Міністерство освіти і науки України
ДВНЗ "Ужгородський національний університет"
Факультет інформаційних технологій
Кафедра інформаційних управляючих систем та технологій

Коцовський В. М.

Спектральний аналіз дискретних нейрофункцій
Частина II

Методичний посібник

Ужгород – 2019

ЗМІСТ

ВСТУП

В підручнику [1] та монографіях [2, 3] зазначено, що на сьогоднішній день основними є два підходи до розробки систем штучного інтелекту у складі сучасних інтелектуальних інформаційних систем:

1) низхідний (Top-Down AI, семіотичний, символьний) — створення експертних систем, баз знань та систем логічного виведення, що моделюють та імітують високорівневі психічні процеси: мислення, міркування, мова, емоції, творчість і т.п.;
2) висхідний (Bottom-Up AI, біологічний, конекціоністський) — вивчення нейронних мереж і еволюційних обчислень, які моделюють інтелектуальну поведінку на основі біологічних елементів, а також створення відповідних обчислювальних систем, таких як нейрокомп'ютери.

Курс за вибором «Спектральний аналіз дискретних нейрофункцій», який протягом двох семестрів читається аспірантам спеціальності 122 «Комп'ютерні науки», присвячений вивченню основних понять, моделей та методів, які були запропоновані у межах висхідного підходу до розробки систем штучного інтелекту. Основну увагу приділено дослідженню нейромережевої парадигми у штучному інтелекті. Наведено основні моделі штучних нейроелементів та нейромереж, а також описано найбільш розповсюджені методи їх навчання.

Очевидно, що матеріал посібника не вичерпує усе різноманіття результатів і застосувань теорії штучних нейронних мереж. Додаткові відомості можуть бути знайдені в [3–11]. В [12, 13] наведено сучасні методи машинного навчання з використанням високоефективних глибинних нейромереж. Матеріал першої частини курсу за вибором наведено в [14].

1. КОНЕКЦІОНІСТСЬКІ МОДЕЛІ ТА МЕТОДИ

1.1. Загальна характеристика конекціоністського підходу та його місце в теорії інтелектуальних систем

Конекціоністський підхід до побудови систем штучного інтелекту розвинувся на противагу символьному, що є характерним для сучасних моделей знань. В основі конекціоністського підходу лежить спроба безпосереднього моделювання розумової діяльності людського мозку. Відомо, що мозок людини складається з величезної кількості нервових клітин (нейронів), що взаємодіють між собою. Ці "обчислювальні елементи" мозку функціонують набагато повільніше, ніж обчислювальні елементи комп'ютерних систем. Але ефективність людського інтелекту досягається як за рахунок паралельної роботи нейронів, так і за рахунок того, що механізми їх взаємодії були вироблені шляхом тривалої еволюції.

Для багатьох цілей нейрон можна розглядати як елемент з певним критичним значенням. Це означає, що він або ж дає на виході деяку постійну величину, якщо сума його входів досягає певного значення, або ж залишається пасивним.

Мак-Каллок і Піттс довели, що будь-яку обчислювану функцію можна реалізувати за допомогою спеціально організованої мережі ідеальних нейронів, логічні властивості яких з високою достовірністю можна приписати реальному нейрону. Але ця мережа буде мати наступні вади. По-перше, проблема полягає в тому, чи можна знайти якийсь розумний принцип реорганізації мережі, який дозволяв би випадково об'єднаній, спочатку, групі ідеальних нейронів самоорганізовуватись в "обчислювальний пристрій", здатний вирішувати довільну задачу розпізнання. По-друге, потрібно використовувати велику кількість нейронів. Так, модель мурашки потребує використання близько 20000 нейронів, людини — 100 млрд. нейронів, що на практиці неможливо.

