

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

В.Д. Дмитрієнко, С.Ю. Леонов, О.Ю. Заковоротний

**НЕЙРОННІ МЕРЕЖІ:
ВІД НАЙПРОСТІШИХ МОДЕЛЕЙ БІОЛОГІЧНИХ СИСТЕМ ДО
СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ**

Навчальний посібник

Затверджено
редакційно-видавничою радою
НТУ "ХПІ",
протокол № 1 від 13.02.2025

Харків,
НТУ "ХПІ"
2025

УДК 004.896

Д 53

Рецензенти:

Аврунін О.Г., д-р техн. наук, проф., Харківський національний університет радіоелектроніки;

Трубчанінова К.А., доктор технічних наук, професор, професор кафедри транспортного зв'язку, Український державний університет залізничного транспорту;

У посібнику викладено основи архітектур та принципи роботи, що найчастіше застосовуються при вирішенні різних завдань нейронних мереж: перцептронів, мереж Хопфілда, Хеммінга, Кохонена, адаптивної резонансної теорії, асоціативної пам'яті тощо. Текст супроводжується великою кількістю прикладів.

Посібник розрахований на студентів, викладачів та фахівців у галузі обчислювальної техніки, а також наукових та практичних працівників, які займаються вирішенням завдань прогнозування, розпізнавання образів та управління.

Іл. 60. Табл. 26. Бібліограф. 68 назв.

Д 53 Дмитрієнко В.Д.

Нейронні мережі: Від найпростіших моделей біологічних систем до систем штучного інтелекту: навчальний посібник / Дмитрієнко В.Д., Леонов С.Ю., Заковоротний О.Ю. – Харків: НТУ "ХПІ", 2025. – 232 с.

© Дмитрієнко В.Д., Леонов С.Ю.

Заковоротний О.Ю., 2025

© НТУ «ХПІ», 2025

СТУП

Навчальний посібник знайомить студентів з основами теорії штучних нейронних мереж та сферами їхнього практичного застосування. Нейронні мережі використовуються в різних галузях науки і техніки для вирішення різноманітних завдань розпізнавання образів, контролю, діагностики та управління практично будь-якими динамічними об'єктами, в тому числі, і з невідомою динамікою, в експертних системах і системах підтримки прийняття рішень, проектуванні та виробництві нейрочіпів і розробці на їх основі суперкомп'ютерів та програмного забезпечення (ПЗ) для них. З одного боку, це ПЗ забезпечує роботу суперкомп'ютерів, роблячи можливим розпаралелювання обчислень нейромережевих алгоритмів та оптимізацію завантаження численних процесорів терафлопних суперкомп'ютерів з трансп'ютерною архітектурою, а, з іншого боку, ПЗ забезпечує вирішення фундаментальних завдань обробки космічних та аерозображень, управління космічними апаратами на інших планетах, дослідження в галузі геному людини та інших живих організмів, вирішення різних систем звичайних диференціальних рівнянь та рівнянь у частинних похідних, що описують різноманітні фізичні та біофізичні процеси тощо. В даний час практично неможливо знайти галузі в науці, техніці, медицині, біології і т.д., де б не застосовувалися в тій чи іншій мірі штучні нейронні мережі [1].

Ідея створення моделей нервових мереж на основі формалізованих нервових клітин для вивчення психологічних функцій мозку вперше була сформульована в 1938 році американським ученим Н. Рашевським в його книзі "Математична біофізика" [6]. Наприкінці тридцятих і на початку сорокових років чиказька школа біофізиків-математиків, що утворилася навколо Рашевського, розпочала перші дослідження мереж з формальних нейронів [3 – 4].

Початком історії нейрокомп'ютерів вважатиметься поява у американському "Бюлетені математичної біофізики" першопрохідницької роботи У.С. Мак-Каллока та У.Х. Піттса "Логічне обчислення ідей, іманентних нервової активності" [5]. У цій роботі американські вчені вперше в історії науки чітко запропонували загальні підходи та конкретні математичні моделі біологічних нейронних мереж та їх компонентів – нейронів, які стали основоположними в теорії штучних нейронних мереж. Як моделі нейронів вони використовували порогові елементи з двома стійкими станами, які згодом отримали назву "нейронів Маккалока – Піттса". Однак завдання розробки моделей та систем на основі порогових елементів виявилися настільки незвичайними та складними, що лише через півтора десятка років з'явилася перша працездатна штучна нейронна мережа (ШНМ) – перцептрон (персептрон) Розенблатта [6]. Перцептрон

Розенблатта викликав величезну цікавість, оскільки продемонстрував можливість створення технічних систем розпізнавання образів на основі моделей мозку людини та вищих тварин. Перші успішні експерименти з перцептроном сприяли залученню до досліджень у галузі нейронних мереж великої кількості дослідників та появи надзвичайно оптимістичних прогнозів щодо створення штучних інтелектуальних систем, що перевершують людину. Однак подальші дослідження перцептронів Розенблатта показали, що їх використання в системах розпізнавання пов'язане із низкою труднощів. Глибокий теоретичний аналіз цих труднощів, виконаний у книзі Мінського та Пейперта "Персептрони", показав, що вони мають принциповий характер, пов'язаний зі структурою цієї нейронної мережі. Виявилося, що така модель мозку людини не може вирішувати багато найпростіших завдань, зокрема, навіть реалізувати логічну функцію "ВИКЛЮЧНЕ АБО". Суворість теоретичних результатів, отриманих для найпростіших перцептронів, авторитет Мінського та його негативний прогноз щодо потенційно можливого прогресу для багат шарових перцептронів викликали розчарування багатьох дослідників та сприяли припиненню урядових субсидій. Після виходу роботи Мінського та Пейперта інтерес до нейронних мереж впав майже до нуля більш ніж на десять років.

Відродження інтенсивних досліджень в області ШНМ почалося лише на початку 80-х років, коли, з одного боку, крім перцептронів вже з'явилися й інші види нейронних мереж та були розроблені принципово нові ідеї навчання нейронних мереж, зокрема, на основі роботи Д. Хопфілда, а, з іншого боку, у зв'язку з відставанням темпів зростання продуктивності послідовних ЕОМ від вимог науки та промисловості зріс інтерес до багатопроцесорних паралельних обчислень і, зокрема, до високопаралельних систем обробки інформації. Нейронні мережі з їх масовим паралелізмом, здатністю до самоорганізації та навчання на прикладах опинилися в центрі уваги багатьох вчених. Реалізовані на інтегральних мікросхемах нового типу – нейромікросхемах, ШНМ несподівано швидко зі стадії попередніх лабораторних досліджень перейшли у сферу уваги фахівців, які вирішують практичні завдання у галузі науки, техніки та економіки. З кінця 80-х років застосування нейронних мереж у Європі, навіть Японії носить вибуховий характер. Якщо 1985 – 1987 роки лише окремі фірми спеціалізувалися на випуску нейронних мереж, то на початку 1990 року їх кількість перевищувало сотню, а в 1992 року лише у США понад 150 різних фірм спеціалізувалися у розробці та виробництві різноманітної нейропродукції.

У 1987 – 1988 роках. у країнах Європейського Економічного Співтовариства (ЄЕС), США та Японії починається фінансування великих наукових проєктів:

1. Програма ЄС "Мозок" (BRAIN = Basic Research in Adaptive Intelligence and Neurocomputing = Базові дослідження в галузі адаптивного інтелекту та нейрообчислень), 1988 – 1996 рр.

2. Програма DARPA на 1987 – 1995 роки. Управління перспективного планування НДР військового застосування Америки.

3. Програма Японії "Кордони людини" на 1988 – 1996 рр.

Великі фінансові вкладення та залучення тисяч висококваліфікованих фахівців призвели до створення численних комерційних продуктів на основі нейромереж, нейрокомп'ютерів та нейротехнологій. Медики використовують нейронні мережі для діагностики та прогнозування лікування, митники – для виявлення наркотиків та пластикових бомб, метеорологи – для прогнозу погоди, фахівці в галузі програмного навчання – для боротьби з комп'ютерними вірусами, банкіри – для оцінки кредитних ризиків, курсів облігацій, акцій та валют, передбачення банкрутств, оптимізації портфелів цінних паперів, забезпечення безпеки транзакцій по пластикових картках, промисловці – для контролю якості виробів та пакування, управління роботами, прокатними станами, атомними станціями, військові – для виявлення, розпізнавання та супроводу цілей, ведення бойових дій, пілотування сильно пошкодженими літаками. Ось лише кілька прикладів.

У медицині ідеальний метод діагностики повинен мати стовідсоткову чутливість, тобто. не пропускати справді хворих людей, і стовідсоткову специфічність, тобто. не відносити до хворих людей, які не мають цього захворювання. Щоб не пропустити хворих людей, намагаються передусім забезпечити стовідсоткову чутливість методу, проте це часто обертається низькою специфічністю методу. Наприклад, рак молочної залози є причиною смерті кожної дев'ятої жінки. Пухлину виявляють в результаті мамографії (первинного рентгенографічного аналізу), а потім проводиться біопсія (аналіз тканини пухлини), оскільки існуючі правила диференціювання злоякісних та доброякісних утворень за даними мамографії сильно завищують відсоток ракових захворювань. Тільки від 10 до 20% діагнозів злоякісних новоутворень підтверджуються хірургічною біопсією.

Нейронна мережа, навчена в університеті Дьюка на восьми особливостях злоякісних новоутворень, на основі яких ставлять діагнози радіологи, здатна вирішувати завдання розпізнавання пухлини з чутливістю близько 100% і специфічністю 59%. Нейросеть клініки Майо (Міннесота) за даними ультразвукових досліджень молочних залоз жінок забезпечує специфічність у сорок відсотків, тоді як у лікарів для тих самих пацієнток вона виявилася близькою до нуля.

Кваліфіковані лікарі при гострому болю в грудях правильно визначають у хворих на інфаркт міокарда в 88% випадків і помилково ставлять цей діагноз у 29% пацієнтів, які страждають на інші захворювання. Нейронна мережа Вільяма Бакста з Каліфорнійського університету в Сан-Дієго має чутливість у 92% при діагностиці інфаркту міокарда і дає лише чотири відсотки помилкових тривог.

Причиною близько 90% всіх вірусних інфекцій програмного забезпечення ЕОМ є віруси завантажувальних секторів. Тому багато фірм приділяють особливу увагу боротьбі саме з вірусами цієї категорії, кількість яких в даний час наближається до 300. Програма AntiVirus корпорації IBM, основана на нейронній технології і навчена на великій кількості нормальних та заражених загрузочних секторів, за рік свого існування самостійно виявила та знашкodiла кілька нових вірусів завантажувальних секторів. Для запобігання розповсюдженню вірусів по мережах в IBM розроблено і нейронну імунну систему, яка використовує програму-приманку, яка першою запускається в мережі, а потім перевіряється на наявність інфекції. За наявності зараження імунна система виявляє сегменти вірусу, аналізує їх коди і потім видаляє їх із заражених файлів.

Японська корпорація Nippon Telegraph and Telephon на основі нейронної мережі, що обробляє інформацію від погодних радарів, що відстежують переміщення хмар у радіусі 240 км, розробила систему прогнозу погоди. Система здатна за кілька хвилин зробити для регіону прогноз випадання опадів на найближчі три години, причому точність прогнозу вдвічі вища, ніж у найкращих систем, що працюють на інших принципах.

Нейронна мережа SNOOPE, розроблена американською компанією SAIC для митних служб великих авіапортів, використовується для виявлення наркотиків та вибухових речовин у багажі авіапасажирів. Розпізнавання здійснюється нейронною мережею на основі аналізу рентгенівських зображень. У найбільшому американському міжнародному аеропорту Лос-Анджелес нова технологія огляду багажу при 95% ймовірності виявлення вибухових речовин зменшила ймовірність помилкових тривог всього до двох відсотків, що втричі нижче, ніж у сучасних альтернативних систем.

Багато Західних банків, фінансові компанії та великі фірми використовують нейронні мережі для прогнозування ринкової кон'юнктури, передбачення курсів валют та акцій, оцінки інвестиційних та кредитних ризиків тощо. У цьому використовуються найрізноманітніші нейромережеві вироби. Такі великі фінансові корпорації, як American Express або банки, як Manhattan Bank, застосовують нейромережеві системи на базі сучасних суперкомп'ютерів CNAPS фірми Adaptive Solutions або SNAP фірми HNC вартістю від п'ятдесяти тисяч доларів. Досить досконалі, але також дорогі нейропакети фірм Nestor і Neuron

Data та ін. поставляються і для звичайних персональних комп'ютерів. Створено і цілий спектр недорогих нейропакетів, наприклад, пакет Rrain Maker фірми California Scientific Software, який багато фахівців вважають одним із найкращих у своєму класі. Базову версію цього пакета вартістю близько тисячі доларів придбало понад 16 тисяч фірм та програмістів.

Американська компанія Nestor, створена ще в 1976 році, постачає нейромережні продукти для зору роботів, розпізнавання мови та відбитків пальців, сортування поштової кореспонденції, читання рукописних підписів на чеках та страхових полісах. Усього за дві-три секунди нейронна система встановлює справжність підпису з ймовірністю понад 0,95.

Для бойових вертольотів та літаків на основі ШНС розроблено системи ведення повітряного бою. У реальному часі нейронна мережа узагальнює такі вхідні дані, як висота, швидкість, відстань між літальними апаратами, наявність боєприпасів і т.д. і вибирає один із класичних бойових маневрів, що використовуються пілотами під час повітряних боїв. Навчання таких нейронних мереж багато в чому нагадує навчання курсантів льотних училищ – їм, як і людям, послідовно пред'являються великі множини сценаріїв повітряних боїв з правильними та неправильними діями льотчиків і вимагають, щоб у процесі навчання вони навчилися завжди правильно визначати необхідний бойовий маневр.

Застосовуються нейронні мережі та в АСУ та інформаційних системах для обробки мовної інформації, автоматичного перекладу, обробки текстів, розв'язання задач кодування та декодування інформації тощо.

Застосовуються нейронні мережі та в Інтернеті для асоціативного пошуку інформації, для адресної реклами та маркетингу в електронній торгівлі, як електронні секретарі та автономні агенти користувача у мережі тощо.

Компанія Autonomu виробляє програмні продукти AGENTWARE, що дозволяють користувачам створювати та застосовувати автономні персональні нейроагенти, які можуть використовуватися як секретарі, що фільтрують інформацію, що надходить по каналах зв'язку. Вони можуть також надсилатися для асоціативного пошуку інформації на місці у віддалених базах даних або використовуватися для постійної фільтрації інформації на сервері провайдера і т.д.

Г л а в а 1

ОСНОВИ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ І НЕРВОВИХ КЛІТОК

1.1. Короткі відомості про мозок людини

За різними оцінками мозок людини містить від 100 до 1000 млрд. нервових клітин – нейронів, кожен з яких у середньому з'єднаний 10 тис. зв'язків з іншими нервовими клітинами. Мільярди нейронів і сотні трильйонів їхніх зв'язків відповідають за підтримку приголомшливо складних функцій людського тіла і за всі загадкові явища, які називаються мисленням, пізнанням, емоціями, почуттями, пам'яттю тощо. Більшість його функцій залишається таємницею розуміння. Незрозуміло навіть, яким чином, складається з величезної кількості клітин, що повільно діють, час реакції кожної з яких не менше кількох мілісекунд, мозок може обробляти величезні обсяги інформації за частки секунд, наприклад, розпізнавати знайомі особи за 100 - 120 мс. У мозку людини існують близько тисячі різних видів нервових клітин, кожен зі своєю характерною формою, проте всі вони в першому наближенні мають приблизно однакову будову. Кожна клітина складається з соми (або тіла), дерева входів (дендритів) та виходів (довгого або короткого аксона з безліччю закінчень). Тіло клітини зазвичай має в діаметрі розміри від 5 до 100 мікрон. Довжина дендритів, що пронизують простір навколо нейрона, сягає одного міліметра. Довжина аксона може бути в сотні разів більшою і досягати кількох метрів. Починається аксон аксонним горбком (потовщенням на тілі клітини, що поступово звужується), а закінчується сотнями або тисячами своїх закінчень на дендритах і на тілах інших нервових клітин синаптичними бляшками або синапсами, потовщеннями, що мають поперечні розміри від одного до декількох мікрон. Через синапс сигнал з аксонного горбка нейрона потрапляє в тіла інших клітин. Ці вхідні сигнали в залежності від виду синапсу можуть бути гальмуючими або збуджуючими активність нервової клітини. Величини сигналів, що генеруються різними синапсами однієї і тієї ж клітини, при однакових величинах вхідних сигналів, у загальному випадку різні і залежать від ефективності або маси синапсу. Ефективність синапсу може змінюватись у процесі функціонування нейронів. Таким чином, сигнали на шляху до аксонного пагорба клітини зважуються, а потім відбувається їх підсумовування. Залежно від величини виваженої суми аксонний горбок або знаходиться у загальмованому стані, або генерує вихідний сигнал, який через гілки аксона та їх синапси трансформується у вхідні сигнали для сотень або тисяч інших нейронів, вихідні сигнали яких у свою чергу можуть збуджувати або гальмувати тисячі інших нервових клітин.

Описана структура та функціонування нейрона дуже схематичні. Це видно навіть при поверхневому розгляді найзагальніших механізмів переробки інформації в окремих нейронах на молекулярному рівні, які включають в себе:

- гігантський паралелізм переробки інформації, в якому одночасно беруть участь мільйони молекул;

- складні нелінійні оборотні динамічні процеси на молекулярному та надмолекулярному рівнях;

- високу інформаційну складність операцій, що виконуються на молекулярному рівні, що пояснюється особливостями функціонування молекул ферментів, для яких характерна мінливість молекулярного кістяка, можливість існування кількох стійких станів, перебудова структури під дією хімічних та фізичних факторів, а також тактильність (виду ключ – замок) характеру переробки інформації молекулами ферменту;

- здатність до навчання, відбору та еволюції молекулярних компонентів переробки інформації;

- високу ефективність переробки інформації, яка за величиною тепловиділення білкових молекул в $10^6 - 10^7$ разів менше, ніж у найкращих напівпровідникових інтегральних схем.

В даний час багато вчених вважають, що ще дуже далеко до повного розуміння процесів функціонування окремих нейронів. Хоча ще кілька років тому здавалося, що їхня структура та функції добре вивчені, проте майже кожен новий рік приносить цінну інформацію про біохімію та електрохімічну поведінку нервових клітин у напрямку розкриття раніше невідомих рівнів складності. У зв'язку з цим слід зазначити, що хоча перші штучні нейронні мережі і з'явилися в результаті вивчення мозку, проте подібності між ними та мозком насправді дуже мало та можливості їх дуже обмежені. Тому в багатьох випадках розробники штучних нейронних мереж змушені виходити за межі біологічної правдоподібності і створювати штучні мережі з корисними властивостями, але неможливими в живій природі або такими, що потребують занадто грубих припущень про функціонування мозку або його анатомію. Це ще більше розширює прірву між мозком та його моделями. Все ж таки навіть ці скромні наслідування організації мозкової діяльності принесли істотні результати: штучні нейронні мережі подібно до мозку здатні навчатися на безлічі прикладів, вирішувати завдання, спираючись на спотворену, неповну і внутрішньо суперечливу інформацію, робити абстрактні висновки, розпізнавати мову і візуальні зображення. Все це сприяє широкому колу досліджень у галузі штучних нейронних мереж, незважаючи на те, що і на рівні структур, і на рівні окремих нейронів вони сильно відрізняються від біологічного прототипу.

1.2. Формальні нейрони штучних нейронних мереж

Детальна модель навіть окремої нервової клітини, що має тисячі зв'язків з іншими клітинами, надзвичайно складна, тому при моделюванні нейронних мереж використовується зазвичай набагато простіший процесорний елемент, зображений на рис. 1.1. На його входи надходить вектор $X = (x_1, \dots, x_n)$ вхідних сигналів, що є вихідними сигналами інших нейронів, а також одиничний сигнал зміщення. Всі вхідні сигнали, включаючи сигнал зміщення, множаться на вагові коефіцієнти своїх зв'язків і підсумовуються:

$$S = \sum_{i=1}^n x_i w_i + w_0, \quad (1.1)$$

де S – сумарний вхідний сигнал; $w_i (i = \overline{1, n})$ – вагові коефіцієнти зв'язків вхідних сигналів $x_1, \dots, x_n$; w_0 – ваговий коефіцієнт зв'язку сигналу зміщення.

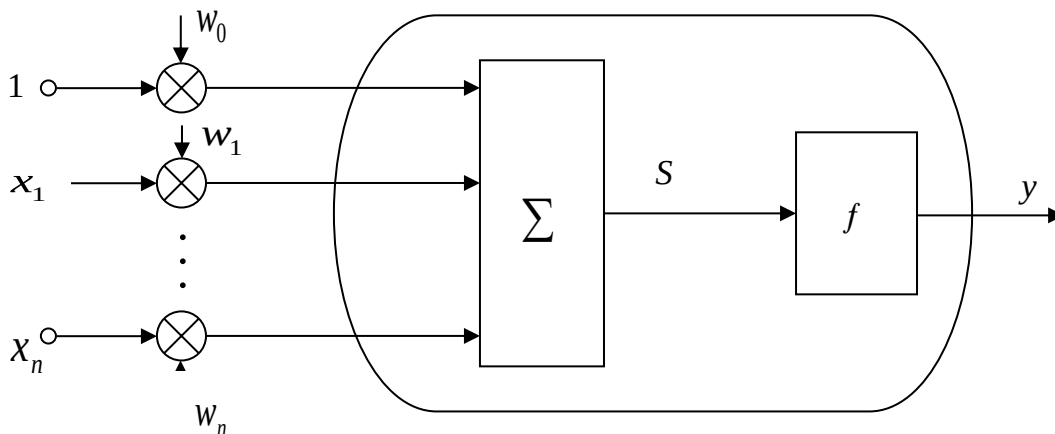


Рис. 1.1. Процесорний елемент, що використовується у звичайних нейромережах

Отриманий сигнал S надходить на вхід блоку, що реалізує функцію f активації нейрона. Типовими функціями активації є бінарна

$$y = \begin{cases} 1, & \text{якщо } S > 0, \\ 0, & \text{якщо } S \leq 0 \end{cases} \quad (1.2)$$

або біполярна

$$y = \begin{cases} 1, & \text{якщо } S > 0, \\ -1, & \text{якщо } S \leq 0. \end{cases} \quad (1.3)$$

Багато авторів при описі моделі нейрона використовують не сигнал усунення, а поріг нейрона θ , що призводить до еквівалентної моделі елемента. У цьому випадку вирази (1.2) та (1.3) приймають відповідно вигляд:

$$y = \begin{cases} 1, & \text{якщо } S > \theta, \\ 0, & \text{якщо } S \leq \theta, \end{cases} \quad (1.4)$$

$$y = \begin{cases} 1, & \text{якщо } S > \theta, \\ -1, & \text{якщо } S \leq \theta, \end{cases} \quad (1.5)$$

де

$$S = \sum_{i=1}^n w_i x_i. \quad (1.6)$$

Графічне зображення бінарної та біполярної функцій активації для цього випадку представлено на рис. 1.2, а і 1.2, б.

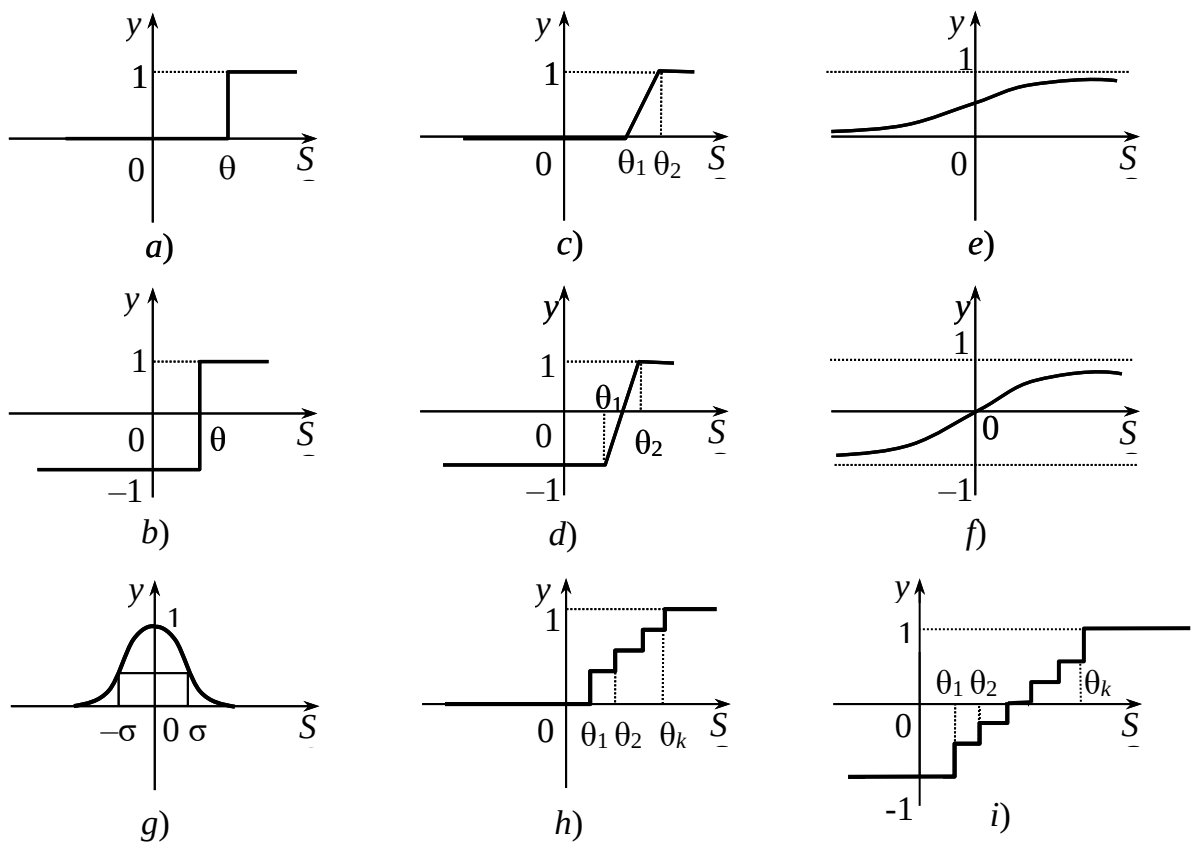


Рис. 1.2. Функції активації нейронів

Зі зіставлення виразів (1.1) – (1.3) і (1.4) – (1.6) випливає, що кожному значенню порога θ нейрона може бути поставлений у відповідність ваговий коефіцієнт w_0 зв'язку сигналу зміщення та навпаки.

Найрідше використовуються лінійні бінарні або біполярні функції активації (рис. 1.2, *c* и 1.2, *d*):

$$y = \begin{cases} -a & \text{при } S < \theta_1, \\ kS + a_0 & \text{при } \theta_1 \leq S \leq \theta_2, \\ 1 & \text{при } S > \theta_2, \end{cases} \quad (1.7)$$

де a дорівнює нулю для бінарних вихідних сигналів нейронів і рівно мінус одиниці для біполярних сигналів; k, a_0 – постійні коефіцієнти.

Крім наведених функцій активації використовуються також такі нелінійні функції активації:

бінарна сигмоїдальна або логічна сигмоїдальна (рис. 1.2, *e*):

$$y = \frac{1}{1 + e^{-\tau S}}, \quad (1.8)$$

де τ – постійний коефіцієнт;

біполярна сигмоїдальна (рис.1.2, *f*):

$$y = \frac{2}{1 + e^{-\tau S}} - 1, \quad (1.9)$$

радіально-симетрична (рис.1.2, *g*):

$$y = e^{-\frac{S^2}{\tau^2}}, \quad (1.10)$$

K -значна бінарна (рис.1.2, *h*):

$$y = \begin{cases} 0 & \text{при } S < \theta_1, \\ 1/(K-1) & \text{при } \theta_1 \leq S < \theta_2, \\ 2/(K-1) & \text{при } \theta_2 \leq S < \theta_3, \\ \dots & \dots \\ \frac{K-2}{K-1} & \text{при } \theta_{k-1} \leq S < \theta_k, \\ 1 & \text{при } S \geq \theta_k, \end{cases} \quad (1.11)$$

K -значна біполярна (рис.1.2, і):

$$y = \begin{cases} -1 & \text{при } S < \theta_1, \\ -1 + 2/(K-1) & \text{при } \theta_1 \leq S < \theta_2, \\ -1 + 4/(K-1) & \text{при } \theta_2 \leq S < \theta_3, \\ \dots & \dots \\ -1 + \frac{2(K-2)}{K-1} & \text{при } \theta_{k-1} \leq S < \theta_k, \\ 1 & \text{при } S \geq \theta_k. \end{cases} \quad (1.12)$$

Наведені моделі штучних нейронів ігнорують багато відомих властивостей біологічних прототипів. Наприклад, вони не враховують тимчасові затримки нейронів, ефекти частотної модуляції, локального збудження і пов'язані з ними явища підпорогової тимчасової і просторової сумації, коли клітина збуджується не одночасно імпульсами, що прийшли, а послідовностями збуджуючих сигналів, що надходять через короткі проміжки часу. Не враховуються також періоди абсолютної рефрактерності, під час яких нервові клітини неможливо збудити, тобто. як би клітини мають нескінченно великий порог збудження, який потім за кілька мілісекунд після проходження сигналу знижується до нормального рівня. Цей список відмінностей, які багато біологів вважають вирішальними, легко продовжити, проте штучні нейронні мережі все ж таки виявляють низку цікавих властивостей, характерних для біологічних прототипів.

1.3. Найпростіші нейронні мережі. Правило Хебба

Штучні нейронні мережі, призначені на вирішення різноманітних конкретних завдань, можуть містити від кількох нейронів до тисяч і навіть мільйонів елементів. Однак, вже окремий нейрон (рис. 1.1) з біполярною або бінарною функцією активації може бути використаний для вирішення простих завдань розпізнавання та класифікації зображень. Вибір біполярного (1, -1) або бінарного (1, 0) подання сигналів у нейромережах здійснюється виходячи з розв'язуваної задачі і в багатьох випадках він рівноцінний. Є спектр завдань, у

яких бінарне кодування сигналів зручніше, проте загалом біполярне подання інформації краще [7].

Оскільки вихідний сигнал у двійкового нейрона (рис 1.1) приймає лише два значення, то нейрон можна використовувати для класифікації зображень, що пред'являються, на два класи.

Нехай ϵ множина M зображень, для яких відома коректна класифікація на два класи $X^1 = \{X^{11}, X^{12}, \dots, X^{1q}\}$, $X^2 = \{X^{21}, X^{22}, \dots, X^{2p}\}$, $X^1 \cup X^2 = M$, $X^1 \cap X^2 = \emptyset$, і нехай першому класу X^1 відповідає вихідний сигнал $y = 1$, а класу X^2 – сигнал $y = -1$. Якщо, наприклад, подано деяке зображення $X^\alpha = (X_1^\alpha, \dots, X_n^\alpha)$, $X^\alpha \in M$ та його зважена сума вхідних сигналів перевищує нульове значення:

$$S = \sum_{i=1}^n x_i^\alpha w_i + w_0 > 0,$$

то вихідний сигнал $y = 1$ і, отже, вхідне зображення X^α належить класу X^1 . Якщо $S \leq 0$, то $y = -1$ і пред'явлене зображення належить другому класу.

Можливе використання окремого нейрона та для виділення з множини класів $M = \{X^1 = \{X^{11}, \dots, X^{1k}\}, \dots, X^i = \{X^{i1}, \dots, X^{iq}\}, \dots, X^p = \{X^{p1}, \dots, X^{pm}\}\}$ зображень єдиного класу X^i . У цьому випадку вважають, що один із двох можливих вихідних сигналів нейрона (наприклад, 1) відповідає класу X^i , а другий – усім іншим класам. Тому, якщо вхідне зображення X^α призводить до появи сигналу $y = 1$, то $X^\alpha \in X^i$, якщо $y = -1$ (або $y = 0$, якщо використовується бінарне кодування), то це означає, що пред'явлене зображення не належить класу, що виділяється.

Система розпізнавання на основі єдиного нейрона поділяє весь простір можливих рішень на дві області за допомогою гіперплощини

$$x_1 w_1 + x_2 w_2 + \dots + x_n w_n + w_0 = 0.$$

Для двовимірних вхідних векторів кордоном між двома класами зображень є пряма лінія: вхідні вектори, розташовані вище цієї прямої, належать до одного класу, а нижче – до іншого.

Для адаптації, налаштування або навчання ваг зв'язків нейрона може використовуватися кілька методів. Розглянемо один із них, який отримав назву “правило Хебба” [7]. Хебб, досліджуючи механізми функціонування центральної нервової системи, припустив, що навчання відбувається шляхом посилення зв'язків між нейронами, активність яких збігається за часом. Хоча в біологічних

системах це припущення виконується далеко не завжди і не вичерпує всіх видів навчання, проте під час навчання одношарових нейромереж з біполярними сигналами воно дуже ефективне.

Відповідно до правила Хебба, якщо пред'явлене біполярне зображення $X = (x_1, \dots, x_n)$ відповідає неправильний вихідний сигнал y , то ваги w_i ($i = \overline{1, n}$) зв'язків нейрона адаптуються за формулою

$$w_i(t+1) = w_i(t) + x_i y, \quad i = \overline{0, n}, \quad (1.13)$$

де $w_i(t)$, $w_i(t+1)$ відповідно вага i -го зв'язку нейрона до та після адаптації; x_i ($i = \overline{1, n}$) – компоненти вхідного зображення; $x_0 \equiv 1$ – сигнал переміщення; y – вихідний сигнал нейрона.

У більш повній і строгій формі алгоритм налаштування ваги зв'язків нейрона з використанням правила Хебба виглядає наступним чином:

Крок 1. Задається множина $M = \{(X^1, t^1), \dots, (X^m, t^m)\}$, що складається з пар (вхідне зображення $X^k = (x_1^k, \dots, x_n^k)$, необхідний вихідний сигнал нейрона t^k), $k = \overline{1, m}$). Ініціюються ваги зв'язків нейрона:

$$w_i = 0, \quad i = \overline{0, n}.$$

Крок 2. Для кожної пари (X^k, t^k) , $k = \overline{1, m}$ доки не дотримуються умови зупинки, виконуються кроки 3 – 5.

Крок 3. Ініціюється множина входів нейрона:

$$x_0 = 1, \quad x_i = x_i^k, \quad i = \overline{1, n}.$$

Крок 4. Ініціюється вихідний сигнал нейрона: $y = t^k$.

Крок 5. Коригуються ваги зв'язків нейрона за правилом

$$w_i(\text{new}) = w_i(\text{old}) + x_i y, \quad i = \overline{0, n}.$$

Крок 6. Перевірка умов зупинки.

Для кожного вхідного зображення X^k розраховується відповідний йому вихідний сигнал y^k :

$$y^k = \begin{cases} 1, & \text{якщо } S^k > 0, \\ -1, & \text{якщо } S^k \leq 0, \end{cases} \quad k = \overline{1, m},$$

де

$$S^k = \sum_{i=1}^n x_i^k w_i + w_0.$$

Якщо вектор $(y^1, \dots, y^m)$ розрахованих вихідних сигналів дорівнює вектору $(t^1, \dots, t^m)$ заданих сигналів нейрона, тобто. кожному вхідному зображенню відповідає заданий вихідний сигнал, то обчислення припиняються (перехід до кроку 7), якщо ж $(y^1, \dots, y^m) \neq (t^1, \dots, t^m)$, то перехід до кроку 2 алгоритма.

Крок 7. Зупинка.

Приклад 1.1. Нехай потрібно навчити біполярний нейрон розпізнавання зображень X^1 , і X^2 , наведених на рис. 1.3.

X^1			X^2		
1	2	3	1	2	3
4	5	6	4	5	6
7	8	9	7	8	9

Рис. 1.3. Вхідні зображення

При цьому потрібно, щоб зображенню X^1 відповідав вихідний сигнал нейрона "+1", а зображенню X^2 – сигнал "-1".

Застосування алгоритму Хебба дає такі результати:

Крок 1. Задається множина

$$M = \{(X^1 = (1, -1, 1, 1, 1, 1, -1, -1, 1), 1), \\ (X^2 = (1, 1, 1, 1, -1, 1, 1, -1, 1), -1)\};$$

ініціюються ваги зв'язків нейрона: $w_i = 0, i = \overline{0, 9}$.

Крок 2. Для кожної з двох пар $(X^1, 1), (X^2, -1)$, виконуються кроки 3 – 5.

Крок 3. Ініціюється множина входів нейрона для зображення першої пари: $x_0 = 1, x_i = x_i^1, i = \overline{0, 9}$.

Крок 4. Ініціюється вихідний сигнал нейрона для зображення першої пари: $y = t^1 = 1$.

Крок 5. Коригуються ваги зв'язків нейрона за правилом Хебба

$$w_i = w_i + x_i^1 y \quad (i = \overline{0, n}):$$

$$w_0 = w_0 + x_0 y = 0 + 1 \cdot 1 = 1;$$

$$w_1 = w_1 + x_1^1 y = 0 + 1 \cdot 1 = 1;$$

$$w_1 = w_3 = w_4 = w_5 = w_6 = w_9 = 1;$$

$$w_2 = w_2 + x_2^1 y = 0 + (-1) \cdot 1 = -1;$$

$$w_7 = w_8 = -1.$$

Крок 3. Ініціюється множина входів нейрона для зображення X^2 другої пари: $x_0 = 1, x_i = x_i^2, i = \overline{0, 9}$.

Крок 4. Ініціюється вихідний сигнал нейрона для зображення другої пари (X^2, t^2) :

$$y = t^2 = -1.$$

Крок 5. Коригуються ваги зв'язків нейрона:

$$w_0 = w_0 + x_0 y = 1 + 1 \cdot (-1) = 0;$$

$$w_1 = w_1 + x_1^2 y = 1 + 1 \cdot (-1) = 0;$$

$$w_3 = w_4 = w_6 = w_9 = 0;$$

$$w_2 = w_2 + x_2^2 y = -1 + 1 \cdot (-1) = -2;$$

$$w_7 = -2;$$

$$w_5 = w_5 + x_5^2 y = 1 + (-1) \cdot (-1) = 2;$$

$$w_8 = w_8 + x_8^2 y = -1 + (-1) \cdot (-1) = 0.$$

Крок 6. Перевіряються умови зупинки.

Розраховуються вхідні та вихідний сигнали нейрона при пред'явленні зображення X^1 :

$$S^1 = \sum_{i=1}^9 x_i^1 w_i + w_0 = 1 \cdot 0 + (-1) \cdot (-2) + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 2 + 1 \cdot 0 + (-1) \cdot (-2) + (-1) \cdot 0 + 1 \cdot 0 + 0 = 6,$$

$$y^1 = 1, \text{ так як } S^1 > 0.$$

Розраховуються вхідний та вихідний сигнали нейрона при пред'явленні зображення X^2

$$S^2 = \sum_{i=1}^9 x_i^2 w_i + w_0 = 1 \cdot 0 + 1 \cdot (-2) + 1 \cdot 0 + 1 \cdot 0 + (-1) \cdot 2 + 1 \cdot 0 + 1 \cdot (-2) + (-1) \cdot 0 + 1 \cdot 0 + 0 = -6,$$

$$y^2 = -1, \text{ так як } S^2 < 0.$$

Оскільки вектор $(y^1, y^2) = (1, -1)$ дорівнює вектору (t^1, t^2) , то обчислення припиняються, оскільки мета досягнута – нейрон правильно розпізнає задані зображення.

Основна ідея правила (1.13) – посилювати зв'язки, які з'єднують нейрони з однаковою за часом активністю, та послаблювати зв'язки, що з'єднують елементи з різною активністю, може бути використана при налаштуванні нейромереж з бінарними елементами. Правило Хебба (1.13) для одношарових бінарних мереж можна записати як:

$$w_i(t+1) = w_i(t) + \Delta w_i, \quad (1.14)$$

де

$$\Delta w_i = \begin{cases} 1, & \text{якщо } x_i y = 1, \\ 0, & \text{якщо } x_i = 0, \\ -1, & \text{якщо } x_i \neq 0 \text{ та } y = 0. \end{cases} \quad (1.15)$$

Приклад 1.2. Нехай потрібно навчити бінарний нейрон розпізнаванню зображень X^1 і X^2 приклада 1.1. При цьому зображенню X^1 нехай відповідає вихідний сигнал нейрона "+1", а зображенню X^2 – "0". Застосування правила Хебба у разі дає такі результати:

Крок 1. Задається множина

$$M = \{(X^1 = (1, 0, 1, 1, 1, 1, 0, 0, 1), 1), (X^2 = (1, 1, 1, 1, 0, 1, 1, 0, 1), 0)\},$$

та ініціюються ваги зв'язків нейрона $w_i = 0, i = \overline{0, 9}$.

Крок 2. Для пар $(X^1, 1), (X^2, 0)$, виконуються кроки 3 – 5.

Крок 3. Ініціюється множина входів нейрона елементами зображення X^1 :

$$x_0 = 1, \quad x_i = x_i^1, \quad i = \overline{0, 9}.$$

Крок 4. Ініціюється вихідний сигнал нейрона для зображення X^1 :

$$y = t^1 = 1.$$

Крок 5. Коригуються ваги зв'язків нейрона за допомогою співвідношень (1.14), (1.15):

$$w_0 = w_0 + \Delta w_0 = 0 + 1 = 1;$$

$$w_1 = w_1 + \Delta w_1 = 0 + 1 = 1;$$

$$w_1 = w_3 = w_4 = w_5 = w_6 = w_9 = 1;$$

$$w_2 = w_2 + \Delta w_2 = 0 + 0 = 0;$$

$$w_2 = w_7 = w_8 = 0.$$

Крок 3. Ініціюється множина входів нейрона елементами зображення X^2 :

$$x_0 = 1, x_i = x_i^2, i = \overline{0, 9}.$$

Крок 4. Ініціюється вихідний сигнал нейрона для зображення X^2 :

$$y = t^2 = 0.$$

Крок 5. Коригуються ваги зв'язків нейрона за допомогою співвідношень (1.14), (1.15):

$$w_0 = w_0 + \Delta w_0 = 1 + (-1) = 0;$$

$$w_1 = w_1 + \Delta w_1 = 1 + (-1) = 0;$$

$$w_1 = w_3 = w_4 = w_6 = w_9 = 0;$$

$$w_2 = w_2 + \Delta w_2 = 0 + (-1) = -1;$$

$$w_5 = w_5 + \Delta w_5 = 1 + 0 = 1;$$

$$w_7 = w_2 = -1;$$

$$w_8 = w_8 + \Delta w_8 = 0 + 0 = 0.$$

Крок 6. Перевірка умов зупинки.

Розраховуються вхідні та вихідні сигнали нейрона при пред'явленні зображень X^1, X^2 :

$$S^1 = \sum_{i=1}^9 x_i^1 w_i + w_0 = 1 \cdot 0 + 0 \cdot (-1) + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot (-1) + \\ + 0 \cdot 0 + 1 \cdot 0 + 0 = 1,$$

$$y^1 = 1, \text{ так як } S^1 > 0.$$

$$S^2 = \sum_{i=1}^9 x_i^2 w_i + w_0 = 1 \cdot 0 + 1 \cdot (-1) + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 1 \cdot (-1) + \\ + 0 \cdot 0 + 1 \cdot 0 = -2;$$

$$y^2 = -1, \text{ так як } S^2 < 0.$$

Оскільки вектор $(y_1, y_2) = (1, 0)$ дорівнює заданому вектору $(t_1, t_2) = (1, 0)$, то мета досягнута та обчислення припиняються.

Використання групи з m біполярних або бінарних нейронів $A_1, \dots, A_m$ (рис. 1.4) дозволяє суттєво розширити можливості нейронної мережі та розпізнавати до 2^m різних зображень. Щоправда, застосування цієї мережі для розпізнавання 2^m (або близьких до 2^m чисел) різних зображень може призводити до нерозв'язних проблем адаптації ваги зв'язків нейромережі. Тому часто рекомендують використовувати цю архітектуру для розпізнавання лише m різних зображень, задаючи кожному з них одиничний вихід тільки на виході одного A -елемента (виходи інших при цьому повинні набувати значення "-1" для біполярних нейронів або "0" – для бінарних).

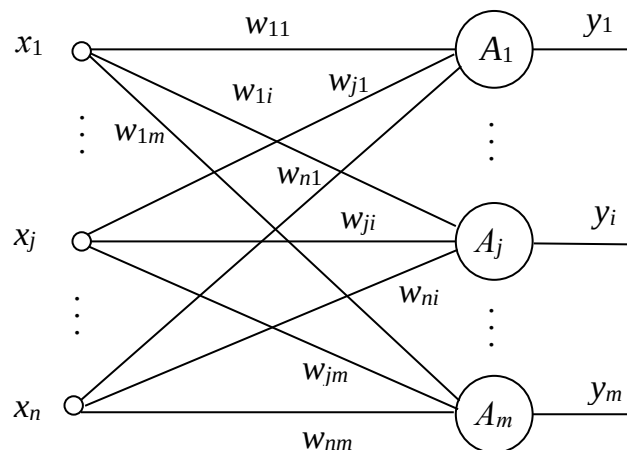


Рис 1.4. Нейронна мережа з m елементів

Одношарова нейронна мережа з двійковими нейронами може бути навчена за допомогою алгоритму на основі правила Хебба. Для біполярного представлення сигналів можливе навчання нейромережі за допомогою наступного алгоритму:

Крок 1. Задається множина $M = \{(X^1, t^1), \dots, (X^m, t^m)\}$, що складається з пар (вхідне зображення $X^k = (x_1^k, \dots, x_n^k)$, необхідний вихідний сигнал нейрона t^k , $k = \overline{1, m}$). Ініціюються ваги зв'язків нейрона:

$$w_{ji} = 0, \quad j = \overline{1, n}, \quad i = \overline{1, m}.$$

Крок 2. Кожна пара (X^k, t^k) , перевіряється на правильність реакції нейронної мережі на вхідне зображення. Якщо отриманий вихідний вектор мережі $(y_1^k, \dots, y_m^k)$, відрізняється від заданого $t^1 = (t_1^k, \dots, t_m^k)$, то виконують кроки 3 – 5.

Крок 3. Ініціюється множина входів нейронів: $x_0 = 1, x_j = x_j^k, j = \overline{1, n}$.

Крок 4. Ініціюються вихідні сигнали нейронів: $y_i = t_i^k, i = \overline{0, m}$.

Крок 5. Коригуються ваги зв'язків нейронів за правилом:

$$w_{ji}(\text{new}) = w_{ji}(\text{old}) + x_j y_i, \quad j = \overline{0, n}, \quad i = \overline{0, m}.$$

Крок 6. Перевіряються умови зупинки, тобто. правильність функціонування мережі за умови пред'явлення кожного вхідного зображення. Якщо умови не виконуються, то перехід до кроку алгоритму 2, інакше – припинення обчислень (перехід до кроку 7).

Крок 7. Зупинка.

1.4. Адалін. Навчання нейронних мереж за допомогою дельта-правила

Часто нейронні мережі з однієї і тієї ж базовою архітектурою носять різні назви. Різниця в назвах, зазвичай, пов'язана з різними методами навчання мереж. Мережа (або окремий нейрон) із входами $x_0, x_1, \dots, x_n$, зображена на рис. 1.5, залежно від методу навчання, що використовується, може називатися мережею Хебба, одношаровим перцептроном або адаліном (*Adaline* – адаптивний лінійний нейрон).

Відмінна риса адаліну – використання для навчання дельта-правила. Це – досить загальний метод навчання одношарових нейронних мереж з багатьма вихідними елементами. Адалін – окремий випадок нейронної мережі, коли є лише один вихідний нейрон. Зазвичай адалін використовує біполярні вхідні та вихідні сигнали, але можливе і бінарне кодування сигналів. Навчальне правило мінімізує квадрат різниці між вхідним сигналом елемента

$$U_{\text{inp.}A} = \sum_{i=1}^n x_i w_i + w_0 \quad (1.16)$$

та необхідним значенням $U_{\text{out.}A}$ сигналу на виході:

$$E = (U_{out.A} - U_{inp.A})^2. \quad (1.17)$$

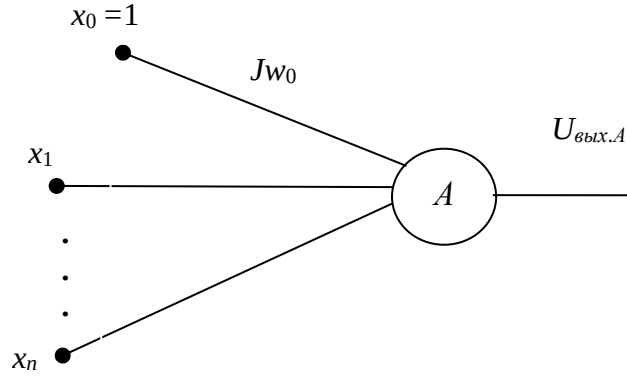


Рис. 1.5. Адалін

Функція E залежить від усіх ваг w_j ($j = \overline{0, n}$) нейрона. Градієнт функції E визначається частинними похідними $\frac{\partial E}{\partial w_j}$ по кожній з ваг і вказує напрямок найбільшого зростання E . Протилежний градієнту напрямок визначає найбільш швидке зменшення мінімізованого квадрата різниці. При адаптації окремої ваги він повинен змінюватися відповідно до значення похідної $-\frac{\partial E}{\partial w_j}$. Використовуючи вирази (1.17) та (1.16) для конкретного зображення $X^k = (x_1^k, \dots, x_n^k)$ маємо:

$$\frac{dE}{dw_j} = -2(U_{out.A}^k - U_{inp.A}^k) \frac{dU_{inp.A}^k}{dw_j} = -2(U_{out.A}^k - U_{inp.A}^k)x_j^k. \quad (1.18)$$

Дельта-правило для налаштування j -ї ваги нейрона (для кожного зображення x^k , $k = \overline{1, p}$) має вигляд

$$\Delta w_j^k = \alpha(U_{out.A}^k - U_{inp.A}^k)x_j^k, \quad j = \overline{0, n}. \quad (1.19)$$

Адалін послідовно навчається на всій множині $M = \{X^1, U_{out}^1\}, \dots, (X^p, U_{out}^p)\}$ пар (вхідне зображення, необхідний вихідний сигнал), мінімізуючи помилку для всієї множини M :

$$E = \sum_{k=1}^p E_k = \sum_{k=1}^p (U_{out.A} - U_{inp.A})^2. \quad (1.20)$$

Важливу роль у правильному і швидкому визначенні ваг зв'язків грає навчальний коефіцієнт α . При великих значеннях α процес може не сходитися чи сходитися з неправильними значеннями (приклад 1.3). При малих значеннях α навчання може дуже затягуватися. Вибір навчального коефіцієнта в загальному випадку непросте завдання. Для адаліну при практичних обчисленнях рекомендується вибирати α таким чином, щоб виконувались нерівності

$$0,1 \leq n \cdot \alpha \leq 1,0, \quad (1.21)$$

де n – число входів нейрона, включаючи вхід сигналу зміщення.

Навчальний алгоритм адаліну наступний:

Крок 1. Задається множина що складається з пар (вхідне зображення

$X^k = (x_1^k, \dots, x_n^k)$, необхідний вихідний сигнал нейрона $U_{out.A}^k$), $k = \overline{1, p}$.

Невеликими випадковими значеннями ініціюються ваги зв'язків нейрона: w_i , $i = \overline{0, n}$, визначається навчальний коефіцієнт α .

Крок 2. Поки не дотримуються умови зупинки кроку 7, виконуються кроки 3 – 7 ітераційного уточнення ваг $w_0, w_1, w_2, \dots$ з послідовним використанням всіх елементів множини M .

Крок 3. Для кожної навчальної пари $(X^k, U_{out.A}^k)$ $k = \overline{1, p}$ виконуються кроки 4 – 6.

Крок 4. Активується множина входів адаліну:

$$x_0 = 1, x_j = x_j^k, j = \overline{1, n}.$$

Крок 5. Обчислюється вхідний сигнал адаліну:

$$U_{inp.A}^k = \sum_{j=1}^n x_j w_j + w_0.$$

Крок 6. Адаптуються вагові коефіцієнти:

$$w_0(new) = w_0(old) + \alpha(U_{out.A}^k - U_{inp.A}^k),$$

$$w_j(new) = w_j(old) + \alpha(U_{out.A}^k - U_{inp.A}^k)x_j^k, j = \overline{1, n}.$$

Крок 7. Перевіряється умова зупинки.

Обчислення припиняються, якщо в процесі виконання ітераційної процедури всі ваги змінюються менше деякої попередньо заданої малої величини ε , в інакше – перехід до кроку 2.

Крок 8. Зупинка.

Приклад 1.3. Зазвичай адалін передбачає використання біполярних сигналів, проте дельта-правило справедливе і при бінарних вхідних сигналах та біполярному вихідному. Функція "І" визначається наступними чотирма елементами множини M , заданими в табл. 1.1. Оскільки вхідні вектори в аналізованому прикладі мають лише дві компоненти, то адалін відповідно до виразу (1.16) матиме лише три навчальні вагові коефіцієнти: w_0, w_1, w_2 . З виразу (1.20) випливає, що ці коефіцієнти мають мінімізувати функцію

$$E = \sum_{k=1}^4 (U_{out.A}^k - (x_1^k w_1 + x_2^k w_2 + w_0))^2, \quad (1.22)$$

де $(x_1^k w_1 + x_2^k w_2 + w_0)$ – вхідний сигнал адаліну, що відповідає вектору $X^k = (x_1^k, x_2^k)$; $U_{out.A}^k$ – необхідний вихідний сигнал нейрона для вхідного вектора X^k .

Таблиця 1.1. Навчальна множина M для функції "І"

Вхідний вектор	Компоненти вхідного вектора (x_1^k, x_2^k)	Необхідний вихідний сигнал $U_{out.}^k$
X^1	(1, 1)	1
X^2	(1, 0)	-1
X^3	(0, 1)	-1
X^4	(0, 0)	-1

Визначимо ваги зв'язків при різних значеннях навчального коефіцієнта α .

Крок 1. Задається множина $M = \{(X^1 = (1, 1), 1), (X^2 = (1, 0), -1), (X^3 = (0, 1), -1), (X^4 = (0, 0), -1)\}$ та ініціюються випадковими значеннями з інтервалу $[-0,5, 0,5]$ ваги зв'язків:

$$w_0 = -0,38246, \quad w_1 = 0,37999, \quad w_2 = 0,42999.$$

Встановлюють навчальний коефіцієнт α та константу ε , що входить до умов припинення ітерацій: $\alpha = 0,5$; $\varepsilon = 0,003$.

Крок 2. Початок ітераційного уточнення ваг w_0, w_1, w_2 . з послідовним використанням всіх елементів множини M та перевіркою умов досягнення заданої точності на кроці 7.

Крок 3. Для навчальної пари $(X^1 = (1, 1), 1)$, виконуються кроки 4 – 6.

Крок 4. Активується множина входів адаліну: $x_0 = 1, x_1 = 1, x_2 = 1$.

Крок 5. Обчислюється вхідний сигнал адаліну для вектора X^1 :

$$U_{inp.A}^1 = x_1^1 w_1 + x_2^1 + w_0 = 1 \cdot 0,37999 + 1 \cdot 0,42999 - 0,38246 = 0,42752.$$

Крок 6. Адаптуються ваги зв'язків:

$$\begin{aligned} w_0(new) &= w_0(old) + \alpha(U_{out.A}^1 - U_{inp.A}^1) = \\ &= 0,38246 + 0,5 \cdot (1 - (0,42752)) = 0,33130, \\ w_1(new) &= w_1(old) + \alpha(U_{out.A}^1 - U_{inp.A}^1)x_1^1 = \\ &= 0,37999 + 0,5 \cdot (1 - (-0,42752)) \cdot 1 = 1,09375, \\ w_2(new) &= w_2(old) + \alpha(U_{out.A}^1 - U_{inp.A}^1)x_2^1 = \\ &= 0,42999 + 0,5 \cdot (1 - (-0,42752)) \cdot 1 = 1,14375. \end{aligned}$$

Крок 3. Для навчальної пари $(X^2 = (1, 0), 1)$ виконуються кроки 4 – 6.

Крок 4. Активується множина входів адаліну:

$$x_0 = 1, x_1 = 1, x_2 = 0.$$

Крок 5. Обчислюється вхідний сигнал адаліну для вектора X^2 :

$$\begin{aligned} U_{inp.A}^2 &= x_1^2 w_1 + x_2^2 w_2 + w_0 = \\ &= 0 \cdot 1,09375 + 0,33130 + 1 \cdot 1,09375 = 1,42505. \end{aligned}$$

Крок 6. Адаптуються ваги зв'язків:

$$\begin{aligned} w_0 &= 0,33130 + 0,5 \cdot (-1 - 1,42505) = -0,881225, \\ w_1 &= 1,09375 + 0,5 \cdot (-1 - 1,42505) \cdot 1 = -0,118775, \\ w_2 &= 1,14375 + 0,5 \cdot (-1 - 1,42505) \cdot 0 = 1,143750. \end{aligned}$$

Крок 3. Для навчальної пари ($X^3 = (0, 1), -1$) виконуються кроки 4 – 6.

Крок 4. Активується множина входів адаліну:

$$x_0 = 1, \quad x_1 = 0, \quad x_2 = 0.$$

Крок 5. Обчислюється вхідний сигнал:

$$U_{inp.A}^3 = -0,118775 + 1,143750 \cdot 1 - 0,881225 = 0,262525.$$

Крок 6. Адаптуємо ваги зв'язків:

$$w_0 = -0,881225 + 0,5 \cdot (-1 - 0,262525) = -1,512488,$$

$$w_1 = -0,118775,$$

$$w_2 = 1,143750 + 0,5 \cdot (-1 - 0,262525) = 0,512488.$$

Крок 3. Для навчальної пари ($X^4 = (0, 0), -1$) виконуються кроки 4 – 6.

Крок 4. Активується множина входів адаліну:

$$x_0 = 1, \quad x_1 = 0, \quad x_2 = 0.$$

Крок 5. Обчислюється вхідний сигнал:

$$U_{inp.A}^4 = 0 \cdot (-0,118775 + 0 \cdot 0,512488 + 1 \cdot (-1,512488)) = -1,512488.$$

Крок 6. Адаптуються ваги зв'язків:

$$w_0 = -1,512488 + 0,5 \cdot (-1 + 1,512488) = -1,256244,$$

$$w_1 = -0,118775, \quad w_2 = 0,5122488.$$

Крок 7. Перевіряються умови зупинки. Якщо

$$\sum_{p=0}^2 |w_p^* - w_p| \leq \varepsilon,$$

то зупинка, інакше – перехід до кроку 2. Тут w_p^* ($p = \overline{0,2}$) – ваги зв'язків перед початком виконання кроку 2; w_p ($p = \overline{0,2}$) – ваги зв'язків, отримані на попередньому кроці 6.

$$\sum_{p=0}^2 |w_p^* - w_p| = |-0,382460 - (-1,256244)| + |0,379990 - (-0,118775)| + \\ + |0,429990 - 0,512488| = 1,290051 > 0,003,$$

отже, перехід до кроку 2.

Крок 8. Зупинка.

У табл. 1.2 наведено результати виконання дванадцяти разів кроку 2. З останніх рядків таблиці випливає, що ваги зв'язків сходяться до наступних значень: $w_0 = -1,5$; $w_1 = 0,5$; $w_2 = 1$.

Таблиця 1.2. Зміни ваги зв'язків адаліну під час ітераційної процедури

Номер кроку ітераційної процедури	Ваги зв'язків адаліну		
	w_0	w_1	w_2
—	-0,38246	0,37999	0,42999
1	-1,25624	-0,11878	0,51249
2	-1,38048	0,07805	0,57095
3	-1,44022	0,22920	0,88041
4	-1,47010	0,33472	0,94021
5	-1,48500	0,42243	0,97012
6	-1,49254	0,44381	0,98503
7	-1,49625	0,46815	0,99250
8	-1,49812	0,48220	0,99625
9	-1,49905	0,49015	0,99810
10	-1,49952	0,49460	0,99905
11	-1,49976	0,49690	0,99925
12	-1,49981	0,49833	0,99978

Аналогічні розрахунки щодо визначення параметрів адаліну, виконані при $\alpha \leq 0,3$, приводять до іншого набору значень ваг: $w_0 = -1,5$; $w_1 = w_2 = 1$.

Для визначення точних значень ваг w_0 , w_1 , w_2 , відповідних точці мінімуму функції E (1.22), розв'яжемо систему рівнянь

$$\frac{dE}{dw_0} = -2 \sum_{k=1}^4 (U_{out}^k - (x_1^k w_1 + x_2^k w_2 + w_0)) = 0,$$

$$\frac{dE}{dw_1} = -2 \sum_{k=1}^4 x_1^k (U_{out}^k - (x_1^k w_1 + x_2^k w_2 + w_0)) = 0, \quad (1.23)$$

$$\frac{dE}{dw_2} = -2 \sum_{k=1}^4 x_2^k (U_{out}^k - (x_1^k w_1 + x_2^k w_2 + w_0)) = 0.$$

Після підстановки вихідних даних з множини M система рівнянь (1.23) перетворюється на вигляд:

$$\begin{aligned} 2w_0 + w_1 + w_2 + 1 &= 0, \\ 2w_0 + 2w_1 + w_2 &= 0, \\ 2w_0 + w_1 + 2w_2 &= 0. \end{aligned} \tag{1.24}$$

Вирішуючи систему лінійних рівнянь (1.24), отримаємо:

$$w_0 = -1,5; \quad w_1 = 1; \quad w_2 = 1.$$

Таким чином, параметри адаліну, отримані при $\alpha \leq 0,3$, відповідають мінімуму функції E , а параметри, розраховані при $\alpha = 0,5$ з порушенням правої нерівності (1.21), не є вирішенням вихідного завдання.

Дельта-правило – це загальний метод навчання одношарових біполярних нейромереж (рис. 1.6). У загальному випадку для вихідних нейронів і навчальної множини

$$M = \{(X^1 = (x_1^1, \dots, x_n^1), U_{out}^1 = (U_{out.A1}^1, \dots, U_{out.Am}^1)), \dots, (X^p = (x_1^p, \dots, x_n^p), U_{out}^p = (U_{out.A1}^p, \dots, U_{out.Am}^p))\}$$

мінімізується сума квадратів різниць:

$$E = \sum_{k=1}^p \sum_{i=1}^m (U_{out.Ai}^k - U_{out.Ai}^k)^2$$

або

$$E = \sum_{k=1}^p \sum_{i=1}^m (U_{out.Ai}^k - \sum_{j=1}^n x_j^k w_{ji} - w_{0i})^2.$$

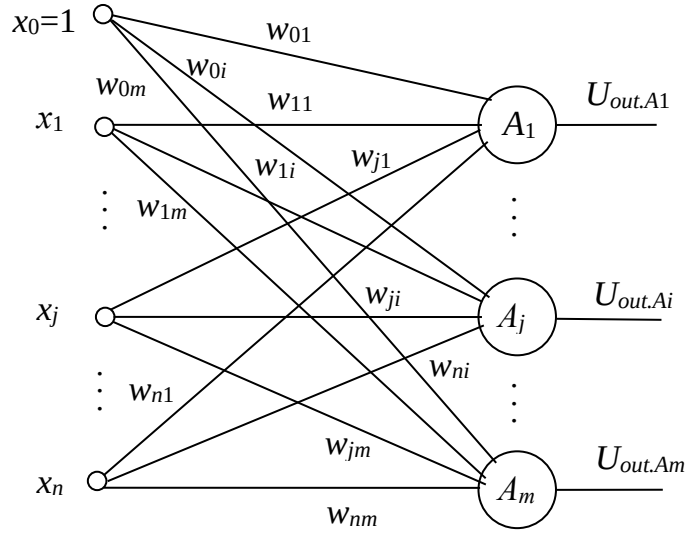


Рис 1.6. Одношарова мережа з m вихідними нейронами

Функція E залежить від усіх ваг w_{ji} , $j = \overline{0, n}$, $i = \overline{1, m}$, а її градієнт формується похідними $\frac{\partial E}{\partial w_{ji}}$ і вказує напрямок найбільшого зростання функції. Вектор, протилежний градієнту, визначає напрям найбільш швидкого зменшення функції.

Вычислим производную $\frac{\partial E}{\partial w_{ji}}$ для произвольного веса w_{ji} , $j = \overline{0, n}$,

$i = \overline{1, m}$:

$$\frac{\partial E}{\partial w_{ij}} = -2 \sum_{k=1}^p (U_{out.Ai}^k - (\sum_{j=1}^n x_j^k w_{ji} + w_0)) x_j^k \quad (1.25)$$

де $x_j^k \equiv 1$ при $j = 0$.

Для довільного, але єдиного вхідного вектора X^k вираз (1.25) набуває вигляду

$$\frac{\partial E}{\partial w_{ij}} = -2 \sum_{k=1}^p (U_{out.Ai}^k - (\sum_{j=1}^n x_j^k w_{ji} + w_0)) x_j^k \quad (1.26)$$

або

$$\frac{\partial E}{\partial w_{ij}} = -2 \sum_{k=1}^p (U_{out.Ai}^k - U_{inp.Ai}^k) x_j^k \quad (1.27)$$

З виразів (1.25) та (1.26) випливає, що вага w_{ji} ($j = \overline{0, n}, i = \overline{1, m}$) впливає лише на один вихідний сигнал елемента A_j . Для кожної точки поверхні функції багатьох змінних E виразів (1.26), (1.27) визначають швидкість її зміни залежно від ваги w_{ji} та компонент навчальної пари $(X^k, U_{out}^k) = ((x_1^k, \dots, x_n^k, \dots, U_{out.A1}^k, \dots, U_{out.Am}^k))$. Для найбільшого зменшення функції E приріст Δw_{ji} ваги w_{ji} слід брати пропорційним значенню похідної $\frac{\partial E}{\partial w_{ji}}$ але з протилежним знаком:

$$\Delta w_{ji} = \alpha (U_{out.Ai}^k - U_{inp.Ai}^k) x_j^k. \quad (1.28)$$

Вираз (1.28) є дельта-правилом для навчання одношарової мережі з багатьма біполярними вихідними елементами, за формою воно аналогічне дельта-правилу (1.19) для окремого адаліну.

Зазначимо, що дельта-правила (1.19), (1.28) можуть бути узагальнені на випадок, коли навчальні дані є лише представницькою вибіркою з безлічі вихідних даних.

Контрольні питання

1. Скільки приблизно нейронів містить мозок людини?
2. Який процесорний елемент використовується у звичайних нейронних мережах?
3. Наведіть кілька функцій активації нейронів.
4. Які завдання можна вирішувати за допомогою одного бінарного або біполярного нейрона?
5. Сформулюйте правило Хебба, що застосовується під час навчання окремих біполярних нейронів.
6. Чим відрізняються правила Хебба, які застосовуються при навчанні біполярних та бінарних нейронів?

7. Чи може одна й та сама архітектура нейронної мережі мати різні назви?
8. Розробити структуру мережі Хебба, здатну розпізнавати п'ять різних букв Вашого імені чи прізвища. При цьому обґрунтуйте вибір:
 - числа вхідних нейронів (число n x -елементів мережі має бути в межах $12 \leq n \leq 30$);
 - числа вихідних нейронів;
 - вибір векторів вихідних сигналів.
9. У чому відмінна риса адаліну?
10. Опишіть алгоритм навчання адаліну.
11. Наведіть дельта-правило для навчання одношарової нейронної мережі.

Г л а в а 2

ПЕРЦЕПТРОНИ

2.1. Перцептрони – клас моделей мозку

Перцептрони або персептрони (від *perceptio* – сприйняття) були першими штучними нейронними мережами, що з'явилися в результаті багаторічних досліджень мозку тварин та людини. Автор першого перцептронну – американський вчений Френк Розенблатт, який вперше опублікував свої дослідження в цій галузі у 1957 році. На думку Ф. Розенблатта [], перцептрони насамперед є класом моделей мозку, які пояснюють деякі його характерні функції. Зокрема, перцептрони, нехай і в самій елементарній формі, пояснюють деякі проблеми організації пам'яті біологічних систем, демонструють механізм набуття знань «пізнаючих (*cognitive*) систем» про навколишній світ і показують, що ці знання залежать як від когнітивної системи, так і від довкілля. За Розенблаттом, для різних видів тварин найпростіше уявлення про анатомічну структуру нервової системи може бути отримано за допомогою схеми, представленої на рис. 2.1.

Кожен із п'яти видів інформації про довкілля сприймається своїми спеціалізованими сенсорними нейронами і передається своїм окремим сенсорним трактам в центральну нервову систему. Через моторні нейрони центральна нервова система пов'язана з м'язами та залозами організму.

У перших роботах Розенблатт розглядав модель лише зорової системи. У найбільш простому вигляді ця модель включає три послідовно з'єднаних безлічі нейронів: чутливих (*S*-елементів), асоціюючих (*A*-елементів) і реагуючих (*R*-елементів). *S* елементам в нервовій системі тварини або людини відповідають сенсорні або рецепторні нейрони, що генерують сигнали на зовнішні подразнення (зображення), що надходять і передають їх *A*-нейронам.

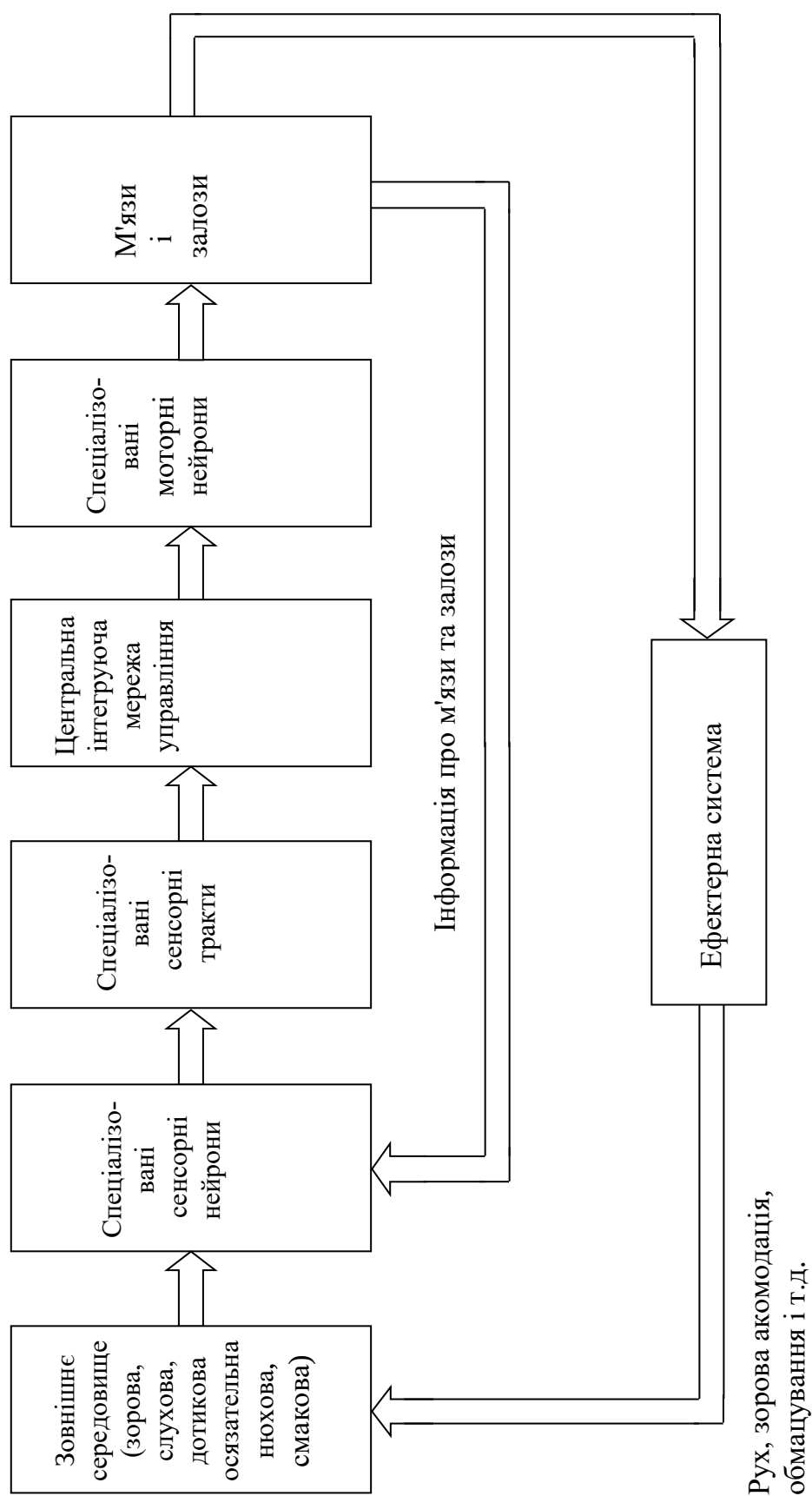


Рис. 2.1 Структура нервової системи

A-елементи аналогічні в нервовій системі живого організму нейронам, що утворюють локальний спеціалізований зоровий центр у корі головного мозку і зв'язує рецепторні нейрони з моторними. *R*-елементам у нервовій системі відповідають ефекторні (моторні) нейрони, впорядковані в обмежені топологічні структури та передаючі сигнали управління центральної нервової системи до м'язів та залоз організму.

