О.Г. Руденко, Є.В. Бодянський

Штучні нейронні мережі

УДК 519.71
ББК 004.338.8
Р 83

Рекомендовано Міністерством освіти і науки України як навчальний посібник для студентів вищих навчальних закладів, які навчаються за спеціальностями «Інтелектуальні системи прийняття рішень», «Комп'ютерні системи і мережі» (лист № 14/18.2-2756 від 06.12.2005 р.)

Видано за рахунок державних коштів. Продаж заборонено

Рецензенти:

В. Д. Дмитрієнко, доктор техн. наук, професор
(Національний технічний університет «ХПІ»);

О. Ю. Соколов, доктор техн. наук, професор
(Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ»);

Е. Г. Петров, доктор техн. наук, професор
(Харківський національний університет радіоелектроніки)

Руденко О. Г., Бодянський Є. В.

Р 83 Штучні нейронні мережі: Навчальний посібник. — Харків: ТОВ «Компанія СМІТ», 2006. — 404 с.

ISBN 966-8530-73-X

У навчальному посібнику викладено основні принципи побудови штучних нейронних мереж (ШНМ) як самостійного напрямку в теорії інтелектуальних систем, подано біологічний аналог штучного нейрона і процеси обробки інформації в біологічних системах. Наведено різноманітні моделі штучних нейронів і розглянуто властивості мереж, побудованих на їх основі, починаючи від персептрона і закінчуючи новітніми розробками в цій галузі. Значну увагу приділено методам навчання ШНМ, питанням раціонального вибору і спрощення їх архітектури. Окремий розділ посібника присвячений розгляду прикладного аспекту використання нейромережевих технологій (фінансове прогнозування, адаптивне управління складними об'єктами за умов невизначеності, обробка відео- та мовних сигналів, фільтрація і стиснення інформації тощо).

Для студентів, аспірантів і науково-технічних співробітників, які займаються створенням сучасних способів обробки інформації.

УДК 519.71
ББК 004.338.8

ISBN 966-8530-73-X

ПЕРЕДМОВА

Виникнення штучних нейронних мереж (ШНМ) пов'язане з розумінням того, що мозок живого організму працює інакше, ніж комп'ютер. Мозок людини — це дуже складна нелінійна паралельна інформаційно-керуюча система, здатна до мислення, нагромадження й відновлення інформації, вирішення проблем. Ця система складається з досить однотипних будівельних блоків — нервових клітин, або нейронів, що являють собою прості елементи обробки сигналів, які отримують і комбінують інформацію від інших нейронів через входи — дендрити.

З інженерної точки зору ШНМ — це паралельно розподілена система обробки інформації, утворена тісно зв'язаними простими обчислювальними вузлами (однотипними або різними), що має властивість накопичувати експериментальні знання, узагальнювати їх і робити доступними для користувача у формі, зручній для інтерпретації й прийняття рішень.

На сьогодні є всі підстави говорити про досягнення певних успіхів нейромережевих технологій у вирішенні складних завдань як суто наукових, так й у сфері техніки, бізнесу, фінансів, медичної діагностики й інших галузей, пов'язаних з інтелектуальною діяльністю.

В основу книги покладено матеріали з курсів, прочитаних авторами протягом декількох років у Харківському національному університеті радіоелектроніки.

Ця книга написана в першу чергу для студентів, що цікавляться ШНМ і мають намір використати їх у різних застовуваннях. Зазначимо, що нейромережі, однак, не є панацеєю, і перед тим як використовувати їх для вирішення практичних завдань, необхідно ще раз оцінити їхні переваги й недоліки. Теорія штучних нейронних мереж інтенсивно розвивається, й ми, природно, не можемо висвітлити всі її аспекти. У посібнику розглянуто основні архітектури мереж і парадигми навчання. Відсутність у ньому такого важливого класу ШНМ, як нейро-фаззі-мережі пояснюється тим, що для його опису необхідно ознайомитися з апаратом нечіткої логіки, що зазвичай не читається у вищих навчальних закладах. Поглибленню знань у галузі ШНМ має сприяти звертання до наявної у посібнику бібліографії.

Автори висловлюють подяку О. Бессонову за проведене моделювання ШНМ і кропітку роботу з набору й оформлення рукопису.

Вважаємо своїм обов'язком виразити вдячність Deutscher Akademischer Austauschdienst (DAAD) і Deutsche Forschungsgemeinschaft (DFG) за здійснену ними підтримку, що дала нам можливість відвідати ряд університетів ФРН, ознайомитися із сучасними досягненнями в галузі нейронних мереж й обмінятися думками із зарубіжними колегами.

Автори

ВСТУП

Уже ні в кого не викликає здивування проникнення комп'ютерів практично в усі сфери людської діяльності. Удосконалювання елементної бази, що визначає архітектуру комп'ютера, і розпаралелювання обчислень дозволяють швидко й ефективно вирішувати задачі все зростаючої складності. Вирішення багатьох проблем немислиме без застосування комп'ютерів. Однак, маючи величезну швидкодію, комп'ютер часто не в змозі впоратися з поставленим перед ним завданням так, як це робить людина. Прикладами подібних завдань є розпізнавання (наприклад, знайоме обличчя людина впізнає за 100–120 мс, а найсучасніший комп'ютер — за хвилини або години), розуміння мови й тексту, написаного від руки тощо. Таким чином, мережа нейронів, що створює мозок людини, будучи, як і комп'ютерна мережа, системою паралельної обробки інформації, у багатьох випадках є більш ефективною. Ідея переходу від обробки закладеним у комп'ютер алгоритмом деяких формалізованих знань до реалізації в ньому властивих людині прийомів обробки інформації (розумової діяльності) призвели до появи штучних нейронних мереж (ШНМ).

Характерною рисою біологічних систем є адаптація, завдяки якій такі системи в процесі навчання розвиваються й здобувають нові властивості. Як і біологічні нейронні мережі, ШНМ складаються із з'єднаних між собою елементів, штучних нейронів, функціональні можливості яких тією чи іншою мірою відповідають елементарним функціям біологічного нейрона. Як і біологічний прототип, ШНМ має такі властивості:

- адаптивне навчання: здатність поліпшувати свої характеристики, закладені у тому або іншому алгоритмі настроювання параметрів мережі, що відпрацьовує подані їй послідовності, що навчають, або використовує набутий досвід;
- самоорганізація: ШНМ здатні змінювати свою структуру (архітектуру) або форму подання інформації;

- узагальнення: після завершення процесу навчання мережа може бути нечутливою до незначних змін вхідних сигналів, що дозволяє застосовувати її при зашумлених або не повністю заданих даних;
- обчислення в реальному часі: нейромережеві обчислення можуть здійснюватися паралельно в часі, що істотно збільшує швидкодію ШНМ;
- стійкість до перебоїв: часткове руйнування мережі призводить до втрати якості, однак деякі її властивості зберігаються навіть у випадку руйнування більшої частини мережі.

Незважаючи на таку подібність, ШНМ ще дуже далекі від дублювання властивостей мозку людини. Однак дивна подібність функціонування деяких ШНМ з їхніми біологічними прототипами наводить на думку про можливості проникнення в людський інтелект уже в близькому майбутньому.

Новітня історія ШНМ бере свій початок з 1943 р. — з моменту виходу праці *У. Маккаллоха* та *У. Піттса* [1], у якій досліджено властивості найпростішої моделі нейрона із двома стійкими станами (модель Маккаллоха — Піттса), на вхід якої надходить прискорювальний і гальмуючий сигнали. Вони довели, що за допомогою такої моделі можуть бути реалізовані різні логічні функції, наприклад, І, АБО, НЕ та ін.

У 1949 р. *Д. Гебб* [2] не тільки встановив, що пам'ять у біологічних системах викликається процесами, що змінюють зв'язки між нейронами, але й запропонував правило зміни цього зв'язку — правило навчання нейрона, назване його ім'ям. І сьогодні правило Гебба в тій або іншій формі присутнє в багатьох алгоритмах навчання ШНМ.

Підбадьорюючими були й результати першого комп'ютерного моделювання ШНМ, проведеного в 1956 р. під керівництвом *Н. Рочестера* [3]. Цей рік вважається роком народження наукового напрямку, що має назву «штучний інтелект». В основу модельованої ШНМ покладено вдосконалену модель Гебба. Дана робота дала поштовх численним роботам з моделювання нейронних мереж.

Найбільш важливі результати для розвитку ШНМ були отримано в цей період у Массачусетському технологічному інституті групою вчених під керівництвом *Ф. Розенблатта* [4]. Ними не тільки досліджено різноманітні варіанти мереж, названі персептронами, та вивчено різні способи їх навчання, але й здійснено технічну реалізацію персептрона. Синтезований ними персептрон

використав граничну логіку, складався із трьох шарів і був здатен вирішувати прості завдання розпізнавання цифр. Більш докладно властивості персептрона ми розглянемо нижче. Зазначимо тільки, що робота *Ф. Розенблатта* [5], у якій наведено різні варіанти персептрона й доведено теорему збіжності для персептрона, викликала величезний інтерес до ШНМ.

Водночас під керівництвом Б. Уідроу розвивався напрямок, пов'язаний із практичним застосуванням ШНМ, побудованих на елементах ADALINE (ADAptive LInear NEuron), аналогічних персептрону. Мережа, що містить багато таких елементів, була названа MADALINE (Multiple ADALINE's). Для навчання такої мережі було використано алгоритм, що мінімізував квадратичну помилку навчання — алгоритм *Уідроу — Гоффа* [6, 7]. Слід зазначити, що згодом Б. Уідроу заснував фірму, що займається розробкою електронних компонентів для нейрокомп'ютерів.

Райдужним надіям щодо систем штучного інтелекту, компонентами яких були ШНМ, поклала кінець монографія *М. Мінські* й *С. Пейперта* [8], присвячена серйозному аналітичному дослідженню властивостей персептрона, що з'явилася у 1969 р. Автори показали, що можливості персептрона (мається на увазі одношаровий) обмежуються реалізацією тільки найпростіших логічних функцій, і то не всіх. Наприклад, з його допомогою неможливо організувати функцію «що виключає АБО» тощо. Хоча дослідники й розуміли, що можна будувати персептрони, що містять більше одного шару й мають, вочевидь, ширші можливості, зовсім незрозуміло було, як можна навчити приховані шари. Тому монографія М. Мінські й С. Пейперта викликала справжній шок серед учених і стала настільки сильним ударом по дослідженнях у цій галузі, що такі роботи практично припинилися.

Увага дослідників переключилася на інші види мереж. У 80-ті роки XX ст. досягнуто значних успіхів у такій галузі ШНМ, як асоціативна пам'ять. Піонерськими тут є роботи *Т. Когонена* [9–14] і *Дж. Андерсона* [15, 16]. Як модель Т. Когонен використав так званий одношаровий лінійний асоціатор, а навчання здійснював за правилом Гебба, при якому деякий пропонований мережі образ асоціювався з іншими. Дж. Андерсон, вирішуючи завдання побудови асоціативної пам'яті, що здійснює пошук інформації, яка зберігається, не за її адресою, а за змістом, також користувався правилом навчання Гебба для побудови матриць, що служать для подання збережених у пам'яті й виклику необхідних образів.

Важливою подією в теорії й практиці ШНМ стала поява алгоритму навчання багатошарових мереж, який отримав назву ***алго-***

ритму зворотного поширення помилки. Вперше запропонований у 1974 р. *П. Вербосом* у його дисертації [17] метод виявився непоміченим. Непоміченою виявилася й праця *Д. Паркера* [18], що вийшла в 1985 р. Лише після появи в 1986 р. праці *Д. Румельхарта, Г. Гінтона* й *Р. Уільямса* [19] цей алгоритм отримав широке розповсюдження. Алгоритм забезпечує мінімізацію вихідної помилки мережі шляхом настроювання вагових коефіцієнтів усіх шарів ШНМ, починаючи з останнього, переходячи до передостаннього й т. д., аж до вхідного шару. Таким чином, під час його роботи помилка немовби поширюється від виходу мережі на її вхід, чим і пояснюється назва алгоритму. Розробка цього алгоритму дозволила не тільки перебороти обмеженість одношарового персептрона й конструювати багатошарові мережі, що мають значно більші можливості, але й дала новий імпульс розвитку ШНМ. Поступово з'явився необхідний для конструювання потужних багатошарових мереж теоретичний фундамент. Оцінка М. Мінські виявилася надто песимістичною, і багато хто з поставлених у його книзі завдань вирішуються в цей час мережами за допомогою стандартних процедур.

Якщо перші структури ШНМ були простими й використовували лінійні статичні моделі нейронів, то з часом сконструйовано мережі, зокрема динамічні, що мають більш складні структури й здатні вирішувати значно складніші завдання. Особливий інтерес викликали праці *С. Ґроссберґа* [20–22]. Так, займаючись дослідженням динамічних властивостей мереж, він разом з М. Коеном довів фундаментальну теорему про глобальну збіжність динамічних мереж — теорему Коена — Ґроссберґа. Дослідження біологічних об'єктів дозволили Ґроссберґу і його колегам установити дилему стабільності — пластичності (запам'ятовування нової інформації без руйнування тієї, що вже зберігається у пам'яті) і сконструювати мережу, що володіє такими властивостями. Ця мережа отримала назву ART (*Adaptive Resonance Theory*). Робота [22] поклала початок цілій низці публікацій С. Ґроссберґа і його колег, присвячених розробці й дослідженню різних видів ART: ART-1, ART-2, ART-3, ART MAP, Fuzzy-ART.

Величезну роль у розвитку ШНМ відіграли й праці фізика *Дж. Гопфілда* [23–25], у яких на основі функцій Ляпунова досліджувалася проблема стійкості симетричних рекурсивних мереж. У цих працях мережа вперше характеризувалася деякою енергетичною функцією, мінімуми якої відповідають збереженим образам. Сконструйована Дж. Гопфілдом мережа, що носить його ім'я,

здатна вирішувати дуже складні оптимізаційні завдання, наприклад відома задача про комівояжера, що й було продемонстровано в його спільній з *Д. Танком* роботі [25]. Слід зазначити, що підкреслена практична спрямованість робіт Дж. Гопфілда послужила основою того, що вже в 1987 р. під його керівництвом був створений ***нейрочип***.

Перша переконлива перевага використання ШНМ під час вирішення складних практичних завдань продемонстрована роботою *Т. Сейновського* й *Ч. Розенберґа* [26] на прикладі автоматичного навчання англійській мові.

Сьогодні існує вже достатньо типів ШНМ, які дозволяють успішно вирішувати завдання розпізнавання образів і мови, класифікації, побудови математичних моделей і керування, оптимізації й прогнозування. З деякими з основних типів ШНМ ми й ознайомимося далі.

Галузі застосування ШНМ

ШНМ знаходять сьогодні широке застосування у будь-яких галузях, починаючи від завдань апроксимації функцій і закінчуючи створенням нейрокомп'ютерів. Важко описати всі можливі сфери застосування нейронних мереж, тому ми коротко зупинимося лише на деяких з них.

Апроксимація функцій

Установлення універсальних апроксимуючих властивостей ШНМ стало досить важливим етапом у становленні загальної теорії й стимулювало дослідження в даній галузі. Багатьма дослідниками було встановлено, що нейронна мережа з одним прихованим й одним вихідним шаром здатна апроксимувати з будь-якою наперед заданою точністю на компактній множині будь-яку неперервну функцію. Так, *Р. Гехт-Нільсен* [27, 28], спираючись на теорему Колмогорова — Арнольда [29, 30] про подання неперервних функцій декількох змінних у вигляді суперпозицій неперервних функцій одного змінного, довів можливість апроксимації функцій багатьох змінних досить загального виду за допомогою двошарової нейронної мережі із прямими повними зв'язками з фіксованою кількістю нейронів з заздалегідь відомими обмеженими функціями активації. *Дж. Цибенко* й *К. Хорник* [31–33], використовуючи теорему Хана-Банаха, довели, що кінцева лінійна комбінація фіксованих одновимірних функцій може однозначно апроксимувати будь-яку неперервну функцію n дійсних змінних на заданому

гіперкубі при досить м'яких припущеннях щодо функцій одного змінного.

Асоціативна пам'ять

Існують архітектури ШНМ, які запам'ятовують образи, що надходять на них, з чимось їх асоціюючи, а при пред'явленні деякого «асоціативного» образу витягують їх з пам'яті. Ця властивість дозволяє організовувати пошук інформації не за адресою, а за її змістом. Навіть якщо пропонована асоціація, що відновлюється, буде спотворена перешкодами, мережа може видати правильний результат. Залежно від того, чи збігається шуканий образ зі збереженим у пам'яті, чи ні, розрізнюють авто- і гетероасоціативну пам'ять.

Дослідженню асоціативної пам'яті присвячена монографія [10].

Стиснення даних

Деякі типи ШНМ мають властивості, що дозволяють використовувати ці мережі для стиснення даних, наприклад перед їхньою передачею, зменшуючи тим самим кількість переданих бітів інформації. Подібні завдання виникають і в кластерному аналізі, коли різні, схожі за певними ознаками образи об'єднуються в деякі групи або кластери, тобто здійснюється перехід від вихідного n-вимірного простору образів до m-вимірного простору кластерів, де $m < n$. Подальша робота в просторі меншої розмірності призводить до економії обчислювальних ресурсів і зменшення обсягу необхідної пам'яті.

Застосування ШНМ для стиснення даних у системах розпізнавання мови й зображень забезпечує стиснення даних у 100 і більше разів [34].

Розпізнавання та класифікація

Розпізнавання образів (зображень, зокрема текстів, друкованих і рукописних, звуку, мови тощо) є тією галуззю, де найбільш яскраво виявляються переваги ШНМ. Вирішення багатьох задач розпізнавання образів ускладнено внаслідок їхньої високої розмірності. Як ми вже зазначали раніше, використання ШНМ шляхом стиснення даних дозволяє знизити розмірність задачі, зберігаючи властивості подільності розподілів, що відповідають різним класам.

Багато важливих застосувань теорії розпізнавання образів відносяться до задач класифікації кривих і геометричних фігур. Такими є, наприклад, завдання діагностики, виявлення несправностей тощо. І в цьому випадку ШНМ дають ефективне вирішення

завдання незалежно від того, існує навчальна множина вже класифікованих об'єктів чи не існує. Однак наявність такої інформації прискорює процес пошуку вирішення [35, 36].

Оптимізаційні задачі

Більшість практично важливих задач можуть бути сформульовані як оптимізаційні, що доставляють екстремум деякому заздалегідь обраному критерію. Тут, однак, йтиметься тільки про одну таку задачу, задачу про комівояжера, що має величезне практичне значення й вирішення якої вимагає значних обчислювальних витрат. Ця задача полягає в тому, що комівояжер має відвідати задану кількість міст, вибравши для цього найкоротший маршрут, причому в кожному місті він повинен побувати не більше одного разу. Така задача виникає, наприклад, під час створення автоматичних технологічних ліній (свердлення отворів у друкованих платах, трасування друкованих плат та ін.), визначення оптимальних маршрутів перевезення тощо.

Доведено, що ця задача відноситься до класу «NP-повних» (недетерміністськи поліноміальних), кращим методом вирішення яких є повний перебір можливих варіантів. Оскільки у випадку n міст існує $n!$ варіантів обходу, очевидно, що із збільшенням кількості міст складність вирішення завдання різко зростає. Одне з можливих ефективних вирішень цієї задачі за допомогою ШНМ, що вимагає значно менших обчислювальних витрат і при цьому збігається з розв'язком, отриманим повним перебором, було запропоновано в роботі [25].

Керування складними процесами

Проблема синтезу ефективної системи керування є досить складною, оскільки реальні процеси характеризуються, як правило, нелінійними залежностями, високим рівнем шумів та їх корельованістю, які змінюються умовами функціонування, що обумовлюють зміну характеристик досліджуваних об'єктів тощо. Необхідність вирішення задач керування в реальному часі висуває певні вимоги як до самих алгоритмів керування, що входять до складу математичного забезпечення проектованої системи, так і до технічних засобів, що їх реалізують. У цих умовах найбільш ефективними виявляються методи й алгоритми, що базуються на теорії адаптації. Однак, як і при будь-якому підході, ці методи вимагають розробки математичних моделей досліджуваних об'єктів.

Слід зазначити, що отримані математичні моделі використовуються не тільки з метою безпосередньо керування, але й для

упередження поведінки об'єкта, що дозволяє підвищити ефективність керування шляхом завчасної корекції керованих параметрів. Якщо одержання математичної моделі ускладнено або вимагає істотних зусиль, доцільне застосування ШНМ. Використанню ШНМ у системах керування присвячені, наприклад, монографії [37–42].

Прогнозування

Будь-яке прогнозування (екстраполяція) спирається на формалізоване уявлення про існуючий зв'язок між причинами й наслідком. Багато процесів формуються під впливом великої кількості факторів, що діють у різних напрямках і нерідко невідомих. Статистичний аналіз цих процесів містить дослідження взаємозв'язків факторів як у статичному стані, так і в часі. Інформацією для вивчення взаємозв'язків служать часові ряди показників, що характеризують розвиток об'єктів. Найпоширенішим підходом до вирішення задачі прогнозування є екстраполяція чинних у цей час зв'язків і закономірностей на майбутнє. Побудовані відповідно до цього принципу моделі прогнозування відрізняються одна від іншої лише гіпотезами про конкретні види збережених зв'язків. Чим більш загальні припущення закладені у форму моделі й чим більший клас процесів можна описати з її допомогою, тим ширше її можливості під час дослідження окремої реалізації.

Таким чином, вибір і основа математичної моделі є центральним моментом прогнозування. На практиці ж нерідко виявляється, що внаслідок тих чи інших причин отримати математичну модель, яка адекватно відбиває властивості досліджуваного об'єкта, надзвичайно складно. І в цих випадках ефективним виявляється використання ШНМ.

У роботах [37, 43–46] наведено численні приклади успішного застосування ШНМ для вирішення завдань економічного, зокрема, фінансового прогнозування.

Нейрокомп'ютери

Нейрокомп'ютери є обчислювачами нового класу, що забезпечують більш швидке, більш дешеве та якісне вирішення конкретних завдань. Нейрокомп'ютер реалізує ідею створення аналого-цифрової ЕОМ, у якій аналогова частина виконує багатовимірні операції в граничному базисі, а цифрова реалізує алгоритми настроювання параметрів нейронних мереж [47].

Деякі порівняльні характеристики комп'ютера й ШНМ наведено в табл. В.1.

Таблиця В.1

	Комп'ютер	ШНМ
Підхід до вирішення задачі	Дедуктивний (використовує відомі правила перетворення вхідних даних у вихідні)	Індуктивний (задані вхідні й вихідні дані використовуються для конструювання правил)
Обчислення	Централізовані, синхронні, послідовні	Розподілені, асинхронні, паралельні
Пам'ять	Пакетована, зосереджена й адресована за розташуванням	Розподілена, адресована за змістом
Стійкість до перебоїв	Нестійкий (вихід з ладу одного елемента призводить до виходу з ладу всього пристрою)	Стійка за рахунок надлишковості й поділу функцій
Швидкодія	Висока (млн оп./с)	Низька (тис. оп./с)
Точність	Висока	Невисока
Архітектура	Фіксована	Що змінюється

Висока швидкодія нейрокомп'ютерів досягається за рахунок розпаралелювання обчислень. Кардинальним напрямком розвитку нейрокомп'ютерів як загального призначення, так і проблемно-орієнтованих є розробка нейрочипів. Досягнення в мікроелектроніці вселяють упевненість у тому, що незабаром будуть створені могутніші й водночас більш дешеві нейронні ЕОМ [47, 48].

1. БІОЛОГІЧНІ ОСНОВИ НЕЙРОННИХ МЕРЕЖ

Уперше думка про те, що живі організми складаються з клітин, була сформульована в 1838 р. *Маттіасом Шлейденом* із Берліна, а в 1839 р. він разом із *Т. Шванном* опублікував працю, у якій було сказано, що «існує універсальний принцип утворення організмів... [який] може бути позначений терміном клітинна теорія» (*перекл. ред.*) [49–50][1]. Упродовж багатьох років (більшу частину XIX століття) тривали суперечки про те, чи застосовна клітинна теорія до нервової системи. Незважаючи на здобуті на той час досягнення, одиночна нервова клітина не розглядалася як єдине ціле. Ключ до мікроскопічного дослідження нервової системи дав лікар *К. Ґольджі*, розробивши в 1875 р. метод вибіркового фарбування нервової тканини, при якому повністю зафарбовується лише невелика частина клітин (і донині ніхто не знає, як і чому спрацьовує метод Ґольджі, забарвлюючи повністю одну зі 100 клітин і зовсім не зачіпаючи всі інші). Скориставшись методом Ґольджі, іспанський гістолог *С. Рамон-і-Кахал* установив, що, по-перше, нервова система є сукупністю окремих, відокремлених клітин, які сполучаються одна з одною за допомогою синапсів, і, по-друге, неймовірно складні зв'язки між нервовими клітинами не випадкові, а високо структуровані й специфічні. Він вивів також основні принципи, згідно з якими нервові сигнали випливають як по дендритах, так і по аксонах клітин, і передача сигналів між клітинами здійснюється в місцях контактів аксонів з дендритами. Зазначимо, що електрична природа нервового імпульсу встановлена *Е. де Буа-Раймондом* і *Г. фон Гельмгольцем* ще в 1850 р.

У 1891 р. *В. Вальдеєр* вперше ввів термін «нейрон», що в перекладі з грецької означає «нерв», запропонувавши назвати так нервову клітину. З цього часу клітинна теорія, застосовувана до нервової системи, стає відомою як «нейронна доктрина».

[1] У даному розділі використані матеріали монографій [49–51].

У рамках цієї доктрини *З. Фройд* пояснював проходження електричного сигналу через клітину й, використовуючи моделі нейронів (рис. 1.1), намагався з'ясувати механізми пам'яті, мислення, задоволення й т. д. [52].

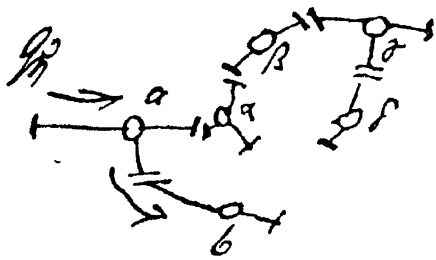


Рис. 1.1. Модель З. Фройда

1.1. Мозок людини

Характерною рисою мозку людини є, по-перше, різноманітність високоспеціалізованих дій, яким він здатний навчатися й, по-друге, розподіл функцій між півкулями. Передбачається, що перша особливість обумовлена структурою кори головного мозку, а розумова діяльність людини пов'язана з певними нервовими мережами, що є спеціалізованими системами в мозку, структури яких поки ще не встановлено. Факт того, що мозок людини не повністю симетричний у своїх функціях, вірогідно був установлений давно.

Дослідження мозку дозволило зробити два важливих висновки:

- різні ділянки мозку виконують різну роботу;
- мозок переробляє інформацію способами, зовсім відмінними від тих, які можна було б уявити (наприклад, упізнання букв і цифр, які, здавалося б, мають відбуватися в тому самому місці, очевидно, здійснюються в різних місцях. Справедливо й протилежне: деякі процеси, що уявляються роздільними, порушуються при ушкодженні однієї й тієї ж ділянки).

На карті кори мозку людини (рис. 1.2) показано ділянки, функціональна спеціалізація яких установлена. Більша частина кори відведена під порівняно елементарні функції: керування рухом

і первинний аналіз подразників. Ці ділянки, що містять моторну й соматосенсорну зони, а також первинні зорові, слухові й нюхові ділянки, є у всіх видів, що мають добре розвинену кору, і залучаються в роботу при багатьох родах діяльності.