Нейрологічна теорія стала також основою системи розпізнавання, яка дістала назву *перцептрон*. В цьому підході основна увага приділялась встановленню характеристик, приписаних фіксованій множині детекторів ознак. Альтернативний підхід розпізнавання зводиться до пошуку "*добрих"* ознак, на основі яких розпізнавання здійснюється найбільш чітко. Наприклад, перцептрон Розенблатта передавав повідомлення від "ока", яке реалізовувалось системою фотоелементів, в блоки електромеханічних комірок пам'яті, які оцінювали відносні величини електричних сигналів. Ці комірки з'єднувались між собою випадковим чином, створюючи мережу з прямими зв'язками. Перцептрон міг навчатись шляхом спроб і помилок, а також корекцією електричних імпульсів.

Мінські і Пейпертом було математично доведено, що перцептрони не в змозі розв'язувати деякі задачі, наприклад, розпізнавання частково затулених предметів.

Один з сучасних напрямків створення розумних машин — це розробка *нейрокомп'ютерів*. *Нейрокомп'ютер* — це програмно-технічна система (спеціалізована ЕОМ), яка реалізує деяку формальну модель природної мережі нейронів.

В основу машин п'ятого покоління покладено ідею паралельної обробки інформації в нейроподібних системах. Не зважаючи на те, що електронний процесор працює в тисячі разів швидше, ніж його нейронний еквівалент у мозку, мережі нейронів вирішують багато задач в тисячі раз швидше, ніж електронний процесор.

Причини цього такі:

1) Характер взаємозв'язків між нейронами дозволяє розв'язувати багато задач на основі паралельної обробки;

2) У нейронній мережі пам’ять не локалізована в одному місці (як в послідовних машинах), а розподілена по всій структурі. В біологічних системах пам’ять реалізується підсиленням або послабленням зв’язків між нейронами, а не зберіганням двійкових символів;
3) Біологічні мережі реагують не на всі, а тільки на визначені зовнішні подразнення. Кожний нейрон виступає як елемент прийняття рішення і як елемент зберігання інформації. Перевага такої структури — “життєздатність” (вихід з ладу декількох нейронів не приводить до значної зміни даних, що зберігаються, або ж до руйнування всієї системи).
4) Можливість адресації за вмістом (асоціативної пам’яті) є ще однією важливою характеристикою систем з розподіленою пам’яттю (кожний елемент відшукується за його вмістом, а не зберігається в комірці пам’яті з визначеним номером).

В основу зв’язків в нейрокомп’ютерах покладено *принцип асоціацій*. *Асоціативні зв’язки* пронизують все мислення людини. Існує думка [1], що *процеси мислення є не що інше, як розповсюдження певного збудження, як деяка ланцюгова реакція*. Навіть найбільш примітивні процеси навчання принципово залежать від послідовності подій в часі. Це й закладено в природу нейронних систем. Тому їм притаманне реагування тільки на жорстко визначені зовнішні подразнення. Наприклад, домашні тварини "навчаються" ігнорувати повторні несуттєві зовнішні подразнення ("цокання" годинника), але посилюють сприйняття подразнень, які можуть мати серйозні наслідки (звук автомобільних гальм).

Багато дослідників вважає, що майбутнє належить комп’ютерам, які базуються на аналізі зв’язків, а не обробці символів [11]. Мінські вважав, що якщо комп’ютер повинен діяти подібно мозку, тоді й його конструкція повинна бути також подібна до мозку.

Моделі штучних нейронних мереж (ШНМ) і схеми з адресацією за вмістом мають і недоліки. Внаслідок нефіксованої організації вони можуть плутати різні об’єкти. Але це аналогічно звиканню, посиленню чуттєвості до асоціацій, які лежать в основі психічних особливостей людини. Іншими словами, будь-який комп’ютер, який претендує на "розумність" повинен мати такі особливості.

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 [11].

1.2. Модель штучного нейрона

Структуру штучного нейрона, запропоновану у [11], зображено на рис. 1.1.