Визначення 2.1. *S*-елемент називається простим, якщо він видає одиничний вихідний сигнал при входному сигналі, що перевищує певний заданий поріг, і нульовий сигнал – інакше.

Визначення 2.2. Простим асоціативним елементом називається *A*-елемент, який видає одиничний вихідний сигнал, якщо алгебраїчна сума його входних сигналів перевищує деякий заданий поріг $\theta > 0$, інакше – вихідний сигнал асоціативного нейрона дорівнює нулю.

Визначення 2.3. Простим біполярним (бінарним) елементом називається *R*-елемент, що видає одиничний вихідний сигнал, якщо алгебраїчна сума його входних сигналів більша або дорівнює пороговому значенню, і негативний одиничний (нульовий) сигнал, якщо сума його входних сигналів менша за заданий поріг.

Чутливі *S*-елементи живого організму (рис. 2.2) збуджуються від впливу енергії світла, якщо величини їх входних сигналів перевищують певний поріг θ_i . Рецепторні нейрони випадковим чином пов'язані з елементами *A*, вихідні сигнали яких відмінні від нуля тільки в тому випадку, коли порушено досить велике число сенсорних нейронів, що впливають на входи одного асоціюючого елемента. Простий *A*-елемент, аналогічно простому *S*-елементу, є активним і видає одиничний вихідний сигнал, якщо алгебраїчна сума сигналів на його вході перевищує задану порогову величину, в іншому випадку нейрон знаходиться в незбудженому стані. Коефіцієнти (ваги) зв'язків між *S*- та *A*-елементами постійні.

Сітківка ока
з S -елементів

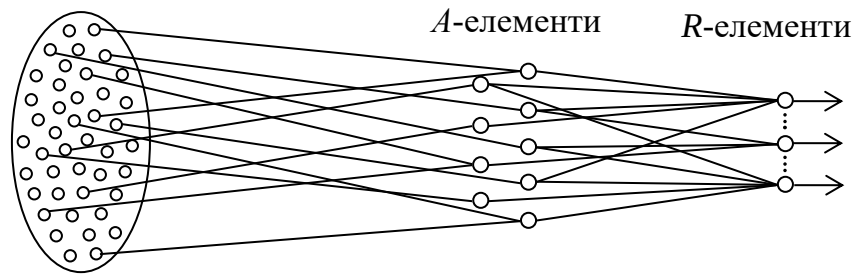


Рис. 2.2. Структура моделі зорової системи

Комбінація виходів всіх A -елементів є реакцією двох перших шарів перцептрону на пред'явлене вхідне зображення, яка за допомогою вихідного шару нейронів перетворюється на необхідну комбінацію вихідних сигналів системи. Часто вимагають, щоб кожному класу вхідних зображень відповідав лише певний активний R -нейрон. Необхідних комбінацій вихідних сигналів на кожен клас зображень домагаються на етапі навчання або адаптації перцептрону за рахунок зміни змінних ваг зв'язків між A - та R -елементами.

Поділ множини G зображень на два класи G_1 і G_2 можна виконати за допомогою одного вихідного елемента. У цьому випадку зображення першого класу може відповідати позитивний вихідний сигнал (+1) R -елемента, а другого класу – негативний (–1). На прикладі найпростішого (елементарного) перцептрону розглянемо різні способи навчання цих нейромереж, які вперше запропоновані та досліджені Розенблаттом.

Визначення 2.4 []. Простим перцептроном називається нейронна мережа, що складається з S -, A - та R -елементів і задовольняє наступним п'яти умовам:

1. У мережі є лише один R -нейрон, який з'єднаний зв'язками зі змінними вагами з усіма A -нейронами.
2. У мережі є тільки послідовні зв'язки від S - до A -елементів і від A -елементів до R -елементу.
3. Ваги зв'язків між S - та A -елементами є фіксованими.

4. Час передачі сигналів кожним зв'язком дорівнює нулю (або фіксованій постійній величині).

5. Вихідні сигнали всіх нейронів мережі формуються як:

$$U_{out} = f(\sum_i U_{inp,i}(t)),$$

де $\sum_i U_{inp,i}(t)$ – алгебраїчна сума сигналів, що надходять одночасно на вхід нейрона.

Визначення 2.5. Простий перцептрон з простими A - та R -елементами та передавальними функціями зв'язків виду:

$$C_{ij}(t) = w_{ij}(t)U_{out,i}(t - \tau_{ij}),$$

де $w_{ij}(t)$ – вага зв'язку між i -м і j -м нейронами у момент часу t ; $U_{out,i}(t - \tau_{ij})$ – вихідний сигнал i -го нейрона в момент часу $(t - \tau_{ij})$; τ_{ij} – час передачі сигналу $U_{out,i}(t - \tau_{ij})$ з виходу i -го нейрона на вхід j -го елемента називається елементарним перцептроном.

Елементарний перцептрон навчається чи налаштовується на розпізнавання двох класів зображень G_1, G_2 шляхом пред'явлення йому деяких послідовностей зображень із цих класів. Вчитель (людина або обчислювальна машина), що спостерігає реакцію перцептрону на кожне вхідне зображення, за наявності помилкових рішень мережі повинен коригувати ваги зв'язків між R - та A -елементами відповідно до деякої системи правил.

Визначення 2.6. Матрицею взаємодії перцептрону називається матриця, елементами якої є ваги зв'язків w_{ij} для всіх пар нейронів U_i, U_j мережі.

Якщо зв'язок між нейронами U_i, U_j відсутній (наприклад, у простому перцептроні немає зв'язків між R - та S -нейронами), то приймають $w_{ij} = 0$.

Матриця взаємодії фактично відображає стан пам'яті перцептрону. Безліч всіх можливих станів пам'яті мережі утворює фазовий простір мережі, який може

бути представлений у вигляді області в n -вимірному евклідовому просторі, кожна координатна вісь якого відповідає одному зв'язку мережі.

2.2. Навчання перцептронів за допомогою α - і γ -систем підкріплення

Визначення 2.7. Системою підкріплення нейронної мережі називається будь-який набір правил, за допомогою яких можна змінювати у часі стан пам'яті мережі (чи матрицю взаємодії).

Визначення 2.8. Позитивним (негативним) підкріпленням називається такий процес корекції ваг зв'язків, при якому вага зв'язку $w_{ij}(t)$, що починається на виході активного i -го елемента і закінчується на вході j -го елемента, змінюється на величину $\Delta w_{ij}(t)$, знак якої збігається зі знаком вихідного сигналу j -го нейрона (знак якої протилежний знаку вихідного сигналу j -го нейрона).

Існує велика кількість різних систем підкріплення [29], більшість з яких становить лише історичний інтерес. Тому зупинимось тільки на системі підкріплення з корекцією помилок, яка є основною в даний час.

У системі підкріплення з корекцією помилок насамперед необхідно визначити, чи реакція перцептрону є правильною. Доки вихідний сигнал R -елемента приймає бажане значення, величина сигналу підкріплення η дорівнює нулю. При появі неправильної реакції перцептрону використовується підкріплення, величина та знак якого в загальному випадку визначається монотонно зростаючою функцією f :

$$\eta = f(R^* - R), \quad (2.1)$$

де R^* – бажана реакція; R – отримана реакція; $f(0) = 0$.

Таким чином, при появі помилки для корекції ваг зв'язків використовуються сигнали, знаки яких протилежні знаку вихідного сигналу R -елемента. У зв'язку з цим розглянутий метод корекції ваг отримав назву системи з негативним підкріпленням.

Конкретним прикладом системи підкріплення із корекцією помилок є альфа-система підкріплення. У цій системі за наявності помилок ваги всіх активних зв'язків, які закінчуються на R -елементі, змінюють на однакову величину η , а ваги всіх неактивних зв'язків залишають без змін. Перцептрони, в яких застосовується альфа-система підкріплення, називаються альфа-перцептронами.

При використанні альфа-системи підкріплення сума ваг всіх зв'язків між R - і A -нейронами може зростати (або зменшуватися) від кроку до кроку, що повинно призводити до небажаних ситуацій, коли багато зв'язків мають максимальні (або мінімальні) ваги і не можуть використовуватися в подальшому процесі навчання нейронної мережі. Для усунення цього недоліку α -системи підкріплення була запропонована гамма-система підкріплення [29], яка має властивість консервативності щодо суми Σ_1 ваг всіх зв'язків між нейронами, тобто сума Σ_1 залишається постійною у процесі навчання перцептрону. Це досягається за рахунок того, що при наявності помилкової реакції перцептрону спочатку ваги всіх активних зв'язків змінюються на однакове значення η , а після цього з ваг всіх активних і пасивних зв'язків віднімається величина, що дорівнює відношенню суми зміни ваг усіх активних зв'язків до всіх зв'язків [29]. Зміна ваги окремих зв'язків при цьому визначається співвідношенням:

$$\Delta w_{ij} = \begin{cases} \eta - \frac{N_{ak}\eta}{N}, & \text{якщо } i - \text{й } A - \text{нейрон возбужден,} \\ -\frac{N_{ak}\eta}{N}, & \text{якщо } i - \text{й } A - \text{нейрон загальмован.} \end{cases} \quad (2.2)$$

де Δw_{ij} – у загальному випадку збільшення ваги зв'язку між i -м A -нейроном і j -м R -нейроном, для елементарного перцептрону $j = \text{const} = 1$; η – величина сигналу підкріплення; N_{ak} – кількість активних зв'язків; N – кількість зв'язків, що закінчуються на вході j -го елемента.

За такої системи корекції ваг зв'язків виконується рівність:

$$\eta N_{ak} - \frac{N_{ak} \eta}{N} \cdot N = 0,$$

з якої і випливає консервативність гамма-системи підкріплення щодо суми ваг усіх навчальних зв'язків.

Зауваження 2.1. Зазначимо, що співвідношення (2.2) у неявній формі передбачає, що ваги, що коригуються w_{ij} зв'язків досить далекі від своїх граничних значень $w_{ij \min} = 0$ і $w_{ij \max} = 1$, т.е.

$$w_{ij \min} \leq w_{ij} + \Delta w_{ij} \leq w_{ij \max} . \quad (2.3)$$

Якщо нерівності (2.3) порушуються, а вимога консервативності щодо суми Σ_1 вагів зв'язків залишається незмінним, то співвідношення (2.2) необхідно уточнити. Нехай, наприклад, серед активних зв'язків частина $N_{a.zp.}$ зв'язків мають граничні значення ваг $w_{ij \max}$ або $w_{ij \min}$ і для них виконуються умови:

$$w_{ij \max} + \Delta w_{ij} > w_{ij \max} \text{ або } w_{ij \min} + \Delta w_{ij} < w_{ij \min} . \quad (2.4)$$

Нехай також $N_{a.бзр.}$ активних зв'язків мають ваги, близькі до граничних, для яких справедливі нерівності

$$w_{ij} + \Delta w_{ij} > w_{ij \max} \text{ або } w_{ij} + \Delta w_{ij} < w_{ij \min} . \quad (2.5)$$

У цьому випадку загальна сума S_a початкових змін ваг активних зв'язків дорівнюватиме:

$$S_a = (N_{ak} - N_{a.zp.} - N_{a.бзр.})\eta + \text{sign}(\eta) \sum_{k=1}^{N_{a.бзр.}} \left| w_{kj}^{zp.} - w_{kj} \right|, \quad (2.6)$$

де w_{kj}^{ep} – граничне значення ваги зв'язку між k -м та j -м нейронами, $w_{kj}^{ep} \in \{0,1\}$;

Δw_{kj} – збільшення ваги зв'язку, що визначається за співвідношенням (2.2) без урахування наявності множини $\{w_{ij \min}, w_{ij \max}\} \equiv \{0,1\}$; $\text{sign}(\cdot)$ – знакова функція.

Якщо припустити, що для всіх пасивних зв'язків виконуються співвідношення (2.3), тоді від ваг пасивних зв'язків та ваг активних зв'язків, для яких не виконується співвідношення (2.4) або (2.5), віднімається величина $S_a = |(N_{ak} - N_{aep} - N_{abep})|\eta$. З урахуванням цих зауважень співвідношення (2.2) набуває вигляду:

$$\Delta w_{ij} = \begin{cases} \eta_1 = \eta - \frac{S_a}{(N_{ak} - N_{aep} - N_{abep})}, \text{ якщо } i\text{-й нейрон збуджений} \\ \quad \text{та не виконуються вимоги (2.4) або (2.5),} \\ 0, \text{ якщо } i\text{-й } A\text{-нейрон збуджений та виконується одна з вимог (2.4),} \\ 1 - w_{ij}, \text{ якщо } w_{ij} + \eta_1 > 1 \text{ та } i\text{-й } A\text{-нейрон збуджений,} \\ 0 - w_{ij}, \text{ якщо } w_{ij} + \eta_1 < 1 \text{ та } i\text{-й } A\text{-нейрон збуджений,} \\ -S_a / (N_{ak} - N_{aep} - N_{abep}), \text{ якщо } i\text{-й } A\text{-нейрон загальмований.} \end{cases}$$

Прикладом ще одного загального способу навчання перцептронів є метод корекції помилок випадковими обуреннями. Він передбачає, як і альфа-система підкріплення, у разі помилок – корекцію ваг активних зв'язків, але знак і величина корекції кожному зв'язку вибирається випадково відповідно до деякого заданого розподіла ймовірностей.

Приклад 2.1. Виконаємо навчання елементарного перцептрону з бінарними S - і A -нейронами та біполярним R -нейроном (рис. 2.3) розпізнаванню зображень літер H та Π (рис. 2.4, а, б) на рецепторному полі з дев'яти елементів (рис 2.4, в).

При цьому вимагатимемо, щоб при пред'явленні зображення літери H на виході R -елемента був сигнал "–1", при появі другого зображення – сигнал "+1".

Задамо в таблицях 2.1 та 2.2 ваги зв'язків w_{ij}^1 ($i = \overline{1,9}$, $j = \overline{1,6}$), w_k^2 ($k = \overline{1,6}$) відповідно між бінарними S - і A -нейронами і між A -нейронами та біполярним

нейроном R за допомогою генератора випадкових чисел, що генерує їх з кінцевої множини $\{0,1; 0,2; \dots; 0,9\}$.

Подамо на вхід перцептрону зображення літери Н (рис 2.4 а). Це зображення збуджує всі S -нейрони, крім другого та восьмого. Поодинокі сигнали з виходів збуджених бінарних S -нейронів через зв'язки, вагові коефіцієнти яких задано табл. 2.1. надходять на входи A -нейронів. Сумарний вхідний сигнал на вході i -го A -елемента визначається співвідношенням:

$$U_{inp.Ai} = \sum_{j=1}^9 U_{out.Sj} w_{ji}^1, \quad j = \overline{1, 6}. \quad (2.8)$$

де $U_{inp.Ai}$ – сигнал на вході i -го A -нейрону; $U_{out.Sj}$ – сигнал на виході j -го S нейрона; w_{ji} – вага зв'язку між j -м S -нейроном та i -м A -елементом.

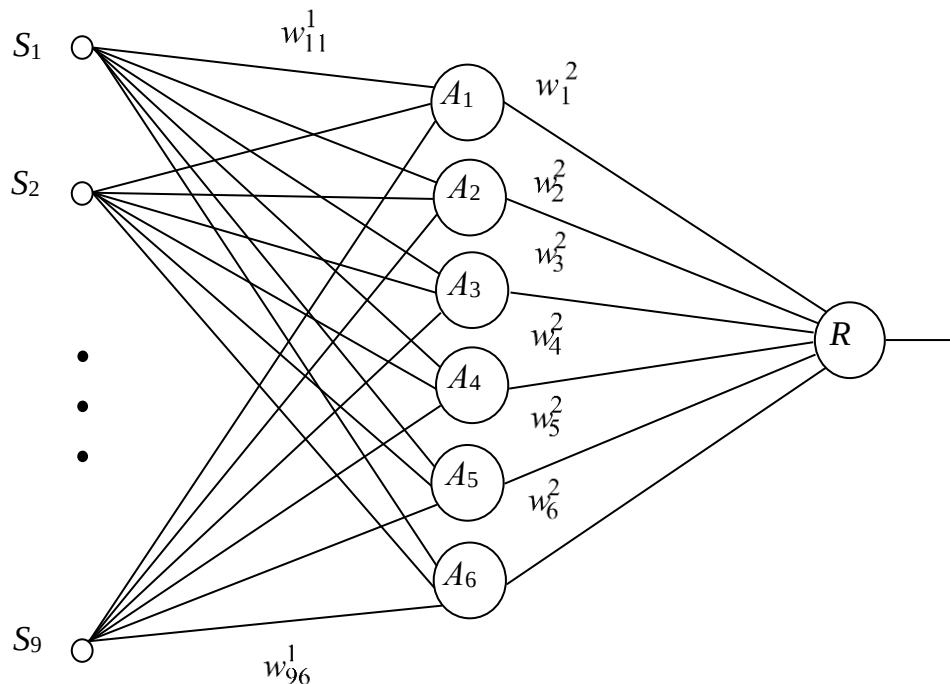


Рис. 2.3. Елементарний перцептрон

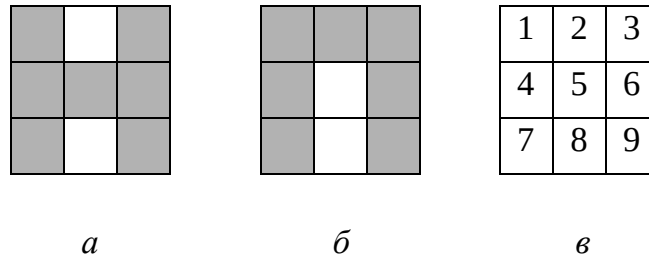


Рис 2.4. Зображення літер Н і П

Таблиця 2.1. Ваги w_{ij}^1 зв'язків перцептрону між S - та A -елементами

w_{ij}^1	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
A_1	0,3	0,2	0,1	0,6	0,5	0,4	0,9	0,6	0,7
A_2	0,2	0,1	0,3	0,4	0,5	0,6	0,7	0,1	0,9
A_3	0,3	0,5	0,1	0,6	0,5	0,4	0,9	0,4	0,7
A_4	0,4	0,3	0,2	0,1	0,8	0,7	0,6	0,6	0,9
A_5	0,5	0,3	0,3	0,6	0,1	0,2	0,9	0,2	0,7
A_6	0,6	0,5	0,4	0,1	0,2	0,3	0,8	0,5	0,8

Таблиця 2.2. Ваги w_k^2 зв'язків перцептрону між R - та A -елементами

w_k^2	A_1	A_2	A_3	A_4	A_5	A_6
R	0,2	0,8	0,6	0,9	0,8	0,1

Для першого A -нейрону маємо

$$U_{inp.A1} = \sum_{j=1}^9 U_{out.Sj} w_{j1}^1 = 1 \cdot 0,3 + 0 \cdot 0,2 + 1 \cdot 0,1 + 1 \cdot 0,6 + \\ + 1 \cdot 0,5 + 1 \cdot 0,4 + 1 \cdot 0,9 + 0 \cdot 0,6 + 1 \cdot 0,7 = 3,5.$$

Аналогічно обчислюються сигнали на входах інших A -елементів. Результати цих обчислень наведено у другому рядку табл. 2.3. У третьому рядку цієї таблиці – результати розрахунків сигналів на входах A -елементів при пред'явленні перцептрон зображення літери П.

Таблиця 2.3. Величини сигналів на входах A -елементів

Зображення	Сигнали на входах A -елементів					
	$U_{inp.A1}$	$U_{inp.A2}$	$U_{inp.A3}$	$U_{inp.A4}$	$U_{inp.A5}$	$U_{inp.A6}$
Літера Н	3,5	3,6	3,5	3,7	3,3	3,2
Літера П	3,2	3,2	3,4	3,2	3,6	3,5

Для спрощення розрахунків припустимо, що пороги θ_i , $i = \overline{1,6}$, всіх A -нейронів однакові

$$\theta_1 = \theta_2 = \dots = \theta_6 = \theta.$$

Якщо величина порога обрана менше 3,2, то при пред'явленні будь-якого зображення будуть збуджені всі A -нейрони, а якщо вибрати $\theta > 3,7$, то на виходах усіх нейронів будуть нульові сигнали. В обох цих випадках перцептрон не може виконувати розпізнавання зображень, що пред'являються.

Очевидно, що для забезпечення працездатності нейронної мережі поріг θ необхідно вибрати між 3,2 і 3,7 і таким чином, щоб при пред'явленні різних зображень порушувалися різні множини M_1 , M_2 A -елементів, причому бажано, щоб ці множини не перетиналися, т.о.:

$$M_1 \cap M_2 = \emptyset. \quad (2.9)$$

Нехай вихідний сигнал A -елементів визначається співвідношенням

$$U_{out.A1} = \begin{cases} 1, & \text{якщо } U_{inp.A} \geq \theta, \\ 0, & \text{якщо } U_{inp.A} < \theta. \end{cases}$$

тоді умова (2.9) виконується при $\theta = 3,5$ та при пред'явленні зображення літери Н будуть збуджені елементи A_1 , A_2 , A_3 и A_4 , а при пред'явленні букви П – нейрони A_5 и A_6 . Розрахуємо з урахуванням даних табл. 2.2 сигнали $U_{inp.RH}$, $U_{inp.РП}$ на вході R -нейрону при пред'явленні зображень літер Н та П:

$$U_{inp.RH} = \sum_{i=1}^6 U_{out.A1} w_i^2 = 1 \cdot 0,2 + 1 \cdot 0,8 + 1 \cdot 0,6 + 1 \cdot 0,9 + 0 \cdot 0,9 + 0 \cdot 0,1 = 2,5,$$

$$U_{inp.R\Pi} = \sum_{i=1}^6 U_{out.A1} w_i^2 = 0 \cdot 0,2 + 0 \cdot 0,8 + 0 \cdot 0,6 + 0 \cdot 0,9 + 1 \cdot 0,8 + 1 \cdot 0,1 = 0,9.$$

При величині порога R -елемента $\theta_R = 1,7$ та пред'явленні зображення літери Н на виході перцептрону буде сигнал "+1", а при пред'явленні другого зображення – сигнал "–1", що не відповідає вимогам до вихідних сигналів нейронної мережі. Використовуємо для налаштування перцептрону α систему підкріплення при величині сигналу підкріплення η рівному 0,1 та при пред'явленні послідовності зображень Н, П, Н, П, ... у моменти часу $t_1, t_2, t_3, \dots$. Процес адаптації ваги зв'язків між R - і A -нейронами ілюструється в табл. 2.4.

Таблиця 2.4. Адаптація ваг зв'язків перцептрону за допомогою α -системи підкріплення

Вагові коефіцієнти та вхідні сигнали	Моменти часу									
	t_0	t_1^H	t_2^P	t_3^H	t_4^P	t_5^H	t_6^P	t_7^P	t_8^P	t_9^P
w_1^2	0,2	0,1	0	0	0	0	0	0	0	0
w_2^2	0,8	0,7	0	0,6	0	0,5	0	0	0	0
w_3^2	0,6	0,5	0	0,4	0	0,3	0	0	0	0
w_4^2	0,9	0,8	0	0,7	0	0,6	0	0	0	0
w_5^2	0,8	0	0,9	0	1	0	1	1	1	1
w_6^2	0,1	0	0,2	0	0,3	0	0,4	0,5	0,6	0,7
$U_{inp.RH}$	2,5	2,1	–	1,7	–	1,4	–	–	–	–
$U_{inp.R\Pi}$	0,9	–	1,1	–	1,3	–	1,4	1,5	1,6	1,7

У другому стовпці таблиці при $t = t_0$ наведено значення вихідних ваг зв'язків та величини сигналів $U_{inp.RH}$, $U_{inp.R\Pi}$ на вході R -елемента при пред'явленні відповідно до зображень літер Н і П. При першому пред'явленні зображення літери Н на момент часу t_1 (позначено t_1^H) через наявність помилкового сигналу

на виході перцептрону коригуються ваги активних зв'язків $w_1^2, \dots, w_4^2$ на величину $\eta = 0,1$. Ця корекція зменшує сумарний вхідний сигнал $U_{inp.RH}$ до величини 2,1.

Нехай функціонування R -елемента описується співвідношенням:

$$U_{out.R} = \begin{cases} +1, & \text{якщо } U_{inp.R} \geq \theta_R, \\ -1, & \text{якщо } U_{inp.R} < \theta_R. \end{cases}$$

де θ_R – поріг R -елемента, тоді для досягнення правильної реакції R -елемента на зображення літери Н необхідні дві повторні корекції ваги зв'язків $w_1^2, \dots, w_4^2$. Результати цих корекцій наведено у табл. 2.4 відповідно у п'ятому та шостому стовпцях при $t = t_3^H$ та $t = t_5^H$. Після моменту часу $t = t_5^H$, оскільки виконується співвідношення (2.9), із вхідної послідовності можуть бути виключені зображення літери Н і пред'являтися лише зображення літери П. Результати корекції ваги зв'язків w_5^2, w_6^2 , визначальних сигнал на вході R -елемента при пред'явленні зображення літери П наведені в останньому рядку таблиці. Оскільки в прикладі, що розглядається, корекція вхідного сигналу R -елемента при пред'явленні зображення літери П здійснюється тільки за допомогою ваг двох зв'язків, причому, після другої корекції вага зв'язку w_5^2 приймає максимальне значення і надалі зростати не може, процес навчання нейронної мережі правильної реакції на друге зображення більш тривалий і закінчується тільки при $t = t_9^P$.

У табл. 2.5 наведено результати налаштування елементарного перцептрону при тих самих вихідних даних, але за допомогою γ -системи підкріплення. У другому стовпці табл. 2.5 при $t = t_0$ наведено вихідні ваги зв'язків та величини сигналів на вході R -елемента при пред'явленні зображень літер Н та П, а також сума Σ_1 ваг всіх зв'язків між R - і A -нейронами. Значення ваг зв'язків у третьому стовпці таблиці отримані після пред'явлення зображення літери Н в момент часу $t = t_1^H$. Так як $\eta = 0,1$, $N_{ак} = 4$ и $N = 6$, то, використовуючи співвідношення (2.2) для розрахунку збільшення ваг активних зв'язків, отримаємо:

$$\Delta w_1 = \Delta w_2 = \Delta w_3 = \Delta w_4 = \eta - N_{ак} \eta / N = 0,1 - 4 \cdot 0,1 / 0,6 = 0,0334, \quad (2.10)$$

а для збільшення ваг пасивних зв'язків маємо:

$$\Delta w_5 = \Delta w_6 = -N_{ак} \eta / N = -4 \cdot 0,1 / 0,6 = 0,0667. \quad (2.11)$$

Знаючи збільшення ваг зв'язків та використовуючи співвідношення:

$$w_i^2(t_1^H) = w_i^2(t_0) - \Delta w_i, \quad i = \overline{1,6},$$

неважко отримати і чисельні значення, наведені у третьому стовпчику табл. 2.5

Таблиця 2.5. Адаптація ваг зв'язків перцептрону за допомогою γ -системи підкріплення

Вагові коефіцієнти та вхідні сигнали	Моменти часу									
	t_0	t_1^H	$t_2^П$	t_3^H	$t_4^П$	t_5^H	$t_6^П$	$t_7^П$	$t_8^П$	$t_9^П$
w_1^2	0,2	0,1666	0,1333	0,1000	0,0800	0,0600	0,0400	0,0200	0,0000	0,0000
w_2^2	0,8	0,7666	0,7333	0,7000	0,6800	0,6600	0,6400	0,6200	0,6000	0,5750
w_3^2	0,6	0,5666	0,5333	0,5000	0,4800	0,4600	0,4400	0,4200	0,4000	0,3750
w_4^2	0,9	0,8666	0,8333	0,8000	0,7800	0,7600	0,7400	0,7200	0,7000	0,6750
w_5^2	0,8	0,8667	0,9334	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
w_6^2	0,1	0,1667	0,2334	0,3000	0,3800	0,4600	0,5400	0,6200	0,7000	0,7750
$U_{inp.RH}$	2,5	2,3664	–	2,1000	–	1,9400	–	1,7800	–	1,6250
$U_{inp.РП}$	0,9	–	1,1682	–	1,3800	–	1,5404	–	1,7004	1,7700
Σ_1	3,4	3,3998	3,4000	3,4000	3,4000	3,4000	3,4000	3,4000	3,4000	3,4000

При пред'явленні зображення літери П на момент часу $t = t_2^П$ активними є лише нейрони A_5 та A_6 , тому $N_{ак} = 2$ і співвідношення (2.2) дає такі чисельні значення приріст ваг зв'язків:

$$\Delta w_5 = \Delta w_6 = \eta - N_{ак} \eta / N = 0,1 - 2 \cdot 0,1 / 6 = 0,0667, \quad (2.12)$$

$$\Delta w_1 = \dots = \Delta w_4 = -N_{ак} \eta / N = -2 \cdot 0,1 / 6 = -0,0333. \quad (2.13)$$

Знаючи збільшення Δw_i ($i = \overline{1,6}$) і використовуючи вираз:

$$w_i^2(t_2^\Pi) = w_i^2(t_1^H) + \Delta w_i, \quad i = \overline{1,6},$$

Неважко отримати дані четвертого стовпця табл. 2.5.

При розрахунку збільшення ваг зв'язків при $t = t_3^H$ за співвідношеннями (2.10), (2.11) виявляється, що збільшення Δw_5 більше, ніж можлива зміна ваги зв'язку w_5^2 :

$$\Delta w_5 = -N_{ak}\eta/N = -4 \cdot 0,1/6 = 0,0667 > 1 - w_5^2(t_2^\Pi) = 1 - 0,9341 = 0,0659.$$

Тому для виконання умови консервативності щодо суми Σ_1 ваг зв'язків необхідно величину різниці

$$\Delta_1 w_5 = \Delta w_5 - \left| 1 - w_5^2(t_2^\Pi) \right| = 0,0667 - 0,0659 = 0,0008$$

використовуватиме для зміни ваг зв'язків, які прийняли граничних значень. Один із можливих способів використання різниці $\Delta_1 w_5$ – змінити кожен з таких $(N - 1)$ активних та пасивних зв'язків на величину:

$$\eta_1 = \frac{\Delta_1 w_5}{N - 1}.$$

Чисельні значення ваг зв'язків при $t = t_3^H$ наведені у п'ятому стовпці табл. 2.5.

При розрахунку ваг зв'язків за допомогою виразів (2.12), (2.13) для $t = t_4^\Pi$, необхідно враховувати, що змінюватись може вага лише одного активного зв'язку:

$$\Delta w_5 = 0, \tag{2.14}$$

$$\Delta w_6 = \eta - \frac{N_{ak} - 1}{(N - 1)} \eta = 0,1 - 1 \cdot 0,1/5 = 0,0800, \tag{2.15}$$

$$\Delta w_1 = \dots = \Delta w_4 = (N_{ак} - 1)\eta / (N - 1) = -1 \cdot 0,1 / 5 = -0,0200. \quad (2.16)$$

Аналогічно при $t = t_5^H$ маємо:

$$\Delta w_1 = \dots = \Delta w_4 = -\eta + N_{ак}\eta / (N - 1) = -0,1 + 4 \cdot 0,1 / 5 = -0,0200, \quad (2.17)$$

$$\Delta w_5 = 0, \quad (2.18)$$

$$\Delta w_6 = N_{ак}\eta / (N - 1) = 4 \cdot 0,1 / 5 = 0,0800. \quad (2.19)$$

Використовуючи співвідношення (2.14) – (2.19), аналогічним чином розраховуються ваги зв'язків при $t = t_6^H, t_7^H, t_8^H$. При $t = t_9^H$ кількість активних A -нейронів при пред'явленні зображення літери Н дорівнюватиме чотирьом, але у виразі (2.17) необхідно використовувати $N_{ак}=3$, оскільки ваговий коефіцієнт $w_1^2 = 0$. Зменшується до чотирьох і кількість ваг зв'язків, які використовуються для забезпечення сталості суми Σ_1 ваг усіх змінюваних зв'язків перцептрону.

В результаті отримуємо:

$$\Delta w_1 = \Delta w_6 = 0,$$

$$\Delta w_2 = \Delta w_3 = \Delta w_4 = -\eta + (N_{ак} - 1)\eta / (N - 2) = -0,1 + 3 \cdot 0,1 / 4 = -0,025,$$

$$\Delta w_5 = (N_{ак} - 1)\eta / (N - 2) = 3 \cdot 0,1 / (6 - 2) = 0,075,$$

$$U_{inp.RH} = 1,6250 < \theta_R, \quad U_{inp.РП} = 1,7754 > \theta_R = 1,7.$$

Таким чином, при пред'явленні літери Н на вході R -елемента сигнал менше величини порога θ_R і, отже, на виході R -нейрону буде необхідний сигнал "–1", а при пред'явленні зображення літери П на вихідному нейроні з'явиться заданий сигнал "+1".

Як впливає із співвідношення (2.2) в γ -системі підкріплення при одній і тій же величині сигналу підкріплення η , що і в α -системі, приращення Δw_{ij} ваги

коригованого активного зв'язку менше, ніж у α -системі. У зв'язку з цим можна очікувати, що в загальному випадку процес налаштування нейронної мережі другим методом потребує більшої кількості ітерацій. Проте аналіз даних таблиць 2.4 і 2.5 прикладу показує, що кількість ітерацій при використанні γ -системи підкріплення не більше, ніж при застосуванні α -системи. Це пояснюється так. Якщо ваги всіх зв'язків далекі від граничних значень, загальна сума S_α зміни ваг в α -системі пов'язані тільки з активними зв'язками:

$$S_\alpha = \eta N_{ak}.$$

У γ -системі аналогічна сума S_γ може бути рівна, менше або більше S_α , оскільки на кожній ітерації коригуються ваги всіх зв'язків:

$$S_\gamma = (\eta - \frac{N_{ak}\eta}{N})N_{ak} + \frac{N_{ak}\eta}{N}(N - N_{ak}).$$

За допомогою першого доданку у цьому виразі підраховується сума зміни ваг активних зв'язків, а за допомогою другого – сума зміни ваг пасивних зв'язків. Після перетворень маємо:

$$S_\gamma = N_{ak}\eta + (N_{ak} - \frac{2N_{ak}^2}{N})\eta. \quad (2.20)$$

З аналізу виразу (2.20) випливає, що залежно від співвідношення величин N_{ak} і N можливі три вирази:

$$S_\gamma = S_\alpha, \text{ якщо } N_{ak} = 0,5N. \quad (2.21)$$

$$S_\gamma > S_\alpha, \text{ якщо } N_{ak} < 0,5N. \quad (2.22)$$

$$S_\gamma < S_\alpha, \text{ якщо } N_{ak} > 0,5N. \quad (2.23)$$

Зі співвідношень (2.21) – (2.23) можна зробити висновок, що в загальному випадку за кількістю ітерацій у процесі навчання елементарного перцептрону жодна з систем підкріплення, що розглядаються, не має помітної переваги.

2.3. Теорема Розенблатта про елементарні перцептрони

Перша теорема Розенблатта [29] доводить існування елементарного перцептрону, здатного виконати будь-яку класифікацію заданої множини чорно-білих зображень, тобто. вона показує, що перцептрон є універсальним пристроєм для вирішення будь-якої задачі класифікації зображень.

Теорема 2.1. Нехай дано множину $W = \{W_1, \dots, W_n\}$ чорно-білих зображень на деякій сітківці $S = \{S_1, \dots, S_m\}$, тоді для будь-якої класифікації $C(W)$ множини W чорно-білих зображень на два підмножини W^1, W^2 існує не порожній клас елементарних перцептронів із сітківкою S , здатних виконати цю класифікацію.

Доказ. Для доказу достатньо показати існування хоч одного елементарного перцептрону, здатного виконати довільну класифікацію $C(W)$. Розглянемо перцептрон, кожному зображенню W_k ($k = \overline{1, n}$) на сітківці S якого відповідає один A -елемент – нейрон A_k , функціонування якого визначається виразом:

$$U_{out.Ak} = \begin{cases} 1, \text{ якщо } U_{inp.Ak} \geq \theta, \\ 0, \text{ якщо } U_{inp.Ak} < \theta, \end{cases} \quad (2.24)$$

де $U_{out.Ak}$ – вихідний сигнал нейрона A_k ; $U_{inp.Ak}$ – сигнал на вході нейрона A_k ,

$$U_{inp.Ak} = \sum_{j=1}^m U_{out.Sj} w_{jk}; \quad (2.25)$$

$U_{out.Sj}$ – вихідний сигнал j -го S -елемента,

$$U_{out.Sj} = \begin{cases} 1, \text{ якщо } S - \text{елемент збуджений,} \\ -1, \text{ якщо } S - \text{елемент загальмован,} \end{cases} \quad (2.26)$$

w_{jk} – вага зв'язку між j -м S -елементом та k -м A -нейроном; θ – поріг спрацьовування k -го A -елемента; $\theta = m$.

Для кожного зображення W_k задамо ваги w_{jk} ($j = \overline{1, m}, k = \overline{1, n}$) співвідношенням:

$$w_{jk} = \begin{cases} 1, \text{ якщо } S_j - \text{елемент збуджений } k - \text{м зображенням,} \\ -1, \text{ якщо } S_j - \text{елемент загальмован } k - \text{м зображенням.} \end{cases} \quad (2.27)$$

При пред'явленні будь-якого зображення W_k ($k = \overline{1, n}$) перцептрону, що задовольняє співвідношень (2.24) – (2.27), тільки на вході одного k -го A -нейрона, буде сигнал, рівний відповідно до співвідношення (2.25) числу m , і тільки на виході цього нейрона відповідно до виразу (2.24) буде одиничний вихідний сигнал. Тепер для правильного виконання поділу вихідної множини W на дві підмножини W^1, W^2 за допомогою елементарного перцептрону необхідно лише всім вагам зв'язків між R - і A -нейронами, які відповідають A -елементам, що збуджуються зображеннями підмножини W^1 , надати позитивних значень, а всіх ваг зв'язків A -нейронів, які порушуються зображеннями підмножини W^2 , – негативні значення, а потім задати вихідний сигнал R -нейрону $U_{out.R}$ виразом виду:

$$U_{out.R} = \begin{cases} 1, \text{ якщо } U_{inp.R} \geq 0, \\ -1, \text{ якщо } U_{inp.R} < 0, \end{cases}$$

де $U_{inp.R}$ – вхідний сигнал R -елемента.

У цьому випадку всі зображення підмножини W^1 кодуватимуться позитивним одиничним вихідним сигналом нейронної мережі, а підмножини W^2

– негативним, тобто. правильно виконуватиметься класифікація $C(W)$ вихідної множини W зображень.

Хоча побудована таким чином нейронна мережа не має суттєвого практичного значення, проте її наявність показує, що елементарний перцептрон є універсальним пристроєм класифікації будь-якої множини зображень на два класи. У тому випадку, коли кількість зображень множини W перевищує число A -нейронів, елементарний перцептрон втрачає свою універсальну здатність класифікувати зображення [24].

Теорема 2.2. Якщо число n зображень у множині W більше за число A -елементів елементарного перцептрону, то існує деяка класифікація $C(W)$ множини W чорно-білих зображень на дві підмножини W^1, W^2 , яка не може бути виконана перцептроном.

Теорема 2.3. Для будь-якого елементарного перцептрону з кінцевим числом A -нейронів ймовірність виконання класифікації $C(W)$, яка вибирається з рівномірного розподілу за всіма можливими класифікаціями множини $W = \{W_1, \dots, W_n\}$ зображень на два класи, що прагне до нуля при n , що прагне до нескінченності.

Теорема 2.1 доводить існування елементарного перцептрону, здатного виконувати будь-яку задану класифікацію $C(W)$ зображень деякої множини W на два класи, проте вона не вказує на алгоритми досягнення цієї здатності в процесі навчання нейронної мережі. Розенблатом було доведено важливу для теорії елементарних перцептронів теорему про наявність таких алгоритмів для α -перцептронів.

Розглянемо деяку класифікацію $C(W)$ множини W зображень на дві підмножини W^1, W^2 , яка може здійснюватися перцептроном з R -елементом, вихідний сигнал якого задовольняє умовам:

$$U_{out.R} = \begin{cases} 1, & \text{якщо } W_k \in W^1, \\ -1, & \text{якщо } W_k \in W^2, \end{cases} \quad k = \overline{1, n}. \quad (2.28)$$

Положим також, що:

$$\rho_k = \begin{cases} 1, & \text{якщо } W_k \in W^1, \\ -1, & \text{якщо } W_k \in W^2. \end{cases} \quad (2.29)$$

Визначення 2.9. Метод корекції помилок без квантування – це метод системи підкріплення з корекцією помилок, коли за помилкової реакції R -елемента з порогом θ на деяке зображення $W_k \in W$ до ваги кожного із зв'язків, що з'єднують активні A -нейрони з R -елементом, додається величина $\eta = \rho_k R_k$, де коефіцієнт R_k вибирається з умови, що після корекції ваги зв'язків виконується співвідношення (2.28), тобто. перцептрон правильно класифікує пред'явлене зображення. У методі корекції помилок з квантуванням застосовується це правило корекції ваг зв'язків, але величина R_k у загальному випадку набагато менше і правильний сигнал на виході R -нейрона, як правило, досягається не за одну ітерацію.

Теорема 2.4. Нехай дано елементарний α -перцептрон, безліч чорно-білих зображень $W = \{W_1, \dots, W_n\}$, деяка класифікація $C(W)$ цих зображень на дві підмножини, яка може бути виконана перцептроном. Зображення $W_1, \dots, W_n \in W$ подаються на вхід перцептрону довільної послідовності, в якій кожне з них з'являється неодноразово, через кінцеве число пред'явлень інших зображень. Тоді процес навчання перцептрону методом корекції помилок (з квантуванням або без квантування підкріплення) незалежно від початкових значень ваги зв'язків між R - і A -елементами завжди призводить за кінцеве число ітерацій до безлічі вагів зв'язків, за допомогою яких α -перцептрон може виконати задану класифікацію зображень.

Теорема 2.4 доводить наявність детермінованого методу навчання з корекцією помилок для елементарного перцептрону з квантуванням або без квантування підкріплення. Наступна теорема доводить, що навчання елементарного перцептрону може бути виконано і при менш жорстких вимогах до виду корекції – методом корекції помилок з випадковим законом підкріплення,

коли при появі помилки сигнал підкріплення формується як у α -системі, але знак підкріплення з ймовірністю 0,5 може бути позитивним чи негативним.

Теорема 2.5. Нехай дан елементарний α -перцептрон з кінцевим числом значень ваги зв'язків між R - і A -нейронами, множина чорно-білих зображень $W = \{W_1, \dots, W_n\}$, деяка класифікація $C(W)$ цих зображень на дві підмножини, яка може бути виконана α -перцептроном при деякому наборі ваги зв'язків між R - і A -нейронами, зображення $W_1, \dots, W_n \in W$ подаються на вхід перцептрону довільної послідовності, в якій кожне з них з'являється неодноразово через кінцеве число пред'явлень інших зображень. Тоді процес навчання перцептрону, величина сигналів підкріплення якого формується як і в системі з квантуванням підкріплення, а знак підкріплення вибирається з ймовірністю 0,5 позитивним або негативним, може бути виконаний за кінцевий час з ймовірністю, що дорівнює одиниці, незалежно від початкових значень ваг зв'язків між R - та A -нейронами.

Природно, що метод із випадковим знаком підкріплення вимагає більшого обсягу обчислень під час навчання перцептрону, ніж пряма корекція помилок із квантуванням чи без квантування підкріплення. Ще більшого обсягу обчислень вимагає метод, у якому виробляється випадковий вибір як знака, так й величини підкріплення. Розенблаттом доведена теорема про те, що з ймовірністю, що дорівнює одиниці, навчання перцептрону може бути виконано за кінцевий час і методом корекції випадковими збуреннями, коли підкріплення формується як в α -системі, але при цьому величина η і знак підкріплення для кожної ваги зв'язку вибираються окремо і незалежно відповідно до деякого заданого закону розподілу ймовірностей.

Менш універсальною системою підкріплення, ніж α -система та системи з випадковим формуванням підкріплення, є γ -система. Це свідчить наступна теорема Розенблатта [29].

Теорема 2.6. Нехай дано елементарний γ -перцептрон, підмножину $W = \{W_1, \dots, W_n\}$ чорно-білих зображень та деяка класифікація $C(W)$ цих зображень на два класи W^1, W^2 . Тоді для виконання класифікації $C(W)$ може існувати набір ваг зв'язків, недосяжний для γ -системи підкріплень.

Доказ. Нехай функціонування A -нейронів визначається виразом:

$$U_{out.Ak} = \begin{cases} 1, & \text{якщо } U_{inp.Ak} \geq \theta, \\ 0, & \text{якщо } U_{inp.Ak} < \theta, \end{cases} \quad k = \overline{1, n},$$

де $U_{out.Ak}$, $U_{inp.Ak}$ – відповідно вихідний та вхідний сигнали k -го A -нейрону; θ – поріг спрацювання A -нейронів.

Нехай також кожен A -нейрон збуджується лише одним зображенням з множини W , а класифікація $C(W)$ здійснюється за допомогою R -елемента, функціонування якого описується співвідношенням:

$$U_{out.R} = \begin{cases} 1, & \text{якщо } U_{inp.R} \geq \theta, \\ -1, & \text{якщо } U_{inp.R} < \theta, \end{cases} \quad k = \overline{1, n},$$

де $U_{out.R}$, $U_{inp.R}$ – відповідно вихідний та вхідний сигнали R -нейрону. Виберемо класифікацію $C(W)$, яка відносить усі зображення до класу W^1 ($U_{out.R}, U_{inp.R} > 0$) або до класу W^2 ($U_{out.R}, U_{inp.R} < 0$). Очевидно, що в першому випадку рішення існує тільки тоді, коли ваги всіх зв'язків між R - і A -елементами позитивні (або негативні, якщо всі зображення відносяться до класу W^2). Якщо для першого випадку початкові ваги всіх зв'язків між R - і A -нейронами негативні, а для другого – позитивні, то через властивість консервативності γ -системи підкріплення щодо суми всіх ваг зв'язків між R - і A -нейронами, вона не зможе виконати правильне налаштування ваг зв'язків перцептрону для аналізованої класифікації $C(W)$.

2.4. Тришарові перцептрони з кількома R -елементами та змінними вагами зв'язків між S - та A -нейронами

Природним узагальненням елементарного перцептрону є перцептрони, які від елементарного відрізняється лише наявністю кількох R -нейронів. Такі нейронні мережі можуть мати три різновиди топологічної структури. Для першої

структури характерне з'єднання кожного A -елемента з кожним R нейроном. Для другої властиво поділ множини A -елементів на непересічні підмножини, кожне з яких пов'язане тільки з одним R -елементом. Ця структура є хіба що об'єднання кількох елементарних перцептронів, які мають одну загальну множину S -елементів. Для третьої топологічної структури характерне з'єднання кожного R -елемента з деяким підмножиною випадково обраних A -елементів.

Перцептрон з N R -елементами, кожен з яких має два стани: "+1" і "-1" або "+1" і "0", може бути використаний для розпізнавання від N до 2^N образів. Число образів, що розпізнаються, залежить від способу їх кодування R елементами. Якщо використовується позиційне кодування, коли кожному образу призначається позитивний вихід лише одного R елемента (і нульовий або негативний вихід решти R -нейронів), то перцептрон може бути навчений розпізнаванню N образів. Якщо кожному образу у відповідність ставиться довільний двійковий код на вихідних нейронах перцептрона, тобто. використовується конфігураційне кодування образів, то перцептрон може бути навчений розпізнавання до 2^N образів.

За наявності можливості вибору між двома видами кодування, краще вибрати позиційний код, так як він за інших рівних умов у багатьох випадках призводить до кращих результатів, хоча за кількістю використовуваних елементів R і є менш економічним [29]. Якщо число N_1 образів перевищує число N R -нейронів, необхідно використання конфігураційного коду. При цьому бажано, щоб N_1 не було близько до 2^N , оскільки при граничних значеннях числа образів, що розпізнаються вибір коду окремих образів може стати вирішальним фактором для успішного навчання перцептрону [29].

Для навчання перцептронів з кількома R -елементами можна використовувати модифікований метод корекції помилок. Як і для елементарного перцептрона, у цьому разі перш за все визначається наявність неправильної реакції нейронної мережі на пред'явлене зображення. Це може виконуватися за допомогою співвідношення виду:

$$\eta = R - r, \quad (2.30)$$

де $\eta = (\eta_1, \dots, \eta_N)$ – вектор підкріплень ваг зв'язків від A - до R -нейронів; η_i ($i = \overline{1, N}$) – підкріплення ваг зв'язків, що йдуть до i -го R -елементу; $R = (R_1, \dots, R_N)$ – вектор бажаних реакцій R -нейронів; $r = (r_1, \dots, r_N)$ – вектор дійсних реакцій вихідних нейронів.

Якщо $\eta = 0$, тобто реакція перцептрону правильна, то ваги зв'язків між A - та R -нейронами залишаються незмінними. Якщо $\eta \neq 0$, то за допомогою скалярних компонентів векторного співвідношення (2.30) виконується корекція тільки ваг зв'язків, що йдуть до R -елементів, реакція яких відрізняється від бажаної.

Система підкріплення на основі співвідношення (2.30) завжди призводить до налаштування перцептрону, якщо таке налаштування взагалі існує для даної безлічі зображень, способу їх кодування та матриці постійних ваг S - A -зв'язків.

Умова фіксованості значень ваги зв'язків між двома першими шарами нейронів може негативно позначатися на можливостях перцептрону. У зв'язку з цим у процесі навчання бажана оптимізація ваг зв'язків як між A - і R -елементами, а й між S - і A -нейронами. На початку 60-х років Розенблатт запропонував кілька методів корекції S - A - зв'язків [29], однак вони або не гарантували визначення оптимального рішення, або вимагали надмірних обчислювальних витрат навіть для нейронних мереж з невеликою кількістю нейронів, оскільки використовували випадковий пошук для вирішення задачі, що відноситься до класу NP -складних. (Завдання відноситься до класу NP -складних, якщо не існує алгоритму, який би міг вирішити її за час, що зростає з розмірністю задачі не швидше за поліному кінцевого ступеня). Істотний прорив у навчанні трьох- та багат шарових перцептронів було отримано тільки після появи методу зворотного поширення помилки [24, 25], який буде розглянуто надалі.

2.5. Загальні висновки про можливості тришарових перцептронів із послідовними зв'язками

Тришаровий перцептрон з послідовними зв'язками між S -, A - і R -елементами є найпростішою нейронною мережею, здатною навчитися розпізнаванню образів. Він має наступні характеристики [29]:

1. Може бути навчений розпізнаванню зображень кінцевого зовнішнього середовища.

2. Не потребує точної розробки своєї структури для вирішення конкретних завдань. Один і той же перцептрон може бути навчений розв'язанню різних завдань для однієї і тієї ж або різних зовнішніх середовищ.

3. Результати навчання перцептрону не залежать від початкових значень ваг його зв'язків (або початкового стану його пам'яті) і не вимагають докладного опису змін його параметрів.

4. Навчання перцептрону можливе на будь-якій навчальній послідовності, в якій кожне зображення з'являється неодноразово протягом кінцевого інтервалу часу.

5. Перцептрон має схильність до прояву однакових реакцій на будь-які два зображення, які в рамках розв'язуваної задачі можуть вважатися в певному сенсі ідентичними. Однак при необхідності він може бути навчений розрізняти ці зображення.

6. Перцептрони значною мірою поступаються біологічним системам у швидкості, ефективності, економічності та надійності навчання і мають такі принципові недоліки:

– для вирішення реальних завдань розпізнавання в науці та техніці вимагають надмірно великої кількості нейронів та часу навчання;

– мають низьку здатність до узагальнення;

– характеризуються недостатньою здатністю до виділення суттєвих елементів у складних зображеннях.

7. Трирядні перцептрони здатні розпізнавати тільки лінійно розділені класи зображень.

Останній факт було встановлено 1969 року у роботі Мінського і Пейперта "Перцептрони" [12], у якій автори зробили глибокий теоретичний аналіз труднощів застосування трирядних перцептронів. Вони показали, що труднощі у використанні перцептронів для розпізнавання образів носять принциповий характер, пов'язаний з тим, що цей тип нейронних мереж може розпізнавати лише класи зображень, що лінійно розділяються. Для зображень на площині це означає, що як роздільник є пряма лінія (рис. 2.5), для зображень у тривимірному просторі – площина, для чотирьох і більш багатовимірних просторів – гіперплощина.

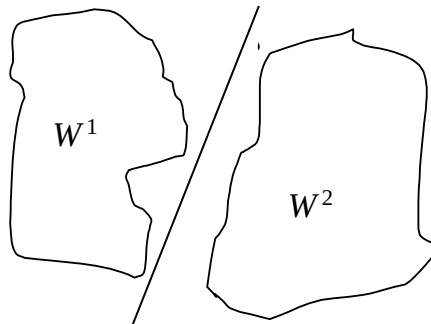


Рис. 2.5. Лінійно розділені класи зображень

Перцептрон із n двійковими чутливими S -нейронами може сприймати 2^n різних вхідних зображень. Оскільки при одному елементі R кожному вхідному зображенню може відповідати два різних виходи, то за допомогою елементарного перцептрону може бути отримано $n_1 = 2^{2^n}$ різних функцій від n двійкових змінних або n_1 різних класифікацій 2^n вхідних зображень. У табл. 2.6 [25] для різних значень n наведено загальне число n_1 функцій та число n_2 класифікацій, що відповідають лінійно розділеним зображенням, а також процентний зміст $\delta = 100n_2/n_1$ лінійно здійснених класифікацій у загальній кількості можливих.

Таблиця 2.6. Лінійно розділені та нероздільні функції

n	n_1	n_2	δ
1	4	4	100
2	16	14	87,5
3	256	104	40,6
4	65536	1882	2,87
5	$4,295 \cdot 10^9$	94572	0,0022
6	$1,845 \cdot 10^{19}$	15028134	$0,0081 \cdot 10^{-10}$

З аналізу даних четвертого шпальти табл. 2.6 слід, що при $n \geq 5$ частка лінійно поділених зображень серед їх загального числа мізерно мала. У зв'язку з цим, область можливого застосування тришарового перцептрону як пристрою розпізнавання дуже обмежена. І вона ще більше звужується через відсутність простого та універсального способу визначення лінійної роздільності конкретних образів.

Як один із яскравих прикладів, що ілюструють обмежені можливості тришарового перцептрону як моделі мозку, Мінський та Пейперт наводять приклад перцептрону з бінарними нейронами: двома S - та одним A -елементом. Вихідний сигнал A -елемента $U_{out.A}$ описується виразом:

$$U_{out.A} = \begin{cases} 1, & \text{якщо } U_{out.A} \geq \theta_A, \\ 0, & \text{якщо } U_{out.A} < \theta_A, \end{cases} \quad (2.31)$$

де $U_{inp.A}$ – сигнал на вході A -нейрону,

$$U_{inp.A} = S_1 w_1 + S_2 w_2 \quad (2.32)$$

S_1, S_2 – вихідні сигнали S -елементів; w_1, w_2 – ваги зв'язків перцептрону.

Задамо за допомогою табл. 2.7 два класи V_1 та V_2 зображень. Дані другого та третього стовпців таблиці на площині $S_1 S_2$ (рис. 2.6) визначають точки A_1, A_2 ,

B_1, B_2 , що відносяться відповідно до першого та другого класу зображень. Співвідношення (2.32) з урахуванням виразу (2.31) при фіксованих значеннях ваг w_1, w_2 зв'язків та порога θ_A визначає на площині S_1S_2 пряму:

$$S_2 = \frac{w_1}{w_2} S_1 + \frac{\theta_A}{w_2}, \quad (2.33)$$

де $w_1, \theta_A \geq 0$ та $w_2 > 0$.

Таблиця 2.7. Таблиця істинності функції ВИКЛЮЧНЕ АБО

Клас зображень	Зображення	Крапки на площині S_1S_2	Вихідний сигнал елемента S_1	Вихідний сигнал елемента S_2	Требуемый сигнал перцептрона
V_1	1. $S_1=0, S_2=1$	A_1	0	1	1
	2. $S_1=1, S_2=0$	A_2	1	0	1
V_2	1. $S_1=0, S_2=0$	B_1	0	0	0
	2. $S_1=1, S_2=1$	B_2	1	1	0

Змінюючи w_1, w_2, θ_A , неважко переконатися, що при їх будь-яких значеннях пряма (2.33) не може бути проведена таким чином, щоб точки A_1, A_2 були по її один бік, а точки B_1, B_2 – по іншу. Таким чином, перцептрон, що розглядається, не може виконати задану класифікацію зображень, що пред'являються.

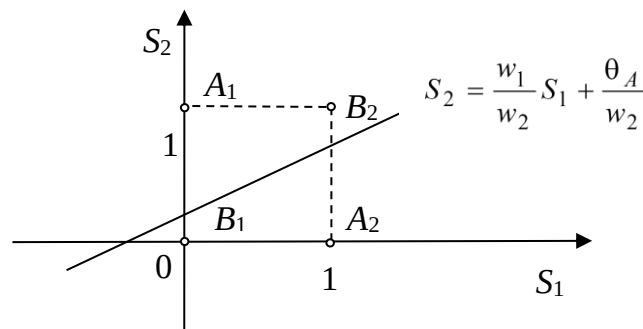


Рис. 2.6. Приклад лінійно нероздільних класів зображень

Оскільки четвертий, п'ятий та шостий стовпці табл. 2.7 можна розглядати як таблицю істинності функції $f_{i\text{ або}}(S_1, S_2)$ ВИКЛЮЧНЕ АБО (або функції суми за модулем два), то аналізований двошаровий перцептрон не може бути використаний для реалізації навіть такої найпростішої функції, а також функції $f_E(S_1, S_2)$ еквівалентності, яку можна визначити як заперечення функції ВИКЛЮЧНЕ АБО:

$$f_E(S_1, S_2) = \overline{f_{i\text{ або}}(S_1, S_2)}.$$

На рис. 2.6 функції $f_E(S_1, S_2)$ будуть відповідати ті самі точки A_1, A_2, B_1, B_2 області визначення, як і функції $f_{i\text{ або}}(S_1, S_2)$ але тільки приймати вона буде в цих точках значення, протилежні значенням функції $f_{i\text{ або}}(S_1, S_2)$

$$f_E(0, 0) = f_E(1, 1) = 1,$$

$$f_E(0, 1) = f_E(1, 0) = 0.$$

Очевидно, що властивості перцептрону, що розглядається, як розпізнаючої системи не поліпшаться, якщо до трьох наявних нейронів додати R -елемент, на входи якого надходитимуть сигнали з виходів A -нейронів. У роботі Мінського та Пейперта не випадково аналізувалися дво-, а не тришарові перцептрони. Збільшення числа шарів мережі з нейронами, що мають лінійну функцію активації, не призводить до якісної зміни властивостей мережі, оскільки обчислення вектора U_{out}^i вихідних сигналів нейронів i -го шару зводиться до множення вектора U_{inp}^i вхідних сигналів на матрицю W^i ваг нейронів i -го шару:

$$U_{out}^i = U_{inp}^i W^i$$

Отже, додавання до мережі $(i + 1)$ шару нейронів призводить до появи на її виході вектора

$$U_{out}^{i+1} = U_{inp}^{i+1} W^{1+1} = U_{out}^i W^{1+1} = (U_{inp}^i W^1) W^{1+1},$$

де U_{inp}^{i+1} , U_{out}^{i+1} – відповідно вектор вхідних та вихідних сигналів нейронів $(i+1)$ -го шару; W^{1+1} – матриця вагів нейронів $(i + 1)$ -го шару.

Оскільки множення матриць асоціативно, то

$$U_{out}^{i+1} = U_{inp}^i (W^i W^{1+1}) = U_{inp}^i W^*),$$

а це означає, що два шари лінійної мережі еквівалентні одному шару нейронів з матрицею ваг W^* , яка дорівнює добутку матриць ваг цих шарів. Звідси випливає, що кожна лінійна багатошарова нейронна мережа еквівалентна певної одношарової мережі. Для розширення можливостей багатошарових нейронних мереж порівняно з одношаровими необхідно використовувати нейрони з нелінійною функцією активації.

Подібні та інші приклади, що розглянуті Мінським та Пейпертом призвели до різкого падіння інтересу дослідників до дво- та трирядних перцептронів, а оскільки на той час були відсутні ефективні методи навчання багаторядних перцептронів з нелінійними функціями активації, то і до всіх перцептронів взагалі.

2.6. Багатошарові перцептрони

Перцептрони, розглянуті раніше, містили не більше трьох шарів нейронів: шар вхідних чутливих S -елементів, шар вихідних R елементів та шар A -нейронів, розташований між S - та R -елементами і часто званий шаром прихованих нейронів.

Визначення 2.10. Багатошаровими називаються перцептрони, що містять два і більше шарів A -нейронів.

Приклад чотиришарового перцептрону наведено на рис. 2.7. Перцептрон наведений на рис. 2.8, а, на перший погляд, також є чотиришаровим, проте інше графічне уявлення цієї нейронної мережі (рис. 2.8, б) показує, що його можна розглядати як тришаровий перцептрон, що має зв'язки між A -елементами всередині прихованого шару.

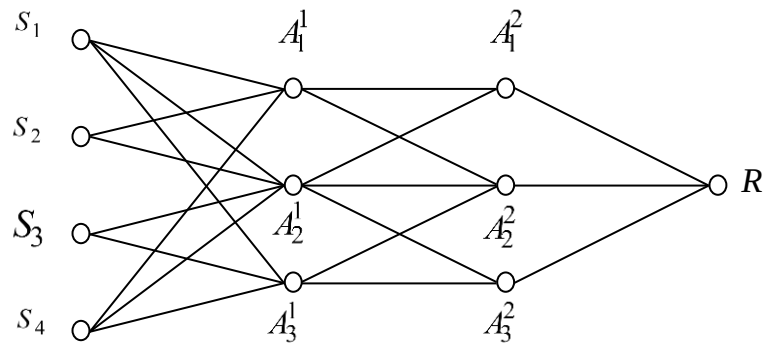


Рис. 2.7. Чотиришаровий перцептрон

Визначення 2.11. Логічною відстанню L від нейрона A до нейрона B називається найменша кількість зв'язків, за допомогою яких сигнал може бути переданий від нейрона A нейрону B .

Визначення 2.12. Нейронна мережа, в якій всі зв'язки з виходами нейронів з логічною відстанню L від найближчого S -елемента закінчуються на нейронах з логічною відстанню $L + 1$ від найближчого S -елемента, називається нейронною мережею з послідовними зв'язками.

Визначення 2.13. Перцептрони, у яких деякі зв'язки з'єднують один з одним нейрони, що знаходяться на однаковій логічній відстані від S -елементів, називаються перцептронами з перехресними зв'язками.

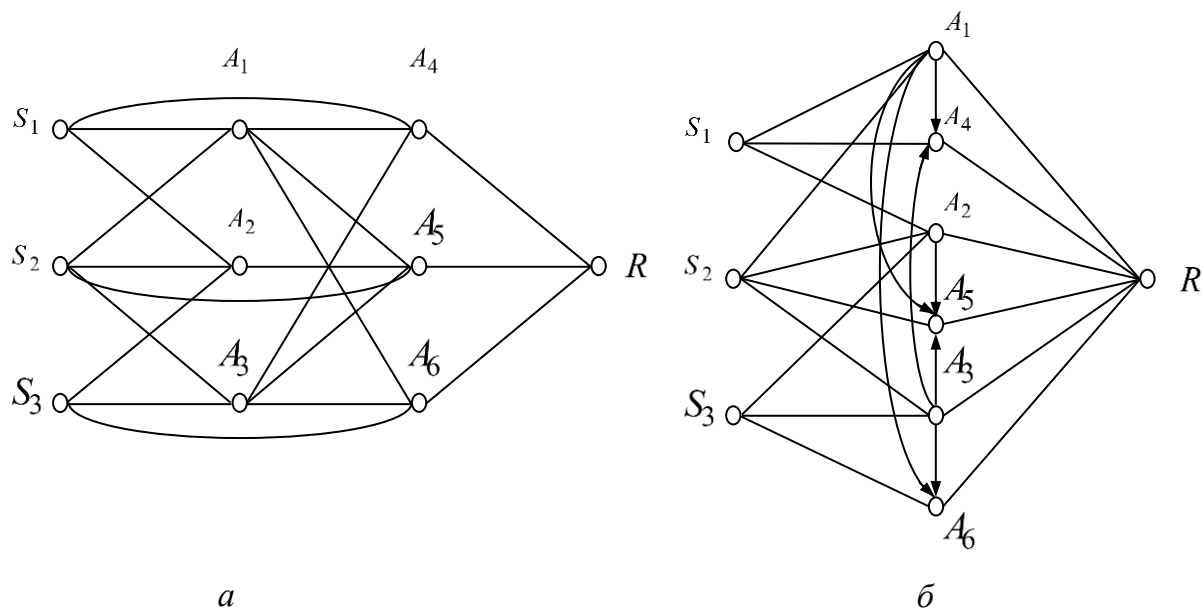


Рис. 2.8. Тришаровий перцептрон з перехресними зв'язками

Визначення 2.14. Перцептрони, які мають деякі зв'язки з виходів нейронів з логічною відстанню L від S -елементів закінчуються на входах нейронів з логічною відстанню L_1 ($L_1 < L$) від S -елементів, називаються перцептронами із зворотними зв'язками.

Збільшення числа шарів у перцептронах, введення перехресних та зворотних зв'язків веде до якісної зміни властивостей перцептронів як систем, що розпізнають. Повернемося до прикладу Мінського і Пейперта, розглянутому у попередньому параграфі, і проаналізуємо можливості нейронної мережі, зображеної на рис. 2.9.

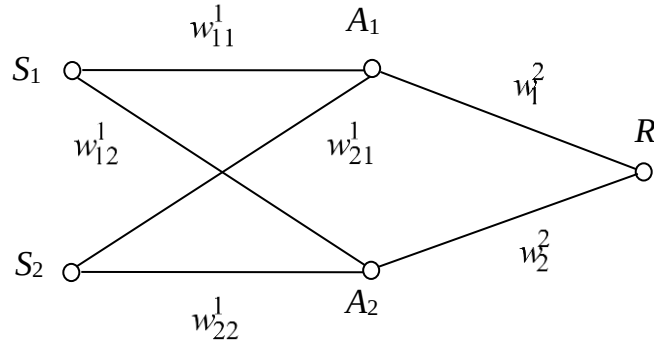


Рис. 2.9. Нейронна мережа з двома А-елементами

У прикладі Мінського та Пейперта площина S_1S_2 за допомогою єдиної прямої ділиться на дві частини. Співвідношення виду (2.34), (2.35), записані для вихідних сигналів нейронів A_1, A_2 , визначають дві прямі:

$$S_{2A1} = \frac{w_{11}^1}{w_{21}^1} S_1 + \frac{\theta_{A1}}{w_{21}^1}, \quad (2.34)$$

$$S_{2A2} = \frac{w_{22}^1}{w_{12}^1} S_1 + \frac{\theta_{A2}}{w_{12}^1}, \quad (2.35)$$

де θ_{A1}, θ_{A2} – пороги нейронів A_1, A_2 .

Два із можливих положень цих прямих зображені на рис. 2.10 де вони ділять площину S_1S_2 на чотири частини. Внаслідок цього виникають якісно нові можливості для поділу зображень на два класи.

Відповідно до зображень на рис. 2.10 і співвідношеннями (2.34), (2.35) можна вимагати, щоб нейрон A_1 забезпечував одиничний вихідний сигнал нижче прямого S_{2A1} , а нейрон A_2 – вище за пряму S_{2A2} . Якщо покласти, що $w_1^2 = w_2^2 = 0,5$, то вихідний сигнал R -елемента дорівнює одиниці тільки всередині заштрихованих V-подібних областей (рис. 2.10).

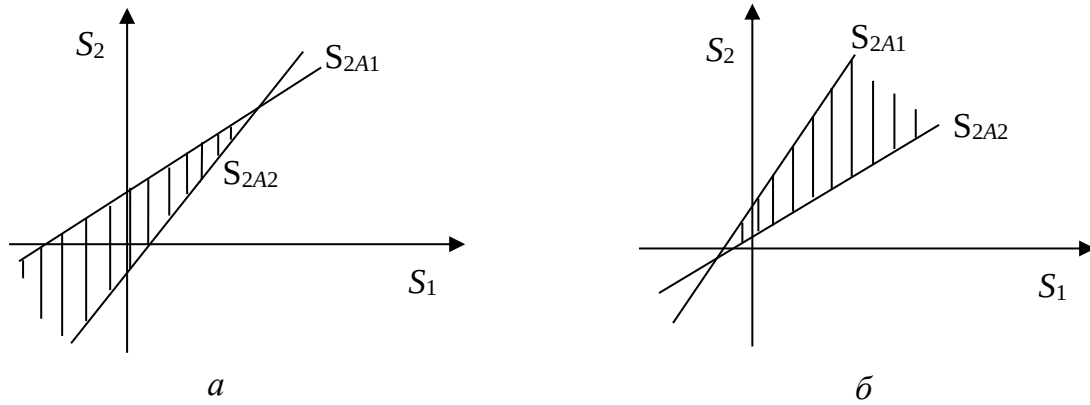


Рис. 2.10. Области рішень, що задаються нейронною мережею з двома А-елементами

Аналіз співвідношень (2.34), (2.35) та рис. 2.10 показує, що перцептрон, зображений на рис. 2.9, може класифікувати об'єкти, що описуються функціями виду ВИКЛЮЧАЄ АБО або еквівалентність. Дійсно, нехай вихідні сигнали S - та A -елементів описуються виразами

$$U_{out.Si} = \begin{cases} -1, & \text{якщо } U_{inp.} \leq 0, \\ 0, & \text{якщо } U_{inp.} > 0, \end{cases} \quad i = \overline{1, 2},$$

$$U_{out.Ak} = \begin{cases} 0, & \text{якщо } U_{inp.Ak} \leq 0, \\ 1, & \text{якщо } U_{inp.Ak} > 0, \end{cases} \quad k = \overline{1, 2},$$

а ваги зв'язків та поріг R -елемента мають такі значення: $w_{11}^1 = w_{22}^1 = -1$; $w_{12}^1 = w_{21}^1 = 1$; $w_1^2 = w_2^2 = 0,5$; $0,5 < \theta_R < 1$, тоді функціонування перцептрон може бути описано за допомогою табл. 2.8. Зіставлення вхідних та вихідних сигналів перцептрон показує, що він реалізує функцію еквівалентності. Аналогічним чином, при належному виборі параметрів нейронної мережі, може бути отриманий і перцептрон, що реалізує функцію ВИКЛЮЧАЮЧНЕ АБО.

Входи S -елементів не обов'язково повинні бути двійковими, вони можуть бути безперервними. При безперервних входах, що класифікуються на площині S_1S_2 нейронної мережі зображення способні займати не тільки дискретні, но і неперервні обмежені або неограничені області.

Таким чином, тришарові перцептрони з двома A -елементами можуть виконувати більш загальну класифікацію, відокремлюючи об'єкти, які лежать у дискретних або безперервних, обмежених або необмежених областях, отриманих при перетині двох прямих. Введення третього A -нейрону дозволяє (за відповідного вибору параметрів перцептрону) отримувати на площині S_1S_2 області трикутної форми, а використання m трійок A -нейронів дає можливість виділяти на площині S_1S_2 щонайменше m довільних трикутних областей. Це ще більше розширює можливості перцептрону, оскільки дозволяє використовувати нейронну мережу навіть з одним R -елементом для класифікації множин об'єктів, які займають на площині одну або кілька трикутних областей, або один клас займає одну або кілька трикутних областей, а інший – всю площину, що залишилася і т.д.

Якщо для формування замкнутої опуклої області у площині S_1S_2 використовувати не три, а більше A -нейронів з відповідними параметрами, то можна домогтися виділення в цій площині будь-якого опуклого багатокутника.

Визначення 2.15. Геометрична фігура називається опуклою, якщо для будь-яких її двох точок відрізок прямих, що з'єднує ці точки, цілком належить розглянутій геометричній фігурі.