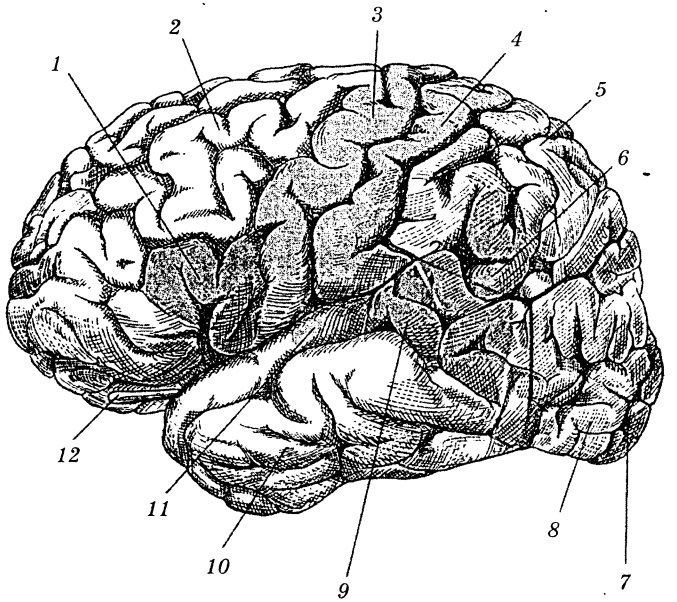


Рис. 1.2. Карта кори головного мозку людини:
1 — зона Брока; *2* — лобова частка; *3* — моторна кора; *4* — соматосенсорна кора; *5* — тім'яна частка; *6* — кутова звивина; *7* — первинна зорова зона; *8* — потилична частка; *9* — зона Верніке; *10* — скронева частка; *11* — первинна слухова зона; *12* — нюхова цибулина

Деякі інші ділянки, зображені на рис. 1.2 темними кольорами, більш вузько спеціалізовані. Зони Брока й Верніке беруть участь у формуванні й сприйнятті мови. Передбачається, що співвіднесення зорового й слухового подання інформації здійснює кутова звивина. Такі функціональні спеціалізації виявлено тільки на лівій половині мозку; відповідні ділянки правої півкулі не мають аналогічного зв'язку з лінгвістичними здібностями. Права півкуля, що тут не показана, визначає свої власні специфічні здібності, зокрема дотичні деяких аспектів сприйняття музики й складних зорових образів. Однак анатомічні зони, що асоціюються із цими

здібностями, визначені не так добре, як мовні зони. Навіть у лівій півкулі співвідношення функцій з ділянками кори лише приблизне; деякі зони можуть мати інші функції, крім зазначених, а у виконанні окремих функцій може брати участь кілька зон.

Останнім часом ці функціональні асиметрії були співвіднесені з анатомічними і був покладений початок дослідженню їхнього значення в інших біологічних видів, крім людини.

У людини, як і в інших ссавців, великі ділянки кори відведені під відносно елементарні сенсорні й моторні функції. Дуга, що йде, грубо кажучи, від вуха до вуха через покрівлю мозку, відповідає первинній моторній корі, що здійснює довільне керування м'язами. Паралельно цій дузі й прямо за нею розташована соматосенсорна зона, що отримує сигнали від шкіри, кісток, суглобів і м'язів. Майже кожна ділянка тіла представлена відповідною ділянкою як у первинній моторній, так й у соматосенсорной корі. У задній частині мозку й, зокрема, на внутрішній поверхні потиличних часток, розташовується первинна зорова кора. Первинні слухові зони знаходяться у скроневих частках, нюх зосереджений у певній ділянці на нижній частині лобових часток.

Спеціалізація соматосенсорної і моторної зон кори мозку виражається в тому, що кожну ділянку цих зон можна з'єднати з певною частиною тіла. Інакше кажучи, більша частина тіла мо-

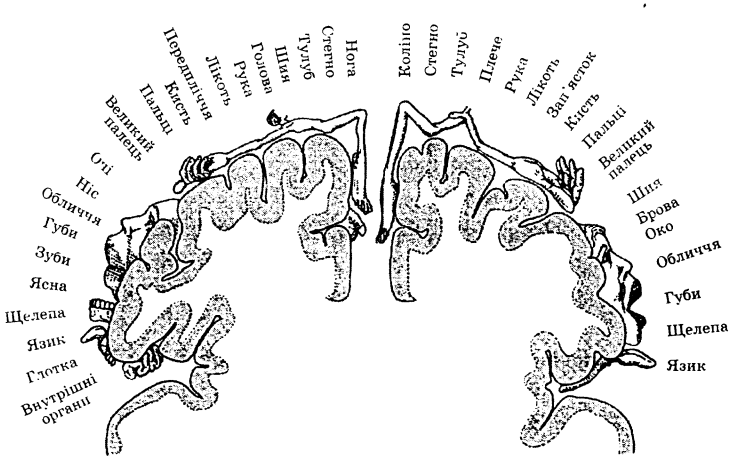


Рис. 1.3. Спеціалізація зон кори мозку

же бути представлена на корі у вигляді схеми; результатом цього будуть два перекручених гомункулуси (рис. 1.3). Перекручування пов'язані з тим, що відведена під дану частину тіла площа кори пропорційна не величині цієї частини, а необхідній точності керування. У людини дуже великі площі моторної й соматосенсорної зон, що відповідають обличчю й рукам. Тут показано тільки половини кожної з таких коркових зон: ліва соматосенсорна зона (яка отримує сигнали переважно від правої половини тіла) і права моторна зона (яка здійснює керування рухами лівої половини тіла).

Структурними елементами мозку є нейрони: гліальні клітини, що скріплюють нейрони й, імовірно, допомагають живити їх і вилучати непотрібні продукти обміну речовин; шванківські клітини, що утворюють захисну (мієлінову) оболонку аксонів; кровоносні судини та клітини, що їх складають; череп, що вміщує інші структури й забезпечує їхній захист.

1.2. Нейрон

Нейрони, або нервові клітини, є основними «будівельними блоками» мозку.

Вважають, що мозок людини складається з 10^{11} нейронів: це приблизно стільки ж, скільки зірок у нашій Галактиці.

Нервові клітини аж ніяк не однакові, вони поділяються на безліч різних типів. Хоча є й проміжні форми, у цілому цей розподіл на типи є досить чітким. Ніхто не знає, скільки типів нейронів існує в головному мозку, — їх, безсумнівно, більше сотні, а можливо, більше тисячі. Не існує двох зовсім однакових нейронів. Дві клітини того самого класу приблизно так само подібні між собою, як два дуби або два клени, а розходження двох класів можна порівняти з відмінністю кленів від дубів або навіть від кульбаб.

Незважаючи на це, їхні форми звичайно вміщуються в невелику кількість широких категорій, і більшості нейронів властиві певні структурні особливості, що дозволяють виділити: клітинне тіло, дендрити й аксон (рис. 1.4). Тіло містить ядро й біохімічний апарат синтезу ферментів та інших молекул, необхідних для життєдіяльності клітини. Зазвичай тіло нейрона має приблизно сферичну або пірамідальну форму. Від тіла клітини відходить кілька дендритів й один аксон. Тіло клітини та дендрити вкриті синапсами — бляшкоподібними структурами, через які надходить інформація від інших нейронів. Мітохондрії живлять клітину енергією.

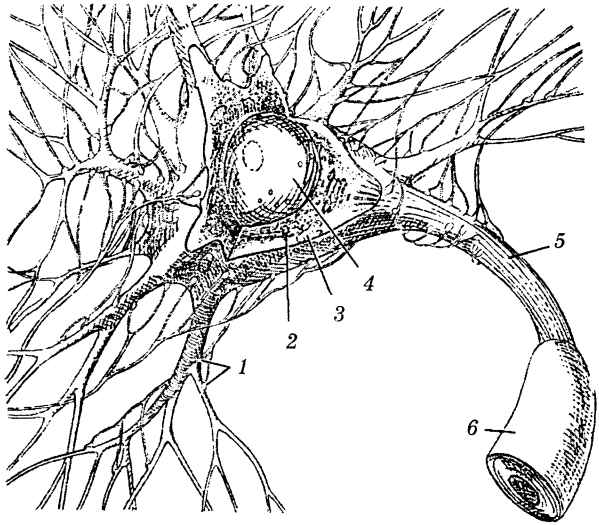


Рис. 1.4. Нейрон:
1 — синапси; *2* — шорсткуватий ендоплазматичний ретикулум; *3* — мітохондрія; *4* — ядро; *5* — аксон; *6* — мієлінова оболонка

Дендрити є тонкими трубчастими виростами, які багаторазово діляться й утворюють гіллясте дерево навколо тіла клітини. Вони створюють ту основну фізичну поверхню, на яку надходять сигнали, що йдуть до даного нейрона. Аксон тягнеться далеко від тіла клітини й служить тією лінією зв'язку, по якій сигнали, генеровані в тілі даної клітини, можуть передаватися на більші відстані в інші ділянки мозку й решти нервової системи. Аксон відрізняється від дендритів як за будовою, так і за властивостями своєї зовнішньої мембрани. Більшість аксонів довше й тонше дендритів і має відмінний від них характер галуження: якщо відростки дендритів в основному групуються навколо клітинного тіла, то відростки аксонів розташовуються на кінці волокна, у тому місці, де аксон взаємодіє з іншими нейронами.

Функціонування мозку пов'язане з рухом потоків інформації по складних ланцюгах, що складаються із нейронних мереж. Інформація передається від однієї клітини до іншої в спеціалізованих місцях контакту — синапсах. Типовий нейрон може мати від 1000 до 10 000 синапсів й отримувати інформацію від тисячі інших нейронів. Хоча у своїй більшості синапси утворюються між аксонами однієї клітини й дендритами іншої, існують інші типи

синаптичних контактів: між аксоном й аксоном, між дендритом і дендритом та між аксоном і тілом клітини.

В області синапса аксон звичайно розширюється, утворюючи на кінці пресинаптичну бляшку, що є частиною контакту, яка передає інформацію (рис. 1.4). Кінцева бляшка містить дрібні сферичні утворення, що називаються синаптичними пухирцями, кожний з яких містить кілька тисяч молекул хімічного медіатора. Після прибуття в пресинаптичне закінчення нервового імпульсу деякі з пухирців викидають свій уміст у вузьку щілину, що відокремлює бляшку від мембрани дендрита іншої клітини, призначеного для прийому таких хімічних сигналів. Таким чином, інформація передається від одного нейрона іншому за допомогою деякого посередника, або медіатора. Імпульсація нейрона відбиває активацію нейронами сотень синапсів. Деякі синапси є збуджувальними, тобто вони сприяють генерації імпульсів, тоді як інші — гальмові — здатні анулювати дію сигналів, які за їх відсутності могли б викликати розряд нейрона.

Для збуджувальних синапсів характерні сферичні пухирці й суцільне стовщення постсинаптичної мембрани, а для гальмових — сплощені пухирці й несуцільне стовщення мембрани. Синапси можна також класифікувати за їхнім розташуванням на поверхні сприймаючого нейрона — на тілі клітини, на стовбурі або «шипику» дендрита, або на аксоні (рис. 1.5).

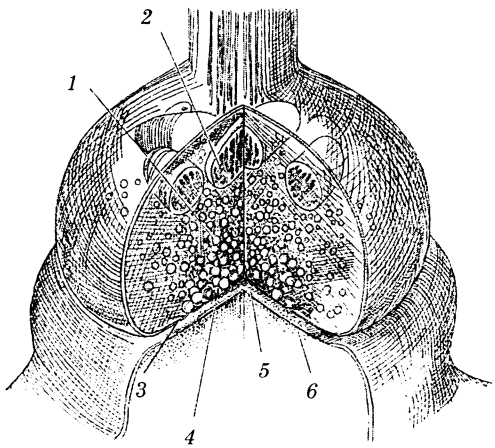


Рис. 1.5. Синапс:
1 — пухирці; *2* — мітохондрія; *3* — пресинаптична мембрана; *4* — постсинаптична мембрана; *5* — синаптична щілина; *6* — іонні канали

1.3. Передавання інформації

Завдання нервової клітини полягає в тому, щоб приймати інформацію від клітин, які її передають, підсумовувати, або інтегрувати, цю інформацію й доставляти інтегровану інформацію іншим клітинам. Інформація звичайно передається у формі короткочасних процесів, що називаються *нервовими імпульсами*. У будь-якій клітині кожен імпульс має бути таким самим, як і будь-який інший, тобто імпульс — це стереотипний процес. У будь-який момент частота імпульсів, що посилається нейроном, визначається сигналами, тільки що отриманими ним від передавальних клітин, і передає інформацію клітинам, стосовно яких цей нейрон є передавальним. Частота імпульсів варіює від одного в кожні кілька секунд або ще нижче до максимуму близько тисячі за секунду.

Сигнали можуть бути двоякими — електричними й хімічними. Сигнал, що генерується нейроном і проводиться по його аксону, є електричним імпульсом, але від клітини до клітини він передається молекулами передавачів, медіаторів.

Нейрони здатні виконувати свою функцію тільки завдяки тому, що їхня зовнішня мембрана має особливі властивості. Мембрана аксона по всій його довжині спеціалізована для проведення електричного імпульсу. Мембрана аксонних закінчень здатна виділяти медіатор, а мембрана дендритів реагує на медіатор. Крім того, мембрана забезпечує впізнавання інших клітин у процесі ембріонального розвитку так, що кожна клітина відшукує призначене їй місце в мережі, що складається з 10^{11} клітин.

Що відбувається, коли інформація передається від однієї клітини до іншої через синапс? У першій — пресинаптичній клітині біля основи аксона виникає електричний сигнал, або імпульс. Імпульс переміщується по аксону до його закінчень. З кожного закінчення в результаті цього імпульсу у вузький (0,02 мкм) заповнений рідиною проміжок, що відокремлює одну клітину від іншої, — *синаптичну щілину* — вивільняється хімічна речовина, що дифундує до другої — *постсинаптичної* — клітини. Вона впливає на мембрану цієї другої клітини таким чином, що ймовірність виникнення в ній імпульсів або зменшується, або зростає. Після цього короткого опису повернемося назад і розглянемо весь процес докладно.

Нервова клітина обмивається сольовим розчином і містить його всередині. До солей відноситься не тільки хлористий натрій, але також хлористий калій, хлористий кальцій і ряд інших, менш

звичайних солей. Оскільки більшість молекул солі дисоційовано, рідини як усередині, так і зовні клітини містять іони хлору, калію, натрію й кальцію (Cl^-, K^+, Na^+ й Ca^{2+}).

У стані спокою електричні потенціали всередині й зовні клітин розрізняються приблизно на одну десяту частку вольта, причому плюс знаходиться зовні (рис. 1.6). Точне значення ближче до величини 0,07 вольта, або 70 мілівольт. Передані нервами сигнали є швидкими змінами потенціалу, що переміщуються по волокну від тіла клітини до закінчень аксона.

Мембрана нервової клітини, що вкриває весь нейрон, — структура надзвичайно складна. Вона не суцільна, а містить мільйони «пор», через які речовини можуть переходити з одного боку на інший. Деякі з них — це дійсно пори різного розміру; вони є білками у формі трубок, що наскрізь пронизують жирову речовину мембрани. В інших випадках це не просто пори, а мініатюрні білкові механізми, що називаються насосами; вони здатні уловлювати іони одного типу й викидати їх із клітини, одночасно захоплюючи інші іони всередину із зовнішнього простору. Таке перекачування вимагає витрати енергії, яку клітина в остаточному підсумку отримує у процесі окислювання глюкози. Існують також пори, що називаються каналами, — це «клапани», які можуть відкриватися й

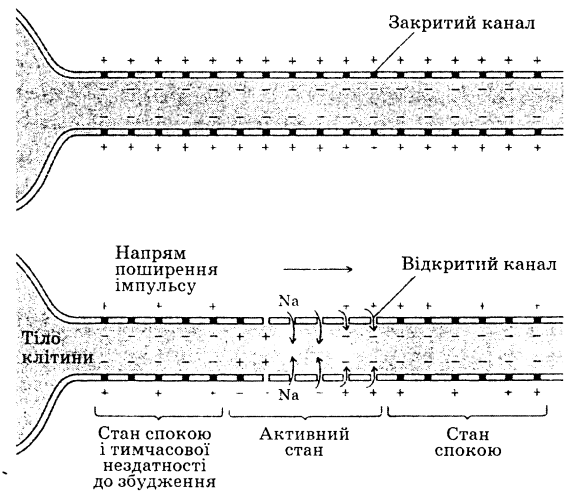


Рис. 1.6. Процес поширення імпульсу

закриватися. Який вплив призводить до їхнього відкриття або закриття, залежить від типу пор. На деякі з них впливає мембранний потенціал, інші відкриваються або закриваються за наявності певних речовин у внутрішній і зовнішній рідині.

Різниця потенціалів на мембрані в будь-який момент визначається концентрацією іонів усередині й зовні, а також тим, відкриті чи закриті різні пори.

Відкриття калієвих пор призводить до виникнення на мембрані різниці потенціалів з позитивним полюсом зовні. Якщо відкрити натрієві пори — зовні виникає негативний заряд.

Коли нерв перебуває в спокої, більшість калієвих каналів відкрито, а більшість натрієвих закрито; тому зовні буде позитивний заряд. Під час імпульсу на короткому відрізку нервового волокна раптово відкривається велика кількість натрієвих каналів, що призводить до короткочасної переваги потоку іонів натрію, і ця ділянка швидко стає електронегативною зовні стосовно внутрішнього простору. Потім натрієві пори знову закриваються, у той час як калієві залишаються відкритими, причому навіть у більшій кількості, ніж у стані спокою. Обидва процеси — закриття натрієвих пор і додаткове відкриття калієвих пор — призводять до швидкого відновлення потенціалу спокою з позитивним полюсом зовні. Вся послідовність подій займає приблизно тисячну частку секунди.

Зменшення потенціалу на мембрані з подальшою зміною його знака (реверсією) не відбувається відразу по всій довжині волокна, оскільки перенос заряду вимагає часу. Активна ділянка виникає в одному місці й переміщується по волокну зі швидкістю від 0,1 до приблизно 10 метрів за секунду. У будь-який момент часу існує одна активна ділянка з реверсованим потенціалом, і ця область реверсії пересувається, віддаляючись від тіла нейрона; попереду неї знаходиться ділянка із ще не відкритими каналами, а позаду — ділянка, де канали знову закрилися й тимчасово нездатні до повторного відкриття. Процес поширення імпульсу представлений на рис. 1.6.

На цьому рисунку вгорі — ділянка аксона в стані спокою. Натрієвий насос перекачав назовні зайві іони натрію, а всередину — відсутні іони калію. Натрієві канали переважно закриті. Оскільки відкрито багато калієвих каналів, клітину покинула достатня кількість калію, щоб мембранний потенціал досяг рівноважного в таких умовах рівня — близько 70 мілівольтів із плюсом зовні. Внизу — зліва направо переміщається нервовий імпульс. На крайньому правому кінці аксон перебуває ще в стані спокою. У серед-

ній ділянці відбуваються процеси, пов'язані з імпульсом: натрієві канали відкриті, іони натрію переходять усередину (хоча й не в такій кількості, щоб їхня концентрація після одного імпульсу помітно змінилася); мембранний потенціал 40 мілівольтів із плюсом усередині. На крайньому лівому кінці мембрана повертається у вихідний стан, оскільки відкрилися (а потім закрилися) додаткові калієві канали, а натрієві канали автоматично закрилися. Оскільки натрієві канали не здатні відразу ж повторно відкритися, другий імпульс не може виникнути раніше, ніж через приблизно мілісекунду. Це дозволяє зрозуміти, чому імпульс не може повернути назад до тіла клітини.

Центральна нервова система, що знаходиться між вхідними й вихідними нейронами, є тим апаратом, що здійснює сприйняття, реагування, запам'ятовування й т. д. Чимало відділів центральної нервової системи організовані у вигляді послідовних шарів-рівнів (рис. 1.7). Клітина одного рівня отримує численні збуджувальні й гальмові входи від попереднього рівня й посилає вихідні сигнали багатьом клітинам наступного рівня. Основну масу вхідної інформації нервова система отримує від рецепторів: очей, вух, шкіри й т. д., які перетворюють такі зовнішні впливи, як світло, тепло або звук в електричні нервові сигнали. Виходом можуть бути скорочення м'язів або залозистих клітин.

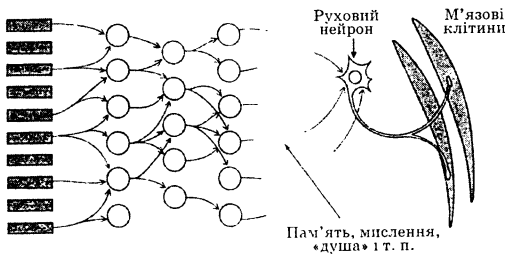


Рис. 1.7. Багаторівнева організація центральної нервової системи

Таким чином, з погляду кібернетики як власне центральна нервова система, так й її відділи являють собою здебільшого «чорний ящик», дослідження якого є надзвичайно складним. Навіть результати найбільш вивченої частини мозку, зорової системи («сірого ящика»), що дозволили простежити шлях інформації від клітин сітківки до шостого-сьомого етапу й установити роль кори півкуль великого мозку, пов'язаної із зором, не дають відповіді на

запитання, як відбувається сприйняття або впізнання предметів. Однак, незважаючи на це, багато виявлених властивостей центральної нервової системи можуть бути промодельовані й реалізовані за допомогою технічних засобів.

Контрольні запитання

1. Які основні висновки зроблено в результаті дослідження ШНМ?
2. Що являє собою карта кори головного мозку?
3. З чого складається нейрон?
4. Яким чином відбувається передача інформації?
5. Що являє собою центральна нервова система з точки зору кібернетики?

2. ОСНОВНІ ПОНЯТТЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

2.1. Структура штучного нейрона

Штучні нейрони, що також називаються *нейронними клітинами*, *вузлами*, *модулями*, моделюють структуру й функції біологічних нейронів. Архітектура й особливості штучних нейронних мереж, утворених нейронами, залежать від конкретних завдань, які мають бути вирішені з їхньою допомогою [34, 53–60].

Структуру штучного нейрона зображено на рис. 2.1.

Вхідними сигналами штучного нейрона $x_i (i = \overline{1,N})$ є вихідні сигнали інших нейронів, кожний з яких узятий зі своєю вагою w_i $(i = \overline{1,N})$, аналогічною синаптичній силі.

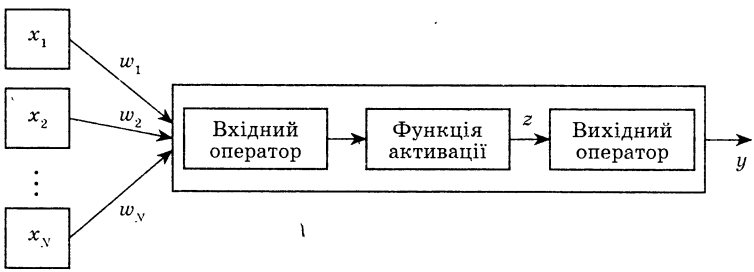


Рис. 2.1. Структура штучного нейрона

Вхідний оператор $f_{вх}$ перетворює зважені входи й подає їх на *оператор активації* f_a. Вихідний сигнал нейрона y являє собою перетворений *вихідним оператором* $f_{вих}$ вихідний сигнал оператора активації. Таким чином, нелінійний оператор перетворення вектора вхідних сигналів x у вихідний сигнал y може бути записаний у такий спосіб:

$$y = f_{вих}(f_a(f_{вх}(\boldsymbol{x},\boldsymbol{w}))). \qquad (2.1)$$

Як ми вже зазначали, вихідний сигнал даного нейрона є вхідним для наступного.

Вхідний оператор

Вхідний оператор (вхідна функція) нейрона задає вигляд використовуваного в нейроні перетворення зважених входів. Відмінність гальмуючих входів від збуджувальних відбивається у знаках відповідних ваг. Звичайно використовуються такі вхідні функції:

— сума зважених входів

$$f(\boldsymbol{x},\boldsymbol{w}) = \sum_{i=1} w_i x_i; \tag{2.2}$$

— максимальне значення зважених входів

$$f(\boldsymbol{x},\boldsymbol{w}) = \max_i (w_i x_i); \tag{2.3}$$

— добуток зважених входів

$$f(\boldsymbol{x},\boldsymbol{w}) = \prod_{i=1}^{N} w_i x_i; \tag{2.4}$$

— мінімальне значення зважених входів

$$f(\boldsymbol{x},\boldsymbol{w}) = \min_i (w_i x_i). \tag{2.5}$$

Функція активації

Функція активації (*activation function*) $f_a(\cdot)$ описує правило переходу нейрона, що перебуває в момент часу k у стані $z(k)$, у новий стан $z(k+1)$ при надходженні вхідних сигналів $\boldsymbol{x}$

$$z(k+1) = f_a(z(k), f_{вх}(\boldsymbol{x}, \boldsymbol{w})). \tag{2.6}$$

Надалі позначатимемо функцію активації без індексу «а».

Найбільш простими активаційними функціями є

— *лінійна*

$$f(z) = Kz, \quad K = \text{const}; \tag{2.7}$$

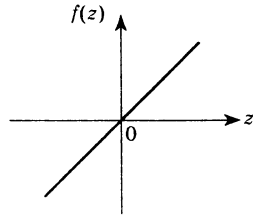


Рис. 2.2. Лінійна функція

— лінійна біполярна з насиченням

$$f(z)=\begin{cases}1 & \text{при} \quad z>\alpha_2;\\ Kz & \text{при} \quad -\alpha_1\le z\le\alpha_2;\\ -1 & \text{при} \quad z<\alpha_1;\end{cases} \tag{2.8}$$

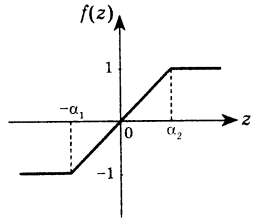


Рис. 2.3. Лінійна біполярна функція з насиченням

— лінійна уніполярна з насиченням

$$f(z)=\begin{cases}1 & \text{при} \quad z\ge\dfrac{1}{2\alpha};\\ \alpha z+0{,}5 & \text{при} \quad |z|<\dfrac{1}{2\alpha};\\ 0 & \text{при} \quad z\le-\dfrac{1}{2\alpha}.\end{cases} \tag{2.9}$$

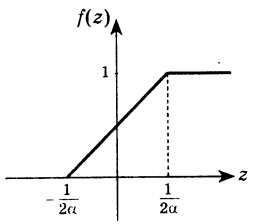


Рис. 2.4. Лінійна уніполярна функція з насиченням

Незважаючи на те, що лінійні функції є найбільш простими, їхнє застосування обмежене, в основному, найпростішими ШНМ, які не мають у своєму складі прихованих шарів, у яких, крім того, існує лінійна залежність між вхідними й вихідними змінними. Такі мережі мають обмежені можливості. Для багатошарової ж

лінійної мережі справедливо наступне. Оскільки після вхідного оператора на оператор активації надходить сукупність зважених вхідних сигналів, записана, наприклад, у матричному вигляді (більш докладно *див.* нижче) $W_1\boldsymbol{x}$, використання лінійної активаційної функції призводить до того, що на виході другого шару з'явиться сигнал $W_2(W_1\boldsymbol{x}) = (W_2W_1)\boldsymbol{x}$. Це означає, що двошарова лінійна мережа еквівалентна одношаровій з ваговою матрицею, що дорівнює добутку вагових матриць першого й другого шарів. Звідси випливає, що будь-яка багатошарова лінійна мережа може бути замінена еквівалентною одношаровою. Хоча, як показано в [34], використання лінійних активаційних функцій не є зайвим у багатошарових ШНМ, для розширення ж можливостей мережі застосовують нелінійні функції активації.

У роботі *У. Маккаллоха* і *У. Піттса* [1] як активаційна використовувалася функція Хевісайда — *уніполярна гранична функція* вигляду

$$f(z) = \begin{cases} 1 & \text{при } z \geq \alpha; \\ 0 & \text{при } z < \alpha. \end{cases} \tag{2.10}$$

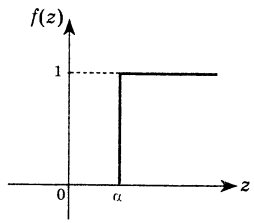


Рис. 2.5. Уніполярна порогова функція

Різновидом даної функції є *біполярна порогова функція*

$$f(z) = \begin{cases} 1 & \text{при } z \geq \alpha; \\ -1 & \text{при } z < \alpha. \end{cases} \tag{2.11}$$

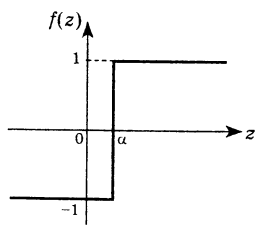


Рис. 2.6. Біполярна порогова функція

Ці функції активації застосовувалися в основному в класичних ШНМ. При побудові нових структур ШНМ найчастіше доводиться працювати як із самою активаційною функцією, так і з її першою похідною. У цих випадках необхідним є використання як активаційної монотонної диференційованої й обмеженої функції. Особливо важливу роль відіграють такі функції під час моделювання нелінійних залежностей між вхідними й вихідними змінними. Це так звані *логістичні*, або *сигмоїдальні* (*S-подібні*), функції.