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

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

$$y = f_{\text{вих}}(f_{\text{а}}\,(f_{\text{вх}}(\boldsymbol{x}, \boldsymbol{w}))) \tag{1.1}$$

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

Рис. 1.1. Структура штучного нейрона

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

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

— максимальне значення зважених входів

$$f(\mathbf{x}, \mathbf{w}) = \max_i (w_i x_i);$$

— мінімальне значення зважених входів

$$f(\mathbf{x}, \mathbf{w}) = \min_i (w_i x_i).$$

1.2.1. Функція активації

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

$$z(k+1) = f_a(z(k), f_{\text{вх}}(\mathbf{x}, \mathbf{w})).$$

Надалі позначатимемо функцію активації без індексу *«a»*. Найбільш простими активаційними функціями є

— лінійна

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

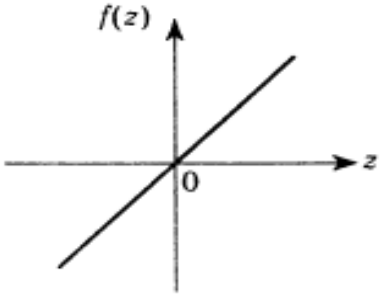


Рис. 1.2. Лінійна функція

— лінійна біполярна з насиченням

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

Незважаючи на те, що лінійні функції є найбільш простими, їхнє застосування обмежене. 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, використання лінійних активаційних функцій не є зайвим у багатошарових ШНМ, для розширення ж можливостей мережі застосовують нелінійні функції активації.

У роботі У. Мак-Каллока і У. Піттса у якості активаційної використовувалася *функція Хевісайда* — уніполярна *порогова* гранична функція вигляду

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

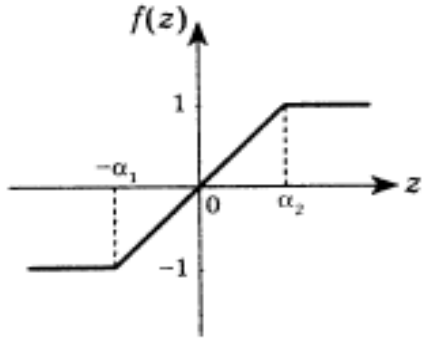


Рис. 1.3. Лінійна біполярна функція з насиченням

Різновидом даної функції є біполярна порогова функція

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

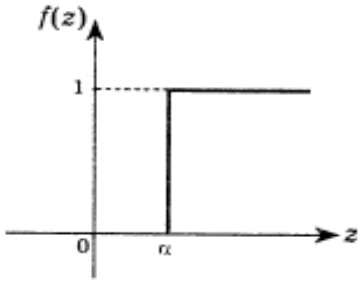


Рис. 1.4. Уніполярна порогова функція

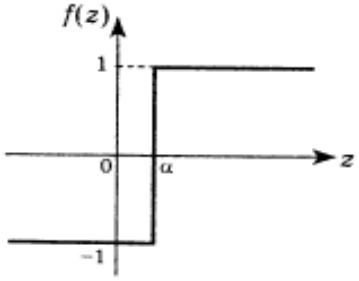


Рис. 1.5. Біполярна порогова функція

Ці функції активації застосовувалися в основному в класичних ШНМ. При побудові нових структур ШНМ найчастіше доводиться працювати як із самою акти-

ваційною функцією, так і з її першою похідною. У цих випадках необхідним є використання як активаційної монотонної диференційованої й обмеженої функції. Це так звані *логістичні*, або *сигмоїдальні* (*S*-подібні), функції. Функція називається сигмоїдальною, якщо вона є монотонно зростаючою, диференційованою і задовольняє умову

$$\lim_{\lambda\to-\infty} f(\lambda) = k_1, \qquad \lim_{\lambda\to+\infty} f(\lambda) = k_2, \qquad k_1 < k_2.$$

До таких функцій належать:
— логістична (уніполярна)

$$f_{\log}(z) = \frac{1}{1+e^{-kz}}; \tag{1.2}$$

— гіперболічний тангенс (біполярна)