Таблиця 2.8. Сигнали нейронів перцептрону, що реалізує функцію еквівалентності

Клас зображень	Зображення (вхідні сигнали S -нейронів)	Вхідні сигнали A -елементів		Вихідні сигнали нейронів				
		$U_{ex.A1}$	$U_{ex.A2}$	S_1	S_2	A_1	A_2	R
V_1	1. $U_{inp.S1} = 0$, $U_{inp.S2} = 1$.	1	-1	-1	0	1	0	0
	2. $U_{inp.S1} = 1$, $U_{inp.S2} = 0$.	-1	1	0	-1	0	1	0
V_2	1. $U_{inp.S1} = 0$, $U_{inp.S2} = 0$.	0	0	-1	-1	1	1	1
	2. $U_{inp.S1} = 1$, $U_{inp.S2} = 1$.	0	0	0	0	1	1	1

Тришарові перцептрони здатні виділяти на площині будь-які опуклі області. Сформувати за їх допомогою довільну неопуклу фігуру не можна, проте можна отримувати неопуклі області за допомогою опуклих фігур. На рис. 2.11 наведено дві неопуклі фігури G_1 та G_2 , перша з яких утворена об'єднанням двох опуклих чотирикутників A і B

$$G_1 = A \cup B, \quad (2.36)$$

а друга отримана шляхом виключення із трикутника C площі трикутника $D_1D_2D_3$, є частиною чотирикутника D , тобто. сформована в результаті перетину трикутника C і $\bar{D}$:

$$G_2 = C \cap \bar{D}. \quad (2.37)$$

Фігура G_1 , а загалом і всі неопуклі фігури, одержувані шляхом об'єднання кінцевого числа опуклих багатокутників A_k ($k = 1, p$):

$$A_1 \cup A_2 \cup \dots \cup A_p, \quad (2.38)$$

можуть бути сформовані за допомогою тришарових перцептронів, а фігури виду G_2 – ні. Для формування неопуклих фігур, що задаються за допомогою співвідношень виду (2.37) або більш загальному випадку за допомогою виразу

$$A_1 \cup A_2 \cup \dots \cup A_k \cap \overline{A_{k+1}} \cap \overline{A_{k+2}} \cap \dots \cap \overline{A_p}, \quad (2.39)$$

необхідні чотиришарові перцептрони. У чотиришарових перцептронах у перших трьох шарах елементів формуються довільні опуклі фігури, а нейрони останнього шару синтезують різні об'єднання та перетину цих фігур, що описуються за допомогою співвідношень виду (2.36) - (2.39) або їм подібних.

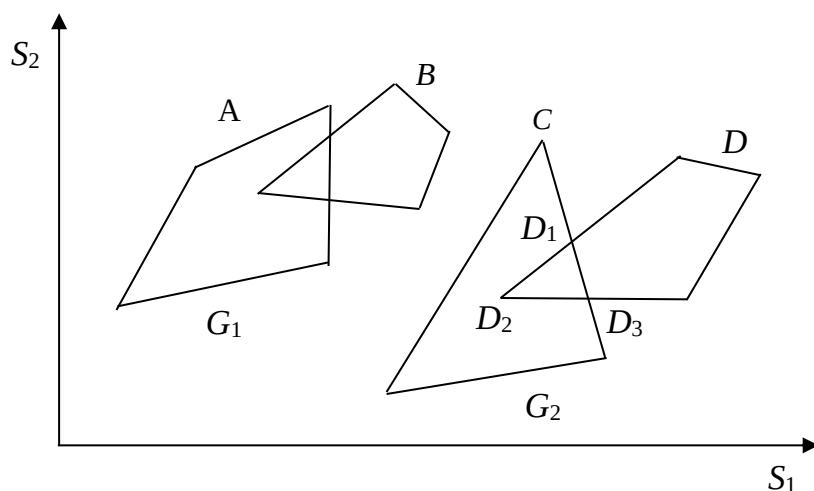


Рис. 2.11. Неопуклі області рішень, що задаються перцептронами

За наявності великої кількості нейронів число багатокутників та його сторін може зростати практично необмежено, що дозволяє апроксимувати області довільної форми з будь-якою заданою точністю.

Несмотря на то, что приведенные возможности многослойных перцептронов были известны давно, их практическое использование для распознавания образов стало возможно только в 80-е годы, когда для настройки или обучения нейронных сетей появились эффективные алгоритмы метода обратного распространения ошибки (back-propagation error), которые будут рассмотрены в следующей главе пособия.

Контрольні питання

1. Хто автор першої штучної нейронної мережі?
2. Яка архітектура моделі зорової системи у людини?
3. Яка нейронна мережа називається елементарним перцептроном?
4. Скільки різних класів зображень може розпізнавати елементарний перцептрон?
5. Нарисуйте архітектуру елементарного перцептрону.
6. Що таке α -система підкріплення?

7. Що таке γ -система підкріплення?
8. Виконайте навчання елементарного перцептрону, який розпізнає дві літери з Вашого імені, за допомогою α -системи підкріплення.
9. Виконайте навчання елементарного перцептрону, який розпізнає дві літери з Вашого імені, за допомогою γ -системи підкріплення.
10. Наведіть теореми Розенблатта про елементарні перцептрони.
11. Яка система підкріплення α -система або γ -система є більш універсальною?
12. Які перцептрони називаються багат шаровими?
13. Які загальні висновки можна сформулювати про можливості тришарових перцептронів?
14. Нарисуйте схему чотиришарового перцептрону.
15. Які перцептрони називаються перцептронами з перехресними зв'язками?
16. Які перцептрони називаються перцептронами із зворотними зв'язками?
17. Яка геометрична фігура називається опуклою?
18. Які перцептрони здатні виділяти на площині будь-які опуклі фігури?
19. За допомогою якого перцептрону можна класифікувати об'єкти, що описуються функціями виду ВИКЛЮЧНЕ АБО?
20. Чи можна за допомогою тришарових перцептронів класифікувати безперервні вхідні елементи?
21. Нарисуйте чотиришаровий перцептрон із зворотними зв'язками.

Г л а в а 3

МЕТОД ЗВОРОТНОГО ПОШИРЕННЯ ПОМИЛКИ

3.1. Функції активації, що використовуються у методі зворотного розповсюдження

Метод зворотного поширення помилки, для назви якого часто-густо використовують лише перші три слова, відіграв важливу роль у відродженні інтересу до нейронних мереж як інструменту розв'язання широкого кола різноманітних проблем. Метод був розроблений незалежно один від одного кількома дослідниками в період з 1974 по 1986 [30 – 33], але отримав широке визнання і застосування тільки після публікації Румельхарта, Хінтона і Вільямса в 1986 [33]. Метод зворотного поширення фактично є узагальнення методу найменших квадратів стосовно багатосарових нейронних мереж. Він за допомогою простого градієнтного спуску мінімізує середньоквадратичну помилку між дійсним та необхідним виходом нейронної мережі.

Використання градієнтного спуску передбачає певні вимоги до активаційних функцій нейронів. Функція активації повинна бути безперервною, диференційованою та монотонно зростаючою, апроксимувати одиничний максимум і мінімум. Її похідна повинна просто обчислюватись і бажано, щоб вона виражалася через значення функції. (Вимога монотонності зростання для радіально-симетричної функції (1.10) виконується лише на інтервалі $(-\infty, 0)$, а на інтервалі $(0, \infty)$ воно замінюється вимогою монотонності спадання).

Однією з найбільш поширених є бінарна функція сигмоїдальна активації (1.8) з областю значень $(0, 1)$ і областю визначення $(-\infty, \infty)$:

$$g(x) = \frac{1}{1 + e^{-x}}.$$

При стремленні x к $-\infty$ функція $g(x)$ прагне до нуля, а при $x \rightarrow \infty$ вона прагне одиниці.

Похідна функції $g(x)$ дорівнює

$$g'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = g(x)(1 - g(x)). \quad (3.1)$$

Оскільки жоден із співмножників виразу (3.1) не перетворюється на нуль в інтервалі $(-\infty, \infty)$, то функція $g(x)$ не має екстремальних точок, а так як в інтервалі, що розглядається, обидва співмножники невід'ємні, то $g(x) > 0$. Отже, функція $g(x)$ в інтервалі $(-\infty, \infty)$ монотонно зростає від нуля до одиниці, оскільки

$$\lim_{x \rightarrow -\infty} (g(x)) = 0 \quad \text{та} \quad \lim_{x \rightarrow \infty} (g(x)) = 1.$$

Функція $g'(x)$ при прагненні x к $-\infty$ або к $+\infty$ має своєю межею нуль. Її похідна

$$(g'(x))' = \frac{e^{-x}(-1+e^{-x})}{(1+e^{-x})^3}$$

позитивна при $x \in (-\infty, 0)$, негативна при $x \in (0, \infty)$ і звертається в нуль тільки при $x = 0$. Функція $g'(x)$ при $x = 0$ має єдиний максимум:

$$g'(0) = \frac{e^0}{(1+e^0)^2} = \frac{1}{4}.$$

Похідна функції $g(x)$ симетрична щодо осі ординат, тобто

$$g'(x) = g'(-x)$$

або

$$\frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^x}{(1+e^x)^2}.$$

Дійсно

$$\frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}}{\left(\frac{e^x+1}{e^x}\right)^2} = \frac{e^x}{(1+e^x)^2}.$$

Для функції $g'(x)$ справедлива також властивість

$$\int_{-\infty}^{\infty} g'(x) dx = \frac{1}{1+e^{-x}} \Big|_{-\infty}^{\infty} = 1. \quad (3.2)$$

Якщо розглядати сигмоїдальну функцію активації

$$g(x) = \frac{1}{1+e^{-kx}}$$

при k , який прагне до ∞ , то отримаємо функцію активації бінарного нейрона (співвідношення (1.2) та рис. 1.2a). При цьому похідна $g'(x)$ сигмоїдальної функції в силу співвідношення (3.2) та наявності єдиного максимуму при $x = 0$ перетворюється в δ -функцію Дірака (або просто в δ -функцію), тобто при $k \rightarrow \infty$, із сигмоїдальної функції можна отримати граничну функцію активації найпростішого бінарного нейрона. Сигмоїдальна характеристика має ще одну важливу гідність, що впливає з властивостей її першої похідної (або тангенса кута нахилу дотичної до графіку функції $g(x)$). Для слабких сигналів ця характеристика вхід-вихід має найбільше посилення, оскільки $g'(x)$ приймає максимальне значення при $x = 0$. Коли ж величина позитивного або негативного сигналу зростає, посилення падає, причому тим більше, чим більше величина

вхідного сигналу відрізняється від нуля. Це з тим, що функція $g'(x)$ монотонна і симетрична щодо осі ординат і зменшується до нуля при $x \rightarrow \pm\infty$.

Таким чином, завдяки властивостям сигмоїдальної кривої автоматично регулюється посилення нейронів мережі та вирішується проблема обробки як слабких, так і сильних сигналів.

Іншою поширеною функцією активації є біполярна функція сигмоїдальна (1.9), яка має область значень $(-1, 1)$ і визначається як

$$g_1(x) = 2g(x) - 1 = \frac{2}{1 + e^{-kx}} - 1.$$

Її похідна дорівнює

$$g'_1(x) = \frac{k}{2} (1 - g_1(x))(1 + g_1(x)).$$

Оскільки

$$g_1(x) = \frac{2}{1 + e^{-kx}} - 1 = \frac{1 - e^{-kx}}{1 + e^{-kx}},$$

а гіперболічний тангенс визначається виразом

$$\text{th}x = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

яке можна перетворити на вигляд

$$\text{th}x = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}},$$

то цю активаційну функцію можна розглядати і як гіперболічний тангенс.

Метод зворотного розповсюдження помилки може використовуватися і для налаштування нейронних мереж, елементи яких мають функції несигмоїда активації. Прикладом такої функції є радіально-симетрична:

$$g_2(x) = e^{-kx^2}.$$

Її похідна визначається співвідношенням

$$g'_2(x) = -kxe^{-kx^2} = 2kxg_2(x).$$

3.2. Основний алгоритм методу зворотного розповсюдження помилки

Метод зворотного поширення помилки розроблявся на навчання нейронних мереж із будь-яким числом шарів. Проте задля викладу суті методу досить розглянути мережу, зображену на рис. 3.1. В даний час у літературі немає єдиного підходу до того, як рахувати число шарів багатошарових нейронних

мереж. Одні автори підраховують кількість шарів нейронів, включаючи і вхідний шар, інші – лише кількість шарів нейронів з безліччю приєднаних до їх входів зв'язків із ваговими коефіцієнтами, стверджуючи, що вхідні нейрони не утворюють повноцінного шару, оскільки виконують лише функцію розподілу вхідних сигналів. При першому підході мережа, зображена на рис. 3.1, має три шари, а при другому – два. Ми притримувалися і будемо дотримуватися першого підходу, вважаючи, що нейронна мережа, зображена на рис. 3.1, має три шари.

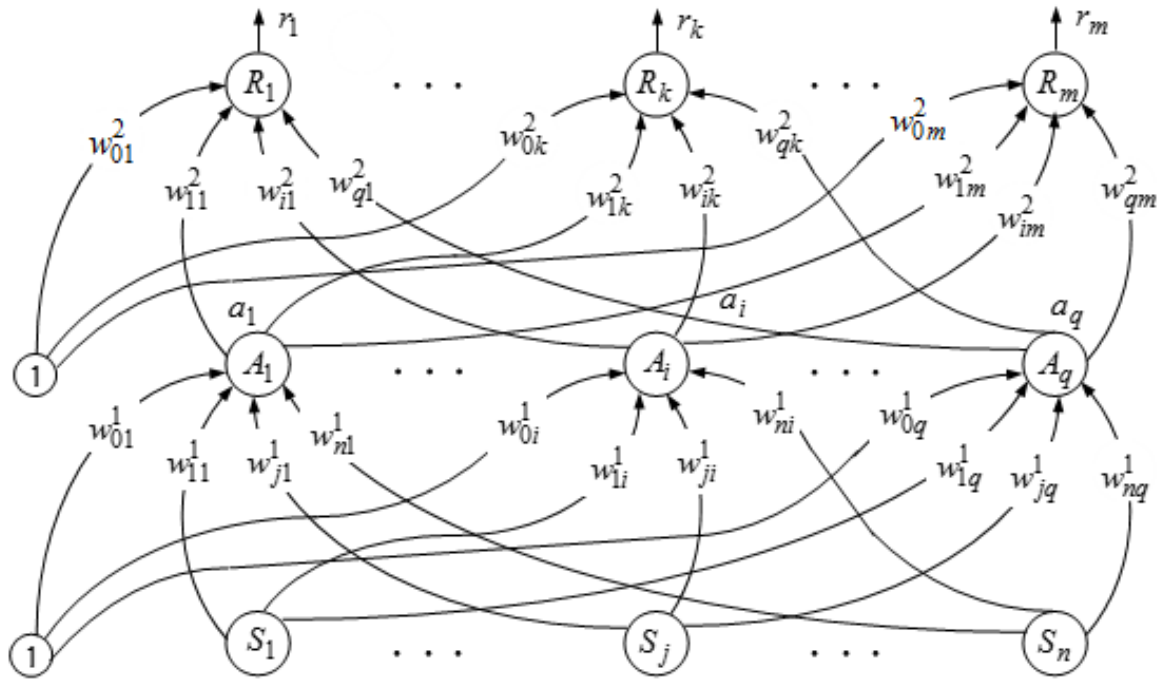


Рис. 3.1. Тришарова мережа для ілюстрації методу зворотного розповсюдження

На рис. 3.1 показано, що вихідні R -нейрони і нейрони прихованого шару (A -елементи) можуть мати зсуви, які діють подібно до нейронів, у яких вихідний сигнал завжди дорівнює одиниці (зазвичай ці елементи нейронних мереж, зображені на рис. 3.1, в явному вигляді не показують). Зміщення на типові елементи R_k , A_i вихідного та прихованого шару подаються за допомогою зв'язків, що мають відповідно вагові коефіцієнти w^2_{0k} , w^1_{0i} ($k = \overline{1, m}$, $i = \overline{1, q}$). З огляду на одиничні сигнали на входах зв'язків величини зміщень на зазначені нейрони також рівні відповідно w^2_{0k} , w^1_{0i} ($k = \overline{1, m}$, $i = \overline{1, q}$).

Вхідні зображення $S^1 = (S^1_1, \dots, S^1_j, \dots, S^1_n)$, ..., $S^p = (S^p_1, \dots, S^p_j, \dots, S^p_n)$, ..., $S^L = (S^L_1, \dots, S^L_j, \dots, S^L_n)$ подаються на чутливі нейрони першого шару (S_j , $j = \overline{1, n}$), які не перетворюють сигнали вхідних зображень

$$S^p_j = U^p_{inp.Sj}, \quad j = \overline{1, n}, \quad p = \overline{1, L},$$

а лише розподіляють їх по входах нейронів прихованого шару, тобто

$$U_{oup.Sj}^p = U_{inp.Sj}^p, \quad j = \overline{1, n}, \quad p = \overline{1, L}.$$

Кожному вхідному зображенню S^p ($p = \overline{1, L}$) ставиться у відповідність вектор $t^p = (t_1^p, \dots, t_k^p, \dots, t_m^p)$, що задає необхідний вихід нейронної мережі. Вхідні зображення S^p та відповідні їм вектори t^p утворюють навчальні пари (S^p, t^p) , $p = \overline{1, L}$ нейронної мережі.

Перед строгим описом конкретного алгоритму пояснимо загальну ідею методу зворотного розповсюдження помилки.

Метод зворотного поширення має три явно виражені етапи або фази: пряму передачу вхідного навчального зображення, зворотне поширення помилки (якщо вона є) і корекцію ваг зв'язків (якщо мала місце помилка).

Протягом прямої передачі кожен вхідний нейрон приймає сигнал від пред'явленого зображення $S^p = (S_1^p, \dots, S_j^p, \dots, S_n^p)$ і передає цей сигнал кожному елементу A_i ($i = \overline{1, q}$) прихований шар. Кожен прихований нейрон підсумовує всі свої вхідні сигнали та за допомогою функції активації обчислює вихідний сигнал (a_i), який посиляє кожному R -елементу. Кожен елемент R_k ($k = \overline{1, m}$) обчислює свій вихідний сигнал r_k . Вектор $(r_1, \dots, r_k, \dots, r_m)$ вихідних сигналів R -елементів є реакцією мережі на дане вхідне зображення. На другому етапі роботи алгоритму кожен вихідний елемент R_k ($k = \overline{1, m}$) порівнює свій вихідний сигнал із компонентами необхідного вихідного вектора $t^p = (t_1^p, \dots, t_k^p, \dots, t_m^p)$ нейронної мережі для цього вхідного зображення S^p з метою визначення величини власної помилки. Ґрунтуючись на розрахованій помилці, для кожного R -елемента обчислюється величина δ_k ($k = \overline{1, m}$), яка використовується для поширення помилки від вихідного нейрона R_k назад до всіх нейронів прихованого шару, з'єднаних з R_k . Пізніше δ_k ($k = \overline{1, m}$) використовуються також для корекції ваги зв'язків між прихованим і вихідним шаром. Подібним чином для кожного прихованого елемента A_i ($i = \overline{1, q}$) обчислюється величина δ_i ($i = \overline{1, q}$). Множина величин δ_i ($i = \overline{1, q}$) використовується для корекції ваг зв'язків між вхідним та прихованим шаром.

На третьому етапі роботи алгоритму за допомогою величин δ_k ($k = \overline{1, m}$), δ_i ($i = \overline{1, q}$) проводиться корекція ваги зв'язків w_{ik}^2 та w_{ji}^1 , а також зміщень w_{0k}^2, w_{0i}^1 нейронів вихідного та прихованого шарів.

Умов закінчення роботи алгоритму має бути не менше двох – для успішного та невдалого його застосування. Прикладом умови успішного закінчення роботи алгоритму може бути таке:

Алгоритм закінчує роботу, якщо при пред'явленні кожного зображення S^p ($p = \overline{1, L}$) без корекції ваг зв'язків та зміщень на виході нейронної мережі з'являється вектор $r^p = (r_1^p, \dots, r_k^p, \dots, r_m^p)$, що задовольняє вимогам

$$\left| t_k^p - r_k^p \right| \leq E_k^p, \quad k = \overline{1, m},$$

де t_k^p, E_k^p – відповідно компоненти цільового вектора $t^p = (t_1^p, \dots, t_k^p, \dots, t_m^p)$, та вектора $E^p = (E_1^p, \dots, E_k^p, \dots, E_m^p)$ допустимої помилки для p -го зображення.

Умовою припинення роботи алгоритму при його невдалому застосуванні може бути досягнення алгоритмом граничного часу T_n навчання нейронної мережі.

Навчальний алгоритм методу зворотного розповсюдження помилки.

Крок 1. Множина невеликих випадкових значень ініціюють ваги всіх зв'язків нейронної мережі.

Крок 2. Перевіряються умови коректності завдання вихідних даних та необхідності роботи алгоритму і, якщо вони виконуються, то реалізуються кроки 3 – 11 алгоритму.

Крок 3. Для кожної навчальної пари (вхідне зображення $S^p = (S_1^p, \dots, S_n^p)$, необхідний вихідний вектор $t^p = (t_1^p, \dots, t_m^p)$) виконуються кроки 4 – 10 алгоритму.

Етап прямої передачі.

Крок 4. Кожен вхідний чутливий нейрон S_j ($j = \overline{1, n}$) отримує вхідний сигнал S_j^p і передає його до всіх елементів A_i ($i = \overline{1, q}$) прихованого шару.

Крок 5. Кожен прихований нейрон A_i ($i = \overline{1, q}$) підсумовує свої зважені вхідні сигнали:

$$U_{inp.Ai} = w_{0i}^1 + \sum_{j=1}^n S_j^p w_{ji}^1,$$

та за допомогою своєї функції активації g_{Ai} розраховує вихідний сигнал

$$a_i = U_{out.Ai} = g_{Ai}(U_{inp.Ai}),$$

та посиляє його на входи всіх R -елементів.

Крок 6. Кожен вихідний нейрон R_k ($k = \overline{1, m}$) підсумовує свої зважені вхідні сигнали:

$$U_{inp.Rk} = w_{0k}^2 + \sum_{i=1}^q a_i w_{ik}^2$$

та за допомогою своєї функції активації розраховує вихідний сигнал

$$r_k = U_{out.Rk} = g_{Rk}(U_{inp.Rk}).$$

Етап зворотного розповсюдження помилки

Крок 7. Кожен вихідний нейрон R_k ($k = \overline{1, m}$) отримує компоненту t_k^p необхідного вихідного вектора $t^p = (t_1^p, \dots, t_m^p)$, та обчислює свою помилку

$$E_k = \frac{1}{2} (t_k^p - r_k)^2$$

та її похідну

$$\delta_k = E_k = (t_k^p - r_k) g'_{Rk}(U_{inp.Rk}),$$

визначає за допомогою коефіцієнта швидкості навчання α (зазвичай α задовольняє нерівностям $0,01 \leq \alpha \leq 1$) збільшення ваг зв'язків w_{ik}^2 ($i = \overline{1, q}, k = \overline{1, m}$) та зміщень w_{0k}^2 ($k = \overline{1, m}$):

$$\Delta w_{ik}^2 = \alpha \delta_k a_i, \quad \Delta w_{0k}^2 = \alpha \delta_k,$$

та посиляє похідні δ_k до елементів прихованого шару.

Крок 8. Кожен прихований елемент A_i ($i = \overline{1, q}$) підсумовує свої дельта-входи від елементів верхнього шару та визначає сумарну помилку на своєму вході

$$\delta_{inp.Ai} = U_{inp.Ai} = \sum_{k=1}^m \delta_k w_{ik}^2,$$

потім обчислює дельта-функцію на своєму виході

$$\delta_i = \delta_{inp.Ai} g'_i(U_{inp.Ai})$$

та збільшення ваг зв'язків w_{ji}^1 ($j = \overline{1, n}$, $i = \overline{1, q}$) та зміщень w_{0j}^1 ($j = \overline{1, n}$):

$$\Delta w_{ji}^1 = \alpha \delta_i S_j^p, \quad \Delta w_{0i}^1 = \alpha \delta_i.$$

Етап корекції ваг зв'язків та зміщень

Крок 9. Кожен вихідний елемент R_k ($k = \overline{1, m}$) адаптує своє зміщення та ваги:

$$w_{ik}^2 = w_{ik}^2 + \Delta w_{ik}^2, \quad i = \overline{0, q}, \quad k = \overline{1, m}.$$

Крок 10. Кожен прихований елемент A_i ($i = \overline{1, q}$) адаптує своє зміщення та ваги:

$$w_{ji}^1 = w_{ji}^1 + \Delta w_{ji}^1, \quad j = \overline{0, n}, \quad i = \overline{1, q}.$$

Крок 11. Перевіряються умови зупинки і, якщо вони не виконуються, то перехід до кроку 3 алгоритму.

Крок 12. Зупинка.

3.3. Проблеми модифікації та узагальнення основного алгоритму методу зворотного розповсюдження

Один із основних недоліків алгоритмів зворотного поширення помилки – повільна збіжність. У багатьох випадках для навчання багатошарової мережі потрібні сотні, тисячі і навіть десятки тисяч пред'явлень усієї навчальної безлічі зображень.

Інший недолік основного алгоритму пов'язаний з використанням градієнтного спуску поверхнею помилки виходу нейронної мережі. Поверхня помилки навіть для відносно простих нейронних мереж, що містять не більше 15 – 20 нейронів, має складний рельєф і складається з пагорбів та піків, ярів та западин, складок та долин у просторі високої розмірності. Алгоритм зворотного поширення легко знаходить точки локальних екстремумів, з яких він не здатний вибратися.

Невдачі в навчанні мережі можуть бути пов'язані і з попаданням в результаті корекції ваги зв'язків багатьох або всіх нейронів в області, де похідні функції активації дуже малі. Оскільки в процесі навчання величини приростів ваг зв'язків, що використовуються для адаптації нейронної мережі, пропорційні цим похідним, процес навчання мережі може практично зупинитися і настане "параліч мережі".

У зв'язку з зазначеними проблемами багатьма дослідниками було розроблено різні поліпшення, модифікації та узагальнення наведеного вище основного алгоритму методу зворотного поширення помилки [24, 25, 28, 33 – 38]. Число запропонованих алгоритмів занадто велике, щоб їх можна було охопити у

навчальному посібнику. Тому зупинимося лише на деяких перспективних розробках.

У роботах [25, 33, 38] описаний алгоритм, що використовує для адаптації ваг зв'язків нейронних мереж збільшення ваг, отриманих не тільки на поточному, але і на попередньому проході алгоритму:

$$w_{pq}^i(t+1) = w_{pq}^i(t) + \Delta w_{pq}^i(t) + \alpha_1 \Delta w_{pq}^i(t-1),$$

де α_1 – постійний коефіцієнт, що часто вибирається навколо 0,9 [25].

Алгоритм з такою корекцією ваг зв'язків дозволяє спускатися дном вузьких ярів поверхні помилки. Він дозволяє ефективно отримувати хороші рішення одних класів завдань, але може давати незначний або навіть негативний ефект при вирішенні інших.

У роботах [24, 37] описаний подібний алгоритм, заснований на експоненційному згладжуванні за рахунок використання як коефіцієнта α_1 експоненційно затухаючої функції.

Для визначення точніших оцінок величин приростів ваг зв'язків і, отже, прискорення збіжності алгоритмів запропоновано метод зворотного поширення помилки другого порядку, у якому передбачені процедури використання других похідних. На кожному проході алгоритму метод вимагає додаткового обсягу обчислень у порівнянні з основним алгоритмом, але через більш точний вибір прирощень ваг загальний обсяг обчислень, необхідних для навчання нейронних мереж, може бути значно меншим. Однак ефективніший цей метод при вирішенні далеко не всіх завдань. Наприклад, при навчанні розв'язання задачі ВИКЛЮЧНЕ АБО на одній із мереж методом зворотного поширення знадобилося 245 пред'явлень навчальної безлічі зображень і 4986 пред'явлень при використанні алгоритму зворотного розповсюдження помилки другого порядку [25].

Для боротьби з попаданням алгоритмів зворотного поширення в точки локальних мінімумів запропоновано методи, що поєднують статистичні підходи з градієнтним спуском у просторі помилки. Розглянемо один з них – метод комбінованого зворотного поширення з навчанням Коші. Збільшення ваг, що використовуються для адаптації нейронної мережі, у цьому методі складаються з двох компонентів:

$$w_{pq}^i(t+1) = w_{pq}^i(t) + \beta \Delta w_{pq}^i + (1-\beta) \Delta w_{pqk}^i,$$

де $w_{pq}^i(t+1)$, $w_{pq}^i(t)$ – значення ваги зв'язків відповідно в майбутній і поточний моменти часу між нейронами i -го та $(i+1)$ -го шарів; Δw_{pq}^i – приріст ваги, обчислений з використанням алгоритму зворотного поширення; Δw_{pqk}^i – збільшення ваги, що визначається алгоритмом навчання Коші; β – коефіцієнт, що задає відносний внесок кожного збільшення в загальне збільшення ваги зв'язку; при $\beta = 1$ маємо основний алгоритм зворотного поширення, а при $\beta = 0$ – алгоритм навчання Коші.

Статистичний алгоритм навчання Коші використовує ідеї, запозичені з фізичних процесів, що протікають при відпалі металів. Атоми металу, нагрітого

до температури, що перевищує його точку плавлення, знаходяться в сильному хаотичному русі, що перешкоджає виникненню низькоенергетичних станів. Однак при зниженні температури ймовірність появи таких станів збільшується, а оскільки кожен атом прагне стану з мінімумом енергії, то зрештою в металі досягається один з найнижчих можливих енергетичних станів. Під час відпалу металу розподіл різних енергетичних рівнів описується виразом

$$P(E) = e^{-E/kT}, \quad (3.4)$$

де $P(E)$ – ймовірність знаходження атома металу в стані з енергією E ; k – постійна Больцмана; T – температура металу в градусах Кельвіна.

Аналіз виразу (3.4) показує, що за високих значеннях температури ймовірність $P(E)$ появи всіх енергетичних станів близька до нуля, але в міру зниження температури ймовірність появи низькоенергетичних кристалічних решіток істотно вище в порівнянні з високоенергетичними станами.

За аналогією з описаним фізичним процесом відпалу металу було запропоновано алгоритм навчання Коші для нейронних мереж, який передбачає виконання наступних кроків:

Крок 1. Задається висока штучна початкова температура T_0 , яка з часом t зменшується відповідно до виразу

$$T(t) = \frac{T_0}{1+t}. \quad (3.5)$$

Крок 2. Множина невеликих випадкових значень ініціюються ваги всіх зв'язків.

Крок 3. Для кожної навчальної пари (вхідне зображення S^p ($p = \overline{1, L}$), необхідний вихідний вектор t^p) виконуються кроки 4 – 6.

Крок 4. Пред'являється вибране вхідне зображення S^p , розраховується вихідний вектор мережі та визначається цільова функція $F_p(w_0)$, де w_0 – множина ваг зв'язків, отриманих на попередньому кроці алгоритму.

Крок 5. Визначаються збільшення ваги зв'язків (залежно від різновиду алгоритму може змінюватися вага одного зв'язку, ваги зв'язків груп або шарів нейронів, всі ваги нейронної мережі) за допомогою співвідношення

$$\Delta w = \rho T(t) \text{tg}(P(\Delta w)), \quad (3.6)$$

де Δw – випадкове збільшення обраної ваги зв'язку; ρ – коефіцієнт швидкості навчання; $T(t)$ – температура, що визначається співвідношенням (3.5); $P(\Delta w)$ – випадкове число з інтервалу $(-\pi/2, \pi/2)$ з рівномірним розподілом.

Розраховується цільова функція з новими значеннями ваги зв'язків: $F(w_0 + \Delta w)$.

Крок 6. Якщо цільова функція покращилася ($F(w_0 + \Delta w) < F(w_0)$), то зберігаються нові значення ваги, якщо цільова функція погіршилася ($F(w_0 + \Delta w) > F(w_0)$), то нові значення ваг зберігаються з ймовірністю, яка визначається за допомогою розподілу Больцмана:

$$P(\Delta F) = e^{-\Delta F / kT}, \quad (3.7)$$

де ΔF – приріст цільової функції; k – коефіцієнт, вибирається в залежності від виду нейронної мережі та розв'язуваної задачі.

Для цього з інтервалу $[0, 1]$ з рівномірним розподілом чисел вибирається випадкове число q . Якщо $P(\Delta F) > q$, то зміна ваги зв'язків зберігається, якщо $P(\Delta F) \leq q$, то ваги набувають попередніх значень.

Крок 7. Для пред'явленої навчальної пари перевіряється умова досягнення заданого значення цільової функції. Якщо значення досягнуто та не виконуються умови припинення роботи алгоритму, то перехід до кроку 3 та пред'явлення наступної навчальної пари, якщо значення цільової функції не досягнуто – то зниження температури та перехід до кроку 4 алгоритму.

Шаг 8. Зупинка.

Зауваження 3.1. Наведений алгоритм носить ім'я Коші, оскільки співвідношення (3.6), що визначає випадкове збільшення ваги зв'язку, отримано з розподілу Коші:

$$P(x) = \frac{T(t)}{T(t)^2 + x^2},$$

де $P(x)$ – ймовірність збільшення величини x (при отриманні співвідношення (3.6) вважали, що $x = \Delta w$).

За рахунок випадкових кроків у напрямках, що погіршують цільову функцію, алгоритм Коші дозволяє вириватися з локальних мінімумів, де невеликий крок у будь-якому напрямку веде до погіршення цільової функції. Однак випадковий пошук, що виконується алгоритмом навчання Коші, має і суттєвий недолік, оскільки він може вимагати в сотні та тисячі разів більше машинного часу, ніж основний алгоритм зворотного розповсюдження.

Основний алгоритм зворотного розповсюдження дає комбінації методів переваги прямого пошуку – корекція ваги зв'язків з його допомогою завжди веде до поліпшення цільової функції і більш швидкого порівняння з випадковим пошуком визначення мінімумів цільової функції. Випадковий пошук методу Коші, хоч і вимагає на порядки більше машинного часу, ніж детермінований пошук, проте дозволяє визначати глобальний мінімум функції помилки, оскільки відповідно до виразу (3.6) можливі великі збільшення ваги зв'язків, тобто. переміщення у багатовимірному та багатоекстремальному просторі помилки на значні відстані. Крім того, при погіршенні цільової функції нові ваги не

відкидаються автоматично, а зберігаються з певною ймовірністю, яка визначається за допомогою розподілу Больцмана (3.7). Це дозволяє здійснювати пошук локальних мінімумів у всіх областях багатовимірного та багатоекстремального простору помилки, а інформація про всі знайдені локальні екстремуми та дозволяє визначити глобальний мінімум цільової функції (або знайти кращий набір вагових коефіцієнтів нейронної мережі)

3.4. Глибокі нейронні мережі

Архітектури глибоких нейронних мереж з'явилися в останні десятиліття двадцятого століття лише на десять років пізніше після появи перцептрону Розенблатта. Ряд вчених [39, 40] появу цих мереж пов'язують із роботами українського вченого О.Г. Івахненка [41 – 43]. У шістдесяті роки ХХ століття А.Г. Івахненко розробив багаторядний (багатоетапний) метод групового врахування аргументів (МГВА) або метод самоорганізації математичних моделей. У багатьох випадках у МГВА як моделі, які синтезувалися за безліччю експериментальних даних, використовувалися поліноми Колмогорова-Габора. При цьому всі вихідні дані ділилися принаймні на дві частини – навчальну та перевірочну послідовності. На першому ряду селекції на точках навчальної послідовності синтезувалося безліч моделей, що містять зазвичай не більше трьох-чотирьох одночленів полінома Колмогорова-Габора. Отримані моделі за допомогою критеріїв оцінювалися на точках перевірочної послідовності і найкращі з них пропускалися в наступний ряд селекції (термінологія МГВА), де з їх допомогою на точках навчальної послідовності з пар найкращих моделей першого ряду селекції синтезувалося безліч моделей другого ряду селекції, які оцінювали за допомогою критеріїв на точках перевірочної послідовності та кращі з них пропускалися в третій ряд селекції і т.д. Цей процес тривав доти, доки досягалось задане число рядів селекції. Оскільки це евристичний метод, він не завжди давав гарне рішення. В цьому разі процес синтезу моделей слідвало повторити з іншими параметрами. Це повторення могло бути багаторазовим.

Іншим джерелом ідей в архітектурах глибоких нейронних мереж з'явилися інтенсивні дослідження зорової кори тварин та людини. Розенблатт зробив лише перший крок у вивченні зорової системи тварин та людини. Фактично, коли він навчав дискретну нейронну мережу розпізнавання якихось зображень, він використовував матрицю чутливих елементів, кожен з яких міг бути в активному або загальмованому стані. Загалом ця матриця мала розміри $n \times m$ елементів, на входи яких подавалися зображення у вигляді векторів довжини $n \times m$. Такі вектори ніяк не враховували структуру зображення та різні його особливості. У той же час дослідження зорових систем тварин (зокрема, жаб і кішок) показували, що зображення сприймаються багато в чому не окремими чутливими клітинами ока, а групами клітин, які спільно з клітинами кори головного мозку тварин виконують набагато складніші функції, будучи детекторами частин зображень: крапок, плям, кутів, кордонів, ліній (світлих на темному тлі або темних на світлому тлі), рухи тощо. [44]. Такі дані мали рано чи пізно призвести до більш складних нейронних мереж, у яких мали враховуватися якимось чином фрагменти зображень. Причому цей облік має бути незалежним від конкретного розташування фрагментів

зображень. Першою такою нейронною мережею, в якій вдалося втілити ідеї нейронних мереж, вивчених у живих організмах, був неокогнітрон [30, 40, 45]. На жаль, минуло понад 20 років, перш ніж ідеї, вкладені в неокогнітрон, вдало втілювалися в нейронних мережах глибокого навчання.

Неокогнітрон має ієрархічну архітектуру (рис. 3.2), що складається з вхідного поля сенсорних елементів та ряду модулів (каскадів, stages) з пар S_i та C_i -шарів ($i = 1, 2, \dots, m$) нейронів [30]. Сигнали від вхідного зображення надходять на шар S_1 клітин першого модуля. Кожна з клітин складається з прямокутного поля чутливих елементів, необхідні розпізнавання конкретних фрагментів (ознак) вхідного зображення. У процесі навчання кожна клітина шару S_1 вчиться реагувати на конкретний фрагмент (ознаку) зображення незалежно від того, де він розташований на зображенні. Число клітин шару S_1 нейронів не повинно бути менше кількості різних фрагментів зображень, що розпізнаються. Шар C_1 також складається з прямокутних клітин, кожна з яких сприймає сигнали від кількох клітин шару S_1 . Ці клітини C_1 -шару можуть реагувати вже кілька фрагментів вхідного зображення. Тим самим збільшується впізнавана область вхідного зображення і можливість розпізнавати фрагменти, наприклад, кути, відрізки прямих і т.д. при різному їх розташуванні у зображенні. Групи клітин S_1 та C_1 шарів утворюють фактично двовимірний шар нейронів, кожен з яких реагує на певні фрагменти вхідного зображення. Кожен наступний двовимірний шар нейронів (шари S_2 та C_2 , S_3 та C_3 тощо) розпізнає вхідне зображення за все повнішою сукупністю його фрагментів. Така архітектура нейронної мережі дозволяє розробляти системи розпізнавання, які стійко працюють у умовах, коли зображення сильно спотворені чи зашумлені різними перешкодами, повернуті чи зсунуті проти еталонних. До недоліків систем розпізнавання на основі неокогнітрону необхідно віднести необхідність формування бібліотек фрагментів зображень, від яких сильно залежить якість функціонування систем розпізнавання.

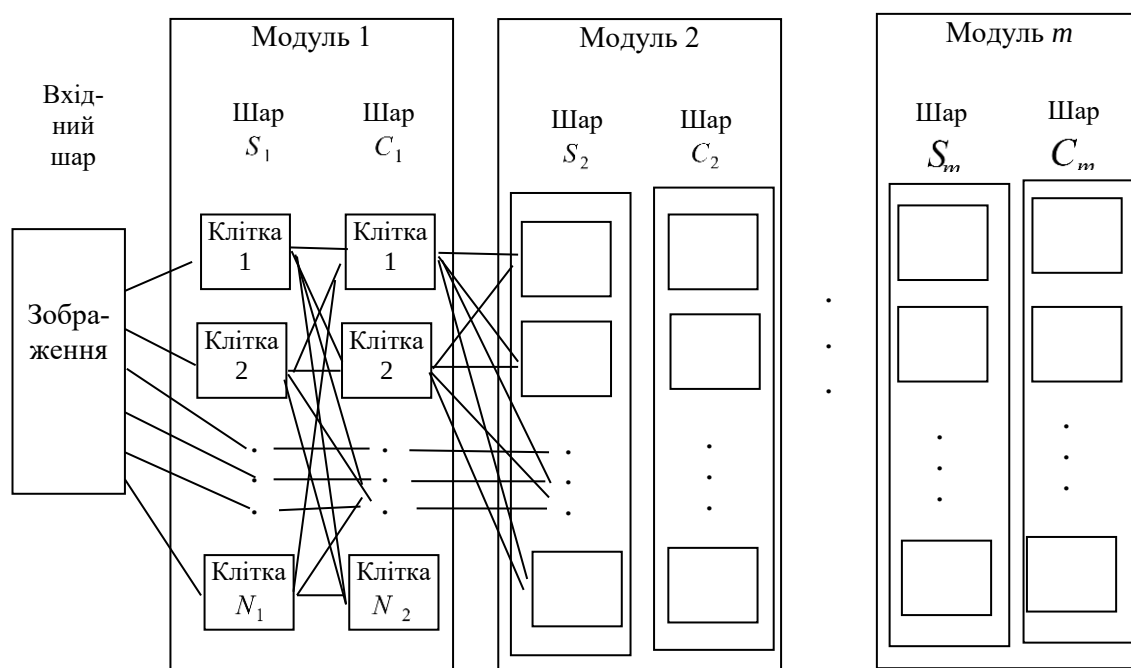


Рис. 3.2. Архітектура неокогнітрону

Багатомодульна (або багатокаскадна) архітектура неокогнітрону багато в чому копіює зорову кору ссавців, яка часто ділиться на сім (або більше) зорових зон від V1 до V6 та V7, які відрізняються один від одного і фізіологією, і функціями, і архітектурою [40]. Коротка інформація про ці зони наступна. Первинна зорова зона V1 виділяє невеликі фрагменти із зображень на сітківці ока. У зоні V2 продовжується виділення фрагментів зображення, узагальнюючи їх та враховуючи, що інформація надходить від двох очей. У зоні V3 виконується розпізнавання кольору та починає виконуватися сегментація та групування фрагментів. У зоні V4 починає виконуватися розпізнавання фрагментів зображень та простих геометричних фігур. Зона V5 займається розпізнаванням рухів об'єктів (або їх фрагментів), які спостерігаються на зображенні, що розглядається. У зоні V6 здійснюється корекція та узагальнення даних у зображенні внаслідок руху ссавця або змін по всьому полю зору. У зоні V7 людини на підставі інформації, отриманої від попередніх зон, виконується розпізнавання людських осіб та інших складних об'єктів, хоча це можуть робити інші ссавці. Ієрархія перелічених зон не є суворою, оскільки існують безлічі зв'язків, які можуть передавати інформацію, безпосередньо на більш високі рівні зорової кори. Наприклад, інформація із зони V1 може передаватися прямо до зони V5 і т.д. Крім того, в живому мозку є сильні зворотні зв'язки від вищих рівнів до нижчих.

Навіть цей короткий опис зорової кори показує, наскільки сильно біологічні нейронні мережі (або зорова кора людського мозку) відрізняються від штучної нейронної мережі неокогнітрону. Але навіть те небагато, що вивчено і зрозуміло у функціонуванні мозку, дозволяє створювати нейронні мережі та системи розпізнавання на їх основі, які не поступаються людині, а часто і перевершують її.

Сучасні нейронні мережі глибокого навчання мають цілу низку архітектур, які певною мірою задовольняють результати, отримані з допомогою неокогнітрона. Однак, на відміну від неокогнітрона, ці мережі самі виділяють ознаки чи фрагменти зображень. Розглянемо це виділення з прикладу однієї з класів мереж глибокого навчання – згорткових мереж [40]. Основна ідея цих нейронних мереж, як і неокогнітронів, полягає в тому, що обробка ознак або фрагментів зображень не повинна залежати від їх розташування на цих зображеннях. У згорткових нейронних мережах це досягається за рахунок того, що вихідне велике зображення (наприклад, картинка розмірами 128×128 пікселів) покривається за допомогою ковзного вікна (наприклад, $5 \times 5 = 25$ або $7 \times 7 = 49$ пікселів і т.д.). При кожному положенні ковзного вікна виділяються ті самі ознаки або фрагменти, і виходять матриці розмірами 5×5 (або 7×7). Потім результати роботи ковзного вікна можна замінити на нову "картинку", замінюючи кожну матрицю 5×5 , отриману за допомогою ковзного вікна на її центральні пікселі. Таким образом виходить перший згортковий шар. До нової матриці застосовують другий згортковий шар, який може бути з іншим ковзним вікном і т.д. [30].

При такому підході ознаки вхідних зображень виділяються автоматично та локально, і не вимагають участі людини.

Інтенсивний розвиток теорії глибоких нейронних мереж почався лише кілька років тому, але їх застосування у навчанні та розпізнаванні призвело до революційних результатів при вирішенні різноманітних завдань штучного інтелекту. Однак теорія цих мереж набагато складніша за такі відомі нейронні мережі як мережі Хопфілда, Хеммінга, Кохонена і т.д. Для її викладу та практичного застосування глибоких нейронних мереж потрібна окрема навчальна дисципліна та ознайомлення з розробленими бібліотеками для застосування глибоких нейронних мереж, а також детальний аналіз успішного застосування цих мереж при вирішенні конкретних завдань.

Контрольні питання

1. Запишіть формулу бінарної функції сигмоїдальної активації.
2. Яка область визначення бінарної сигмоїдальної функції активації?
3. Яка область значень бінарної функції сигмоїдальної активації?
4. Яка похідна бінарної сигмоїдальної функції активації?
5. Запишіть формулу біполярної функції сигмоїдальної активації.
6. Яка похідна біполярної сигмоїдальної функції активації?
7. Яка загальна ідея алгоритму зворотного розповсюдження помилки?
8. Які три основні етапи має метод зворотного розповсюдження помилки?
9. Запишіть основні кроки алгоритму методу зворотного розповсюдження помилки.
10. Які недоліки алгоритмів методу зворотного розповсюдження помилки?
11. Які найбільш відомі модифікації алгоритмів методу зворотного розповсюдження помилки?
12. Опишіть основну ідею статистичного алгоритму навчання Коші.

13. Навіщо поєднують два методи – метод зворотного розповсюдження та навчання Коші?
14. З яких кроків складається алгоритм навчання Коші для нейронних мереж?
15. У чому полягає основна ідея методу зворотного розповсюдження помилки другого порядку?
16. Що таке "параліч мережі"?
17. Який рельєф має поверхню помилки у методі зворотного розповсюдження навіть для відносно простих нейронних мереж?
18. Хто заклав теоретичні основи методу групового обліку аргументів та глибоких нейронних мереж?
19. З яких модулів складається зорова кора ссавців?
20. Нарисуйте архітектуру неокогнітрону.
21. З яких компонентів складаються модулі неокогнітрону?
22. Чим принципово відрізняються глибокі нейронні мережі від неокогнітрону?

НЕЙРОНІ МЕРЕЖІ, ОСНОВАНІ НА ЗМАГАННІ

Багато нейронних мереж використовують ідеї конкуренції між нейронами для посилення контрасту в активності нейронів. У більшості випадків або екстремальних ситуаціях, які часто називають "Переможцем отримує все" (Winner-Take-All), залишається лише один нейрон, що має найбільший вихідний сигнал. За способом виділення нейрона-переможця конкуруючі мережі можна розділити на два класи: мережі, ваги зв'язків яких залишаються фіксованими, та мережі, ваги зв'язків яких змінюються в ході ітераційного процесу виділення нейрона-переможця. Розглянемо спочатку три нейронні мережі, які представляють перший клас, та був – дві мережі другого класу

4.1. Maxnet

Загальновизнаної назви цієї мережі російською мовою немає. Найчастіше зустрічаються такі: мережу пошуку максимуму, максимізатор, мережу пошуку максимуму з конкуренцією за допомогою латерального гальмування (maxnet with competition through lateral inhibition).

Maxnet – приклад найпростішої нейронної мережі, заснованої на конкуренції та використовує механізм латерального гальмування. Вона часто використовується в інших нейронних мережах як модуль виділення нейронів, що мають найбільші вихідні сигнали. Архітектура мережі наведена на рис. 4.1

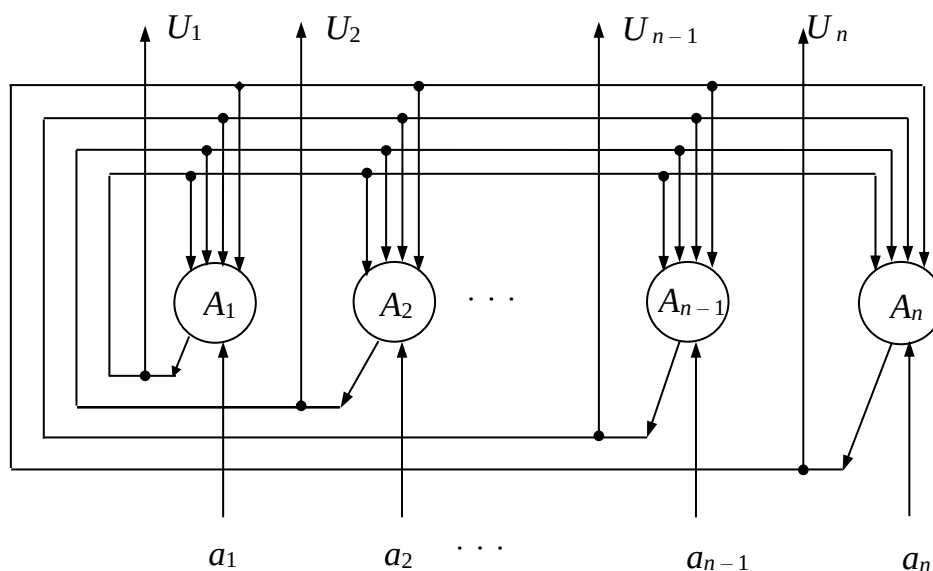


Рис. 4.1. Нейронна мережа Maxnet

На рисунку прийняті такі позначення: a_i , U_i – відповідно вхідний та вихідний сигнал нейрона A_i ($i = \overline{1, n}$). Функція активації нейронів A_i задається виразом

$$g(U_{inp}) = \begin{cases} U_{inp}, & \text{якщо } U_{inp} > 0, \\ 0, & \text{якщо } U_{inp} \leq 0. \end{cases} \quad (4.1)$$

Ваги зв'язків мережі визначаються співвідношенням

$$w_{ij} = \begin{cases} 1, & \text{якщо } i = j, \\ -\varepsilon, & \text{якщо } i \neq j, \end{cases} \quad i, j = \overline{1, n}, \quad (4.2)$$

де ε – константа, що задовольняє нерівності $0 < \varepsilon \leq 1/n$.

Мережа функціонує циклічно, динаміка нейронів описується ітераційним виразом

$$U_i(t+1) = g(U_i(t) - \varepsilon \sum_{j=1, j \neq i}^n U_j(t)), \quad i = \overline{1, n}, \quad (4.3)$$

за початкових умов

$$U_i(0) = a_i, \quad i = \overline{1, n}. \quad (4.4)$$

На кожній ітерації сигнал з виходу будь-якого нейрона відповідно до виразу (4.3) прагне придушити вихідні сигнали всіх інших нейронів, але спочатку більш сильні сигнали пригнічують найбільш слабкі, а потім триває конкуруюча взаємодія між рештою. Якщо на початку функціонування один з нейронів мав найбільший вхідний сигнал, то в результаті ітераційного процесу в мережі він тільки один і залишиться з вихідним сигналом, відмінним від нуля, тобто. стане "переможцем". Усі інші нейрони матимуть нульові вихідні сигнали.

Число циклів функціонування мережі заздалегідь невідоме. Критерієм зупинки ітераційного процесу в мережі є відсутність змін на виходах усіх нейронів мережі.

Мережа Махнет виділяє максимальний вхідний сигнал, проте в процесі функціонування втрачає саме значення цього сигналу, що є помітним недоліком мережі. Інший недолік пов'язаний з тим, що мережа розрахована на виділення лише одного максимального сигналу, і не дає жодного рішення, якщо на її входах є два або більше однакових максимальних сигналів. Ще одним недоліком мережі Махнет можна вважати квадратичне зростання числа зв'язків між нейронами зі збільшенням числа входів мережі.

Приклад 4.1. Розглянемо роботу мережі Махнет з п'ятьма нейронами, що мають активаційну функцію виду (4.1), та вагами зв'язків, що визначаються співвідношенням (4.2) при $\varepsilon = 0,2$. Нехай також вектор вхідних сигналів, має вигляд:

$$a_1 = U_1(0) = 0,30; \quad a_2 = U_2(0) = 0,60; \quad a_3 = U_3(0) = 0,70;$$

$$a_4 = U_4(0) = 0,90; \quad a_5 = U_5(0) = 0,80.$$

Результати функціонування мережі, розраховані за допомогою співвідношення (4.3), наведено в табл. 4.1.

Таблиця 4.1. Результати функціонування мережі Махnet із п'ятьма нейронами

Час	Величини вихідних сигналів нейронів				
	U_1	U_2	U_3	U_4	U_5
0	0,300	0,600	0,700	0,900	0,800
1	0	0,060	0,180	0,420	0,300
2	0	0	0,024	0,312	0,168
3	0	0	0	0,274	0,101
4	0	0	0	0,254	0,056
5	0	0	0	0,243	0,005
6	0	0	0	0,242	0

4.2. Нейронна мережа Mexican Hat

У перекладі російською мовою Mexican Hat означає "мексиканська капелюх". Така назва мережі пов'язана з функцією $w_{jk} = f(R)$ взаємодії нейронів мережі, що має вигляд мексиканської капелюхи (рис. 4.2) Тут прийняті такі позначення: j – обраний нейрон; R , w_{jk} – відповідно відстань та вага зв'язку між нейронами j та k .

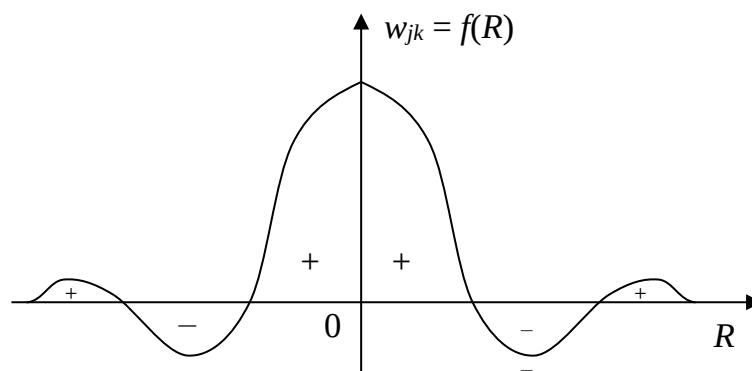


Рис. 4.2. Функція взаємодії нейронів мережі Mexican Hat

Наведений вид функції взаємодії нейронів відповідає деяким фізіологічним даним, за якими сусідні нейрони збуджують одне одного. Зі збільшенням відстані збудження зменшується і перетворюється на гальмування, яке, своєю чергою, змінюється слабкими збуджуючими зв'язками [24]. З виду функції $f(R)$ витікає, що у цій мережі більш загальної форми, ніж у Махнет, використовується взаємодія нейронів. Якщо Махнет застосовується лише конкуруючий вплив нейронів друг на друга, то Mexican Hat – конкуруючий і кооперативний.

Приклад схеми взаємодії нейронів для одновимірного випадку наведено на рис. 4.3 щодо вибраного нейрона A_j .

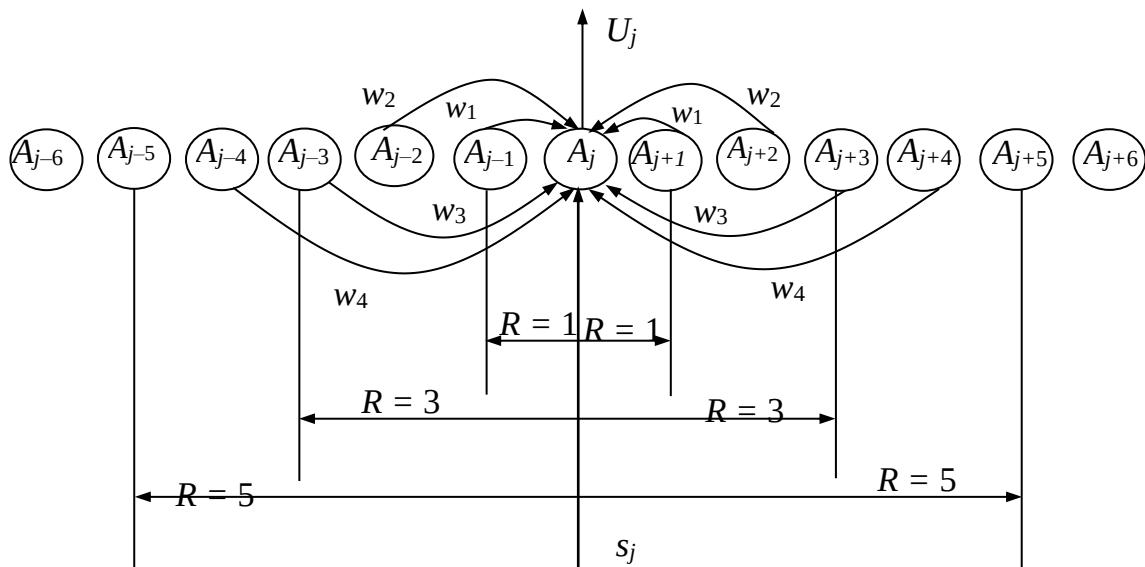


Рис. 4.3. Схема взаємодії нейронів у одновимірному випадку

Два найближчих до нейрона A_j сусіда зліва (A_{j-2} , A_{j-1}) і праворуч (A_{j+1} , A_{j+2}) пов'язані з ним збудливими зв'язками з вагами $w_{(j-2)j} = w_{(j+2)j} = w_2 > 0$, $w_{(j-1)j} = w_{(j+1)j} = w_1$, $w_1 > w_2$. Нейрони A_{j-4} , A_{j-3} , A_{j+3} , A_{j+4} перебувають у конкуруючій взаємодії з нейроном A_j і мають гальмуючі зв'язки з негативними вагами: $w_{j-3} = w_{j+3} = w_3 < 0$, $w_{j-4} = w_{j+4} = w_4 < 0$. У цьому прикладі архітектури нейронної мережі елементи A_{j-k} , A_{j+k} при $k \geq 5$ не взаємодіють із нейроном. Подібну топологію свого оточення, що складається із двох симетричних областей, має кожен нейрон у лінійній структурі. Її відмінності для окремих елементів можуть бути пов'язані тільки з обмеженою кількістю елементів у шарі і, отже, з тим, що крайні ліві та праві нейрони (у нашому випадку A_{j-6} , A_{j+6}), а також їхні найближчі сусіди (у цьому прикладі A_{j-5} , A_{j-4} і A_{j-3} для нейрона A_{j-6} , і A_{j+5} , A_{j+4} , A_{j+3} для нейрона A_{j+6}) не мають відповідно зліва і справа всіх або тільки частини сусідніх елементів, що на них впливають.

Величини областей кооперації (позитивні взаємодії) та конкуренції (негативні взаємодії) нейронної мережі можуть змінюватися, як можуть змінюватися і величини ваг позитивних та негативних зв'язків, а також топологія оточення нейрона (лінійна (рис. 4.3), прямокутна або гексагональна (рис 4.4) та і т.д.).

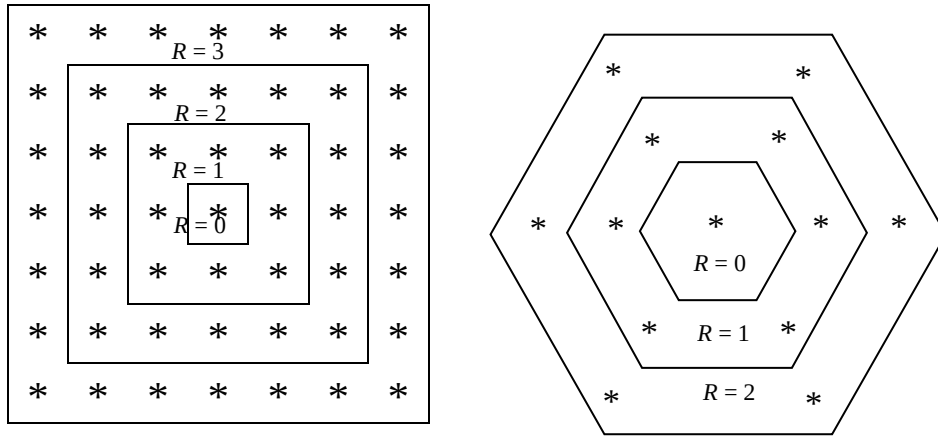


Рис. 4.4. Топологія оточення нейрона у двовимірному випадку

Контрастне посилення вхідного сигналу s_j нейрона A_j досягається за кілька ітерацій, при цьому величина вихідного сигналу U_j визначається як функція часу за допомогою співвідношення

$$U_j(t) = g_j[s_j(t) + \sum_{\substack{r=-k \\ r \neq 0}}^k w_{(j-r),j} U_{j-r}(t-1)] \quad (4.5)$$

з початковими умовами

$$U(0) = s(0), \quad (4.6)$$

де $U_j(t)$, $s_j(t)$ – відповідно вихідний та вхідний сигнал j -го нейрона в момент часу t ; g_j – функція активації нейрона A_j ; $U_{j-r}(t-1)$ – вихідний сигнал $(j-r)$ -го елемента в момент часу $(t-1)$; $j-r = \overline{j-k, j+k}$; $2k$ – число нейронів, що взаємодіють з нейроном j ; $w_{(j-r)j}$ – вага зв'язку між нейронами j та $j-r$ ($r = \overline{-k, k}$); $U(0) = (U_{j-k}(0), \dots, U_j(0), \dots, U_{j+k}(0))$, $s(0) = (s_{j-k}(0), \dots, s_j(0), \dots, s_{j+k}(0))$ – вектори вихідних та вхідних сигналів у момент часу $t = 0$.

Мережа працює без вчителя, вхідні сигнали і ваги зв'язків повністю визначають процес самоорганізації мережі, отже, і вихідні сигнали її нейронів. Критерієм зупинки ітераційного процесу в мережі є досягнення заданої кількості ітерацій.

Приклад 4.2. Розглянемо алгоритм роботи Mexican Hat для простої мережі з одинадцятьма нейронами, активаційна функція яких описується співвідношенням

$$g(U_{inp.}) = \begin{cases} 0, & \text{якщо } U_{inp.} < 0, \\ U_{inp.}, & \text{якщо } 0 \leq U_{inp.} \leq 2, \\ 2, & \text{якщо } U_{inp.} > 2, \end{cases} \quad (4.7)$$

Нехай загальний радіус галузі взаємодії нейрона A_j ($j = \overline{1,11}$) дорівнює чотирьом, і з позитивним підкріпленням – двом, у своїй ваги зв'язків рівні: $w_{(j-1)j} = w_{(j+1)j} = w_1 = 0,6$; $w_{(j-2)j} = w_{(j+2)j} = w_2 = 0,5$; $w_{(j-3)j} = w_{(j+3)j} = w_3 = -0,3$; $w_{(j-4)j} = w_{(j+4)j} = w_4 = -0,1$.

Вектор вхідного сигналу в різні моменти часу визначатиметься:

$$\begin{aligned} S(t=0) &= (0,0; 0,1; 0,2; 0,4; 0,6; 0,8; 0,6; 0,4; 0,2; 0,1; 0,0); \\ S(t>0) &= (0,0; 0,0; 0,0; 0,0; 0,0; 0,0; 0,0; 0,0; 0,0; 0,0; 0,0). \end{aligned} \quad (4.8)$$

Максимальна кількість ітерацій дорівнює трьом.

Використовуючи співвідношення (4.5), (4.6), розрахуємо вхідні та вихідні сигнали нейронів A_j ($j = \overline{1,11}$) при $t = 1$.

Оскільки при $t = 0$ вектор вхідного сигналу мережі визначається співвідношеннями (4.8), то маємо

$$U(0) = (0,0; 0,1; 0,2; 0,4; 0,6; 0,8; 0,6; 0,4; 0,2; 0,1; 0,0). \quad (4.9)$$

Використовуючи вирази (4.5) та (4.8) при $t = 1$ можна записати співвідношення, що визначають вхідні сигнали всіх нейронів мережі:

$$\begin{aligned} U_{inp.1}(t=1) &= w_1 U_2(0) + w_2 U_3(0) + w_3 U_4(0) + w_4 U_5(0); \\ U_{inp.2}(t=1) &= w_1 U_1(0) + w_1 U_3(0) + w_2 U_4(0) + w_3 U_5(0) + w_4 U_6(0); \\ U_{inp.3}(t=1) &= w_2 U_1(0) + w_1 U_2(0) + w_1 U_4(0) + w_2 U_5(0) + w_3 U_6(0) + w_4 U_7(0); \\ U_{inp.4}(t=1) &= w_3 U_1(0) + w_2 U_2(0) + w_1 U_3(0) + w_1 U_5(0) + w_2 U_6(0) + w_3 U_7(0) + \\ &\quad + w_4 U_8(0); \\ U_{inp.5}(t=1) &= w_4 U_1(0) + w_3 U_2(0) + w_2 U_3(0) + w_1 U_4(0) + w_1 U_6(0) + w_2 U_7(0) + \\ &\quad + w_3 U_8(0) + w_4 U_9(0); \\ U_{inp.6}(t=1) &= w_4 U_2(0) + w_3 U_3(0) + w_2 U_4(0) + w_1 U_5(0) + w_1 U_7(0) + w_2 U_8(0) + \\ &\quad + w_3 U_9(0) + w_4 U_{10}(0); \\ U_{inp.7}(t=1) &= w_4 U_3(0) + w_3 U_4(0) + w_2 U_5(0) + w_1 U_6(0) + w_1 U_8(0) + w_2 U_9(0) + \\ &\quad + w_3 U_{10}(0) + w_4 U_{11}(0); \\ U_{inp.8}(t=1) &= w_4 U_4(0) + w_3 U_5(0) + w_2 U_6(0) + w_1 U_7(0) + w_1 U_9(0) + w_2 U_{10}(0) + \\ &\quad + w_3 U_{11}(0); \\ U_{inp.9}(t=1) &= w_4 U_5(0) + w_3 U_6(0) + w_2 U_7(0) + w_1 U_8(0) + w_1 U_{10}(0) + w_2 U_{11}(0); \end{aligned}$$

$$U_{inp.10}(t = 1) = w_4 U_6(0) + w_3 U_7(0) + w_2 U_8(0) + w_1 U_9(0) + w_1 U_{11}(0);$$

$$U_{inp.11}(t = 1) = w_4 U_7(0) + w_3 U_8(0) + w_2 U_9(0) + w_1 U_{10}(0).$$

Використовуючи вектор вхідних сигналів (4.9) при $t = 0$ та задані значення ваг зв'язків $w_k (k = \overline{1,4})$, нескладно визначити і чисельні значення вхідних сигналів мережі при $t = 1$, а потім за допомогою функції активації нейронів (4.7) обчислити вектор вихідних сигналів. Дані цих розрахунків наведено у третьому та четвертому рядках табл. 4.2. Аналогічно обчислюються вектори вхідних і вихідних сигналів мережі при $t = 2$, $t = 3$ и $t = 9$. Чисельні значення цих розрахунків наведено у п'ятому – десятому рядках табл. 4.2.

Таблиця 4.2. Результати функціонування нейронної мережі Mexican Hat

Вектори вхідних та вихідних сигналів	Величини вхідних та вихідних сигналів нейронів мережі										
	1	2	3	4	5	6	7	8	9	10	11
$S(t = 0)$	0,00	0,10	0,20	0,40	0,60	0,80	0,60	0,40	0,20	0,10	0,00
$U(t = 0)$	0,00	0,10	0,20	0,40	0,60	0,80	0,60	0,40	0,20	0,10	0,00
$U_{ex.}(t = 1)$	-0,02	0,06	0,30	0,71	0,95	0,98	0,95	0,71	0,30	0,06	-0,02
$U(t = 1)$	0,00	0,06	0,30	0,71	0,95	0,98	0,95	0,71	0,30	0,06	0,00
$U_{ex.}(t = 2)$	-0,122	0,152	0,568	0,914	1,378	1,658	1,378	0,914	0,568	0,152	-0,122
$U(t = 2)$	0,000	0,152	0,568	0,914	1,378	1,658	1,378	0,914	0,568	0,152	0,000
$U_{ex.}(t = 3)$	-0,037	0,219	0,693	1,568	2,140	2,196	2,140	1,568	0,693	0,219	-0,037
$U(t = 3)$	0,000	0,219	0,693	1,568	2,000	2,000	2,000	1,568	0,693	0,219	0,000
$U_{ex.}(t = 9)$	1,400	2,240	3,000	3,180	2,860	2,800	2,860	3,180	3,000	2,240	1,420
$U(t = 9)$	1,400	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	1,420

На рис 4.5 за даними табл. 4.2 наведено графічну ілюстрацію зміни величини вихідних сигналів мережі Mexican Hat при різних кількості ітерацій.

При великій кількості ітерацій мережа може досягти стаціонарного стану, коли виходи більшості або навіть усіх нейронів набувають максимально можливих значень вихідних сигналів. Для прикладу архітектури мережі і наведених вихідних даних стаціонарний стан досягається при $t = 9$. Вхідні та вихідні сигнали нейронів мережі для цього випадку наведені у двох останніх рядках табл. 4.2.

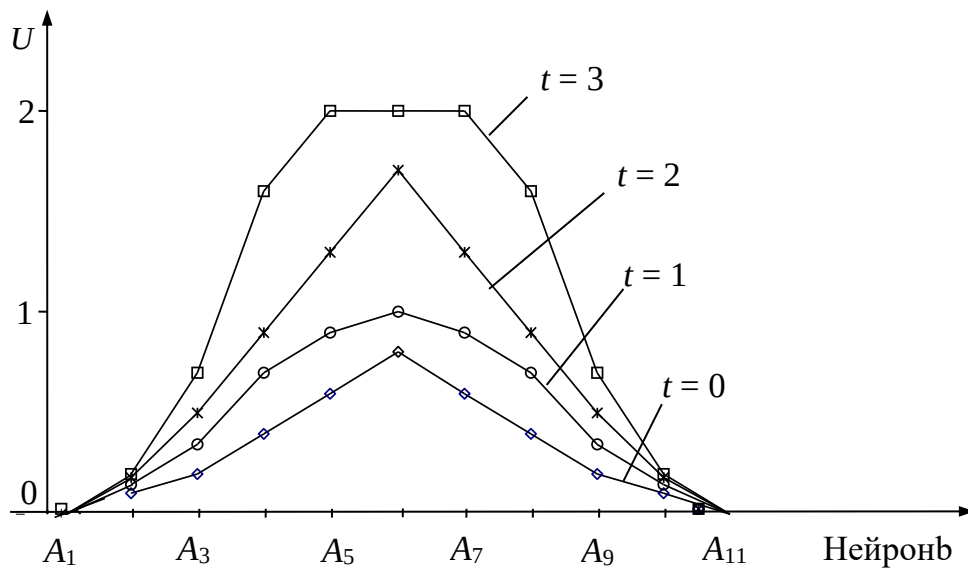


Рис. 4.5. Зміна вихідних сигналів нейронів мережі Mexican Hat за різних ітерацій

4.3. Мережа Хеммінга

На думку L. Fausett [28] мережа Хеммінга - це одна з найбільш перспективних нейронних мереж, що розпізнають і класифікують. У цій мережі чорно-білі зображення подаються як m -мірних біполярних векторів. Свою назву вона отримала від відстані Хеммінга, яка використовується в мережі в міру подібності R зображень вхідного та еталонних, що зберігаються за допомогою ваг мережі. Захід подібності визначається співвідношенням

$$R = m - R_x, \quad (4.10)$$

де m – число компонентів вхідного та еталонних векторів; R_x – відстань Хеммінгу між векторами.

Визначення 4.1. Відстанню Хеммінгу між двома двійковими векторами називається число компонент, у яких вектори різні.

Через визначення відстані Хеммінгу міра подібності зображень (4.10) може бути задана і як число a компонент двійкових векторів, в яких вони збігаються: $R = a$.

Запишемо для біполярних векторів $S = (s_1, \dots, s_m)$ та $Z = (z_1, \dots, z_m)$ їх скалярний добуток через число збігаються і відмінних компонент:

$$SZ = \sum_{i=1}^m s_i z_i = a - d, \quad (4.11)$$

де a – число однакових компонентів векторів; d – кількість різних компонентів

векторів S та Z .

Оскільки m – розмірність векторів, то $m = a + d$, отже, скалярний добуток (4.11) можна записати як

$$SZ = 2a - m.$$

Звідси нескладно отримати:

$$a = m/2 + \frac{SZ}{2} = \frac{m}{2} + \frac{1}{2} \sum_{i=1}^m s_i z_i. \quad (4.12)$$

Праву частину виразу (4.12) можна розглядати як вхідний сигнал нейрона, що має m синапсів з ваговими коефіцієнтами $z_i/2$ ($i = \overline{1, m}$) та зміщенням $m/2$. Синапси нейрона сприймають m компонент вхідного вектора $S = (s_1, \dots, s_m)$. Така інтерпретація правої частини виразу (4.12) призводить до архітектури нейронної підмережі, що зображена в нижній частині рис. 4.6. Одні автори (наприклад, [46]) мережу, зображену на рис 4.6, називають мережею Хеммінга, інші (наприклад, [28]) мережею Хеммінга називають лише її нижню частину, вважаючи, що наведена мережа і двох підмереж – Хеммінга і Махnet. Ми дотримуватимемося першої точки зору.