Функція $f(\cdot)$ називається *сигмоїдальною*, якщо вона є монотонно зростаючою, диференційованою і задовольняє умові

$$\lim_{\lambda \to -\infty} f(\lambda) = k_1, \quad \lim_{\lambda \to \infty} f(\lambda) = k_2, \quad k_1 < k_2. \tag{2.12}$$

До таких функцій належать:

— *логістична* (*уніполярна*)

$$f_{\log}(z) = \frac{1}{1+e^{-\alpha z}}; \tag{2.13}$$

$$\frac{d}{dz} f_{\log}(z) = \alpha f_{\log}(z)\left(1 - f_{\log}(z)\right);$$

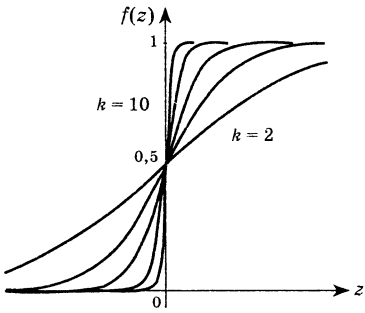


Рис. 2.7. Логістична функція

— *гіперболічного тангенса* (*біполярна*)

$$f_{\text{th}}(z) = \tanh(\alpha z) = \frac{e^{\alpha z} - e^{-\alpha z}}{e^{\alpha z} + e^{-\alpha z}}; \tag{2.14}$$

$$\frac{d}{dz} f_{\text{th}}(z) = 1 - \tanh^2(\alpha z).$$

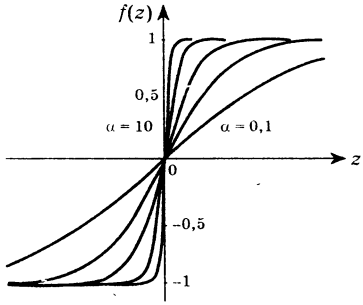


Рис. 2.8. Функція гіперболічного тангенса

Функції (2.13) і (2.14) можуть бути виражені одна через одну. Наприклад,

$$f_{\text{th}}(z) = \tanh(\alpha z) = \frac{e^{\alpha z}(1-e^{-2\alpha z})}{e^{\alpha z}(1+e^{-2\alpha z})} = \frac{2-(1+e^{-2\alpha z})}{1+e^{-2\alpha z}} = 2f_{\log}(z) - 1.$$

Аналогічно можна показати, що $f_{\log}(z) = \frac{1}{2}(\tanh(\frac{z}{2}) - 1)$.

Слід також зазначити, що перевага функції $f_{\text{th}}(z)$ перед $f_{\log}(z)$ полягає в її симетричності відносно початку координат (у деяких випадках це істотно полегшує обчислення).

— *синусоїдальна з насиченням* (*біполярна*)

$$f_{\sin}(z) = \begin{cases} 1 & \text{при} \quad z \geq \alpha; \\ \sin z & \text{при} \quad |z| < \alpha; \\ -1 & \text{при} \quad z \leq -\alpha, \end{cases} \tag{2.15}$$

$$\frac{d}{dz} f_{\sin}(z) = \sqrt{1 - f_{\sin}^2(z)};$$

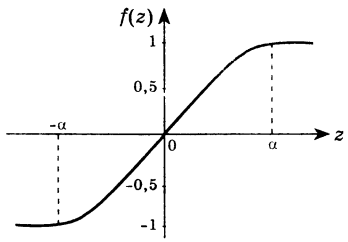


Рис. 2.9. Функція синусоїдальна з насиченням

— *косинусоїдальна з насиченням (уніполярна)*

$$f_{\cos}(z)=\begin{cases}1 & \text{при } z\geq\dfrac{\pi}{2};\\ \dfrac{1}{2}\left(1+\cos\left(z-\dfrac{\pi}{2}\right)\right) & \text{при } |z|<\dfrac{\pi}{2};\\ 0 & \text{при } z\leq-\dfrac{\pi}{2}.\end{cases} \tag{2.16}$$

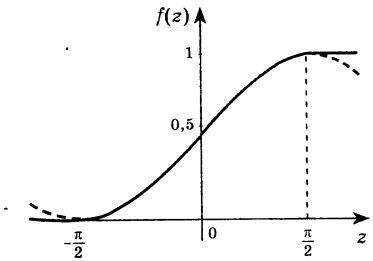


Рис. 2.10. Функція косинусоїдальна з насиченням

— *модульована сигмоїда*

$$f(x,y)=\frac{1}{1+e^{-x-y-\theta}}-\frac{1}{1+e^{x-y-\theta}}. \tag{2.17}$$

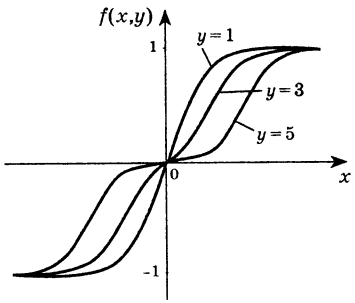


Рис. 2.11. Функція «модульована сигмоїда»

Сигмоїдальну функцію за аналогією з електронними системами можна вважати нелінійною підсилювальною характеристикою штучного нейрона. Центральна область такої функції, що має великий коефіцієнт підсилення, вирішує проблему обробки слабких сигналів, а області зі спадним посиленням на позитивному й негативному кінцях служать для обробки значних збуджень. Таким

чином, нейрон функціонує з більшим підсиленням у широкому діапазоні рівнів вхідного сигналу.

На завершення зазначимо, що вибір конкретного виду активаційної функції специфічний для кожного виду ШНМ і залежить від розв'язуваного завдання.

Вихідний оператор

Вихідний оператор служить для представлення стану нейрона в бажаній області значень. Зазвичай у більшості робіт цей оператор не виділяють, а під вихідним сигналом нейрона розуміють сигнал після оператора активації. Однак під час аналізу й синтезу ШНМ, що містять різні активаційні функції, які мають різні області значень й області визначення, виникає необхідність використання такого оператора $f_{вих}$.

2.2. Моделі штучних нейронів

Моделі штучних нейронів залежать від конкретних застосувань. Тому синтез моделі в кожному окремому випадку є нетривіальним завданням.

2.2.1. Формальна модель нейрона Маккаллоха — Піттса

Формальний штучний нейрон (його називають також *нейроном Маккаллоха — Піттса*) може бути поданий як багатовхідний нелінійний перетворювач із ваговими коефіцієнтами w_{ji}, які також називаються синаптичними вагами або підсилювачами (рис. 2.12). *Клітина тіла* (*сома*) описується нелінійною обмежувальною або пороговою функцією $f(u_j)$. Найпростіша модель штучного нейрона підсумує N ваг входів і здійснює нелінійне перетворення (*див.* рис. 2.1)

$$y_j = f(\sum_{i=1}^{N} w_{ji} x_i + \theta_j), \qquad (2.18)$$

де y_j — вихідний сигнал j-го нейрона; f — обмежувальна або порогова функція (активаційна); N — кількість входів; w_{ji} — синаптичні ваги; x_i — вхідні сигнали ($i = \overline{1,N}$); θ_j, ($\theta_j \in R$) — пороговий сигнал, що також називається *зсувом*.

Позначаючи $\theta_j = w_{j0} x_0$ (зазвичай $x_0 = 1$), вираз (2.18) можна переписати у вигляді

$$y_j = f(\sum_{i=0}^{N} w_{ji} x_i) = f(\boldsymbol{w}_j^T \boldsymbol{x}), \qquad (2.19)$$

де $\boldsymbol{x} = (1, x_1, ..., x_N)^T$; $\boldsymbol{w}_j = (w_{j0}, w_{j1}, ..., x_{jN})^T$ — відповідні вектори входів і ваг розмірності $(N + 1) \times 1$.

Як випливає з (2.19), будь-який формальний нейрон характеризується своєю активаційною функцією й *порогом* θ_j. Перша модель нейрона, запропонована У. Маккаллохом і У. Піттсом, використовувала тільки бінарні (жорстко обмежені) функції (2.10). У цій моделі сума всіх зважених входів порівнювалася із пороговим значенням θ_j, і якщо вона перевищувала це порогове значення, вихід нейрона встановлювався у «верхнє значення» або логічну 1, в іншому випадку — в «нижнє» або логічний 0.

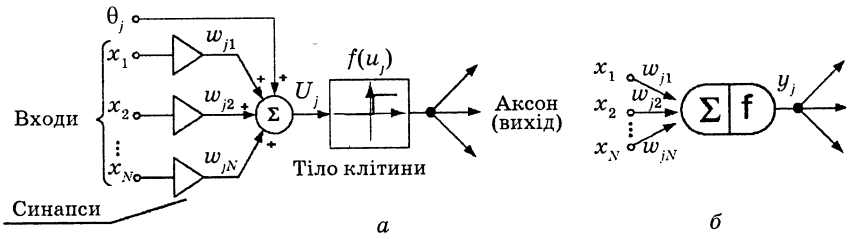


Рис. 2.12. Модель штучного нейрона Маккаллоха — Піттса та її позначення

У сучасних застосуваннях порогова функція заміняється більш загальною нелінійною функцією й, отже, вихід нейрона y_i може приймати як дискретне значення на множині (наприклад, $\{-1, 1\}$), так і неперервне (наприклад, між -1 й 1 або в загальному випадку між $y_{\min}$ й $y_{\max}$). Рівень активації або стан нейрона залежить від значення його вихідного сигналу y_i (наприклад, $y_i = 1$, якщо нейрон активний, і $y_i = 0$ — нейрон перебуває в незбудженому стані) і звичайно визначається монотонно зростаючою сигмоїдальною функцією.

Модель сигмоїдальної функції може бути побудована на основі традиційних електронних компонент. В електричному ланцюзі напруга імітує *тіло клітини* (*сому*), провідності замінюють вхідну структуру (*дендрити*) і вихідну (*аксон*) та різні резистори *моделюють синаптичні ваги* (*синапси*). Вихідна напруга y_i імітує вихідну реакцію біологічного нейрона. Сигмоїдальна активаційна функція

представлена вольт-амперною характеристикою підсилювача з насиченням. Сигнали x_i надходять у формі вхідної напруги у провідники-дендрити у співвідношенні сум вихідних напруг і відповідних електропровідностей (рис. 2.13).

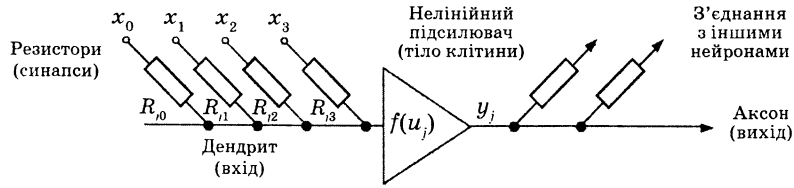


Рис. 2.13. Електронна аналогова модель нейронної клітини

Струм у вхідних ланцюгах схеми (дендритах) прямо пропорційний вхідним напругам x_i і провідностям w_{ji}. Напругу на виході може бути отримано відповідно до закону Кірхгофа

$$y_j = f(u_j) = f(\sum_{i=0}^{N} G_{ji} x_i / \sum_{i=0}^{N} G_{ji}), \qquad (2.20)$$

де y_i — вихідна напруга j-го нейрона; $f(\cdot)$ — сигмоїдальна функція підсилювача; u_j — вхідна напруга j-го підсилювача; $G_{ji} = R_{ji}^{-1}$ — провідність резисторів на вході схем; $x_j (j = \overline{0,N})$ — вхідні напруги.

Вираз (2.20) можна переписати в компактній формі у вигляді (2.19) з $w_{ji} = G_{ji} / \sum_{i=0}^{n} G_{ji}$.

2.2.2. Модель нейрона Адаліни

Однією з найважливіших властивостей біологічних і штучних нейронів є їхня здатність до навчання за допомогою зміни синаптичних ваг, здійснювана за тим чи іншим алгоритмом. Навчання штучних нейронних мереж полягає у виборі їхньої правильної реакції на пропоновані навчальні сигнали настроювання синаптичних ваг.

Практично одночасно з моделлю Маккаллоха — Піттса з'явилася модель штучного нейрона, що навчається, у вигляді *адаптивного лінійного елемента*, розробленого *Б. Уідроу* й названого *Адаліною* (***AD****Aptive* ***LI****near* ***NE****uron*) [6,7]. Адаліна складається з двох частин: лінійного підсилювача, що настроюється, і дворівневого

квантувача (з жорстко обмеженою або релейною активаційною функцією). Докладно Адаліна буде розглянута нижче.

2.2.3. Модель нейрона Фукушіми

У загальному випадку, як й у випадку моделі Маккалло-ха — Піттса, синаптичні ваги нейрона можуть бути позитивними, рівними нулю або негативними залежно від того, як впливають сигнали, що надходять, на його реакцію. Сигнали називаються *збуджувальними*, якщо ваги позитивні ($w_{ji} > 0$), і *гальмуючими* — якщо негативні ($w_{ji} < 0$). Однак у моделі штучного нейрона, запропонованого *К. Фукушімою*, всі синаптичні ваги й всі вхідні й вихідні сигнали є ненегативними, тобто вони можуть бути рівними нулю або приймати будь-яке позитивне значення [61–64]. У цій моделі входи й відповідні синаптичні ваги поділяються на дві групи: збуджувальні й гальмуючі (рис. 2.14).

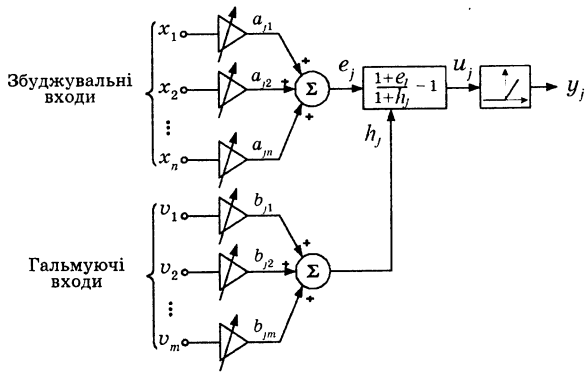


Рис. 2.14. Спрощена модель нейрона Фукушіми

Збуджувальний ефект e_j визначається зваженою сумою всіх збуджувальних входів. Вихід нейрона може бути описаний виразом

$$y_j = f\left[\frac{1+\sum_{i=1}^{n} a_{ji}x_i}{1+\sum_{i=1}^{m} b_{ji}\upsilon_i} - 1\right], \qquad (2.21)$$

де $f(u_j) = \begin{cases} u_j, & \text{якщо } u_j \geq 0; \\ 0 & \text{в іншому випадку.} \end{cases}$

У даному виразі a_{ji} відповідає збуджувальним синаптичним вагам, реалізованим у вигляді негативних електропровідностей, і b_{ji} — гальмуючим, також реалізованим у вигляді негативних електропровідностей. Синаптичні ваги звичайно є змінними, і вони змінюються у процесі навчання нейронної мережі.

2.2.4. Модель штучного нейрона Гопфілда

Усі описані вище структури нейронів належать до статичних структур і не здатні моделювати динамічний процес. Альтернативою їм може виступати модель нейрона *Дж. Гопфілда*, що є сьогодні найбільш популярною динамічною моделлю біологічного нейрона [23]. На рис. 2.15 зображено електричну й функціональну структури нейрона. Електрична схема складається з конденсатора C_j, резистора R_{ji} і нелінійного підсилювача із сигмоїдальною функцією. Передбачається, що підсилювач має два асиметричних виходи, при цьому всі резистори, що моделюють синаптичні ваги, мають позитивні значення. Це означає, що позитивна синаптична вага реалізується з'єднанням резистора R_{ji} з $(+v_i)$ і негативною вагою — з'єднанням R_{ji} із сигналом $(-v_i)$. Струм I_j представляє зсув (незалежний зовнішній вхідний сигнал).

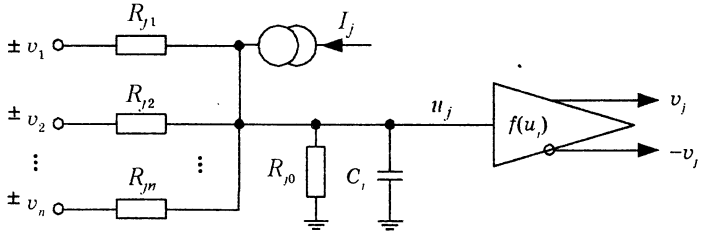


Рис. 2.15. Модель Гопфілда динамічної нейронної клітини

Застосовуючи закон Кірхгофа, отримуємо таке диференціальне рівняння, що описує нейрон:

$$C_j \frac{du_j}{dt} = -\frac{u_j}{R_j} + \sum_{i=1}^{N} \frac{v_i}{R_{ji}} + I_j, \qquad (2.22)$$

де $\frac{1}{R_j} = \frac{1}{R_{j0}} + \sum_{n=1}^{n} \frac{1}{R_{ji}} = G_{j0} + \sum_{i=1}^{N} G_{ji}$. Тут G_{ji} визначає електропровідності $(i = \overline{1,N})$; $v_j = f(u_j)$, $(j = \overline{1,N})$.

Вираз (2.22) може бути переписаний у більш загальній формі

$$T_j \frac{du_j}{dt} = -a_j u_i + \left(\sum_{i=0}^{N} w_{ji} x_i + \theta_j \right), \tag{2.23}$$

$$y_j = f(u_j), \tag{2.24}$$

де $T_j = r_j C_j$ — стала інтегрування; u_j — внутрішній сигнал, що називається внутрішнім потенціалом, або потенціалом дії; $a_j = r_j / R_j$ — коефіцієнт загасання; $w_{ji} = \pm r_j / R_{ji}$ — синаптичні ваги (з позитивним знаком, якщо R_{ji} з'єднаний з $(+x)$, з негативним знаком, якщо R_{ji} з'єднаний з $(-x)$); $x_i = v_i \, (i = \overline{1,N})$ — вхідні сигнали (напруги, потенціали); $\theta_j = r_j I_j$ — сигнал зсуву. Тут r_j — масштабуючий опір.

Схему нейрона, що відповідає виразам (2.22), (2.23), зображено на рис. 2.16. Схема містить суматор налаштовуваних синаптичних ваг w_{ji}, інтегратор і нелінійний елемент із сигмоїдальною активаційною функцією.

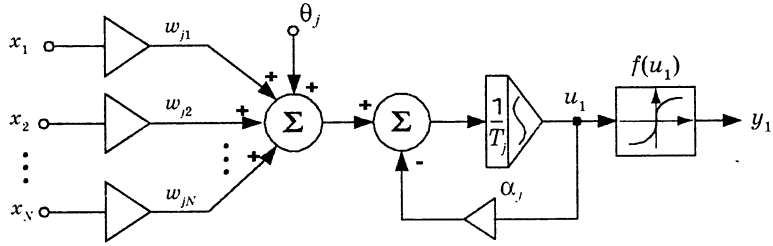


Рис. 2.16. Структура нейрона Гопфілда

Динаміка нейрона в цілому визначається ємністю C_j й опором R_{j0}. Синаптичні ваги визначаються вхідними електропровідностями $G_{ji} = 1/R_{ji}$, з'єднаними з одним з виходів $(+v$ або $-v)$ j-го підсилювача. Видно, що позитивні синаптичні ваги $(w_{ji} > 0)$ вимагають, щоб резистор R_{ji} був з'єднаний з позитивним, а негативні $(w_{ji} < 0)$ — з негативним входом j-го підсилювача.

Дискретну модель нейрона Гопфілда зображено на рис. 2.17.

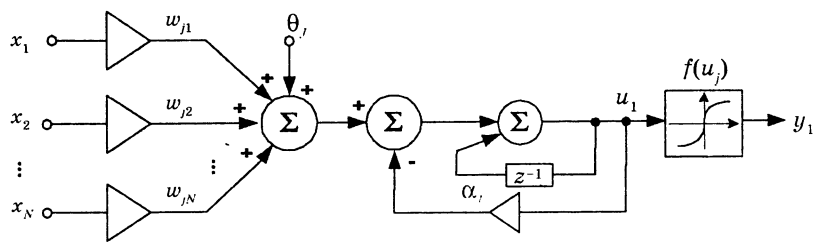


Рис. 2.17. Дискретна модель нейрона Гопфілда

2.2.5. Модель Ґроссберґа

Штучний нейрон, розроблений *С. Ґроссберґом* [20], є узагальненням моделі Гопфілда й може бути описаний таким диференціальним рівнянням:

$$T_j \frac{du_j}{dt} = -\alpha_j u_j + (\gamma_j - \beta_j u_j)\left(\sum_{i=1}^{n} w_{ji} f_i(u_i, \theta_j)\right), \qquad (2.25)$$

де u_i — внутрішня активність j-го нейрона; α, β, γ — константи, що визначають динаміку нейрона.

Настроювання параметрів нейрона здійснюється за допомогою алгоритму навчання вигляду

$$\frac{dw_{ji}}{dt} = \left[-\theta_{ji} w_{ji} + d_{ji} f_i(u_i)\right] h_i(u_i). \qquad (2.26)$$

Спрощена функціональна схема моделі штучного нейрона Ґроссберґа, що відповідає виразу (2.25), показана на рис. 2.18. Нейрон може перебувати у двох станах: збудженому і загальмованому, які описуються рівняннями вигляду:

$$T_j \frac{du_j}{dt} = -\alpha_j u_j + (\gamma_{jE} - \beta_{jE} u_j)\left(\sum_{i=1}^{N_E} w_{jiE} f_{iE}(u - i) + \theta_{jE}\right) -$$

$$- (\gamma_{jI} + \beta_{jI} u_j)\left(\sum_{i=1}^{N_I} w_{jiI} f_{iI}(u - i) + \theta_{jI}\right). \qquad (2.27)$$

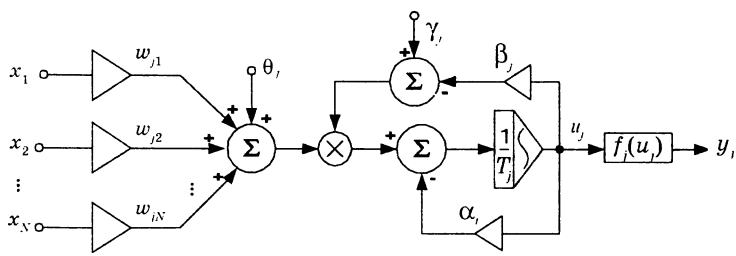


Рис. 2.18. Функціональна модель нейрона Ґроссберґа

На рис. 2.19 зображено дискретну модель нейрона Ґроссберґа.

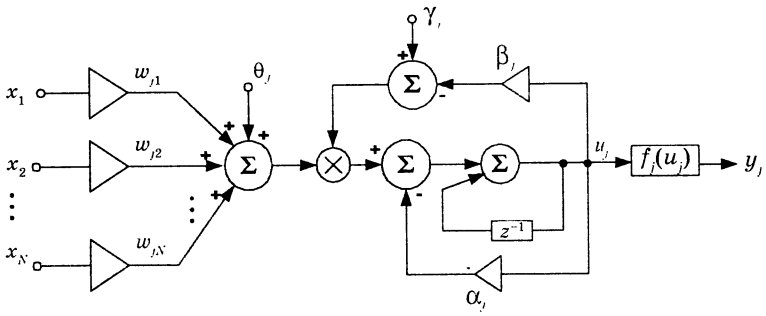


Рис. 2.19. Дискретна модель нейрона Ґроссберґа

Більшість відомих моделей штучних нейронів може бути розглянуто як окремі випадки описаних вище моделей.

2.2.6. Узагальнена модель нейрона

У деяких застосуваннях використовуються узагальнені моделі, які містять у собі більш складні комплексні математичні операції, ніж підсумовування. Найбільш загальне подання може бути записане у вигляді системи нелінійних диференціальних рівнянь

$$T_j \frac{du_j}{dt} = \varphi_j\left(J_j(u_j),\, w_{j1},\, w_{j2} f_2(u_2),\, \ldots,\, w_{jN} f_N(u_N)\right); \qquad (2.28)$$

$$T_{ji} \frac{dw_{ji}}{dt} = g_i(w_{ji},\, u_i,\, u_j), \qquad (2.29)$$

де $\varphi_j(\cdot)$ — нелінійне перетворення на вході нейрона; $J_j(\cdot)$ — функція, що описує внутрішню динаміку j-го нейрона; g_j — передавальна функція, що визначає динаміку зміни синаптичних ваг.

Усі наведені динамічні моделі штучних нейронів подано в неперервному часі. Динамічна поведінка таких нейронів звичайно описується системою диференціальних рівнянь, але під час моделювання роботи ШНМ використовується їхнє дискретне подання.

У загальному випадку дискретна математична модель нейрона має вигляд

$$x_j(k+1) = f\left(\sum_{i=1}^{N} w_{ji} x_i(k) + \theta_j\right). \qquad (2.30)$$

На рис. 2.20 зображено ще одну дискретно-часову модель штучного нейрона, описану виразами

$$u_j(k+1) = u_j(k) + \left(\sum_{i=1}^{N} w_{ji} x_i(k) + \theta_j \right),$$

$$x_j(k+1) = f(u_j(k+1)) \qquad j = \overline{1,N}, \tag{2.31}$$

де $k = 0, 1, 2, \ldots$ — індекс поточного дискретного часу.

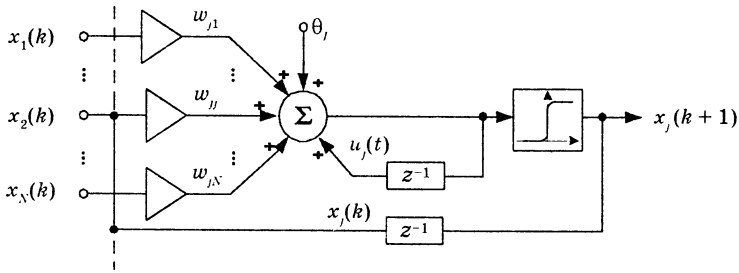


Рис. 2.20. Дискретно-часова модель нейрона

У цій моделі інтегратор замінюється суматором, охопленим елементом чистого запізнювання — дигратором. Дискретно-часова модель нейрона може бути отримана з аналогової моделі шляхом конвертування диференціальних рівнянь у відповідні різницеві рівняння.

Хоча можливості нейронів обмежені, мережі, побудовані з їхньою допомогою, мають практично необмежені можливості.

2.2.7. Σ-П-нейрон

Нейрон даного типу покликаний більш точно відбити властивості біологічного нейрона, а саме здатність моделювати певні синаптичні контакти. З цією метою в цьому нейроні, на відміну від лінійної моделі (2.18), використовується більш складне перетворення вхідних сигналів — їхній добуток або кореляція [37]

$$z(w,x) = \sum_{i=1}^{m} w_i \prod_{j=1}^{N_i} x_j. \tag{2.32}$$

Вихідний сигнал формується так само, як й у розглянутому в підрозділі 2.2.1 нейроні:

$$y = f(w,x) = f\left(\sum_{i=1}^{m} w_i \prod_{j=1}^{N_i} x_j\right), \tag{2.33}$$

де $f(\cdot)$ — функція активації (найчастіше гранична).

Σ-П-нейрон наведено на рис. 2.21.

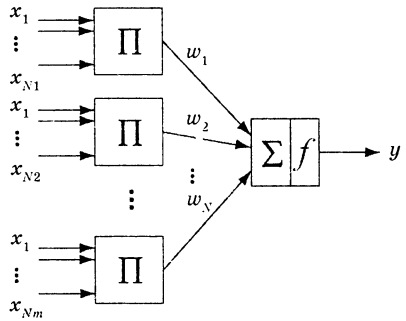


Рис. 2.21. Σ-П-нейрон

2.2.8. Стохастичний нейрон

Розглянуті вище моделі нейронів є детермінованими, оскільки при подачі на їхній вхід деяких сигналів вони видаватимуть однозначно обумовлений використовуваною активаційною функцією вихідний сигнал. У роботі [65] була розглянута модель стохастичного нейрона, у якій активаційна функція є випадковою й залежить від активності нейрона z. При цьому вихідний сигнал нейрона формується за таким правилом:

$$y = \begin{cases} +1 & \text{з імовірністю } P(z \mid y = 1); \\ -1 & \text{з імовірністю } P(z \mid y = -1), \end{cases} \tag{2.34}$$

де $P(z|y = 1) + P(z|y = -1) = 1$.