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

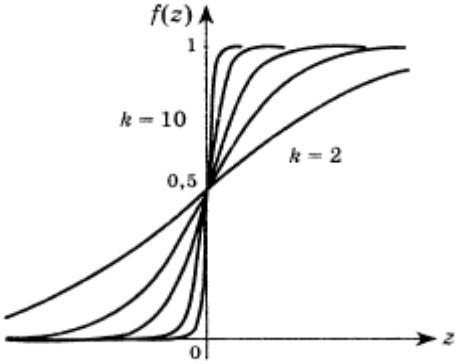


Рис. 1.6. Логістична функція

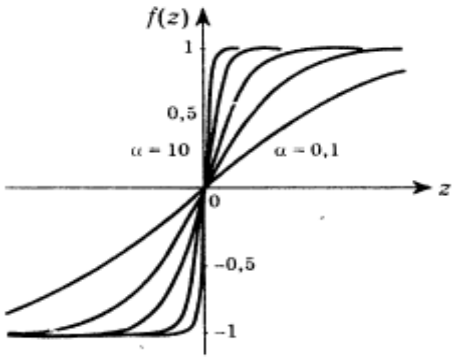


Рис. 1.7. Гіперболічний тангенс

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

1.2.2. Формальна модель нейрона Маккаллока-Піттса

Формальний штучний нейрон (його називають також нейроном Мак-Каллока-Піттса) [11] може бути поданий як нелінійний перетворювач із ваговими коефіцієнтами w_{ji}, які також називаються синаптичними вагами або підсилювачами. Клітина тіла (сома) описується нелінійною обмежувальною або пороговою функцією $f(u_j)$. Найпростіша модель штучного нейрона додає N зважених входів і здійснює нелінійне перетворення (див. рис. 1.1)

$$y_j = f\left(\sum_{i=1}^{N} w_{ji}x_i + \theta_j\right), \tag{1.3}$$

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 $x_0 = 1$) формулу (1.3) можна переписати у вигляді

$$y_j = f\left(\sum_{i=0}^{N} w_{ji}\, x_i\right) = f\left(\mathbf{w}_j^T x\right),$$

де $x = (1, x_1, \ldots, x_N)^T$; $w_j = (w_{j0}, w_{j1}, \ldots, x_{jN})^T$ — відповідні вектори входів і ваг.

2. АРХІТЕКТУРА ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

2.1. Поняття штучної нейромережі

З'єднані між собою нейрони утворюють *штучну нейронну мережу* (ШНМ). Таким чином, ШНМ — пара (N, C), де N — множина нейронів; C — 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. 2.1). Така пошарова організація є аналогом шаруватих структур певних відділів мозку.

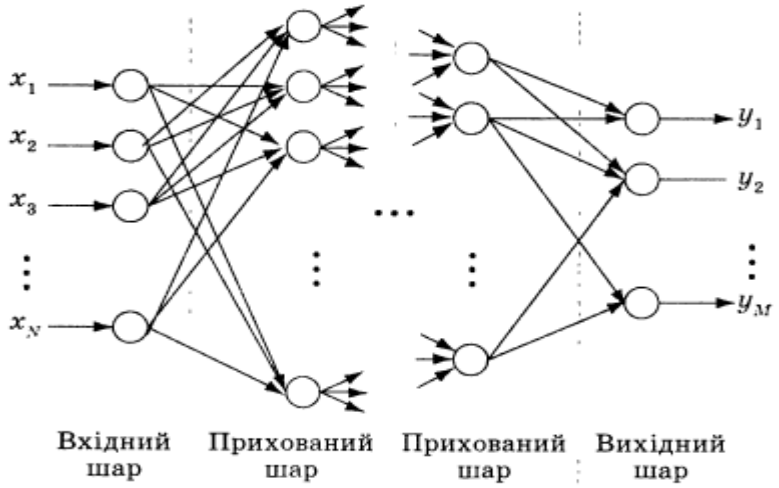


Рис. 2.1. Структура ШНМ

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.

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

- ШНМ без зворотних зв'язків (прямого поширення, Feed forward)
- 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 [1].