Мережа Хеммінга має m вхідних нейронів $S_1, \dots, S_m$, сприймають біполярні компоненти $s_1^q, \dots, s_m^q$ вхідних зображень S^q ($q = \overline{1, L}$). Вихідні сигнали S -елементів визначаються співвідношенням

$$U_{out.Si} = \begin{cases} +1, & \text{якщо } s_i^q = 1, \\ -1, & \text{якщо } s_i^q = -1, \end{cases} \quad (4.13)$$

тобто вихідний сигнал S -елемента повторює його вхідний сигнал:

$$U_{out.Si} = U_{inp.Si} = s_i^q$$

Кожен нейрон S_j ($j = \overline{1, m}$) пов'язаний із входом кожного елемента Z_k ($k = \overline{1, n}$). Вага цих зв'язків $w_{1k}, \dots, w_{mk}$ містять інформацію про k -е еталонне зображення $V^k = (v_1^k, \dots, v_m^k)$:

$$w_{1k} = v_1^k / 2, \dots, w_{mk} = v_m^k / 2. \quad (4.14)$$

Функції активації Z -елементів описується співвідношенням

$$g_Z(U_{inp}) = \begin{cases} 0, & \text{якщо } U_{inp} \leq 0, \\ k_1 U_{inp}, & \text{якщо } 0 \leq U_{inp} \leq U_n, \\ U_n, & \text{якщо } U_{inp} > U_n, \end{cases} \quad (4.15)$$

де $U_{inp.}$ – вхідний сигнал нейрона; k_1, U_n – константи.

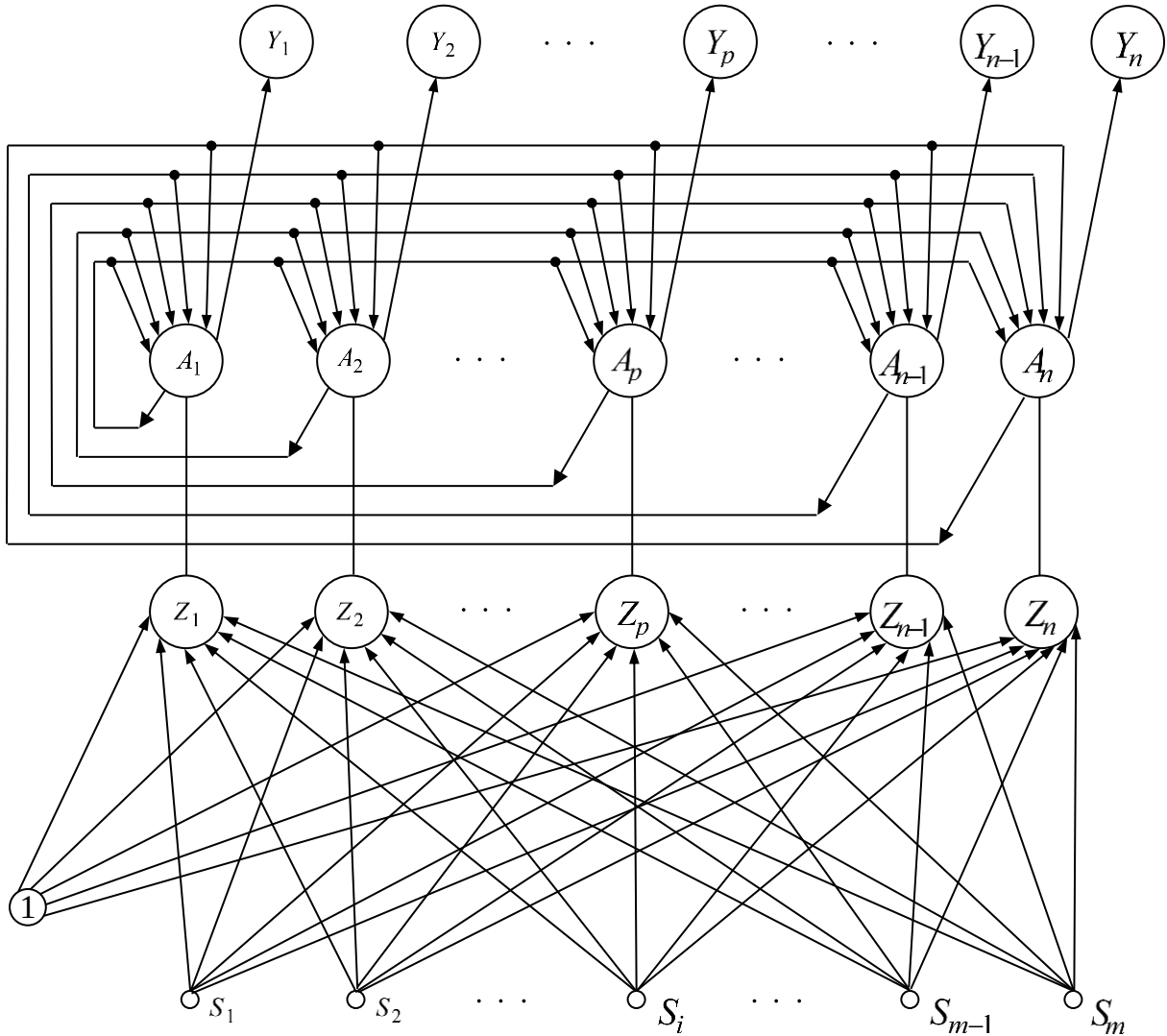


Рис. 4.6. Мережа Хеммінгу

При пред'явленні вхідного зображення $S^* = (s_1^*, \dots, s_m^*)$ кожен Z -нейрон розраховує свій вхідний сигнал відповідно до виразу виду (4.12):

$$U_{inp.Zk} = m/2 + \sum_{i=1}^m w_{ik} s_i^*, \quad (4.16)$$

та за допомогою функцій активації, визначає вихідний сигнал $U_{out.Zk}$. Вихідні сигнали $U_{out.Z1}, \dots, U_{out.Zn}$ Z -елементів є вхідними сигналами $a_1, \dots, a_n$ верхньої підмережі, якою є мережа Махнет, описана у підрозділі 4.1. Функції активації нейронів A_p ($p = \overline{1, n}$) та ваги їх зв'язків задаються співвідношеннями (4.1) та (4.2). Процес функціонування нейронів A_p описується виразом (4.3) за початкових умов

$$U_i(0) = a_i = U_{out.Zi}, \quad i = \overline{1, n}. \quad (4.17)$$

Якщо серед вхідних сигналів $a_1, \dots, a_n$ нейронів $A_1, \dots, A_n$ є один максимальний сигнал a_p ($p \in \{1, 2, \dots, n\}$), то в результаті ітераційного процесу в підмережі Махnet лише один нейрон A_p залишиться з вихідним сигналом, більшим за нуль, тобто. стане "переможцем". Оскільки вихідні сигнали $U_1, \dots, U_p, \dots, U_n$ А-елементів надходять на входи Y-нейронів, які мають функцію активації виду

$$g_Y(U_{inp}) = \begin{cases} 1, & \text{якщо } U_{inp} \geq 0, \\ 0, & \text{якщо } U_{inp} < 0, \end{cases} \quad (4.18)$$

то в результаті на виході мережі Хеммінга лише один нейрон Y_p виявиться з одиничним вихідним сигналом. Одиничний вихід цього нейрона та нульові всіх інших і будуть вказувати на те, що пред'явлене зображення $S^* = (s_1^*, \dots, s_m^*)$ найбільш близько, у сенсі заданої міри близькості (4.10), до еталонного зображення $V^p = (v_1^p, \dots, v_m^p)$.

Істотна перевага мережі Хеммінга полягає в тому, що вона не вимагає трудомістких обчислювальних процедур для свого навчання. Помітний недолік мережі: вона виділяє два і більше еталонних зображень, мають з пред'явленим однакові максимальні заходи близькості.

Приклад 4.3. Розробити мережу Хеммінга, що має як еталонні п'ять чорно-білих зображень $V^1, \dots, V^5$, наведених на рис. 4.7. Визначити реакцію мережі зображення, наведені на рис. 4.8.

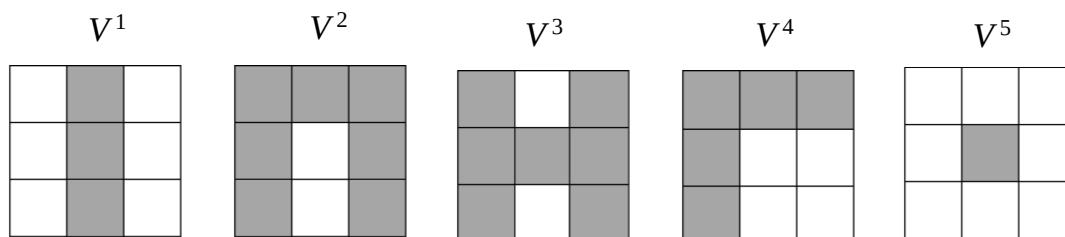


Рис. 4.7. Еталонні зображення

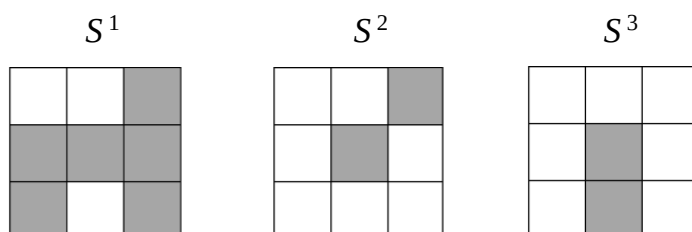


Рис. 4.8. Пред'явлені зображення

1	2	3
4	5	6
7	8	9

Рис. 4.9. Нумерація елементів зображень

Оскільки задано всього п'ять еталонних зображень, мережа повинна мати по п'ять Z-, A- та Y-нейронів. Наявність дев'яти чорно-білих елементів зображення рис. 4.7 та рис. 4.8 визначають дев'ять S-нейронів, що сприймають елементи вхідних зображень.

Пронумеруємо елементи зображення рис. 4.7 та 4.8 відповідно до рис. 4.9 і представимо зображення V^p ($p = \overline{1,5}$) у векторній формі за допомогою біполярного представлення векторів:

$$V^1 = (-1, 1, -1, -1, 1, -1, -1, 1, -1),$$

$$V^2 = (1, 1, 1, 1, -1, 1, 1, -1, 1),$$

$$V^3 = (1, -1, 1, 1, 1, 1, 1, -1, 1),$$

$$V^4 = (1, 1, 1, 1, -1, -1, 1, -1, -1),$$

$$V^5 = (-1, -1, -1, -1, 1, -1, -1, -1, -1).$$

Знаючи вектор еталонних зображень та їх число, за співвідношенням (4.14), розрахуємо матрицю $|W_{ik}|$ ($i = \overline{1,9}$, $k = \overline{1,5}$) ваг зв'язків нижньої підмережі мережі Хеммінга:

$$|W_{ik}| = \begin{matrix} & V^1(Z_1) & V^2(Z_2) & V^2(Z_3) & V^4(Z_4) & V^5(Z_5) \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \end{matrix} & \begin{vmatrix} -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ 0,5 & 0,5 & -0,5 & 0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ 0,5 & -0,5 & 0,5 & -0,5 & 0,5 \\ -0,5 & 0,5 & 0,5 & -0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ 0,5 & -0,5 & -0,5 & -0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & -0,5 & -0,5 \end{vmatrix} \end{matrix}, \quad (4.19)$$

де для наочності рядки та стовпці матриці пронумеровані відповідно за допомогою S-елементів та еталонних зображень V^p або нейронів Z_p , взятих у круглі дужки.

Зміщення $b_1, \dots, b_5$ Z-нейронів розраховуються за допомогою виразу (4.20):

$$b_1 = b_2 = \dots = b_5 = \frac{m}{2} = \frac{9}{2} = 4,5. \quad (4.20)$$

Функції активації Z-нейронів задамо співвідношенням (4.15) при $k_1 = 0,1$ та $U_n = 1/k_1 = 1/0,1 = 10$. Функції активації A- та Y-нейронів визначимо як функції (4.1) та (4.18). Константу ε , визначальну ваги негативних зв'язків у підмережі Maxnet,

знайдемо з рівності $\varepsilon = 1/n$, а так як $n = 5$, то $\varepsilon = 0,2$.

Знаючи всі параметри мережі Хеммінга, розглянемо її функціонування при пред'явленні зображення $S^1 = (-1, -1, 1, 1, 1, 1, 1, -1, 1)$ рис. 4.8.

Після пред'явлення зображення S^1 на виходах S -нейронів через те, що їх вихідні сигнали повторюють вхідні (співвідношення (4.13)), з'явиться вектор сигналів $S_{out.S} = S^1$. Використовуючи вихідні сигнали S -елементів, кожен Z -нейрон розраховує свій вхідний сигнал відповідно до виразу (4.16), матриці ваг (4.19) та зсуву b_k , $k = \overline{1,5}$ (4.20):

$$U_{inp.Z1} = \frac{9}{2} + \sum_{i=1}^9 w_{i1} S_i^1 = 4,5 + (-0,5) \cdot (-1) + 0,5 \cdot (-1) \cdot (-0,5) \cdot 1 + (-0,5) \cdot 1 + 0,5 \cdot 1$$

$$U_{inp.Z2} = \frac{9}{2} + \sum_{i=1}^9 w_{i2} S_i^1 = 4,5 + 0,5 \cdot (-1) + 0,5 \cdot (-1) + 0,5 \cdot 1 + (-0,5) \cdot 1 + 0,5 \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot (-1) + 0,5 \cdot 1 = 6,$$

$$U_{inp.Z3} = \frac{9}{2} + \sum_{i=1}^9 w_{i3} S_i^1 = 4,5 + 0,5 \cdot (-1) + (-0,5) \cdot (-1) + 0,5 \cdot 1 + 0,5 \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot (-1) + 0,5 \cdot 1 = 8,$$

$$U_{inp.Z4} = \frac{9}{2} + \sum_{i=1}^9 w_{i4} S_i^1 = 4,5 + 0,5 \cdot (-1) + 0,5 \cdot (-1) + 0,5 \cdot 1 + (-0,5) \cdot 1 + (-0,5) \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot (-1) + (-0,5) \cdot 1 = 4,$$

$$U_{inp.Z5} = \frac{9}{2} + \sum_{i=1}^9 w_{i5} S_i^1 = 4,5 + (-0,5) \cdot (-1) + (-0,5) \cdot (-1) + (-0,5) \cdot 1 + (-0,5) \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot 1 + (-0,5) \cdot (-1) + (-0,5) \cdot 1 = 4.$$

За вхідним сигналом $U_{ex.Zk}$, використовуючи свою функцію активації (4.15) при $k_1 = 0,1$ и $U_n = 10$, кожен Z -нейрон розраховує свій вихідний сигнал:

$$U_{out.Z1} = k_1 U_{inp.Z1} = 0,1 \cdot 2 = 0,2,$$

$$U_{out.Z2} = k_1 U_{inp.Z2} = 0,1 \cdot 6 = 0,6,$$

$$U_{out.Z3} = k_1 U_{inp.Z3} = 0,1 \cdot 8 = 0,8,$$

$$U_{out.Z4} = k_1 U_{inp.Z4} = 0,1 \cdot 4 = 0,4,$$

$$U_{out.Z5} = k_1 U_{inp.Z5} = 0,1 \cdot 4 = 0,4.$$

Вектор

$$U_{out.Z} = (0,2; 0,6; 0,8; 0,4; 0,4) \quad (4.21)$$

є вхідним вектором підмережі Махnet, яка починає ітераційний процес виділення максимального вихідного сигналу за допомогою виразів (4.1) та (4.3) за початкових умов (4.21). Для $t = 1$ відповідно до виразів (4.3) та (4.1) маємо

$$U_{out.A1}(1) = g(U_{out.A1}(0) - \varepsilon \sum_{k=2}^5 U_{out.Ak}(0)) = g(0,2 - 0,2(0,6 + 0,8 + 0,4 + 0,4)) = \\ = g(-0,24) = 0,$$

$$U_{out.A2}(1) = g(U_{out.A2}(0) - \varepsilon \sum_{k=1, k \neq 2}^5 U_{out.Ak}(0)) = g(0,6 - 0,2(0,2 + 0,8 + 0,4 + 0,4)) \\ = g(0,24) = 0,24,$$

$$U_{out.A3}(1) = g(U_{out.A3}(0) - \varepsilon \sum_{k=1, k \neq 3}^5 U_{out.Ak}(0)) = g(0,8 - 0,2(0,2 + 0,6 + 0,4 + 0,4)) \\ = g(0,48) = 0,48.$$

$$U_{out.A4}(1) = g(U_{out.A4}(0) - \varepsilon \sum_{k=1, k \neq 4}^5 U_{out.Ak}(0)) = g(0,4 - 0,2(0,2 + 0,6 + 0,8 + 0,4)) \\ = g(0) = 0,$$

$$U_{out.A5}(1) = g(U_{out.A5}(0) - \varepsilon \sum_{k=1, k \neq 5}^5 U_{out.Ak}(0)) = g(0,4 - 0,2(0,2 + 0,6 + 0,8 + 0,4)) = \\ = g(0) = 0.$$

Використовуючи вектор $U_{out.A1}(1), \dots, U_{out.A5}(1)$ вихідних сигналів A -елементів при $t = 1$ та вирази (4.3) та (4.1), аналогічним чином розраховують вихідні сигнали A -нейронів при $t = 2, 3$ та 4 . Результати розрахунків наведено у табл. 4.3.

Таблиця 4.3. Результати розрахунків ітераційного процесу у підмережі Махnet

Час	Величина вихідних сигналів нейронів A_k ($k = \overline{1,5}$)				
	$U_{out.A1}$	$U_{out.A2}$	$U_{out.A3}$	$U_{out.A4}$	$U_{out.A5}$
0	0,200	0,600	0,800	0,400	0,400
1	0,000	0,240	0,480	0,000	0,000
2	0,000	0,144	0,432	0,000	0,000
3	0,000	0,058	0,403	0,000	0,000
4	0,000	0,000	0,402	0,000	0,000

Ітераційний процес у підмережі Махnet закінчується при $t = 5$, оскільки на цьому кроці функціонування підмережі не змінює жодного вихідного сигналу A -елементів. Вектор вихідних сигналів A -елементів записаний в останньому рядку

табл. 4.3 надходить на входи Y -елементів.

Так як Y -нейрони мають функцію активації виду (4.18), то на виході лише одного елемента Y_3 з'явиться одиничний сигнал. Поява цього сигналу свідчить, що пред'явлене зображення S^1 найбільш близько до еталонного зображення V^3 . Візуальне зіставлення рис. 4.7 та 4.8 підтверджує правильність роботи мережі.

Визначимо тепер реакцію мережі при пред'явленні зображення $S^2 = (-1, -1, 1, -1, 1, -1, -1, -1)$. Оскільки розрахунки аналогічні, то наведемо лише основні проміжні результати:

$$U_{inp.Z1}(S^2) = 6, U_{inp.Z2}(S^2) = 2, U_{inp.Z3}(S^2) = 4, U_{inp.Z4}(S^2) = 4, U_{inp.Z5}(S^2) = 8;$$

$$U_{out.Z1}(S^2) = a_1 = 0,6, U_{out.Z2}(S^2) = a_2 = 0,2,$$

$$U_{out.Z3}(S^2) = U_{out.Z4}(S^2) = a_3 = a_4 = 0,4,$$

$$U_{out.Z5}(S^2) = a_5 = 0,8.$$

Оскільки вхідний вектор $(0,6; 0,2; 0,4; 0,4; 0,8)$ підмережі Махнет містить єдиний максимальний елемент $a_5 = 0,8$, то в результаті ітераційного процесу, що визначається співвідношеннями (4.3), (4.1), на виході лише елемента A_5 виявиться позитивний вихідний сигнал, який викликає одиничний сигнал на виході нейрона Y_5 . Отже, пред'явлене зображення найближче до еталонного зображення V^5 , що підтверджує і візуальне зіставлення рис. 4.7 та 4.8.

Визначимо тепер реакцію мережі Хеммінга на вхідне зображення $S^3 = (-1, -1, -1, 1, -1, -1, 1, -1)$ (рис. 4.8). При пред'явленні зображення S^3 маємо:

$$U_{inp.Z1}(S^3) = 8, U_{inp.Z2}(S^3) = 0, U_{inp.Z3}(S^3) = 1, U_{inp.Z4}(S^3) = 2, U_{inp.Z5}(S^3) = 8.$$

Оскільки сигнали $U_{inp.Z1}(S^3) = U_{inp.Z5}(S^3)$ є однаковими максимальними вхідними сигналами, однак однаковими будуть і максимальні сигнали на виходах Z -елементів $U_{inp.Z1}(S^3) = U_{out.Z5}(S^3) = 0,8$ та на входах A -нейронів

$$a_1(S^3) = a_5(S^3) = 0,8,$$

отже, підмережа Махнет не зможе виділити єдиного максимального сигналу і в результаті її функціонування на всіх виходах A -і Y -нейронів з'являться нульові сигнали.

Таким чином, мережа Хеммінга не може визначити, до якого з еталонних зображень найближче пред'явлене зображення S^3 .

4.4. Нейронна мережа Хеммінгу, здатна визначати кілька рішень

Розробимо нейронну мережу, що використовує відстань Хеммінга і класифікуючу зображення, що знаходяться на мінімальній відстані Хеммінга від одного, двох або трьох еталонних зображень зв'язків нейронної мережі, що зберігаються у вагах. Архітектура цієї мережі наведено на рис. 4.10.

Кожен нейрон S_i ($i = 1, \dots, n$) пов'язаний із входом кожного елемента Z_p ($p = 1, \dots, m$) зв'язками з вагами w_{ip} . Ваги зв'язків $w_{1p}, \dots, w_{np}$ містять інформацію о p -м еталонном зображенні $V^p = (v_1^p, \dots, v_n^p)$:

$$w_{1p} = v_1^p / 2, \dots, w_{np} = v_n^p / 2.$$

При пред'явленні вхідного зображення $S^* = (S_1^*, \dots, S_m^*)$ кожен Z -нейрон розраховує свій вхідний та вихідний сигнали. Вихідні сигнали $U_{out Z_1}, \dots, U_{out Z_m}$ Z -елементів є вхідними сигналами $a_1, \dots, a_m$ підмережі Махnet.

Мережа функціонує циклічно, динаміка нейронів описується ітераційним співвідношенням (4.3) за початкових умов (4.4). Вихідні сигнали нейронів у результаті ітераційної процедури поступово зменшуються.

Якщо серед вхідних сигналів $a_1, \dots, a_m$ нейронів $A_1, \dots, A_m$ є один максимальний сигнал a_p ($p \in \{1, 2, \dots, m\}$), то в результаті ітераційного процесу в підмережі Махnet лише один нейрон A_p залишиться з вихідним сигналом, більшим за нуль.

Якщо максимальних сигналів два або більше, то на виходах всіх нейронів у результаті ітераційного процесу будуть нульові вихідні сигнали. Для подолання нестачі мережі Хеммінгу необхідно до закінчення ітераційного процесу виділяти однакові максимальні вихідні сигнали за допомогою додаткових нейронів.

Архітектура запропонованої нейронної мережі відрізняється від архітектури відомої мережі Хеммінга, наявністю двох додаткових шарів нейронів: X -шару та Σ -шару (рис. 4.10).

Нейрони X - та Σ -шарів мають функцію активації виду

$$g(U_{inp}) = \begin{cases} 1, & \text{якщо } U_{inp} \geq 0, \\ 0, & \text{якщо } U_{inp} < 0, \end{cases} \quad (4.21)$$

де U_{inp} – вхідний сигнал нейрона, що має функцію активації $g(U_{inp})$.

Вхідні сигнали $U_{inp X_p}$ ($p = 1, \dots, m$), нейронів X -шару описуються наступним виразом:

$$U_{inp X_p} = U_{out X_p} W_{X_p X_p} + U_{out X_p} W_{Y_p X_p} + \sum_{i=0}^3 W_{\Sigma_i X_p} U_{out \Sigma_i} + W_{0p}^3 \cdot 1 \quad (4.22)$$

де $U_{out X_p}, U_{out Y_p}$ ($p = 1, \dots, m$) – відповідно вихідні сигнали нейронів X - та Y шарів; $W_{X_p X_p}$ – вага зворотного зв'язку нейрона X_p ($p = 1, \dots, m$); $W_{Y_p X_p}$ – вага зв'язку з виходу нейрона Y_p на вхід нейрона X_p ($p = 1, \dots, m$); $W_{\Sigma_i X_p}$ – вага зв'язку від нейрона Σ_i ($i = 0, 1, 2, 3$) до нейрону X_p ($p = 1, \dots, m$); $U_{out \Sigma_i}$ ($i = 0, 1, 2, 3$) –

вихідні сигнали нейронів Σ -шару; W_{0p}^3 – вага зв'язку сигналу зміщення нейрона X_p ($p = 1, \dots, m$).

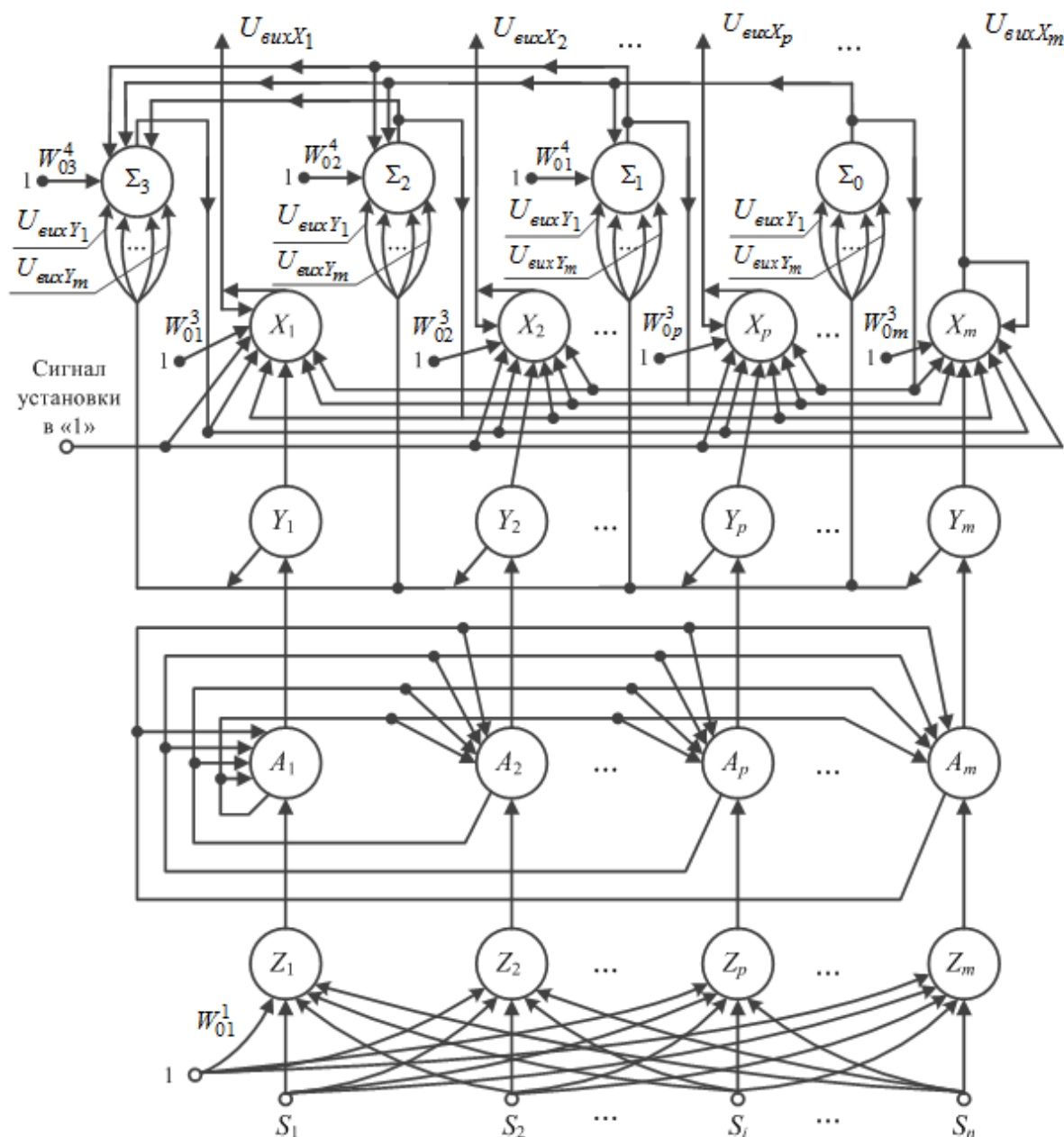


Рис. 4.10. Нейронна мережа, яка використовує відстань Хеммінга та класифікує зображення, що знаходяться на мінімальній відстані від одного, двох або трьох еталонних зображень

Вхідні сигнали нейронів Σ -шару описуються співвідношеннями:

$$U_{inp\Sigma_0} = \sum_{p=1}^m U_{outY_p} W_{Y_p} W_{Y_p\Sigma_0}; \quad (4.23)$$

$$U_{inp\Sigma_1} = \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_1} + U_{out\Sigma_0} W_{\Sigma_0\Sigma_1} + W_{01}^4 \cdot 1; \quad (4.24)$$

$$U_{inp\Sigma_2} = \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_2} + U_{out\Sigma_1} W_{\Sigma_1\Sigma_2} + U_{out\Sigma_0} W_{\Sigma_0\Sigma_2} + W_{02}^4 \cdot 1; \quad (4.25)$$

$$U_{inp\Sigma_3} = \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_3} + U_{out\Sigma_2} W_{\Sigma_2\Sigma_3} + U_{out\Sigma_1} W_{\Sigma_1\Sigma_3} + \\ + U_{out\Sigma_0} W_{\Sigma_0\Sigma_3} + W_{03}^4 \cdot 1 \quad (4.26)$$

де U_{outX_p} – вихідний сигнал нейрона Y_p , $p=1, \dots, m$; $W_{Y_p\Sigma_j}$ – вага зв'язку від нейрона Y_p до нейрону Σ_j , $W_{Y_p\Sigma_j} = -1$, $p=1, \dots, m$, $j=0, 1, 2, 3$; $U_{out\Sigma_j}$ – вихідний сигнал нейрона Σ_j , $j=0, 1, 2, 3$; $W_{\Sigma_k\Sigma_{k+1}}$ – вага зв'язку від нейрона Σ_k до нейрона Σ_{k+1} , $k=0, 1, 2$; W_{0j}^4 – вага зв'язку сигналу зміщення нейрона Σ_j , $j=1, 2, 3$, $W_{0j}^4 = j$.

Якщо на виходах чотирьох або більше A -нейронів є позитивні вихідні сигнали, які викликають одиничні вихідні сигнали на виходах відповідних Y -нейронів, то сигнали $U_{inp\Sigma_1}$, $U_{inp\Sigma_2}$, $U_{inp\Sigma_3}$ – негативні. Відповідно до функції активації (4.21) на виходах нейронів Σ_1 , Σ_2 , Σ_3 будуть нульові вихідні сигнали. Якщо на виходах лише трьох A -нейронів будуть позитивні сигнали, наприклад, у нейронів A_{p1} , A_{p2} і A_{p3} , то тоді відповідно до виразу (4.26) маємо: $U_{inp\Sigma_3} = 0$.

При нульовому входньому сигналі $U_{inp\Sigma_3} = 0$ та функції активації нейрона Σ_3 виду (4.21) на виході нейрона з'явиться одиничний вихідний сигнал $U_{out\Sigma_3} = 1$, який надійде на входи всіх нейронів X -шару. Відповідно до виразу (4.22) на входах X -нейронів будуть наступні входні сигнали: $U_{inpX_p} = -1$, $p=1, \dots, m$, $p \neq p_1, p_2, p_3$, $U_{inpX_p} = 0$, $p = p_1, p_2, p_3$.

Таким чином, на виходах нейронів X_p ($p=1, \dots, m$, $p \neq p_1, p_2, p_3$) будуть нульові вихідні сигнали, а на виходах нейронів X_{p1} , X_{p2} , X_{p3} будуть поодинокі вихідні сигнали.

З появою на виходах нейронів X_{p1} , X_{p2} , X_{p3} одиничних сигналів ці сигнали по ланцюгу зворотного зв'язку з вагою $W_{X_pX_p} = 2$ ($p = p_1, p_2, p_3$) надходять на входи цих нейронів. Сигнали зворотного зв'язку цих нейронів підтримуватимуть невід'ємні входні сигнали навіть тоді, коли виконуватимуться умови $U_{out\Sigma_3} = 0$; $U_{outY_{p1}} = U_{outY_{p2}} = U_{outY_{p3}} = 0$. Таким чином, якщо входне зображення знаходиться на однаковій мінімальній відстані Хеммінгу від трьох еталонних зображень, що зберігаються у вагах нейронів Z_{p1} , Z_{p2} , Z_{p3} , то вихідні

сигнали нейронів A_{p1} , A_{p2} , A_{p3} одночасно стають рівними нулю. При цьому $U_{outA_{p1}} = U_{outA_{p2}} = U_{outA_{p3}} = U_{out\Sigma_3} = 0$ и $U_{outX_{p1}} = U_{outX_{p2}} = U_{outX_{p3}} = 1$. Отже, на виході нейронної мережі залишаються поодинокі сигнали нейронів, що вказують від яких трьох зображень знаходиться вхідне зображення на мінімальній відстані.

Якщо сигнали на виходах A -нейронів не одночасно стають рівними нулю, наприклад, спочатку на виході одного нейрона A_p ($p = p_1, p_2, p_3$) з'являється нульовий сигнал, наприклад, $U_{outA_{p3}} = 0$, тоді одиничний сигнал на виході нейрона X_{p3} повинен бути скинутий у нуль. Це досягається наступним чином:

1. В соответствии с выражением (4.25) срабатывает нейрон Σ_2 : $U_{inp\Sigma_2} = 0$ и $U_{out\Sigma_2} = 1$.

2. Вихідний сигнал нейрона Σ_2 загальмовує нейрон Σ_3 , оскільки за співвідношенням (4.26) при $W_{\Sigma_2\Sigma_3} = -4$ маємо $U_{inp\Sigma_3} = -3$ и $U_{out\Sigma_3}(-3) = 0$.

3. Відповідно до виразу (4.22) вхідний сигнал нейрона X_{p3} буде негативним, а вихідний – нульовим.

4. Відповідно до виразу (4.22) вхідні сигнали нейронів X_{p1} , X_{p2} будуть невід'ємними, а вихідні – одиничними.

5. Якщо сигнали $U_{outA_{p1}}$ и $U_{outA_{p2}}$ одночасно стануть нульовими, то на виходах нейронів X_{p1} , X_{p2} залишаться поодинокі вихідні сигнали, що вказують на те, що вхідне зображення знаходиться на однаковій відстані Хеммінга від еталонних зображень, що зберігаються у вагах зв'язків нейронів Z_{p1} , Z_{p2} .

Якщо сигнали $U_{outA_{p1}}$ и $U_{outA_{p2}}$ не звертаються одночасно на нуль, а спочатку, наприклад, стає нульовим сигнал $U_{outA_{p2}}$, то в цьому випадку одиничний сигнал залишиться тільки на виході нейрона X_{p1} , вказуючи, що вхідне зображення знаходиться на мінімальній відстані Хеммінга від еталонного зображення, що зберігається у вагах зв'язків нейрона Z_{p1} . Справді, у цьому випадку маємо:

$$\begin{aligned}
 U_{outA_{p1}} &\neq 0; U_{outX_p} = 0, p = 1, \dots, m, p \neq p_1; \\
 U_{inp\Sigma_0} &= \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_0} = U_{outY_{p1}} W_{Y_{p1}\Sigma_0} = 1 \cdot (-1) = -1; \\
 U_{out\Sigma_0} &= g(U_{inp\Sigma_0} = -1) = 0; \\
 U_{inp\Sigma_1} &= \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_1} + U_{out\Sigma_0} W_{\Sigma_0\Sigma_1} + W_{01}^4 \cdot 1 = 0; \\
 U_{out\Sigma_1} &= g(U_{inp\Sigma_1} = 0) = 1;
 \end{aligned}$$

$$U_{inp\Sigma_2} = \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_2} + U_{out\Sigma_1} W_{\Sigma_1\Sigma_2} + U_{out\Sigma_0} W_{\Sigma_0\Sigma_2} + W_{02}^4 \cdot 1 = -1;$$

$$U_{out\Sigma_2} = g(U_{inp\Sigma_2} = -1) = 0;$$

$$U_{inp\Sigma_3} = \sum_{p=1}^m U_{outY_p} W_{Y_p\Sigma_3} + U_{out\Sigma_2} W_{\Sigma_2\Sigma_3} + U_{out\Sigma_1} W_{\Sigma_1\Sigma_3} + \\ + U_{out\Sigma_0} W_{\Sigma_0\Sigma_3} + W_{03}^4 \cdot 1 = -1;$$

$$U_{out\Sigma_3} = g(U_{inp\Sigma_3} = -1) = 0;$$

$$U_{inpX_p} = \sum_{i=0}^3 U_{out\Sigma_i} W_{\Sigma_i X_p} + U_{outX_p} W_{X_p X_p} + U_{outY_p} W_{Y_p Y_p} + W_{0p}^3 \cdot 1 = -1;$$

$$U_{outX_p} = g(U_{inpX_p} = -1) = 0;$$

$$U_{inpX_{p_1}} = \sum_{i=0}^3 U_{out\Sigma_i} W_{\Sigma_i X_{p_1}} + U_{outX_{p_1}} W_{X_{p_1} X_{p_1}} + U_{outY_{p_1}} W_{Y_{p_1} Y_{p_1}} + W_{0p_1}^3 \cdot 1 = 2;$$

$$U_{outX_{p_1}} = g(U_{inpX_{p_1}} = 2) = 1.$$

де $W_{\Sigma_0\Sigma_1} = -2$; $W_{\Sigma_0\Sigma_2} = -3$; $W_{\Sigma_0\Sigma_3} = -4$; $W_{\Sigma_1\Sigma_2} = -2$; $W_{\Sigma_1\Sigma_3} = -3$.

Отже, у цьому випадку тільки на виході одного нейрона X -шару буде одиничний вихідний сигнал, що вказує на те, що вхідне зображення знаходиться на мінімальній відстані Хеммінгу від еталонного зображення, що зберігається у вагах нейрона Z_{p1} .

Таким чином, розроблена нейронна мережа, що використовує відстань Хеммінга при розпізнаванні чорно-білих зображень і здатна розпізнавати зображення, що знаходяться на мінімальній відстані Хеммінга від одного, двох або трьох еталонних зображень, що зберігаються у вагах нейронної мережі.

4.5. Нейронна мережа Кохонена

На відміну від трьох розглянутих у цьому розділі нейронних мереж, які відносяться до першого класу мереж, що змагаються і, отже, ваги зв'язків яких залишаються фіксованими в процесі їх функціонування, мережа Кохонена - представник другого класу конкуруючих або змагаються мереж, тобто. ваги її зв'язків змінюються під час ітераційного процесу виділення нейронів-переможців. Структура цієї нейронної мережі, що має й інші назви – самоорганізоване відображення Кохонена, що топологічно зберігає перетворення, карти ознак Кохонена, що самоорганізуються, – була запропонована Кохоненом у 1982 році [47]. Відмінна риса мережі – відображати вхідну інформацію, зберігаючи відношення сусідства вхідних елементів, тобто. зберігаючи її топологічну структуру. Ця властивість властива мозку, але використовується лише в небагатьох нейронних мережах [48 – 51], хоча воно часто буває необхідним при

встановленні характеру взаємозв'язків, які людським оком важко вловлюються. У цьому випадку можливе застосування відображення Кохонена, яке дозволяє від важко сприйманих людським зором взаємозв'язків елементів перейти, наприклад, до впорядкованого розташування елементів прямокутної, гексагональної або будь-якої іншої сітки. Це широко використовується для перетворення багатовимірних вихідних даних в одно-або двовимірні карти ознак Кохонена (або карти Кохонена, що самоорганізуються).

Інший можливий ефективний випадок застосування Кохонена перетворення. Є безліч об'єктів, частина з яких схожа між собою, інша частина схожа відносно або не схожа зовсім, а необхідно виконати класифікацію таким чином, щоб було видно, як групуються об'єкти і наскільки вони близькі або далекі один від одного.

Можливе також застосування відображення Кохонена для перетворення багатовимірних вихідних даних на одно- або двовимірні "карти ознак Кохонена", які можна розглядати як результат передобробки для інших нейронних мереж.

Структура мережі Кохонена представлена на рис. 4.11.

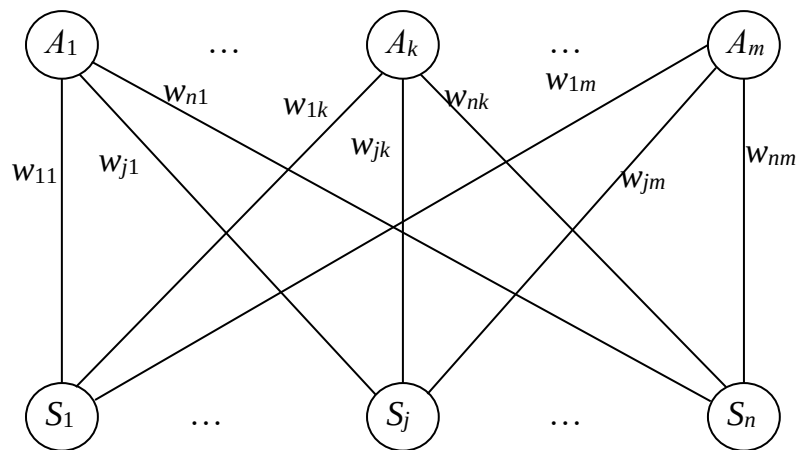


Рис. 4.11. Мережа Кохонена

Мережа має два шари нейронів. Нейрони першого шару сприймають вхідну інформацію як n -мірних безперервних векторів і передають її A -нейронам, які упорядковані в одно- чи двовимірному масиві як елементи деяких кластерів. Під час процесу самоорганізації при пред'явленні вектора деякого вхідного зображення виділяється A -елемент кластера, який у сенсі мінімуму заданої відстані (наприклад, квадрата евклідової відстані) найбільше відповідає цьому вхідному вектору. Цей виділений чи переможний елемент та її найближчі сусіди (чи термінах топології – покриття елемента) за заданим правилом змінюють свої ваги, щоб краще відповідати вхідному вектору, тобто. елемент переможець та його покриття в деякому сенсі близькі до вхідного вектора та прагнуть цю близькість збільшити.

Приклад покриття елемента A_j для одновимірного випадку наведено на рис.

4.3, а двовимірного випадку – на рис. 4.4. На рис. 4.3 покриття елемента A_j с радіусом $R = 1$ включає крім A_j всього два елементи: A_{j-1} і A_{j+1} , а покриття з $R = 5$ – десять елементів: $A_{j-5}, \dots, A_{j-1}, A_{j+1}, \dots, A_{j+5}$. У двовимірному випадку (рис 4.4) при прямокутній решітці і $R = 1$ покриття входять вісім елементів, найближчих до виділеного, а у випадку гексагональної решітки – шість. Переможні елементи, розташовані близько до краю решітки, мають менше найближчих сусідів, відсутні сусіди в процесі функціонування мережі просто ігноруються.

Алгоритм роботи мережі Кохонена

Крок 1. Ініціюються ваги w_{jk} ($j = \overline{1, n}$, $k = \overline{1, m}$) зв'язків мережі випадковими числами з інтервалу $[0, 1]$. Задається безліч, що визначає топологічну близькість елементів, та почальний коефіцієнт α .

Крок 2. Поки не виконуються умови зупинки, реалізуються кроки 3 – 9 алгоритму.

Крок 3. Для кожного вхідного вектора $S^p = (s_1^p, \dots, s_n^p)$, $p = \overline{1, L}$ виконуються кроки 4 – 6.

Крок 4. Для кожного нейрона A_j обчислюється відстань

$$D(A_j) = \sum_{i=1}^n (w_{ij} - s_i^p)^2.$$

Крок 5. Визначається нейрон A_j , для якого відстань $D(A_j)$ мінімальна.

Крок 6. Для всіх A -елементів у межах заданої околиці R елемента A_j і для всіх i виконується зміна ваги зв'язків:

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(s_i^p - w_{ij}(\text{old}), \quad i = \overline{1, n}.$$

Крок 7. Модифікується навчальний коефіцієнт α .

Крок 8. Зменшується радіус топологічної близькості як функція часу.

Крок 9. Перевіряються умови зупинки.

Крок 10. Зупинка.

Зауваження 4.1. Розглянемо деякі можливі альтернативні зміни у структурі алгоритму.

Вихідним терезам у наведеному алгоритмі задаються випадкові значення. Однак, якщо є деяка апіорна інформація про розташування або розподіл кластерів, що стосується конкретної проблеми, що вивчається, то її можна і бажано врахувати у вихідних вагах зв'язків нейронів.

Можливі структури алгоритму з різними законами зміни радіусу R -околиці найближчих елементів, однак у будь-якому випадку радіус повинен зменшуватись у міру прогресу процесу кластеризації. Аналогічно навчальний коефіцієнт може бути спадною функцією часу чи періодів навчання. Кохонен показав, що лінійного зменшення достатньо для практичних додатків, проте процес навчання мережі може бути досить тривалим.

Можливими умовами закінчення ітераційного процесу можуть бути: задана кількість ітерацій (пред'явлень вхідної множини зображень) або зменшення навчального коефіцієнта α до наперед заданого значення, або зміни величин елементів матриці ваг між двома ітераціями менше заданого значення ε і т.д.

Приклад 4.4. Застосування самоорганізованого перетворення Кохонена для кластеризації чотирьох векторів (0 0 0 1), (0 0 1 1), (1 0 0 0), (1 1 0 0) на два класи.

Оскільки вектора 4-мірні, то число вхідних S -нейронів дорівнює 4, оскільки задано два класу, то $m = 2$ і число A -нейронів дорівнює 2, звідси також слід, що покриття нейронів складається тільки з самого нейрона, тобто радіус околиці нейронів R дорівнює нулю. Покладемо також, що повчальний коефіцієнт змінюється як функція часу:

$$\alpha(t+1) = k\alpha(t),$$

де $k = 0,5$ і $\alpha(0) = 0,6$.

Використовуємо наведений вище алгоритм для вирішення цієї задачі кластеризації:

Крок 1. Ініціюється матриця M_q ваг зв'язків випадково вибраними числами з інтервалу $[0, 1]$:

$$\begin{vmatrix} 0,7 & 0,6 \\ 0,4 & 0,1 \\ 0,5 & 0,5 \\ 0,2 & 0,9 \end{vmatrix}.$$

Ініціюється радіус та навчальний коефіцієнт: $R = 0$, $\alpha = 0,6$.

Встановлюються умови закінчення ітераційного процесу: процес закінчується, якщо виконуються вісім нерівностей:

$$\left| a_{ij}^q - a_{ij}^{q-1} \right| \leq 0,0005, \quad i = \overline{1,4}, \quad j = \overline{1,2}, \quad (4.27)$$

де a_{ij}^q, a_{ij}^{q-1} – елементи матриці ваг M_q після завершення поточної та попередньої ітерацій.

Крок 2. Починаються обчислення, що реалізують кроки 3 – 9 алгоритму.

Крок 3. Для першого вектора $S^1 = (0 \ 0 \ 0 \ 1)$ виконуємо кроки 4 – 6.

Крок 4. Для нейронів A_1 і A_2 обчислюються відстані:

$$D(A_1) = (0,7 - 0)^2 + (0,4 - 0)^2 + (0,5 - 0)^2 + (0,2 - 1)^2 = 1,54,$$

$$D(A_2) = (0,6 - 0)^2 + (0,1 - 0)^2 + (0,5 - 0)^2 + (0,9 - 1)^2 = 0,63.$$

Крок 5. Визначається, що $D(A_1) > D(A_2)$, отже, нейрон A_2 є переможцем.

Крок 6. Обчислюються нові ваги нейрона, що переміг A_2 :

$$w_{i2}(new) = w_{i2}(old) + 0,6(S_i^1 - w_{i2}(old)) = 0,4w_{i2}(old) + 0,6S_i^1.$$

Після розрахунків виходить нова матриця ваг:

$$\begin{vmatrix} 0,7 & 0,24 \\ 0,4 & 0,04 \\ 0,5 & 0,20 \\ 0,2 & 0,96 \end{vmatrix}.$$

Крок 3. Для другого вектора $S^2 = (0 \ 0 \ 1 \ 1)$ виконуємо кроки 4 – 6.

Крок 4. Для нейронів A_1 і A_2 обчислюються відстані:

$$D(A_1) = (0,7 - 0)^2 + (0,4 - 0)^2 + (0,5 - 1)^2 + (0,2 - 1)^2 = 1,54,$$

$$D(A_2) = (0,24 - 0)^2 + (0,04 - 0)^2 + (0,2 - 0)^2 + (0,96 - 1)^2 = 0,70.$$

Крок 5. Визначається, що нейрон A_2 є переможцем.

Крок 6. Адаптуються ваги зв'язків нейрона, що переміг, і виходить матриця ваг з новим другим стовпцем:

$$\begin{vmatrix} 0,7 & 0,096 \\ 0,4 & 0,016 \\ 0,5 & 0,680 \\ 0,2 & 0,984 \end{vmatrix}.$$

Крок 3. Для третього вектора $S^3 = (1 \ 0 \ 0 \ 0)$ виконуються кроки 4 – 6.

Крок 4. Для нейронів A_1 і A_2 обчислюються відстані:

$$D(A_1) = (0,7 - 1)^2 + (0,4 - 0)^2 + (0,5 - 0)^2 + (0,2 - 0)^2 = 0,54,$$

$$D(A_2) = (0,096 - 1)^2 + (0,016 - 0)^2 + (0,680 - 0)^2 + (0,984 - 0)^2 = 1,888.$$

Крок 5. Визначається, що нейрон A_1 є переможцем.

Крок 6. Адаптуються ваги зв'язків нейрона, що переміг, і виходить матриця ваг з новим першим стовпцем:

$$\begin{vmatrix} 0,880 & 0,096 \\ 0,160 & 0,016 \\ 0,200 & 0,680 \\ 0,080 & 0,984 \end{vmatrix}.$$

Крок 3. Для четвертого вектора $S^4 = (1 \ 1 \ 0 \ 0)$ виконуємо кроки 4 – 6.

Крок 4. Обчислюються відстані:

$$D(A_1) = (0,880 - 1)^2 + (0,160 - 1)^2 + (0,200 - 0)^2 + (0,080 - 0)^2 = 0,766,$$

$$D(A_2) = (0,096 - 1)^2 + (0,016 - 1)^2 + (0,680 - 0)^2 + (0,984 - 0)^2 = 3,216.$$

Крок 5. Визначається, що переможцем є нейрон A_1 .

Крок 6. Адаптуються ваги зв'язків нейрона, що переміг, і виходить нова матриця ваг:

$$\begin{vmatrix} 0,952 & 0,096 \\ 0,664 & 0,016 \\ 0,080 & 0,680 \\ 0,032 & 0,984 \end{vmatrix}.$$

Крок 7. Модифікується навчальний коефіцієнт α :

$$\alpha(t = 1) = 0,5 \cdot \alpha(t = 0) = 0,5 \cdot 0,6 = 0,3.$$

Крок 8. Оскільки $R = 0$, цей крок не виконується.

Крок 9. Перевіряються умови зупинки і оскільки вони не виконуються, то перехід на крок 2 алгоритму (інакше – зупинка).

Аналогічно першою виконуються друга та наступні ітерації алгоритму. В результаті цих ітерацій маємо такі матриці ваг M_q ($q = \overline{1,9}$):

$$M_2 = \begin{vmatrix} 0,9760 & 0,0470 \\ 0,6250 & 0,0080 \\ 0,0330 & 0,6330 \\ 0,0150 & 0,9920 \end{vmatrix}, \quad M_3 = \begin{vmatrix} 0,982 & 0,034 \\ 0,601 & 0,006 \\ 0,028 & 0,607 \\ 0,011 & 0,994 \end{vmatrix}, \quad \dots, \quad M_8 = \begin{vmatrix} 0,98642 & 0,02516 \\ 0,57916 & 0,00448 \\ 0,02080 & 0,58263 \\ 0,00882 & 0,99562 \end{vmatrix},$$

$$M_9 = \begin{vmatrix} 0,98648 & 0,02504 \\ 0,57879 & 0,00448 \\ 0,02070 & 0,58224 \\ 0,00878 & 0,99564 \end{vmatrix}.$$

Після виконання дев'ятої ітерації виконуються умови зупинки (4.27) та алгоритм припиняє свою роботу.

Для векторів $(0 \ 0 \ 0 \ 1)$, $(0 \ 0 \ 1 \ 1)$, $(1 \ 0 \ 0 \ 0)$, $(1 \ 1 \ 0 \ 0)$ через те, що для кожного з двох класів тільки один з одиничних компонентів є в обох представників класу, а друга – тільки в одному з двох, а решта компонентів векторів нульові, ідеальна матриця M_u ваг має вигляд:

$$M_u = \begin{vmatrix} 1 & 0 \\ 0,5 & 0 \\ 0 & 0,5 \\ 0 & 1 \end{vmatrix}.$$

Якщо порівнювати елементи w_{21} , w_{32} матриць M_u і M_9 , максимальна похибка елементів, отриманих в результаті ітераційної процедури, становить 16,5%. Така відносно велика похибка розрахунків пояснюється дуже швидкою зміною повчального коефіцієнта α . Збільшимо у виразі $\alpha(t+1) = k\alpha(t)$ коефіцієнт k з 0,5 до 0,75 і повторимо весь ітераційний процес за тих же вихідних даних та умов закінчення ітерацій. В результаті отримаємо, що кількість ітерацій до виконання умов зупинки зросте з дев'яти до сімнадцяти, а похибка визначення елементів w_{21} , w_{32} матриці ваг зменшиться до 8%:

$$M'_1 = \begin{vmatrix} 0,952 & 0,096 \\ 0,664 & 0,016 \\ 0,080 & 0,680 \\ 0,032 & 0,984 \end{vmatrix}, \quad M'_2 = \begin{vmatrix} 0,98548 & 0,02904 \\ 0,65086 & 0,00484 \\ 0,02420 & 0,65570 \\ 0,00968 & 0,99516 \end{vmatrix}, \quad M'_3 = \begin{vmatrix} 0,99363 & 0,01275 \\ 0,62316 & 0,00213 \\ 0,01061 & 0,62529 \\ 0,00425 & 0,99787 \end{vmatrix}, \dots,$$

$$M'_{16} = \begin{vmatrix} 0,99930 & 0,00141 \\ 0,54003 & 0,00024 \\ 0,01173 & 0,54022 \\ 0,00046 & 0,99976 \end{vmatrix}, \quad M'_{17} = \begin{vmatrix} 0,99931 & 0,00139 \\ 0,53957 & 0,00024 \\ 0,01159 & 0,53975 \\ 0,00046 & 0,99976 \end{vmatrix}.$$

Зі порівняння елементів матриць M_9 і M'_{17} слід, що поліпшилися як елементи w_{21} , w_{32} , але й інші компоненти матриці. При збільшенні числа ітерацій до 70 вдається отримати похибку елементів w_{21} , w_{32} в межах одного відсотка.

4.6. Навчальне векторне розбиття

Навчальне векторне розбиття (Learning vector quantization (LVQ)) запропоновано Кохоненом в 1989 [52]. Це метод класифікації образів, в якому кожен вихідний нейрон (іноді їх група) представляють певний клас або категорію зображень, що пред'являються. Початковий вектор ваг вихідного нейрона часто є еталонним для класу, що відображається. Під час навчання ваги вихідних елементів адаптуються за допомогою вчителя (супервізора) для того, щоб апроксимувати весь простір рішень.

Навчальне векторне розбиття передбачає, що є безліч навчальних зображень з відомою класифікацією і є набір еталонних, можливо, не зовсім точних зображень або векторів, які представляють відому або задану класифікацію. За відсутності еталонних векторів беруть перші m (за кількістю класів зображень) і використовують їх як еталонні для завдання ваг зв'язків вихідних нейронів, вектори, що залишилися, використовують як навчальну

множину.

Архітектура нейронної мережі LVQ така сама, як і у самоорганізованого відображення Кохонена (рис. 4.11), проте алгоритм використання цієї архітектури в цьому випадку інший.

Мотивація для алгоритму навчання мережі LVQ полягає у визначенні вихідних нейронів, які були б у певному сенсі найбільш близькі до вхідних зображень. Якщо вхідний вектор $S^* = (s_1^*, \dots, s_j^*, \dots, s_n^*)$ та вектор ваг $W_k = (w_{1k}, \dots, w_{jk}, \dots, w_{nk})$ вихідного нейрона A_k відносяться до одного класу, то під час навчання ваги зв'язків нейрона змінюються у бік зменшення різниці. Якщо ж вихідний нейрон A_k та вхідне зображення S^* відносяться до різних класів, то ваги зв'язків елемента A_k змінюються з метою збільшення відмінностей.

Алгоритм роботи нейронної мережі LVQ

Крок 1. Ініціюються початкові ваги зв'язків нейронів A_k ($k = \overline{1, m}$) (ініціюються вихідні еталонні вектори) та відповідне нейронам безліч класів $K_n = \{k_{n1}, \dots, k_{nm}\}$ вхідних зображень.

Ініціюється множина $S = \{S^1, \dots, S^L\}$ вхідних навчальних зображень з відповідними класами $K_u = \{k_{u1}, \dots, k_{uL}\}$. Ініціюється навчальний коефіцієнт $\alpha(t)$.

Крок 2. Поки не виконуються умови зупинки, реалізуються кроки 3 – 7 алгоритму.

Крок 3. Для кожного навчального вхідного вектора S^p ($p = \overline{1, L}$) виконуються кроки 4 та 5.

Крок 4. Визначається нейрон A_j , для якого евклідова відстань $|S^p - W_j|$ між вхідним вектором $S^p = (s_1^p, \dots, s_n^p)$ і вектором ваг $W_j = (w_{1j}, \dots, w_{nj})$ мінімально.

Крок 5. Коригується вектор W_j ваг зв'язків наступним чином:
Якщо клас k_{nj} нейрона j збігається з класом k_{up} вхідного зображення S^p , то

$$W_j(\text{new}) = W_j(\text{old}) + \alpha \cdot (S^p - W_j(\text{old})),$$

якщо класи k_{nj}, k_{up} не збігаються, то

$$W_j(\text{new}) = W_j(\text{old}) - \alpha \cdot (S^p - W_j(\text{old})).$$

Крок 6. Зменшується навчальний коефіцієнт α .

Крок 7. Перевіряються умови зупинки.

Умовами зупинки може бути: досягнення заданого першому кроці алгоритму числа ітерацій чи зменшення величини α до наперед заданого значення, або відсутність істотних змін ваг зв'язків нейронів, що адаптуються і т.д.

Крок 8. Зупинка.

Приклад 4.5. Виконати класифікацію семи векторів у три класи за допомогою мережі LVQ.

Нехай задано сім векторів $(1, 1, 0, 0, 0)$, $(0, 0, 0, 1, 1)$, $(0, 0, 1, 1, 0)$, $(0, 1, 1, 0, 0)$, $(0, 0, 1, 0, 0)$, $(1, 0, 0, 0, 0)$, $(0, 0, 0, 0, 1)$, перший та шостий з яких відносяться до першого класу, другий та сьомий – до другого, третій, четвертий та п'ятий – до третього.

Візьмемо як мережу LVQ з трьома виходами і п'ятьма вхідними нейронами ($m = 3$, $n = 5$). Початкові ваги зв'язків нейронів та їх класи визначимо першими трьома заданими векторами:

$$\begin{aligned}W_1 &= (1, 1, 0, 0, 0), \text{ перший клас, } k_{n1} = 1; \\W_2 &= (0, 0, 0, 1, 1), \text{ другий клас, } k_{n2} = 2; \\W_3 &= (0, 0, 1, 1, 0), \text{ третій клас, } k_{n3} = 3.\end{aligned}$$

Використовуємо наведений вище алгоритм для вирішення цієї задачі класифікації:

Крок 1. Ініціюємо початкові ваги нейронів A_j ($j = \overline{1,3}$) та відповідні їм класи:

$$\begin{aligned}W_1 &= (1, 1, 0, 0, 0), W_2 = (0, 0, 0, 1, 1), W_3 = (0, 0, 1, 1, 0), \\k_{n1} &= 1; k_{n2} = 2, k_{n3} = 3.\end{aligned}$$

Ініціюємо безліч вхідних зображень та відповідні їм класи:

$$\begin{aligned}S^1 &= (0, 1, 1, 0, 0), k_{u1} = 3; S^2 = (0, 0, 1, 0, 0), k_{u2} = 3; \\S^3 &= (1, 0, 0, 0, 0), k_{u3} = 1; S^4 = (0, 0, 0, 0, 1), k_{u4} = 2.\end{aligned}$$

Ініціюємо навчальний коефіцієнт $\alpha(t)$: $\alpha(0) = 0,1$.

Крок 2. Починаємо виконання кроків 3 – 7 алгоритму.

Крок 3. Для вхідного вектора $S^1 = (0, 1, 1, 0, 0)$, $k_{u1} = 3$ виконуємо кроки 4 – 5.

Крок 4. Обчислюємо евклідові відстані між вхідним вектором S^1 та векторами ваг W_j ($j = \overline{1,3}$):

$$\begin{aligned}d_{11} &= \sqrt{\sum_{i=1}^5 (S_i^1 - w_{i1})^2} = \\&= \sqrt{(0-1)^2 + (1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2} = \sqrt{2} = 1,414;\end{aligned}$$

$$\begin{aligned}d_{12} &= \sqrt{\sum_{i=1}^5 (S_i^1 - w_{i2})^2} = \\&= \sqrt{(0-0)^2 + (1-0)^2 + (1-0)^2 + (0-1)^2 + (0-1)^2} = 2;\end{aligned}$$

$$d_{13} = \sqrt{\sum_{i=1}^5 (S_i^1 - w_{i3})^2} =$$

$$= \sqrt{(0-0)^2 + (1-0)^2 + (1-1)^2 + (0-1)^2 + (0-0)^2} = \sqrt{2} = 1,414.$$

Визначаємо, що вектори ваг нейронів A_1 і A_3 мають однакову мінімальну відстань до вектора S^1 .

Крок 5. Коригуємо ваги зв'язків нейронів A_1 і A_3 . Так як $k_{u1} \neq k_{u1}$ і $k_{u1} = k_{u3}$, то маємо:

$$W_1 (new) = W_1 (old) - 0,1 \cdot (S^1 - W_1 (old)) = (1, 1, 0, 0, 0) -$$

$$- 0,1 \cdot ((0, 1, 1, 0, 0) - (1, 1, 0, 0, 0)) = (1, 1, 1, 0, -0,1, 0, 0, 0, 0).$$

$$W_3 (new) = W_3 (old) + 0,1 \cdot (S^1 - W_3 (old)) = (0, 0, 1, 1, 0) +$$

$$+ 0,1 \cdot ((0, 1, 1, 0, 0) - (0, 0, 1, 1, 0)) = (0, 0, 0, 1, 1, 0, 0, 9, 0, 0, 0, 0).$$

Крок 3. Для вхідного вектора $S^2 = (0, 0, 1, 0, 0)$, $k_{u2} = 3$ виконуємо кроки 4 – 5.

Крок 4. Обчислюємо евклідові відстані між вхідним вектором S^2 та векторами ваг W_j ($j = \overline{1,3}$): $d_{21} = 1,849$, $d_{22} = 1,733$, $d_{23} = 0,906$.

Визначаємо, що вектор ваг нейрона A_3 має меншу відмінність від вектора S^2 , ніж вектори ваги інших нейронів.

Крок 5. Коригуємо ваги зв'язків нейронів A_3 . Так як $k_{u2} = k_{u3}$, то маємо:

$$W_3 = (0, 0, 0, 1, 1, 0, 0, 9, 0, 0) + 0,1 \cdot ((0, 0, 0, 0, 1, 0, 0, 0, 0, 0) -$$

$$- (0, 0, 0, 1, 1, 0, 0, 9, 0, 0)) = (0, 0, 0, 0, 9, 1, 0, 0, 0, 8, 1, 0, 0, 0).$$

Крок 3. Для вхідного вектора $S^3 = (1, 0, 0, 0, 0)$, $k_{u2} = 1$ виконуємо кроки 4 – 5.

Крок 4. Обчислюємо евклідові відстані між вхідним вектором S^3 та векторами ваг W_j ($j = \overline{1,3}$): $d_{31} = 1,01$, $d_{32} = 1,73$, $d_{33} = 1,63$.

Оскільки мінімальна відстань d_{31} , то вхідний вектор S^3 відповідає вектору вагів нейрона A_1 .

Крок 5. Коригуємо ваги зв'язків нейронів A_3 . Так як $k_{u3} = k_{u1}$, то маємо:

$$W_1 = (1, 1, 1, 0, -0,1, 0, 0, 0, 0) + 0,1 \cdot ((1, 0, 0, 0, 0, 0, 0, 0, 0, 0) -$$

$$- (1, 1, 1, 0, -0,1, 0, 0, 0, 0)) = (1, 0, 9, 0, 9, 0, -0, 0, 9, 0, 0, 0, 0, 0, 0).$$

Крок 3. Для вхідного вектора $S^4 = (0, 0, 0, 0, 1)$, $k_{u4} = 2$ виконуємо кроки 4 – 5.

Крок 4. Обчислюємо евклідові відстані між вхідним вектором S^4

та векторами ваг W_j ($j = \overline{1,3}$): $d_{41} = 1,73$, $d_{42} = 1,00$, $d_{43} = 1,63$.

Оскільки мінімальна відстань d_{42} , то вхідному зображенню відповідає вектор вагів нейрона A_2 .

Крок 5. Коригуємо ваги зв'язків нейронів A_2 . Так як $k_{u4} = k_{n2}$, то маємо:

$$W_2 = (0, 0, 0, 1, 1) + 0,1 \cdot ((0, 0, 0, 0, 1) - (0, 0, 0, 1, 1)) = (0,0, 0,0, 0,0, 0,9, 1,0).$$

Крок 6. Зменшуємо навчальний коефіцієнт α :

$$\alpha(t+1) = \alpha(t) - 0,01 = 0,1 - 0,01 = 0,09.$$

Крок 7. Перевіряємо умови зупинки.

Якщо $\alpha(t+1) = 0$, то зупинка, інакше перехід до кроку 2.

У табл. 4.4 наведено вектори $W_1 = (w_{11}, \dots, w_{15})$, $W_2 = (w_{21}, \dots, w_{25})$, $W_3 = (w_{31}, \dots, w_{35})$ ваги зв'язків вихідних нейронів мережі при зміні значень навчального коефіцієнта від 0,1 до 0,01.

Порівняння векторів ваги зв'язків вихідних нейронів, наведених в останньому рядку табл. 4.4, з ідеальними векторами ваг $W_{1и} = (1, 0,5, 0, 0, 0)$, $W_{2и} = (0, 0, 0, 0,5, 1)$, $W_{3и} = (0, 1/3, 1, 1/3, 0)$ для трьох зазначених класів показує, що у випадках другий компонент вектора W_1 та четвертої компоненти вектора W_2 похибка досягає 13%. Збільшення числа ітерацій з 10 до 13 – 15 за рахунок повільнішого зменшення величини коефіцієнта α підвищує точність визначення зазначених компонентів. Однак збільшення кількості ітерацій до тридцяти при законі зміни навчального коефіцієнта $\alpha(t+1) = \alpha(t) - 0,033$ приводить наприкінці ітераційного процесу до векторів $W_1 = (1, 0, 0, 0, 0)$, $W_2 = (0, 0, 0, 0, 1)$, $W_3 = (0, 0,5, 1, 0, 0)$.

Таблиця 4.4. Зміни векторів ваг зв'язків мережі LVQ у процесі навчання

Навчальний коефіцієнт α	Вектора ваги зв'язків вихідних нейронів		
	$W_1 = (w_{11}, \dots, w_{15})$	$W_2 = (w_{21}, \dots, w_{25})$	$W_3 = (w_{31}, \dots, w_{35})$
–	(1; 1; 0; 0; 0)	(0; 0; 0; 1; 1)	(0; 0; 1; 1; 0)
0,10	(1,090; 0,900; –0,090; 0; 0)	(0; 0; 0; 0,900; 1)	(0; 0,090; 1; 0,810; 0)
0,09	(1,082; 0,819; –0,082; 0; 0)	(0; 0; 0; 0,819; 1)	(0; 0,156; 1; 0,671; 0)
0,08	(1,075; 0,753; –0,075; 0; 0)	(0; 0; 0; 0,753; 1)	(0; 0,206; 1; 0,568; 0)
0,07	(1,070; 0,701; –0,070; 0; 0)	(0; 0; 0; 0,701; 1)	(0; 0,243; 1; 0,491; 0)
0,06	(1,066; 0,659; –0,066; 0; 0)	(0; 0; 0; 0,659; 1)	(0; 0,271; 1; 0,434; 0)
0,05	(1,063; 0,626; –0,063; 0; 0)	(0; 0; 0; 0,626; 1)	(0; 0,292; 1; 0,392; 0)

0,04	(1,060; 0,601; -0,060; 0;0)	(0; 0; 0; 0,601;1)	(0; 0,307; 1; 0,361; 0)
0,03	(1,058; 0,583; -0,058; 0;0)	(0; 0; 0; 0,583; 1)	(0; 0,318; 1; 0,340; 0)
0,02	(1,057; 0,571; -0,57; 0; 0)	(0; 0; 0; 0,571;1)	(0; 0,325; 1; 0,326; 0)
0,01	(1,056; 0,565; -0,057; 0;0)	(0; 0; 0; 0,565; 1)	(0; 0,329; 1; 0,314;0)

Отримані результати свідчать про явні недоліки алгоритму визначення параметрів нейромережі. Ці недоліки в першу чергу пов'язані з тим, що вектори вихідних даних, які використовувалися для завдання початкових ваг мережі, в процесі навчання нейромережі їй не пред'являлися, а оскільки деякі компонент цих векторів унікальні, то при великій кількості ітерацій вони губляться. Це і не гарантує отримання найкращого рішення. Отже, для завдання початкових ваг мережі бажано використовувати вектори, які мають унікальних властивостей, або вектори, отримані шляхом усереднення вихідних даних у кожному із заданих класів. Якщо ж для завдання початкових ваг використовуються вектори з унікальними властивостями, їх обов'язково необхідно включати в навчальну послідовність нейросети.

Приклад 4.6. Класифікація 81 двокомпонентного вектора у чотири класи за допомогою LVQ мережі.

У роботі [28] розглянуто приклад використання мережі LVQ для класифікації двокомпонентних векторів, наведених на рис. 4.12 як точки квадратних ґрат. Приналежність векторів до одного з чотирьох класів показана на рисунку символами $\oplus$, 0, $\otimes$, $\star$. Завдяки квадратним ґратам компоненти вхідних векторів легко обчислюються за допомогою рисунка.

Навчальна нейронна мережа містить два елементи в першому шарі і чотири в другому. Кожен із нейронів другого шару є представником одного класу елементів. Навчання починається з початковими векторами терезів, наведеними в перших чотирьох рядках табл. 4.5

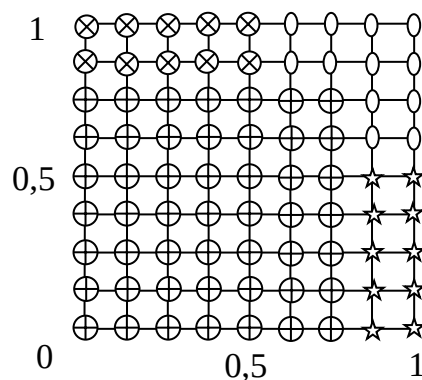


Рис. 4.12. Класифіковані двокомпонентні вектори

Після першої епохи (тобто після одноразового пред'явлення всіх навчальних векторів для адаптації ваги зв'язків вихідних нейронів) нейромережа класифікувала вхідні вектори відповідно до рис. 4.12, *а*, після другої, десятої та сорокової епох – відповідно до рис. 4.13, *б* – 4.13, *г*. Вектори ваги зв'язків нейронів і число помилок класифікації вхідних векторів після цих епох наведені в табл. 4.5.

З вихідних даних прикладу та обчислювальних експериментів випливає, що мережа з чотирма вихідними нейронами не може виконувати без помилок класифікацію вхідних векторів, причому ітераційний процес не зменшив, а збільшив кількість помилок. При цьому практично всі помилки пов'язані з неправильною класифікацією елементів першого класу.