Модель такого нейрона зображено на рис. 2.22.

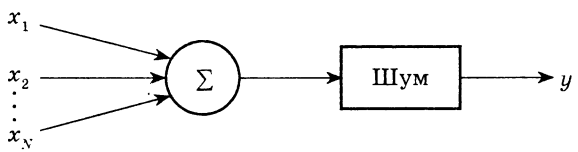


Рис. 2.22. Модель стохастичного нейрона

Так, якщо $P(z|y=1)=(1+\exp(-2\alpha z)^{-1})$, то $P(z|y=-1)=1-P(z|y=1)=$ $=-\exp(-2\alpha z)$, а математичне очікування вихідного сигналу $<y>$ визначається в такий спосіб:

$$<y> = 1P(z|y=1) + (-1)\,P(z|y=-1) = \tanh(\alpha z).$$

Зазначимо, що ця модель є основою *машини Больцмана* й *машини Коші*, а також лежить в основі навчання із підкріплюванням. Ці питання більш докладно будуть висвітлені нижче.

2.3. Топологія ШНМ

З'єднані між собою нейрони утворюють ШНМ. Таким чином, ШНМ — пара (M, V), де M — множина нейронів; V — множина зв'язків. Структура мережі задається у вигляді *графа*, у якому вершини є нейронами, а ребра являють собою зв'язки (з'єднання).

Кожен нейрон мережі має вхідні ланцюги, причому їхня кількість є довільною для кожного нейрона.

У загальному випадку ШНМ складається з декількох шарів, серед яких обов'язково є *вхідний*, що отримує зовнішні сигнали, *вихідний*, що відбиває реакцію нейронів на комбінації вхідних сигналів, і в багатошарових ШНМ — *приховані шари* (рис. 2.23). Така пошарова організація є аналогом шаруватих структур певних відділів мозку.

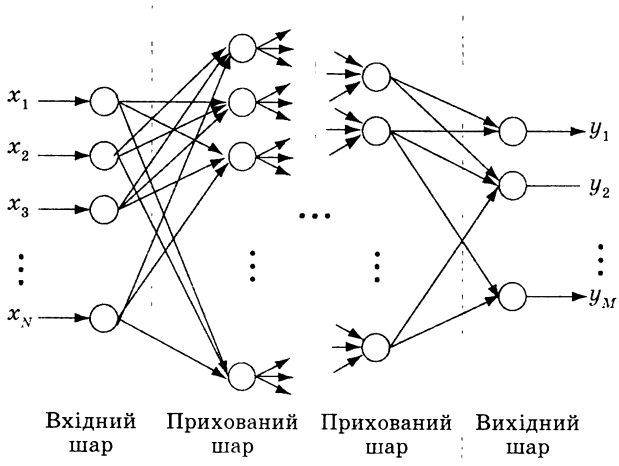


Рис. 2.23. Структура ШНМ

Зв'язки між нейронами задаються у вигляді векторів і матриць. Ваги зручно подавати елементами матриці $W = [w_{ij}]$ розмірності $N \times M$, де N — кількість входів; M — кількість нейронів. Елемент w_{ij} відбиває зв'язок між i-м й j-м нейронами. При цьому, якщо

$w_{ij} = 0$ — зв'язок між i-м й j-м нейронами відсутній;

$w_{ij} < 0$ — гальмуючий сигнал зв'язок;

$w_{ij} > 0$ — прискорювальний сигнал (збуджувальний) зв'язок.

Приклад 2.1. На рис. 2.24 зображено ШНМ і відповідну їй матрицю зв'язків.

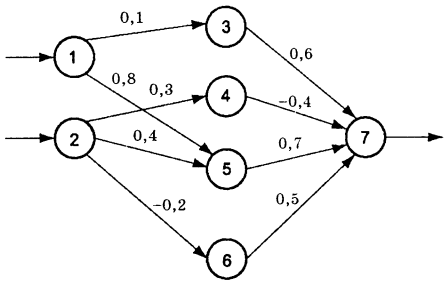


Рис. 2.24. Приклад представлення ШНМ

$$W = \begin{vmatrix} 0 & 0 & 0{,}1 & 0 & 0{,}8 & 0 & 0 \\ 0 & 0 & 0 & 0{,}3 & 0{,}4 & -0{,}2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}6 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0{,}4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0{,}5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Залежно від того, чи містять ШНМ зворотні зв'язки, чи ні, розрізнюють такі їхні топології:

- ШНМ без зворотних зв'язків (прямого поширення, *Feed forward*)
  — першого порядку;
  — другого порядку (з «*shortcut connections*»);
- ШНМ зі зворотними зв'язками (зворотного поширення, рекурентні, *Feedback*)

— з прямими зворотними зв'язками (*direct feedback*);
— з непрямими зворотними зв'язками (*indirect feedback*);
— з латеральними зв'язками (*lateral feedback*);
— повнозв'язні.

2.3.1. ШНМ прямого поширення

Дана топологія припускає наявність декількох шарів зі зв'язками між нейронами різних шарів. У мережах *першого порядку* існують тільки зв'язки між двома сусідніми шарами, тобто між i-м й $(i+1)$-м шарами. У цьому випадку говорять, що *зв'язки* ШНМ *пошарові*. Приклад такої мережі зображено на рис. 2.23. Якщо в мережі цього типу кожен нейрон шару i пов'язаний з кожним нейроном $(i+1)$-го шару, мережа називається *повнозв'язною прямого поширення*.

У мережах *другого порядку* поряд зі зв'язками між нейронами сусідніх i-го й $(i+1)$-го шарів присутні зв'язки між нейронами шарів i-го й $(i+l)$-го, де $l > 1$. Такий зв'язок називається «*shortcut*». Приклад такої ШНМ наведено на рис. 2.25.

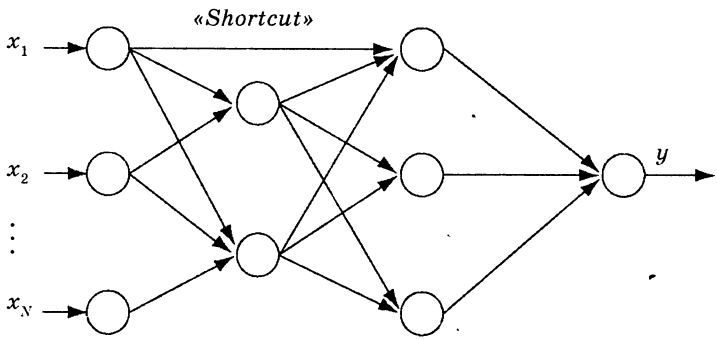


Рис. 2.25. ШНМ прямого поширення другого порядку

Зазначимо, що для мереж прямого поширення матриця зв'язків W є верхньою трикутною матрицею.

2.3.2. ШНМ зворотного поширення

Мережі цього типу припускають наявність зворотних зв'язків як між нейронами різних шарів, так і між нейронами одного шару. Використання мереж зі зворотними зв'язками необхідне у процесі

вивчення складних динамічних об'єктів, наприклад об'єктів, що змінюють свій стан при надходженні нових вхідних сигналів. Такі ШНМ можуть мати властивості, подібні до короткочасної людської пам'яті.

У ШНМ із *прямими зворотними зв'язками* (рис. 2.26) на вхід нейрона деякого i-го шару подається його вихідний сигнал, тобто даний нейрон підсилює або послаблює сигнал, перетворений його активаційною функцією, завдяки чому досягається його граничний активаційний стан.

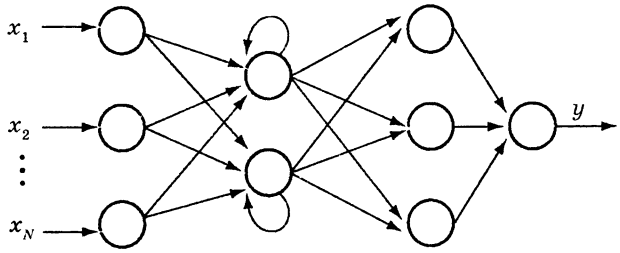


Рис. 2.26. ШНМ із прямими зворотними зв'язками

У ШНМ із *непрямими зворотними зв'язками* існують зв'язки нейрона i-го шару з нейронами $(i-k)$-го шару $k > 0$. При цьому одночасно можуть бути прямі зв'язки цього ж нейрона з нейроном $(i+l)$-го шару $(l > 0)$. Введення таких зворотних зв'язків необхідно, щоб виділити певну особливо важливу для даної ШНМ область вхідних сигналів (рис. 2.27).

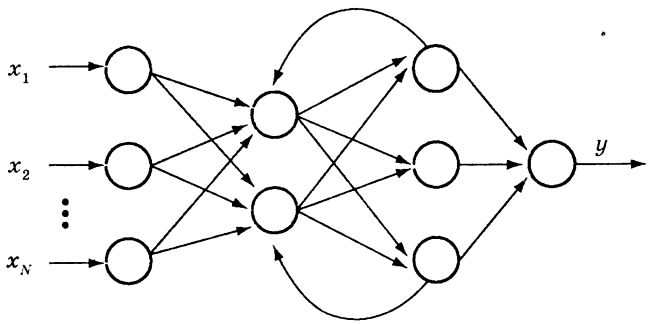


Рис. 2.27. ШНМ із непрямими зворотними зв'язками

ШНМ із *латеральними зв'язками* має зв'язки між нейронами одного шару (рис. 2.28). Такий тип зворотних зв'язків використовується у тому випадку, якщо тільки один нейрон з даної групи нейронів має бути активним. У цьому випадку на вхід кожного нейрона надходять *гальмуючий* (*ослабляючий, інгібіторний*) сигнал від інших нейронів і *звичайно збуджувальний* (*посилюючий, ексгібіторний*) сигнал власного зворотного зв'язку. Нейрон із найбільшою активністю (переможець) придушує інші нейрони. Тому цю топологію називають також топологією мережі *«переможець отримує все»* (*WTA — Net*).

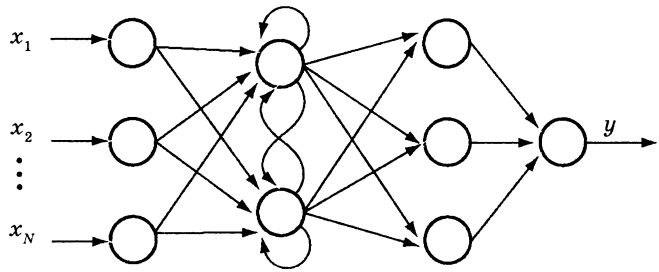


Рис. 2.28. ШНМ із латеральними зв'язками

2.3.3 Повнозв'язні ШНМ

Повнозв'язні ШНМ характеризуються наявністю зв'язків між усіма нейронами мережі (рис. 2.29). Цей вид топології відомий також як мережа Гопфілда.

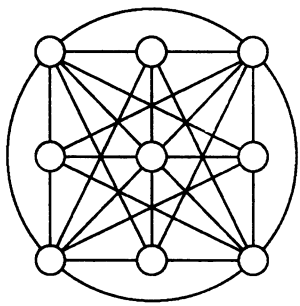


Рис. 2.29. Повнозв'язна ШНМ

Особливістю даної топології є те, що матриця зв'язків W має бути симетричною з нульовими діагональними елементами.

2.4. Навчання ШНМ

Характерною властивістю ШНМ є її здатність до навчання, що полягає у виробленні правильної реакції на подані їй різні вхідні сигнали. Існують такі можливості навчання ШНМ:

— зміна конфігурації мережі шляхом утворення нових або виключення деяких існуючих зв'язків між нейронами;
— зміна елементів матриці зв'язку (ваг);
— зміна характеристик нейронів (виду й параметрів активаційної функції й т. д.).

Найбільшого поширення сьогодні отримав підхід, при якому структура мережі задається апріорно, а мережа навчається шляхом настроювання матриці зв'язків (вагових коефіцієнтів) W. Від того, наскільки вдало побудована ця матриця, залежить ефективність даної мережі. У цьому випадку навчання полягає у зміні за певною процедурою елементів матриці W при послідовному поданні мережі деяких векторів, що навчають.

У зв'язку з цим штучний нейрон може бути представлений у такий спосіб (рис. 2.30).

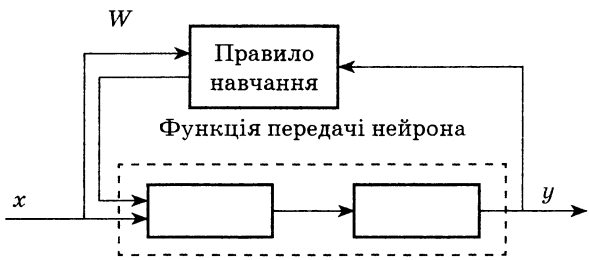


Рис. 2.30. Модель штучного нейрона

У процесі навчання ваги стають такими, що під час надходження вхідних сигналів мережа виробляє відповідні необхідні вихідні сигнали. Розрізнюють *навчання з учителем* і *без учителя*. Перший тип навчання припускає, що є «учитель», що задає пари, які навчають — для кожного вхідного вектора, що навчає, необхідний

вихід мережі. Для кожного вхідного вектора, що навчає, обчислюється вихід мережі, порівнюється з відповідно необхідним, визначається помилка виходу, на основі якої й коректуються ваги. Пари, що навчають, подаються мережі послідовно й ваги уточнюються доти, поки помилка за такими парами не досягне необхідного рівня.

Цей вид навчання неправдоподібний з біологічної точки зору. Дійсно, важко уявити зовнішнього «учителя» мозку, що порівнює реальні й необхідні реакції того, кого навчають, і коригує його поведінку (поведінку нейронів) за допомогою негативного зворотного зв'язку. Більш природним є навчання без учителя, коли мережі подаються тільки вектори вхідних сигналів, і мережа сама, використовуючи деякий алгоритм навчання, підстроювала б ваги так, щоб при поданні їй досить близьких вхідних векторів вихідні сигнали були б однаковими. У цьому випадку в процесі навчання виділяються статистичні властивості безлічі вхідних векторів, що навчають, і відбувається об'єднання близьких (подібних) векторів у класи. Подання мережі вектора з даного класу викликає її певну реакцію, яка до навчання є непередбаченою. Тому в процесі навчання виходи мережі мають трансформуватися в деяку зрозумілу форму. Це не є серйозним обмеженням, оскільки зазвичай нескладно ідентифікувати зв'язок між вхідними векторами й відповідною реакцією мережі.

Існує ще один вид навчання — *з підкріплюванням* (*reinforcement learning*), при якому також передбачається наявність учителя, що не підказує, однак, мережі правильної відповіді. Учитель тільки повідомляє, правильно чи неправильно відпрацювала мережа поданий образ. На основі цього мережа корегує свої параметри, збільшуючи значення ваг зв'язків, що правильно реагують на вхідний сигнал, і зменшуючи значення інших ваг.

Сьогодні існує велика кількість алгоритмів навчання. Деякі з них розглядатимуться пізніше, тут же коротко зупинимося на найбільш відомих.

2.4.1. Правило навчання Гебба (корелятивне, співвідносне навчання)

Більшість сучасних алгоритмів навчання виросло із *правила Гебба* [2]. Наприкінці 40-х років XX ст. років *Д. О. Гебб* теоретично встановив, що асоціативна пам'ять у біологічних системах викликається процесами, що змінюють зв'язки між нервовими клітина-

ми. Відповідно до установленого їм правила, що називається «правилом Гебба», при одночасній активації (порушенні) двох нейронів синаптична сила (вага їхнього зв'язку) зростає. Таким чином, часто використовувані зв'язки в мережі підсилюються, що пояснює феномен звички й навчання повторенням.

У ШНМ зростання синаптичної сили еквівалентне збільшенню ваги зв'язку між нейронами i та j на величину

$$\Delta w_{ij} = \gamma x_i y_j, \tag{2.35}$$

де x_i — вихід i-го й вхід j-го нейронів; y_j — вихід j-го нейрона; γ — коефіцієнт, що впливає на швидкість навчання.

У векторному вигляді це правило може бути записане в такий спосіб:

для неперервного часу $\dfrac{\partial \boldsymbol{w}}{\partial t} = \gamma \boldsymbol{x} y$, (2.36)

для дискретного часу $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma \boldsymbol{x}(k) y(k)$. (2.37)

Правило Гебба використовується у зв'язках асоціативної пам'яті, а також у деяких інших, заснованих на навчанні без учителя (без підкріплювання). У мережах *асоціативної пам'яті* приймають $\boldsymbol{y} = \boldsymbol{x}$. У *гетероасоціативних мережах* $\boldsymbol{x}$ й $\boldsymbol{y}$ в загальному випадку різняться.

Існують різні варіанти реалізації правила Гебба, наприклад засновані на мінімізації енергетичної або ентропійної функції.

Так, якщо як енергетичну вибрати функцію вигляду

$$I(\boldsymbol{w}) = -\Phi(\boldsymbol{w}^T \boldsymbol{x}) + 0{,}5\alpha \|\boldsymbol{w}\|^2, \tag{2.38}$$

де $\|\boldsymbol{w}\|^2 = \sum_{i=1}^{N} w_i^2$ — евклідова норма; $\alpha \geq 0$ — деякий коефіцієнт; $\Phi(\cdot)$ — диференційована функція така, що

$$y = \frac{d\Phi(\boldsymbol{w}^T \boldsymbol{x})}{d(\boldsymbol{w}^T \boldsymbol{x})} = f(\boldsymbol{w}^T \boldsymbol{x}),$$

то правило Гебба у векторній формі записується в такий спосіб:

$$\frac{d\boldsymbol{w}}{dt} = \gamma(y\boldsymbol{x} - \alpha \boldsymbol{w}). \tag{2.39}$$

Тут $\gamma > 0$ — деякий параметр, що впливає на швидкість навчання.

Для випадку дискретного часу правило (2.39) приймає вигляд

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma[y(k)\boldsymbol{x}(k) - \alpha \boldsymbol{w}(k)]. \tag{2.40}$$

Якщо замість фактичного значення реакції нейрона y в (2.39), (2.40) використовується необхідне значення y^*, то відповідні алгоритми

$$\frac{dw}{dt} = \gamma(y^* \boldsymbol{x} - \alpha w) \tag{2.41}$$

і

$$w(k+1) = w(k) + \gamma\left[y^*(k)\boldsymbol{x}(k) - \alpha w(k) \right] \tag{2.42}$$

називають *корелятивними правилами* навчання.

Використання як енергетичного функціоналу, що мінімізується, квадратичного вигляду

$$I(w) = 0{,}5\|e\|^2,$$

де $\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}}$ — вектор помилок; $\hat{\boldsymbol{x}}$ — оцінка вхідного вектора, призводить до *правила навчання Ойя* [66–67]. При цьому передбачається, що, по-перше, вектор ваг нормалізований й, по-друге, нейрон має лінійну активаційну функцію, тобто $y = \boldsymbol{w}^T\boldsymbol{x}$. Оцінка вектора вхідного сигналу є зваженим вектором ваг

$$\hat{\boldsymbol{x}} = wy. \tag{2.43}$$

Саме ж правило навчання Ойя записується у такий спосіб:

— для неперервного часу

$$\frac{dw}{dt} = \gamma(\boldsymbol{x}y - \boldsymbol{w}y^2), \tag{2.44}$$

— для дискретного часу

$$w(k+1) = w(k) + \gamma y(k)\left[\boldsymbol{x}(k) - w(k)y(k)\right]. \tag{2.45}$$

Під час використання цього правила виникають деякі проблеми, про які йтиметься згодом. Слід лише зазначити, що це правило досить активно використовувалося в мережах, однак за останні роки виникло багато більш ефективних алгоритмів, що забезпечують значно вищу швидкість навчання.

2.4.2. Дельта-правило

Це важливе правило навчання було запропоновано *Б. Уідроу* й *М. Е. Гоффом* [6] і найбільше відповідає одношаровим ШНМ прямого поширення. Ідея його полягає в тому, що якщо під час навчання мережі можна встановити розбіжність між її бажаною й наявною реакціями, ця розбіжність може бути усунута або змен-

шена шляхом зміни певним чином вагових коефіцієнтів зв'язку. Для цього й використовується *дельта-правило*, відповідно до якого зміна ваги зв'язку між i-м й j-м нейронами визначається у такий спосіб:

$$\Delta w_{ij} = \gamma x_i (y_j^* - y_j), \tag{2.46}$$

де x_i — вихід попереднього i-го нейрона; y_j^*, y_j — бажана й реальна реакції j-го нейрона відповідно; γ — коефіцієнт, що впливає на швидкість навчання.

З (2.46) видно, що якщо різниця $(y_j^* - y_j)$ мала, тобто реакція j-го нейрона незначною мірою відрізняється від бажаної, зміна ваги зв'язку між цими нейронами також буде незначною.

2.4.3. Розширене дельта-правило

Розширене дельта-правило, на відміну від простого, знаходить застосування у багатошарових ШНМ прямого поширення другого порядку. Більш докладно це розглядатиметься нижче. Тут же зазначимо, що відповідно до цього правила зміна ваг здійснюється в такий спосіб:

$$\Delta w_{ij} = \gamma x_i \delta_j, \tag{2.47}$$

де $\delta_j = \begin{cases} f_j'(\boldsymbol{x}, \boldsymbol{w})(y_j^* - y_j), & \text{якщо } j\text{-нейрон вихідного шару;} \\ f_j'(\boldsymbol{x}, \boldsymbol{w}) \sum_m \delta_m w_{im} & \text{в іншому випадку;} \end{cases}$

$f_j'(\boldsymbol{x}, \boldsymbol{w})$ — похідна активаційної функції, використовувана в j-му нейроні; індекс j використовується для позначення всіх нейронів наступних шарів, пов'язаних з нейроном j.

2.4.4. Конкурентне навчання

У цьому виді навчання всі нейрони одного шару є конкуруючими, а перевага віддається тому нейрону, що найбільш сильно реагує на вхідний (дратівний) сигнал, тобто настроюються ваги тільки одного нейрона — *нейрона-переможця* (*winner-takes-all*).

Правило ж настроювання ваг звичайно є деякою модифікацією правила Гебба. Початковим значенням вагових коефіцієнтів привласнюються малі, відмінні від нуля й різні значення. При поданні образу, що навчає, реакція одного з нейронів буде найбільш сильною. Його вагові коефіцієнти підсилюються або змінюються таким чином, щоб найповніше відповідати поданому образу, ваги ж інших

(неактивних) нейронів або не змінюються, або зменшуються. Процес навчання завершується, коли вага активного нейрона дорівнюватиме загальній сумі ваг нейронів одного шару.

Реалізація цього виду навчання багатоваріантна.

2.4.5. Стохастичне навчання

Розглянуті вище методи є *детерміністськими*, у яких на кожному такті за певним алгоритмом відбувається корекція ваг, заснована на використанні значень вхідних і вихідних бажаних і фактичних сигналів.

Стохастичні методи навчання базуються на псевдовипадкових змінах ваг зі збереженням тих змін, які ведуть до поліпшень. Для корекції ваг використовується деяка імовірнісна функція. Більші первинні і випадкові корекції зі збереженням певних змін ваг поступово зменшують, досягаючи при цьому мети навчання. Це нагадує процес відпалювання металу, коли атоми розплавленого металу, які перебувають у хаотичному русі при його поступовому охолодженні, гублячи енергію, досягають нижчого з можливих енергетичних станів (глобального мінімуму). Тому для опису цього виду навчання часто використовують термін *«імітація відпалювання»* (*simulated annealing*).

Таке стохастичне навчання використовується в машинах Больцмана й Коші.

2.4.6. Градієнтні методи навчання

Багато методів навчання засновано на мінімізації деякої цільової (вартісної, енергетичної й т. д.) функції I, що являє собою звичайно деяку опуклу функцію. Якщо використовувані функції активації $f(\cdot)$ диференційовані, зручно застосовувати *градієнтні методи мінімізації*. У цьому випадку корекція ваг зв'язку між i-м й j-м нейронами відбувається за правилом

$$\Delta w_{ij} = -\gamma \nabla_w I(\boldsymbol{w}), \tag{2.48}$$

де γ — коефіцієнт, що впливає на швидкість навчання;

$$\nabla_w I(\boldsymbol{w}) = \frac{\partial I(\boldsymbol{w})}{\partial w_{ij}}.$$

Ці методи найчастіше використовуються при контрольованому навчанні, коли відома необхідна реакція нейронів y^*.

Більшість градієнтних методів засновано на мінімізації квадратичного функціонала

$$I(\boldsymbol{w}) = 0{,}5e^2(k) = 0{,}5(y^*(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k))^2. \qquad (2.49)$$

У цьому випадку

$$\nabla_w I(\boldsymbol{w}) = -e(k)\boldsymbol{x}(k). \qquad (2.50)$$

Підстановка (2.50) у (2.48) призводить до *алгоритму методу найменших квадратів* (*МНК*)

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma e(k)\boldsymbol{x}(k). \qquad (2.51)$$

Градієнтні методи застосовуються також і для навчання стохастичних нейронів, функціонування яких описується формулою (2.34). У цьому випадку замість критерію (2.49) мінімізується критерій

$$I(\boldsymbol{w}) = 0{,}5(y^*(k) - < y(k) >)^2,$$

де $< y(k) > = (+1)P(z(k) \mid y(k) = 1) + (-1)P(z(k) \mid y(k) = -1)$ — математичне сподівання виходу мережі.

Якщо $P(z(k) \mid y(k)) = (1 + \exp(-2\alpha z(k))^{-1}$, то, як було показано в підрозділі 2.2. 8, $< y(k) > = \tanh(\alpha z(k))$ і (*див.* (2.14))

$$\nabla_w I(\boldsymbol{w}) = \alpha(y^*(k) - < y(k) >)(1 - < y^2(k) >)\boldsymbol{x}(k),$$

що призводить до градієнтного алгоритму

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma\alpha(y^*(k) - < y(k) >)(1 - < y^2(k) >))\boldsymbol{x}(k). \qquad (2.52)$$

Як видно, цей алгоритм не відрізняється від градієнтного алгоритму навчання детермінованого нейрона, що має активаційну функцію $f(z) = \tanh(z)$.

Очевидно, що властивості (2.51) істотно залежать від вибору коефіцієнта γ. Існують різні рекомендації з вибору γ. Так, у теорії стохастичної апроксимації, що вивчає особливості роботи алгоритмів такого типу за наявності завад вимірів, цей коефіцієнт вибирається змінним і таким, що задовольняє умовам Дворецького [68–70]

$$\lim_{k\to\infty}\gamma(k) = 0, \quad \sum_{k=1}^{\infty}\gamma(k) = \infty, \quad \sum_{k=1}^{\infty}\gamma^2(k) < \infty, \qquad (2.53)$$

зміст яких полягає в тому, що для забезпечення збіжності послідовності (2.51) у деяку точку $\boldsymbol{w}^*$ довжина кроку $\gamma(k)$ має з одного боку спадати досить повільно (для забезпечення власне збіжності),

а з іншого — досить швидко (з метою придушення завад). Таким умовам відповідає, наприклад, гармонійний ряд $\gamma(k) = \gamma_0 k^{-\alpha}$, де γ_0 — деяка константа, $0{,}5 < \alpha \le 1$. У теорії стохастичної апроксимації немає рекомендацій з вибору константи γ_0, крім її позитивності. Теорія оптимальної фільтрації, тісно пов'язана з теорією стохастичної апроксимації, дозволяє вибрати цю константу, що виявляється залежною від статистичних характеристик сигналів і завад. Зокрема, значення γ_0 може бути обране з умови

$$0 < \gamma < \frac{2}{\lambda_{\max}}$$

або з умови

$$0 < \gamma < \frac{2}{tr[R_x]},$$

де $R_x = M\{\boldsymbol{x}\boldsymbol{x}^T\}$ — коваріаційна матриця; $M\{\cdot\}$ — символ математичного сподівання; $tr[R_x]$ — слід матриці R_x

$$tr[R_x] = \sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} r_{i,xx} \ge \lambda_{\max};$$

λ_i — власні числа матриці R_x; $r_{i,xx}$ — діагональні елементи коваріаційної матриці R_x; $\lambda_{\max}$ — найбільш власне число R_x.