2.2. ШНМ прямого поширення

ШНМ прямого поширення 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. 1.10. Якщо в мережі цього типу кожен нейрон шару і пов’язаний з кожним нейроном (i+1)-го шару, мережа називається повнозв’язною прямого поширення.

У мережах другого порядку поряд із 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. 2.2.

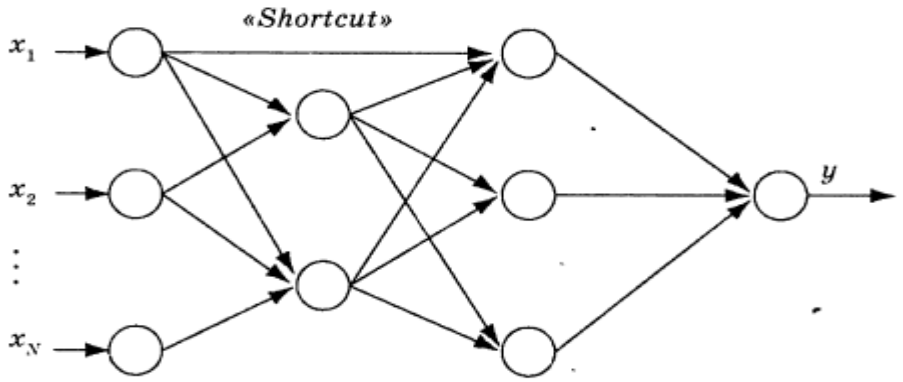


Рис. 2.2. ШНМ прямого поширення другого порядку

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

2.3. 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. 2.3) 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.

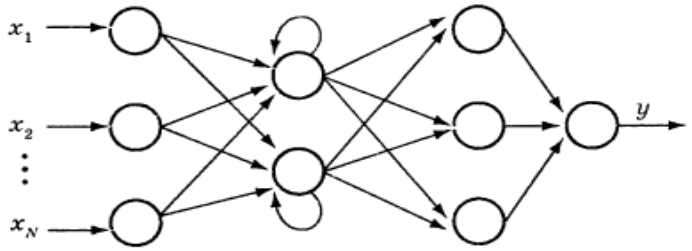


Рис. 2.3. ШНМ із прямими зворотними зв’язками

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 звя’зків необхідно, щоб виділити певну особливо важливу для даної ШНМ область вхідних сигналів (рис. 2.4).

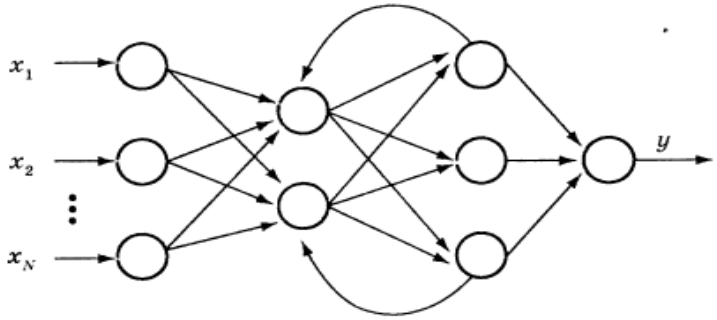


Рис. 2.4. ШНМ із непрямими зворотними зв’язками

ШНМ із латеральними зв’язками має зв’язки між нейронами одного шару (рис. 2.5). 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 (послаблюючий, 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).

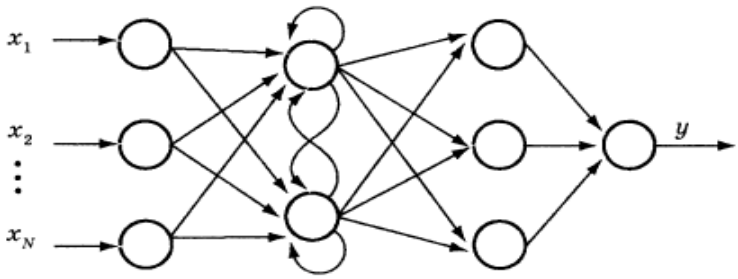


Рис. 2.5. ШНМ із латеральними зв’язками

2.3.1. Повнозв’язні ШНМ

Повнозв’язні ШНМ характеризуються наявністю зв’язків між усіма нейронами

мережі (рис. 2.6). Цей вид топології відомий також як мережа Хопфілда.