Таблиця 4.5. Результати навчання мережі LVQ

Епоха	Номер класу векторів або вихідного нейрона	Позначення векторів класу	Ваги зв'язків вихідних нейронів (w_{1i} , w_{2i})	Число помилок класифікації
0	1	$\oplus$	0, 0	–
	2	0	1, 1	–
	3	☆	1, 0	–
	4	$\otimes$	0, 1	–
1	1	$\oplus$	0,44, 0,52	9
	2	0	0,90, 0,93	0
	3	☆	1,03, 0,17	0
	4	$\otimes$	0,13, 1,02	1
2	1	$\oplus$	0,41, 0,55	10
	2	0	0,88, 0,92	0
	3	☆	1,03, 0,24	0
	4	$\otimes$	0,22, 1,02	0
10	1	$\oplus$	0,34, 0,44	12
	2	0	0,89, 0,91	0
	3	☆	1,10, 0,26	0
	4	$\otimes$	0,30, 1,03	0
40	1	$\oplus$	0,30, 0,31	14
	2	0	0,92, 0,93	0
	3	☆	1,11, 0,26	0
	4	$\otimes$	0,27, 1,09	0

Аналіз цього прикладу дозволяє зробити кілька пропозицій щодо покращення алгоритмів навчання LVQ нейромереж:

1. У алгоритми навчання LVQ нейромереж доцільно ввести критерії, що оцінюють якість класифікації навчальних вибірок та передбачити процедуру збереження наборів вагових коефіцієнтів, що відповідають кращим розв'язкам задач класифікації.

2. Якщо виявлено класи вхідних векторів, які вносять основну частку помилок у роботу мережі, доцільно вводити у вихідний шар мережі додаткові нейрони, що відповідають цим класам. Це можна робити, використовуючи наступний алгоритм:

Крок 1. Визначається набір терезів, що відповідають кращому розв'язанню задачі класифікації.

Крок 2. Фіксуються ваги зв'язків вихідних нейронів тих класів, які вимагають додаткових вихідних елементів, і видаляються з навчальних даних вектори, які правильно класифікуються цими нейронами.

Крок 3. Вводяться додаткові вихідні нейрони та проводиться повторне навчання мережі (без зміни зафіксованої ваги).

Крок 4. Оцінюється якість класифікації повної навчальної вибірки всіма нейронами другого шару мережі. Якщо необхідно, то виконується додаткове навчання нейромережі на всій кількості вихідних даних.

Крок 5. Оцінюється необхідність повторного збільшення числа елементів вихідного шару і за необхідності цього – перехід до виконання кроку 1, інакше – припинення роботи алгоритму.

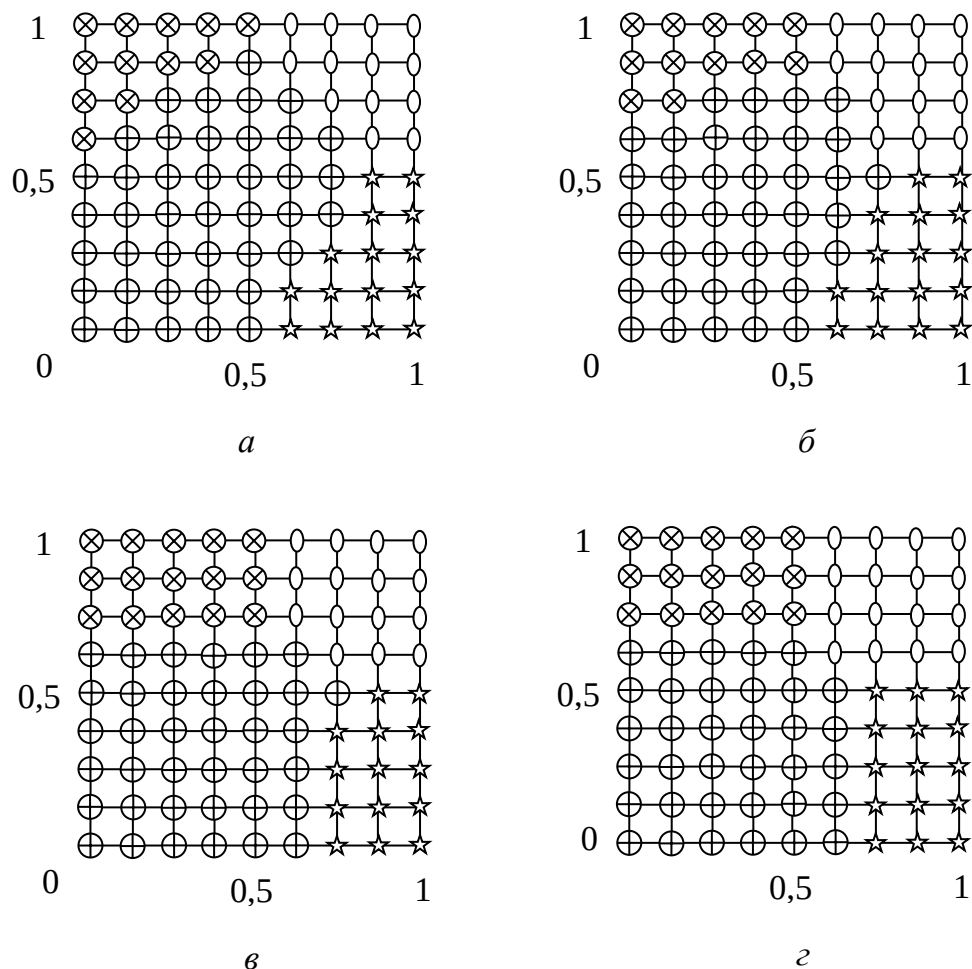


Рис. 4.13. Результати класифікації вхідних векторів після однієї, двох, десяти та сорока епох навчання

Зауваження 4.2. На кроці 2 у складних випадках можливе введення додаткових елементів різними способами для того, щоб надалі з них вибирати один або кілька кращих і, можливо, об'єднати їх у подальшому пошуку оптимального рішення вихідної задачі класифікації.

Контрольні питання

1. Нарисуйте архітектуру нейронної мережі Maxnet.
2. Який ітераційний вираз описує динаміку нейронної мережі Maxnet?
3. Яка причина назви нейронної мережі Maxican Net?
4. Нарисуйте можливі топології оточення нейрона у двовимірному випадку для мережі Maxican Net.
5. З яких шарів нейронів складається архітектура нейронної мережі Хеммінга?
6. Дайте визначення відстані Хеммінгу.
7. Як відбувається навчання нейронної мережі Хеммінгу?
8. Розробте мережу Хеммінга, що має як еталонні зображення шість чорно-білих зображень розміром 3×4 .
9. Який недолік має нейронна мережа Хеммінгу?
10. З яких шарів нейронів складається нейронна мережа, що використовує відстань Хеммінга та класифікує зображення, що знаходяться на мінімальній відстані від одного, двох або трьох еталонних зображень?
11. Нарисуйте архітектуру нейронної мережі Кохонена.
12. Опишіть алгоритм функціонування нейронної мережі Кохонена як навчання.
13. Які ще є назви у нейронної мережі Кохонена?
14. Нарисуйте архітектуру нейронної мережі LVQ.
15. Що таке навчальний векторний простір (LVQ)?
16. Опишіть алгоритм роботи нейронної мережі LVQ.
17. Які умови зупинки використовуються в алгоритмі функціонування нейронної мережі LVQ?
18. Як впливає величина навчального коефіцієнта $\alpha(t)$ на точність класифікації?

Г л а в а 5

МЕРЕЖІ ХОПФІЛДУ

Мережі цього типу були запропоновані американським біофізиком Джоном Хопфілдом у 1982 році, коли йому вдалося залучити до аналізу нейронних мереж потужний апарат статистичної фізики на основі аналогій між нейронними мережами та особливими фізичними системами – спиновим склом. Спінове скло – це ансамблі електромагнітних частинок, що знаходяться на переході фаз, наприклад, нижній точці плавлення скла. Кожна частка має власний момент кількості руху, званім спином. Спин може мати лише дві орієнтації S_1, S_2 щодо зовнішнього магнітного поля, спрямованого по деякій осі x . Одна з них збігається з напрямком поля ($S_1 = 1$), а інша – йому протилежна ($S_2 = -1$). Спинові скла мають деяку структуру атракторів (стійких станів) у своєму фазовому просторі. Кожен атрактор можна розглядати як запис певної інформації, яку можна вважати за певних умов. Пізніше з'ясувалося, що не тільки спінове скло, а й інші фізичні системи мають у своїх фазових просторах придатними для нейронних мереж безлічі атракторів. Це поняття часто використовується в теорії нейронних мереж, тому в наступному підрозділі дамо основні відомості про атрактори.

5.1. Атрактори в нейронних мережах

Визначення 5.1. Топологією в деякій множині X називається будь-яка система τ його підмножин M , що відповідає вимогам:

1. Множина X і порожня множина $\emptyset$ належать τ .

2. Об'єднання $\bigcup_{i=1}^m M_i$ будь-якого кінцевого чи нескінченного числа множин з τ належить τ .

3. Перетин $\bigcap_{r=1}^n M_r$ будь-якого кінцевого числа множин з τ належить τ .

Визначення 5.2. Топологічним простором називається пара (X, τ) , т.е. множина X із заданою у ньому топологією.

Визначення 5.3. C^0 -поток називається відображення $F_t(x)$, що задовольняє умовам:

1. $F_t(x)$ – безперервно по t і x .

2. $F_0: (X, \tau) \rightarrow (X, \tau)$, F_0 – тотожне відображення; $F_{t+s} = F_s \circ F_t$ для всіх $t, s \in R$, R – множина дійсних чисел.

Визначення 5.4. C^0 -напівпоток називається C^0 -потік, визначений тільки для $t \geq 0$.

Визначення 5.5. Нехай M_k – множина і $M_k \subset X$, F_t – C^0 -напівпотік на топологічному просторі (X, τ) , що задовольняє умові $F_t(M_k) \subset M_k$. Множина M_k називається стійким у топологічному просторі (X, τ) , якщо для будь-якої досить малої околиці U множини M_k існує така його околиця $V \supset U$, що траєкторія $x(x_0, t) \equiv F_t(x_0)$ при деякому $t \geq 0$ належить U , якщо $x_0 \in V$.

Визначення 5.6. Стійка множина M_k називається асимптотично стійким або атрактором, якщо для будь-якої околиці U множини M_k існує така його околиця V , що перетин усіх траєкторій $x(x_0, t) \equiv F_t(x_0)$, $x_0 \in V$ задовольняє рівності $\bigcap_{t>0} F_t(V) = M_k$.

Іншими словами, множина M_k стійко (асимптотично стійко), якщо будь-яка траєкторія, що проходить через точку, близьку до множини M_k , залишається поблизу цієї множини (прагне в множину M_k).

Стійкі точки і множини часто зустрічаються в різних галузях природознавства. Можна розглянути порожню нерухому напівсферу, в яку кинута маленька важка кулька (рис. 5.1 а). Кулька після низки рухів зупиняється або в нижній точці A півсфери, або в одній з точок її досить малої околиці. У цьому випадку ми маємо стійку точку.

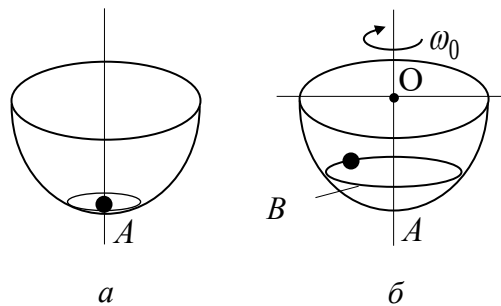


Рис. 5.1. Стійкі множини в нерухомій та
полусфері, що обертається

Якщо напівсферу почати обертати навколо деякої вертикальної осі OA (рис. 5.1, б) з деякою кутовою швидкістю ω_0 , то кулька почне рухатися і її стійке становище тепер буде окрема точка, а ціла орбіта B , яку він потрапляє за будь-яких киданнях кульки всередину цієї півсфери. Якщо напівсфера і кулька оброблені ідеально і стосуються один одного тільки в одній точці, то точка A і траєкторія B перетворюються на стійкі асимптотично. Для будь-якого околиці (яка знаходиться в межах напівсфери або поза нею, але забезпечує попадання кульки в півсферу з подальшим рухом усередині неї) точки A при нерухомій напівсфері або орбіти B при траєкторії, що обертається, кульки закінчується в асимптотично стійкому стані.

Можна дати і дещо іншу інтерпретацію атрактора. Уявімо собі певний природний ландшафт, коли йде сильний дощ і місцевістю тече вода. Атрактор можна зіставити озеро або басейн, що збирає воду. А якщо аттрактор – це точка, через яку вода йде під землю, то кожна крапля води прагне потрапити до цієї точки.

Розглянемо тепер приклад області нейронних мереж, зокрема, навчання елементарного перцептрона розпізнаванню двох зображень, наприклад, букв Π і Γ (рис. 5.2).

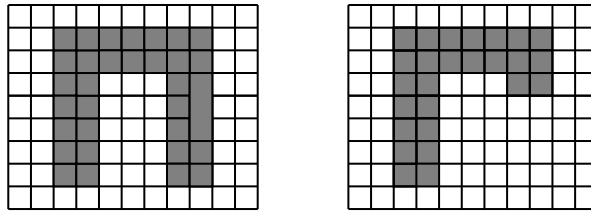


Рис. 5.2. Еталонні навчальні зображення букв

У процесі навчання мережі виникають дві стаціонарні точки або два атрактори. Басейни атракторів створюються всіма зображеннями літер П і Г із різними дефектами. Ці зображення разом з еталонними багаторазово пред'являються перцептрону в процесі навчання мережі, коли фактично по двох заданих стаціонарних точках, використовуючи ітераційну процедуру, намагаються знайти відповідну матрицю ваги зв'язків нейромережі.

Елементарний перцептрон – це мережа тільки з прямою передачею сигналів, а в таких нейромережах атракторами можуть бути лише стаціонарні точки. Якщо ж у перцептроні чи будь-якій іншій нейронній мережі є зворотні зв'язки, то при зовнішньому обуренні мережі в ній можуть виникати динамічні процеси, що не згасають. Розглянемо докладніше процеси в мережах із зворотними зв'язками. Нехай деяка мережа складається з n біполярних нейронів із матрицею ваг $W = |w_{ij}|$, $i, j = \overline{1, n}$ і час змінюється дискретно: $t = 0, 1, 2, \dots$. Припустимо, що функціонування кожного нейрона такої мережі описується рівнянням:

$$U_{out.i}(t) = \text{Sign}\left(\sum_{j=1}^n U_{out.j}(t)w_{ji} - \theta_i\right), U_{out.j}(t-1)), i = \overline{1, n}, \quad (5.1)$$

де $U_{out.i}(t)$, $U_{out.i}(t-1)$ – вихідні сигнали i -го нейрона у моменти часу t , $t-1$; $U_{out.i}(t)$ – вихідний сигнал i -го нейрона; w_{ji} – вага зв'язку між j -м та i -м нейронами; Sign – функція двох змінних, що визначається співвідношенням:

$$U_{out.i}(t) = \text{Sign}(U_{inp.i}(t), U_{out.i}(t-1)) = \begin{cases} 1, & \text{якщо } U_{inp.i}(t) > 0, \\ -1, & \text{якщо } U_{inp.i}(t) < 0, \\ U_{out.i}(t-1), & \text{якщо } U_{inp.i}(t) = 0. \end{cases} \quad (5.2)$$

$$U_{inp.i}(t) = \sum_{j=1}^n U_{out.j}(t)w_{ji} - \theta_i$$

Рівняння (5.1) часто називають нейрорівняння. Система з n нейрорівнянь визначає еволюцію мережі в часі, яка багато в чому залежить від режимів спрацьовування її елементів. Можна виділити три характерні режими спрацьовування – всі нейрони спрацьовують одночасно;

– для кожного моменту часу $t = 0, 1, 2, \dots$ задана деяка підмножина M_t з елементів мережі і в будь-який момент часу, що розглядається, t спрацьовують нейрони тільки множини M_t ;

– нейрони, що вибираються випадково (іноді послідовно або циклічно), спрацьовують строго по одному.

Останній режим функціонування компонентів нейромережі біологічно більш природний, оскільки експерименти на живих організмах показують, що нервові клітини спрацьовують незалежно один від одного і асинхронно.

Визначення 5.7. Стан $S^* = (U_{out,1}^*, U_{out,2}^*, \dots, U_{out,n}^*)$ нейромережі називається стійким по відношенню до довільного підмножини M^1 елементів мережі, якщо воно не змінюється при одночасному спрацьовуванні будь-яких окремих нейронів.

Визначення 5.8. Стан S^* нейромережі називається стійким або стаціонарним, якщо воно стійке по відношенню до асинхронного спрацьовування будь-яких окремих нейронів і одночасного спрацьовування всіх елементів мережі.

Визначення 5.9. Пряме завдання стаціонарності для нейронних мереж полягає у визначенні всіх стаціонарних станів нейромережі по заданій матриці ваг зв'язків W та вектору порогів θ (або вектору зміщень W_0) нейронів.

Визначення 5.10. Зворотне завдання стаціонарності для нейронних мереж полягає в тому, щоб за заданою множиною $S^* = \{S^1 = (U_{out,1}^1, U_{out,2}^1, \dots, U_{out,n}^1), \dots, S^k = (U_{out,1}^k, U_{out,2}^k, \dots, U_{out,n}^k)\}$ стаціонарних станів мережі визначити матрицю ваг зв'язків W та вектор порогів θ (або вектор зміщень W_0), що забезпечують ці стаціонарні стани.

Усі стаціонарні стани нейромережі теоретично можна визначити із системи рівнянь (5.1), що описує всі компоненти мережі. Для цього достатньо вибрати початковий стан S^0 всіх нейронів мережі та, використовуючи заданий режим їх спрацьовування, визначити новий стан S^1 мережі, потім, скориставшись S^1 як наступний початковий стан, визначити стан S^2 і т.д. Внаслідок цього процесу виникає послідовність станів мережі:

$$S^0, S^1, S^2, \dots, S^k, \dots \quad (5.3)$$

Ця послідовність може закінчитися або будь-яким стаціонарним станом S^p или некоторым бесконечным динамическим процессом. У разі досягнення стаціонарного стану S^p з деякого моменту часу $t = t_q$ спостерігаються стани $S^q, S^{q+1}, S^{q+2}, \dots, S^{q+m}$, які задовольняють умові $S^q = S^{q+1} = S^{q+2} = \dots = S^{q+m} = S^p$.

Стаціонарний стан мережі S^p часто називають асоційованим з її початковим станом S^0 . Взаємозв'язок станів S^0 і S^p використовується для вирішення завдань розпізнавання образів за допомогою нейромереж зі зворотними зв'язками. На такому взаємозв'язку побудовано і функціонування мереж із прямим поширенням сигналів, наприклад, перцептронів.

Якщо послідовність (5.3) не закінчується стаціонарним станом, тобто нескінченна, то в загальному випадку при випадковому спрацьовуванні окремих

нейронів або їх груп виникають послідовності (5.3) станів мережі, що важко аналізуються. У зв'язку з цим обмежимося лише режимами спрацьовування, які однозначно визначають наступний стан мережі за поточним. Одним з таких режимів є режим синхронного спрацьовування всіх елементів мережі (режим маршування), іншим режим послідовного або циклічного спрацьовування по одному елементу: i -й, $(i + 1)$ -й, ..., n -й, ..., 1 -й, 2 -й, ..., i -й, $(i + 1)$ -й, При цих режимах спрацьовування нейронів ланцюжок станів (5.3) однозначно визначається своїм початковим станом S^0 . З огляду на кінцевого числа n двійкових елементів мережі звичайно і загальне число (2^n) станів нейромережі, тому в нескінченній послідовності (5.3) знайдуться принаймні два однакові стани S^q и S^p , $q < p$. Оскільки $S^q = S^p$ і наступний стан мережі однозначно визначається поточним станом та режимом спрацьовування нейронів, то маємо:

$$S^q = S^p, S^{q+1} = S^{p+1}, S^{q+2} = S^{p+2}, \dots, S^{p-1} = S^{2p-q-1}, \dots, S^p = S^{2p-q},$$

тобто цикл повторюється. В окремому випадку, коли $p = q + 1$, отримуємо одночленний цикл або розглянутий вище стаціонарний стан.

Визначення 5.11. Цикл станів нейромережі, що не має в собі інших циклів, називається мінімальним.

Очевидно, що будь-які два мінімальні цикли або збігаються, або не мають загальних станів.

Нехай $C = \{C_1, \dots, C_p\}$ – множина всіх мінімальних циклів (або атракторів) нейромережі N із заданою матрицею W ваг зв'язків. Тоді для будь-якого стану S^* цієї нейромережі існує атрактор або мінімальний цикл $C^* \in C$, до якого воно притягується.

Визначення 5.12. Басейном B_S тяжіння циклу C_r (або басейном атрактора) називається сукупність усіх станів нейронної мережі, які притягуються до цього циклу: $B_S = \{S_{C_r}^1, \dots, S_{C_r}^{kr}\}$.

Басейни тяжіння будь-яких двох циклів $C_k, C_m \in C$ не перетинаються, тобто

$$B_S(C_k) \cap B_S(C_m) = \emptyset, \text{ якщо } k \neq m,$$

а об'єднання басейнів всіх циклів дорівнює безлічі всіх станів S нейронної мережі:

$$\bigcup_{r=1}^p B_S(C_r) = S.$$

У графічній інтерпретації басейни атракторів включають стани (або точки), відповідні мініимальному циклу, і дерева, що стягуються до станів або точок циклу (рис. 5.3).

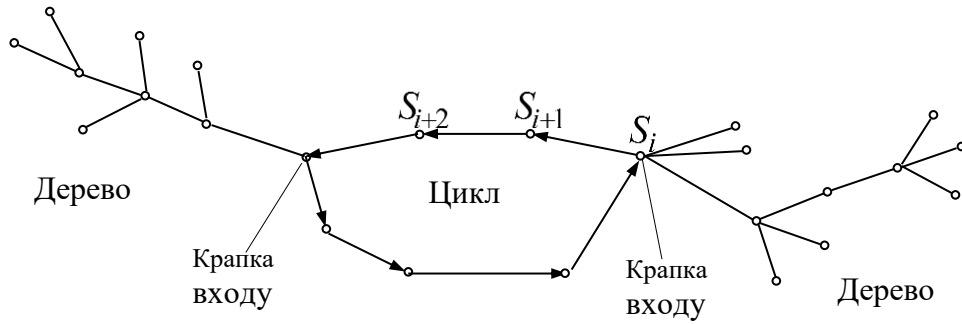


Рис. 5.3. Басейн атрактора

Точки циклу, що мають непусті дерева, називаються точками входу до циклу. Дерева різних точок входу не перетинаються, а їхня сукупність покриває весь басейн атрактора.

Кожен стан $S_{Cr}^1, \dots, S_{Cr}^{kr}$ басейна $B_S(C_r)$ тяжіння циклу C_r характеризується числом кроків $q(S_{Cr}^i)$, $i = \overline{1, kr}$, за яке мережа зі стану S_{Cr}^i потрапляє до циклу C_r , а найважливішою характеристикою басейну є максимальна висота h його дерев:

$$h(B_S(C_r)) = \max_{i \in \{1, \dots, kr\}} q(S_{Cr}^i).$$

У гіршому випадку, коли цикл має одну точку входу, а дерево лінійне, висота дерева дорівнює кількості станів мережі в басейні атрактора.

Якщо нейронні мережі N_1, N_2 мають однакові цикли $C_1, \dots, C_p$ та басейни тяжіння $B_S(C_1), \dots, B_S(C_p)$, але різні набори чисел $h^{N_1}(B_S(C_i))$, $h^{N_2}(B_S(C_i))$, $i = \overline{1, p}$, то мережа N_1 краще за мережу N_2 , якщо виконуються співвідношення

$$h^{N_1}(B_S(C_i)) \leq h^{N_2}(B_S(C_i)), \quad i = \overline{1, p}. \quad (5.4)$$

Якщо співвідношення (5.4) виконуються не для всіх циклів C_i , то мережі вважаються незрівнянними. Висота дерев басейнів атракторів використовується і для оцінки якості процесів чи алгоритмів навчання. Нехай є два алгоритми навчання A_1, A_2 , які за заданими стаціонарними станами S -мережі отримують матриці ваг зв'язків $W_{A_1}(S)$, $W_{A_2}(S)$. Якщо басейни атракторів, породжені матрицею ваг $W_{A_1}(S)$ виходять регулярно (може, з рідкісними винятками) краще басейнів нейромереж з матрицею $W_{A_2}(S)$, то алгоритм навчання A_1 краще алгоритму A_2 .

5.2. Дискретна мережа Хопфілда

У своїй першій роботі [53], присвяченій нейронним мережам, Хопфілд розглянув повнозв'язкову нейронну мережу з бінарних елементів із симетричними зв'язками. Структура цієї мережі наведено на рис. 5.4.

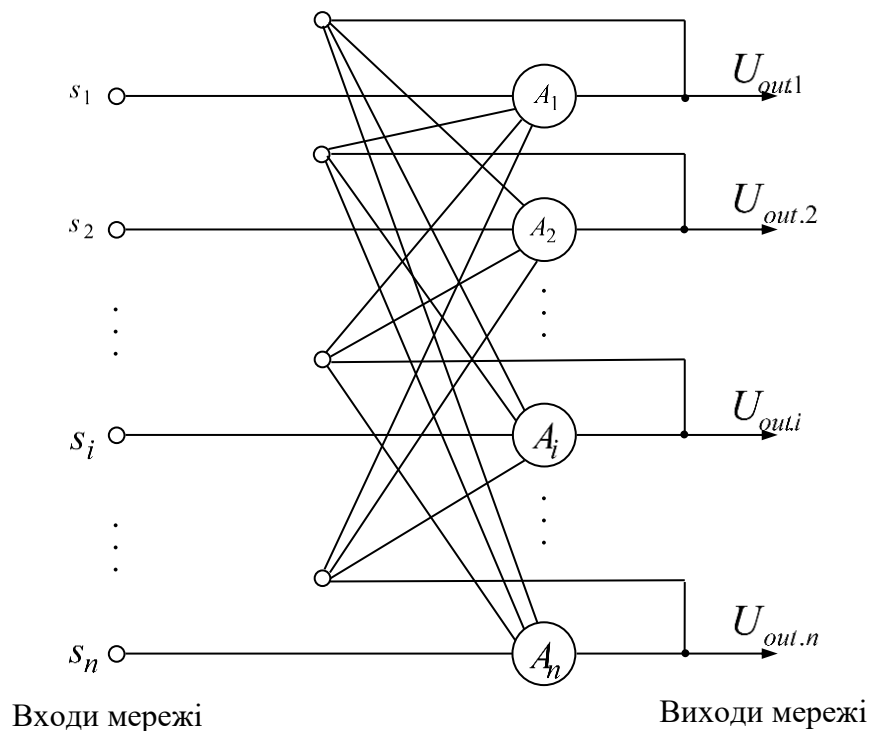


Рис. 5.4. Нейронна мережа Хопфілда

Дискретна мережа Хопфілда складається з єдиного шару нейронів, кожен з яких пов'язаний з усіма іншими та має мережні вхід та вихід. Сигнали на мережних входах нейронів визначають їх вихідні сигнали:

$$U_{\text{вх},i} = s_i, \quad i = \overline{1,n}.$$

При відсутності сигналів на мережних входах елементи мережі функціонують в асинхронному режимі, при якому кожен з них визначає свій вихідний сигнал у випадкові моменти часу із заданою середньою частотою відповідно до виразу

$$U_{out,i} = \begin{cases} 1, & \text{якщо } \sum_{j=1}^n w_{ji} U_{out,j} > \theta_i, \\ 0, & \text{якщо } \sum_{j=1}^n w_{ji} U_{out,j} < \theta_i, \\ U_{out,i}, & \text{якщо } \sum_{j=1}^n w_{ji} U_{out,j} = \theta_i, \end{cases} \quad (5.5)$$

де $U_{out,i}$, θ_i – відповідно вихідний сигнал та поріг i -го нейрона; w_{ji} – вага зв'язку між j -м та i -м нейронами.

Матриця $W = \|w_{ij}\|$ ($i, j = \overline{1, n}$) ваг зв'язків нейромережі симетрична і має нульові компоненти на головній діагоналі, тобто

$$w_{ij} = w_{ji}, \quad w_{ii} = 0, \quad i, j = \overline{1, n}. \quad (5.6)$$

Такий вид матриці ваги забезпечує стійкість мережі: при подачі на її входи зовнішніх сигналів виникає послідовність станів мережі виду (5.3), яка закінчується стаціонарним станом. Процес досягнення стаціонарного стану можна описати за допомогою мінімізації спеціальної енергетичної функції:

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} U_{out,i} - \sum_{j=1}^n U_{inp,j} U_{out,j} + \sum_{j=1}^n \theta_j U_{out,j} \quad (5.7)$$

де E – штучна енергія мережі, задана як функції Ляпунова; $U_{inp,j}$ – зовнішній вхідний сигнал j -го нейрона.

Енергію всієї мережі можна як суму енергій її окремих нейронів:

$$E = \sum_{j=1}^n E_j. \quad (5.8)$$

З виразів (5.8) та (5.7) маємо:

$$E_j = \left(\sum_{i=1, i \neq j}^n w_{ij} U_{out,i} + U_{inp,j} - \theta_j \right) U_{out,j}.$$

Розглянемо зміну енергії ΔE_j довільного j -го елемента при його спрацьовуванні:

$$\begin{aligned} \Delta E_j &= E_j(U_{out,j} + \Delta U_{out,j}) - E_j(U_{out,j}) = \\ &= - \left(\sum_{i=1, i \neq j}^n w_{ij} U_{out,i} + U_{inp,j} - \theta_j \right) \Delta U_{out,j}. \end{aligned} \quad (5.9)$$

Для бінарних нейронів збільшення їх вихідних сигналів може приймати лише три значення: +1, 0, -1. При цьому знак збільшення $\Delta U_{out,j}$ для j -го елемента збігається зі знаком виразу у круглих дужках. Справді, якщо $\Delta U_{out,j} = -1$, тобто нейрон переходить з одиничного стану в нульовий, то це означає, що відповідно до виразу (5.5) виконується нерівність

$$\sum_{j=1}^n w_{ij} U_{out,i} + U_{inp,j} < \theta_j ,$$

тобто вираз у круглих дужках співвідношення (5.9) негативний. Якщо $\Delta U_{out,j} = -1$, то аналізований j -й нейрон переходить з нульового стану в одиничний і, отже,

$$\sum_{j=1}^n w_{ij} U_{out,i} + U_{inp,j} > \theta_j$$

або

$$\sum_{j=1}^n w_{ij} U_{out,i} + U_{inp,j} - \theta_j > 0 .$$

Зі збігу знаків співмножників у виразі (5.9) випливає, що при спрацьовуванні будь-якого j -го нейрона ні його енергія, ні енергія всієї мережі збільшитися не може. Вона або залишається колишньою, якщо $\Delta U_{out,j} = 0$, або зменшується, якщо $\Delta U_{out,j} \neq 0$. Отже, у міру спрацьовування нейронів енергія монотонно зменшуватиметься, поки не досягне одного зі своїх локальних мінімумів, якому відповідає одна зі стаціонарних точок нейромережі. Еволюція мережі з будь-якого початкового стану через існування функції Ляпунова (5.7) завжди закінчується у одній з її стаціонарних точок, тобто атракторами в дискретної бінарної мережі Хопфілда є лише стаціонарні точки. Це ж твердження справедливе і для дискретної мережі Хопфілда з біполярними нейронами.

Для зберігання деякої множини зображень $S = \{S^1 = (s_1^1, \dots, s_n^1), \dots, S^P = (s_1^P, \dots, s_n^P)\}$ в біполярній мережі Хопфілда використовується матриця W ваги зв'язків з елементами

$$w_{ij} = \sum_{k=1}^P s_i^k s_j^k, \quad w_{ij} = w_{ji} \text{ при } i \neq j, \quad w_{ii} = 0, \quad (5.10)$$

де індекс k вказує на належність вхідних сигналів k -му зображенню.

При переході до бінарних нейронів елементи w_{ij} матриці W визначаються співвідношенням

$$w_{ij} = \sum_{k=1}^P (2s_i^k - 1)(2s_j^k - 1), \quad w_{ij} = w_{ji} \text{ при } i \neq j, \quad w_{ii} = 0, \quad (5.11)$$

Пороги θ_i всіх бінарних елементів зазвичай приймаються рівними нулю, а пороги біполярних нейронів часто визначаються сумою елементів матриці ваг [46]:

$$\theta_i = \frac{1}{2} \sum_{j=1}^n w_{ij}. \quad (5.12)$$

Для мережі Хопфілда число p зображень, що запам'ятовуються, не повинно перевищувати величини, приблизно рівної $0,15n$, де n – число нейронів мережі. Крім того, якщо є пари дуже схожих зображень, наприклад, S^k, S^q , вони можуть викликати у мережі перехресні асоціації, тобто пред'явлення на входи мережі зображення S^k може призводити до появи на її виході зображення S^q і навпаки.

Завдання, що вирішуються дискретною мережею Хопфілда з бінарними або біполярними нейронами як асоціативна пам'ять, формулюються наступним чином. Відомий набір еталонних двійкових зображень чи сигналів. Мережа повинна вміти за частковою інформацією неідеальних зображень, що подаються на її вхід, виділяти еталонні зображення або давати інформацію про те, що вхідний вектор не відповідає жодному зі збережених у пам'яті. Коли мережа розпізнає якесь зображення, то на її виходах з'являється саме це зображення. В іншому випадку вектор вихідних сигналів не збігається з жодним з еталонних.

Приклад 5.1. Розглянемо можливості дискретної мережі Хопфілда з дев'ятьма біполярними нейронами щодо розпізнавання неідеальних зображень літер Н та Т. Еталонні зображення S^1 і S^2 цих букв наведено на рис. 5.5 там же дана нумерація елементів зображень, що відповідає нейронам мережі Хопфілда та їх векторному уявленню:

$$S^1 = (s_1^1, \dots, s_9^1) = (1, -1, 1, 1, 1, 1, 1, -1, 1),$$

$$S^2 = (s_1^2, \dots, s_9^2) = (1, 1, 1, -1, 1, -1, -1, 1, -1).$$

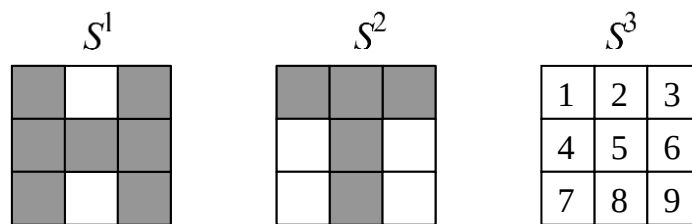


Рис. 5.5. Зображення S^1, S^2, S^3

Відповідно до вихідних даних вираз (5.10) для прикладу, що розглядається, набуває вигляду:

$$w_{ij} = \sum_{k=1}^2 s_i^k s_j^k = s_i^1 s_j^1 + s_i^2 s_j^2, \quad i \neq j.$$

Розрахуємо вагу зв'язку w_{12} :

$$w_{12} = 1 \cdot (-1) + 1 \cdot 1 = 0.$$

В силу загальної рівності $w_{ij} = w_{ji}$ також отримаємо, що $w_{21} = w_{12} = 0$. Аналогічно розраховуються і інші елементи w_{ij} ($i, j = \overline{1,9}$, $i \neq j$) матриці W ваги зв'язків. Елементи головної діагоналі матриці W визначаються виразом (5.10) при $i = j$: $w_{11} = w_{22} = \dots = w_{99} = 0$.

Результати розрахунків матриці W наведено у табл. 5.1.

Таблиця 5.1. Матриця ваг зв'язків

Номера нейронів	Номера нейронів								
	1	2	3	4	5	6	7	8	9
1	0	0	2	0	2	0	0	0	0
2	0	0	0	-2	0	-2	-2	2	-2
3	2	0	0	0	2	0	0	0	0
4	0	-2	0	0	0	2	2	-2	2
5	2	0	2	0	0	0	0	0	0
6	0	-2	0	2	0	0	2	-2	2
7	0	-2	0	2	0	2	0	-2	2
8	0	2	0	-2	0	-2	-2	0	-2
9	0	-2	0	2	0	2	2	-2	0

Пороги біполярних нейронів мережі Хопфілда розраховуються за допомогою співвідношення (5.12) та даних табл. 5.1:

$$\theta_k = \frac{1}{2} \sum_{i,j=1}^9 w_{ij} = 4, \quad k = \overline{1,9}.$$

Пред'явимо мережі Хопфілда зображення S^1 літери Н (рис. 5.5) і розрахуємо вихідні сигнали мережі після його зняття за двох значень порогів: $\theta_k = 4$ і $\theta_k = -4$. Результати розрахунків наведено у табл. 5.2.

З аналізу даних табл. 5.2 випливає, що вектор вихідного зображення мережі повторює зображення S^1 у широкому діапазоні значень порогів. Аналогічна ситуація виходить при пред'явленні зображення S^2 літери Т (табл. 5.3), тобто розрахована дискретна мережа Хопфілда, як і належить, повторює своєму виході ідеальні вхідні зображення.

Таблиця 5.2. Результати розрахунків вихідних сигналів мережі Хопфілда після пред'явлення зображення S^1 літери Н

Номера нейронів	Компоненти зображення S^1	Вхідні сигнали нейронів	Пороги нейронів	Вихідні сигнали нейронів
1	1	4	-4 або 4	1
2	-1	-10	-4 або 4	-1
3	1	4	-4 або 4	1
4	1	10	-4 або 4	1
5	1	4	-4 або 4	1
6	1	10	-4 або 4	1
7	1	10	-4 або 4	1
8	-1	-10	-4 або 4	-1
9	1	10	-4 або 4	1

Таблиця 5.3. Результати розрахунків вихідних сигналів мережі Хопфілда після пред'явлення зображення S^2 літери Т

Номера нейронів	Компоненти зображення S^2	Вхідні сигнали нейронів	Пороги нейронів	Вихідні сигнали нейронів
1	1	4	-4 або 4	1
2	1	10	-4 або 4	1
3	1	4	-4 або 4	1
4	-1	-10	-4 або 4	-1
5	1	4	-4 або 4	1
6	-1	-10	-4 або 4	-1
7	-1	-10	-4 або 4	-1
8	1	10	-4 або 4	1
9	-1	-10	-4 або 4	-1

Пред'явимо тепер мережі зображення S^{3I} , інверсне зображенню S^3 (рис. 5.5). Зображення S^{3I} можна розглядати як спотворене уявлення літери Н, у якого втрачено дві негативні компоненти. Результати розрахунків для цього випадку при $\theta_k = -4$ наведено у табл. 5.4 (при $\theta_k > 2$ зображення не відновлюється.). Зіставлення п'ятих стовпців таблиць 5.4 та 5.2 показує, що мережа відновила еталонне зображення літери Н.

Таблиця 5.4. Результати розрахунків вихідних сигналів мережі Хопфілда після пред'явлення зображення S^3

Номера нейронів	Компоненти зображення S^{3I}	Вхідні сигнали нейронів	Пороги нейронів	Вихідні сигнали нейронів
1	1	4	-4	1
2	1	-6	-4	-1
3	1	4	-4	1
4	1	2	-4	1
5	1	4	-4	1
6	1	2	-4	1
7	1	2	-4	1
8	1	-6	-4	-1
9	1	2	-4	1

При пред'явленні зображень S^1, S^2, S^{3I} мережа потрапляла до стаціонарної точки за один такт часу при синхронному спрацьовуванні всіх її елементів. Однак таке ідеально швидке досягнення стійкого стану можливе далеко не завжди. Пред'явимо мережі зображення S^4 (рис. 5.6), яке можна розглядати як сильно спотворений еталон літери Н, у якого п'ять одиничних компонентів замінені на протилежні "1". Для досягнення стаціонарної точки, що відповідає зображенню S^1 літери Н, у цьому випадку при $\theta_k = -4$ необхідно два такти часу. При цьому мережа проходить через стани S^5 (рис. 5.6). Вхідні та вихідні сигнали нейронів мережі під час цього динамічного процесу наведені у табл. 5.5.

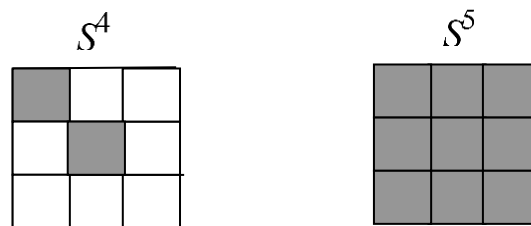


Рис. 5.6. Зображення S^4, S^5

Таблиця 5.5. Результати розрахунків вихідних сигналів мережі Хопфілда після пред'явлення зображення S^4

Номери нейронів	Компоненти зображення $S^4(t = 0)$	Вхідні сигнали нейронів		Вихідні сигнали нейронів	
		$t = 1$	$t = 2$	$t = 1$	$t = 2$
1	1	0	4	1	1
2	-1	6	-6	1	-1
3	-1	4	4	1	1
4	-1	-2	2	1	1
5	1	0	4	1	1
6	-1	-2	2	1	1
7	-1	-2	2	1	1
8	-1	6	-6	1	-1
9	-1	-2	2	1	1

5.3. Безперервні мережі Хопфілда

Подібно до людської пам'яті, дискретна мережа Хопфілда має асоціативну пам'ять, тому вона може в частині заданої інформації відновлювати всю інформацію, якщо вона зберігається в її пам'яті.

Для створення асоціативної пам'яті ваги зв'язків мережі вибираються таким чином, щоб мінімум енергетичної функції був у потрібних вершинах одиничного гіперкуба. Якщо бінарну або біполярну функції активації нейронів дискретної мережі Хопфілда замінити на відповідно бінарну сигмоїдальну

$$f(S) = \frac{1}{1 + e^{-\sigma S}}$$

або біполярну сигмоїдальну

$$f(S) = \frac{2}{1 + e^{-\sigma S}} - 1$$

функції, то виходить безперервна мережа Хопфілда. Якщо коефіцієнт σ , що задає крутість сигмоїдальної функції, великий, то властивості безперервних і дискретних елементів і нейромереж близькі і безперервні мережі набувають стійких станів біля вершин гіперкуба, тобто коли сигнали на виходах всіх

нейронів мережі близькі до двійкових (одиницям і мінус одиницям для біполярних мереж або одиниць і нул для бінарних). При зменшенні коефіцієнта $\square$ вихідні сигнали нейронів починають помітно відрізнятися від дискретних значень та стійкі стани нейромережі віддаляються від вершин гіперкубу.

Як і у випадку двійкових нейромереж, стійкість безперервних мереж Хопфілда гарантується, якщо матриця ваги зв'язків мережі симетрична і має нульові компоненти на головній діагоналі, тобто при виконанні умов (5.6). Енергетична функція нейромережі може задаватися співвідношенням (5.7), але запропоновані й інші вирази, наприклад, Хопфілд в оригінальному поданні безперервних нейронних мереж в 1984 використовував функцію

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} U_{out,i} U_{out,j} - \sum_{i=1}^n \theta_i U_{out,i} + \frac{1}{\tau} \sum_{i=1}^n \int_0^{U_{out,i}} f_i^{-1}(U_{out,i}) dU_{out,i}$$

де E – енергія мережі; $U_{out,i}$, $U_{out,j}$, $(i, j = \overline{1, n})$ – вихідні сигнали нейронів з номерами i і j ; w_{ij} – вага зв'язку між i -м та j -м нейронами; θ_i – поріг i -го нейрона; τ – часова константа; f – функція активації нейрона.

Наявність зв'язків між усіма нейронами мережі Хопфілда та безперервність їх зміни у певному інтервалі дозволяє використовувати безперервну мережу Хопфілда для вирішення різних оптимізаційних завдань. Одне з таких завдань – завдання комівояжера або торговця, що подорожує.

Комівояжер повинен відвідати n різних міст з координатами $A_1(x_1, y_1), \dots, A_n(x_n, y_n)$. При цьому потрібно, щоб у кожне місто він потрапив лише один раз, а потім повернувся до міста, з якого розпочалася поїздка. Необхідно також, щоб відстань, яку він має проїхати, була мінімальною.

Це завдання має численні практичні програми, наприклад, при формуванні оптимальних траєкторій руху інструмента роботів, що здійснюють свердління складних деталей або точкове зварювання будь-яких виробів, при визначенні мінімальних довжин ліній передач при з'єднанні засобів обчислювальної техніки або електроенергетики в кільце і т.д.

Знаючи координати міст i та j , які має відвідати комівояжер, можна обчислити відстані між ними

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

та побудувати матрицю відстаней

	A_1	A_2	$\dots$	A_n
A_1	0	d_{12}	$\dots$	d_{1n}
A_2	d_{21}	0	$\dots$	d_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_n	d_{n1}	d_{n2}	$\dots$	0

Число возможных решений задачи коммивояжера для случая n городов равно $n!$. Так как для определения оптимального решения этой задачи не известно лучшего метода, чем полный перебор всех возможных траекторий движения, то установление оптимальной траектории реально неосуществимо при больших значениях n . В связи с этим предложено множество эвристических методов решения этой задачи, в том числе и на основе нейронных сетей.

Для решения задачи коммивояжера функция энергии может быть задана различными способами. В форме Хопфилда-Танка для нейронов с бинарной сигмоидальной функцией активации она имеет вид:

$$E = \frac{A}{2} \sum_{x=1}^n \sum_{i=1}^n \sum_{j=1, j \neq i}^n U_{x,i} U_{x,j} + \frac{B}{2} \sum_{i=1}^n \sum_{x=1}^n \sum_{y=1, y \neq x}^n U_{x,i} U_{y,j} + \frac{C}{2} \left[n - \sum_{x=1}^n \sum_{i=1}^n U_{x,i} \right]^2 + \frac{D}{2} \sum_{x=1}^n \sum_{y=1, y \neq x}^n \sum_{i=1}^n d_{xy} U_{x,i} (U_{x,j} + U_{y,i-1}) + \frac{1}{\tau} \sum_{i=1}^n \int_0^{U_i} f^{-1}(U) dU, \quad (5.12)$$

де A, B, C, D, n, τ – позитивні константи; x, y – індексні змінні для перебору міст; i, j – індексні змінні для перебору нейронів мережі; $U_{x,i}$ – вихідний сигнал i -го нейрона мережі, що моделює x -е місто у вихідному завданні; f_i – функція активації i -го нейрона; $U_i = f(U_{inp.})$; $U_{inp.}$ – вхідний сигнал нейрона; d_{xy} – відстань між містами x, y шляхом масштабування зведено до значень з інтервалу $[0, 1]$, тобто $0 \leq d_{xy} \leq 1$.

Перший доданок у виразі енергії (5.12) дорівнює нулю тільки у тому випадку, коли кожен рядок табл. 5.6 містить не більше однієї одиниці. Другий доданок дорівнює нулю тільки у випадку, коли кожен стовпець таблиці містить не більше однієї одиниці. Третя складова функції енергії дорівнює нулю, коли табл. 5.6 містить рівно n одиниць. Таким чином, перші три компоненти функції E при великих значеннях коефіцієнтів A, B, C забезпечують вимогу, щоб кожне місто було послідовно відвідане один раз. Якщо ця вимога виконується, то вони звертаються в нуль, забезпечуючи низькоенергетичний стан системи, і мінімізуються лише два останні складові виразу (5.12).

Таблиця 5.6. Матриця, на основі якої формуються перші три компоненти функції енергії (5.12)

Міста	Порядок проходження міст					
	1	2	...	x	...	n
A_1	0	1	...	0	...	0
A_2	0	0	...	1	...	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
A_x	0	0	...	0	...	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
A_n	1	0	...	0	...	0

Як і у випадку дискретної мережі Хопфілда, для досягнення мінімуму енергії необхідно змінити вихідні сигнали $U_{x,i}$ ($x, i = \overline{1, n}$) не вело до збільшення енергії, тобто щоб виконувалася нерівність виду

$$\frac{\partial E}{\partial U_{x,i}} \leq 0. \quad (5.13)$$

Так як

$$\begin{aligned} \frac{\partial E}{\partial U_{x,i}} = & A \sum_{j=1, j \neq i}^n U_{x,j} + B \sum_{y=1, y \neq x}^n U_{y,i} + C \left[n - \sum_{x=1}^n \sum_{i=1}^n U_{x,i} \right] + \\ & + D \sum_{y=1, y \neq x}^n d_{xy} (U_{y,i+1} + U_{y,i-1}) + \frac{U_{x,i}}{\tau} \end{aligned} \quad (5.14)$$

і кожен доданок у правій частині виразу (5.14) для бінарної сигмоїдальної функції невід'ємний, то це з урахуванням виразу (5.13) призводить до вимоги:

$$-\frac{\partial E}{\partial U_{x,i}} \leq 0. \quad (5.15)$$

Вага зв'язку $w(x, i, y, j)$ між нейронами i і j ($i, j = \overline{1, n}$, $i \neq j$), відповідних міст x та y , визначається співвідношенням

$$w(x, i, y, j) = -A\delta_{xy}(1 - \delta_{ij}) - B\delta_{ij}(1 - \delta_{xy}) - C - Dd_{xy}(\delta_{i,j+1} + \delta_{i,j-1}),$$

де δ_{kq} – дельта-функція Дірака:

$$\delta_{kq} = \begin{cases} 1, & \text{якщо } k = q, \\ 0, & \text{якщо } k \neq q. \end{cases}$$

Кожен елемент мережі отримує сигнал зсуву:

$$U_{cm} = CN,$$

де параметр N зазвичай вибирається дещо більше, ніж кількість міст n .

Алгоритм роботи мережі Хопфілда при вирішенні завдання комівояжера наступний:

Крок 1. Задаються ваги зв'язків нейронів та ініціюється їхня активність.

Задається мале значення Δt кроку за часом.

Крок 2. Поки не дотримуються умов зупинки, виконуються кроки 3 – 7.

Крок 3. Кроки 4 - 6 виконуються протягом інтервалу часу $[0, n^2]$, де n – число городів.

Крок 4. Випадковим чином вибирається нейрон.

Крок 5. Коригується вхідний сигнал вибраного нейрона:

$$U_{inp,x,j}(t + \Delta t) = U_{inp,x,j}(t) + \Delta t [-U_{inp,x,j}(t) - A \sum_{j=1, j \neq i}^n U_{x,j} - \\ - B \sum_{j=1, j \neq i}^n U_{x,j} - C[N - \sum_{x=1}^n \sum_{i=1}^n U_{x,i}] - D \sum_{j=1, j \neq i}^n d_{xy}(U_{y,i+1} + U_{y,i-1})].$$

Крок 6. Визначається вихідний сигнал нейрона $U_{x,i} = f(U_{inp,x,i})$.

Крок 7. Перевіряються умови зупинки.

Крок 8. Зупинка.

У роботі [28] під час вирішення завдання з десятьма містами використовувалися такі константи: $A = B = D = 500$, $C = 200$, $N = 15$, $\Delta t = 10^{-5}$. У випадку немає систематичного методу визначення цих констант, хоча збіжність методу сильно залежить від своїх чисельних значень.

Контрольні питання

1. Що таке атрактор?
2. Наведіть приклади атракторів у нейронних мережах.
3. Нарисуйте архітектуру дискретної мережі Хопфілда.
4. Чим відрізняється стійка множина від асимптотично стійкого?
5. Які рівняння називаються нейрорівняннями?
6. Які три характерні режими спрацювання нейронів Ви знаєте?
7. Сформулюйте пряме завдання стаціонарності для нейронних мереж.
8. Сформулюйте зворотне завдання стаціонарності для нейронних мереж.
9. Що таке цикл станів нейронної мережі?
10. Що таке басейн атрактора (або басейн тяжіння циклу)?
11. Чи перетинаються басейни будь-яких двох циклів?
12. Як спрацювають нейрони у мережі Хопфілда?

13. Яка енергетична функція пов'язана з дискретною нейронною мережею Хопфілда?
14. Як визначаються ваги зв'язків дискретної нейронної мережі Хопфілда?
15. Як визначаються пороги нейронів у мережі Хопфілда?
16. Навчіть дискретну мережу Хопфілда розпізнаванню двох літер з Вашого імені.
17. У чому відмінність безперервної мережі Хопфілда від дискретної мережі Хопфілда?
18. Опишіть основний алгоритм розв'язання задачі комівояжера за допомогою мережі Хопфілда.

Г л а в а 6

ДВОНАПРАВЛЕНА АСОЦІАТИВНА ПАМ'ЯТЬ

6.1. Двонаправлена асоціативна пам'ять на двійкових елементах

Будь-яка нейронна мережа може розглядатися як асоціативна пам'ять, якщо зображення, що зберігаються у вагах її зв'язків, є атракторами, оскільки будь-яке вхідне зображення, що потрапило в область тяжіння атрактора, може бути використане для відтворення збереженої інформації. Подібна пам'ять уже аналізувалася в п'ятому розділі, присвяченому мережам Хопфілда. Мережі Хопфілда дозволяють зашумленим або неповним зображенням відновлювати збережені, тобто виправляти і доповнювати вхідні зображення, але не асоціювати їх з іншими образами. Отже, мережі Хопфілда мають лише автоасоціативну пам'ять. Це пояснюється однорівневою структурою мережі, в якій вхідні та вихідні зображення з'являються на виходах тих самих елементів.

Двонаправлена асоціативна пам'ять (bidirectional associative memory, BAM) на двійкових елементах було запропоновано Коско 1988 року [47]. Архітектура мережі представлена на рис. 6.1 [46].

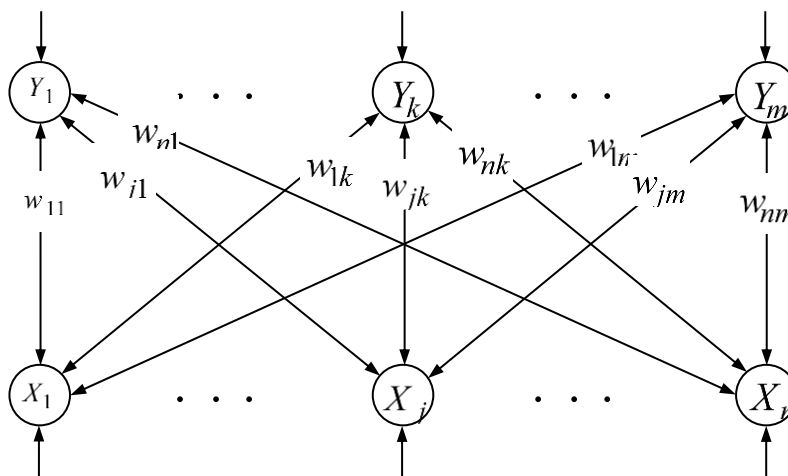


Рис. 6.1. Двонаправлена асоціативна пам'ять

Мережа складається з двох шарів нейронів, пов'язаних парами двонаправлених завислих зв'язків. Зображення (або n -вимірні або m -вимірні вхідні вектори) можуть подаватися відповідно на входи X - або Y -елементів. При цьому не передбачається подача зображень на обидва шари елементів одночасно. Якщо вагова матриця для сигналів, що посилюються з X -шару елементів в Y -шар, є

$$W = \begin{bmatrix} w_{11} & \dots & w_{1k} & \dots & w_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ w_{j1} & \dots & w_{jk} & \dots & w_{jm} \\ \dots & \dots & \dots & \dots & \dots \\ w_{n1} & \dots & w_{nk} & \dots & w_{nm} \end{bmatrix}, \quad (6.1)$$

то вагова матриця для сигналів від Y -елементів у X -шар має вигляд:

$$W_1 = W^T = \begin{bmatrix} w_{11} & \dots & w_{1j} & \dots & w_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{k1} & \dots & w_{kj} & \dots & w_{kn} \\ \dots & \dots & \dots & \dots & \dots \\ w_{m1} & \dots & w_{mj} & \dots & w_{mn} \end{bmatrix}.$$

Деякі автори (наприклад, [25]) для того, щоб підкреслити особливості двонаправленої асоціативної пам'яті (ДАП): схожість з мережами Хопфілда, двошаровість та можливість збільшення числа верств мережі, використовують інше графічне уявлення архітектури (рис. 6.2)

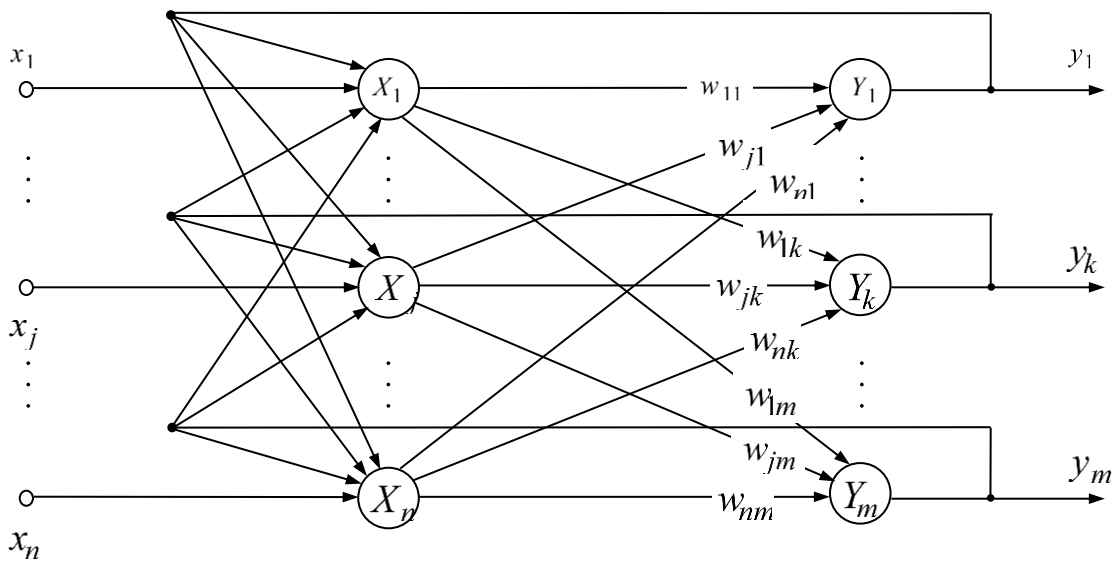


Рис. 6.2. Структура, що відображає функціонування ДАП при подачі зображень на X -шар елементів

Недолік цього графічного представлення ДАП полягає в тому, що тут шари нейронів перебувають у нерівноправному положенні - створюється видимість, що мережа може реагувати лише при пред'явленні зображень на шар X -елементів. Однак ДАП реагує на вхід кожного шару як X -шар, так і Y -шар елементів може бути вхідним і вихідним, тобто на рис. 6.2 або повинні бути додані мережні входи

для Y -нейронів та мережеві виходи для X -елементів, або обумовлено, що ця структура відповідає функціонуванню мережі при подачі вхідного зображення на шар X -елементів.

Мережа здатна запам'ятовувати пари асоційованих один з одним образів $S^p = (s_1^p, \dots, s_n^p)$, $T^p = (t_1^p, \dots, t_m^p)$ з деяких заданих множин образів $S = \{S^1, \dots, S^p, \dots, S^L\}$, $T = \{T^1, \dots, T^p, \dots, T^L\}$, L – кількість асоційованих пар.

Існують різні варіанти ДАП, у тому числі бінарний, біполярний та безперервний. Двійкові форми ДАП (бінарна та біполярна) тісно взаємопов'язані. Процес навчання ДАП з бінарними нейронами полягає в попередньому налаштуванні ваги зв'язків між X - і Y -нейронами (елементів матриці $W = |w_{ij}|$) відповідно до формули

$$w_{ij} = \sum_{p=1}^L (2s_i^p - 1)(2t_j^p - 1), \quad i = \overline{1, n}, \quad j = \overline{1, m}. \quad (6.2)$$

Для біполярних нейронів елементи матриці (6.1) визначаються виразом

$$w_{ij} = \sum_{p=1}^L s_i^p t_j^p, \quad i = \overline{1, n}, \quad j = \overline{1, m}. \quad (6.3)$$

Динаміка двонаправленої асоціативної пам'яті є ітераційною. Процес зміни вихідних сигналів нейронів кожного шару ДАП здійснюється синхронно на початку ітерації, при цьому сигнали надсилаються з шару в шар послідовно, а не одночасно в обох напрямках. При бінарних вхідних векторах вихідні сигнали X - та Y -нейронів визначаються функціями активації $f_p(U_{inp.p})$

$$U_{out.p}(t+1) = \begin{cases} 1, & \text{якщо } U_{inp.p}(y) > 0, \\ U_{out.p}(t), & \text{якщо } U_{inp.p}(y) = 0, \\ 0, & \text{якщо } U_{inp.p}(y) < 0, \end{cases} \quad (6.4)$$

де $U_{out.p}(t+1)$ – вихідний сигнал p -го ($p = 1, \dots, n, n+1, \dots, n+m$) бінарної елементи в момент часу $t+1$; $U_{inp.p}(t)$ – вхідний сигнал p -го елемента на момент часу t .

Для біполярних векторів вхідних функції активації $f_p(U_{inp.p})$ для елементів X - та Y -шару задаються виразом

$$U_{out.p}(t+1) = \begin{cases} 1, & \text{якщо } U_{inp.p}(y) > \theta_p, \\ U_{out.p}(t), & \text{якщо } U_{inp.p}(y) = \theta_p, \\ -1, & \text{якщо } U_{inp.p}(y) < \theta_p, \end{cases} \quad (6.5)$$

де θ_p – поріг p -го елемента ДАП, $p = 1, \dots, n, n+1, \dots, n+m$.

З виразів (6.4), (6.5) слід, що й вхідний сигнал елемента точно дорівнює пороговому значенню, то функція активації залишає на виході нейрона попереднє значення вихідного сигналу. У зв'язку з цим початкова активація нейронів зазвичай провадиться нульовими вхідними сигналами. Функціонування мережі може починатися із завдання зображення (вхідних сигналів) будь-якого з шарів ДАП. Опишемо алгоритм роботи двійкової мережі при першому пред'явленні зображення S^k шару X-елементів.

Алгоритм функціонування ДАП

Крок 1. Ініціюються ваги зв'язків, що визначаються за допомогою співвідношень (6.2) або (6.3), для множини з L пар асоційованих один з одним двійкових зображень (S^p, T^p) , $p = \overline{1, L}$. Задається початковий час: $t = 0$. Ініціюються нульовими вхідними сигналами всі нейрони ДАП:

$$U_{inp.Xi}(0) = 0, U_{out.Xi}(0) = f_1(U_{inp.Xi} = 0, i = \overline{1, n},$$

$$U_{inp.Xj}(0) = 0, U_{out.Xj}(0) = f_1(U_{inp.Xj} = 0, j = \overline{1, m}.$$

Крок 2. Для заданого зображення $S^k = (s_1^k, ..., s_n^k)$ виконуються кроки 3 – 7.

Крок 3. Задаються вхідні сигнали нейронів X-шару:

$$U_{inp.Xi}(0) = s_j^k, i = \overline{1, n}.$$

Задається час $t = t + 1$ та обчислюються вихідні сигнали X-елементів

$$U_{out.Xj}(t+1) = f_1(U_{inp.Xi}(t)), i = \overline{1, n}$$

Крок 4. Доки не встановляться вихідні сигнали всіх X- та Y-нейронів, виконуються кроки 5 – 7 алгоритму.

Крок 5. Адаптується активність елементів Y-шару.

Обчислюються вхідні та вихідні сигнали Y-елементів:

$$U_{inp.Yj}(t+1) = \sum_{i=1}^n w_{ij} U_{out.Xi}(t+1), j = \overline{1, m},$$

$$U_{out.Yj}(t+2) = f_j(U_{inp.Yj}(t+1)), j = \overline{1, m}.$$

Вихідні сигнали Y-нейронів надсилаються на входи елементів X-шару.

Крок 6. Адаптується активність елементів X-шару.

Обчислюються вхідні та вихідні сигнали X-елементів:

$$U_{inp.Xi}(t+2) = \sum_{j=1}^m w_{ij} U_{out.Yj}(t+2), i = \overline{1, n}$$

Вихідні сигнали X -нейронів посиляються на входи елементів Y -шару.

Крок 7. Перевіряється тест на збіжність. Порівнюються вихідні сигнали X -нейронів $U_{out.Xi}(t+3)$ та $U_{out.Xi}(t+1)$ $i = \overline{1, n}$, а також Y -нейронів $U_{out.Yj}(t+2)$ та $U_{out.Yj}(t)$, $j = \overline{1, m}$ отримані на поточній та попередній ітераціях. Якщо не виконується хоча б одна з $(n + m)$ рівностей

$$U_{out.Xi}(t+3) = U_{out.Xi}(t+1), i = \overline{1, n},$$

$$U_{out.Yj}(t+2) = U_{out.Yj}(t), j = \overline{1, m}$$

то здійснюється перехід до кроку 5 алгоритму, інакше – перехід до кроку 8.

Крок 8. Зупинка.

Приклад 6.1. Розглянемо використання ДАП із біполярними нейронами для запам'ятовування двох пар асоційованих зображень (S^1, t^1) , (S^2, t^2) , представлених на рис. 6.3. Нумерацію елементів зображень наведено на рис. 6.4.

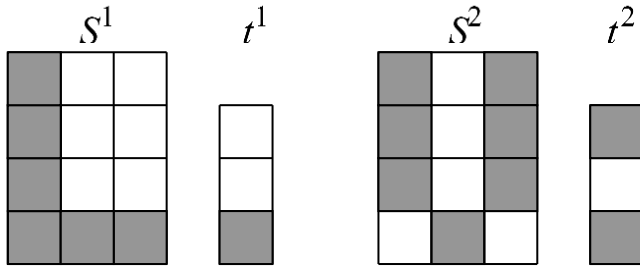


Рис. 6.3. Пари асоційованих зображень

1	2	3
4	5	6
7	8	9
10	11	12

1
2
3

Рис. 6.4. Нумерація елементів зображень

У векторній формі зображення S^1, t^1, S^2, t^2 мають вигляд:

$$S^1 = (1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1), t^1 = (-1, -1, 1),$$

$$S^2 = (1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1), t^2 = (1, -1, 1).$$

Нехай зображення S^1, S^2 пред'являються на входи X -елементів, а зображення t^1, t^2 – на входи нейронів Y -шару, тоді число елементів X -шару має дорівнювати дванадцяти ($n = 12$), а Y -шару – трьом ($m = 3$). Використовуючи

векторне представлення зображень та співвідношення (6.3), розрахуємо елементи матриць W и W^T . Для ваг w_{11} и w_{12} маємо:

$$w_{11} = \sum_{p=1}^2 s_1^p t_1^p = s_1^1 t_1^1 + s_1^2 t_1^2 = 1 \cdot (-1) + 1 \cdot 1 = 0,$$

$$w_{12} = \sum_{p=1}^2 s_1^p t_2^p = s_1^1 t_2^1 + s_1^2 t_2^2 = 1 \cdot (-1) + 1 \cdot (-1) = -2.$$

Аналогічно розраховуються та інші елементи матриць W и W^T . В результаті отримаємо:

$$W = \begin{pmatrix} 0 & -2 & 2 \\ 0 & 2 & -2 \\ 2 & 0 & 0 \\ 0 & -2 & 2 \\ 0 & 2 & -2 \\ 2 & 0 & 0 \\ 0 & -2 & 2 \\ 0 & 2 & -2 \\ 2 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & -2 & 2 \\ -2 & 0 & 0 \end{pmatrix}, \quad W^T = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & -2 & 0 & -2 \\ -2 & 2 & 0 & -2 & 2 & 0 & -2 & 2 & 0 & 0 & -2 & 0 \\ 2 & -2 & 0 & 2 & -2 & 0 & 2 & -2 & 0 & 0 & 2 & 0 \end{pmatrix}.$$

Для вхідного зображення S^1 , асоційованого з вектором t^1 , маємо:

$$\begin{aligned} ((U_{ex.Y1}(S^1), U_{ex.Y2}(S^1), U_{ex.Y3}(S^1))) &= S^1 W = \\ &= (1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1) W = (-10, -14, 14), \end{aligned}$$

тобто $U_{inp.Y1} = -10$, $U_{inp.Y2} = -14$, $U_{inp.Y3} = 14$.

Використовуючи функцію активації біполярних нейронів при нульовому порозі, визначимо фактичний вектор $t^{1\Phi}$ вихідних сигналів Y -нейронів:

$$t^{1\Phi} = (U_{out.Y1}(S^1), U_{out.Y2}(S^1), U_{out.Y3}(S^1)) = (-1, -1, 1) = t^1$$

Подаючи вектор t^1 на входи Y -нейронів, отримаємо вектор сигналів

$$\begin{aligned} U_{inp.X1}(t^1), U_{inp.X2}(t^1), \dots, U_{inp.X12}(t^1) &= t^1 W^T = (-1, -1, 1) W^T = \\ &= (4, -4, -2, 4, -4, -2, 4, -4, -2, 2, 4, 2). \end{aligned}$$

При нульових порогах X -елементів за допомогою функції активації (6.5) розраховується вектор $S^{1\Phi}$ вихідних сигналів нейронів X -шару:

$$S^{1\Phi} = (1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1) = S^1.$$

Аналогічно для зображення S^2 отримаємо:

$$U_{inp.Y1}(S^2), U_{inp.Y2}(S^2), U_{inp.Y3}(S^2) = S^2 W = \\ (1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1) W = 10, -14, 14,$$

$$t^{2\Phi} = U_{out.Y1}(S^2), U_{out.Y2}(S^2), U_{out.Y3}(S^2) = (1, -1, 1) = t^2,$$

де $t^{2\Phi}$ – фактичний вектор вихідних сигналів Y -елементів при пред'явленні на входи X -шару нейронів зображення S^2 . Неважко перевірити, що за вхідного вектора t^2 на виході нейронів X -шару ДАП з'явиться зображення S^2 . Таким чином, мережа правильно визначає асоційовані вектори для будь-якого з цих векторів S^1, t^1, S^2, t^2 .

ДАП має гарну здатність до узагальнень і може правильно відтворювати асоційовані вектори при пред'явленні на її входи зашумлених зображень. Пред'явимо мережі послідовно зображення S^{11} та S^{21} (рис. 6.5), є спотвореними описами зображень S^1, S^2 (рис. 6.3).

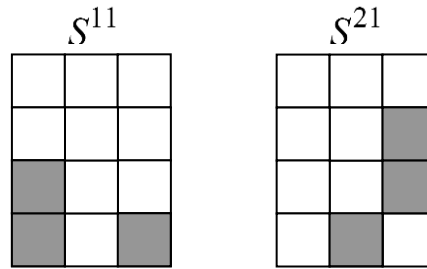


Рис. 6.5. Зображення S^{11}, S^{21}

Після пред'явлення зображень S^{11} та S^{21} неважко отримати:

$$U_{inp.Y1}(S^{11}), U_{inp.Y2}(S^{11}), U_{inp.Y3}(S^{11}) = S^{11} W =$$

$$(-1, -1, -1, -1, -1, 1, 1, -1, -1, -1, 1, -1) W = -10, -2, 2,$$

$$t(S^{11}) = U_{out.Y1}(S^{11}), U_{out.Y2}(S^{11}), U_{out.Y3}(S^{11}) = (-1, -1, 1) = t^1,$$

$$U_{inp.Y1}(S^{21}), U_{inp.Y2}(S^{21}), U_{inp.Y3}(S^{21}) = S^{21} W =$$

$$(-1, -1, -1, -1, -1, 1, -1, 1, -1, 1, -1) W = 8, -2, 2,$$

$$t(S^{21}) = U_{out.Y1}(S^{21}), U_{out.Y2}(S^{21}), U_{out.Y3}(S^{21}) = (1, -1, 1) = t^2,$$

Таким чином, мережа правильно визначає асоційовані вектори, хоча в першому зображенні змінено три елементи з дванадцяти або 25% компонентів зображення, а в другому – 33% спотворених елементів. Подібним чином мережа поводить при пред'явленні спотворених векторів шару Y -нейронів. Нехай, наприклад, мережі послідовно пред'являються вектора $t^{11} = (-1, -1, 0)$ та $t^{21} = (1, 0, 1)$. В результаті аналогічних обчислень отримаємо:

$$\begin{aligned} U_{inp.X1}(S^{11}), U_{inp.X2}(S^{11}), \dots, U_{inp.X12}(S^{11}) &= \tau^{11} W^\tau = \\ &= (-1, -1, 0) W^\tau = \\ &= 2, -2, -2, 2, -2, 2, -2, -2, 2, 2, 2, 2, \end{aligned}$$

$$S(t^{11}) = 1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1 = S^1,$$

$$\begin{aligned} U_{inp.X1}(t^{21}), U_{inp.X2}(t^{21}), \dots, U_{inp.X12}(t^{21}) &= \tau^{21} W^\tau = \\ &= 2, -2, 2, 2, -2, 2, 2, -2, 2, -2, 2, -2. \end{aligned}$$

$$S(t^{21}) = 1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1 = S^2,$$

тобто і в цих випадках мережа правильно визначає асоційовані вектори.

6.2. Оцінка ємності двонаправленої асоціативної пам'яті

Як і інші нейронні мережі, ДАП має обмеження на максимальну кількість інформації, що запам'ятовується. Якщо ця межа перевищена, то мережа може виробляти неправильні асоціації. Оцінки, наведені в роботі Коско [55], обмежують кількість асоціацій, що запам'ятовуються, аналізованої ДАП числом m нейронів у меншому шарі при спеціальному кодуванні зображень, що максимізує ємність пам'яті. Це кодування передбачає отримання кожного із зображень, що запам'ятовуються, з однаковими (або різними не більше, ніж на одиницю) числами біполярних компонент з різними знаками. Точніші оцінки істотно зменшують ємність пам'яті нейромережі [25]. Для N випадково вибраних та оптимально закодованих векторів, кожен з яких повинен бути відновлюваним, у загальному випадку необхідне виконання нерівності

$$N < m / (4 \log_2 m). \quad (6.6)$$

При заміні вимоги відновлення кожного з N образів, що запам'ятовуються, слабшим – відновлюються всі запам'ятовані вектори за винятком їх незначної частини, число N може бути збільшено в два рази:

$$N < m / (2 \log_2 m). \quad (6.7)$$

Якщо, наприклад, $m = 1024$, то кількість зображень, що запам'ятовуються, за співвідношенням (6.6) не повинна перевищувати 25, а за нерівністю (6.7) – 51. Таким чином, розглянута ДАП може запам'ятовувати дуже обмежену кількість асоціацій.