Вибір коефіцієнта γ у вигляді

$$\gamma(k) = \left\|\boldsymbol{x}(k)\right\|^{-2}$$

призводить до *алгоритму Качмажа*, відомому в теорії ШНМ як *алгоритм Уідроу — Гоффа* [6]. У теорії ж оцінювання цей алгоритм називають *нормалізованим алгоритмом МНК*. Більш докладно цей алгоритм розглядатиметься нижче.

Найбільш відомим є *рекурентний алгоритм МНК* (*РМНК*), що виходить із (2.51) при виборі замість скалярного коефіцієнта $\gamma(k)$ матричного змінного коефіцієнта $\Gamma(k)$

$$\Gamma(k) = P(k)\,[\lambda + \boldsymbol{x}^T(k)P(k)\boldsymbol{x}(k)]^{-1}, \qquad (2.54)$$

де $P^{-1}(k) = \sum_{i=1}^{k} \lambda^{k-i}\boldsymbol{x}(i)\boldsymbol{x}^T(i)$ — коваріаційна матриця; $0 < \lambda \le 1$ — параметр зважування інформації.

Матриця $P(k)$ допускає рекурентне обчислення. Тому в остаточному вигляді *РМНК із експонентним зважуванням інформації* має вигляд

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \alpha(k)P(k)\,\boldsymbol{x}(k)e(k); \tag{2.55}$$

$$P(k+1) = \frac{1}{\lambda}\Big[I - \alpha(k)P(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\Big]P(k), \tag{2.56}$$

де $\alpha(k) = [\lambda + \boldsymbol{x}^T(k)P(k)\boldsymbol{x}(k)]^{-1}$; I — одинична матриця.

Використовуваний у (2.55), (2.56) параметр λ забезпечує зважування інформації, тобто надання більшого значення інформації, що знову надходить.

Даний механізм оцінювання важливості інформації особливо ефективний під час настроювання ваг, що змінюються в часі (дослідження нестаціонарних об'єктів). При $\lambda = 1$ алгоритм (2.55), (2.56) переходить у РМНК.

Різновидом РМНК є *багатокроковий* (*l-кроковий*) *проекційний алгоритм*, що використовує, на відміну від (2.55–2.56), фіксовану кількість інформації та має вигляд [71]:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma(k)\boldsymbol{r}_l(k); \tag{2.57}$$

$$\begin{aligned}\boldsymbol{r}_l(k) &= R_{l-1}(k-1)\boldsymbol{x}(k)\Gamma^{-1}(k)e(k) + \\ &+\big(1-\gamma(k-1)\big)\big(I - R_l(k-1)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\Gamma^{-1}(k)\big)\boldsymbol{r}_{l-1}(k-1);\end{aligned} \tag{2.58}$$

$$\begin{aligned}R_i(k-l+i) &= \big(I - R_{i-1}(k-l+i-1)\boldsymbol{x}(k-l+i)\times \\ &\times \boldsymbol{x}^T(k-l+i)\Gamma^{-1}(k-l+i)\big)R_{i-1}(k-l+i-1),\end{aligned} \tag{2.59}$$

де $r_0 = (0\ \ 0\ \ \ldots\ \ 0)^T$; $R_0 = I$; $i = \overline{1,l}$; l = const — пам'ять алгоритму (ця величина вибирається меншою, ніж розмірність завдання); $\gamma(i) \in (0, 2)$.

Використання в (2.57–2.59) фіксованої пам'яті робить цей алгоритм особливо привабливим у процесі навчання мереж, параметри яких змінюються в часі.

Якщо нейрони описуються нелінійними активаційними функціями $f(\boldsymbol{w},\boldsymbol{x}(k))$, то, скориставшись розкладанням вихідних сигналів нейронів у ряд Тейлора

$$y(k+1) = f(\boldsymbol{w}(k),\boldsymbol{x}(k+1)) + H^T(k+1)\big(\boldsymbol{w}^* - \boldsymbol{w}(k)\big) + \rho(k+1), \tag{2.60}$$

де $H(k+1)$ — матриця розмірності $S \times M$ вигляду

$$H(k+1) = \left.\frac{\partial f(\boldsymbol{w},\boldsymbol{x}(k+1)}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}(k)};$$

S — розмірність вектора ваг; M — розмірність вихідного сигналу; $\rho(k)$ — залишок розкладання, що враховує члени більш високих порядків, можна отримати такий алгоритм навчання:

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+K(k+1)\left(y^{*}(k+1)-f(\boldsymbol{w}(k),\boldsymbol{x}(k+1))\right); \quad (2.61)$$

$$K(k+1)=\frac{1}{\lambda}P(k)H(k+1)\left[I+\frac{1}{\lambda}H^{T}(k+1)P(k)H(k+1)\right]^{-1}; \quad (2.62)$$

$$P(k+1)=\frac{1}{\lambda}\left[I-K(k+1)H^{T}(k+1)\right]^{-1}, \quad (2.63)$$

відомий у теорії оцінювання як *фільтр Калмана*. Матриця $K(k+1)$, що обчислюється відповідно до (2.62), називається *матрицею посилення Калмана*.

Алгоритм навчання (2.55–2.56) випливає з (2.61–2.63) у процесі використання лінійної апроксимації активаційної функції нейронів.

Якщо вихідний сигнал i-го нейрона подати у вигляді

$$y_i(k+1)=f(\boldsymbol{w}_i^{T}(k)\boldsymbol{x}_i(k+1)), \quad (2.64)$$

то може бути отриманий такий алгоритм навчання його ваг, що заснований на *розширеному фільтрі Калмана* [72]:

$$\boldsymbol{w}_i(k+1)=\boldsymbol{w}_i(k)+K_i(k+1)e_i(k+1); \quad (2.65)$$

$$K_i(k+1)=\frac{1}{\lambda}P_i(k)Z_i(k+1)\left[I+\frac{1}{\lambda}Z_i^{T}(k+1)P_i(k)Z_i(k+1)\right]^{-1}; \quad (2.66)$$

$$P_i(k+1)=\frac{1}{\lambda}\left[I-K_i(k+1)Z_i^{T}(k+1)\right]P_i(k+1), \quad (2.67)$$

де $Z_i(k+1)=x_i(k+1)\nabla_{w}f\left(\boldsymbol{w}_i^{T}(k)x_i(k+1)\right)$; $\nabla_{w}f(\cdot)=\dfrac{\partial f(\cdot)}{\partial \boldsymbol{w}}$.

Наведені вище алгоритми навчання реалізують методи *оптимізації першого порядку*, у яких використовується обчислення градієнта функціонала. Серед цих методів найбільшу швидкість збіжності має *метод сполучених градієнтів* [73–74].

Ще більшу швидкість збіжності мають *методи другого порядку*, що вимагають обчислення других похідних мінімізованого

функціонала. Серед таких методів у першу чергу слід відзначити *метод Ньютона*:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - H_k^{-1} g_k, \tag{2.68}$$

де H_k^{-1} — матриця, зворотна матриці Гессе

$$H_k = \nabla_w^2 I(\boldsymbol{w}) = \begin{bmatrix} \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_1^2} & \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_1 w_2} & \cdots & \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_1 w_N} \\ \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_2 w_1} & \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_2^2} & \cdots & \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_2 w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_N w_1} & \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_N w_2} & \cdots & \dfrac{\partial^2 I(\boldsymbol{w})}{\partial w_N^2} \end{bmatrix}, \tag{2.69}$$

$$g_k = \left(\frac{\partial I(\boldsymbol{w})}{\partial w_1} \quad \frac{\partial I(\boldsymbol{w})}{\partial w_2} \quad \cdots \quad \frac{\partial I(\boldsymbol{w})}{\partial w_N} \right)^T. \tag{2.70}$$

Ці похідні обчислюються у точці $\boldsymbol{w} = \boldsymbol{w}(k)$.

Якщо як критерій навчання використовується

$$I(\boldsymbol{w}) = 0{,}5 \sum_{i=1}^{M} \left(y_i^* - y_i \right)^2 = 0{,}5 \sum_{i=1}^{M} e_i^2, \tag{2.71}$$

то

$$g_k = \frac{\partial I(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0{,}5 \left[\frac{\partial \sum_{i=1}^{M} e_i^2}{\partial \boldsymbol{w}_1} \quad \cdots \quad \frac{\partial \sum_{i=1}^{M} e_i^2}{\partial \boldsymbol{w}_N} \right]^T =$$

$$= \left[\sum_{i=1}^{M} e_i \frac{\partial e_i}{\partial \boldsymbol{w}_1} \quad \cdots \quad \sum_{i=1}^{M} e_i \frac{\partial e_i}{\partial \boldsymbol{w}_N} \right]^T = J^T \boldsymbol{e}, \tag{2.72}$$

де J — якобіан, що має вигляд

$$J = \begin{bmatrix} \dfrac{\partial e_1}{\partial w_1} & \dfrac{\partial e_1}{\partial w_2} & \cdots & \dfrac{\partial e_1}{\partial w_N} \\ \dfrac{\partial e_2}{\partial w_1} & \dfrac{\partial e_2}{\partial w_2} & \cdots & \dfrac{\partial e_2}{\partial w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial e_M}{\partial w_1} & \dfrac{\partial e_M}{\partial w_2} & \cdots & \dfrac{\partial e_M}{\partial w_N} \end{bmatrix}. \tag{2.73}$$

Оскільки елементи матриці $H = \nabla_w^2 I(\boldsymbol{w})$ визначаються за формулою

$$[H]_{l,m} = \left[\nabla_w^2 I(\boldsymbol{w})\right]_{l,m} = \frac{\partial^2 I(\boldsymbol{w})}{\partial w_l \partial w_m} = \sum_{i=1}^{M}\left(\frac{\partial e_i}{\partial w_l}\frac{\partial e_i}{\partial w_m} + e_i \frac{\partial^2 e_i}{\partial w_l \partial w_m}\right),$$

то з урахуванням (2.69), (2.73) можна записати

$$H = J^T J + S,$$

де $S = \sum_{i=1}^{M} e_i \nabla_w^2 e_i$.

З наближенням до мінімуму елементи матриці S стають малими, і матриця Гессе може бути апроксимована

$$H \approx J^T J. \qquad (2.74)$$

Підставляючи (2.74) у (2.68), отримуємо такий алгоритм *методу Ньютона*:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \left[J_k^T J_k\right]^{-1} J_k^T \boldsymbol{e}_k, \qquad (2.75)$$

де $\boldsymbol{e}_k = (e_1, e_2, \ldots, e_m)^T$; індекс $k = 0, 1, 2, \ldots$ означає прискорений машинний час.

Для підвищення обчислювальної стійкості алгоритму навчання замість (2.74) використовують апроксимацію

$$H \approx J^T J + \mu I, \qquad (2.76)$$

де μ — досить мале додатне число.

Підстановка (2.76) в (2.68) призводить до *алгоритму Левенберга — Маркуардта*

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \left[J_k^T J_k + \mu I\right]^{-1} J_k^T \boldsymbol{e}_k. \qquad (2.77)$$

Ще один метод, що використовує обчислення градієнта й широко застосовуваний у процесі навчання багатошарових ШНМ — *метод зворотного поширення помилки* (*backpropagation*) — буде розглянутий у наступному розділі. Зазначимо тільки, що дельта-правило є оптимальним у певному сенсі для одношарових ШНМ, тому що з його допомогою визначається вагова матриця, що точно відтворює реакцію нейрона у випадку лінійно незалежних вхідних векторів. У багатошарових ШНМ це правило не застосовується, оскільки не зрозуміло, яким чином настроювати вагові коефіцієнти нейронів прихованих шарів, які б зменшували помилку вихідного сигналу мережі. Ця проблема називається «*credit assignment-проблеми*» і вирішується за допомогою алгоритму зворотного поши-

рення помилки. Цей алгоритм може бути застосований до ШНМ, що має будь-яке число прихованих шарів. Відповідно до цього алгоритму шляхом мінімізації вихідної помилки спочатку настроюються ваги останнього (вихідного) шару, потім попереднього й т. д. до вхідного шару. Однак, якщо дельта-правило гарантує збіжність процесу навчання, то алгоритм зворотного поширення помилки такої гарантії не дає, і навчання може застрягнути в одному з локальних мінімумів.

2.4.7. Навчання з підкріплюванням

Цей вид навчання також заснований на оптимізації деякого критерію. Крім того, у випадку навчання з підкріплюванням, як ми вже зазначали раніше, також є присутнім учитель або зовнішній арбітр, що дає лише вказівку про зменшення або збільшення значень параметрів мережі. Оскільки для забезпечення адаптації мережі у всьому заданому просторі сигналів необхідні випадкові її вихідні сигнали, то для настроювання мережі звичайно використовують алгоритми, застосовувані для навчання стохастичних нейронів.

Так, у роботі [75] запропоновано алгоритм навчання з підкріплюванням, аналогічний (2.52) і що використовує для настроювання мережі послідовність навчальних пар вигляду $\{\boldsymbol{x}(k), r(k)\}$, де $\boldsymbol{x}(k)$ — вектори вихідних сигналів, а $r(k) \in \{-1,1\}$ — вказівки вчителя ($r(k) = 1$ свідчить про правильний напрямок у зміні ваг і про необхідність збільшення їхнього значення; $r(k) = -1$ сигналізує про помилковий напрямок і про необхідність зменшення значення ваг). Крім того, у цьому алгоритмі, на відміну від (2.52), використовується змінний коефіцієнт навчання $\alpha(r(k))$

$$w(k+1) = w(k) + \alpha(r(k))[y^*(k) - < y(k) >][1 - < y^2(k) >]\boldsymbol{x}(k), \quad (2.78)$$

де

$$\alpha(r(k)) = \begin{cases} \alpha^+ \\ \alpha^-, \end{cases} \quad y^*(k) = \begin{cases} y(k) & \text{при } r(k) = +1 \quad \text{(заохочування)} \\ -y(k) & \text{при } r(k) = -1 \quad \text{(штраф)}, \end{cases}$$

$\alpha^+ >> \alpha^- > 0$.

Вибір $\alpha^+ >> \alpha^-$ служить для забезпечення швидкого навчання й повільного забування.

У роботі [76] запропоновано модифікацію алгоритму (2.78) такого вигляду:

$$w(k+1) = w(k) + \alpha(r(k))[r(k)(y^*(k) - < y(k) >) - \\ - (1 - r(k))(y(k) + < y^2(k) >)]x(k), \tag{2.79}$$

де $r(k) \in [0,1]$.

На завершення зазначимо, що хоча цей вид навчання й досліджено у ряді робіт, він не належить до числа популярних. Тому надалі ми розглядатимемо перші два види навчання: навчання з учителем і навчання без учителя.

Докладніше методи й алгоритми навчання висвітлено в монографії [77].

Контрольні запитання та завдання

1. Яка структура штучного нейрона?
2. Наведіть приклади вхідних операторів.
3. Що таке функція активації? Наведіть приклади різних функцій активації.
4. Дайте характеристику нейрона Маккаллоха — Піттса.
5. У чому особливість нейрона Фукушіми?
6. Поясніть принцип функціонування нейрона Гопфілда.
7. Поясніть особливості нейрона Ґроссберґа.
8. Наведіть приклади топологій ШНМ.
9. Які існують підходи до навчання ШНМ?
10. Поясніть правило навчання Гебба.
11. На чому засноване дельта-правило?
12. У чому суть конкурентного навчання?
13. Що є основою стохастичного навчання?
14. Наведіть приклади градієнтних методів навчання.

3. РАННІ АРХІТЕКТУРИ ШНМ

Перші ШНМ, розроблені у 50–60-х роках XX ст., мали просту одношарову архітектуру й могли вирішувати досить обмежене коло задач. Однак результати досліджень властивостей цих мереж виявилися настільки цікавими, що викликали цілий потік робіт, присвячених створенню мереж, які мають більш складну структуру й здатні вирішувати значно складніші задачі.

3.1. Одношарові ШНМ

3.1.1. Одношаровий персептрон

Термін «персептронний нейрон» введено у роботі *У. Маккалоха* й *У. Піттса* [1]. У більшій частині їхньої роботи використовувалася модель (2.17) з активаційною функцією вигляду (2.10). Із досягненням зваженою сумою значення, більшого заданого порога θ, на виході нейрона з'являвся одиничний сигнал, якщо ж зважена сума була менше θ, сигнал був відсутній. Системи, що використовують дані моделі нейронів, отримали назву *персептронів*.

Значний інтерес до персептронів викликаний роботою *Ф. Розенблатта* [4], у якій він досліджував нейромережеву модель сітківки (RETINA) — *фотоперсептрон*. Згодом такий підхід широко використовувався для моделювання обробки оптичних сигналів. Фотоперсептрон зображений на рис. 3.1 і складається, відповідно до концепції Розенблатта, із трьох шарів, що послідовно здійснюють попередню обробку (розбивання) образу, оцінку його характеристик і розпізнавання:

— сітківка (RETINA);
— асоціативний шар;
— вихідний шар.

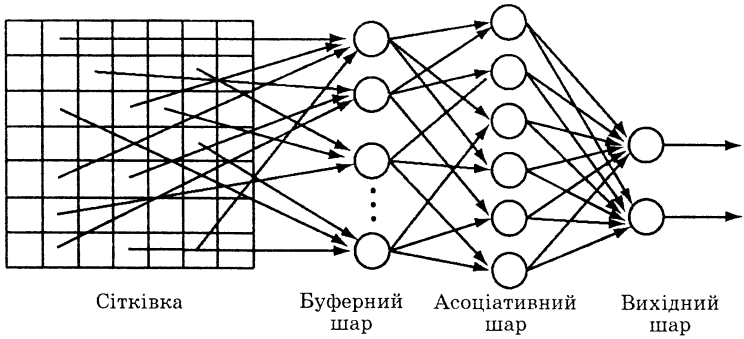


Рис. 3.1. Персептрон Розенблатта

Попередня обробка образу не залежить від його виду. Однак результат цієї обробки має забезпечити можливість розпізнавання образів на основі аналізу їхніх характеристик. Нарешті, вихідний шар (класифікатор) аналізує характеристики знову пропонованого образу й установлює його відповідність одному з раніше поданих.

Сигнали першого шару, сітківки, подані у двійковій формі, надходять на асоціативний шар, причому в загальному випадку не всі нейрони першого шару пов'язані з усіма нейронами другого шару. При встановленні цих зв'язків виникає можливість структуризації вхідних даних, тобто виділення й об'єднання в так звані *рецептивні поля* найбільш важливих ознак (областей).

У зв'язку з цим під рецептивним полем розуміють множину всіх нейронів вхідного шару, пов'язаних з одним нейроном асоціативного шару. Зв'язки між нейронами асоціативного й вихідного шарів варіабельні й можуть модифікуватися шляхом зміни вагових коефіцієнтів.

Нейрони асоціативного шару мають лінійні активаційні функції, тому під час надходження із сітківки вхідних (дратівних) сигналів вони посилають імпульси на вихідний шар, де й відбувається додавання зважених імпульсів. У вихідному шарі використовуються уні- або біполярна активаційні функції (2.10) або (2.11). Якщо сума зважених імпульсів перевищує деяке задане порогове значення, виробляється одиничний вихідний сигнал, якщо не перевищує — нульовий (для уніполярної) або –1 (для біполярної функції активації). У зв'язку з цим персептрон може розглядатися як двошарова ШНМ прямого поширення.

Можливості одношарового персептрона можна проаналізувати на прикладі реалізації булевих функцій двох змінних (табл. 3.1).

Таблиця 3.1

x_1	x_2	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Відповідний спрощений персептрон зображено на рис. 3.2.

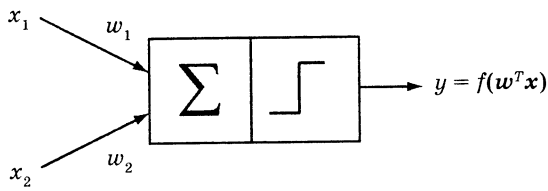


Рис. 3.2. Персептрон:
w_1 й w_2 — вагові коефіцієнти; $\theta > 0$ — поріг вихідного нейрона

Приклад 3.1. Для реалізації функції «І» необхідне виконання таких нерівностей:

Якщо $x_1 = 0, x_2 = 0$, $w_1x_1 + w_2x_2 = 0 < \theta$,

$x_1 = 0, x_2 = 1$, $w_1x_1 + w_2x_2 = w_2 < \theta$,

$x_1 = 1, x_2 = 0$, $w_1x_1 + w_2x_2 = w_1 < \theta$,

$x_1 = 1, x_2 = 1$, $w_1x_1 + w_2x_2 = w_1 + w_2 \geq \theta$.

Як видно з наведених співвідношень, існує нескінченна множина значень w_1 й w_2, при яких дані нерівності виконуються.

Особливістю персептрона є здатність до навчання його вагових коефіцієнтів. У режимі навчання йому подають образи — пари, що навчають $(\boldsymbol{x}, \boldsymbol{y}^*)$, на основі яких він настроює свої параметри так, щоб з появою деякого вхідного вектора $\boldsymbol{x}$ на його виходах з'являлися відповідні цьому входу сигнали $\boldsymbol{y}$. Процес навчання завершується, якщо всі пари $(\boldsymbol{x}, \boldsymbol{y}^*)$ асоціюються правильно. Алгоритм навчання персептрона реалізує спосіб навчання з учителем, при якому вихідна помилка мінімізується. Досліджуючи клас самонастроювальних мереж, правильніше мереж, що навчаються без учителя, Ф. Розенблатт довів, що за певних умов алгоритм навчання завжди збігається. Однак виявилося, що персептрони не здатні

навчатися вирішенню низки простих задач. Точний доказ цього факту був здійснений *М. Мінські* [8].

Найбільш вражаюча обмеженість найпростішого персептрона виявляється при спробі реалізації функції «що виключає АБО» (функції y_6 у табл. 3.1).

Приклад 3.2. Застосування персептрона, зображеного на рис. 3.2, для реалізації функції y_6 вимагає виконання таких умов:

$$\begin{aligned} x_1 = 0,\ x_2 = 0, &\qquad w_1x_1 + w_2x_2 = 0 < \theta, \\ x_1 = 0,\ x_2 = 1, &\qquad w_1x_1 + w_2x_2 = w_2 \geq \theta, \\ x_1 = 1,\ x_2 = 0, &\qquad w_1x_1 + w_2x_2 = w_1 \geq \theta, \\ x_1 = 1,\ x_2 = 1, &\qquad w_1x_1 + w_2x_2 = w_1 + w_2 < \theta. \end{aligned}$$

З перших трьох нерівностей випливає, що необхідно задати $\theta > 0$. Однак у цьому випадку отримуємо

$$w_1 + w_2 \geq 2\theta > \theta,$$

тобто остання умова не виконується. Отже, на найпростішому персептроні реалізувати функцію «що виключає АБО» неможливо.

Обмежені можливості найпростішого персептрона, виявлені й доведені Мінські, призвели до того, що дослідження в цій галузі на певний час припинилися. Це можна пояснити ще й тим, що в той час хоча й розуміли необхідність переходу до багатошарових персептронів, методи їхнього навчання ще не були відомі.

3.1.2. Навчання персептрона

Існують різні шляхи реалізації процесу навчання персептрона, однак у їхній основі лежить таке правило: вагові коефіцієнти персептрона змінюються тільки тоді, коли виникає розбіжність між його фактичною й бажаною реакціями.

Схему навчання персептрона наведено на рис. 3.3.

Передбачається, що активаційна функція має вигляд (2.11). Процес навчання полягає в послідовному поданні множини пар, що навчають $(\boldsymbol{x}_p, y_p^*)$, $p = \overline{1,P}$, де $\boldsymbol{x}_p$, y_p^* — $(N \times 1)$-вхідний вектор і бажаний вихідний сигнал p-ї пари, що навчає, відповідно, за допомогою яких визначається необхідний вектор вагових коефіцієнтів $\boldsymbol{w}^*$ такий, що

$$y_p = \operatorname{sgn}(w^{*T}\boldsymbol{x}_p) = y_p^*,\ p = \overline{1,P}. \tag{3.1}$$

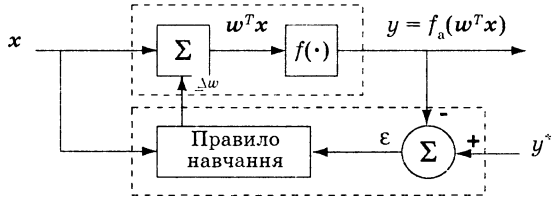


Рис. 3.3. Схема навчання персептрона

У цьому випадку вектор w^* забезпечить правильну класифікацію персептроном усіх пар, що навчають, із поданої множини (завершується цикл навчання).

З (3.1) випливає, що гіперплощина $w^{*T}x_p = 0$ ділить вхідний простір на два підпростори. Для $y_p^* = 1$ має виконуватися умова $w^{*T}x_p > 0$, а для $y_p^* = -1$ — умова $w^{*T}x_p < 0$.

Алгоритм навчання може бути записаний у такий спосіб:

$$w_{p+1} = w_p + \gamma e_p x_p, \tag{3.2}$$

де $e_p = y_p^* - y_p$ — помилка класифікації; γ — параметр, що впливає на швидкість збіжності алгоритму (тривалість процесу навчання). Корекція ваг відповідно до (3.2) може відбуватися в режимах *online* й *offline*.

У режимі *online* корекція відбувається при поданні кожної навчальної пари $(x_p, y_p^*), p = \overline{1,P}$.

У режимі *offline* у M-му циклі (епосі) навчання подаються всі пари (x_p, y_p^*) й обчислюється середнє значення помилки класифікації

$$\overline{e}_M = \frac{1}{P}\sum_{p=1}^{P}\left(y_{P,M}^* - y_{P,M}\right), \tag{3.3}$$

що й використовується в алгоритмі навчання. Тут $y_{P,M}^*, y_{P,M}$ — бажані й реальні вихідні сигнали при пред'явленні p-ї пари, що навчає, в M-му циклі навчання відповідно.

Алгоритм навчання в M-му циклі приймає вигляд

$$w_{M+1} = w_M + \gamma \overline{e}_M x_P . \tag{3.4}$$

Реалізація даного навчання пов'язана з необхідністю попереднього обчислення значень вихідних змінних для всіх пар, що навчають.

3.1.3. Теорема збіжності для персептрона

Ця теорема, доведена Розенблаттом, стверджує, що якщо задача має розв'язок, то він може бути отриманий за кінцеве число тактів (ітерацій). З погляду класифікації це означає, що персептрон може навчитися правильно класифікувати подані йому образи на кінцевому числі пар, що навчають.

Це може бути показано в такий спосіб.

Нехай $y_k^* \in \{0,1\}$ і вхідні сигнали обмежені за величиною, тобто

$$\|x(k)\|^2 \le A, \tag{3.5}$$

де $\|x(k)\|^2 = x^T(k)x(k)$ — евклідова норма; k — деякий такт навчання.