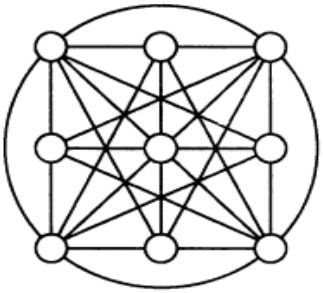


Рис. 2.6. Повнозв’язна ШНМ

3. НАВЧАННЯ ШНМ

3.1. Поняття про 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. Штучний нейрон може бути представлений у такий спосіб (рис. 3.1).

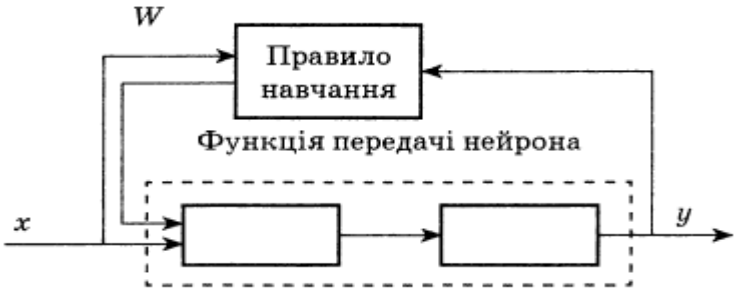


Рис. 3.1. Модель штучного нейрона

У процесі навчання ваги стають такими, що під час надходження вхідних сигналів мережа виробляє відповідні необхідні вихідні сигнали. Розрізняють 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 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 множини 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.

3.2. Правило навчання Гебба (корелятивне, співвідносне навчання)

Більшість сучасних алгоритмів навчання виросло із правила Гебба. 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 клітинами. 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

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

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

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 **y** = **x**. У гетероасоціативних мережах **x** й **y** в загальному випадку різняться.

3.3. Дельта-правило

Це важливе правило навчання було запропоновано Б. Уідроу й М. Е. Гоффом 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 зменшена 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:

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

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.

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.

3.4. 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

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

де γ — коефіцієнт, що впливає на швидкість навчання;

$$\nabla_{\boldsymbol{w}} I(\boldsymbol{w}) = \frac{\partial I(\boldsymbol{w})}{\partial \boldsymbol{w}_{ij}}.$$

Ці методи найчастіше використовуються при контрольованому навчанні, коли відома необхідна реакція нейронів $\boldsymbol{y}^*$.

Більшість градієнтних методів засновано на мінімізації квадратичного функціонала

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

У цьому випадку

$$\nabla_{\boldsymbol{w}} I(\boldsymbol{w}) = -e(k)\boldsymbol{x}(k).$$

Таким чином приходимо до алгоритму методу найменших квадратів (МНК)

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

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

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

зміст яких полягає в тому, що для забезпечення збіжності послідовності 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, ряд $\gamma(k) = \gamma_0 k^{-\alpha}$, де γ_0 — деяка константа, $0{,}5 < \alpha \leq 1$.

3.5. Одношаровий перцептрон

3.5.1. Будова перцептрона

Значний інтерес до перцептронів викликаний роботою Ф. Розенблатта, у якій він досліджував нейромережеву модель сітківки (RETINA) — фотоперцептрон. Перцептрон зображений на рис. 3.2 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.