Зауваження 6.1. Для конкретних асоційованих пар зображень нерівності (6.6), (6.7) можуть давати занижені оцінки. Наприклад, з нерівності (6.6) для $m = 3$ маємо $N < 1$, а з нерівності (6.7) випливає, що $N < 2$, однак приклад 6.1 переконливо показує, що ДАП з трьома нейронами в меншому шарі запам'ятовує та відновлює дві асоціації чи чотири зображення, наведені на рис. 6.3. Тим не менш і в розглянутому прикладі ємність ДАП невелика, вже при трьох парах зображень, що запам'ятовуються (рис. 6.6) нейромережа не здатна відновлювати без помилок все запам'ятоване.

Приклад 6.2. Збільшимо кількість пар зображень прикладу 6.1 до трьох (рис. 6.6) і досліджуємо можливості ДАП відновлювати асоційовані компоненти.

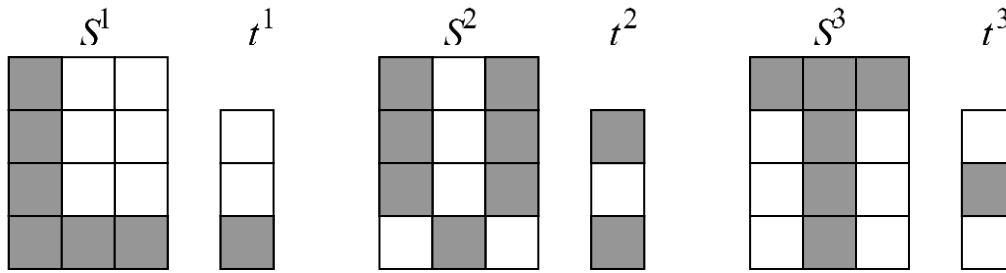


Рис. 6.6. Пари асоційованих зображень

Запишемо асоційовані пари зображень S^1 и t^1 , S^2 и t^2 , S^3 и t^3 у векторній формі:

$$\begin{aligned} S^1 &= (1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1), \quad t^1 = (-1, -1, 1), \\ S^2 &= (1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1), \quad t^2 = (1, -1, 1), \\ S^3 &= (1, 1, 1, -1, 1, -1, -1, 1, -1, -1, 1, -1), \quad t^3 = (-1, 1, -1). \end{aligned} \quad (6.8)$$

Використовуючи векторну форму представлення зображень (6.8) та співвідношення (6.3) при $L = 3$, $n = 12$, $m = 3$, розрахуємо елементи матриць W та W^T

$$w_{11} = w_{11}^T = \sum_{p=1}^3 s_1^p t_1^p = s_1^1 t_1^1 + s_1^2 t_1^2 + s_1^3 t_1^3 = 1 \cdot (-1) + 1 \cdot 1 + 1 \cdot (-1) = -1,$$

.....

$$\cdot w_{12,3} = w_{3,12}^T = \sum_{p=1}^3 s_{12}^p t_3^p = s_{12}^1 t_3^1 + s_{12}^2 t_3^2 + s_{12}^3 t_3^3 = 1 \cdot 1 + (-1) \cdot 1 + (-1) \cdot (-1) = 1.$$

В результаті отримаємо наступні матриці W та W^T вагових коефіцієнтів:

$$W = \begin{pmatrix} -1 & -1 & 1 \\ -1 & 3 & -3 \\ 1 & 1 & -1 \\ 1 & -3 & 3 \\ -1 & 3 & -3 \\ 3 & -1 & 1 \\ 1 & -3 & 3 \\ -1 & 3 & -3 \\ 3 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \end{pmatrix}, \quad W^T = \begin{pmatrix} -1 & -1 & 1 & 1 & -1 & 3 & 1 & -1 & 3 & -1 & -1 & -1 \\ -1 & 3 & 1 & -3 & 3 & -1 & -3 & 3 & -1 & -1 & -1 & -1 \\ 1 & -3 & 1 & 3 & -3 & 1 & 3 & -3 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Для вхідних зображень S^1, S^2, S^3 , асоційованих відповідно до векторів t^1, t^2, t^3 , визначаємо вхідні та вихідні сигнали біполярних Y -нейронів при нульових порогах елементів:

$$U_{inp.Y1}(S^1), U_{inp.Y2}(S^1), U_{inp.Y3}(S^1) = S^1 W = \\ (1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1)W = -6, -18, 18,$$

$$t(S^1) = U_{out.Y1}(S^1), U_{out.Y2}(S^1), U_{out.Y3}(S^1) = (-1, -1, 1) = t^1,$$

$$U_{inp.Y1}(S^2), U_{inp.Y2}(S^2), U_{inp.Y3}(S^2) = S^2 W = \\ (1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1)W = 12, -16, 16,$$

$$t(S^2) = U_{out.Y1}(S^2), U_{out.Y2}(S^2), U_{out.Y3}(S^2) = (1, -1, 1) = t^2,$$

$$U_{inp.Y1}(S^3), U_{inp.Y2}(S^3), U_{inp.Y3}(S^3) = S^3 W = \\ = (1, 1, 1, -1, 1, -1, -1, 1, -1, -1, 1, -1)W = -12, 18, 20,$$

$$t(S^3) = U_{out.Y1}(S^3), U_{out.Y2}(S^3), U_{out.Y3}(S^3) = (-1, 1, -1) = t^3,$$

Таким чином, ДАП за пред'явленими зображеннями S^1, S^2, S^3 правильно визначає асоційовані ним зображення t^1, t^2, t^3 . Пред'явимо тепер послідовно зображення t^1, t^2, t^3 Y -шару елементів і визначимо за допомогою

транспонованої матриці ваг W^T вектора вхідних та вихідних сигналів біполярних X -нейронів:

$$U_{inp.Y1}(\tau^1), U_{inp.Y2}(\tau^1), U_{inp.Y3}(\tau^1) = \tau^1 W = \\ = 3, -5, -3, 5, -5, -1, 5, -5, -1, 3, 3, 3,$$

$$S(t^1) = 1, -1, -1, 1, -1, -1, 1, -1, -1, 1, 1, 1 = S^1,$$

$$U_{inp.X1}(t^2), U_{inp.X2}(t^2), \dots, U_{inp.X12}(t^2) = t^2 W^T = \\ (1, -1, 1) W^T = \\ 1, -7, -1, 7, -7, 5, 7, -7, 5, 1, 1, 1,$$

$$S(t^2) = (1, -1, -1, 1, -1, 1, 1, -1, 1, 1, 1, 1) \neq S^2,$$

$$(U_{ex.X1}(t^3), U_{ex.X2}(t^3), \dots, U_{ex.X12}(t^3)) = t^3 W^T = (-1, 1, -1) W^T = \\ = (-1, 7, 1, -7, 7, -5, -7, 7, -5, -1, -1, 1),$$

$$S(t^3) = -1, 1, 1, -1, 1, -1, -1, 1, -1, -1, -1, 1 \neq S^3,$$

З виконаних обчислень випливає, що за пред'явленими зображеннями t^1, t^2, t^3 правильно відновлюється лише одне зображення S^1 . Два інших виявляються спотвореними (рис. 6.7).

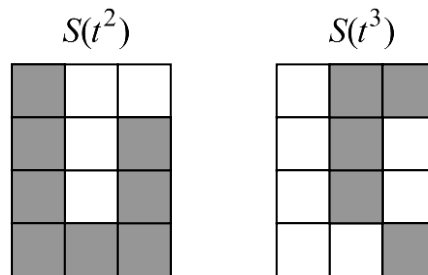


Рис. 6.7. Невдале відновлення зображень S^2, S^3

Вимога приблизної рівності чисел позитивних і негативних біполярних компонентів у кожному з ДАП зображень, що запам'ятовуються, дуже істотно і його не можна ігнорувати, так як це зменшує ємність пам'яті. Продемонструємо це на конкретному прикладі.

Приклад 6.3. Замінімо у прикладі 6.1 вектори $t^1 = (-1, -1, 1)$, $t^2 = (1, -1, 1)$, кожен з яких має біполярні компоненти різних знаків відповідно на вектори $t^1 = (-1, -1, -1)$, $t^2 = (1, 1, 1)$, мають всі біполярні компоненти одного знака, і перевіримо здатність ДАП правильно відновлювати асоційовані зображення.

Використовуючи вихідні зображення та співвідношення (6.3), розрахуємо елементи матриць W та W^T

$$w_{11} = w_{11}^T = \sum_{p=1}^2 s_1^p t_1^p = s_1^1 t_1^1 + s_1^2 t_1^2 = 1 \cdot (-1) + 1 \cdot 1 = 0,$$

.....

$$w_{12,3} = w_{3,12}^T = \sum_{p=1}^2 s_{12}^p t_3^p = s_{12}^1 t_3^1 + s_{12}^2 t_3^2 = 1 \cdot (-1) + (-1) \cdot 1 = -2.$$

В результаті отримаємо наступні матриці W та W^T вагових коефіцієнтів:

$$W = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & 2 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & 2 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & 2 & 2 \\ -2 & -2 & -2 \\ 0 & 0 & 0 \\ -2 & -2 & -2 \end{pmatrix}, \quad W^T = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & -2 & 0 & -2 \\ 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & -2 & 0 & -2 \\ 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & -2 & 0 & -2 \end{pmatrix}.$$

Зображення t^{11} та t^{21} відновлюються за зображеннями S^1 та S^2 поскільки

$$U_{inp.Y1}(S^1), U_{inp.Y2}(S^1), U_{inp.Y3}(S^1) = S^1 W = (1, -1, -1, 1, -1, 1, 1, -1, 1, -1, 1, 1) W = (-10, -10, -10),$$

$$U_{inp.Y1}(S^2), U_{inp.Y2}(S^2), U_{inp.Y3}(S^2) = S^2 W = (1, -1, 1, 1, -1, 1, 1, -1, 1, -1, 1, -1) W = (10, 10, 10)$$

і, отже,

$$t(S^1) = U_{out.Y1}(S^1), U_{out.Y2}(S^1), U_{out.Y3}(S^1) = (-1, -1, -1) = t^{11},$$

$$t(S^2) = U_{out.Y1}(S^2), U_{out.Y2}(S^2), U_{out.Y3}(S^2) = (1, 1, 1) = t^{21}.$$

Однак зображення S^1 та S^2 відповідно за зображеннями t^{11} та t^{21} не відновлюються. У цьому випадку маємо:

$$\begin{aligned} U_{inp.X1}(t^{11}), U_{inp.X2}(t^{11}), \dots, U_{inp.X12}(t^{11}) &= t^{11}W^\tau = \\ &(-1, -1, -1) W^\tau = \\ &(0, 0, -6, 0, 0, -6, 0, 0, -6, 6, 0, 6), \end{aligned} \quad (6.9)$$

$$\begin{aligned} U_{inp.X1}(t^{21}), U_{inp.X2}(t^{21}), \dots, U_{inp.X12}(t^{21}) &= t^{21}W^\tau = \\ &(1, 1, 1) W^\tau = \\ &(0, 0, 6, 0, 0, 6, 0, 0, -6, -6, 0, -6), \end{aligned} \quad (6.10)$$

Використовуючи біполярну функцію активації (6.5), із співвідношень (6.9), (6.10) отримаємо векторне представлення зображень $S(t^{11}), S(t^{21})$:

$$S(t^{11}) = 0, 0, -1, 0, 0, -1, 0, 0, -1, 1, 0, 1 \neq S^1,$$

$$S(t^{21}) = 0, 0, 1, 0, 0, 1, 0, 0, 1, -1, 0, -1 \neq S^2,$$

тобто кількість зображень, що запам'ятовуються і правильно відновлюються, дійсно залежить від виду цих зображень і воно зменшується, якщо ігнорується вимога приблизної рівності чисел позитивних і від'ємних біполярних компонентів у кожному із запам'ятовуваних зображень.

Теоретично ємність ДАП може бути збільшена до 2^m зображень, де m – число нейронів у меншому шарі, якщо для кожного елемента мережі з біполярною функцією активацією (6.5) ввести свій поріг θ . Така нейромережа отримала назву негомогенної ДАП на відміну розглянутої гомогенної ДАП, у якій пороги всіх нейронів приймаються рівними нулю. На жаль, практичне використання негомогенної ДАП такої ємності утруднено через жорсткі обмеження, що накладаються на зображення, що запам'ятовуються. Якщо зображення вибираються випадковим чином, але при цьому кожне з них містить $(4 + \log_2 m)$ позитивних біполярних компонент, а число N асоціацій, що запам'ятовуються менше $0,34m^2/(4 + \log_2 m)$, то негомогенна пам'ять може правильно відновлювати до 98% зображень, що запам'ятовуються. При $m = 1024$ число зображень, що запам'ятовуються, не повинно перевищувати 3636, що на два порядки більше в порівнянні з розглянутою гомогенною ДАП.

Через відсутність загальної теорії та зручних алгоритмів для перекодування довільних наборів векторів, що запам'ятовуються, у вектора з лімітованим числом одиниць обмеження числа позитивних компонент у вхідних зображеннях є серйозним недоліком негомогенної ДАП. Інший істотний недолік ДАП полягає в тому, що при зашумлених вхідних векторах мережа може відтворювати асоціації, що не відповідають заданим векторам вхідним. Незважаючи на ці недоліки, ДАП,

в силу її логічної простоти та зручності реалізації у вигляді НВІС, є дуже перспективним класом інтенсивно досліджуваних нейромереж.

6.3. Безперервна ДАП

Розглянуті варіанти ДАП використовували нейрони з простими пороговими функціями активації, проте розривність передавальних функцій нейронів біологічно малоправдоподібною. У зв'язку з цим були розроблені варіанти безперервної ДАП на нейронах з безперервними функціями активації. Безперервна двонаправлена асоціативна пам'ять перетворює безперервні вхідні сигнали в безперервні вихідні сигнали, що змінюються в інтервалі $[0, 1]$ або $[-1, 1]$, використовуючи відповідні функції активації сигмоїду для всіх своїх елементів.

Для бінарних пар вхідних векторів, (S^k, t^k) , $k = \overline{1, 4}$ елементи матриці ваг $W = \|w_{ij}\|$ визначаються формулою (6.2), а для біполярних векторів – формулою (6.3).

Функція активації для нейронів X - та Y -шару з логічною сигмоїдою має вигляд:

$$f(U_{inp}) = \frac{1}{1 + \exp(-U_{inp})},$$

де для елементів X - та Y -шарів сигнал U_{inp} визначається відповідно до формул:

$$U_{inp.Xi} = w_{0i} + \sum_{j=1}^m U_{out.Yj} w_{ji}, \quad i = \overline{1, n},$$

$$U_{inp.Yj} = w_{0j} + \sum_{i=1}^n U_{out.Xi} w_{ji}, \quad j = \overline{1, m},$$

де w_{0i}, w_{0j} – відповідно зміщення i -го нейрона X -шару та j -го нейрона Y -шару.

Аналогічно визначається функція активації для шарів нейронів з біполярними парами вхідних векторів.

Безперервні ДАП, що навіть функціонують в асинхронному режимі, в основному аналогічні дискретним синхронним версіям двонаправленої асоціативної пам'яті.

Наведені різновиди ДАП не вичерпують все відоме безліч нейромереж цього класу. Існують адаптивні, конкуруючі або змагаються ДАП, ДАП, засновані на диференціальних рівностях Кохонена-Гроссберга, багатоспрямована асоціативна пам'ять, багат шарова асоціативна пам'ять, асоціативна пам'ять, що

зберігає ланцюжки асоціацій і т.д. Архітектура та алгоритми функціонування цих мереж описані у роботах [25, 28, 55–58].

6.4. Гетероасоціативна пам'ять

Якщо в архітектурі двонаправленої асоціативної пам'яті (рис. 6.1) залишити лише зв'язки, що передають інформацію тільки в одному напрямку (наприклад, з X -шару нейронів в Y -шар або, навпаки, з Y -шару в X -шар), то ДАП перетвориться у гетероасоціативну пам'ять (рис. 6.8). Якщо в мережі число n вхідних нейронів більше числа m вихідних, то мережа може використовуватися як асоціативна пам'ять, пристрій, що розпізнає, як пристрій кодування або стиснення інформації. При цьому коефіцієнт стиснення може досягати 100. При $m > n$ мережа може також використовуватися для відновлення закодованої або стиснутої інформації.

Вага зв'язків мережі можуть визначатися за допомогою алгоритму Хебба, описаного в першому розділі.

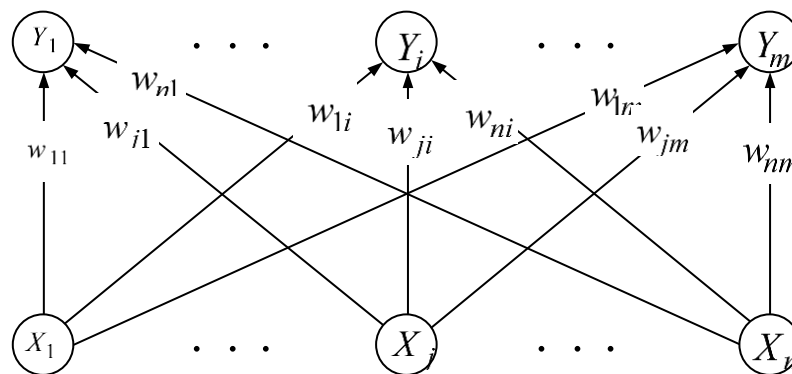


Рис. 6.8. Гетероасоціативна пам'ять

Алгоритм використання мережі як асоціативної пам'яті при біполярних нейронах та вхідних зображеннях наступний:

Крок 1. Ініціюються ваги зв'язків мережі та задається вхідне зображення

$$S^k = (S_1^k, \dots, S_n^k).$$

Крок 2. Визначаються вхідні та вихідні сигнали нейронів X -шару:

$$U_{inp.Xj} = S_j^k, \quad j = \overline{1, n},$$

$$U_{out.Xj} = U_{inp.Xj}, \quad j = \overline{1, n}$$

Крок 3. Обчислюються вхідні сигнали нейронів Y -шару:

$$U_{inp.Yi} = \sum_{j=1}^n U_{out.Xj} w_{ji}, \quad i = \overline{1, m},$$

Крок 4. Визначаються вихідні сигнали Y -елементів:

$$U_{out.Yi} = \begin{cases} 1, & \text{якщо } U_{inp.Yi} > 0, \\ 0, & \text{якщо } U_{inp.Yi} = 0, \\ -1, & \text{якщо } U_{inp.Yi} < 0, \end{cases} \quad (6.11)$$

Вектор $U_{out.Y1}, \dots, U_{out.Ym}$ і є вектором, асоціативним вхідному зображенню $S^k = (S_1^k, \dots, S_n^k)$.

Приклад 6.4. Навчимо нейронну мережу з п'ятьма вхідними ($n = 5$) та двома вихідними ($m = 2$) елементами наступним асоціаціям:

Вхідне зображення

Асоційоване зображення

$$S^1 = (-1, -1, 1, 1, 1)$$

$$T^1 = (-1, 1)$$

$$S^2 = (-1, -1, 1, -1, 1)$$

$$T^1 = (-1, 1)$$

$$S^3 = (-1, -1, 1, 1, -1)$$

$$T^1 = (-1, 1)$$

$$S^4 = (1, 1, -1, -1, -1)$$

$$T^2 = (1, -1)$$

$$S^5 = (1, -1, -1, -1, -1)$$

$$T^2 = (1, -1)$$

Крок 1. Задається множина $M = \{(S^1, T^1), (S^2, T^1), (S^3, T^1), (S^4, T^2), (S^5, T^2)\}$, що складається з пар асоційованих зображень. Ініціюються ваги зв'язків нейронів:

$$w_{ji} = 0, \quad j = \overline{1, n}, \quad i = \overline{1, m}.$$

Крок 2. Першу пару, $(S^1 = (-1, -1, 1, 1, 1), T^1 = (-1, 1))$ з множини M перевіряють на правильність реакції нейронної мережі. Оскільки всі вагові коефіцієнти дорівнюють нулю, то для першої пари необхідно виконати кроки 3 – 5.

Крок 3. Ініціюється множина входів X -нейронів:

$$U_{inp.Xj} = S_j^1, \quad j = \overline{1, 5}, \quad U_{inp.X1} = U_{inp.X2} = -1,$$

$$U_{inp.X3} = U_{inp.X4} = U_{inp.X5} = 1.$$

Крок 4. Ініціюються вихідні сигнали Y -нейронів:

$$U_{out.Y1} = t_1^1 = -1, U_{out.Y2} = t_2^1 = 1.$$

Крок 5. По правилу

$$w_{ij}(new) = w_{ij}(old) + S_j^1 t_i^1, \quad j = \overline{1, 5}, \quad i = 1, 2,$$

коригуються ваги зв'язків нейронів:

$$w_{11}(new) = 0 + (-1) \cdot (-1) = 1, \quad w_{12}(new) = 0 + (-1) \cdot 1 = -1,$$

$$w_{21} = w_{32} = w_{42} = w_{52} = w_{11} = 1,$$

$$w_{22} = w_{31} = w_{41} = w_{51} = w_{12} = -1.$$

Крок 2. Другу пару ($S^2 = (-1, -1, 1, -1, 1)$, $T^1 = (-1, 1)$) з множини M перевіряють на правильність реакції нейронної мережі. Спочатку розраховують вхідні та вихідні сигнали Y -нейронів:

$$U_{inp.Y1} = \sum_{j=1}^5 w_{j1} U_{out.Xj} = 1 \cdot (-1) + 1 \cdot (-1) + (-1) \cdot 1 + (-1) \cdot (-1) + (-1) \cdot 1 = -3,$$

$$U_{inp.Y2} = \sum_{j=1}^5 w_{j2} U_{out.Xj} = (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot 1 + 1 \cdot (-1) + 1 \cdot 1 = 3,$$

$$U_{inp.Y1}, U_{inp.Y2} = 1.$$

Оскільки $(U_{out.Y1}, U_{out.Y2}) = T^1$ то реакція мережі правильна.

Правильна реакція мережі та при пред'явленні зображень S^3, S^4 та S^5 , тобто мережа навчена.

Приклад 6.5. Перевіримо правильність функціонування гетероасоціативної пам'яті, отриманої в прикладі 6.4, при пред'явленні зображень $S^6 = (-1, -1, -1, -1, 1)$ та $S^7 = (-1, 1, -1, -1, -1)$, які людина асоціює відповідно із зображеннями T^1 та T^2 .

Крок 1. Ініціюються ваги зв'язків нейронної мережі та задається вхідне зображення S^6 .

Крок 2. Визначаються вхідні та вихідні сигнали нейронів X -шару:

$$U_{out.Xj} = U_{inp.Xj} = S_j^6, \quad j = \overline{1, 5},$$

$$U_{out.X1} = U_{out.X2} = U_{out.X3} = U_{out.X4} = -1, \quad U_{out.X5} = 1.$$

Крок 3. Визначаються вхідні сигнали нейронів Y -шару:

$$U_{inp.Y1} = \sum_{j=1}^5 U_{out.Xj} w_{j1} = (-1) \cdot 1 + (-1) \cdot 1 + (-1) \cdot (-1) + (-1) \cdot (-1) + 1 \cdot (-1) = -1,$$

$$U_{inp.Y2} = \sum_{j=1}^5 U_{out.Xj} w_{j2} = (-1) \cdot (-1) + (-1) \cdot (-1) + (-1) \cdot 1 + (-1) \cdot 1 + 1 \cdot 1 = 1.$$

Крок 4. Визначаються вихідні сигнали Y -елементів:

$$U_{out.Y1} = -1, \quad U_{out.Y2} = 1.$$

Таким чином, $(U_{out.Y1}(S^6), U_{out.Y2}(S^6)) = T^1$. Аналогічно отримаємо, що $(U_{out.Y1}(S^7), U_{out.Y2}(S^7)) = T^2$, отже, гетероасоціативна пам'ять працює правильно і при пред'явленні зображень, які не входили до навчальних даних.

6.5. Автоасоціативна мережа

Гетероасоціативна пам'ять при $n = m$, тобто при однаковій кількості нейронів у X - та Y -шарах, перетворюється на автоасоціативну пам'ять (або автоасоціативну мережу). Ця нейронна мережа може використовуватися для розпізнавання або фільтрації вхідних зображень, а також для відновлення зображень їх частинами. Для навчання мережі може використовуватися алгоритм Хебба, або краще алгоритм Хебба з подальшою корекцією елементів w_{ii} ($i = \overline{1, n}$) головної діагоналі матриці ваг

$$W = \begin{vmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{vmatrix}.$$

Приклад 6.6. Навчимо автоасоціативну мережу з біполярними нейронами за допомогою алгоритму Хебба при $n = 4$ та вхідному векторі $S^1 = (-1, 1, 1, 1)$:

Крок 1. Задається вхідне зображення S^1 та ініціюються початкові ваги зв'язків мережі: $w_{ij} = 0$, $i, j = \overline{1, 4}$.

Крок 2. Ініціюється множина входів X -нейронів:

$$U_{inp.Xj} = S_j^1, \quad j = \overline{1, 4}$$

$$U_{inp.X1} = -1, \quad U_{inp.X2} = U_{inp.X3} = U_{inp.X4} = 1.$$

Крок 3. Ініціюються вихідні сигнали Y -нейронів:

$$U_{out.Yj} = S_j^1, \quad j = \overline{1, 4},$$

$$U_{out.Y1} = -1, \quad U_{out.Y2} = U_{out.Y3} = U_{out.Y4} = 1.$$

Крок 4. По правилу

$$w_{ji}(new) = w_{ji}(old) + S_j^1 S_i^1, \quad j, i = \overline{1, 4},$$

коригуються ваги зв'язків нейронів:

$$w_{11} = 0 + (-1) \cdot (-1) = 1, \quad w_{12} = 0 + (-1) \cdot 1 = -1,$$

$$w_{12} = w_{13} = w_{14} = w_{21} = w_{31} = w_{41} = -1,$$

$$w_{11} = w_{22} = w_{23} = w_{24} = w_{32} = w_{33} = w_{34} = w_{42} = w_{43} = w_{44} = 1.$$

В результаті виходить матриця ваг

$$W = \begin{vmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{vmatrix}.$$

Перевіримо, як мережа реагує на вхідне зображення S^1 :

$$U_{inp.Y} = (U_{inp.Y1}, U_{inp.Y2}, U_{inp.Y3}, U_{inp.Y4}) = S^1 W =$$

$$(-1, 1, 1, 1) \cdot \begin{vmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{vmatrix} = (-4, 4, 4, 4).$$

При біполярних Y -елементах (6.11) маємо, що $U_{out.Y} = (-1, 1, 1, 1)$ тобто, як і вимагалось, вихідний вектор мережі збігається з вхідним.

Неважко перевірити, що викривлення будь-якої однієї компоненти вхідного вектора S^1 призводить до правильного відновлення зображення, наприклад:

$$U_{inp.Y} = (1, 1, 1, 1)W = (1, 1, 1, 1) \cdot \begin{vmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{vmatrix} = (-2, 2, 2, 2).$$

або

$$U_{inp.Y} = (-1, -1, 1, 1)W = (-1, -1, -1, -1) \cdot \begin{vmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{vmatrix} = (-2, 2, 2, 2).$$

тому при біполярних Y -елементах (6.11) маємо $U_{out.Y} = S^1$.

Двонаправлена асоціативна пам'ять дозволяє вирішувати певний клас завдань, але клас завдань дуже обмежений. За виконання реальних науково-дослідних робіт на кафедрі комп'ютерної інженерії та програмування НТУ "ХПІ" виникла потреба по одному вхідному вектору отримувати не одну, а N асоціацій, отримувати ланцюжки асоціацій тощо. Це призвело до розробки N -спрямованої асоціативної пам'яті [56], а також багатополарної асоціативної пам'яті [57, 58].

6.6. Архітектура та алгоритми функціонування N -спрямованої асоціативної пам'яті

Архітектура N -спрямованої асоціативної пам'яті виходить з архітектури двоспрямованої асоціативної пам'яті (рис. 6.1) шляхом введення в її архітектуру додаткових $(N - 1)$ сенсорних Y -шарів нейронів, пов'язаних з першим шаром сенсорних X -нейронів парами двонаправлених зважених зв'язків з відповідними рис.6.9). В обох нейронних мережах при вхідних біполярних зображеннях (векторах) функції активації всіх нейронів задаються співвідношенням

$$U_{out.p}(t+1) = \begin{cases} 1, & \text{якщо } U_{inp.p}(t) > \theta_p, \\ 0, & \text{якщо } U_{inp.p}(t) = \theta_p, \\ -1, & \text{якщо } U_{inp.p}(t) < \theta_p, \end{cases}$$

де $U_{out.p}(t+1)$ – вихідний сигнал p -го нейрона на момент часу $(t + 1)$; $U_{inp.p}(t)$ – вхідний сигнал p -го нейрона на момент часу t ; θ_p – поріг p -го нейрона.

Пропонована нейронна мережа складається із сенсорного шару елементів X_j ($j = \overline{1, n}$) та N вихідних полів нейронів: Y_{k1}^1 ($k1 = \overline{1, m1}$), Y_{k2}^2 ($k2 = \overline{1, m2}$), ..., Y_{kN}^N ($kN = \overline{1, mN}$), елементи яких пов'язані парами двонаправлених зважених зв'язків із ваговими коефіцієнтами $w_{j,k1}^{11}$, $w_{k1,j}^{12}$ ($j = \overline{1, n}$; $k1 = \overline{1, m1}$); $w_{j,k2}^{21}$, $w_{k2,j}^{22}$ ($j = \overline{1, n}$; $k2 = \overline{1, m2}$), ..., $w_{j,kN}^{N1}$, $w_{kN,j}^{N2}$ ($j = \overline{1, n}$; $kN = \overline{1, mN}$), відповідними парам нейронів, що з'єднуються. Тут перші верхні індекси вагових коефіцієнтів вказують номер вихідного поля, з яким з'єднаний X -шар елементів, а другий верхній індекс (1 або 2) вказує на напрям передачі сигналів. Якщо він дорівнює

"1", сигнал передається з шару X -нейронів до поля Y -нейронів, якщо другий верхній індекс дорівнює двом, то сигнал передається з шару Y -елементів на входи X -елементів. Перші нижні індекси вагових коефіцієнтів зв'язків вказують на нейрони, що видають вихідні сигнали, а другі нижні індекси вказують на нейрони, що сприймають сигнали.

Розроблена N -спрямована асоціативна пам'ять повинна забезпечувати запам'ятовування вихідної інформації та визначення асоціативних векторів (зображень) за вхідними векторами (зображеннями).

Процес запам'ятовування інформації нейронної мережі зводиться до обчислення N матриць вагових коефіцієнтів зв'язків, що передають сигнали від вхідного X -шару нейронів до входів N полів Y -нейронів.

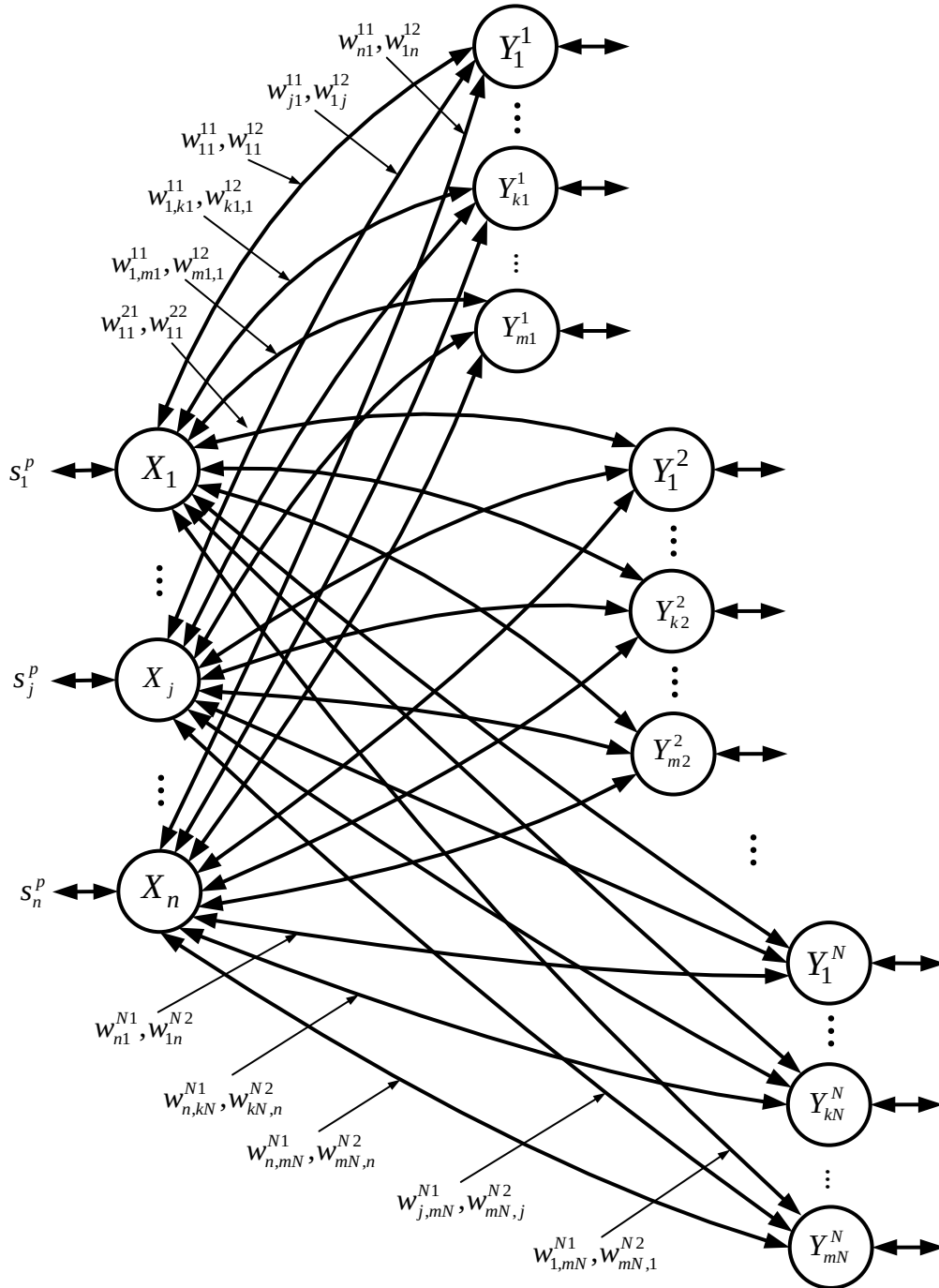


Рис. 6.9. Архітектура N -спрямованої асоціативної пам'яті

Оскільки шар X -нейронів та будь-яке з N полів Y -нейронів є двонаправленою асоціативною пам'яттю, то для обчислення матриць w^q $q = \overline{1, N}$ ваг зв'язків можна скористатися відомим з теорії ДАП співвідношенням, модифікованим для обчислення ваг зв'язків N -спрямованої асоціативної пам'яті

$$w_{j,kq}^{q1} = \sum_{p=1}^{Lq} s_j^p t_{kq}^p, \quad (6.12)$$

де $w_{j,kq}^{q1}$ – елемент матриці ваг зв'язків, що передають сигнали з виходів X -нейронів на входи нейронів Y^q -поля, $q = \overline{1, N}$, $j = \overline{1, n}$, $kq = \overline{1, mq}$, mq – число нейронів у Y^q -полі елементів; Lq – число пар асоціативних зображень, що запам'ятовуються у вагах зв'язків між X -нейронами та нейронами поля Y^q ; (s_j^p, t_{kq}^p) – елементи p -ї пари асоціативних зображень, що запам'ятовуються $(S^p = (s_1^p, s_2^p, \dots, s_n^p), t_q^p = (t_{1q}^p, t_{2q}^p, \dots, t_{mq}^p))$; S^p – вхідний вектор N -спрямованої асоціативної пам'яті, що подається на входи нейронів X -шару; t_q^p – вектор сигналів на виходах нейронів Y^q -поля. Матриці W_1^q ваг зв'язків, що передають сигнали з Y^q -полей ($q = \overline{1, N}$), мають вигляд $W_1^q = (W^q)^T$, де символ T – символ транспонування матриці.

У режимі визначення безлічі векторів, асоціативних вектору, що подається на входи X -нейронів, N -спрямована асоціативна пам'ять функціонує за наступним алгоритмом.

Крок 1. Визначається початкове значення змінної, за допомогою якої визначається активне поле Y^q -нейронів: $q = 0$, задається початковий час $t = 0$ при роботі з полем Y^q -нейронів.

Крок 2. Визначається поточне значення змінної q : $q = q + 1$.

Ініціюються ваги зв'язків Y^q -нейронів, розраховані за допомогою співвідношення (6.12) для безлічі пар асоціативних зображень $(S^p = (s_1^p, s_2^p, \dots, s_n^p), t_q^p = (t_{1q}^p, t_{2q}^p, \dots, t_{mq}^p))$, $p = \overline{1, Lq}$, обнуляються ваги зв'язків полів Y^r ($r = \overline{1, N}$, $r \neq q$).

Ініціюються нульовими вхідними сигналами всі вхідні X -нейрони та нейрони поля Y^q :

$$U_{inp.Xj} = 0, U_{out.Xj} = f_j(U_{inp.Xj}) = 0, j = \overline{1, n};$$

$$U_{inp.Y_{kq}^q} = 0, U_{out.Y_{kq}^q} = f_{kq}(U_{inp.Y_{kq}^q}) = 0, kq = \overline{1, mq},$$

де $U_{inp.Xj}$, $U_{out.Xj}$ – вхідний та вихідний сигнали нейрона X_j ; $f_j(U_{inp.Xj})$ – функція активації нейрона X_j .

Крок 3. Для заданого вхідного зображення $S^p = (s_1^p, s_2^p, \dots, s_n^p)$ виконуються кроки 4 – 9.

Крок 4. Задаються вхідні сигнали нейронів X -шару:

$$U_{inp.Xj} = s_j^p, \quad j = \overline{1, n}.$$

Задається час $t = t + 1$ та обчислюються вихідні сигнали X -елементів

$$U_{out.Xj}(t+1) = f_j(U_{inp.Xj}(t)), \quad j = \overline{1, n}.$$

Вихідні сигнали X -елементів посилаються на входи нейронів Y^q -поля.

Крок 5. Доки не встановляться вихідні сигнали всіх X - та Y^q -нейронів, виконуються кроки 6 – 8 алгоритму.

Крок 6. Адаптується активність елементів Y^q -поля.

Обчислюються вхідні та вихідні сигнали Y^q -елементів:

$$U_{inp.Y_{kq}^q}(t+1) = \sum_{j=1}^n w_{j,kq}^{q1} U_{out.Xj}(t+1), \quad kq = \overline{1, mq}; \quad (6.13)$$

$$U_{out.Y_{kq}^q}(t+2) = f_{kq}(U_{inp.Y_{kq}^q}(t+1)), \quad kq = \overline{1, mq}. \quad (6.14)$$

Вихідні сигнали Y^q -нейронів посилаються на входи елементів X -шару.

Крок 7. Адаптується активність елементів X -шару.

Обчислюються вхідні та вихідні сигнали X -елементів:

$$U_{inp.Xj}(t+2) = \sum_{kq=1}^{mq} w_{kq,j}^{q2} U_{out.Y_{kq}^q}(t+2), \quad j = \overline{1, n}; \quad (6.15)$$

$$U_{out.Xj}(t+3) = f_j(U_{inp.Xj}(t+2)), \quad j = \overline{1, n}. \quad (6.16)$$

Вихідні сигнали X -нейронів посилаються на входи елементів Y^q -шару.

Крок 8. За співвідношеннями (6.13) та (6.14), але для моментів часу $(t+3)$ та $(t+4)$ обчислюються вхідні та вихідні сигнали $U_{inp.Y_{kq}^q}(t+3)$, $U_{out.Y_{kq}^q}(t+4)$, $kq = \overline{1, mq}$ елементів Y^q -шару. Вихідні сигнали нейронів Y^q -шару посилаються на входи X -елементів.

Крок 9. Перевіряється тест на збіжність. Порівнюються вихідні сигнали X -нейронів $U_{inp.Xj}(t+3)$ та $U_{inp.Xj}(t+1)$, $j = \overline{1, n}$, а також нейронів Y^q -шару у моменти часу $(t+4)$ и $(t+2)$ $U_{out.Y_{kq}^q}(t+4)$, $U_{out.Y_{kq}^q}(t+2)$, $kq = \overline{1, mq}$, отримані на поточній та попередніх ітераціях. Якщо не виконується хоча б одна з нерівностей

$$U_{out.Xj}(t+1) = U_{out.Xj}(t+3), \quad j = \overline{1, n};$$

$$U_{out.Y_{kq}^q}(t+2) = U_{out.Y_{kq}^q}(t+4), \quad kq = \overline{1, mq},$$

то обчислюються вхідні та вихідні сигнали X -нейронів за співвідношенням (6.15), (6.16), але для моментів часу $(t+4)$ и $(t+5)$: $U_{inp.Xj}(t+4)$, $U_{out.Xj}(t+5)$. Потім

виконується привласнення $t = t + 1$ та обчислюються вхідні та вихідні сигнали Y -нейронів $U_{out.Xj}(t+1) = U_{out.Xj}(t+5)$ і здійснюється перехід на крок 6 алгоритму, інакше – на виходах нейронів Y^q -шару фіксуються їх вихідні сигнали та здійснюється перехід до кроку 10 алгоритму.

Крок 10. Перевіряється умова $q = N$, тобто на виходах чи всіх Y^q -шарів елементів отримані зображення, асоціативні вхідному зображенню S^P . Якщо умова не виконується, то перехід до кроку алгоритму 2, інакше – до кроку 11.

Крок 11. Зупинка.

Мережа може також функціонувати в режимі подачі зображень на входи нейронів одного з Y^q -шарів ($q = \overline{1, N}$). При цьому на виході нейронів X -шару з'являється зображення S^* , асоціативне зображення, що подається на входи Y^q -шару. Якщо це зображення подати на входи нейронів X -шару, то на виходах всіх Y^q -шарів з'явиться N векторів, асоціативних вектору S^* .

Таким чином, на основі двоспрямованої асоціативної пам'яті вперше розроблено архітектуру та алгоритми функціонування багатоспрямованої асоціативної пам'яті, яка здатна запам'ятовувати та відновлювати за вхідним зображенням (вектором) N асоціативних йому зображень.

6.7. Багатшарова двоспрямована асоціативна пам'ять

Двонаправлена асоціативна пам'ять (ДАП) складається з двох сенсорних шарів елементів (рис. 6.1), пов'язаних між собою парами зважених двонаправлених зв'язків із відповідними ваговими коефіцієнтами. ДАП здатні запам'ятовувати та відновлювати зі своєї пам'яті пари асоціативних зображень. Однак часто буває необхідно запам'ятовувати не пари, а ланцюжки асоціацій (рис. 6.10) [57, 58].

Кожен нейрон $(Z_1^1, \dots, Z_{q1}^1, \dots, Z_{g1}^1)$ першого з N шарів введених у ДАП нейронів з'єднується двонаправленими зваженими зв'язками з кожним нейроном $(X_1, \dots, X_i, \dots, X_n)$ першого сенсорного шару. Ці два шари нейронів, за відповідних ваг зв'язків між ними, фактично є двонаправленою асоціативною пам'яттю (рис. 6.1), яка за наявності зображення на входах нейронів X_i ($i = \overline{1, n}$) першого сенсорного шару визначає асоціативне зображення на виходах нейронів першого шару Z^1 -елементів.



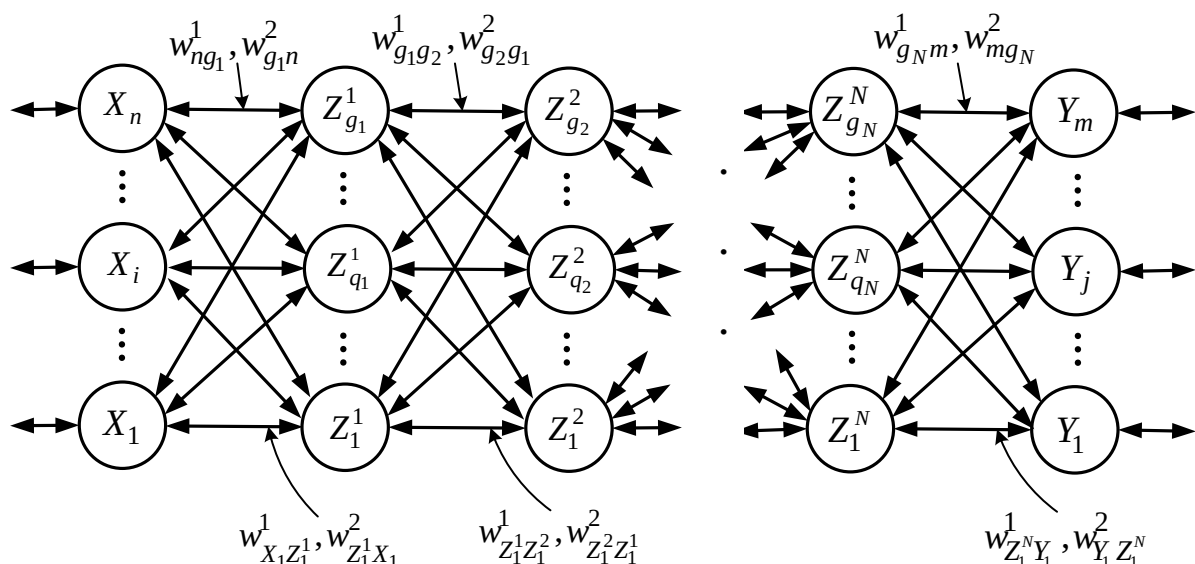


Рис. 6.10. Багатошарова асоціативна пам'ять

І навпаки, за наявності зображення, яке надійшло з виходів елементів Z^2 -шля на входи нейронів Z^1 -шля, визначає асоціативне зображення на виходах елементів X -шля. Оскільки кожен нейрон останнього N шарів Z -нейронів з'єднується двонаправленими зваженими зв'язками з кожним нейроном Y_j ($j=1, m$) другого сенсорного шару, то при відповідних вагах зв'язків між нейронами цих шарів вони також утворюють двонаправлену асоціативну пам'ять ДАП $_{(N/2)+1}$ (рис. 6.11), пари асоціативних зображень, що зберігаються у вагах своїх зв'язків, які можуть надходити відповідно на входи шару Y -елементів або на входи шару Z^N -елементів. Аналогічно блоки ДАП можуть бути отримані з нейронів шарів Z^2 і Z^3 , Z^4 і Z^5 (відповідно блоки ДАП $_2$ і ДАП $_3$ на рис. 6.11). Таким чином, виходить архітектура нейронної мережі, що складається з послідовно з'єднаних блоків ДАП.

Якщо окрема ДАП здатна запам'ятовувати лише пари асоційованих зображень один з одним $S^p = (s_1^p, \dots, s_n^p)$, $T^p = (t_1^p, \dots, t_m^p)$ з деяких заданих множин зображень $S = (s^1, \dots, s^p, \dots, s^L)$, $T = (T^1, \dots, T^p, \dots, T^L)$, де L – кількість асоційованих пар зображень, то багатошарова асоціативна пам'ять здатна запам'ятовувати ланцюжки асоціацій з елементів множин зображень S , Z^1 , Z^2 , ..., $Z^{N/2}$, T . Тут S – множина вхідних зображень, що подаються на входи X -шля нейронів (рис. 6.10) або на входи ДАП $_1$ (рис. 6.11); $Z^1 = \{z^{11}, \dots, z^{1p}, \dots, z^{1L}\}$ – множина вихідних зображень ДАП $_1$, які є одночасно вхідними зображеннями для ДАП $_2$ (або шару Z^2 -нейронів) і породжують на виході ДАП $_2$ (або шару

Z^3 -нейронів) множину зображень $Z^2 = \{z^{21}, \dots, z^{2p}, \dots, z^{2L}\}$, які є вхідними зображеннями для ДАП₃ і т.д.

Зазначимо, що багатошарова асоціативна пам'ять (рис. 6.10, рис. 6.11) є двонаправленою пам'яттю, тобто вхідним шаром нейронної мережі може бути не тільки шар X -нейронів, а й шар Y -нейронів.

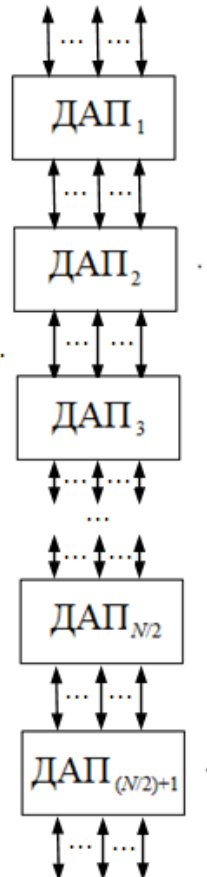


Рис. 6.11. Багатошарова асоціативна пам'ять у вигляді блоків ДАП

Розроблена мережа функціонує відповідно до двох алгоритмів: навчання та розпізнавання. Алгоритм навчання являє собою процес визначення набору навчальних зображень та розрахунок матриць ваг зв'язків між шарами нейронів X та Z^1, Z^2 та $Z^3, \dots, Z^N$ і Y за відомими співвідношення виду (6.2) та (6.3).

Алгоритм функціонування першого або останнього блоку багатошарової мережі при подачі зображення відповідно на перший або другий сенсорні входи багатошарової пам'яті не відрізняється від алгоритму функціонування ДАП. Асоціативні зображення, отримані на виході першого або останнього блоку ДАП, служать вхідними зображеннями відповідно для другого або передостаннього блоку ДАП багатошарової пам'яті і т.д.

Контрольні питання

1. Нарисуйте архітектуру нейронної мережі двонаправлена асоціативна пам'ять.
2. Як визначається матриця W ваг зв'язків для сигналів з X -шару елементів Y -шару?
3. Як визначається матриця W_1 ваг зв'язків для сигналів з Y -шару елементів у X -шар?
4. Як пов'язані матриці W та W_1 ?
5. Нарисуйте структуру, яка відображає функціонування ДАП при подачі зображень на X -шар елементів.
6. Як визначаються елементи матриці W при бінарних елементах ДАП?
7. Як визначаються елементи матриці W при біполярних елементах ДАП?
8. Як визначається функція активації нейронів ДАП при бінарних входних векторах?
9. Наведіть алгоритми функціонування ДАП на бінарних нейронах.
10. Наведіть приклад навчання ДАП на бінарних нейронах запам'ятовування двох пар асоціативних зображень.
11. Наведіть оцінки ємності двонаправленої асоціативної пам'яті.
12. У чому відмінність безперервних та дискретних ДАП?
13. Чим відрізняється автоасоціативна пам'ять від гетероасоціативної?
14. Чим відрізняється гетероасоціативна пам'ять від ДАП?
15. Як виконується навчання гетеро асоціативної пам'яті?
16. Наведіть архітектуру N -спрямованої асоціативної пам'яті.
17. Як виконується навчання автоасоціативної пам'яті?
18. Як виконується навчання N -спрямованої асоціативної пам'яті?

НЕЙРОННІ МЕРЕЖІ НА ОСНОВІ ТЕОРІЇ АДАПТИВНОГО РЕЗОНАНСУ

7.1. Введення в теорію адаптивного резонансу

У більшості нейронних мереж, які навчаються методом зворотного розповсюдження, генетичними алгоритмами, у двонаправленій асоціативній пам'яті, мережах Хопфілда і т.д. дуже часто навчання новому образу, ситуації чи асоціації помітно спотворює чи навіть знищує плоди попереднього навчання, вимагаючи зміни значної частини ваги зв'язків чи повного перенавчання мережі. У цьому відношенні зазначені нейронні мережі різко відрізняються від мозку людини, який, безперервно обробляючи потоки інформації із зовнішнього середовища, може як модифікувати і уточнювати образи, що зберігаються в пам'яті, так і створювати нові, не знищуючи те, що вже зберігається. Таким чином, мозок людини має високу пластичність до інформації, що надходить, що дозволяє їй сприймати нові образи і уточнювати збережену інформацію за вже відомими, і в той же час вона має і високу стабільність, зберігаючи раніше отримані знання. Неможливість за допомогою вже відомих нейронних мереж вирішити проблему стабільності-пластичності стала однією з основних причин розробки нових конфігурацій нейромереж. Приклад таких мереж є нейромережі, отримані на основі адаптивної резонансної теорії (adaptive resonance theory (ART)), розробленої Гроссбергом і Карпентером [59 – 61]. Ці мережі певною мірою дозволяють вирішувати суперечливі завдання чутливості до нових даних та збереження отриманих знань.

Нейронна мережа адаптивної резонансної теорії (АРТ) відносить вхідне зображення до одного з відомих класів, якщо воно достатньо подібне або резонує з прототипом цього класу. Якщо знайдений прототип з певною точністю, що задається спеціальним параметром подібності, відповідає вхідному зображенню, він модифікується, щоб стати більш схожим на пред'явлене зображення. Коли вхідне зображення недостатньо подібне до жодного з наявних прототипів, то на його основі створюється новий клас. Це можливо завдяки тому, що мережа має велику кількість надлишкових або нерозподілених елементів, які не використовуються доти, поки в цьому немає необхідності (якщо немає нерозподілених нейронів, то вхідне зображення не викликає реакції мережі). Таким чином, нові образи можуть створювати нові класи, але не можуть спотворити існуючу пам'ять.

резонансної теорії, зокрема мережі АРТ-1 та АРТ-2. АРТ-1 призначена для роботи з двійковими вхідними зображеннями або векторами, а АРТ-2 – для класифікації як двійкових, так і безперервнозначних векторів. Хоча деталі архітектури та алгоритмів роботи для АРТ-1 та АРТ-2 різні, проте вони мають загальну базову архітектуру.

7.2. Базова архітектура мереж АРТ

Базова архітектура мереж АРТ включає три групи нейронів: поле F_1 вхідних обробних нейронів, що складається з двох шарів елементів, шар нейронів, що розпізнають, і групу нейронів управління (рис. 7.1).

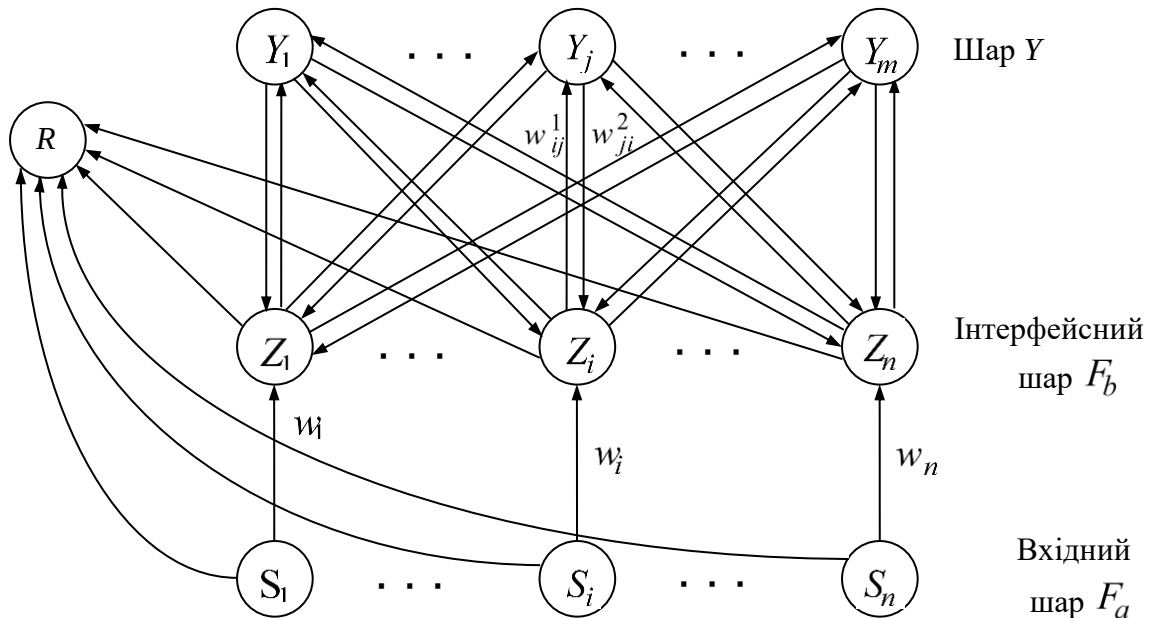


Рис. 7.1. Базова архітектура мереж АРТ

Поле F_1 нейронів складається з двох шарів: вхідний шар F_a та інтерфейсний шар F_b . Вхідний шар сприймає зображення і передає отриману інформацію нейронам інтерфейсного шару F_b и управляючому нейрону R . Кожен елемент інтерфейсного шару F_b пов'язаний з кожним елементом шару, що розпізнає Y двома множинами зв'язаних зв'язків. Сигнали з інтерфейсного шару шар Y передаються зв'язками, що йдуть знизу вгору (з вагами w_{ij}^1), а з шару, що розпізнає, в інтерфейсний – зв'язками, що йдуть зверху вниз (з вагами w_{ji}^2). Через наявність великої кількості зв'язків на малюнку наведено позначення лише однієї пари зв'язків між інтерфейсними та елементами, що розпізнають.

Шар Y є шаром нейронів, що конкурують або змагаються. Будь-коли кожен елемент Y_j ($j = \overline{1, m}$) розпізнає шар знаходиться в одному з трьох станів:

- активний ($U_{out.Yj} = 1$; $d = 1$ для АРТ-1 і $0 < d < 1$ для АРТ-2);
- неактивний ($U_{out.Yj} = 0$, але здатний брати участь у змаганні);
- загальмований ($U_{out.Yj} = -1$ та не допущений до змагань при пред'явленні поточного вхідного вектора).

Після пред'явлення вхідного зображення активним залишається лише один нейрон, що розпізнає, всі інші Y -елементи мають нульові або негативні вихідні сигнали. Виділений нейрон, що розпізнає, допускається до навчання вхідним зображенням тільки в тому випадку, якщо його ваговий вектор зв'язків з шару Y в шар F_b подібний до вхідного вектора. Це рішення приймається за допомогою R -нейрону на основі спеціального параметра, що отримав назву параметра подібності, і сигналів, що надходять із вхідного та інтерфейсного шару елементів. Через допоміжні елементи, різні для мереж АРТ-1 і АРТ-2, проводиться або навчання виділеного Y -елемента, що розпізнає, або його загальмовування з подальшим виключенням з числа змагаються при повторних пред'явленнях цього ж вхідного зображення, коли виділяються нові кандидати для навчання вхідним зображенням.

У найбільш загальному вигляді алгоритм функціонування АРТ нейронної мережі можна подати у такому вигляді [25, 28]:

Крок 1. Ініціюються параметри мережі.

Крок 2. Доки не дотримуються умов зупинки, виконуються кроки 3 – 10.

Крок 3. Для кожного вхідного вектора або зображення виконуються кроки 4 – 9.

Крок 4. Пред'являється вхідний вектор та обчислюються вихідні сигнали нейронів вхідного шару F_a .

Крок 5. Поки не дотримуються умови скидання або повернення до пошуку нового Y -нейрону, виконуються кроки 6 – 8.

Крок 6. Знаходиться незагальмований Y -елемент, що має найбільший вихідний сигнал.

Крок 7. Обчислюються вихідні сигнали нейронів інтерфейсного шару F_b .

Крок 8. За допомогою параметра подібності перевіряються умови скидання чи повернення (вони різні мереж АРТ-1 і АРТ-2). Якщо вони виконуються, тоді виділений Y -елемент загальмовується і провадиться повернення до кроку 5. Якщо умови скидання не виконуються, тоді виділений кандидат із Y -шару допускається до навчання на кроці 9.

Крок 9. Проводиться навчання виділеного Y -елемента.

Крок 10. Перевіряються умови зупинки. Якщо вони не виконуються, то перехід до кроку 2, інакше – перехід до кроку 11.

Крок 11. Зупинка.

Хоча мережі АРТ не висувають вимог до порядку появи вхідних зображень і не вимагають появи всіх зображень з однаковою частотою, при їх навчанні використовується поняття епохи (послідовного пред'явлення кожного з навчальних зображень). Процес навчання мереж АРТ може займати багато епох.

Нейронні мережі АРТ – динамічні об'єкти, що описуються системами звичайних диференціальних рівнянь, тому їхнє навчання в загальному випадку досить трудомістке. Однак моделі мереж АРТ можуть бути спрощені, якщо

припустити, що зміна вихідних сигналів нейронів відбувається набагато швидше, ніж зміна векторів вагових їх зв'язків. Тому в нейромережах теорії адаптивного резонансу можна вважати, що після виділення для навчання прийнятного Y -елемента (настання резонансу між зображенням, що пред'явлене і зберігається в пам'яті), вихідні сигнали всіх нейронів залишаються незмінними протягом тривалого періоду часу, протягом якого відбуваються зміни ваг зв'язків.

В АРТ розрізняють два типи навчання: швидке та повільне. Вони відрізняються як теоретичними передумовами, і своїми робочими характеристиками. У методі швидкого навчання ваги протягом тривалого часу резонансу досягають рівноважного стану при кожному пред'явленні зображень. Ця форма навчання є типовою для двійкових нейронних мереж АРТ-1 і буде детально розглянута в наступних параграфах. У методі повільного навчання має місце істотно повільніша зміна ваг під час резонансу і вони не досягають рівноважних значень при кожному пред'явленні вхідних зображень. Цей метод більш властивий нейромереж АРТ-2, хоча теоретично може застосовуватися (але, практично не застосовується) для навчання та дискретних мереж АРТ-1. Він вимагає великої кількості пред'явлень вхідних зображень, але щодо невеликого об'єму обчислень при кожному пред'явленні. Докладніше метод повільного навчання та мережі АРТ-2 описані в роботах [28, 61, 62].

7.3. Архітектура нейронних мереж АРТ-1

Нейронные сети АРТ-1 рассчитаны на работу с бинарными входными изображениями или векторами. Их общая архитектура приведена на рис. 7.2 и отличается от базовой архитектуры сетей АРТ (рис. 7.1) наличием дополнительных элементов G_1 и G_2 , обеспечивающих управление процессом обучения. На рис. 7.2, как и на предыдущем рисунке, из-за наличия большого числа связей между Z - и Y -слоями элементов приведены обозначения только одной обобщенной пары весов w_{ij}^1, w_{ji}^2 связей между интерфейсными и распознающими нейронами. Большинство связей, приведенных на рис. 7.2, являются возбуждающими: от входного слоя элементов F_a к нейронам R, G_1 и F_b -слоя, от нейронов G_1, G_2 соответственно к нейронам слоев F_b и Y . Тормозящие сигналы передают только множества связей от интерфейсных элементов к R -нейрону, от Y -нейронов к элементу G_1 , и от R -нейрона к Y -нейрону для затормаживания победившего Y -нейрона. Все связи сети АРТ-1 передают только бинарные сигналы 0 или 1.

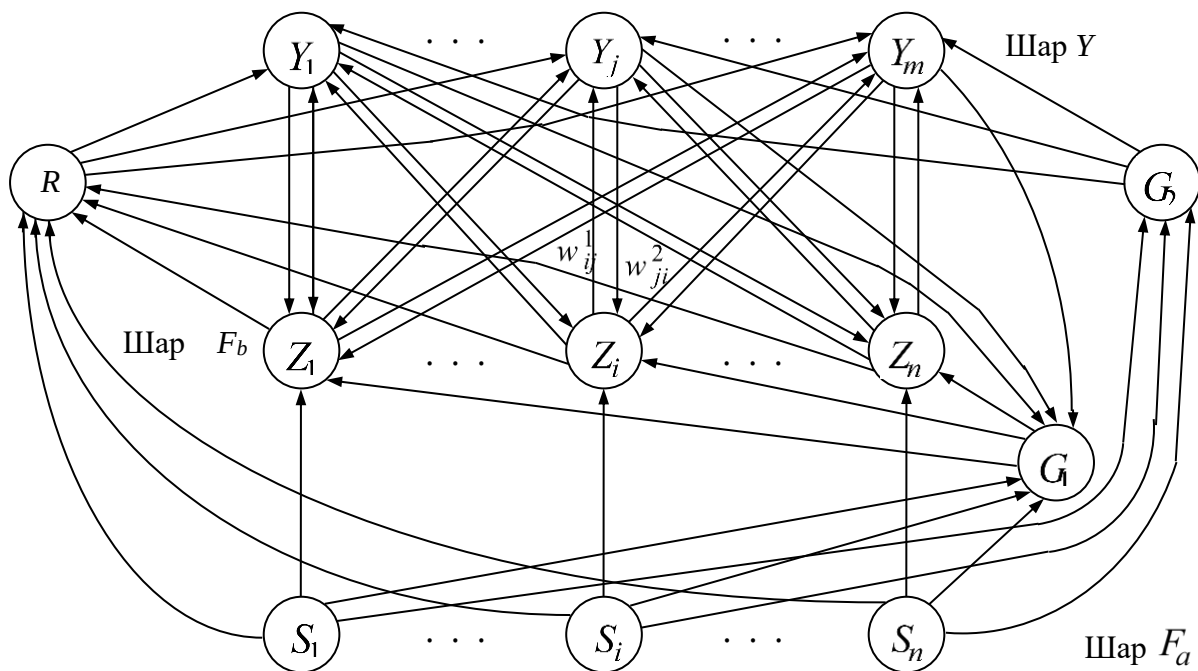


Рис. 7.2. Архітектура нейронних мереж АРТ-1

Кожен елемент в інтерфейсному або Y шарі мережі АРТ-1 має три джерела вхідних сигналів. Довільний інтерфейсний елемент Z_i ($i = \overline{1, n}$) може отримувати сигнали від елемента S_i вхідного шару, з вершин Y -шару та від нейрона G_1 . Аналогічно, елемент Y_j ($j = \overline{1, m}$) може отримувати сигнали від інтерфейсних елементів, нейронів R та G_2 . Для перекладу нейронів інтерфейсного або розпізнає шарів в активний одиничний стан необхідна наявність вхідних сигналів збудливих з двох джерел. Оскільки кожен з аналізованих нейронів має три можливі джерела сигналів, то умова порушення цих нейронів отримала назву "правила два з трьох".

У вихідному стані нейрони R , G_1 , G_2 та вхідного шару F_a мають нульові вихідні сигнали (перебувають у стані "0"). При подачі на входи S -елементів бінарних компонент зображення, що пред'являється частина з них, що отримали одиничні вхідні сигнали, переходить в стан "1". Збудливі сигнали з виходів цих нейронів переводять у стан "1" нейрони G_1 , G_2 та R , а також надходять на входи відповідних нейронів інтерфейсного шару. Нейрони інтерфейсного шару, що отримали одиничні сигнали від нейронів вхідного шару та елемента G_1 , за правилом два з трьох переходять в активний стан і посиляють свої збудливі сигнали зв'язків з вагами w_{ij}^1 на входи нейронів розпізнає шар. Нейрони шару, що

розпізнає, переходять в активний стан також за правилом два з трьох, отримуючи збудливі сигнали не тільки від елементів інтерфейсного шару, але і від елемента G_2 . Після цього у шарі Y -нейронів відбувається латеральний процес та виділяється єдиний Y -нейрон. Усі нейрони Y -шару, крім перемігшого, переходять у стан "0", а нейрон, що переміг, – у стан "1". Одиничний сигнал нейрона, що переміг, загальмовує керуючий нейрон G_1 , а також надходить по зв'язках з вагами w_{ji}^2 на входи елементів інтерфейсного шару. Оскільки елементи інтерфейсного шару підпорядковуються правилу два з трьох, то без збудливих сигналів від нейрона G_1 , в активному стані залишаться тільки ті інтерфейсні елементи, які отримують одиничні сигнали і від елемента вхідного шару, і від нейрона, що переміг розпізнає шару. Гальмівні сигнали активних елементів інтерфейсного шару надходять на входи R -елемента, який також отримує збуджуючі сигнали від нейронів вхідного шару. Залежно від співвідношення величин збуджуючих та гальмівних сигналів R -елемент переходить або в стан "0", або в стан "1". При нульовому вихідному сигналі R -елемента в мережі настає резонанс і відбувається навчання ваг зв'язків Y -нейрона, що переміг, а при одиничному вихідному сигналі – переможець Y -нейрон загальмовується і позбавляється можливості брати участь у змаганнях при повторних пред'явленнях поточного зображення. Потім у Y -шарі проводиться вибір нового нейрона, що переміг.

7.4. Алгоритм навчання мереж АРТ-1

В основу аналізованого алгоритму покладено метод швидкого навчання, що передбачає, що ваги Y -нейрона, що переміг, досягають рівноважних значень при кожному пред'явленні навчального вектора або зображення. Диференціальні рівняння, що описують зміни ваги переміг Y -елемента (позначеного індексом J) в загальному випадку мають вигляд [28]:

$$\frac{dw_{iJ}^1}{dt} = A_1[-E_{ij}w_{ij}^1 + U_{out.Zi}], \quad (7.1)$$

$$\frac{dw_{iJ}^2}{dt} = A_2[-E_{ij}w_{ij}^2 + U_{out.Zi}],$$

де w_{iJ}^1 – вага зв'язку від елемента Z_i до елемента, що переміг Y_J ; w_{ji}^2 – вага зв'язку від елемента Y_J до елемента Z_i ; t – час; A_1, A_2 – постійні коефіцієнти; E_{ij}, E_{ji} – визначувані функції; $U_{out.Zi}$ – вихідний сигнал i -го Z -нейрону.

Функція E_{ij} у першому диференціальному рівнянні системи (7.1) задається прямо пропорційно нормі вектора вихідних сигналів нейронів інтерфейсного шару [28]:

$$E_{ij} = U_{out.Zi} + L^{-1} \sum_{\substack{k=1 \\ K \neq i}}^n U_{out.Zk} \quad (7.2)$$

де L – позитивна константа та $L = A_1 / K$, а норма вектора окреслюється сума його компонент.

З урахуванням співвідношення (7.2) перше рівняння системи (7.1) перетворюється на вид:

$$\frac{dw_{ij}^1}{dt} = KL[-w_{ij}^1 U_{out.Zi} - w_{ij}^1 L^{-1} \sum_{\substack{k=1 \\ K \neq i}}^n U_{out.Zk} + U_{out.Zi}]$$

або

$$\frac{dw_{ij}^1}{dt} = K[(1 - w_{ij}^1)L U_{out.Zi} - w_{ij}^1 L^{-1} \sum_{\substack{k=1 \\ K \neq i}}^n U_{out.Zk}], \quad (7.3)$$

Якщо $U_{out.Zi} = 0$, то рівняння (7.3) набуває вигляду:

$$\frac{dw_{ij}^1}{dt} = -K w_{ij}^1 \|U_{out.Z}\|, \quad (7.4)$$

де $\|U_{out.Z}\| = \sum_{\substack{k=1 \\ k \neq i}}^n U_{out.Zk}$ – норма вектора вихідних сигналів нейронів інтерфейсного шару.

Оскільки у рівноважному стані

$$\frac{dw_{ij}^1}{dt} = 0, \quad (7.5)$$

то із співвідношень (7.4) і (7.5) випливає, що

$$w_{ij}^1 = 0 \quad (7.6)$$

Якщо елемент Z_i активний, тобто $U_{out.Zi} = 1$, то $\sum_{\substack{k=1 \\ k \neq i}}^n U_{out.Zk} = \|U_{out.Z}\| - 1$ і

рівняння (7.3) перетворюється на вигляд:

$$\frac{dw_{ij}^1}{dt} = K[(1 - w_{ij}^1)L - w_{ij}^1 \|U_{out.Z}\| - 1]. \quad (7.7)$$

З рівняння (7.7) за умови (7.5) неважко отримати співвідношення для рівноважних значень ваг

$$w_{ij}^1 = \frac{L}{L-1 + \|U_{out.Z}\|}. \quad (7.8)$$

Формули (7.6) та (7.8) для рівноважних значень ваг можна об'єднати за допомогою одного виразу:

$$w_{ij}^1 = \frac{LU_{out.Zi}}{L-1 + \|U_{out.Z}\|}, \quad (7.9)$$

так як $U_{out.Zi} = 0$ якщо нейрон Z_i пасивен и $U_{out.Zi} = 1$, якщо нейрон активний.

Вираз для рівноважних ваг w_{ji}^2 ($i = \overline{1, n}$) виходить простіше, оскільки у другому рівнянні системи (7.1) згідно з даними роботи [28] можна покласти: $A_2 = E_{ji} = 1$. У зв'язку з цим рівняння перетворюється на вигляд

$$\frac{dw_{ij}^2}{dt} = w_{ij}^2 + U_{out.Zi}. \quad (7.10)$$

Оскільки у рівноважному стані похідна у лівій частині співвідношення (7.10) має бути рівною нулю, то маємо:

$$w_{ij}^2 = U_{out.Zi}. \quad (7.11)$$

На основі методу швидкого навчання для нейромереж АРТ-1 може використовуватися наступний алгоритм [28].