Використовувана в алгоритмі (3.2) помилка $e = y^* - y$ може приймати значення ± 1 або 0. При $e = 0$ (правильна класифікація) зміни ваг не відбувається. Позначимо розв'язок: вектор ваг, що може правильно класифікувати всі пропоновані образи, через w^*. Для цього вектора ваг маємо

$$\begin{aligned} w^{*T} w(k) > \delta > 0, &\quad \text{якщо } y^*(k) = 1; \\ w^{*T} w(k) < -\delta < 0, &\quad \text{якщо } y^*(k) = 0. \end{aligned} \tag{3.6}$$

Помноживши обидві частини (3.2) зліва на $w^T(k+1)$, отримаємо

$$\|w(k+1)\|^2 = \|w(k)\|^2 + 2\gamma e(k) w^T(k)x(k) + \gamma^2 e^2(k)\|x(k)\|^2. \tag{3.7}$$

Для простоти приймемо, що $w(0) = 0$. Оскільки

$$w^T(k)x(k) \begin{cases} > 0 \text{ для } e(k) = -1; \\ < 0 \text{ для } e(k) = +1 \end{cases} \tag{3.8}$$

(корекція коефіцієнтів відбуватиметься тільки в цьому випадку), то $2\gamma e(k) w^T(k)x(k) < 0$ і з урахуванням (3.5) можна записати

$$\begin{aligned} \|w(k+1)\|^2 &\le \|w(k)\|^2 + \gamma^2 e^2(k)\|x(k)\|^2 \le \|w(k)\|^2 + \gamma^2 A = \\ &= \gamma^2 \sum_{i=1}^{k} e^2(k)\|x(i)\|^2 \le \gamma^2 kA. \end{aligned} \tag{3.9}$$

Звідси отримуємо верхню межу для $\|w(k+1)\|$

$$\|w(k+1)\| \le \gamma (kA)^{1/2}. \tag{3.10}$$

З нерівності Коші — Шварца випливає, що

$$(w^{*T} w(k)) \le \|w^*\| \, \|w(k)\|.$$

Тому з урахуванням (3.6) отримуємо нижню межу довжини вектора ваг на k-й ітерації

$$\|\boldsymbol{w}(k+1)\| \geq \frac{\left(\boldsymbol{w}^{*T}\boldsymbol{w}(k+1)\right)}{\|\boldsymbol{w}^*\|} = \frac{\sum_{i=1}^{K}\gamma e(i)\boldsymbol{w}^{*T}\boldsymbol{x}(i)}{\|\boldsymbol{w}^*\|} > \frac{\gamma k\delta}{\|\boldsymbol{w}^*\|}. \qquad (3.11)$$

Поєднуючи (3.10) і (3.11), отримуємо

$$k\gamma\delta\|\boldsymbol{w}^*\|^{-1} \leq \|\boldsymbol{w}(k)\| \leq \gamma(kA)^{0,5}. \qquad (3.12)$$

Ліва частина нерівності (3.12) зростає лінійно зі збільшенням k і швидше правої частини. Тому число k обмежене. Таким чином, алгоритм навчання персептрона збігатиметься за кінцеве число ітерацій.

Приклад 3.3. Заданий тривходовий $\boldsymbol{x} = (x_1\ x_2\ x_3)^T$ одношаровий персептрон з вектором ваг $\boldsymbol{w}(0) = (0{,}5\ \ 0{,}1\ \ 1)^T$ й активаційною функцією виду $f(\cdot) = \operatorname{sgn}(\cdot)$. Слід змінити ваговий вектор так, щоб з появою вхідного сигналу $x = (-1\ -1\ 1)^T$ на виході персептрона з'являвся сигнал $y^* = -1$.

З наявними ваговими коефіцієнтами персептрон відпрацьовує вихідний сигнал

$$y(0) = \operatorname{sgn}(\boldsymbol{w}^T(0)\,\boldsymbol{x}) = \operatorname{sgn}\left[(0{,}5\ \ 0{,}1\ \ 1)\begin{pmatrix}-1\\-1\\1\end{pmatrix}\right] = \operatorname{sgn}(0{,}4) = 1,$$

що не відповідає необхідному.

Корекція ваг за алгоритмом (3.2) з $\gamma = 0{,}01$ дає

$$\boldsymbol{w}(1) = \boldsymbol{w}(0) - \gamma(y^* - y(1))\boldsymbol{x} = \begin{pmatrix}0{,}5\\0{,}1\\1\end{pmatrix} - 0{,}01\cdot 2\begin{pmatrix}-1\\-1\\1\end{pmatrix} = \begin{pmatrix}0{,}52\\0{,}12\\0{,}98\end{pmatrix}.$$

У цьому випадку

$$y(1) = \operatorname{sgn}(\boldsymbol{w}^T(1)\boldsymbol{x}) = \operatorname{sgn}\left[(0{,}52\ \ 0{,}12\ \ 0{,}98)\begin{pmatrix}-1\\-1\\1\end{pmatrix}\right] = \operatorname{sgn}(0{,}34) = 1.$$

Через те, що реакція персептрона на вхідний сигнал неправильна, знову корегуються ваги

$$w(2) = w(1) - \gamma(y^* - y(1))x = \begin{pmatrix} 0{,}52 \\ 0{,}12 \\ 0{,}98 \end{pmatrix} - 0{,}01 \cdot 2 \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0{,}54 \\ 0{,}14 \\ 0{,}96 \end{pmatrix}.$$

При цьому значення вихідного сигналу персептрона дорівнюватиме

$$y(2) = \operatorname{sgn}(w^T(2)x) = \operatorname{sgn}\left[(0{,}54 \;\; 0{,}14 \;\; 0{,}96) \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \right] = \operatorname{sgn}(0{,}26) = 1.$$

Неправильна реакція персептрона свідчить про необхідність подальшої корекції ваг. Діючи за аналогією, отримуємо, що вектор ваг і вихідний сигнал персептрона змінюватимуться в такий спосіб:

$$w(4) = (0{,}58 \;\; 0{,}18 \;\; 0{,}92)^T; \;\; y(4) = 1;$$

$$w(5) = (0{,}60 \;\; 0{,}20 \;\; 0{,}90)^T; \;\; y(5) = 1;$$

$$w(6) = (0{,}62 \;\; 0{,}22 \;\; 0{,}88)^T; \;\; y(6) = 1;$$

$$w(7) = (0{,}64 \;\; 0{,}24 \;\; 0{,}86)^T; \;\; y(7) = -1.$$

Після сьомого такту $y^* - y(7) = 0$. А оскільки реакція персептрона на заданий вхідний сигнал зараз правильна, навчання завершене.

Нескладно побачити, що вибір параметра γ впливає на тривалість процесу навчання. Так, вибираючи в розглянутому прикладі $\gamma = 0{,}02$, отримаємо

$$w(1) = (0{,}6 \;\; 0{,}2 \;\; 0{,}9)^T; \;\; y(1) = 1;$$

$$w(2) = (0{,}70 \;\; 0{,}20 \;\; 0{,}90)^T; \;\; y(5) = 1;$$

тобто навчання завершується за 2 такти.

При виборі ж $\gamma = 0{,}05$ відразу отримуємо один із можливих розв'язків

$$w(1) = (0{,}90 \;\; 0{,}50 \;\; 0{,}60)^T; \;\; y(1) = -1.$$

Мейсом запропоновано модифікації правила настроювання персептрона з лінійною активаційною функцією $z = w^T x$, у яких враховувалася величина реакції (активації) мережі, визначена деяким порогом Ф, що апріорно задається. Перше таке модифіковане правило має вигляд

$$w(k+1)=\begin{cases} w(k)+0{,}5\gamma e(k)x(k)\|x(k)\|^2 & \text{при } z(k)\geq \Phi; \\ w(k)+\gamma y^*(k)x(k)\|x(k)\|^2 & \text{при } z(k)<\Phi, \end{cases} \tag{3.13}$$

де $e(k)=y^*(k)-y(k)=y^*(k)-f(w^T(k)x(k))$.

У другому запропонованому ним правилі враховується той факт, що під час виконання умови $(z(k)\geq\Phi)\wedge(e(k)=0)$ ваги мережі не мають змінюватися. У випадку ж слабкої реакції мережі (при незначному ступені її активації), визначеної, як і раніше, порогом Ф або при виникненні помилки між поточною й необхідною реакціями мережі, здійснюється корекція ваг. Вищенаведене враховується в алгоритмі у такий спосіб:

$$w(k+1)=\begin{cases} w(k) \text{ при } (z(k)\geq\Phi)\wedge(e(k)=0; \\ w(\mathrm{k})+\gamma e(k)x(k)\|x(k)\|^2 \text{ в іншому випадку.} \end{cases} \tag{3.14}$$

Якщо множина поданих персептрону образів $\{x(k), y^*(k)\}$ є лінійно роздільною, то обидва ці алгоритми збігаються до розв'язку за кінцеве число кроків при $\gamma\in(0,2)$. Хоча в деяких випадках алгоритм (3.14) збігається швидше, ніж алгоритм (3.13), теоретичного обґрунтування цьому немає. Слід зазначити, що якщо алгоритм (3.13) ідейно ближчий до стандартного алгоритму навчання персептрона, то алгоритм (3.14) має багато спільною з алгоритмом *Уідроу — Гоффа*, що буде розглянутий нижче.

Згодом був запропонований персептрон із сигмоїдальною активаційною функцією (рис. 3.4).

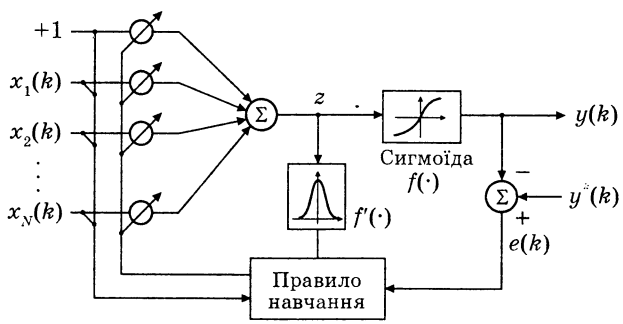


Рис. 3.4. Персептрон із сигмоїдальною активаційною функцією

Наявність у даному персептроні диференційованої активаційної функції дозволяє використати для його навчання градієнтні алгоритми

$$w(k+1)=w(k)-\gamma\nabla_w I(w). \tag{3.15}$$

Якщо в якості мінімізувального функціонала вибрати квадратичний (2.47), а активаційна функція має вигляд (2.14), то

$$\nabla_w I(\boldsymbol{w}) = -e(k)f'_{\text{th}}[z(k)]\boldsymbol{x}(k) = -e(k)\alpha[1 - y^2(k)]\boldsymbol{x}(k)$$

і алгоритм (3.15) записується в такий спосіб:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma\alpha[1 - y^2(k)]e(k)\boldsymbol{x}(k). \tag{3.16}$$

Якщо ж активаційна функція обрана вигляду (2.13), то

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma\alpha y(k)[1 - y(k)]e(k)\boldsymbol{x}(k). \tag{3.17}$$

З наведених формул видно, що вибір сигмоїдальних функцій активації дозволяє отримати прості алгоритми навчання.

3.1.4. Адаліна

Першу лінійну нейронну мережу запропоновано *Б. Уідроу* й *М. Е. Гоффом* у роботі [6] 1960 року. Розроблену ними модель було названо *адаптивним нейроном*, а потім *ADALINE* — **ADA***ptive* **LI***near* **NE***uron*. З часом свою популярність як ШНМ вона втратила, і зараз *ADALINE* — це **ADA***ptive* **LIN***ear* **E***lement*. Адаліна мала структуру, аналогічну персептрону. Зважені вхідні сигнали $x_i \in \{-1,1\}$ підсумовувалися й перетворювалися біполярною активаційною функцією (2.11) у вихідний сигнал $y \in \{-1,1\}$. Зазначимо, що вхідні сигнали можуть бути також булевими змінними або реальними числами. Основна відмінність Адаліни від персептрона полягала у використовуваному алгоритмі навчання. Хоча навчання Адаліни також полягало в корекції її вагових коефіцієнтів на основі поданих пар, що навчають, і порівнянні реальних вихідних сигналів з бажаними (необхідними), Уідроу й Гофф, ґрунтуючись на працях Н. Вінера, пов'язаних з дослідженням фільтрів, використали для цього градієнтний алгоритм.

Алгоритм Уідроу — Гоффа є так званим *нормалізованим алгоритмом методу найменших квадратів* (МНК) або дельта-правилом і може бути отриманий у такий спосіб (рис. 3.5).

Подамо бажаний й реальний вихідні сигнали лінійної частини Адаліни на $(k+1)$-му такті навчання відповідно як

$$y^*(k) = \boldsymbol{w}^{*T}\boldsymbol{x}(k); \tag{3.18}$$

$$u(k) = \boldsymbol{w}^T(k-1)\boldsymbol{x}(k), \tag{3.19}$$

де $\boldsymbol{w}^*$, $\boldsymbol{w}(k)$ — оптимальний вектор вагових коефіцієнтів і його оцінка, отримана на попередньому k-му такті навчання.

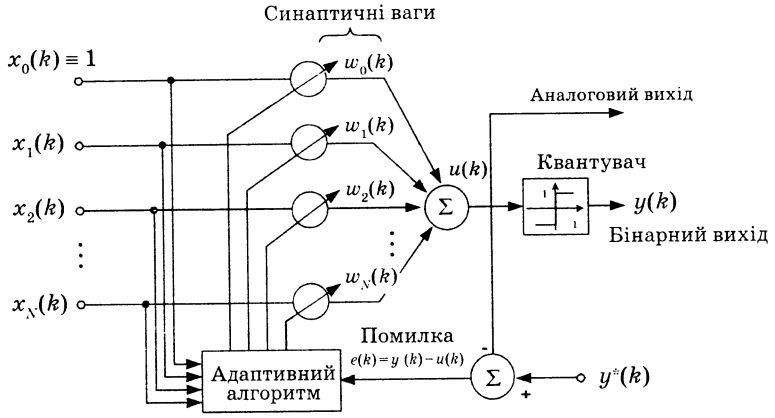


Рис. 3.5. Адаліна

Визначимо помилку реакції

$$e(k) = y^*(k) - u(k) = y^*(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k), \tag{3.20}$$

яку використаємо для формування квадратичного критерію якості навчання

$$I(\boldsymbol{w},\boldsymbol{x}) = e^2(k) = (y^*(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k))^2. \tag{3.21}$$

Визначаючи шукані вагові коефіцієнти з умови $I(\boldsymbol{w},\boldsymbol{x}) \to \min\limits_{\boldsymbol{w}}$, тобто з умови

$$\frac{\partial I(\boldsymbol{w},\boldsymbol{x})}{\partial \boldsymbol{w}} = 0, \tag{3.22}$$

і використовуючи градієнтний алгоритм (2.35), отримуємо таке правило корекції ваг:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \gamma(k)(y^*(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k))\boldsymbol{x}(k). \tag{3.23}$$

Тут γ — деякий позитивний параметр, що впливає на швидкість збіжності алгоритму (тривалість процесу навчання). Оптимальне значення γ, що забезпечує максимальну швидкість збіжності (3.18), визначається в такий спосіб.

Віднімемо з обох частин (3.21) $\boldsymbol{w}^*$, тобто запишемо алгоритм стосовно помилок оцінювання $\theta(i) = \boldsymbol{w}(i) - \boldsymbol{w}^*$. Тоді з урахуванням (3.16)– (3.18)

$$\theta(k+1) = \theta(k) - \gamma(k)(\theta^T(k)\boldsymbol{x}(k))\boldsymbol{x}(k). \tag{3.24}$$

Помножимо (3.24) ліворуч на $\theta^T(k)$

$$\|\theta(k+1)\|^2 = \|\theta(k)\|^2 - 2\gamma(k)(\theta^T(k)\boldsymbol{x}(k)) + \gamma^2(k)(\theta^T(k)\boldsymbol{x}(k))^2\|\boldsymbol{x}(k)\|^2.$$

Визначаючи γ_k з умови мінімуму $\|\theta(k+1)\|^2$, тобто з умови

$$\frac{\partial\|\theta(k+1)\|^2}{\partial\gamma(k)} = -2\left(\theta^T(k)\boldsymbol{x}(k)\right)^2 + 2\gamma(k)\left(\theta^T(k)\boldsymbol{x}(k)\right)^2\|\boldsymbol{x}(k)\|^2 = 0,$$

отримуємо такий вираз для $\gamma^{\text{опт}}(k)$ (*см.* (2.51)):

$$\gamma^{\text{опт}}(k) = \|\boldsymbol{x}(k)\|^{-2}. \tag{3.25}$$

Підстановка (3.25) у (3.23) дає алгоритм

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \frac{y^*(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|^2}\boldsymbol{x}(k), \tag{3.26}$$

що називається внаслідок використання в ньому нормалізації $\boldsymbol{x}(k)\|\boldsymbol{x}(k)\|^{-2}$ *нормалізованим алгоритмом МНК*. Зазначимо, що даний алгоритм уперше застосовано *Качмажем* [47] для розв'язання систем лінійних алгебраїчних рівнянь.

Записаний стосовно помилок оцінювання θ алгоритм (3.26) має вигляд

$$\theta(k+1) = \left(I - \frac{\boldsymbol{x}(k)\boldsymbol{x}^T(k)}{\|\boldsymbol{x}(k)\|^2}\right)\theta(k). \tag{3.27}$$

Тут I — одинична матриця $N \times N$; $\boldsymbol{x}(k)\boldsymbol{x}^T(k)\|\boldsymbol{x}(k)\|^{-2}$ — матриця проеціювання на вектор $\boldsymbol{x}(k)$. Внаслідок використання операції проеціювання швидкість збіжності цього градієнтного алгоритму максимальна.

Алгоритм Уідроу — Гоффа має вигляд, аналогічний (3.26)

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \alpha\frac{y^*(k) - \boldsymbol{w}^T(k)\,\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|^2}\boldsymbol{x}(k), \tag{3.28}$$

де $\alpha \in (0,2)$.

Неважко показати, що оптимальне значення $\alpha = 1$ і $\alpha < 1$, якщо змінні вимірюються неточно.

Приклад 3.4. Розглянемо персептрон, заданий у прикладі 3.3, і здійснимо корекцію його параметрів за алгоритмом (3.26). Тоді

$$\|\boldsymbol{x}(1)\|^2 = 3, \quad \boldsymbol{w}^T(0)\,\boldsymbol{x}(1) = (0{,}5 \quad 0{,}1 \quad 1)\begin{pmatrix}-1\\-1\\1\end{pmatrix} = 0{,}4,$$

$$w(1)=\begin{pmatrix}0,5\\0,1\\1\end{pmatrix}+\frac{(-1-0,4)}{3}\begin{pmatrix}-1\\-1\\1\end{pmatrix}=\begin{pmatrix}0,96\\0,56\\0,54\end{pmatrix};$$

і

$$y(1)=\operatorname{sgn}\left(w^T(1)x(1)\right)=\operatorname{sgn}\left[\begin{pmatrix}0,96 & 0,56 & 0,54\end{pmatrix}\begin{pmatrix}-1\\-1\\1\end{pmatrix}\right]=\operatorname{sgn}(-0,98)=-1.$$

У роботі [79] наведено моделі Адаліни, що містять замість порогової активаційної функції сигмоїдальну (рис. 3.6), а також має як сигмоїдальну, так і порогову активаційні функції (рис. 3.7). Відповідні сигнали помилок використовуються для корекції вагових коефіцієнтів Адаліни.

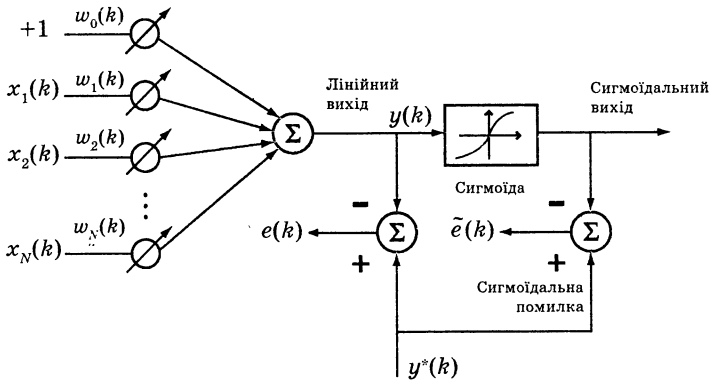


Рис. 3.6. Адаліна із сигмоїдальною функцією активації

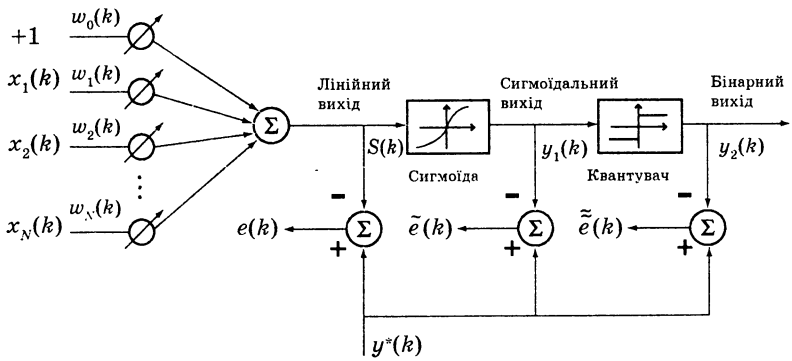


Рис. 3.7. Узагальнена Адаліна

3.1.5. Н-Адаліна

Н-Адаліна (N-ADALINE) є різновидом Адаліни, на вхід якої крім звичайних вхідних сигналів подаються їхні різні комбінації, тобто використовуються нелінійні перетворення вхідних сигналів. У ній реалізовано розроблений *О. Г. Івахненком* метод групового урахування аргументів, МГУА (*GMDH — Group Method of Data Handling*) [37, 46, 80, 81]. Н-Адаліну, вихідний сигнал якої обчислюється за формулою

$$y = w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_1 x_2 + w_4 x_2^2 + w_5 x_2, \qquad (3.29)$$

зображено на рис. 3.8.

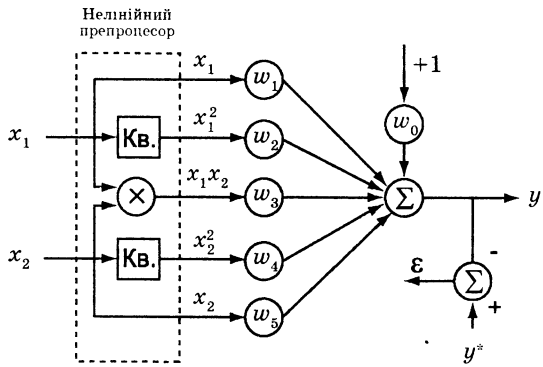


Рис. 3.8. Н-Адаліна

Як випливає з (3.29), вихід y є лінійною функцією відносно вагових коефіцієнтів w_i $(i = \overline{1,5})$ і нелінійною щодо вхідних сигналів x_1 й x_2. За аналогією можуть бути побудовані Н-Адаліни більш високих порядків. МГУА дозволяє вибрати оптимальну складність (кількість членів, степінь і т. д.) математичної моделі об'єкта.

Для навчання Адаліни використовують алгоритми навчання з учителем, наприклад, алгоритм Уідроу — Гоффа, що для Н-Адаліни, наведеної на рис. 3.8, має вигляд

$$\boldsymbol{w}(k+1)(k) = \boldsymbol{w}(k) + \alpha \frac{(\hat{y}(k) - \boldsymbol{w}^T(k)\boldsymbol{z}(k))}{\|\boldsymbol{z}(k)\|^2}\boldsymbol{z}(k), \qquad (3.30)$$

де $\boldsymbol{w} = (w_0, w_1, \ldots, w_5)^T$; $z = (1, x_1, x_1^2, x_1x_2, x_2^2, x_2)^T$; $\hat{y}(k)$ — необхідний вихідний сигнал; α — параметр навчання.

3.1.6. Вхідна зірка Ґроссберґа

Вхідна зірка (*Instar*) *С. Ґроссберґа* є нейроном, що за структурою аналогічний Адаліні, призначений для розв'язання найпростіших задач розпізнавання образів і здійснює перетворення

$$y_j = f(\boldsymbol{w}_j^T \boldsymbol{x} + \theta_j), \tag{3.31}$$

де

$$f(u_j) = \begin{cases} 1, & \text{якщо } u_j \geq 0, \\ 0 & \text{в іншому випадку.} \end{cases} \tag{3.32}$$

Схему вхідної зірки зображено на рис. 3.10.

Нескладно побачити, що цей нейрон активізується (на виході з'являється 1) у випадку, якщо вектор вхідних сигналів $x(k)$ у деякому значенні близький до поточного вектора синаптичних ваг $w_i(k)$, тобто з виконанням умови

$$\boldsymbol{w}_j^T(k)\boldsymbol{x}(k) = \|\boldsymbol{w}_j(k)\|\|\boldsymbol{x}(k)\|\cos\varphi \geq \theta_j, \tag{3.33}$$

де φ — кут між векторами $w_i(k)$ і $x(k)$; θ_i — сигнал зміщення, що задає поріг «близькості» векторів, що визначає спрацьовування вхідної зірки.

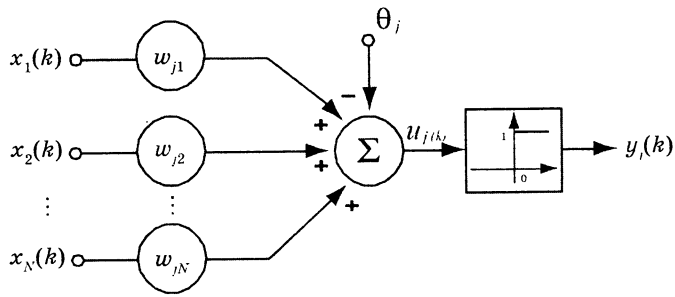


Рис. 3.10. Вхідна зірка

Якщо прийняти

$$\theta_j = \|\boldsymbol{w}_j\|\|\boldsymbol{x}\|, \tag{3.34}$$

то зірка активується тільки у випадку, якщо вхідний сигнал співпадає з вектором синаптичних ваг, тобто розпізнається тільки

один образ. Чим менше значення θ_i, тим більше можливих образів можуть активувати нейрон, що стає при цьому все менш «розбірливим».

Навчання вхідної зірки здійснюється за допомогою модифікованого алгоритму (3.9), що набуває у цьому випадку вигляду

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \gamma y_j(k)\boldsymbol{x}(k) - \alpha y_j(k)\boldsymbol{w}_j(k). \qquad (3.35)$$

Необхідність модифікації пов'язана з тим, що у випадку подачі на вхід нейрона послідовності $\boldsymbol{x}(k)$, що не активує зірку ($y_j(k) = 0$), відбувається поступове забування всієї накопиченої інформації. Дійсно, в цьому випадку алгоритм (3.9) набуває вигляду

$$\boldsymbol{w}_j(k+1) = (1-\alpha)\boldsymbol{w}_j(k). \qquad (3.36)$$

Характерною особливістю правила (3.35) є те, що самонавчання відбувається тільки в активованому стані, коли $y_j(k) = 1$.

Поклавши для простоти $\alpha = \gamma$, отримуємо так зване *стандартне правило самонавчання вхідної зірки*

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \gamma y_j(k)(\boldsymbol{x}(k) - \boldsymbol{w}_j(k)), \qquad (3.37)$$

геометричну ілюстрацію якого наведено на рис. 3.10.

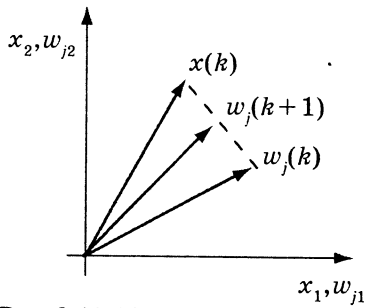


Рис. 3.10. Навчання вхідної зірки

При $y_j(k) = 0$ згідно з (3.37) навчання не відбувається, тобто $\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k)$. При $y_j(k) = 1$ алгоритм природно набуває вигляду

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \gamma(\boldsymbol{x}(k) - \boldsymbol{w}_j(k)) = (1-\gamma)\boldsymbol{w}_j(k) + \gamma\boldsymbol{x}(k), \qquad (3.38)$$

тобто вектор синаптичних ваг «підтягується» до вхідного образу на відстань, пропорційний параметру кроку γ. Чим більше γ, тим ближче $\boldsymbol{w}_j(k+1)$ до $\boldsymbol{x}(k)$ і при $\gamma = 1$ збігається з ним. Зазвичай у реальних задачах використовується змінне значення $\gamma(k)$, обумовле-

не відповідно до умов *Дворецького* [115]. Можна також зазначити, що для зручності обчислювання замість вектора $\boldsymbol{x}(k)$ частіше використовують його нормований аналог

$$\tilde{\boldsymbol{x}}(k) = \frac{\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|}.$$

3.1.7. Вихідна зірка

Своєрідним антиподом вхідної зірки є *вихідна зірка* (*Outstar*), призначена для розв'язання задач відновлення образів, а її схему зображено на рис. 3.11.