Алгоритм навчання нейромереж АРТ-1

В алгоритмі прийнято такі позначення:

m – максимальна кількість елементів, що розпізнають, у Y -шарі або максимальна кількість розпізнаваних класів вхідних зображень;

n – число компонентів у вхідному векторі або зображенні;

S^k – n -вимірний бінарний вхідний вектор; $k = \overline{1, q}$;

q – кількість навчальних вхідних векторів;

$U_{out.Z}$ – n -вимірний бінарний вектор вихідних сигналів інтерфейсного шару елементів;

$\|X\|$ – норма вектора X ;

w_{ij}^1 – вага зв'язку від інтерфейсного елемента Z_i ($i = \overline{1, n}$) до елемента Y_j ($j = \overline{1, m}$), діапазон допустимих початкових значень: $0 < w_{ij}^1 < \frac{1}{L-1+n}$;

рекомендоване початкове значення: $\frac{1}{1+n}$;

w_{ij}^2 – вага зв'язку від елемента Y_j ($j = \overline{1, m}$) до елемента Z_i ($i = \overline{1, n}$), рекомендоване початкове значення: $w_{ij}^2 = 1$;

p – параметр подібності між пред'явленим вектором і вектором ваг нейрона, що переміг Y -шару, діапазон допустимих значень параметра: $0 < p \leq 1$; рекомендоване значення: $p \approx 0,9$.

Крок 1. Ініціюються параметри L, p та ваги w_{ij}^1 і w_{ji}^2 ($i = \overline{1, n}, j = \overline{1, m}$).

Крок 2. Поки не виконуються умови зупинки, реалізуються кроки 3 – 14.

Крок 3. Для кожного навчального вхідного вектора S^k ($k = \overline{1, q}$) виконуються кроки 4 – 13.

Крок 4. Задається нульова активація всіх елементів Y -шару, що розпізнають: $U_{out.Yj} = 0, j = \overline{1, m}$. Вхідним навчальним вектором $S^k = (s_1^k, \dots, s_n^k)$ встановлюється активація S -елементів вхідного шару:

$$U_{out.Si} = s_i^k, i = \overline{1, n}.$$

Крок 5. Обчислюється норма вектора вихідних сигналів вхідного шару:

$$\|U_{out.S}\| = \|S^k\| = \sum_{i=1}^n s_i^k.$$

Крок 6. Формують вхідні та вихідні сигнали елементів інтерфейсного шару

$$U_{inp.Zi} = U_{out.Si}, U_{out.Zi} = U_{inp.Zi}, i = \overline{1, n}.$$

Крок 7. Для кожного незагальмованого Y -нейрона, тобто у якого вихідний сигнал не дорівнює -1 , розраховуються його вхідний та вихідний сигнали:

$$\text{якщо } U_{out.Yi} \neq -1, \text{ то } U_{inp.Yi} = U_{out.Yi} = \sum_{i=1}^n w_{ij}^1 U_{out.Zi}.$$

Крок 8. Поки не знайдено Y -нейрон, ваговий вектор якого при заданому параметрі подібності p відповідає вхідному вектору S^k , виконуються кроки 9 – 12.

Крок 9. У Y -шарі визначається нейрон Y_j , що задовольняє умові $U_{out.Yj} \geq U_{out.Yi}$, $j = \overline{1, m}$. Якщо таких нейронів кілька, то вибирається елемент із найменшим індексом. Якщо $U_{out.Yj} = -1$, то всі елементи загальмовані та вхідне зображення не може бути класифіковано.

Крок 10. Розраховуються вихідні сигнали Z -елементів:

$$U_{out.Zi} = U_{out.Si} w_{ji}^2.$$

Крок 11. Обчислюється норма вектора вихідних сигналів інтерфейсного шару:

$$\|U_{out.Z}\| = \sum_{i=1}^n U_{out.Zi}.$$

Крок 12. Перевіряється умова можливість навчання виділеного Y -нейрона:

$$\text{якщо } \frac{\|U_{out.Z}\|}{\|S^k\|} < p, \text{ то } U_{out.Yj} = -1.$$

тобто загальмовується елемент Y_j і продовжується виконання з кроку 9;

$$\text{якщо } \frac{\|U_{out.Z}\|}{\|S^k\|} > p \text{ то перехід до кроку 13.}$$

Крок 13. Адаптуються ваги зв'язків елемента Y_j :

$$w_{ij}^1 = \frac{L U_{out.Zi}}{L - 1 + \|U_{out.Z}\|}, w_{ij}^2 = U_{out.Zi}, i = \overline{1, n}.$$

Крок 14. Перевіряються умови зупинки. Умовами зупинки можуть бути: відсутність змін ваг w_{ij}^1, w_{ji}^2 мережі, досягнення заданої кількості епох тощо. буд. Якщо умови зупинки не виконуються, то перехід до кроку 2 алгоритму, інакше – до кроку 15.

Крок 15. Зупинка.

Розглянемо у деталях застосування описаного алгоритму на простих прикладах.

Приклад 7.1. Використання алгоритму АРТ-1 для класифікації чотирьох векторів $(1, 1, 0, 0, 0)$, $(0, 0, 1, 1, 0)$, $(1, 0, 0, 0, 0)$, $(0, 0, 0, 1, 1)$.

Для розв'язання задач класифікації використовуємо алгоритм АРТ-1 з наступними параметрами:

$m = 4$ – максимальна кількість створюваних кластерів;

$n = 5$ – розмірність вхідних векторів;

$q = 4$ – число вхідних векторів;

$p = 0,45$ – параметр подібності;

$L = 2$ – параметр для адаптації ваг w_{ij}^1 , $i = \overline{1,5}$, $j = \overline{1,4}$;

$w_{ij}^1(t=0) = \frac{1}{1+n} = 0,167$ – початкові ваги w_{ij}^1 , $i = \overline{1,5}$, $j = \overline{1,4}$;

$w_{ji}^2(t=0) = 1$ – початкові ваги w_{ji}^2 , $j = \overline{1,4}$, $i = \overline{1,5}$.

Застосування алгоритму дає таке:

Крок 1. Ініціюються параметри та початкові ваги зв'язків: $L = 2$; $p = 0,45$;

$w_{ij}^1(0) = 0,167$, $w_{ji}^2(0) = 1$, $i = \overline{1,5}$, $j = \overline{1,4}$.

Крок 2. Початок обчислень.

Крок 3. Для першого вхідного вектора $S^1 = (1, 1, 0, 0, 0)$ виконуються кроки 4 – 13.

Крок 4. Визначаються вихідні сигнали елементів Y-шару:

$$U_{out.Yj} = 0, i = \overline{1,5}, j = \overline{1,4}.$$

Вхідним навчальним вектором визначаються вихідні сигнали S-елементів:

$$U_{out.Si} = s_i^k, i = \overline{1,5}.$$

$$U_{out.S1} = 1, U_{out.S2} = 1, U_{out.S3} = U_{out.S4} = U_{out.S5} = 0.$$

Крок 5. Обчислюється норма вектора вихідних сигналів вхідного шару:

$$\|U_{out.S}\| = \sum_{i=1}^5 U_{out.Si} = 2.$$

Крок 6. Формуються вектори вхідних та вихідних сигналів елементів інтерфейсного шару:

$$U_{inp.Z} = (1, 1, 0, 0, 0), U_{out.Z} = (1, 1, 0, 0, 0).$$

Крок 7. Обчислюються вхідні сигнали всім елементів Y-шару:

$$U_{inp.Y1} = \sum_{i=1}^5 w_{i1} U_{out.Zi} = 0,167 \cdot 1 + 0,167 \cdot 1 + 0,167 \cdot 0 + 0,167 \cdot 0 + 0,167 \cdot 0 = 0,334,$$

$$U_{inp.Y2} = U_{inp.Y3} = U_{inp.Y4} = U_{inp.Y1} = 0,334.$$

Крок 8. Поки не знайдено Y -нейрон, ваговий вектор якого відповідно до заданого параметра подібності p відповідає пред'явленому вектору, виконуються кроки 9 – 12.

Крок 9. Оскільки всі Y -елементи мають однакові вхідні сигнали, то нейроном-переможцем вибирається перший незагальмований Y -нейрон.

Крок 10. Розраховуються вихідні сигнали Z -елементів:

$$U_{out.Zi} = w_{ji}^2 U_{out.Si}, \quad i = \overline{1, 5}$$

Так як $U_{out.S} = (1, 1, 0, 0, 0)$ та $w_{ij}^2 = (1, 1, 1, 1, 1)$, то маємо $U_{out.Z} = (1, 1, 0, 0, 0)$.

Крок 11. Обчислюється норма вектора $U_{out.Z}$

$$\|U_{out.Z}\| = \sum_{i=1}^5 U_{out.Zi} = 2.$$

Крок 12. Перевіряється подібність векторів $S^1 = (1, 1, 0, 0, 0)$ і $U_{out.Z}$, оскільки

$$\frac{\|U_{out.Z}\|}{S^1} = \frac{2}{2} = 1 > p = 0,45,$$

то на наступному кроці виконується адаптація ваг нейрона Y_1 .

Крок 13. За допомогою формули (7.9) визначаються рівноважні ваги w_{i1}^1 :

$$w_{ij}^1 = \frac{L U_{out.Zi}}{L - 1 + \|U_{out.Z}\|} = \frac{2 U_{out.Zi}}{3}, \quad i = \overline{1, 5},$$

$$w_{11}^1 = w_{21}^1 = 0,667, \quad w_{31}^1 = w_{41}^1 = w_{51}^1 = 0.$$

За допомогою формули (7.11) визначаються рівноважні ваги w_{li}^2 : $w_{11}^2 = w_{12}^2 = 1, \quad w_{13}^2 = w_{14}^2 = w_{15}^2 = 0$.

В результаті адаптації ваг отримаємо наступні вагові матриці:

$$W^1 = \begin{vmatrix} 0,667 & 0,167 & 0,167 & 0,167 \\ 0,667 & 0,167 & 0,167 & 0,167 \\ 0 & 0,167 & 0,167 & 0,167 \\ 0 & 0,167 & 0,167 & 0,167 \\ 0 & 0,167 & 0,167 & 0,167 \end{vmatrix}, \quad W^2 = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Крок 3. Для другого вхідного вектора $S^2 = (0, 0, 1, 1, 0)$ виконуються кроки 4 – 13.

Крок 4. Визначаються вихідні сигнали елементів Y-шару:

$$U_{out.Y} = (U_{out.Y1}, \dots, U_{out.Y4}) = (0, 0, 0, 0).$$

Вхідним навчальним вектором $S^2 = (0, 0, 1, 1, 0)$ визначаються вихідні сигнали S-елементів:

$$U_{out.S} = (U_{out.S1}, \dots, U_{out.S4}) = (0, 0, 1, 1, 0).$$

Крок 5. Обчислюється норма вектора вихідних сигналів вхідного шару

$$\|U_{out.S}\| = \sum_{i=1}^5 U_{out.Si} = 2.$$

Крок 6. Визначаються вихідні сигнали елементів інтерфейсного шару: $U_{out.Z} = (0, 0, 1, 1, 0)$.

Крок 7. Обчислюються вхідні сигнали для елементів Y-шару:

$$U_{inp.Y1} = \sum_{i=1}^5 w_{i1} U_{out.Zi} = 0,667 \cdot 0 + 0,667 \cdot 0 + \\ + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 = 0.$$

$$U_{inp.Y2} = \sum_{i=1}^5 w_{i2} U_{out.Zi} = 0,667 \cdot 0 + 0,667 \cdot 0 + 0,667 \cdot 1 + \\ + 0,667 \cdot 1 + 0,667 \cdot 0 = 0,334.$$

$$U_{inp.Y3} = U_{inp.Y4} = U_{inp.Y2} = 0,334.$$

Крок 8. Виконуються кроки 9 – 12.

Крок 9. Оскільки три елементи Y_2, Y_3, Y_4 і мають однакові вхідні, а отже, і вихідні сигнали, то нейроном переможцем вибирається нейрон з меншим номером: $J = 2$.

Крок 10. Розраховуються вихідні сигнали Z-елементів:

$$U_{inp.Z1} = w_{Ji}^2 U_{out.Si}, \quad U_{out.Z} = (0, 0, 1, 1, 0).$$

Крок 11. Обчислюється норма вектора $U_{out.Z}$

$$\|U_{out.Z}\| = \sum_{i=1}^5 U_{out.Zi} = 2.$$

Крок 12. Перевіряється подібність векторів $S^2 = (0, 0, 1, 1, 0)$ і $U_{out.Z}$

$$\frac{\|U_{out.Z}\|}{S^2} = \frac{2}{2} = 1 > p.$$

Крок 13. Визначаються рівноважні ваги w_{i2}^1 , і w_{2i}^2 :

$$w_{i2}^1 = \frac{LU_{out.Zi}}{L - 1 + \|U_{out.Z}\|} = \frac{2U_{out.Zi}}{3}, \quad i = \overline{1, 5},$$

$$w_{12}^1 = w_{22}^1 = w_{52}^1 = 0, \quad w_{32}^1 = w_{42}^1 = 0,667;$$

$$w_{2i}^1 = U_{out.Zi}, \quad i = \overline{1, 5}$$

$$w_{21}^2 = w_{22}^2 = w_{25}^2 = 0, \quad w_{23}^2 = w_{24}^2 = 1.$$

В результаті виходять наступні вагові матриці:

$$W^1 = \begin{vmatrix} 0,667 & 0 & 0,167 & 0,167 \\ 0,667 & 0 & 0,167 & 0,167 \\ 0 & 0,667 & 0,167 & 0,167 \\ 0 & 0,667 & 0,167 & 0,167 \\ 0 & 0 & 0,167 & 0,167 \end{vmatrix},$$

$$W^2 = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Крок 3. Для третього вхідного вектора $S^3 = (1, 0, 0, 0, 0)$ виконуються кроки 4 – 13.

Крок 4. Визначаються вихідні сигнали елементів Y-шару:

$$U_{out.Y} = (0, 0, 0, 0).$$

Визначаються вихідні сигнали S -елементів:

$$U_{out.S} = (1, 0, 0, 0, 0).$$

Крок 5. Обчислюється норма вектора $U_{out.S}$:

$$\|U_{out.S}\| = \sum_{i=1}^5 U_{out.Si} = 1.$$

Крок 6. Визначається вектор вихідних сигналів інтерфейсного шару: $U_{out.Z} = (1, 0, 0, 0, 0).$

Крок 7. Обчислюються вхідні сигнали для елементів Y -шару:

$$U_{inp.Y1} = \sum_{i=1}^5 w_{i1} U_{out.Zi} = 0,667, U_{inp.Y2} = \sum_{i=1}^5 w_{i2} U_{out.Zi} = 0,$$

$$U_{inp.Y3} = U_{inp.Y4} = 0,167.$$

Крок 8. Виконуються кроки 9 – 12.

Крок 9. Визначається нейрон із максимальним вихідним сигналом: $J = 1.$

Крок 10. Розраховуються вихідні сигнали Z -елементів:

$$U_{out.Zi} = U_{out.Si} w_{ji}^2, U_{out.Z} = (1, 0, 0, 0, 0).$$

Крок 11. Обчислюється норма вектора $U_{out.Z}$, $\|U_{out.Z}\| = 1$

Крок 12. Перевіряється подібність векторів $S^3 = (1, 0, 0, 0, 0)$ і $U_{out.Z}$

$$\frac{\|U_{out.Z}\|}{S^3} = 1 > p.$$

Крок 13. Визначаються рівноважні ваги w_{i1}^1, w_{i1}^2 :

$$w_{i1}^1 = \frac{L U_{out.Zi}}{L - 1 + \|U_{out.Z}\|} = U_{out.Zi}, i = \overline{1, 5},$$

$$w_{11}^1 = 1, w_{21}^1 = w_{31}^1 = w_{41}^1 = w_{51}^1 = 0;$$

$$w_{i1}^2 = U_{out.Zi}, w_{i2}^2 = 1, w_{i2}^2 = w_{i4}^2 = w_{i5}^2 = 0.$$

В результаті виходять наступні вагові матриці:

$$W^1 = \begin{vmatrix} 1 & 0 & 0,167 & 0,167 \\ 0 & 0 & 0,167 & 0,167 \\ 0 & 0,667 & 0,167 & 0,167 \\ 0 & 0,667 & 0,167 & 0,167 \\ 0 & 0 & 0,167 & 0,167 \end{vmatrix}, \quad W^2 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}. \quad (7.12)$$

Крок 3. Для четвертого вхідного вектора $S^4 = (0, 0, 0, 1, 1)$ виконуються кроки 4 – 13.

В результаті виконання кроків 4 – 12 обчислюється норма вхідного вектора ($\|S^4\| = 2$), вхідні сигнали елементів Y -шару ($U_{inpY1} = 0, U_{inpY2} = 0,667, U_{inpY3} = U_{inpY4} = 0,334$), визначається нейрон Y_J з максимальним вихідним сигналом ($J = 2$), розраховується вектор вихідних сигналів Z -елементів $U_{out.Z} = (0, 0, 0, 1, 0)$ та його норма $\|U_{out.Z}\| = 1$, а потім перевіряється подібність векторів $U_{out.Z}$ і S^4 :

$$\frac{\|U_{out.Z}\|}{S^4} = 0,5 > p = 0,45.$$

Крок 13. Визначаються рівноважні ваги w_{i2}^1, w_{2i}^2 :

$$w_{12}^1 = w_{22}^1 = w_{32}^1 = w_{52}^1 = 0, \quad w_{42}^1 = 1,$$

$$w_{21}^2 = w_{22}^2 = w_{23}^2 = w_{25}^2 = 0, \quad w_{24}^2 = 1.$$

В результаті виходять наступні вагові матриці:

$$W^1 = \begin{vmatrix} 1 & 0 & 0,167 & 0,167 \\ 0 & 0 & 0,167 & 0,167 \\ 0 & 0 & 0,167 & 0,167 \\ 0 & 1 & 0,167 & 0,167 \\ 0 & 0 & 0,167 & 0,167 \end{vmatrix}, \quad W^2 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Крок 14. Перевірка умов зупинки.

На цьому закінчується перша доба навчання. Легко переконатися, що друга та наступні епохи не можуть внести в отримані матриці при параметрі подібності $p = 0,45$ будь-яких змін. Тому, якщо як умови зупинки використовувати незмінність матриць ваг W^1, W^2 протягом епохи навчання, то алгоритм припинить свою роботу після виконання другої епохи навчання.

Приклад 7.2. Застосування алгоритму АРТ-1 для класифікації векторів прикладу 7.1, що пред'являються в тому самому порядку, але при параметрі подібності $p = 0,8$.

Неважко переконатися, що для векторів $S^1 = (1, 1, 0, 0, 0)$, $S^2 = (0, 0, 1, 1, 0)$, $S^3 = (1, 0, 0, 0, 0)$ навчання виконуватиметься так само, як і в прикладі 7.1, і будуть отримані вагові матриці (7.12) прикладу 7.1. Однак після пред'явлення четвертого вхідного вектора $S^4 = (0, 0, 0, 1, 1)$ виходять результати, відмінні від результатів прикладу 7.1.

Крок 3. Для четвертого вхідного вектора $S^4 = (0, 0, 0, 1, 1)$ виконуються кроки 4 – 13.

Крок 4. Встановлюються вихідні сигнали елементів Y-шару:

$$U_{out.Y} = (0, 0, 0, 0).$$

визначаються вихідні сигнали S-елементів:

$$U_{out.S} = (0, 0, 0, 1, 1).$$

Крок 5. Обчислюється норма вектора $U_{out.S}$:

$$\|U_{out.S}\| = \sum_{i=1}^5 U_{out.Si} = 2.$$

Крок 6. Визначається вектор вихідних сигналів інтерфейсного шару $U_{out.S} = (0, 0, 0, 1, 1)$.

Крок 7. Обчислюються вхідні сигнали для елементів Y-шару:

$$U_{inp.Y1} = \sum_{i=1}^5 w_{i1} U_{out.Zi} = 0, U_{inp.Y2} = \sum_{i=1}^5 w_{i2} U_{out.Zi} = 0,667,$$

$$U_{inp.Y3} = U_{inp.Y4} = 0,334.$$

Крок 8. Виконуються кроки 9 – 12.

Крок 9. Визначається Y-нейрон із максимальним вихідним сигналом: $J = 2$.

Крок 10. Розраховуються вихідні сигнали Z-елементів:

$$U_{out.Zi} = U_{out.Si} w_{2i}^2, U_{out.Z} = (0, 0, 0, 1, 0).$$

Крок 11. Обчислюється норма вектора: $\|U_{out.Z}\| = 1$.

Крок 12. Перевіряється подібність векторів $S^4 = (0, 0, 0, 1, 1)$ та $U_{out.Z}$

$$\frac{\|U_{out.Z}\|}{S^4} = 0,5 < p = 0,8.$$

Оскільки величина відношення норм векторів $U_{out.Z}$ та S^4 менше параметра подібності, то нейрон Y_2 загальмовується та починається пошук нового нейрона з максимальним вихідним сигналом.

Крок 8. Так як не знайдено відповідний Y -нейрон для навчання, то виконуються кроки 9 – 12.

Крок 9. Визначається нейрон із вихідним сигналом, що задовольняє умову: $U_{out.YJ} \geq U_{out.Yj}$

Оскільки $U_{out.Y1} = 0$, $U_{out.Y2} = -1$, $U_{out.Y3} = 0,334$, $U_{out.Y3} = 0,334$, то з двох елементів з однаковим максимальним вихідним сигналом вибирається елемент Y_3 , тому що він має менший індекс, тому $J = 3$.

Крок 10. Розраховуються вихідні сигнали Z -елементів:

$$U_{out.Zi} = U_{out.Si} w_{3i}^2, U_{out.Z} = (0, 0, 0, 1, 1).$$

Крок 11. Розраховується норма вектора $U_{out.Z}$: $\|U_{out.Z}\| = 2$.

Крок 12. Перевіряється подібність векторів $S^4 = (0, 0, 0, 1, 1)$ та $U_{out.Z}$

$$\frac{\|U_{out.Z}\|}{S^4} = 1 > p = 0,8.$$

Оскільки відношення норм векторів $U_{out.Z}$, S^4 перевищує величину параметра подібності, то на наступному кроці визначаються рівноважні ваги.

Крок 13. Визначаються рівноважні ваги w_{i3}^1, w_{3i}^2 :

$$w_{i3}^1 = \frac{LU_{out.Zi}}{L - 1 + \|U_{out.Z}\|} = \frac{2}{3} U_{out.Zi}, i = \overline{1, 5},$$

$$w_{13}^1 = w_{23}^1 = w_{33}^1 = 0, w_{43}^1 = w_{53}^1 = 0,667,$$

$$w_{3i}^2 = U_{out.Zi}, i = \overline{1, 5}$$

$$w_{31}^2 = w_{32}^2 = w_{33}^2 = 0, \quad w_{34}^2 = w_{35}^2 = 1.$$

В результаті виконаних розрахунків виходять такі матриці ваг:

$$W^1 = \begin{vmatrix} 1 & 0 & 0 & 0,167 \\ 0 & 0 & 0 & 0,167 \\ 0 & 0,667 & 0 & 0,167 \\ 0 & 0,667 & 0,667 & 0,167 \\ 0 & 0 & 0,667 & 0,167 \end{vmatrix}, \quad W^2 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Крок 14. Перевірка умов зупинки.

Нехай як умови зупинки використовується незмінність матриць ваг W^1 , W^2 протягом доби навчання. Оскільки ця умова не виконується, то перехід до кроку 2 алгоритму та початок розрахунків другої епохи навчання.

Крок 2. Початок другої доби роботи алгоритму.

Крок 3. Для першого вхідного вектора $S^1 = (1, 1, 0, 0, 0)$ виконуються кроки 4 – 13.

Крок 4. Визначаються вихідні сигнали елементів Y-шару

$$U_{out.Y} = (0, 0, 0, 0).$$

визначаються вихідні сигнали S-елементів:

$$U_{out.S} = (1, 1, 0, 0, 0).$$

Крок 5. Обчислюється норма вектора $U_{out.S}$: $\|U_{out.S}\| = 2$.

Крок 6. Визначається вектор вихідних сигналів інтерфейсного шару:

$$U_{out.Z} = (1, 1, 0, 0, 0).$$

Крок 7. Обчислюються вхідні сигнали для елементів Y-шару:

$$U_{inp.Y_1} = \sum_{i=1}^5 w_{i1} U_{out.Zi} = 1, \quad U_{inp.Y_2} = \sum_{i=1}^5 w_{i2} U_{out.Zi} = 0,$$

$$U_{inp.Y_3} = \sum_{i=1}^5 w_{i3} U_{out.Zi} = 0, \quad U_{inp.Y_4} = \sum_{i=1}^5 w_{i4} U_{out.Zi} = 0,334.$$

Крок 8. Виконуються кроки 9 – 12.

Крок 9. Визначається нейрон із максимальним вихідним сигналом: $J = 1$.

Крок 10. Розраховуються вихідні сигнали Z-елементів:

$$U_{out.Z} = (1, 0, 0, 0, 0).$$

Крок 11. Обчислюється норма вектора $U_{out.Z}$

$$\|U_{out.Z}\| = 1.$$

Крок 12. Перевіряється подібність векторів $S^1 = (1, 1, 0, 0, 0)$ та $U_{out.Z}$:

$$\frac{\|U_{out.Z}\|}{S^1} = 0,5 < p = 0,8.$$

Оскільки відношення норм векторів $U_{out.Z}$ та S^1 менше параметра подібності, то нейрон Y_1 загальмовується і має розпочатися пошук нового нейрона з максимальним вихідним сигналом. Перехід до кроку 8 алгоритму.

Крок 8. Виконуються кроки 9 – 12.

Крок 9. Визначається нейрон із максимальним вихідним сигналом. Оскільки $U_{out.Y1} = -1$, $U_{out.Y2} = U_{out.Y3} = 0$, $U_{out.Y3} = 0,334$, $U_{out.Y4} = 0,334$, то маємо: $J = 4$.

Крок 10. Розраховуються вихідні сигнали Z-елементів

$$U_{out.Zi} = (1, 1, 0, 0, 0).$$

Крок 11. Розраховується норма вектора $U_{out.Z}$

$$\|U_{out.Z}\| = 2.$$

Крок 12. Перевіряється подібність векторів $U_{out.Z}$ та S^1 .
Оскільки

$$\frac{\|U_{out.Z}\|}{S^1} = 1 > p = 0,8.$$

то наступним кроком визначаються рівноважні ваги.

Крок 13. Визначаються рівноважні ваги w_{i4}^1 , w_{4i}^2 :

$$w_{i4}^1 = \frac{LU_{out.Zi}}{L - 1 + \|U_{out.Z}\|} = \frac{2}{3} U_{out.Zi}, \quad i = \overline{1, 5},$$

$$w_{14}^1 = w_{24}^1 = 0,667, \quad w_{34}^1 = w_{44}^1 = w_{54}^1 = 0,$$

$$w_{4i}^2 = U_{out.Zi}, \quad i = \overline{1, 5}$$

$$w_{41}^2 = w_{42}^2 = 1, \quad w_{43}^2 = w_{44}^2 = w_{45}^2 = 0.$$

В результаті розрахунків виходять такі матриці ваг

$$W^1 = \begin{vmatrix} 1 & 0 & 0 & 0,667 \\ 0 & 0 & 0 & 0,667 \\ 0 & 0,667 & 0 & 0 \\ 0 & 0,667 & 0,667 & 0 \\ 0 & 0 & 0,667 & 0 \end{vmatrix} \quad W^2 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{vmatrix}.$$

Таким чином, при великому значенні параметра подібності ($p = 0,8$) чотири вхідні вектори породжують чотири кластери. Цей результат залишається незмінним за будь-якої кількості епох навчання мережі.

7.5. Дискретна нейронна мережа АРТ, що дозволяє вирішувати завдання з кількома рішеннями

Коли на вхід нейронної мережі подаються дані, близькі до даних, на яких відбувалося навчання нейронної мережі, мережа відносить вхідне зображення одного з відомих класів і рішення в цьому випадку єдине. Однак, якщо вхідне зображення не є в певному сенсі близьким до навчальних зображень, то можливо не одне, а кілька рішень. Класичні нейронні мережі АРТ що неспроможні виділяти кілька рішень навіть у тому випадку, коли є кілька рівноцінних рішень. У зв'язку з цим актуальною є розробка нейронних мереж АРТ, що дозволяють вирішувати завдання з кількома рішеннями.

У роботах [63, 64] запропоновано дискретну нейронну мережу АРТ-1s, що дозволяє визначати кілька рішень (якщо вони є). Архітектура цієї мережі наведено на рис. 7.3. У нижній частині малюнка, обмеженою пунктирною лінією, зображено дискретну нейронну мережу АРТ-1, описану на початку цього розділу.

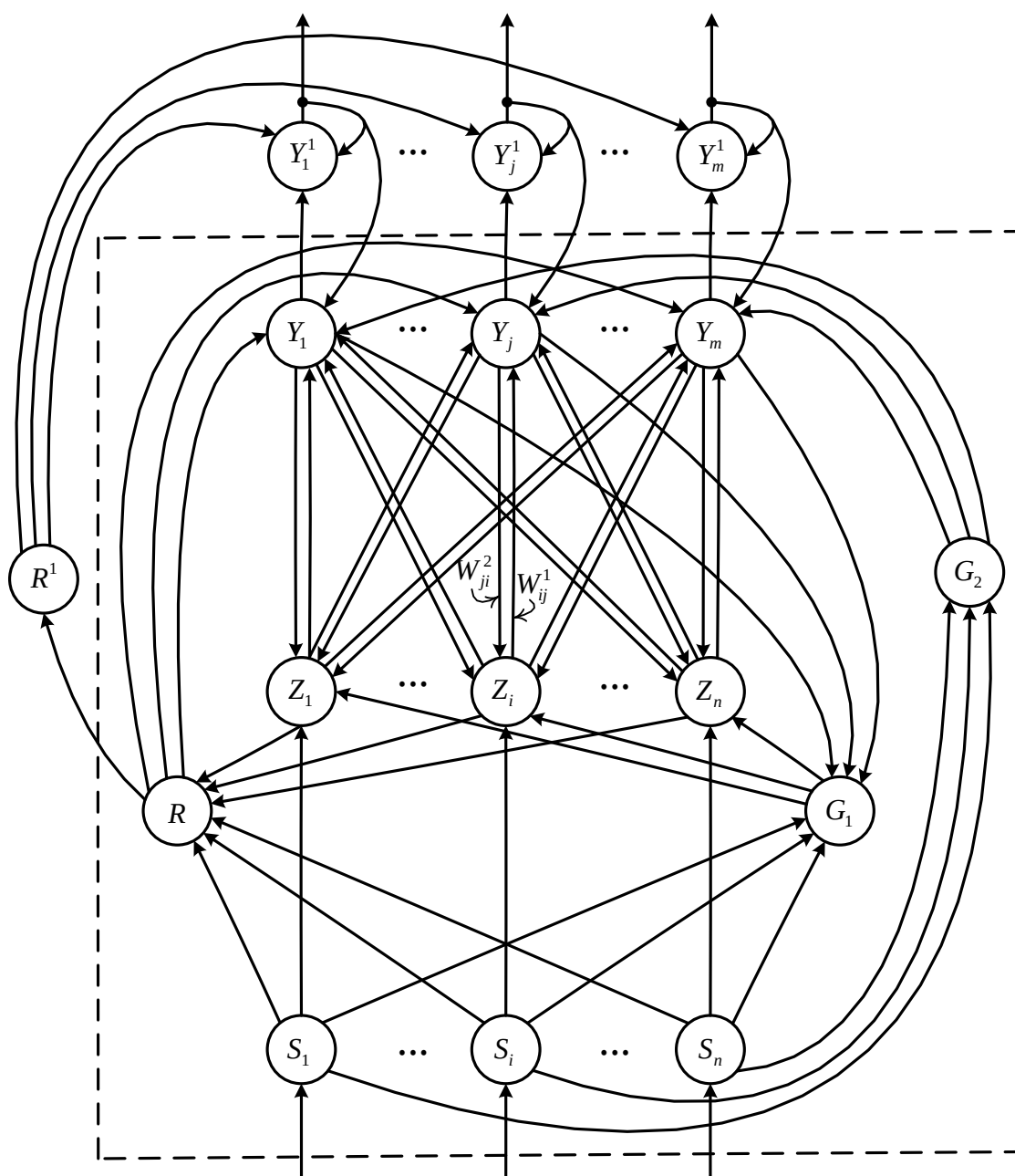


Рис. 7.3. Архітектура дискретної нейронної мережі АРТ-1s
для пошуку кількох рішень

Розглянемо функціонування нейронної мережі АРТ-1 у режимі розпізнавання при $n = 8$, параметрі подібності p , рівному 0,8, і вагах зв'язків між шарами розпізнаючих та інтерфейсних нейронів, отриманих внаслідок запам'ятовування нейронною мережею векторів $S^1 = (0\ 0\ 1\ 1\ 1\ 1\ 0\ 1)$, $S^2 = (0\ 0\ 0\ 1\ 1\ 1\ 1\ 1)$, $S^3 = (1\ 0\ 1\ 1\ 1\ 1\ 0\ 0)$. Ваги зв'язків нейронної мережі мають такі значення:

$$W_{ij}^1 = \frac{1}{n+1} = 0,111, \quad i = \overline{1, 8}, \quad j = \overline{3, m};$$

$$\begin{aligned}
W_{i1}^1 &= 0,333, \quad i = 3, 4, 5, 6, 8; \quad W_{i1}^1 = 0, \quad i = 1, 2, 7; \\
W_{i2}^1 &= 0,333, \quad i = \overline{4, 8}; \quad W_{i2}^1 = 0, \quad i = \overline{1, 3}; \\
W_{i3}^1 &= 0,333, \quad i = 1, 3, 4, 5, 6; \quad W_{i3}^1 = 0, \quad i = 2, 7, 8; \\
W_{i1}^2 &= 1, \quad i = 3, 4, 5, 6, 8; \quad W_{i1}^2 = 0, \quad i = 1, 2, 7; \\
W_{i2}^2 &= 1, \quad i = \overline{4, 8}; \quad W_{i2}^2 = 0, \quad i = \overline{1, 3}; \\
W_{i3}^2 &= 1, \quad i = 1, 3, 4, 5, 6; \quad W_{i3}^2 = 0, \quad i = 2, 7, 8.
\end{aligned}$$

При поданні на вхід дискретної нейронної мережі АРТ-1 вектора $S^4 = (0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0)$ та спрацьовування нейронів S -шару та Z -шару, та нейронів G_1 та G_2 на входах Y -нейронів з'являються сигнали:

$$\begin{aligned}
U_{inp.Y1} &= \sum_{i=1}^8 U_{out.Zi} W_{i1}^1 = 4 \cdot 0,333 = 1,332, \\
U_{inp.Y2} &= \sum_{i=1}^8 U_{out.Zi} W_{i2}^1 = 4 \cdot 0,333 = 1,332, \\
U_{inp.Y3} &= \sum_{i=1}^8 U_{out.Zi} W_{i3}^1 = 4 \cdot 0,333 = 1,332, \\
U_{inp.Y4} &= U_{inp.Y5} = \dots = U_{inp.Ym} = 5 \cdot 0,111 = 0,555. \\
U_{inp.Y_4} &= U_{inp.Y_5} = \dots = U_{inp.Y_m} = 5 \cdot 0,111 = 0,555.
\end{aligned}$$

Оскільки при однакових максимальних вихідних сигналах нейроном-переможцем стає елемент, що розпізнає, з мінімальним індексом, то переможцем буде нейрон Y_1 . Параметр подібності у цьому випадку має таке значення:

$$p = \|U_{out.Z}(S^4)\| / \|S^4\| = 4 / 5 = 0,8.$$

Легко переконається, що такий самий за величиною параметр подібності виходить і при нейронах-переможцях Y_2 та Y_3 . Отже, вхідний вектор S^4 дискретною нейронною мережею АРТ-1 відноситься тільки до одного класу, а два інші можливі рішення губляться. Для усунення цього помітного недоліку нейронної мережі АРТ-1 у її архітектуру вводиться шар бінарних нейронів, що реєструють $Y_1^1, Y_2^1, \dots, Y_m^1$, а також додатковий керуючий нейрон R^1 (см. рис. 7.3). Кожен елемент реєструючого шару нейронів має три входи. Перший вхід пов'язаний односпрямованим зв'язком з одиничним ваговим коефіцієнтом з виходом відповідного елемента шару, що розпізнає. Другий вхід з'єднаний з виходом керуючого нейрона R^1 , який інвертує вихідні сигнали нейрона R . Третій вхід кожного реєструючого нейрона з'єднаний з його виходом.

У режимі навчання нейронної мережі АРТ-1s (рис. 7.3) нейрони реєструючого шару не використовуються. І режим навчання мережі АРТ-1s нічим не відрізняється від режиму швидкого навчання мережі АРТ-1.

Перед початком режиму розпізнавання по ланцюгах, що керують, не відображеним на малюнку, всі нейрони мережі, включаючи і елементи реєструючого шару, переводяться в стан з нульовим вихідним сигналом. В активний стан нейрони реєструючого шару $Y_1^1, Y_2^1, \dots, Y_m^1$, як і нейрони Y - та Z -слоя, перекладаються за правилом "два з трьох", коли поодинокі сигнали одночасно з'являються на виході нейрона-переможця Y_i розпізнаючого шару і на виході керуючого нейрона R^1 . Ці сигнали вказують на те, що вхідний вектор та вектор, що зберігається у вагах зв'язків нейрона-переможця, відповідають один одному за величиною параметра подібності і, отже, знайдено один із класів, до якого належить вхідний вектор. Поодиноким вихідний сигнал нейрона Y_i^1 по ланцюгу зворотного зв'язку з ваговим коефіцієнтом два надходить на його третій вхід і фіксує одиничний сигнал на виході елемента Y_i^1 . Вихідний сигнал нейрона Y_i^1 також загальмовує нейрон-переможець Y_i^1 ($U_{outY_i} = -1$). Потім у шарі нейронів, що розпізнають, починається пошук нового нейрона-переможця. Цей процес пошуку нейронів-переможців та їх загальмовування триває доти, доки не будуть загальмовані всі розподілені розпізнавальні Y -нейрони. При цьому на виходах нейронів реєструючого шару може не виявитись жодного одиничного сигналу, один або кілька одиничних сигналів. У першому випадку вхідний вектор при заданому значенні параметра подібності не відповідає жодному з векторів, що зберігаються у вагах зв'язків розподілених Y -нейронів, що розпізнають. У другому випадку вхідний вектор класифікується як той, що належить одному класу, а в третьому випадку – вхідний вектор може бути віднесений до кількох класів векторів, що зберігаються в пам'яті нейронної мережі.

Алгоритм навчання нейронної мережі АРТ-1s практично не відрізняється від швидкого алгоритму навчання нейронної мережі АРТ-1. Тому тут розглядається лише алгоритм розпізнавання.

Алгоритм функціонування нейронної мережі АРТ-1s у режимі розпізнавання під час пошуку кількох рішень

В алгоритмі прийнято такі позначення:

n – розмірність вхідних векторів (число S та Z -нейронів мережі);

m – число нейронів шарів Y та Y^1 ;

q – кількість розподілених Y -нейронів;

p – параметр подібності між вхідним вектором і векторами, що зберігаються у вагах зв'язків переможців Y -нейронів.

Крок 1. Встановлюються вказані вище параметри, ініціюються ваги зв'язків нейронної мережі, задаються нульові вихідні сигнали всіх нейронів мережі.

Крок 2. З'являється вхідне зображення (вектор) $S^* = (s_1^*, \dots, s_n^*)$ та починається функціонування нейронної мережі в режимі розпізнавання.

Крок 3. Виконуються кроки 4 – 12 для пред'явленого зображення S^* .

Крок 4. Визначаються вхідні та вихідні сигнали елементів S -шару:

$$U_{inp.S_i} = s_i^* \quad U_{out.S_i} = U_{inp.S_i}; \quad i = 1, \dots, n.$$

Для вектору $U_{out S} = (U_{out S_1}, \dots, U_{out S_n})$ розраховується норма

$$\|U_{out S}\| = \sum_{i=1}^n U_{out S_i}.$$

Крок 5. За наступними формулами визначаються вхідні та вихідні сигнали нейронів інтерфейсного та розпізнаючого шарів:

$$U_{inp.Z_i} = U_{out.S_i}, \quad U_{out.Z_i} = U_{inp.Z_i}, \quad i = 1, \dots, n;$$

$$U_{inp.Y_j} = \sum_{i=1}^n w_{ij}^1 U_{out Z_i}, \quad U_{out Y_j} = U_{inp Y_j}, \quad j = 1, \dots, m.$$

Крок 6. Починається цикл пошуку в шарі нейронів, що розпізнають елемента, що зберігає у вагах своїх зв'язків вектор, найбільш близький за параметром подібності вхідному вектору (зображенню) S^* .

Крок 7. У шарі Y -нейронів визначається нейрон-переможець Y_J .

Крок 8. Розраховується вектор вихідних сигналів елементів інтерфейсного шару та його норма:

$$U_{out Z_i} = U_{out S_i} W_{ji}^2, \quad i = 1, \dots, n;$$

$$\|U_{out Z}\| = \sum_{i=1}^n U_{out Z_i}.$$

Крок 9. Розраховується параметр подібності p_1 для векторів $U_{out Z}$ и $U_{out S}$

$$p_1 = \|U_{out Z}\| / \|U_{out S}\|.$$

Якщо $p_1 < p$, то вектори $U_{out S}(S_1^*)$ и $U_{out Z}$ не відповідають один одному за величиною параметра подібності. Отже, нейрон Y_J повинен бути загальмований ($U_{out Y_j} = -1$) вихідним сигналом керуючого нейрона R . Після цього шляхом переходу на крок 5 алгоритму в шарі нейронів, що розпізнають, починається пошук нового елемента-переможця, для якого потім розраховується величина параметра подібності p_1 . Якщо $p_1 > p$, то вектора $U_{out S}(S_1^*)$ та $U_{out Z}$ відповідають один одному за величиною параметра подібності, тобто. вхідний вектор S_1^* відповідає класу векторів, еталонний представник яких зберігається у вагах зв'язків нейрона-переможця Y_J .

Крок 10. Нейрон Y_J^1 реєструючого шару вихідними сигналами нейронів R^1 та Y_J переводиться в стан: $U_{out Y_j^1} = 1$.

Крок 11. Сигнал $U_{out Y_J^1}$ загальмовує нейрон Y_J ($U_{out Y_J} = -1$) та по ланцюгу зворотного зв'язку з виходу елемента Y_J^1 подається на його вхід, фіксуючи активний стан нейрона Y_J^1 ($U_{out Y_J^1} = 1$).

Крок 12. Перевіряється умова припинення роботи алгоритму $U_{out Y_k} = -1, k = 1, \dots, q$.

Якщо умова зупинки роботи алгоритму не виконується, то перехід на п'ятий крок алгоритму та пошук нового нейрона-переможця.

Якщо умова зупинки виконується, то перехід на наступний крок алгоритму.

Крок 13. Зупинка.

В результаті роботи алгоритму можуть бути отримані наступні варіанти розв'язання:

- на виходах елементів Y^1 -шару всі сигнали нульові, в цьому випадку на вхід нейронної мережі пред'явлено вектор, не схожий на жоден з векторів, що зберігаються в пам'яті мережі;

- на виходах елементів Y^1 -шару лише один одиничний сигнал, що вказує на належність вхідного вектора до одного з класів векторів, що зберігаються в пам'яті мережі;

- на виходах елементів Y^1 -шару є кілька одиничних сигналів, що вказують на те, що вхідний вектор належить кільком класам векторів або знаходиться на межі кількох класів.

Таким чином, розроблена дискретна нейронна мережа АРТ, що дозволяє визначати одно-, два або більше рішень (якщо вони існують) у завданнях розпізнавання та класифікації чорно-білих зображень.

7.6. Безперервні нейронні мережі АРТ-2

Основною відмінністю нейронної мережі АРТ-2 [28, 62] є можливість роботи з аналоговими векторами і значне зменшення, порівняно з нейронними мережами АРТ-1, кількості використовуваних нейронів. Порівняно з АРТ-1 в архітектурі мережі зроблено деякі зміни, що дозволяють окремим підсистемам функціонувати асинхронно, що є важливим для апаратних реалізацій. Важливою відмінністю аналогових сигналів від дискретних є важлива можливість аналогових векторів бути як завгодно близькими друг до друга (у той час як простір двійкових векторів дискретно). Це накладає додаткові вимоги на функціонування нейронів шару порівняння, оскільки потрібно більш тонкий і чутливий механізм виділення областей резонансу. Загальним рішенням тут є перехід до багатошарової архітектури з точнішим налаштуванням при переході від шару до шару, що й застосовано в АРТ-2. Функціонування шару розпізнавання принципово не змінюється. Мережі АРТ-2 можуть застосовуватися для розпізнавання зображень, що рухаються. Оскільки нейронні системи АРТ-2 не містять механізму інваріантного розпізнавання (на відміну від неокогнітрона), то в поєднанні з ними можна застосовувати спеціалізовані (часто не нейромереві)

системи інваріантного представлення образів, наприклад, двовимірне перетворення Фур'є або складніші алгоритми.

Нейронним мережам АРТ-2 властиві й недоліки: такі як

- можливість розпізнавання лише одного окремого процесу;
- складніший і повільніший, порівняно з нейронними мережами АРТ 1, алгоритм навчання нейронної мережі;
- відсутність можливості одночасного порівняння вхідного зображення з двома або більшими числами зображень, що зберігаються в пам'яті мережі;
- неможливість точного розпізнавання процесів, близьких за формою, але різних за амплітудою.

Всі перелічені вище особливості функціонування нейронних мереж, їх переваги та недоліки, явно виділяють нейронні мережі АРТ-2 як пріоритетні для застосування їх як БЗ для безперервних об'єктів.

Класична архітектура безперервної нейронної мережі АРТ-2 зображено на рис. 7.4. Вона включає три групи нейронів: поле F_1 вхідних обробних нейронів, що складається із шести типів елементів ($W_i, X_i, V_i, U_i, R_i, Q_i, i = \overline{1, n}$) та нормалізуючих модулів (WN, VN, ZN), поле F_2 розпізнаючих Y -нейронів та групи керуючих нейронів $R_1, \dots, R_n, R$.

Нехай на вхід нейронної мережі (рис. 7.4) у режимі навчання подається вхідне зображення $S = (s_1, \dots, s_n)$. Нейрони $S_1, \dots, S_n$ сприймають сигнали $s_1, \dots, s_n$ вхідного зображення S та передають їх на свої виходи: $U_{out.Si} = U_{inp.Si} = s_i, \dots; U_{out.Sn} = U_{inp.Sn} = s_n$.

Нейрони $W_1, \dots, W_n$ поля F_1 сприймають сигнали $s_1, \dots, s_n$ зображення, що пред'являється, і підсумовують їх з вихідними сигналами $u_1, \dots, u_n$ нейронів $U_1, \dots, U_n$:

$$w_i = s_i + au_i, i = \overline{1, n},$$

де a – ваги зв'язків від нейронів U -шару до нейронів W -шару.

Вихідні сигнали w_i ($i = \overline{1, n}$) нейронів $W_1, \dots, W_n$ надходять на входи елементів $X_1, \dots, X_n$ та модуля WN , що обчислює норму $\|w\| = \sqrt{w_1^2 + \dots + w_n^2}$ вектора сигналів $w = (w_1, \dots, w_n)$. Компоненти вектора w та його норма необхідні обчислення вхідних сигналів елементів X -шару.

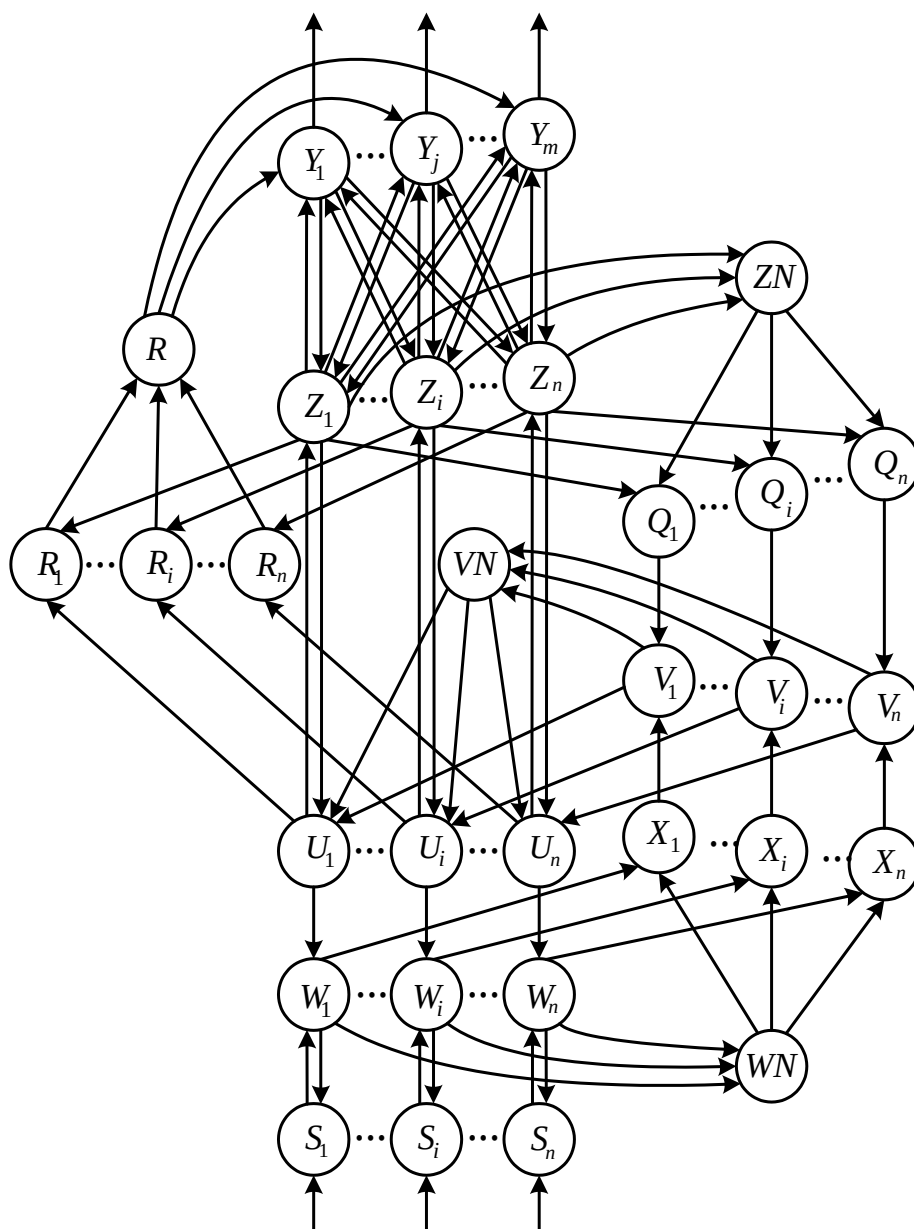


Рис. 7.4. Класична архітектура нейронної мережі АРТ-2

Вхідні сигнали $x_1, \dots, x_n$ нейронів $X_1, \dots, X_n$ визначаються співвідношенням

$$x_i = \frac{w_i}{e + \|w\|},$$

де e – невелика позитивна константа, що запобігає поділу на нуль у випадках, коли $\|w\| = 0$.

Функція активації нейронів $W_1, \dots, W_n, Q_1, \dots, Q_n$ задається наступним співвідношенням:

$$f(u_{inp}) = \begin{cases} U_{inp}, & \text{якщо } U_{inp} \geq \theta, \\ 0, & \text{якщо } U_{inp} < \theta, \end{cases}$$

де θ – параметр, що визначає поріг придушення шумових сигналів. Якщо величина вхідного сигналу U_{inp} менше порогового значення θ , то он рассматривается как шум и подавляется ($f(U_{inp}) = 0$).

Вихідні сигнали $v_1, \dots, v_n$ нейронів $V_1, \dots, V_n$ визначаються виразом

$$v_i = f(x_i) + bf(q_i),$$

де b – константа; q_i ($i = 1, \dots, n$) – вихідні сигнали нейронів $Q_1, \dots, Q_n$.

Нейрони $U_1, \dots, U_n$, використовуючи відповідно вихідні сигнали $v_1, \dots, v_n$ V -нейронів та модуля VN , обчислювального норму $\|v\|$ вектора $v = (v_1, \dots, v_n)$, визначають свої вихідні сигнали щодо співвідношення

$$u_i = \frac{v_i}{e + \|v\|}, \quad i = 1, \dots, n.$$

Якщо нейрон-переможець у Y -шарі ще не визначено, то сигнали u_i однозначно визначають вихідні сигнали $z_1, \dots, z_n$, $q_1, \dots, q_n$ відповідно нейронів $Z_1, \dots, Z_n$ та $Q_1, \dots, Q_n$:

$$z_i = u_i, \quad q_i = \frac{z_i}{e + \|z\|}, \quad i = 1, \dots, n.$$

Стійкий стан обробних нейронів у полі F_1 при вихідних нульових вихідних сигналах нейронів $U_1, \dots, U_n$, $Z_1, \dots, Z_n$, $Q_1, \dots, Q_n$ досягається після двох модифікацій вихідних сигналів нейронів цього поля. Після досягнення рівноваги у полі F_1 Z -нейрони надсилають свої вихідні сигнали на входи Y -нейронів

$$U_{inpY_j} = \sum_{i=1}^n b_{ij} z_i, \quad j = 1, \dots, m,$$

де b_{ij} – вага зв'язку від елемента Z_i ($i = 1, \dots, n$) до елементу Y_j ($j = 1, \dots, m$); початкове значення: $b_{ij} = 1$.

Серед розпізнаючих Y -нейронів визначається нейрон-переможець Y_j , що має найбільший вихідний сигнал. Під час визначення нейрона Y_j вихідні сигнали всіх нейронів поля F_1 залишаються незмінними доти, доки сигнал з виходу нейрона-переможця не надійде на входи Z -нейронів. Після цього за сигналами нейронів Z_i , U_i кожен нейрон R_i ($i = 1, \dots, n$) розраховує свій вихідний сигнал

$$r_i = \frac{u_i + cz_i}{e + \|u\| + c\|z\|},$$

де c – вага зв'язку від нейрона Z_i до нейрона R_i ($i = 1, \dots, n$).

Отримавши вихідні сигнали r_i ($i = \overline{1, n}$) R_i -нейронів, елемент R розраховує норму вектора сигналів $r = (r_1, \dots, r_n)$:

$$\|r\| = \frac{\|u_i + cz_i\|}{\|u\| + c\|z\|}$$

і порівнює її з параметром подібності p між вхідним зображенням і зображенням, що зберігається у вагах зв'язків нейрона, що переміг. Якщо $\|r\| < p$, то нейрон Y_j загальмовується ($U_{out Y_j} = -1$) і не бере надалі участі у змаганнях при пред'явленні поточного зображення. Якщо $\|r\| \geq p$, то відбувається навчання ваг зв'язків переміг Y -нейрона одним з відомих алгоритмів навчання: швидким, повільним або миттєвим.

Нейронна мережа АРТ-2 непогано зарекомендувала себе при розпізнаванні різних зображень. Однак безпосередньо використовувати мережу АРТ-2 для розпізнавання режимів функціонування динамічних об'єктів, коли режими розпізнаються за поведінкою у часі кількох змінних, що належать певним областям своєї зміни, неможливо з кількох причин.

По-перше, нормування компонентів вхідного вектора, що застосовується в нейронній мережі АРТ-2 (зображення) $S = (s_1, \dots, s_n)$ за допомогою співвідношення:

$$s_i^H = s_i / \sqrt{\sum_{i=1}^n s_i^2}$$

дозволяє сприймати будь-які два вектори $S^1 = (s_1^1, \dots, s_n^1)$, $S^2 = (s_1^2, \dots, s_n^2) = (ks_1^1, \dots, ks_n^1)$, де $k \in (0, \infty)$, як однакові. Однак розпізнавання динамічних режимів, де процеси однакової або близької форми суттєво відрізняються за амплітудою, характерне для різних режимів функціонування динамічних об'єктів.

По-друге, у мережі АРТ-2 у вагах зв'язків кожного розподіленого Y -нейрон зберігається тільки одне зображення, отримане в результаті виділення загальних властивостей всіх навчальних зображень. Ознаки, притаманні лише окремим зображенням, не зберігаються в пам'яті мережі. Фактично у вагах зв'язків будь-якого навченого нейрона Y_k зберігається інформація у вигляді перетину нормованих навчальних зображень:

$$S_{Y_k}^H = S_{1,Y_k}^H \cap S_{2,Y_k}^H \cap \dots \cap S_{H_k,Y_k}^H,$$

де $S_{Y_k}^H$ – нормоване зображення, що зберігається у вагах зв'язків розподіленого нейрона Y_k ; S_{i,Y_k}^H ($i=1, \dots, H_k$) – нормовані зображення, що використовуються для навчання нейрона, що розпізнає Y_k ; H_k – кількість навчальних зображень для нейрона Y_k .

При значенні параметра подібності, близького до одиниці, зображення, сприймані людиною як однакові чи дуже близькі, через свої індивідуальні особливості нейронною мережею класифікуються як такі, що належать до різних класів. Тому пам'ять мережі зберігатиме практично кожне зображення у вагах зв'язків окремого нейрона, що розпізнає, тобто. у пам'яті мережі інформація буде зберігатися у вигляді об'єднання нормованих навчальних зображень:

$$S_{1,k}^H \cup S_{2,k}^H \cup \dots \cup S_{m,k}^H,$$

де $S_{j,k}^H$ ($j=1, \dots, m$) – навчальні зображення, що належать до одного k -му образу; m – кількість навчальних зображень.

Архітектура та алгоритми функціонування мережі АРТ-2 не дозволяють на окремих нейронах забезпечити зберігання інформації у вигляді поєднання нормованих навчальних зображень.

По-третє, у нейронній мережі АРТ-2 відсутня можливість одночасного порівняння вхідного зображення з двома або більшим числом зображень, що зберігаються в пам'яті мережі.

7.7. Архітектура та алгоритми функціонування нової безперервної штучної нейронної мережі АРТ-2m

В результаті проведеного аналізу штучних нейромережевих структур, в яких вирішена проблема стабільності-пластичності інформації, що зберігається і запам'ятовується, безперервні нейронні мережі адаптивної резонансної теорії АРТ-2 (рис. 7.4) виділяються для створення БЗ для безперервних об'єктів. Однак мережа АРТ-2 має низку недоліків [62], до яких, в першу чергу, відноситься: неможливість розпізнавання процесів, близьких за формою, але різних за амплітудою. У зв'язку з цим потрібна модифікація мережі АРТ 2 з метою усунення цього обмеження.

Для розпізнавання процесів, близьких за формою, але різних за амплітудою, необхідно змінити нормування компонентів вхідного вектора та вихідних сигналів поля F_1 – поля вхідних обробних нейронів. Пропоноване нормування компонентів вхідного вектора $S = (S_1, \dots, S_n)$, має такий вигляд

$$S_i^H = S_i / S_{i_{\max}}, \quad (7.13)$$

де s_i^H – нормований компонент вхідного вектора; S_i – вихідний компонент вхідного вектора; $S_{i\max}$ – максимально можливе значення i -го компонента для всіх допустимих вхідних векторів нейронної мережі.

Подібне нормування вводиться для інших вихідних векторів поля F_1 , що призводить до зміни архітектури та алгоритму функціонування нейронної мережі.

Нова архітектура безперервної нейронної мережі АРТ-2 m з нормуванням вхідних даних, максимальними значеннями сигналів, зображена на рис. 7.5.

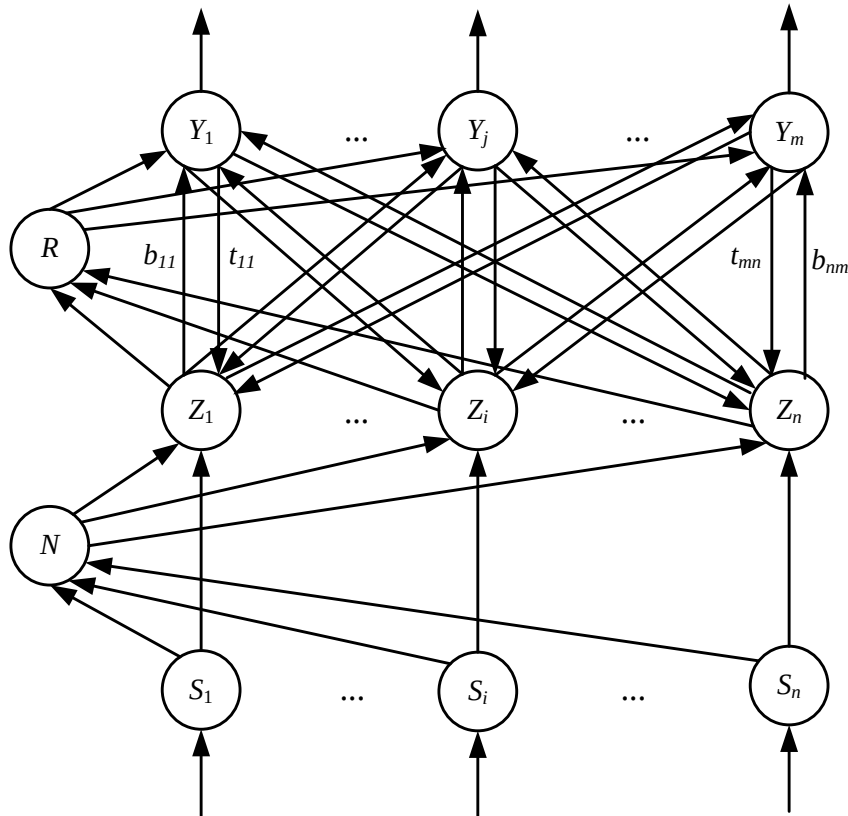


Рис. 7.5. Архітектура нової нейронної мережі АРТ-2 m

Вона включає три шари нейронів: шар вхідних S -нейронів, сигнали яких нормуються за допомогою нормалізуючого модуля N , шар інтерфейсних Z -нейронів, шар розпізнаючих Y -нейронів, а також керуючий нейрон R .

В основу навчання розробленої нейронної мережі покладено алгоритм методу швидкого навчання, що передбачає, що ваги Y -нейрона, що переміг, досягають рівноважних значень при кожному пред'явленні навчального вектора або зображення.

В алгоритмі прийнято такі позначення:

m – максимальна кількість розпізнаючих елементів у Y -шарі або максимальна кількість розпізнаваних образів;

n – число компонентів у вхідному векторі;

S^k – n -мірний вхідний вектор, $k = 1, \dots, q$;

q – число вхідних векторів;

p – параметр подібності, діапазон допустимих значень параметра: $0 < p < 1$;

p_1 – параметр подібності нейрона-переможця Y_J ;

$U_{inp.Y}$ – m -мірний вектор вхідних сигналів шару елементів, що розпізнає;

$U_{inp.Z}$, $U_{out.Z}$ – відповідно вхідний та вихідний n -вимірний вектор сигналів інтерфейсного шару елементів;

b_{ij} – вага зв'язку від елемента Z_i ($i = 1, \dots, n$) до елемента Y_j ($j = 1, \dots, m$); початкове значення: $b_{ij} = 1$;

t_{ji} – вага зв'язку від елемента Y_j ($j = 1, \dots, m$) до елемента Z_i ($i = 1, \dots, n$); початкове значення: $t_{ji} = 1$.

Алгоритм навчання нової безперервної нейронної мережі АРТ-2 m передбачає виконання наступних кроків:

Крок 1. Ініціюється загальний параметр p подібності зображень, максимально можливе значення $S_{i \max}$ компонент для вхідних векторів нейронної мережі та ваги зв'язків b_{ij} та t_{ji} ($i = 1, \dots, n$; $j = 1, \dots, m$).

Крок 2. Для кожного навчального вхідного вектора S^k ($k = 1, \dots, q$) виконуються кроки 3 – 13 алгоритму.

Крок 3. Встановлюються нульові вихідні сигнали всіх нейронів мережі. Вхідним вектором S^k активуються всі S -елементи вхідного шару: $U_{inp.S_i} = s_i^k$, $i = 1, \dots, n$.

Крок 4. Нормуються вихідні сигнали всіх нейронів вхідного шару: $U_{inp.Z_i} = U_{out.S_i} / U_{out.S_{i \max}}$, $i = 1, \dots, n$.

Крок 5. Формуються вихідні сигнали всіх елементів інтерфейсного шару: $U_{out.Z_i} = U_{inp.Z_i}$, $i = 1, \dots, n$.

Крок 6. Для кожного Y -нейрона розраховується його вихідний сигнал: $U_{out.Y_j} = U_{inp.Y_j} = \sum_{i=1}^n b_{ij} U_{out.Z_i}$, $j = 1, \dots, m$.

Крок 7. Поки не знайдено Y -нейрон, ваговий вектор якого відповідно до заданого значення параметра подібності p відповідає вхідному вектору S^k , виконуються кроки 8 – 12 алгоритму.

Крок 8. У Y -шарі визначається нейрон Y_J , що задовольняє умові: $U_{out.Y_j} \geq U_{out.Y_J}$, $j = 1, \dots, m$. Якщо таких елементів кілька, то вибирається елемент із найменшим індексом. Якщо $U_{out.Y_j} = -1$, всі елементи загальмовані, і вхідне зображення не може бути класифіковано або збережено.

Крок 9. Вихідний сигнал нейрона-переможця Y_J задається рівним одиниці: $U_{out.Y_J} = 1$.

Крок 10. Розраховуються вхідні сигнали всіх елементів інтерфейсного шару: $U_{inp.Z_i} = U_{out.Y_J} t_{ji}$, $i = 1, \dots, n$.

Крок 11. Визначається параметр подібності p_1 для нейрона-переможця Y -шару.

Крок 12. Перевіряється умова навчання виділеного нейрона Y_J . Якщо $p_1 < p$, то умова не виконується, нейрон Y_J загальмовується ($U_{out.Y_J} = -1$) та виключається з подальшої участі у змаганнях при пред'явленні даного зображення, потім визначається новий нейрон-переможець (перехід до кроку 8 алгоритму). Якщо $p_1 \geq p$, та умова можливого навчання нейрона Y_J виконується та здійснюється перехід на наступний крок алгоритму.

Крок 13. Визначаються ваги зв'язків елемента Y_J : $b_{iJ} = U_{inp.Z_i}$, $t_{Ji} = U_{inp.Z_i}$, $i = 1, \dots, n$.

Крок 14. Зупинка.

У режимі розпізнавання на відміну режиму навчання алгоритм функціонування безперервної штучної нейронної мережі виробляє лише виділення нейрона-переможця Y_J , при цьому ваги зв'язків нейронної мережі не змінюються.

Таким чином, розроблено архітектуру та алгоритми роботи нової безперервної штучної нейронної мережі АРТ-2 m , яка на відміну від мережі АРТ-2 дозволяє відносити до різних класів процеси, близькі за формою, але різні за амплітудою, що дозволяє на основі штучних нейронних мереж АРТ-2 m розробляти більш універсальні системи розпізнавання.

Контрольні питання

1. Які відмінні риси нейронних мереж адаптивної резонансної теорії?
2. Нарисуйте архітектуру дискретної нейронної мережі АРТ-1.
3. Які шари нейронів складають основу архітектури мережі АРТ-1?
4. Навіщо потрібні нейрони G_1 та G_2 в архітектурі нейронної мережі АРТ-1?
5. Наведіть алгоритм навчання нейронної мережі АРТ-1.
6. Які початкові ваги зв'язків від елементів Z -шару до елементів Y -шару на початку роботи алгоритму навчання нейронної мережі АРТ-1?
7. Як формуються вхідні та вихідні сигнали елементів інтерфейсного шару?
8. Які умови зупинення роботи алгоритму навчання нейронної мережі АРТ-1?
9. Чим визначається максимальна кількість створюваних кластерів під час навчання мережі АРТ-1?
10. Нарисуйте архітектуру дискретної нейронної мережі АРТ, що дозволяє вирішувати завдання розпізнавання з кількома рішеннями.
11. Наведіть алгоритм розпізнавання нейронної мережі АРТ-1 s .
12. Чим відрізняються алгоритми розпізнавання нейронних мереж АРТ-1 та АРТ-1 s ?
13. Наведіть архітектуру нейронної мережі АРТ-2.
14. Які шари нейронів використовуються в нейронній мережі АРТ-2?
15. Наведіть алгоритм навчання нейронної мережі АРТ-2.

16. Як відбувається нормування вхідної інформації в нейронній мережі АРТ-2?
17. Наведіть архітектуру нової нейронної мережі АРТ-2 m .
18. Опишіть алгоритм навчання нейронної мережі АРТ-2 m .

СТАБІЛЬНО-ПЛАСТИЧНІ НЕЙРОННІ МЕРЕЖІ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ ХЕММІНГУ, ХЕББА І ПЕРЦЕПТРОНУ

8.1. Стабільно-пластичні нейронні мережі, що використовують відстань Хеммінга та здатні розпізнавати нову інформацію

Одна з основних переваг нейронних мереж АРТ - можливість донавчатись або можливість запам'ятовувати нову інформацію без спотворення мережі, що вже зберігається в пам'яті. Через цю властивість нейронні мережі АРТ отримали назву стабільно-пластичних нейронних мереж. Ця здатність нейронних мереж АРТ забезпечується наявністю додаткових нейронів, на які записується нова інформація. Якщо розглядати дискретну нейронну мережу АРТ-1 (рис. 7.2), то частина нейронів розпізнає Y -шару зберігає інформацію (еталонні зображення) про відомі класи зображень, а Y -нейрони, що залишилися, можуть використовуватися для запису нової інформації. При цьому кожен Y -нейрон зберігає або зберігатиме інформацію у вагах своїх зв'язків лише про одне зображення. Подібний шар нейронів має мережу Хеммінга (рис. 4.6), де кожен Z -нейрон у вагах своїх зв'язків зберігає інформацію про одне зображення, яке є представником свого класу. Якщо шар Z -нейронів мережі Хеммінга ввести додаткові Z -нейрони (і відповідні їм A - і Y -нейрони), здатні запам'ятовувати нову інформацію (еталонні зображення нових класів), то така мережа також зможе донавчатись і матиме властивості стабільності та пластичності, тобто властивості дискретної нейронної мережі АРТ-1.

Ще одна перевага нейронних мереж АРТ – можливість виділяти нову інформацію. Це досягається тим, що вхідне зображення порівнюється за величиною параметра подібності з еталонними зображеннями, що зберігаються у вагах зв'язків нейронів, що розпізнають. Якщо вхідне зображення відноситься до нового класу, то при порівнянні з будь-яким еталонним зображенням, що зберігається у вагах зв'язків розподілених нейронів, що розпізнають, виходить параметр подібності менше мінімально допустимого. Тому в результаті такого порівняння всі розподілені Y нейрони, що розпізнають, будуть загальмовані, що і є ознакою того, що вхідне зображення несе нову інформацію, якої немає в пам'яті нейронної мережі АРТ.

У нейронній мережі Хеммінга порівняння зображень (або біполярних векторів) виконується за кількістю компонентів, в яких вхідне та еталонне зображення збігаються: $a = n - R_x$, де n – число компонентів зображення; R_x – відстань Хеммінгу між зображеннями. Тому за аналогією з нейронними мережами АРТ можна задати мінімально допустиме значення числа компонентів, що збігаються $a_{\min}$, при яких можуть порівнюватися вхідне та еталонне зображення, а при $a < a_{\min}$ вхідне зображення та еталонне незрівнянні або, іншими словами, вхідне зображення несе нову інформацію порівняно із зображенням, що зберігається у вагах зв'язків нейронної мережі. Врахувати значення $a_{\min}$ в архітектурі нейронної мережі нескладно - достатньо ввести додаткове зміщення

на нейрони Z-шару (точніше - змінити величину зміщення $n/2$). При цьому вихідні сигнали Z-нейронів описуватимуться модифікованим співвідношенням (4.12):

$$U_{out.Z_p} = \frac{n}{2} - a_{\min} + 1 + \sum_{i=1}^n W_{ip} U_{out.S_i} = a - a_{\min} + 1, \quad p = 1, \dots, q, \quad (8.1)$$

де q – число Z-нейронів, що зберігають еталонні зображення.