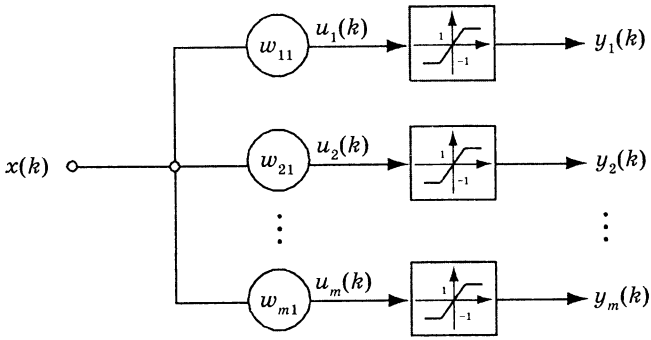


Рис. 3.11. Вихідна зірка

Цей нейрон має скалярний вхід і векторний вихід і здійснює перетворення

$$y_j = f(w_{j1}x),\ j = 1, 2, \ldots, m \tag{3.39}$$

з функцією активації

$$f(u_j) = \begin{cases} u_j, & \text{якщо } -1 \le u_j \le 1, \\ 1, & \text{якщо } 1 < u_j, \\ -1, & \text{якщо } u_j < -1. \end{cases} \tag{3.40}$$

Правило самонавчання вихідної зірки має вигляд

$$w_{ji}(k+1) = w_{ji}(k) + \gamma y_j(k)x_i(k) - \alpha x_i(k)w_{ji}(k), \tag{3.41}$$

а при $\gamma = \alpha$ —

$$w_{ji}(k+1) = w_{ji}(k) + \gamma x_i(k)(y_j(k) - w_{ji}(k)), \tag{3.42}$$

тобто настроювання синаптичних ваг відбувається тільки у випадку $x_i(k) \neq 0$. Тут, як видно, у процесі самонавчання синаптичні ваги «підтягуються» до вихідного вектора $y(k)$.

У векторній формі правило самонавчання має вигляд

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \gamma x_i(k)(\mathbf{y}(k) - \boldsymbol{w}_i(k)), \tag{3.43}$$

де $\boldsymbol{w}_i(k)$ — i-й стовпець матриці коефіцієнтів $W(k+1)$.

3.2. Лінійна роздільність

Нехай N-вимірний простір X складається з двох підпросторів X_1 і X_2. Ці підпростори називаються *лінійно роздільними*, якщо є $(N+1)$-вимірний вектор $\boldsymbol{w}$, такий що

$$\sum_{i=1}^{N} w_i x_i = \begin{cases} \geq w_0 & \forall x = (x_1, \ldots, x_N) \in X_1; \\ < w_0 & \forall x = (x_1, \ldots, x_N) \in X_2. \end{cases} \tag{3.44}$$

Якщо

$$\sum_i w_i x_i = \begin{cases} > w_0 & \forall x \in X_1; \\ < w_0 & \forall x \in X_2, \end{cases} \tag{3.45}$$

то X_1 і X_2 називаються *абсолютно лінійно роздільними*.

Приклади розбивання на два класи наведено на рис. 3.12, 3.13.

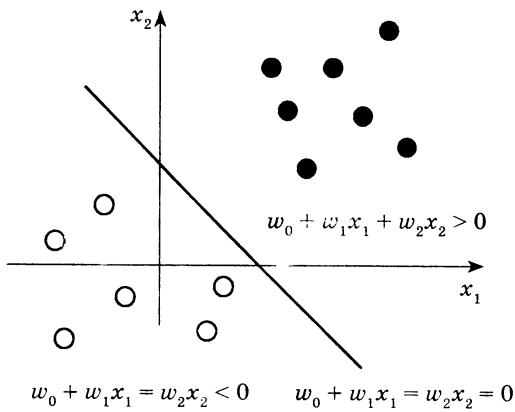


Рис. 3.12. Лінійно роздільні множини

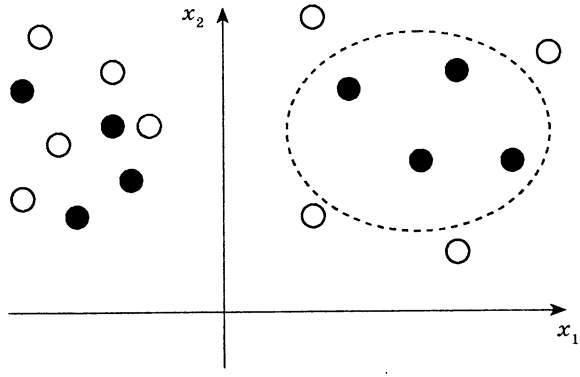


Рис. 3.13. Лінійно нероздільні множини

На рис. 3.14, 3.15 наведено графічну ілюстрацію реалізації найпростішим персептроном функцій «І» й «що виключає АБО». Темні кільця на рисунках відповідають значенню функції $f = 1$.

Як видно з рисунків, можна провести нескінченну безліч прямих, що відповідають різним значенням ваг w_1 й w_2, які відповідають нерівностям із прикладу 3.1 і реалізують функцію «І», і неможливо провести одну пряму, що розділяє площину X_1–X_2 так, щоб реалізовувалася функція «що виключає АБО» (приклад 3.2).

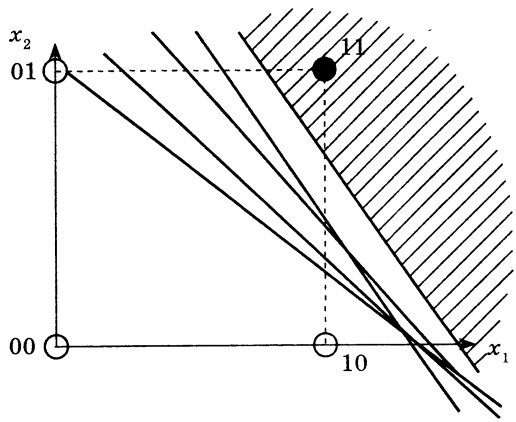


Рис. 3.14. Реалізація функції «І»

Слід зазначити, що вектор ваг $\boldsymbol{w}$ буде завжди ортогональним граничній площині.

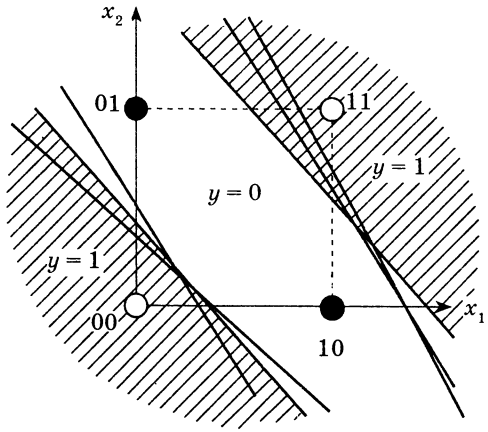


Рис. 3.15. Реалізація функції «що виключає АБО»

Приклад 3.5. Нехай є двовходовий персептрон (рис. 3.2) з вагами $w_1 = 0{,}5$, $w_2 = 1$ і зсувом $\theta = -1$. Тоді межа розв'язків є лінією, що описана рівнянням

$$0{,}5x_1 + 1x_2 - 1 = 0$$

і наведена на рис. 3.16.

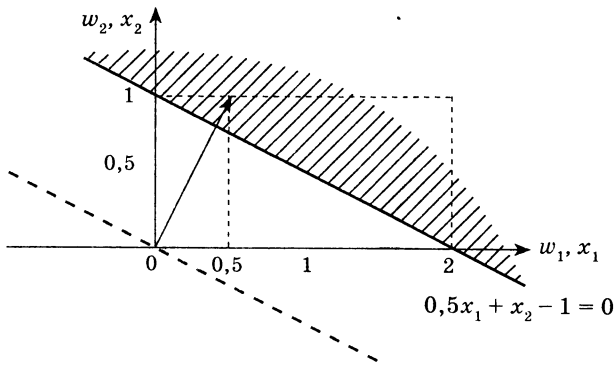


Рис. 3.16. Розташування межі розв'язків і вектора ваг

Нескладно перевірити, що область над межею відповідає $y = 1$, а під межею — $y = 0$.

Зобразивши на цьому ж графіку вектор $w = (0{,}5 \quad 1)^T$, неважко переконатися, що він буде ортогональний межі розв'язків.

Зазначимо, що це справедливо й для випадку $\theta = 0$ (пунктирна пряма на рис. 3.16).
Графічна ілюстрація дозволяє наочно уявити процес навчання персептрона.

Приклад 3.6. Нехай двовходовому персептрону з $\boldsymbol{w}_0 = (0{,}8\ -0{,}6)^T$ (рис. 3.2) послідовно подаються такі пари, що навчають:
$(\boldsymbol{x}_1 = (1\ \ 1{,}5)^T, y_1^* = 1)$, $(\boldsymbol{x}_2 = (-1\ \ 2{,}5)^T, y_2^* = 0)$ $(x_3 = (-0{,}5\ \ -2)^T, y_3^* = 0)$.
На рис. 3.17 відповідні координати навчальних векторів позначені темними ($y^* = 1$) і світлими ($y^* = 0$) кільцями. Тут також наведено вихідний вектор ваг $\boldsymbol{w} = (0{,}8\ \ -0{,}6)^T$.

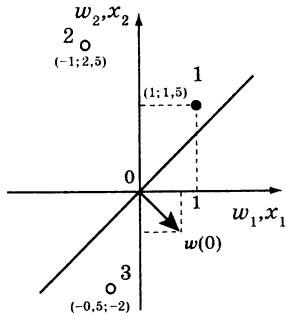


Рис. 3.17. Розташування початкового вектора ваг і векторів, що навчають

Подання вхідного вектора $\boldsymbol{x}_1 = (1\ 1{,}5)^T$ викликає на виході персептрона сигнал

$$y_1 = f_a\left(\boldsymbol{w}_0^T \boldsymbol{x}_1\right) = f_a\left((0{,}8\ \ -0{,}6)\begin{pmatrix}1\\1{,}5\end{pmatrix}\right) = f_a(-0{,}1) = 0,$$

тобто даний образ класифікований неправильно.
Дійсно, з рис. 3.17 видно, що перший і третій образи розташовано по одну сторону від межі розв'язків.
Корекція ваг за алгоритмом (3.2) з $\gamma = 1$ дає

$$\boldsymbol{w}_1 = \boldsymbol{w}_0 + \boldsymbol{x}_1 = \begin{pmatrix}0{,}8\\-0{,}6\end{pmatrix} + \begin{pmatrix}1\\1{,}5\end{pmatrix} = \begin{pmatrix}1{,}8\\0{,}9\end{pmatrix}.$$

Після подання $\boldsymbol{x}_2 = (-1 \;\; 2{,}5)^T$ отримуємо

$$y_2 = f_a\left(\boldsymbol{w}_1^T \boldsymbol{x}_2\right) = f_a\left((1{,}8 \quad 0{,}9)\begin{pmatrix} -1 \\ 2{,}5 \end{pmatrix}\right) = f_a(0{,}445) = 1,$$

а оскільки класифікація здійснена неправильно, знову коректуємо ваги

$$\boldsymbol{w}_2 = \boldsymbol{w}_1 - \boldsymbol{x}_2 = \begin{pmatrix} 1{,}8 \\ 0{,}9 \end{pmatrix} - \begin{pmatrix} -1 \\ 2{,}5 \end{pmatrix} = \begin{pmatrix} 2{,}8 \\ -1{,}6 \end{pmatrix}.$$

Подання $\boldsymbol{x}_3 = (-0{,}5 \;\; -2)^T$ дає

$$y_3 = f_a\left((2{,}8 \quad -1{,}6)\begin{pmatrix} -0{,}5 \\ -2 \end{pmatrix}\right) = f_a(0{,}8) = 1,$$

тому ваги коректуємо знову

$$\boldsymbol{w}_3 = \boldsymbol{w}_2 - \boldsymbol{x}_3 = \begin{pmatrix} 2{,}8 \\ -1{,}6 \end{pmatrix} - \begin{pmatrix} -0{,}5 \\ -2, \end{pmatrix} = \begin{pmatrix} 3{,}3 \\ 0{,}4 \end{pmatrix}.$$

На рис. 3.18 показано послідовні розташування вектора коефіцієнтів після кожного подання персептрону пар, що навчають. Як видно з рисунка, після подання $(\boldsymbol{x}_3, y_3^*)$ персептрон остаточно навчився, тобто налаштував вагові коефіцієнти так, щоб подані йому вектори класифікувалися правильно. Про це свідчить і положення межі розв'язків — прямої, перпендикулярної w_3 і тієї, що ділить всю площину на два класи.

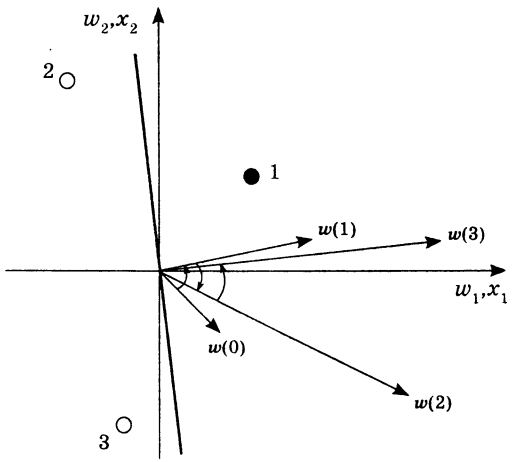


Рис. 3.18. Послідовна зміна розташування вектора ваг

Таким чином, як найпростіший (одношаровий) персептрон, так і Адаліна можуть класифікувати лише лінійно роздільні множини. На жаль, приклад, наведений на рис. 3.15, не єдиний, що свідчить про обмежену можливість одношарових мереж. Тому практичне застосування персептронних схем вимагає впевненості в тому, що розглянута функція є лінійно роздільною. Для загального випадку проблема лінійної роздільності не вирішена.

Р. Віднер досліджував цю проблему теоретично й визначив число лінійно роздільних функцій для n нейронів, входами яких є двійкові змінні. Оскільки в цьому випадку загальне число комбінацій вхідних сигналів становить 2^n, то загальне число вихідних комбінацій (булевих функцій) дорівнюватиме 2^{2^n} [34].

Таблиця 3.2. Лінійно роздільні функції

n	Число булевих функцій	Число лінійно роздільних функцій
1	4	4
2	16	14
3	256	104
4	65536	1772
5	$4{,}3 \cdot 10^9$	94572
6	$1{,}8 \cdot 10^{19}$	5028134

Як видно з таблиці, імовірність того, що випадково обрана функція виявиться лінійно роздільною, досить мала навіть для невеликого числа змінних.

Ефективність схем персептронного типу значно підвищується шляхом переходу до багатошарових персептронів.

3.3. Багатошарові ШНМ

3.3.1. Багатошаровий персептрон

Загальний вигляд багатошарового персептрона наведено на рис. 3.19.

Даний персептрон, що має N входів, M виходів й K шарів, з яких перший, що містить S нейронів, є вхідним, k-й, що складається з M нейронів, — вихідним, а інші — прихованими, описується рівнянням

$$y = f^{(k)}\left[\boldsymbol{w}^{(k)} \ldots f^{(2)}\left[\boldsymbol{w}^{(2)} f^{(1)}\left[\boldsymbol{w}^{(1)} x + \boldsymbol{b}^{(1)}\right] + \boldsymbol{b}^{(2)}\right] \ldots + \boldsymbol{b}^{(k)}\right],$$

де $\boldsymbol{w}^{(i)}$ — матриця ваг i-го шару; $\boldsymbol{b}^{(i)}$ — вектор зсувів i-го шару.

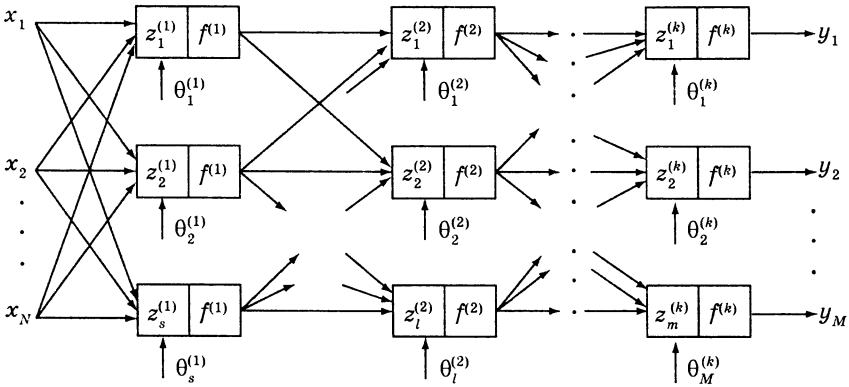


Рис. 3.19. Багатошаровий персептрон

За допомогою багатошарових персептронов можна вирішувати значно складніші задачі, зокрема можна реалізувати нерозв'язну за допомогою одношарового персептрона функцію «що виключає АБО».

Приклад 3.7. Одну з можливих реалізацій функції «що виключає АБО» наведено на рис. 3.20.

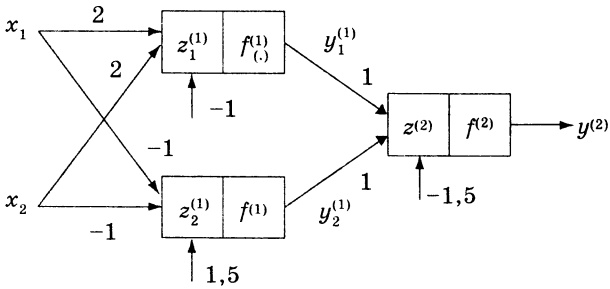


Рис. 3.20. Реалізація функції «що виключає АБО»

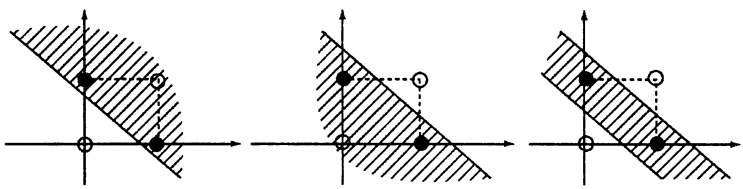
На рисунку позначено: $z_i^{(j)}$ — зважена сума входів i-го нейрона j-го шару; $y_i^{(j)}$ — вихідний сигнал i-го нейрона j-го шару; f — порогова активаційна функція.

Функціонування цієї мережі описується в табл. 3.3.

Таблиця 3.3

x_1	x_2	$y_1^{(1)}$	$y_2^{(1)}$	$y^{(2)}$
0	0	$2\cdot0+2\cdot0-1=0$	$(-1)\cdot0+(-1)\cdot0+1{,}5=1$	$1\cdot0+1\cdot1-1{,}5=0$
0	1	$2\cdot0+2\cdot1-1=1$	$(-1)\cdot0+(-1)\cdot1+1{,}5=1$	$1\cdot1+1\cdot1-1{,}5=1$
1	0	$2\cdot1+2\cdot0-1=1$	$(-1)\cdot1+(-1)\cdot0+1{,}5=1$	$1\cdot1+1\cdot1-1{,}5=1$
1	1	$2\cdot1+2\cdot1-1=1$	$(-1)\cdot1+(-1)\cdot1+1{,}5=0$	$1\cdot1+1\cdot0-1{,}5=0$

Кожний з нейронів першого шару здійснює лінійне розбивання на два класи (рис. 3.21, *а* й *б*), а нейрон другого шару реалізує функцію «І» (рис. 3.21, *в*).



а *б* *в*

Рис. 3.21. Послідовний поділ на два класи

Цей приклад свідчить про те, що використання більшої кількості шарів дозволяє отримувати області розв'язків у вигляді багатокутників кожної наперед заданої форми.

Ці області завжди будуть опуклими, оскільки вони утворені з використанням операції «І» над областями, що задаються лініями.

Неопуклі області розв'язків можуть бути реалізовані на тришаровій мережі, коли опуклі багатокутники, що надходять на входи нейрона третього шару шляхом певних логічних комбінацій, здійснюваних цим нейроном, утворюють неопуклий багатокутник. Слід зазначити, що зі збільшенням кількості нейронів кількість сторін багатокутника необмежено зростає. Це дозволяє апроксимувати області будь-якої форми з необхідною точністю. На рис. 3.22 показано можливості персептронів різної структури [82].

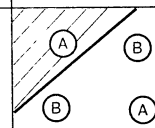
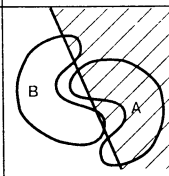
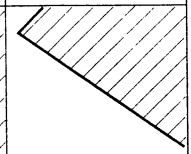
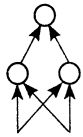
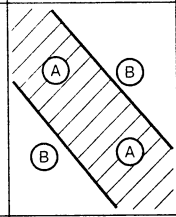
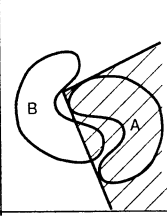
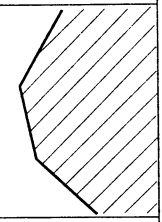
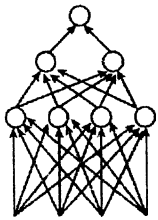
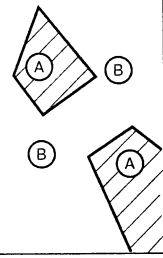
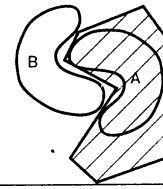
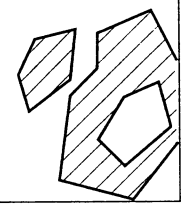
Структура персептрона	Область розв'язання	«Що виключає АБО»	Вільно розташовані області	Області найбільш загальної форми
Одношарова 	Напів-площина, обмежена гіпер-площиною			
Двошарова 	Опуклі відкриті або закриті області			
Тришарова 	Будь-якої складності (обмежені числом вузлів)			

Рис. 3.22. Можливості персептронів

Приклад 3.8. Розглянемо задачу стиснення інформації (*Encoding-Problem*). На вхід мережі подається n-розрядна двійкова послідовність, що містить тільки одну 1. Вихід багатошарового персептрона має повторити вхідний сигнал. При n нейронах вхідного й вихідного шарів прихований шар містить $(\log_2 n)$ нейронів. Таким чином, мережа має навчитися кодувати n-розрядний вхідний сигнал у $(\log_2 n)$-розрядний, а потім його знову декодувати.

На рис. 3.23 наведено тришаровий персептрон, що вирішує проблему 8-3-8.

У таблиці 3.4 наведено значення ваг (прихованого шару), отримані за допомогою градієнтного алгоритму навчання після 3150 поданих образів [56].

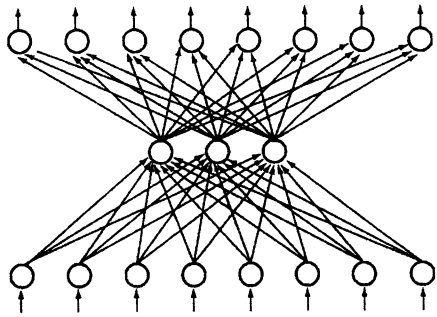


Рис. 3.23. Персептрон, що вирішує проблему 8-3-8

Таблиця 3.4

Вхідний образ	Прихований шар	Вихідний образ
10000000	0,0 1,0 0,6	10000000
01000000	1,0 1,0 1,0	01000000
00100000	0,2 0,6 0,0	00100000
00010000	1,0 1,0 0,2	00010000
00001000	1,0 0,0 0,1	00001000
00000100	0,0 0,0 0,3	00000100
00000010	1,0 0,0 1,0	00000010
00000001	0,0 0,3 1,0	00000001

На рис. 3.24 наведено графік залежності помилки мережі від кількості циклів навчання. Після завершення навчання помилка становила 8 %.

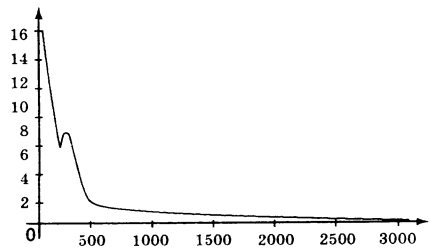


Рис. 3.24. Графік зміни помилки

3.3.2. Мадаліна

Мадаліна (*Many ADALINES*) є в загальному випадку багатошаровою мережею, утвореною багатьма Адалінами (рис. 3.25).

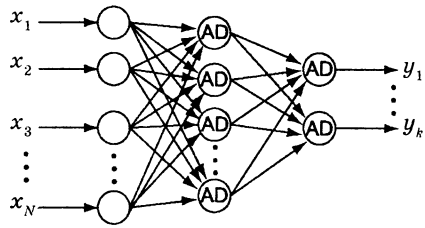


Рис. 3.25. Мадаліна

Цей тип мереж був покликаний усунути недоліки, властиві як Адаліні, так й одношаровому персептрону, і вирішувати задачі, які не могли бути вирішені з їхньою допомогою, зокрема здійснювати класифікацію лінійно нероздільних класів, вирішувати проблему «що виключає АБО» і т. д.

Приклад 3.9. Мадаліну, яка реалізує функцію «що виключає АБО», наведено на рис. 3.26.

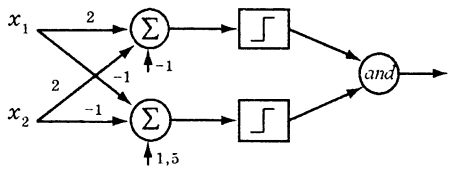


Рис. 3.26. Реалізація логічної функції «що виключає АБО» на двох Адалінах

На вхід мережі подаються сигнали $\boldsymbol{x}_1$ й $\boldsymbol{x}_2$, що приймають значення +1 й −1. Кожна з Адалін, показаних на рисунку, виробляє вихідний сигнал, що являє собою біполярне перетворення зважених сум вхідних сигналів. Ці вихідні сигнали надходять на вхід елемента «І», на вході якого, як нескладно переконатися, з'являється сигнал «+1», якщо значення вхідних сигналів $\boldsymbol{x}_1$ й $\boldsymbol{x}_2$ не збігаються, і «0», коли збігаються, тобто реалізується логічна функція «що виключає АБО».

Якщо первинні схеми Мадаліни використовували настроювання вагових параметрів Адалін першого шару й порогові (релейні) функції активації, то згодом були розроблені алгоритми настроювання ваг усіх шарів Мадаліни, що реалізують навчання з учителем, і використані неперервні нелінійні, зокрема, сигмоїдальні функції активації.

Адаптивний алгоритм навчання Мадаліни заснований на принципі, названому Уідроу «*принципом мінімального збурювання*», і полягає в тому, що корекція вагових коефіцієнтів при поданні нового образу, що забезпечує зменшення виникаючої вихідної помилки мережі, має бути мінімальною. Це пов'язано з тим, що наявні до моменту подання нового образу значення коефіцієнтів, обчислені на основі минулих образів, правильно відбивали роботу мережі. Робота алгоритму полягає в наступному.

При поданні нового образу у вхідному шарі вибирається нейрон, значення вихідного сигналу якого близьке до нуля, і змінюються його вагові коефіцієнти таким чином, щоб його бінарний стан змінився на протилежний. Далі простежується, як впливає ця зміна на стан нейронів наступних шарів і чи зменшується при цьому вихідна помилка: якщо зменшується, то внесені зміни ваг залишаються, якщо ні — значення коефіцієнтів залишаються попередніми й шукається інший нейрон вхідного шару. Цей процес триває доти, поки не будуть переглянуті всі нейрони першого шару. Після цього вибираються різні пари нейронів першого шару, для яких процедура зміни ваг повторюється й визначається вплив цієї зміни на вихідну помилку. Якщо помилка зменшується, зміни залишають, якщо ні — значення вагових коефіцієнтів пар залишають попередніми. Після аналізу всіх пар переходять до аналізу різних трійок, однак на практиці обмежуються звичайно парами.

Після цього переходять до нейронів другого шару й повторюють всю процедуру корекції ваг з урахуванням зміни ваг нейронів першого шару. Потім розглядають третій шар і т. д. аж до останнього (вихідного), вагові коефіцієнти якого корегують інакше, а саме з використанням дельта-правила або будь-якого іншого алгоритму навчання з учителем. Це можливо внаслідок того, що виявляються помилки кожної окремої Адаліни, що перебуває у вихідному шарі. Весь процес корекції вагових коефіцієнтів повторюється для кожного знову поданого образу.

Розглянутий адаптивний алгоритм навчання Мадаліни був згодом модифікований й удосконалений, і сьогодні існує багато його різних варіантів, які використовують, зокрема, крім порогової активаційної функції й сигмоїдальні, що надає мережі додаткові

позитивні властивості, а сам алгоритм стає практично еквівалентним найбільш відомому й досконалому *алгоритму зворотного поширення помилки*, який розглядатиметься нижче.