З (8.1) випливає:

$$U_{out.Z_p} = \begin{cases} 0, & \text{якщо } a < a_{\min}, \\ 1, & \text{якщо } a = a_{\min}, \\ U_{out.Z_p} > 1, & \text{якщо } a > a_{\min}. \end{cases}$$

Таким чином, якщо на виході всіх розподілених Z-нейронів вихідні сигнали дорівнюють нулю ($U_{out.Z_p} = 0, \quad p = 1, \dots, q$), то вхідне зображення за прийнятим критерієм близькості зображень не є схожим на жодне з відомих еталонних зображень і несе нову інформацію. Для апаратного визначення новизни вхідного зображення в архітектуру нейронної мережі Хеммінга (або мережі Хеммінга, що розпізнає зображення на межах кількох класів), достатньо ввести один нейрон Σ , підсумовує вихідні сигнали всіх нейронів Z-шару (рис. 8.1). Якщо після пред'явлення вхідного зображення на виході нейрона Σ нульовий вихідний сигнал, це означає, що вхідне зображення по відстані Хеммінга знаходиться далеко від еталонних зображень, що зберігаються у вагах зв'язків нейронів Z-шару, і є представником нового класу зображень. Якщо після пред'явлення вхідного зображення вихідний сигнал нейрона Σ позитивний, це означає, що вхідне зображення є представником відомого класу зображень, що зберігається в пам'яті нейронної мережі.

8.2. Узагальнення результатів мереж Хеммінгу: стабільно-пластичні нейронні мережі на основі нейронної мережі Хебба

Результати, отримані по мережі Хеммінгу, можуть бути узагальнені інші нейронні мережі. Проілюструємо це з прикладу нейронної мережі Хебба. Архітектура цієї мережі наведено на рис. 8.2 [65]. Основу архітектури складає група з m бінарних або біполярних нейронів $Y_1, \dots, Y_m$. Мережа може запам'ятовувати до 2^m чорно-білих зображень. Однак, застосування цієї мережі для запам'ятовування та розпізнавання 2^m (або близьких до 2^m чисел) різних зображень у багатьох випадках призводить до нерозв'язних проблем адаптації ваги зв'язків нейромережі. Тому часто рекомендують використовувати цю мережу для запам'ятовування та розпізнавання тільки m різних класів зображень, задаючи кожне з них, одиничним вихідним сигналом на виході лише одного нейрона Y-шару (вихідні сигнали інших $(m - 1)$ елементів повинні приймати значення "–1" для біполярних нейронів або "0" для бінарних).

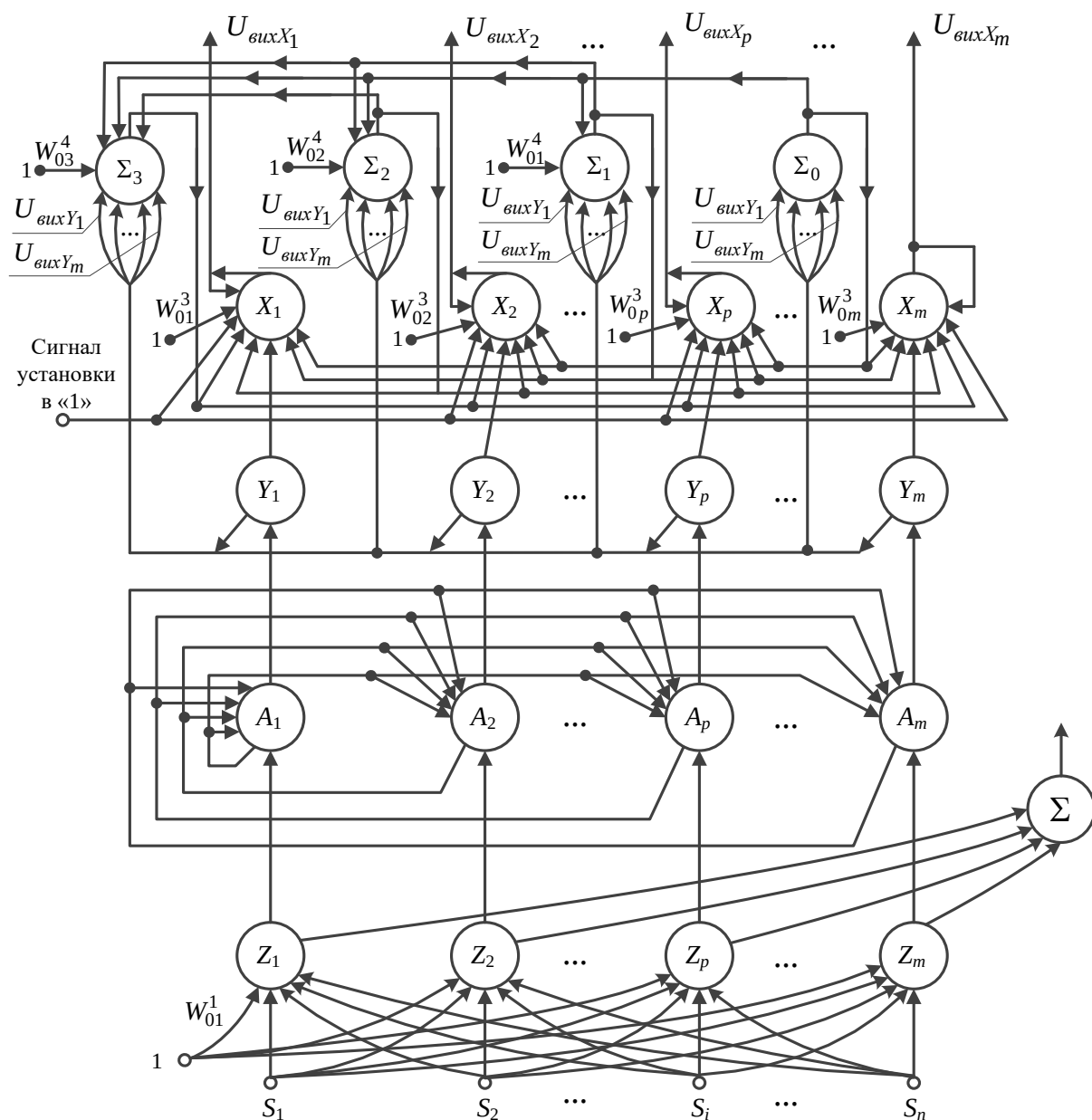


Рис. 8.1. Нейронна мережа розпізнає зображення на мінімальній відстані Хеммінгу від одного, двох або трьох еталонних зображень, зв'язків мережі, що зберігаються у вагах

Нейронна мережа навчається з допомогою алгоритму з урахуванням правила Хебба. Для біполярних нейронів алгоритм навчання має такі основні кроки:

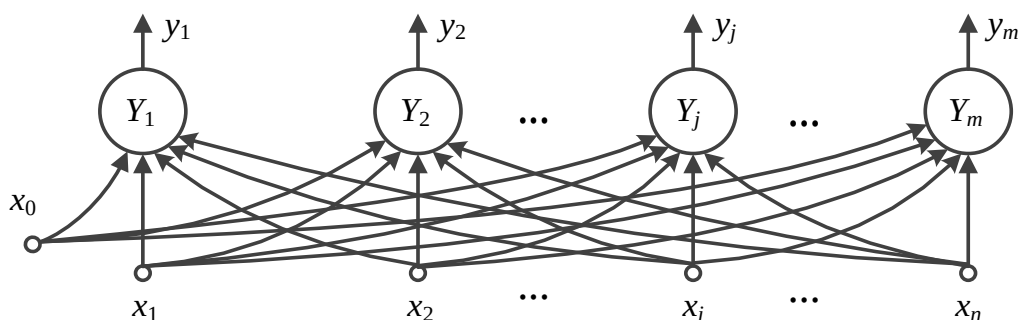


Рис. 8.2. Нейронна мережа Хебба

Крок 1. Задається множина $M = \{(S^1, t^1), \dots, (S^m, t^m)\}$, що складається з m пар (вхідне зображення $S^k = (s_1^k, \dots, s_n^k)$, необхідний вектор вихідних сигналів $t^k = (t_1^k, \dots, t_m^k)$, $k = 1, \dots, m$), задаються нульові вихідні сигнали нейронів Y -шару: $U_{out Y_j} = 0$, $j = \overline{1, m}$. Задається граничне число епох $N_{пр}$ навчання мережі. Ініціюються ваги зв'язків нейронів:

$$W_{ij} = 0, \quad i = 0, 1, \dots, n, \quad j = 1, \dots, m.$$

Крок 2. Починається навчання нейронної мережі правильної реакції кожне вхідне зображення. Послідовно кожна пара (S^k, t^k) , $k = 1, \dots, m$, перевіряється на правильність реакції нейронної мережі на вхідне зображення. Якщо отриманий вихідний вектор мережі $(y_1^k, \dots, y_m^k)$ відрізняється від заданого $t^k = (t_1^k, \dots, t_m^k)$, то виконуються кроки 3 – 5 алгоритму. Якщо вихідний вектор мережі збігається з необхідним, здійснюється перехід до наступної пари. Якщо вихідний вектор збігається із заданим і пара, що перевіряється, є останньою, та перехід на крок 7 алгоритму.

Крок 3. Ініціюється множина вхідних нейронів:

$$x_0 = 1, \quad x_i = s_i^k, \quad i = 1, \dots, n.$$

Крок 4. Ініціюється множина вихідних нейронів мережі:

$$U_{out Y_j} = t_j^k, \quad j = \overline{1, m}.$$

Крок 5. Коригуються ваги зв'язків нейронів мережі за правилом Хебба:

$$W_{ij}(new) = W_{ij}(old) + x_i t_j^k, \quad i = 0, 1, \dots, n, \quad j = 1, \dots, m.$$

Крок 6. Якщо скориговано ваги зв'язків для останньої пари (S^m, t^m) , то перехід крок 7 алгоритму. Інакше – на крок 2 алгоритму.

Крок 7. Перевіряється умова зупинки, тобто правильність функціонування мережі за умови пред'явлення кожного вхідного зображення. Якщо мережа функціонує правильно, перехід на крок 8 алгоритму. Якщо мережа функціонує неправильно, перевіряється досягнення граничного числа епох навчання мережі $N_{\text{пр}}$. Якщо $N_{\text{пр}}$ не досягнуто, по перехід на крок 2 алгоритму, інакше – перехід на наступний крок алгоритму.

Крок 8. Зупинка.

Для забезпечення стабільності в нейронних мережах АРТ при запам'ятовуванні нової інформації використовуються додаткові нейрони, введення яких дозволяє запам'ятовувати нову інформацію, і не руйнувати вже наявну. Покажемо, що цей ефект може бути досягнутий і в нейронній мережі Хебба [66] при введенні додаткових нейронів, що розпізнають. Продемонструємо це на конкретному прикладі.

Приклад 8.1. Нехай задано нейронну мережу Хебба, наведену на рис. 8.3. Потрібно запам'ятати зображення (S^1, S^2, S^3) , наведені на рис. 8.4, за допомогою трьох нейронів Y_1, Y_2, Y_3 . Нумерація елементів зображень відображена на зображенні S^0 (рис. 8.4).

Для навчання нейронної мережі зображеннями S^1, S^2, S^3 необхідно виконати такі кроки:

Крок 1. Задається множина M пар, що складаються з вхідних зображень S^1, S^2, S^3 та необхідних вихідних векторів t^1, t^2, t^3 :

$$M = \{(S^1, t^1), (S^2, t^2), (S^3, t^3)\} = \{((1, -1, 1, 1, 1, 1, -1, -1, 1), (1, -1, -1)), ((1, 1, 1, 1, -1, 1, 1, -1, 1), (-1, 1, -1)), ((1, 1, 1, -1, 1, -1, -1, 1, -1), (-1, -1, 1))\}.$$

Задаються нульові вихідні сигнали всіх Y -нейронів і нульові значення ваги всіх зв'язків, задається початковий момент часу:

$$U_{out.Y_j} = 0, j = 1, \dots, 8;$$

$$W_{ij} = 0, i = 0, 1, \dots, 9, j = 1, \dots, 8.$$

$$t = t_0.$$

Крок 2. Починається навчання нейронної мережі парою (S^1, t^1) .

Крок 3. Ініціюється безліч входів нейронної мережі зображенням S^1 :

$$x_0 = 1, x_1 = 1, x_2 = -1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1, x_7 = -1, x_8 = -1, x_9 = 1.$$

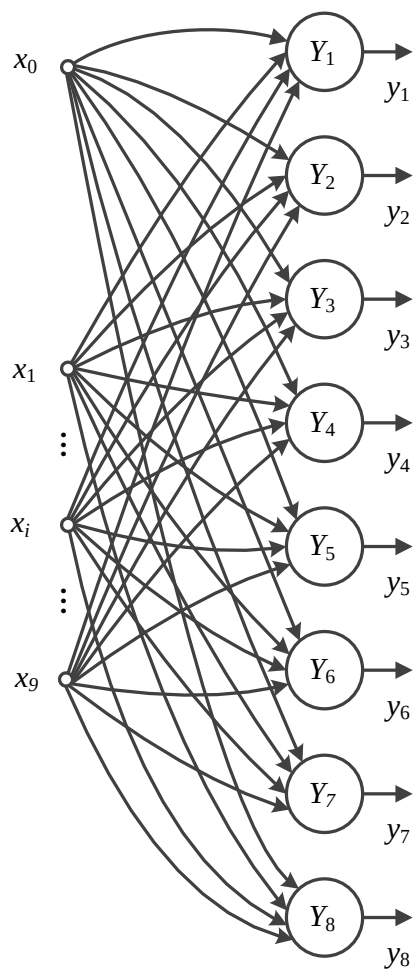


Рис. 8.3. Нейронна мережа Хебба з додатковими нейронами, що запам'ятовують

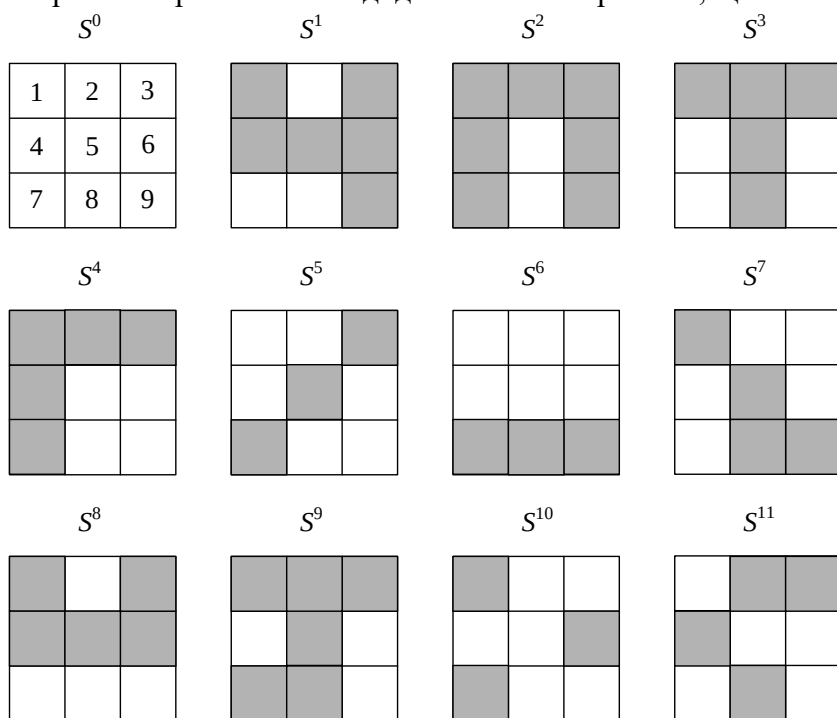


Рис. 8.4. Зображення, що запам'ятовуються нейронною мережею Хебба

Крок 4. Ініціюється множина вихідних нейронів мережі вектором вихідних сигналів t^1 пари (S^1, t^1) :

$$U_{out Y_1} = y_1 = t_1^1 = 1;$$

$$U_{out Y_2} = y_2 = t_2^1 = -1;$$

$$U_{out Y_3} = y_3 = t_3^1 = -1.$$

Крок 5. Визначаються ваги зв'язків нейронів Y_1, Y_2, Y_3 мережі за правилом Хебба:

$$W_{01}(t_1) = W_{01}(t_0) + x_0 y_1 = 0 + 1 \cdot 1 = 1;$$

$$W_{11}(t_1) = W_{11}(t_0) + x_1 y_1 = 0 + 1 \cdot 1 = 1;$$

$$W_{11}(t_1) = W_{31}(t_1) = W_{41}(t_1) = W_{51}(t_1) = W_{61}(t_1) = W_{91}(t_1) = 1;$$

$$W_{21}(t_1) = W_{21}(t_0) + x_2 y_1 = 0 + (-1) \cdot 1 = -1;$$

$$W_{21}(t_1) = W_{71}(t_1) = W_{81}(t_1) = -1;$$

$$W_{02}(t_1) = W_{02}(t_0) + x_0 y_2 = 0 + 1 \cdot (-1) = -1;$$

$$W_{12}(t_1) = W_{12}(t_0) + x_1 y_2 = 0 + 1 \cdot (-1) = -1;$$

$$W_{12}(t_1) = W_{32}(t_1) = W_{42}(t_1) = W_{52}(t_1) = W_{62}(t_1) = W_{92}(t_1) = -1;$$

$$W_{22}(t_1) = W_{22}(t_0) + x_2 y_2 = 0 + (-1) \cdot (-1) = 1;$$

$$W_{22}(t_1) = W_{72}(t_1) = W_{82}(t_1) = 1;$$

$$W_{03}(t_1) = W_{03}(t_0) = -1;$$

$$W_{13}(t_1) = W_{33}(t_1) = W_{43}(t_1) = W_{53}(t_1) = W_{63}(t_1) = W_{93}(t_1) = W_{12}(t_1) = -1;$$

$$W_{23}(t_1) = W_{73}(t_1) = W_{83}(t_1) = W_{22}(t_1) = 1.$$

Потім слід виконувати кроки алгоритму, пов'язані з пред'явленням зображень S^2 і S^3 . У табл. 8.1 наведено результати цих розрахунків.

У табл. 8.2 представлені результати пред'явлення навченої нейронної мережі зображень $S^1, \dots, S^9$. Результати отримані для випадку, коли Y -нейрони мають наступну функцію активації:

$$U_{out Y_j} = \begin{cases} 1, & \text{якщо } U_{inp Y_j} \geq 0, \\ -1, & \text{якщо } U_{inp Y_j} < 0, \end{cases}$$

де $U_{out Y_j}$ та $U_{inp Y_j}$ – відповідно вихідні та вхідні сигнали нейронів Y_j , $j = 1, 2, 3$.

Таблиця 8.1. Результати розрахунків ваги зв'язків мережі Хебба після пред'явлення зображень S^1 , S^2 , S^3

Ваги зв'язків	Початковий момент часу t_0	Момент часу t_1 (після пред'явлення зображення S^1)	Момент часу t_2 (після пред'явлення зображення S^2)	Момент часу t_3 (після пред'явлення зображення S^3)
W_{01}	0	1	0	-1
W_{11}	0	1	0	-1
W_{21}	0	-1	-2	-3
W_{31}	0	1	0	-1
W_{41}	0	1	0	1
W_{51}	0	1	2	1
W_{61}	0	1	0	1
W_{71}	0	-1	-2	-1
W_{81}	0	-1	0	-1
W_{91}	0	1	0	1
W_{02}	0	-1	0	-1
W_{12}	0	-1	0	-1
W_{22}	0	1	2	1
W_{32}	0	-1	0	-1
W_{42}	0	-1	0	1
W_{52}	0	-1	-2	-3
W_{62}	0	-1	0	1
W_{72}	0	1	2	3
W_{82}	0	1	0	-1
W_{92}	0	-1	0	-1
W_{03}	0	-1	-2	-1
W_{13}	0	-1	-2	-1
W_{23}	0	1	0	1
W_{33}	0	-1	-2	-1
W_{43}	0	-1	-2	-3
W_{53}	0	-1	0	1
W_{63}	0	-1	-2	-3
W_{73}	0	1	0	-1
W_{83}	0	1	2	3
W_{93}	0	-1	-2	-3

Таблиця 8.2. Результати розпізнавання зображень $S^1, \dots, S^9$

Нейрони		Вихідні сигнали нейронів при пред'явленні зображень								
		S^1	S^2	S^3	S^4	S^5	S^6	S^7	S^8	S^9
Y_1		1	-1	-1	-1	1	1	1	1	-1
Y_2		-1	1	-1	1	-1	1	-1	-1	-1
Y_3		-1	-1	1	-1	1	1	1	-1	1
Результат розпізнавання		+	+	+	+	-	-	-	+	+

З аналізу таблиці випливає, що вхідні зображення $S^1, S^2, S^3, S^4, S^8, S^9$ викликають реакцію мережі, при якій при пред'явленні будь-якого із зображень на виході нейронної мережі спостерігається один позитивний одиничний сигнал і два негативні одиничні сигнали, тобто мережа відносить вхідне зображення до одного з відомих класів. Аналіз даних табл. 8.2 та зображень показує, що відбулося правильне розпізнавання всіх шести зображень, незважаючи на те, що три з них мають дефекти (S^4, S^8, S^9).

Неважко побачити, що при віднесенні вхідного зображення до одного з трьох класів, представники яких зберігаються у вагах нейронів Y_1, Y_2, Y_3 , сумарний вихідний сигнал цих нейронів дорівнює мінус одиниці: $\sum_{j=1}^3 U_{out.Y_j} = -1$.

Якщо вхідне зображення не відноситься до класів зображень нейронів, що зберігаються у вагах зв'язків Y_1, Y_2, Y_3 , то маємо $\sum_{j=1}^3 U_{out.Y_j} \neq -1$, що може бути ознакою нової інформації (зображень) на вхідному шарі нейронів та необхідності навчання наступної групи Y -нейронів розпізнавати цю вхідну інформацію. При цьому навчені нейрони, що розпізнають, повинні бути заблоковані в режимі розпізнавання і заблоковані або ігноруватися в режимі навчання.

Нейрони Y_4, Y_5, Y_6 нескладно навчити розпізнавання зображень S^5, S^6, S^7 алгоритмом Хебба. Результат навчання наведено у табл. 8.3. При цьому множина M_1 з повчальних пар задавали наступним чином:

$$M_1 = \{(S^5, t^5), (S^6, t^6), (S^7, t^7)\} = \{((-1, -1, 1, -1, 1, -1, 1, -1, -1), (1, -1, -1)), ((-1, -1, -1, -1, -1, -1, 1, 1, 1), (-1, 1, -1)), ((1, -1, -1, -1, 1, -1, -1, 1, 1), (-1, -1, 1))\}.$$

Таблиця 8.3. Результати розрахунків ваг зв'язків нейронів Y_4, Y_5, Y_6 після пред'явлення зображень S^5, S^6, S^7

Ваги зв'язків	Початковий момент часу t_0	Момент часу t_1 (після пред'явлення зображення S^5)	Момент часу t_2 (після пред'явлення зображення S^6)	Момент часу t_3 (після пред'явлення зображення S^7)
W_{04}	0	1	0	-1
W_{14}	0	-1	0	-1
W_{24}	0	-1	0	1
W_{34}	0	1	2	3
W_{44}	0	-1	0	1
W_{54}	0	1	2	1
W_{64}	0	-1	0	1
W_{74}	0	1	0	1
W_{84}	0	-1	-2	-3
W_{94}	0	-1	-2	-3
W_{05}	0	-1	0	-1
W_{15}	0	1	0	-1
W_{25}	0	1	0	1
W_{35}	0	-1	-2	-1
W_{45}	0	1	0	1
W_{55}	0	-1	-2	-3
W_{65}	0	1	0	1
W_{75}	0	-1	0	1
W_{85}	0	1	2	1
W_{95}	0	1	2	1
W_{06}	0	-1	-2	-1
W_{16}	0	1	2	3
W_{26}	0	1	2	1
W_{36}	0	-1	0	-1
W_{46}	0	1	2	1
W_{56}	0	-1	0	1
W_{66}	0	1	2	1
W_{76}	0	-1	-2	-3
W_{86}	0	1	0	1
W_{96}	0	1	0	1

В табл. 8.4 наведено результати розпізнавання зображень S^5, S^6, S^7, S^{10} и S^{11} .

Таблиця 8.4. Результати розпізнавання зображень S^5 , S^6 , S^7 , S^{10} та S^{11}

Нейрони	Вихідні сигнали нейронів при пред'явленні зображень				
	S^5	S^6	S^7	S^{10}	S^{11}
Y_4	1	-1	-1	1	1
Y_5	-1	1	-1	1	1
Y_6	-1	-1	1	-1	-1
Результат розпізнавання	+	+	+	-	-

З аналізу табл. 8.4 випливає, що нейрони Y_4 , Y_5 , Y_6 відносять зображення S^5 , S^6 , S^7 до відомих класів, зразки яких зберігаються у терезах зв'язків цих нейронів. Зображення S^{10} і S^{11} нейронами Y_4 , Y_5 , Y_6 розпізнаються як такі, що не належать до цих класів, це ж спостерігається і при пред'явленні зображень S^{10} , S^{11} та першій групі нейронів Y_1 , Y_2 , Y_3 . Таким чином, для розпізнавання зображень S^{10} , S^{11} необхідно навчити ще два Y -нейрони: Y_7 , Y_8 .

Нейрони Y_7 , Y_8 будуть розпізнавати вказані зображення після того, як при спробі їх розпізнавання будуть заблоковані спочатку нейрони Y_1 , Y_2 , Y_3 , а потім нейрони Y_4 , Y_5 , Y_6 . Результати навчання наведено у табл. 8.5. При цьому множина M_2 з повчальних пар (вхідне зображення – необхідний вихідний вектор) ставилося таким чином:

$$M_2 = \{(S^{10}, t^{10}), (S^{11}, t^{11})\} = \{((1, -1, -1, -1, -1, 1, 1, -1, -1), (1, -1)), ((-1, 1, 1, 1, -1, -1, -1, 1, -1), (-1, 1))\}.$$

Неважко перевірити, що після пред'явлення зображення S^{10} на виходах нейронів Y_7 , Y_8 буде вектор сигналів $(1, -1)$, а після пред'явлення зображення S^{11} – вектор $(-1, 1)$.

Узагальним результати прикладу 8.1 на випадок, коли спочатку нейронна мережа Хебба навчається розпізнаванню множини n_0 ($n_0 \geq 3$) зображень $M_0 = \{S^1, \dots, S^{n_0}\}$, кожне з яких кодується на виходах Y -нейронів $(Y_1^0, \dots, Y_{n_0}^0)$ одним одиничним сигналом та $(n_0 - 1)$ сигналами "-1". Сумарний вихідний сигнал нейронів $Y_1^0, \dots, Y_{n_0}^0$, правильно розпізнаючий одне з зображень, що відносяться до множини M_0 , у цьому випадку дорівнює

$$\sum_{i=1}^{n_0} U_{out Y_i^0} = -(n_0 - 2). \quad (8.2)$$

Таблиця 8.5. Результати розрахунків ваг зв'язків нейронів Y_7, Y_8 після пред'явлення зображень S^{10}, S^{11}

Ваги зв'язків	Початковий момент часу t_0	Момент часу t_1 (після пред'явлення зображення S^{10})	Момент часу t_2 (після пред'явлення зображення S^{11})
W_{07}	0	1	0
W_{17}	0	1	2
W_{27}	0	-1	-2
W_{37}	0	-1	-2
W_{47}	0	-1	-2
W_{57}	0	-1	0
W_{67}	0	1	2
W_{77}	0	1	2
W_{87}	0	-1	-2
W_{97}	0	-1	0
W_{08}	0	-1	0
W_{18}	0	-1	-2
W_{28}	0	1	2
W_{38}	0	1	2
W_{48}	0	1	2
W_{58}	0	1	0
W_{68}	0	-1	-2
W_{78}	0	-1	-2
W_{88}	0	1	2
W_{98}	0	1	0

Якщо при пред'явленні нейронної мережі зображення S^* співвідношення (8.2) не виконується, це ознакою появи на вході нейронної мережі нової інформації (представника нового класу зображень). При накопиченні деякої множини таких зображень $M_1 = \{S^{n_0+1}, S^{n_0+2}, \dots, S^{n_0+n_1}\}$, де $n_1 \geq 3$, можуть бути навчені за допомогою алгоритму Хебба наступні n_1 нейронів мережі. Ознакою того, що деяке вхідне зображення S^* відноситься до множини M_1 є співвідношення:

$$\sum_{i=1}^{n_0} U_{out.Y_i^0}(S^*) > -(n_0 - 2); \quad \sum_{i=1}^{n_1} U_{out.Y_i^1}(S^*) = -(n_1 - 2). \quad (8.3)$$

Поява зображень, для яких не виконуються співвідношення (8.2) та (8.3), вказує на необхідність наступного донавчання нейронної мережі. Число N кроків

донавчання практично не обмежене. Однак при великому N і апаратній реалізації нейронної мережі мережа Хебба з донавчанням буде програвати швидкодії мережі Хебба, навченої на всій кількості зображень, що розпізнаються. В останньому випадку розпізнавання вхідного зображення вимагає однієї спроби (визначаються вхідні та вихідні сигнали нейронів вхідного шару та шару нейронів, що розпізнають). У разі нейронної мережі, яка донавчається N разів, вдала спроба розпізнавання в гіршому випадку може бути здійснена тільки $(N + 1)$ -го разу.

На рис. 8.5 наведено архітектуру нейронної мережі Хебба, яка може донавчатися N разів. Мережа складається з шару вхідних нейронів $x_1, \dots, x_n$ та Y -шару розпізнаючих нейронів. Нейрони Y -шару розбиті на $(N + 1)$ групу, кожна з яких може містити різне число нейронів (але не менше трьох) та формуватися у процесі функціонування мережі.

Перша група нейронів $Y_1^0, \dots, Y_{n_0}^0$ використовується для запам'ятовування вихідної інформації у вигляді безлічі пар зображень M_0 . Друга та наступні групи нейронів $(Y_1^1, \dots, Y_{n_1}^1), \dots, (Y_1^N, \dots, Y_{n_N}^N)$ використовуються для навчання нейронної мережі в міру накопичення нової інформації та послідовного запам'ятовування множин пар зображень $M_1, M_2, \dots, M_N$. У кожній групі нейронів вихідні сигнали Y -нейронів надходять на входи нейрона, що управляє, що має функцію активації виду

$$U_{out.\Sigma_d} = \begin{cases} 0, & \text{якщо } U_{inp.\Sigma_d} = \sum_{k=1}^{n_d} U_{вих.Y_k^d} = -(n_d - 2), \\ 1, & \text{якщо } U_{inp.\Sigma_d} \neq -(n_d - 2), \quad d = 0, 1, \dots, N, \end{cases} \quad (8.4)$$

де $U_{out.\Sigma_d}, U_{inp.\Sigma_d}$ – відповідно вихідний та вхідний сигнал нейрона Σ_d ($d = 0, 1, \dots, N$); $U_{вих.Y_k^d}$ – вихідний сигнал нейрона Y_k^d ($d = 0, 1, \dots, N$; $k = 1, \dots, n_d$); n_d – число нейронів у групі $Y_1^d, \dots, Y_{n_d}^d$.

На початку режиму розпізнавання обнулюються вихідні сигнали Σ -нейронів і блокується робота нейронів $Y_1^1, Y_2^1, \dots, Y_{n_N}^N$.

При подачі на вхід нейронної мережі, яка донавчалася L разів ($1 < L < N$), деякого зображення S^* спочатку активуються нейрони першої групи $Y_1^0, \dots, Y_{n_0}^0$, вихідні сигнали яких надходять на входи керуючого нейрона Σ_0 . Якщо виконується співвідношення

$$U_{inp.\Sigma_0} = \sum_{k=1}^{n_0} U_{out.Y_k^0} = -(n_0 - 2), \quad (8.5)$$

то $U_{out.\Sigma_0} = 0$ та вхідне зображення S^* відноситься до множини пар зображень M_0 , а конкретний клас зображень визначають вихідні сигнали нейронів першої

групи $Y_1^0, \dots, Y_{n_0}^0$, наведені на рис. 8.5.

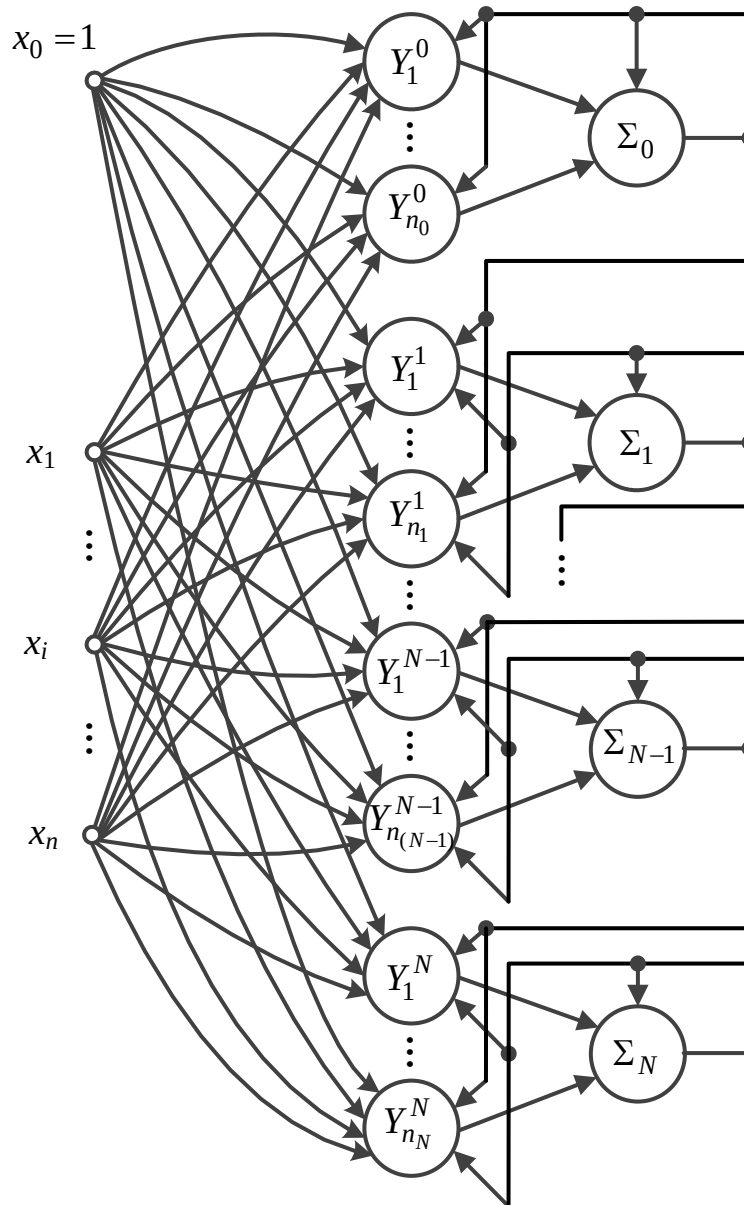


Рис. 8.5. Архітектура нейронної мережі Хебба, яка може донавчатися N разів

Якщо співвідношення (8.5) не виконується, то $U_{out.\Sigma_0} = 1$, що вказує на те, що вхідне зображення S^* не належить множині M_0 . Поодиноким вихідний сигнал нейрона Σ_0 по ланцюгу зворотний зв'язок фіксує одиничний вихідний сигнал нейрона Σ_0 і надходить на входи нейронів першої групи, блокуючи їх. Одночасно цей же одиничний сигнал керуючого нейрона надходить на входи нейронів, що розпізнають $Y_1^1, \dots, Y_{n_1}^1$ другої групи та переводить їх в активний стан. Вихідні сигнали нейронів $Y_1^1, \dots, Y_{n_1}^1$ надходять на входи керуючого нейрона Σ_1 , за

допомогою якого визначається приналежність зображення S^* до множини M_1 . Якщо виконується співвідношення

$$U_{inp.\Sigma_1} = \sum_{k=1}^{n_1} U_{out.Y_k^1} = -(n_1 - 2), \quad (8.6)$$

то зображення S^* належить множині M_1 , а конкретний клас зображень визначають вихідні сигнали нейронів $Y_1^1, \dots, Y_{n_1}^1$. Якщо рівність (8.6) не виконується, зображення S^* не належить множині M_1 та $U_{out.\Sigma_1} = 1$. Сигнал $U_{out.\Sigma_1}$ блокує нейрони $Y_1^1, \dots, Y_{n_1}^1$ та підключає до розпізнавання нейрони третьої групи тощо. В результаті цього процесу вхідне зображення або буде віднесено до одного з відомих множин зображень $M_0, M_1, M_2, \dots, M_L$, або буде сприйнято як нове зображення, яке необхідно запам'ятати на $(L + 1)$ -м донавчання нейронної мережі.

8.3. Стабільно-пластичні нейронні мережі на основі перцептронів

Розроблений підхід для донавчання нейронної мережі Хебба може бути узагальнений і на інші нейронні мережі, зокрема на перцептрони з нейронами, що мають безперервні функції активації, наприклад, біполярні функції сигмоїдної активації:

$$U_{out} = 2/(1 + e^{-\tau U_{inp}}) - 1,$$

де τ – константа. При таких функціях активації отримати на виходах нейронів розпізнає шару вихідні сигнали, точно рівні "1" або "-1", неможливо. Отже, утруднено і безпосереднє використання співвідношень виду (8.2) та (8.3), якщо перцептрон з n_0 вихідними нейронами використовувати як і мережу Хебба для запам'ятовування тільки n_0 різних зображень, задаючи кожне їх одним сигналом, близьким до одиниці, лише з виході одного нейрона, інші вихідні сигнали перцептрона задавати сигналами, близькими до "-1". Однак такі безперервні сигнали можна округлити до цілих значень і використовувати надалі біполярні сигнали.

Архітектуру мережі умовно можна розбити на $(N + 1)$ однотипних блоків (рис. 8.6), кожен із яких зберігає у вагах своїх зв'язків безліч зображень $M_l = \{S^{l1}, S^{l2}, \dots, S^{ln_l}\}$, $l = 0, 1, \dots, N$.

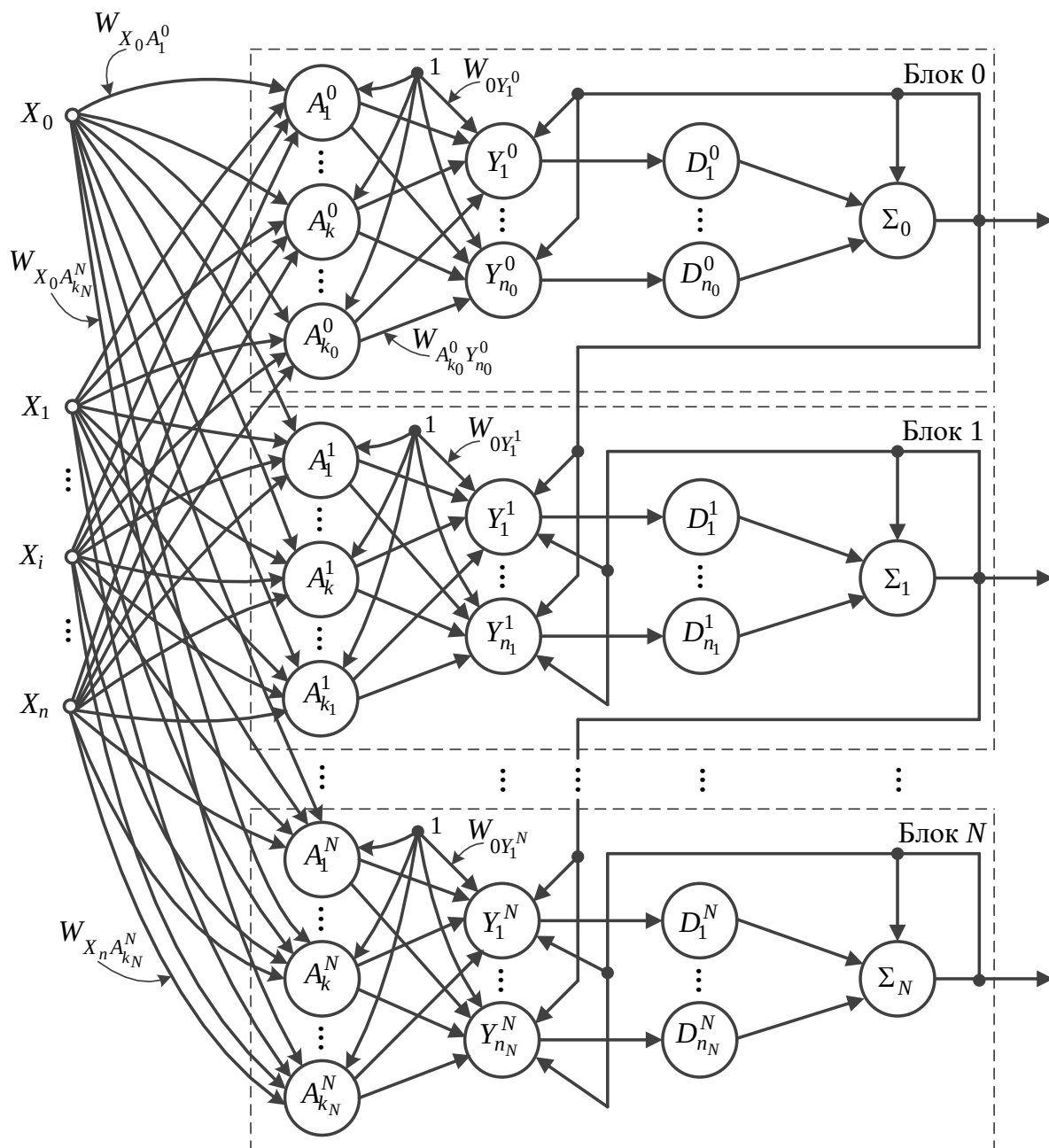


Рис. 8.6. Архітектура тришарового перцептрону, який може донавчатись N разів

Любой l -й ($l = 0, 1, \dots, N$) блок містить k_l A -нейронів прихованого шару: $A_1^l, A_2^l, \dots, A_{k_l}^l$, n_l Y -нейронів розпізнавального або вихідного шару: $Y_1^l, Y_2^l, \dots, Y_{n_l}^l$, і стільки ж D -нейронів ($D_1^l, D_2^l, \dots, D_{n_l}^l$). D -нейрони мають функцію активації виду

$$U_{out.D} = \begin{cases} 1, & \text{якщо } (1-\delta) \leq U_{inp.D} \leq 1, \\ 0, & \text{якщо } (-1+\delta) < U_{inp.D} < 1-\delta, \\ -1, & \text{якщо } -1 < U_{inp.D} \leq -1+\delta, \end{cases} \quad (8.7)$$

де $U_{out.D}$, $U_{inp.D}$ – відповідно вихідний та вхідний сигнал D -нейрону; δ – максимально припустима помилка наближення сигналів "1" та "-1" за допомогою нейронів з біполярною сигмоїдальною функцією активації.

Кожен блок нейронної мережі містить також керуючий Σ -нейрон, що виконує такі ж функції, що й управляючі нейрони в мережі Хебба, здатної донавчати. Входи A -нейронів всіх $(N + 1)$ блоків мережі з'єднані з виходами сенсорних нейронів $X_1, X_2, \dots, X_n$.

Нейрони $A_1^0, \dots, A_{k_0}^0, Y_1^0, \dots, Y_{n_0}^0$ нульового блоку використовуються для запам'ятовування вихідної інформації у вигляді множини зображень M_0 . Блок 1 та наступні блоки нейронної мережі використовуються для навчання перцептрону в міру накопичення нової інформації та послідовного запам'ятовування множин зображень $M_1, M_2, \dots, M_N$. У кожному блоці вихідні сигнали Y -нейронів надходять на входи D -нейронів, що мають функцію активації виду (8.7) і перетворюють безперервні сигнали Y -нейронів на дискретні сигнали: 1, 0 і -1. У кожному блоці нейронів сигнали D -нейронів надходять на входи керуючого нейрона, що має функцію активації виду

$$U_{out.\Sigma_d} = \begin{cases} 0, & \text{якщо } U_{inp.\Sigma_d} = \sum_{k=1}^{n_d} U_{out.D_k^d} = -(n_d - 2), \\ 1, & \text{якщо } U_{inp.\Sigma_d} \neq -(n_d - 2), \quad d = 0, 1, \dots, N, \end{cases} \quad (8.8)$$

де $U_{out.\Sigma_d}$, $U_{inp.\Sigma_d}$ – відповідно вихідний та вхідний сигнал нейрона Σ_d , $d = 0, 1, \dots, N$; $U_{out.D_k^d}$ – вихідний сигнал нейрона D_k^d ($d = 0, 1, \dots, N$; $k = 1, \dots, n_d$); n_d – число Y - і D нейронів у блоці d ($d = 0, 1, \dots, N$).

Якщо $U_{out.\Sigma_d} = 0$, то це є ознакою, що вхідне зображення належить множині M_d , а його конкретний клас визначається комбінацією вихідних сигналів D -нейронів. Якщо $U_{out.\Sigma_d} = 1$, то це означає, що вхідне зображення не належить множині M_d . У цьому випадку вихідний сигнал $U_{out.\Sigma_d}$ по ланцюгу зворотний зв'язок фіксує одиничний вихідний сигнал нейрона Σ_d , блокує нейрони $Y_1^d, \dots, Y_{n_d}^d$ та підключає до розпізнавання нейрони групи $(d + 1)$.

Розглянемо алгоритми функціонування нейронної мережі.

Укрупнений алгоритм навчання перцептрону, здатного донавчатись

Крок 1. Задаються:

- множина M_0 з n_0 навчальних пар (вхідне зображення – заданий вихідний вектор мережі);
- максимальна кількість N можливих донавчень нейронної мережі;
- початкові ваги зв'язків нейронної мережі;
- нульові вихідні сигнали нейронів блоку 0 нейронної мережі;

– початкове значення параметра l , за допомогою якого підраховується кількість донавчання мережі, $l = 0$;

– початкове значення параметра r за допомогою якого підраховується число нерозпізнаних вхідних зображень;

Блокується робота Y -нейронів всіх блоків мережі, крім нульового.

Крок 2. Одним з алгоритмів методу зворотного поширення помилки блок 0 навчається розпізнавання всіх зображень з множини M_0 .

Крок 3. Нейронна мережа переводиться в режим розпізнавання або класифікації вхідної інформації, задається початкове значення параметра $r = 0$. Формується безліч $M_{l+1} = \{S_1^{(l+1)}, S_2^{(l+1)}, \dots, S_{n_{(l+1)}}^{(l+1)}\}$ вхідних зображень, інформація про які немає в пам'яті нейронної мережі та які є представниками нових класів зображень. При записі до множини M_{l+1} нового зображення $S_k^{(l+1)}$ параметр r збільшується на одиницю і перевіряється умова необхідності донавчання нейронної мережі (наприклад, за величиною параметра r і числом донавчень нейронної мережі). Якщо умова не виконується, мережа продовжує функціонувати в заданому режимі, інакше – перехід на наступний крок алгоритму.

Крок 4. Одним із алгоритмів методу зворотного розповсюдження помилки блок $(l+1)$ навчається розпізнавання всіх зображень з множини M_{l+1} . Число l збільшується на одиницю: $l = l+1$. Перевіряється умова закінчення роботи алгоритму, якщо вона не виконується, то перехід на крок 3 алгоритму, інакше – перехід на останній крок алгоритму.

Крок 5. Зупинка.

Можлива архітектура мережі та алгоритм навчання, коли число груп нейронів та число нейронів у кожній групі не задається заздалегідь, а формується в процесі навчання.

Укрупнений алгоритм функціонування перцептрону, здатного донавчати, в режимі розпізнавання

Крок 1. Ініціюються ваги зв'язків та обнулюються вихідні сигнали всіх нейронів нейронної мережі. Задається число L ($1 < L < N$) донавчання нейронної мережі. Задається число l ($l = 0$), за допомогою якого підраховується кількість блоків, які можуть бути використані для розпізнавання вхідного зображення. Блокується робота Y -нейронів всіх блоків, крім нульового.

Крок 2. На вхід мережі подається деяке зображення $S^* = (S_1^*, S_2^*, \dots, S_n^*)$.

Крок 3. Визначаються вхідні та вихідні сигнали нейронів X -шару:

$$U_{inp.X_i} = S_i^*, \quad U_{out.X_i} = U_{inp.X_i}, \quad i = 1, \dots, n.$$

Крок 4. Визначаються вхідні та вихідні сигнали нейронів прихованого шару нульового блоку:

$$U_{inp.A_k^0} = 1 \cdot W_{X_0 A_k^0} + \sum_{i=1}^n U_{out.X_i} W_{X_i A_k^0}, \quad k = 1, \dots, k_0;$$

$$U_{out.A_k^0} = f_{A_k^0}(U_{inp.A_k^0}), \quad k = 1, \dots, k_0,$$

де $U_{inp.A_k^0}, U_{out.A_k^0}, k = 1, \dots, k_0$ - відповідно вхідні та вихідні сигнали нейронів прихованого A^0 -шару; $W_{X_0 A_k^0}, k = 1, \dots, k_0$ - ваги зв'язків для сигналу зміщення на нейрони $A_1^0, \dots, A_{k_0}^0$; $W_{X_i A_k^0}, i = 1, \dots, n; k = 1, \dots, k_0$ - ваги зв'язків від нейронів вхідного шару до нейронів прихованого шару; $f_{A_k^0}$ - функція активації нейронів A^0 -шару.

Крок 5. Визначаються вхідні та вихідні сигнали нейронів нульового блоку, що розпізнають:

$$U_{inp.Y_p^0} = 1 \cdot W_{0Y_p^0} + \sum_{k=1}^{k_0} U_{out.A_k^0} W_{A_k^0 Y_p^0}, \quad p = 1, \dots, n_0;$$

$$U_{out.Y_p^0} = f_{Y_p^0}(U_{inp.Y_p^0}), \quad p = 1, \dots, n_0,$$

де $U_{inp.Y_p^0}, U_{out.Y_p^0}, p = 1, \dots, n_0$ - відповідно вхідні та вихідні сигнали нейронів Y -шару; $W_{0Y_p^0}, p = 1, \dots, n_0$ - ваги зв'язків для сигналу зміщення на нейрони $Y_1^0, \dots, Y_{n_0}^0$; $W_{A_k^0 Y_p^0}, k = 1, \dots, k_0, p = 1, \dots, n_0$ - ваги зв'язків від елементів A -шару до нейронів Y -шару; $f_{Y_p^0}$ - функція активації нейронів Y -шару.

Крок 6. Визначаються вхідні та вихідні сигнали D -нейронів нульового блоку:

$$U_{inp.D_p^0} = U_{out.Y_p^0}, \quad p = 1, \dots, n_0;$$

$$U_{out.D_p^0} = \begin{cases} 1, & \text{якщо } (1-\delta) \leq U_{inp.D_p^0} < 1, \\ 0, & \text{якщо } (-1+\delta) < U_{inp.D_p^0} < 1-\delta, \\ -1, & \text{якщо } -1 < U_{inp.D_p^0} \leq -1+\delta, \end{cases}$$

де $U_{inp.D_p^0}, U_{out.D_p^0}, p = 1, \dots, n_0$ - відповідно вхідні та вихідні сигнали нейронів D -шару нульового блоку; δ - максимально припустима помилка наближення сигналів "1" та "-1" за допомогою нейронів з біполярною функцією активації.

Крок 7. Визначаються вхідні та вихідні сигнали керуючого нейрона Σ_0 :

$$U_{out.\Sigma_0} = \begin{cases} 0, & \text{якщо } U_{inp.\Sigma_0} = \sum_{p=1}^{n_0} U_{out.D_p^0} = -(n-2), \\ 1, & \text{якщо } U_{inp.\Sigma_0} = \sum_{p=1}^{n_0} U_{out.D_p^0} \neq -(n-2). \end{cases}$$

Якщо $U_{out.\Sigma_0} = 0$, то за допомогою вихідних сигналів D^0 -нейронів визначається клас, до якого належить вхідне зображення. Потім здійснюється перехід крок 1 алгоритму. Якщо $U_{out.\Sigma_0} = 1$, то – вхідне зображення не належить множині M_0 . У цьому випадку ланцюгом зворотного зв'язку фіксується одиничний вихідний сигнал нейрона Σ_0 . Цей сигнал блокує нейрони $Y_1^0, \dots, Y_{n_0}^0$ та підключає до розпізнавання нейрони першої групи.

Крок 8. Число l збільшується на одиницю: $l = l + 1$. Визначаються вхідні та вихідні сигнали нейронів шарів A, Y, D блока l :

$$\begin{aligned} U_{inp.A_k^l} &= 1 \cdot W_{X_0 A_k^l} + \sum_{i=1}^{n_l} U_{out.X_i} W_{X_i A_k^l}, \quad k=1, \dots, k_l; \\ U_{out.A_k^l} &= f_{A_k^l}(U_{inp.A_k^l}), \quad k=1, \dots, k_l; \\ U_{inp.Y_p^l} &= 1 \cdot W_{0 Y_p^l} + \sum_{k=1}^{k_l} U_{out.A_k^l} W_{A_k^l Y_p^l}, \quad p=1, \dots, n_l; \\ U_{out.Y_p^l} &= f_{Y_p^l}(U_{inp.Y_p^l}), \quad p=1, \dots, n_l; \\ U_{inp.D_p^l} &= U_{out.Y_p^l}, \quad p=1, \dots, n_l; \\ U_{out.D_p^l} &= \begin{cases} 1, & \text{якщо } (1-\delta) \leq U_{inp.D_p^l} < 1, \\ 0, & \text{якщо } (-1+\delta) < U_{inp.D_p^l} < 1-\delta, \\ -1, & \text{якщо } -1 < U_{inp.D_p^l} \leq -1+\delta, \end{cases} \end{aligned}$$

де $U_{inp.A_k^l}, U_{out.A_k^l}$ ($k=1, \dots, k_l$), $U_{inp.Y_p^l}, U_{out.Y_p^l}, U_{inp.D_p^l}, U_{out.D_p^l}$ ($p=1, \dots, n_l$) – соответственно входные и выходные сигналы нейронів A^l -шару, Y^l -шару і D^l -шару блоку l нейронної мережі; $W_{X_0 A_k^l}$ ($k=1, \dots, k_l$), $W_{0 Y_p^l}$ ($p=1, \dots, n_l$) – ваги зв'язків для сигналів зміщення відповідно на нейрони A^l і Y^l шарів; $W_{X_i A_k^l}, W_{A_k^l Y_p^l}$ ($i=1, \dots, n$; $k=1, \dots, k_l$; $p=1, \dots, n_l$) – відповідно ваги зв'язків від нейронів вхідного шару до нейронів прихованого A^l -шару та від нейронів A^l -шару до нейронів Y^l -шару.

Крок 9. Визначаються вхідний $U_{inp.\Sigma_l}$ та вихідний $U_{out.\Sigma_l}$ сигнали керуючого нейрона Σ_l :

$$U_{inp.\Sigma_l} = \sum_{p=1}^{n_l} U_{out.D_p^l};$$

$$U_{out.\Sigma_l} = \begin{cases} 0, & \text{якщо } U_{inp.\Sigma_l} = -(n_l - 2), \\ 1, & \text{якщо } U_{inp.\Sigma_l} \neq -(n_l - 2). \end{cases}$$

Якщо $U_{out.\Sigma_l} = 0$, то за допомогою вихідних сигналів D^l - нейронів визначається клас, до якого належить вхідне зображення. Потім здійснюється перехід на крок 1 алгоритму.

Якщо $U_{out.\Sigma_l} = 1$ і $l = L$, то – перехід на крок 1 алгоритму, оскільки вхідне зображення не належить множині M_l і в пам'яті мережі немає класу зображень, до якого може належати вхідне зображення. Воно сприймається як представник нового класу зображень, який необхідно запам'ятати $(L + 1)$ -м донавчання нейронної мережі в $(L + 1)$ -м блоці.

Якщо $U_{out.\Sigma_l} = 1$ і $l < L$, то вхідне зображення не належить множині M_l , але може належати множині M_{l+1} . Сигнал із виходу нейрона Σ_l по ланцюгу зворотний зв'язок фіксує свій одиничний вихідний сигнал, блокує роботу нейронів $Y_1^l, \dots, Y_{n_l}^l$ та підключає до розпізнавання нейрони групи $(l + 1)$.

Крок 10. Перевіряються умови припинення роботи алгоритму. Якщо вони виконуються, то перехід на крок 11 алгоритму, інакше – перехід на крок 8 алгоритму.

Крок 11. Зупинка.

Таким чином, розроблено архітектуру та алгоритми функціонування тришарового перцептрону, здатного донавчатись у процесі функціонування нейронної мережі. Отримані результати нескладно узагальнити і на перцептрони, що мають понад три шари нейронів.

Контрольні питання

1. Якими унікальними перевагами володіє нейронна мережа АРТ-1?
2. У чому подібність нейронних мереж Хеммінгу та АРТ-1?
3. Нарисуйте архітектуру стабільно-пластичної мережі, що використовує відстань Хеммінгу.
4. Нарисуйте архітектуру нейронної мережі Хебба.
5. Наведіть алгоритм навчання нейронної мережі Хебба.
6. Запишіть правило Хебба, яке використовується в алгоритмі навчання мережі Хебба.
7. Нарисуйте архітектуру нейронної мережі Хебба, здатної донавчатися N разів.
8. Наведіть алгоритм функціонування нейронної мережі Хебба, здатної донавчати в режимі розпізнавання зображень.
9. Нарисуйте архітектуру стабільно-пластичної нейронної мережі на основі перцептрону.

10. Наведіть алгоритм навчання перцептрону, здатного донавчатись.
11. Наведіть алгоритм функціонування перцептрону, здатного донавчатись в режимі розпізнавання зображень.

ВИСНОВОК

У цьому навчальному посібнику описані основні моделі штучних нейронних мереж, що використовуються для розпізнавання образів, виконання прогнозів, асоціативної пам'яті та управління, розв'язання різноманітних оптимізаційних завдань. Відомі й інші, більш традиційні і, можливо, більш досконалі підходи та методи вирішення перелічених завдань, проте вони не завжди мають необхідну гнучкість при аналізі конкретних практичних проблем з їх різноманітними обмеженнями. Нейронні мережі дають альтернативний і перспективний підхід і багато програм виграють від їх використання. Це є однією з причин стрімкого розвитку як теорії нейронних мереж, так і їх використання у всіх сферах людської діяльності [1 – 5, 10 – 22, 53 – 55]. За прогнозами фахівців спектр завдань, що вирішуються за допомогою нейронних мереж, і далі активно розширюватиметься також тому, що існують завдання підвищеної складності, які часто просто не вирішуються сучасними математичними методами або вимагають швидкодії обчислювальних систем, що наразі недосяжна технологічно. Практично єдиним засобом на вирішення цих завдань є нейромережеві алгоритми обробки інформації, реалізовані нейрокомп'ютерах. Саме завдяки цьому нейрокомп'ютери знайшли широке застосування у військовій галузі. Водночас реальне поширення та практичне застосування нейромережевих технологій в Україні та країнах СНД поки що дуже обмежене. Почасти це можна пояснити такими причинами:

- відсутністю суттєвої вузівської підготовки фахівців у галузі нейронних мереж, а також відсутністю загальнодоступних сучасних підручників та монографій з теорії нейронних мереж та їх практичним додатком;
- використання технологій нейронних мереж має свої особливості, невластиві традиційним інформаційним технологіям;
- перехід від теорії нейронних мереж та модельних прикладів до їх практичного використання потребує високої кваліфікації фахівців та певних тимчасових та фінансових витрат;
- сучасна обчислювальна техніка з послідовною обробкою інформації не найкраще відповідає нейромережевим технологіям.

Однак Україна має необхідний науково-технічний потенціал для того, щоб подолати своє відставання у сфері практичного застосування нейромережевих технологій за лічені роки.

Підтвердженням цього є і дослідження, які проводять співробітники кафедри "Комп'ютерна інженерія та програмування" НТУ "ХПІ", частина з яких опублікована в роботах [56 – 58, 62 – 68]

СПИСОК ЛІТЕРАТУРИ

1. Ямпольський Л.С. Нейротехнології та нейрокомп'ютерні системи: підручник / Л.С. Ямпольський, В.В. Лісовиченко, В.В. Олійник. – Київ: Дорадо-Друк, 2016. – 576 с.
2. Ямпольський Л.С. Нейротехнології та нейросистеми. Монографія/Л.С. Ямпільський. - Київ: Дорадо-Друк, 2015. - 508 с.
3. Proc. IEEE First International conference on neural networks. – San-Diego, Calif., June 21 – 24, 1987. – Vol. I – IV.
4. Greenwood Dan. NASA JSC neural network survey results."1st Annu. Workshop Space Oper. Autom. and Rob. (SOAR'87). Proc. Workshop (Houston, Tex.), Aug. 5 – 7, 1987". – Houston, Tex., 1987.
5. Proc. IEEE Second International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. I – IV.
6. Proc. IEEE International conference on neural networks. – Washington, NJ, – 1989. – Vol. I – II.
7. International Workshop "Neurocomputers and attention", 18 – 22 September 1989 / Extended abstracts. – Moscow, Pushino, 1989.
8. McMahon D.C. A neural network trained to select air craft maneuvers during air combat: A comparison of network and rule based performance / D.C. McMahon // IJCNN – 90, San Diego, Calif. – Vol. 1. – 1990. – P. 107 – 112.
9. Shea P. Application of artificial neural network in the SAIC explosive detection system / P. Shea, B. Yedidia, F. Liu // IJCNN–91, Seattle, Washington, July 8–12, 1991. – Vol. 2. – New York. – 1991. – P. 910.
10. Нейрокомп'ютери та інтелектуальні роботи / Н.М. Амосов, Т.М. Байдик, А.Д. Гольцев та ін. Під ред. Амосова Н.М. - Київ: Наукова думка, 1994. - 272 с.
11. Вороновський Г.К. Генетичні алгоритми, штучні нейронні мережі та проблеми віртуальної реальності / Вороновський Г.К., Махотило К.В., Петрашев С.М., Сергєєв С.А. - Харків: Основа, 1997. - 112 с.
12. Fausett L. Fundamentals of Neural Networks. Architectures, Algorithms, and Applications / L. Fausett – New Jersey: Prentice Hall International, Inc., 2006. – 483 P.
13. Руденко О.Г. Основи теорії штучних нейронних мереж/О.Г. Руденко, Є.В. Бодянський. - Харків: ТЕЛІТЕХ, 2002. - 317 с.
14. Werbos P.J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Masters thesis / P.J. Werbos. – Harward: Harward University, 1974.
15. Parker D.B. Learning logic / D.B. Parker / Invention Report S. 64–81. File 1.–Office of Technology Licensing. – Stanford, CA: Stanford University, 1982. – S. 64–81.
16. Rumelhart D.E. Learning internal representations by error propagation / D.E. Rumelhart, G.E. Hinton, R.J. Williams // Parallel distributed processing. – Vol. 1. – Cambridge, MA: MIT Press, 1986. – P. 318 – 362.
17. Pineda F.J. Generalization of backpropagation to recurrent and higher order networks / F.J. Pineda // Neural information processing systems. – New York: American Institute of Phisycs, 1988. – P. 602 – 611.

18. Werbos P.J. Backpropagation: past and future / P.J. Werbos // IEEE International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. 1. – San-Diego: CA, 1988. – P. 343 – 353.
19. Werbos P.J. Backpropagation and neurocontrol: A review and prospectus / P.J. Werbos // IEEE International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. 1. – San-Diego: CA, 1988. – P. 209 – 216.
20. Sejnowski T.J. Parallel networks that learn to pronounce English text / T.J. Sejnowski, C.R. Rosenberg // Complex Systems. – 1987. – P. 145 – 168.
21. Ivakhnenko F.G. Cybernetics and Forecasting Techniques / A.G. Ivakhnenko, V.G. Lara, R.N. McDonough // American Elsevier, N.Y. – 1967.
22. Івахненко О.Г. Кібернетичні пророчі пристрої / А.Г. Івахненко, В.Г. Лапа. – Київ: Наукова думка, 1965.
23. Ivakhnenko F.G. The Group Method of Data Handling – a Rival of the Method of Stochastic Approximation / A.G. Ivakhnenko // Soviet Automatic Control, 1968. – Vol. 13. – No. 3. – P. 43 – 55.
24. Kohonen T. Self-organized formation of topologically correct feature maps / T. Kohonen // Biol. Cybern. – 1982. – 43. – №1. – P. 59 – 69.
25. Ritter H. Kohonen's self-organizing maps: Exploring their computational capabilities / H. Ritter, K. Shulter // IEEE International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. 1. – San-Diego: CA, 1988. – P. 109 – 116.
26. Luttrell S.P. Self-organizing multilayer topographic mappings / S.P. Luttrell // IEEE International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. 1. – San-Diego: CA, 1988. – P. 93 – 100.
27. Luttrell S.P. Self-organization: A derivation from first principles of a class of learning algorithms / S.P. Luttrell // IEEE International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. 1. – San-Diego: CA, 1988. – P. 495 – 498.
28. Kangas I. Variant of self-organizing maps / I. Kangas, T. Kohonen, J. Laaksonen, O. Simula, O. Venta // IEEE International conference on neural networks. – San-Diego, Calif., July 24 – 27, 1988. – Vol. 1. – San-Diego: CA, 1988. – P. 517 – 522.
29. Kohonen T. Improved Versions of Learning Vector Quantization / T. Kohonen // International. Joint Conference on Neural Networks. – Vol.1. – San-Diego: CA, 1990. – P. 545 – 550.
30. Hopfield J.J. Neural networks and physical systems with emergent collective computation abilities / J.J. Hopfield // Proceedings of the National Academy of Science USA. – 1982. – 79. – P. 2554 – 2558.
31. Kosko B. Bidirectional associative memories / B. Kosko // IEEE Transactions on Systems, Man, and Cybernetics. – 1988. – Vol.18. – P. 19 – 60.
32. Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence / B. Kosko. – New Delhi: Prentice Hall of India, 1996. – 450 p.
33. Патент України на винахід № 108949, МПК G06G 7/60, G06N 3/04, G06F 15/18 Пристрій N-направленої асоціативної пам'яті / Дмитрієнко В.Д.,

Заковоротний О.Ю., Бречко В.О. – Замовник та володар патенту Національний технічний університет "ХПІ". Заявлено 17.03.2014; Опубліковано 25.06.2015. Бюл. № 12.

34. Патент України на винахід № 108947, МПК G06G 7/60, G06F 15/18 Пристрій багатошарової двонаправленої асоціативної пам'яті / Дмитрієнко В.Д., Заковоротний О.Ю., Хавіна І.П., Бречко В.О. – Замовник та власник патенту Національний технічний університет "ХПІ". Заявлено 17.03.2014; Опубліковано 25.06.2015. Бюл. № 12.

35. Патент України на винахід № 108712, МПК G06G 7/60, G06N 3/04, G06F 15/18 Пристрій багатошарової двонаправленої асоціативної пам'яті з керуючими нейронами / Дмитрієнко В.Д., Заковоротний О.Ю., І.П. Хавіна, Бречко В.О. – Замовник та власник патенту Національний технічний університет "ХПІ". Заявлено 22.04.2014; Опубліковано 25.05.2015. Бюл. № 10.

36. Grossberg S. Competitive learning: From interactive activation to adaptive resonance / S. Grossberg // Cognitive Science. – 1987. – Vol. 11. – P. 23 – 63.

37. Дмитрієнко В.Д. Обчислювальний пристрій розпізнавання режимів функціонування динамічних об'єктів / В.Д. Дмитрієнко, В.М. Терьохіна, А.Ю. Заковоротний // Вісник НТУ "ХПІ". Збірник наукових праць. Тематичний випуск: Інформатика та моделювання. - Харків: НТУ "ХПІ". - 2004. - № 34. - С. 70 - 81

38. Dmitrienko V.D. Neural Networks Art: Solving problems with multiple solutions and new teaching algorithm / V.D. Dmitrienko, A.Yu. Zakovorotnyi, S.Yu. Leonov, I.P. Khavina // Open Neurology Journal. – 2014. – Vol. 8. – P. 15 – 21.

39. Дмитрієнко В.Д. Архітектура та алгоритми функціонування нейронних мереж Хеммінга та Хебба, здатних донавчатися та розпізнавати нову інформацію / В.Д. Дмитрієнко, А.Ю. Заковоротний // Радіoeлектроніка, інформатика, керування. - Запоріжжя: ЗНТУ, 2014. - № 2. - С. 100 - 109.

40. Дмитрієнко В.Д. Двонаправлена багатошарова нейронна мережа, що зберігає ланцюжки асоціативних зображень / В.Д. Дмитрієнко, А.Ю. Заковоротний // Вісник Ташкентського державного технічного університету. - Ташкент: ТДТУ, 2015. - № 3. - С. 3 - 10.

41. Дмитрієнко В.Д. Дискретна нейронна мережа АРТ із використанням відстані Хеммінга / В.Д. Дмитрієнко, А.Ю. Заковоротний, Н.В. Мезенців, Г.В. Гейко // Вісник НТУ "ХПІ". – Харків: НТУ "ХПІ", 2016. – Вип. 21. - С. 29 - 40.

Зміст

Вступ	3
Глава 1. Основи штучних нейронних мереж та нервових клітин	8
1.1. Короткі відомості про мозок людини	8
1.2. Формальні нейрони штучних нейронних мереж.	10
1.3. Найпростіші нейронні мережі. Правило Хебба.	13
1.4. Адалін. Навчання нейронних мереж за допомогою дельта-правила	20
Контрольні питання.	29
Глава 2. Перцептрони	30
2.1. Перцептрони – клас моделей мозку	31
2.2. Навчання перцептронів за допомогою α - та γ -систем підкріплення	34
2.3. Теореми Розенблатта про елементарні перцептрони.	43
2.4. Тришарові перцептрони з кількома R -елементами та змінними вагами зв'язків між S - та A -нейронами	48
2.5. Загальні висновки про можливості тришарових перцептронів з послідовними зв'язками.	49
2.6. Багатошарові перцептрони.	54
Контрольні питання	59
Глава 3. Метод зворотного розповсюдження помилки.	60
3.1. Функції активації, що використовуються в методі зворотного поширення	60
3.2. Основний алгоритм методу зворотного розповсюдження помилки.	63
3.3. Проблеми модифікації та узагальнення основного алгоритму методу зворотного розповсюдження	67
3.4. Глибокі нейронні мережі	71
Контрольні питання.	75
Глава 4. Нейронні мережі, що базуються на змаганні	76
4.1. Махпет	76
4.2. Нейронна мережа Mexican Hat	78
4.3. Мережа Хеммінгу.	83
4.4. Нейронна мережа Хеммінгу, здатна визначати кілька рішень.	90
4.5. Нейронна мережа Кохонена.	95
4.6. Навчальне векторне розбиття.	102
Контрольні питання.	110
Глава 5. Мережі Хопфілда.	111
5.1. Атрактори в нейронних мережах	111
5.2. Дискретна мережа Хопфілда	117

5.3. Непрерывные сети Хопфилда	124
Контрольні питання.	128
Глава 6. Двонаправлена асоціативна пам'ять	130
6.1. Двонаправлена асоціативна пам'ять на двійкових елементах	130
6.2. Оцінка ємності двонаправленої асоціативної пам'яті ...	137
6.3. Безперервна ДАП	142
6.4. Гетероасоціативна пам'ять.	143
6.5. Автоасоціативна мережа.	147
6.6. Архітектура та алгоритми функціонування N -спрямованої асоціативної пам'яті	149
6.7. Багатошарова двоспрямована асоціативна пам'ять	153
Контрольні питання.	156
Глава 7. Нейронні мережі на основі теорії адаптивного резонансу. .	157
7.1. Вступ до теорії адаптивного резонансу.	157
7.2. Базова архітектура мереж АРТ	158
7.3. Архітектура нейронних мереж АРТ-1	160
7.4. Алгоритм навчання мереж АРТ-1	162
7.5. Дискретна нейронна мережа АРТ-1, що дозволяє вирішувати завдання з кількома рішеннями	177
7.6. Безперервні нейронні мережі АРТ-2	182
7.7. Архітектура та алгоритми функціонування нової безперервної штучної нейронної мережі АРТ-2 m	187
Контрольні питання.	190
Глава 8. Стабільно-пластичні мережі на основі нейронних мереж Хеммінгу, Хеббу та перцептронів	192
8.1. Стабільно-пластичні нейронні мережі, що використовують відстань Хеммінга та здатні розпізнавати нову інформацію	192
8.2. Узагальнення результатів мереж Хеммінгу: стабільно- пластичні нейронні мережі на основі нейронної мережі Хебба.	193
8.3. Стабільно-пластичні нейронні мережі на основі перцептронів	206
Контрольні питання.	212
Висновок	214
Список літератури	215

Навчальне видання

Дмитрієнко Валерій Дмитрович
Леонов Сергій Юрійович
Заковоротний Олександр Юрійович

**НЕЙРОННІ МЕРЕЖІ:
ВІД НАЙПРОСТІШИХ МОДЕЛЕЙ БІОЛОГІЧНИХ СИСТЕМ ДО
СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ**

Навчальний посібник

для студентів та аспірантів комп'ютерних спеціальностей
вищих технічних навчальних закладів

Роботу до видання рекомендував *М. Й. Заполовський*
Відповідальний за випуск *С. Ю. Гавриленко*
В авторській редакції

План 2025 р., п. 13

Підп. до друку 13.02.2025. Формат $60 \times 84 \frac{1}{16}$. Папір Сору Paper.
Гарнітура Times New Roman. Обл.-вид. арк. 15,8.
Електронне видання.

НТУ "ХПІ", 61002, Харків, вул. Кирпичова, 2
Видавничий центр НТУ "ХПІ"
Свідоцтво ДК № 116 від 10.07.2000 р.