3.3.3. ШНМ, що заснована на МГУА

За аналогією з Мадаліною, елементами якої є Адаліни, може бути побудована мережа, що складається з *Н-Адалін*.

В алгоритмах МГУА аргументи й проміжні змінні поєднуються попарно, тобто в групи, що складаються із двох аргументів (тому МГУА, очевидно, більш точно може бути названий методом попарного урахування аргументів). У такий спосіб мережа, що складається з m шарів, може реалізовувати поліном степеня 2^m.

Дана мережа, запропонована в [46], називається *мережею, заснованою на МГУА* (*GMDH Neural Network*) (рис. 3.27).

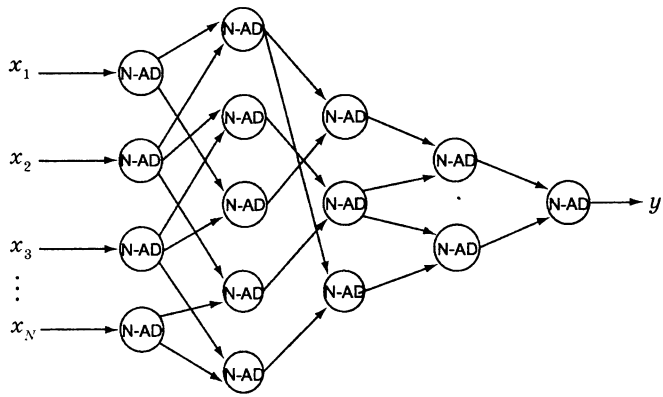


Рис. 3.27. ШНМ, заснована на МГУА

Основним критерієм оптимізації в МГУА є *критерій мінімуму середньоквадратичної помилки* (СКП), а питання, пов'язані зі знаходженням цього мінімуму, вирішуються за допомогою перебирання варіантів, що використовують навчальні й перевіркові послідовності даних. Для вибору кращих варіантів використовується пороговий принцип.

Критерій мінімуму СКП використовується в алгоритмах МГУА принаймні в чотирьох випадках:

1) для обчислення вагових коефіцієнтів кожної Н-Адаліни;

2) для выбору кращих комбінацій пар аргументів, тобто входів Н-Адалін;

3) для визначення величини порогів, на основі яких відбувається відбір кращих варіантів;

4) для визначення ступеня нелінійності використовуваного опису й вибору функцій, що його утворюють.

Якщо у звичайних методах, як правило, з умови мінімуму СКП знаходять вагові коефіцієнти *повного* рівняння, що описує об'єкт, то МГУА забезпечує такий вибір вагових коефіцієнтів *окремих* рівнянь, що становлять повне, при якому досягається мінімум СКП у просторі цих коефіцієнтів. Коефіцієнти ж повного рівняння визначаються за допомогою виключення проміжних змінних із окремих рівнянь. Крім того, для зменшення СКП у МГУА здійснюється повний або частковий перебір можливих комбінацій пар аргументів (вхідних сигналів Н-Адалін), виконуваний також за критерієм мінімуму СКП. У результаті вибираються комбінації, для яких помилка мінімальна.

Кількість враховуваних пар, змінних у кожному шарі багатошарової мережі визначається порогами, що обчислюють також за мінімумом СКП.

Повний опис МГУА, у тому числі поділ даних на послідовності, що навчають і перевіряють, оптимізацію порогів, правила селекції змінних і т. д., наведено в монографії [81]. Тут зазначимо лише, що оскільки ця мережа призначена для моделювання й прогнозування складних процесів, вхідні й вихідні сигнали яких є випадковими, перед початком навчання мережі здійснюється стандартизація (центрування й нормування) змінних за формулами

$$\hat{x}_i = \frac{x_i - \overline{x}_i}{\sigma_{x_i}}; \quad \hat{y}_i = \frac{y_i - \overline{y}_i}{\sigma_{y_i}}, \tag{3.46}$$

де x_i, y_i, $\overline{x}_i$, $\overline{y}_i$ — вхідні й вихідні змінні та їхні середні значення відповідно; σ_{x_i}, σ_{y_i} — середньоквадратичні відхилення x_i й y_i.

Потім формуються різні пари вхідних сигналів Н-Адалін першого шару та здійснюється корекція їхніх ваг за правилом Уідроу — Гоффа.

Після досягнення мінімального значення сумарної СКП першого шару корекція ваг цього шару завершується, тобто вагам привласнюються відповідні значення, які надалі не змінюються. Процедура навчання (селекція сигналів, корекція тощо) повторюється для другого шару й т. д. до досягнення всією мережею необхідного або припустимого значення СКП.

3.4. Алгоритм зворотного поширення помилки

Алгоритм зворотного поширення помилки, згодом названий просто *алгоритмом зворотного поширення* (*Backpropagation, BP*), що являє собою розширене дельта-правило, був запропонований у роботі [17] і застосований для навчання ШНМ у роботах [18, 19]. Він реалізує градієнтний метод мінімізації опуклого (звичайного квадратичного) функціонала помилки в багатошарових мережах прямого поширення, що використовують моделі нейронів з диференціальними функціями активації. Застосування сигмоїдальних функцій активації, що є монотонно зростаючими і що мають відмінні від нуля похідні на всій області визначення, забезпечує правильне навчання й функціонування мережі. Процес навчання полягає у послідовному поданні мережі пар $(\boldsymbol{x}(i), \boldsymbol{y}^*(i))$ $i=\overline{1,P}$, що навчають, де $\boldsymbol{x}(i)$ і $\boldsymbol{y}^*(i)$ — вектор вхідних і бажаних вихідних сигналів мережі відповідно, вивченні реакції на них мережі й корекції відповідно до реакції вагових параметрів (елементів вагової матриці).

Перед початком навчання всім вагам привласнюються невеликі різні випадкові значення (якщо задати всі значення однакові, а для правильного функціонування мережі знадобляться нерівні значення, мережа не навчатиметься).

Для реалізації алгоритму зворотного поширення необхідно:

1. Вибрати із заданої навчальної множини чергову пару $(\boldsymbol{x}(i), \boldsymbol{y}^*(i))$, $i=\overline{1,P}$, що навчає, і подати на вхід мережі вхідний сигнал $\boldsymbol{x}(i)$.
2. Обчислити реакцію мережі $\boldsymbol{y}(i)$.
3. Порівняти отриману реакцію $\boldsymbol{y}(i)$ з необхідною $\boldsymbol{y}^*(i)$ і визначити помилку $\boldsymbol{y}^*(i) - \boldsymbol{y}(i)$.
4. Скорегувати ваги так, щоб помилка була мінімальною.
5. Кроки 1–4 повторити для всієї множини пар $(\boldsymbol{x}(i), \boldsymbol{y}^*(i))$ $i=\overline{1,P}$, що навчають, доти, поки на заданій множині помилка не досягне необхідної величини.

Таким чином, у процесі навчання мережі подача вхідного сигналу й обчислення реакції відповідає *прямому* проходу сигналу від вхідного шару до вихідного, а обчислення помилки й корекція вихідних параметрів — *зворотному*, коли сигнал помилки поширюється по мережі від її виходу до входу. При зворотному проході здійснюється пошарова корекція ваг, починаючи з вихідного шару. Якщо корекція ваг вихідного шару здійснюється за допомогою модифікованого дельта-правила порівняно просто, оскільки

необхідні значення вихідних сигналів відомі, то корекція ваг прихованих шарів відбувається трохи складніше, оскільки для них не відомі необхідні вихідні сигнали.

Алгоритм зворотного поширення застосовують також щодо мереж з будь-якою кількістю шарів: як мереж прямого поширення, так і таких, що містять зворотні зв'язки. Ми обмежимося розглядом випадку навчання двох шарів мережі прямого поширення. Фрагмент такої мережі зображено на рис. 3.28.

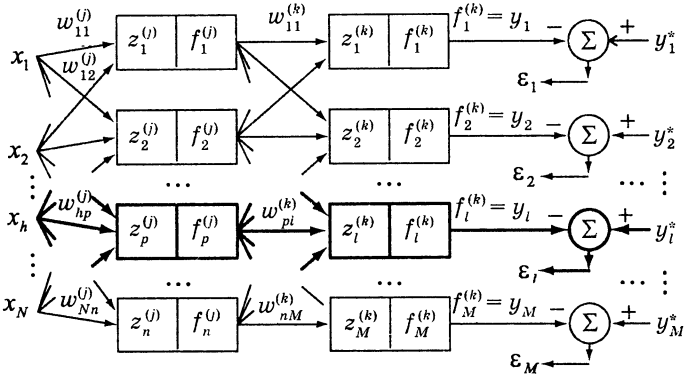


Рис. 3.28. Фрагмент мережі прямого поширення

На рисунку позначено: $w_{pl}^{(k)}$ — вага зв'язку p-го нейрона попереднього шару (j-го) з l-м нейроном наступного (k-го) шару; $f_p^{(j)}$ — активаційна функція p-го нейрона i-го шару; $z_p^{(j)}$ — зважена сума вихідних сигналів попереднього (i-го) шару, що надходять на вхід p-го нейрона наступного (j-го) шару.

3.4.1. Обчислення ваг нейронів вихідного шару

На рис. 3.29 використовуються ті позначення, що й на рис. 3.28. Розглянемо l-й нейрон вихідного (k-го) шару. Вихід даного нейрона є виходом мережі, тому його сигнал $y_l = f_l^{(k)}$ порівнюється з необхідним y_l^* й обчислюється помилка

$$\zeta_l = y_l^* - f_l^{(k)}. \qquad (3.47)$$

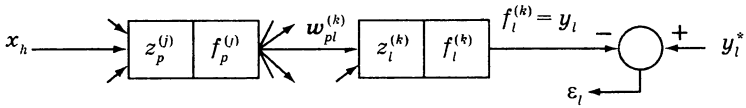


Рис. 3.29. Фрагмент багатошарової мережі

Використання квадратичного критерію якості навчання

$$\zeta_l^2 = (y_l^* - f_l^{(k)})^2 \tag{3.48}$$

дозволяє отримати градієнтний алгоритм корекції ваг, що у цьому випадку набуває вигляду

$$\Delta w_{pl}^{(k)} = -\gamma_{pl} \frac{\partial \zeta_l^2}{\partial w_{pl}^{(k)}}, \tag{3.49}$$

де γ_{pl} — коефіцієнт, що впливає на швидкість навчання.

Обчислення складової (3.49) частинної похідної здійснюється за правилом

$$\frac{\partial \zeta_l^2}{\partial w_{pl}^{(k)}} = \frac{\partial \zeta_l^2}{\partial f_l^{(k)}} \frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} \frac{\partial z_l^{(k)}}{\partial w_{pl}^{(k)}}. \tag{3.50}$$

З урахуванням (3.48), виду активаційних функцій і того, що

$$z_l^{(k)} = \sum_{p=1}^{n} w_{pl}^{(k)} f_p^{(j)},$$

отримуємо

$$\frac{\partial \zeta_l^2}{\partial f_l^{(k)}} = -2(y_l^* - y_l) = -2\zeta_l; \tag{3.51}$$

$$\frac{\partial f_l^{(k)}}{\partial Z_l^{(k)}} = \alpha f_l^{(k)}(1 - f_l^{(k)}); \tag{3.52}$$

$$\frac{\partial z_l^{(k)}}{\partial w_{pl}^{(k)}} = f_p^{(k)}. \tag{3.53}$$

Підстановка (3.51)–(3.53) у (3.50) дає

$$\Delta w_{pl}^{(k)} = 2\alpha\gamma_{pl}\zeta_l f_l^{(k)}(1 - f_l^{(k)}) f_p^{(k)} \tag{3.54}$$

або

$$w_{pl}^{(k)}(i+1) = w_{pl}^{(k)}(i) + \gamma_{pl}\delta_{pl}^{(k)} f_p^{(k)}, \tag{3.55}$$

де

$$\delta_{pl}^{(k)} = 2\alpha\zeta_l f_l^{(k)}(1 - f_l^{(k)}), \tag{3.56}$$

i — номер ітерації.

Аналогічно обчислюються ваги інших нейронів вихідного шару. Всі величини, що входять до складу алгоритму (3.55), є відомими, тому його реалізація труднощів не викликає. Присутня в (3.54), (3.55) помилка ζ_l відмінна від нуля внаслідок того, що нейрони вихідного шару виробляють помилкові вихідні сигнали,

тому що, по-перше, їм привласнені випадкові вагові коефіцієнти й, по-друге, нейрони прихованих шарів також виробляють помилкові сигнали.

Як ми вже зазначали, наявна помилка ζ_l поширюється назад на попередні приховані шари й використовується для корекції ваг цих шарів.

3.4.2. Обчислення ваг нейронів прихованого шару

Фрагмент мережі для обчислення ваг нейронів прихованого шару наведено на рис. 3.30.

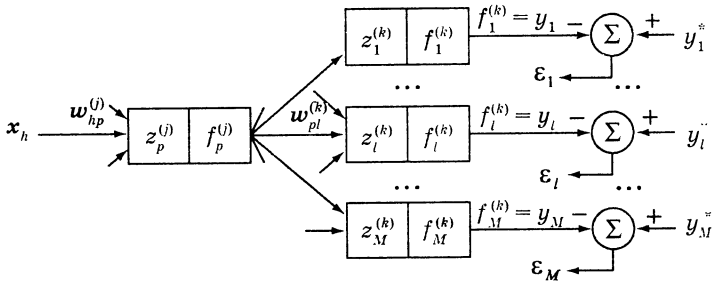


Рис. 3.30. Фрагмент мережі

Розглянемо p-й нейрон прихованого (j-го) шару. Для обчислення ваги $w_{hp}^{(j)}$ даного нейрона також використовується дельта-правило, аналогічне (3.48). Але оскільки вихідний сигнал цього нейрона надходить на виходи всіх нейронів вхідного шару, алгоритм настроювання записується в такий спосіб:

$$\Delta w_{hp}^{(j)} = -\gamma_{hp} \frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = -\gamma_{hp} \sum_{l=1}^{M} \frac{\partial \zeta_l^{(2)}}{\partial w_{hp}^{(j)}}. \qquad (3.57)$$

Тут ζ_l — помилка на i-му виході; γ_{hp} — параметр, що виконує ту саму роль, що й γ_{pl} у (3.55).

Оскільки квадратичний критерій якості в цьому випадку приймає вигляд

$$\zeta = \sum_{l=1}^{M} \left[y_l^* - f_l^{(k)} \right]^2, \qquad (3.58)$$

частинна похідна, використовувана в (3.57), обчислюється так:

$$\frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = \sum_{l=1}^{M} \frac{\partial \zeta_l^2}{\partial f_l^{(k)}} \frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} \frac{\partial z_l^{(k)}}{\partial f_p^{(j)}} \frac{\partial f_p^{(j)}}{\partial z_p^{(j)}} \frac{\partial z_p^{(j)}}{\partial w_{hp}^{(j)}}. \qquad (3.59)$$

Для розглянутого випадку за аналогією з вищевикладеним отримуємо

$$\frac{\partial \zeta_l^2}{\partial f_l^{(k)}} = -2[y_l^* - f_l^{(k)}] = -2\zeta_l; \tag{3.60}$$

$$\frac{\partial f_i^{(k)}}{\partial z_i^{(k)}} = \alpha f_i^{(k)}(1 - f_i^{(k)}); \tag{3.61}$$

$$\frac{\partial z_l^{(k)}}{\partial f_p^i} = w_{pl}^{(k)}; \tag{3.62}$$

$$\frac{\partial f_p^{(j)}}{\partial z_p^{(j)}} = \alpha f_p^{(j)}(1 - f_p^{(j)}); \tag{3.63}$$

$$\frac{\partial z_p^{(j)}}{\partial w_{hp}^{(j)}} = x_h. \tag{3.64}$$

Тут враховано, що

$$z_l^{(k)} = \sum_{p=1}^{m} w_{pl}^{(k)} f_p^{(j)};$$

$$z_l^{(j)} = \sum_{h=1}^{N} w_{hp}^{(j)} x_h,$$

а функції активації є сигмоїдальними.

Таким чином,

$$\frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = \sum_{l=1}^{M}(-2)\alpha\zeta_l\left[f_l^{(k)}(1 - f_l^{(k)})\right] w_{pl}^{(k)} \alpha\left[f_p^{(j)}(1 - f_p^{(j)})\right] x_h =$$

$$= -\sum_{l=1}^{M} \delta_{pl}^{(k)} w_{pl}^{(k)} \frac{\partial f_p^{(j)}}{\partial z_p^{(j)}} x_h, \tag{3.65}$$

де δ_{pl}^{k} визначається виразом (3.56).

Позначимо

$$\delta_{hp}^{(j)} = \delta_{pl}^{(k)} w_{pl}^{(k)} \frac{\partial f_p^{(j)}}{\partial z_p^{(j)}}. \tag{3.66}$$

Тоді

$$\frac{\partial \zeta^2}{\partial w_{hp}^{(j)}} = -\sum_{l=1}^{M} \delta_{hp}^{(j)} x_h \tag{3.67}$$

і алгоритм корекції ваг приймає вигляд

$$\Delta w_{hp}^{(j)} = -\gamma_{hp} x_h \sum_{l=1}^{M} \delta_{hp}^{(j)} \tag{3.68}$$

або

$$w_{hp}^{(j)}(i+1) = w_{hp}^{(j)}(i) + \gamma_{hp} x_h \sum_{l=1}^{M} \delta_{hp}^{(j)}. \tag{3.69}$$

Приклад 3.10. Розглянемо корекцію вагових коефіцієнтів мережі прямого поширення, зображеної на рис. 3.31, за допомогою алгоритму зворотного поширення помилки. Для простоти приймемо, що всі нейрони мають однакову функцію активації з $\alpha = 1$, а коефіцієнт навчання, використовуваний в алгоритмах настроювання ваг, $\gamma = 0{,}5$. Бажаний вихід нейрона E $y^* = 0{,}2$.

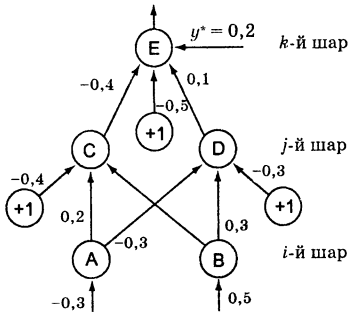


Рис. 3.31. Приклад мережі

Ваги між шарами j й k корегуються за правилом (3.55), а між шарами i й j — за правилом (3.69).

Спочатку обчислимо виходи кожного нейрона з огляду на вигляд функції активації $f = (1 + e^{-\alpha x})^{-1}$ при $\alpha = 1$:

$$z_C = -0{,}3 \cdot 0{,}2 + 0{,}5 \cdot 0{,}1 + 1 \cdot (-0{,}4) = -0{,}06 + 0{,}05 - 0{,}4 = -0{,}41;$$

$$y_C = f_C = 0{,}399$$

$$z_D = -0{,}3 \cdot (-0{,}3) + 0{,}5 \cdot 0{,}3 + 1 \cdot (-0{,}3) = 0{,}09 + 0{,}15 - 0{,}3 = -0{,}06;$$

$$y_D = f_D = 0{,}485$$

$$z_E = 0{,}399 \cdot (-0{,}4) + 0{,}485 \cdot 0{,}1 + 1 \cdot (-0{,}5) = -0{,}16 + 0{,}485 - 0{,}5 = -0{,}17;\ y_E = f_E = 0{,}352$$

Підстановка цих й інших значень у вираз (3.54) і (3.68) дає

$$\Delta w_{CE} = -0{,}5 \cdot (-2) \cdot 1 \cdot (0{,}2 - 0{,}352) \cdot 0{,}352 \cdot (1 - 0{,}352) \cdot 0{,}399 = -0{,}01381;$$

$$\Delta w_{DE} = (0{,}2 - 0{,}352) \cdot 0{,}352 \cdot (1 - 0{,}352) \cdot 0{,}485 = -0{,}01679;$$

$$\Delta w_{1E} = (0{,}2 - 0{,}352) \cdot 0{,}352 \cdot (1 - 0{,}352) \cdot 1 = -0{,}03462;$$

$$\Delta w_{AC} = (0{,}2 - 0{,}352) \cdot 0{,}352 \cdot (1 - 0{,}352) \cdot (-0{,}4) \cdot 1 \cdot 0{,}399 \cdot (1 - 0{,}399) \cdot (-0{,}3) = -0{,}001;$$

$$\Delta w_{AD} = (0{,}2 - 0{,}352) \cdot 0{,}352 \cdot (1 - 0{,}352) \cdot 0{,}1 \cdot 1 \cdot 0{,}485 \cdot (1 - 0{,}485) \cdot (-0{,}3) = 0{,}00026;$$

$$\Delta w_{BC} = (0,2 - 0,352) \cdot 0,352 \cdot (1 - 0,352) \cdot (-0,4) \cdot 1 \cdot 0,399 \cdot (1 - 0,399) \cdot 0,5 = \\ = 0,00166;$$

$$\Delta w_{BD} = (0,2 - 0,352) \cdot 0,352 \cdot (1 - 0,352) \cdot 0,1 \cdot 1 \cdot 0,485 \cdot (1 - 0,485) \cdot 0,5 = \\ = -0,00043;$$

$$\Delta w_{1C} = (0,2 - 0,352) \cdot 0,352 \cdot (1 - 0,352) \cdot (-0,4) \cdot 1 \cdot 0,399 \cdot (1 - 0,399) \cdot 1 = \\ = 0,00332;$$

$$\Delta w_{AD} = (0,2 - 0,352) \cdot 0,352 \cdot (1 - 0,352) \cdot 0,1 \cdot 1 \cdot 0,485 \cdot (1 - 0,485) \cdot 1 = \\ = -0,00086.$$

Додавши ці прирости до існуючих ваг, тобто скориставшись формулами (3.55) і (3.68), отримуємо такі нові значення ваг:

$$\Delta w_{CE} = -0,4 - 0,01381 = -0,414;$$

$$\Delta w_{DE} = 0,1 - 0,01679 = 0,083;$$

$$\Delta w_{1E} = -0,5 - 0,03462 = -0,535;$$

$$\Delta w_{AC} = 0,2 - 0,001 = 0,199;$$

$$\Delta w_{AD} = -0,3 + 0,00026 = -0,03;$$

$$\Delta w_{BC} = 0,1 + 0,00166 = 0,102;$$

$$\Delta w_{BD} = 0,3 - 0,00043 = -0,03;$$

$$\Delta w_{1C} = -0,4 + 0,00332 = -0,397;$$

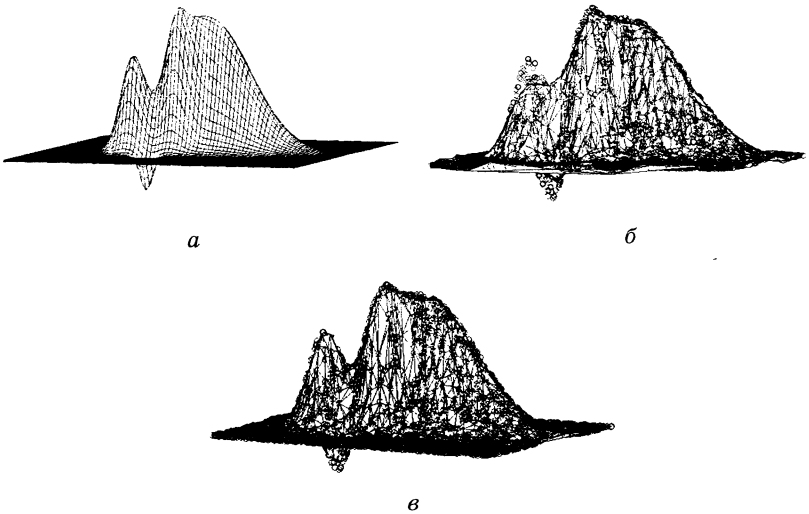
$$\Delta W_{AD} = -0,3 - 0,00086 = -0,301.$$

Приклад 3.11. На рис. 3.32, 3.33 наведено результати апроксимації функції (рис. 3.32, *а*)

$$f(x,y) = 0,5\exp\left[-\frac{(9x-2)^2 + (9y-2)^2}{4}\right] + 0,75\exp\left[-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}\right] + \\ + 0,5\exp\left[-\frac{(9x-7)^2 + (9y-3)^2}{4}\right] - 0,2\exp\left[-(9x-4)^2 - (9y-7)^2\right],$$

багатошаровим персептроном з одним (рис. 3.32,*б*) і двома (рис. 3.33) прихованими шарами. У першому випадку прихований шар містив 30 нейронів, у другому — перший прихований шар — 10, а другий — 6 нейронів, тобто двошаровий персептрон мав вигляд 2-10-6-1. Ця функція була промодельована із

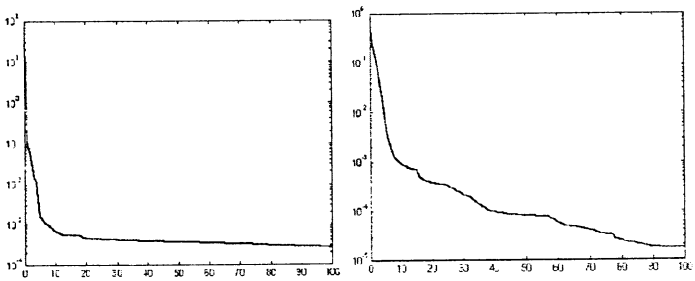
кроком дискретизації 0,1. Під час навчання багатошарового персептрона за допомогою алгоритму зворотного поширення було згенеровано 5000 двовимірних випадкових величин. Самі пари, що навчають, показано на рисунку точками. Задовільна збіжність процесу навчання була досягнута після 100 тактів.



а *б* *в*

Рис. 3.32. Апроксимація функції $f(x, y)$
за допомогою багатошарового персептрона

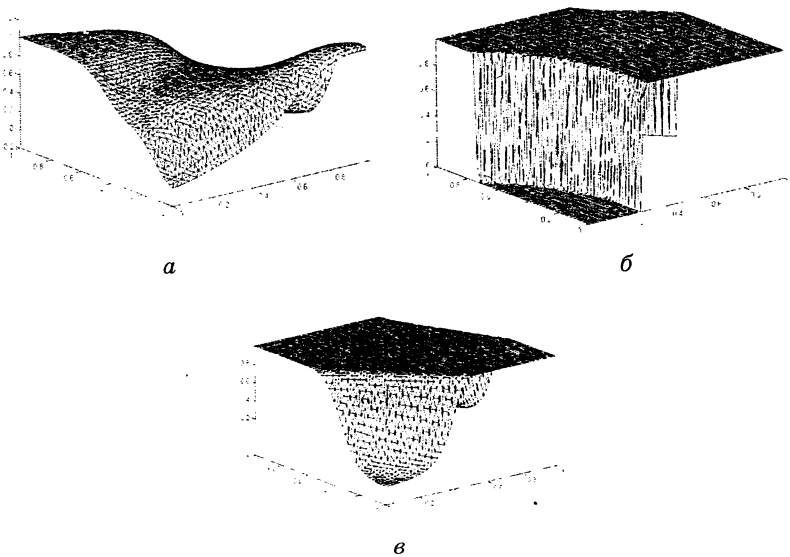
Зміну величин помилок апроксимації зображено на рис. 3.33.



а *б*

Рис. 3.33. Зміна помилок для персептрона, що містить:
а — один і *б* — два прихованих шари

Приклад 3.12. На рис. 3.34 наведено результати моделювання логічної функції «що виключає АБО» на персептроні 2-7-2-1, що має два прихованих шари, причому рисунки *а*, *б* та *в* відбивають реалізації даних функцій на персептронах з активаційними функціями (2.7), (2.13) і (2.14) відповідно.



а *б* *в*

Рис. 3.34. Реалізація функції «що виключає АБО» на персептроні 2-7-2-1

Слід зазначити, що призначення випадкових початкових значень вагових параметрів призводить до того, що при повторному моделюванні з іншими початковими значеннями форма поверхонь може бути іншою.

Контрольні запитання та завдання

1. Що являє собою одношаровий персептрон?
2. Поясніть теорему збіжності для персептрона.
3. Які можливості одношарового персептрона?
4. У чому відмінність Адаліни від персептрона?
5. Які існують різновиди Адаліни?