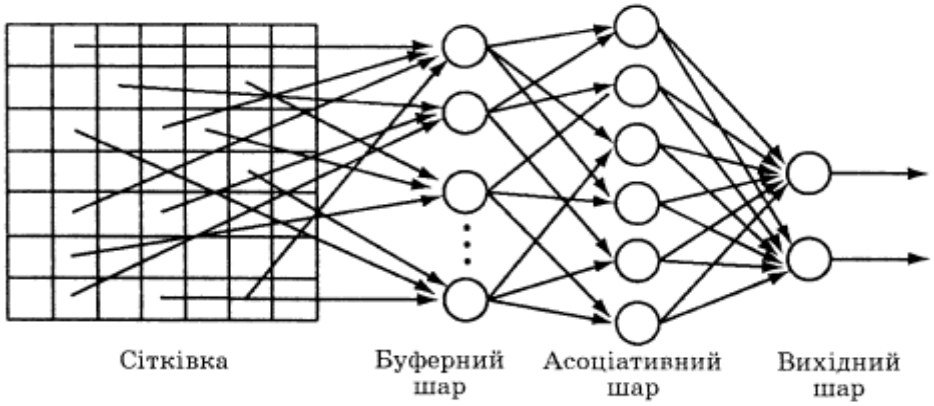


Рис. 3.2. Перцептрон Розенблатта

Результат попередньої 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 сигналів вони посилають імпульси на вихідний шар, де й відбувається додавання зважених імпульсів. У вихідному шарі використовуються уніабо біполярна активаційні функції. 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 перцептрон може розглядатися як двошарова ШНМ прямого поширення.

3.5.2. Навчання перцептрона

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

Схему навчання перцептрона наведено на рис. 3.3.

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

$$y_p = \operatorname{sgn}\left(\mathbf{w}^{*T}\mathbf{x}_\mathrm{p}\right) = y_p^*, \qquad p = \overline{1,P}.$$

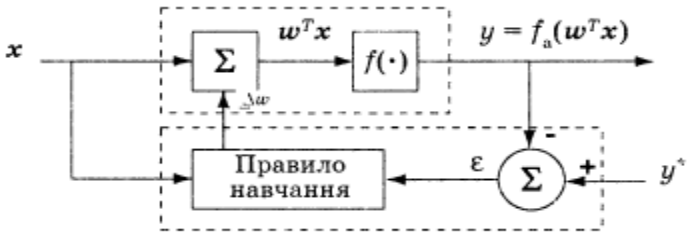


Рис. 3.3. Схема навчання перцептрона

У цьому випадку вектор $\mathbf{w}^*$ забезпечить правильну класифікацію перцептроном усіх пар, що навчають, із поданої множини (завершується цикл навчання).

Гіперплощина $\mathbf{w}^{*T}\mathbf{x}_p = 0$ ділить вхідний простір на два підпростори. Для $y_p^* = \mathbf{1}$ має виконуватися умова $\mathbf{w}^{*T}\mathbf{x}_p > 0$, а для $y_p^* = -1$ — умова $\mathbf{w}^{*T}\mathbf{x}_p < 0$.

Алгоритм навчання може бути записаний у такий спосіб:

$$\mathbf{w}_{p+1} = \mathbf{w}_p + \gamma e_p \mathbf{x}_p,$$

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

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

У режимі offline у M-му циклі (епосі) навчання подаються всі пари $(\mathbf{x}_p, y_p^*)$ й обчислюється середнє значення помилки класифікації

$$\overline{e}_m = \frac{1}{P}\sum_{p=1}^{p}(y_{p,M}^* - y_{p,M}),$$

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.

Теорема збіжності для перцептрона, доведена Розенблаттом, стверджує, що *перцептрон може навчитися правильно класифікувати подані йому образи на скінченному числі навчальних пар*.

3.6. Алгоритм зворотного поширення помилки навчання багатошарових нейромереж прямого поширення

Алгоритм зворотного поширення помилки описаний у [11]. 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, які мають відмінні від нуля похідні на всій області визначення, забезпечує правильне навчання й функціонування мережі. Процес навчання полягає у послідовному поданні мережі пар $\left(\mathbf{x}(i), y^*(i)\right)\ i = \overline{1,P}$, що навчають, де $\mathbf{x}(i)$і $y^*(i)$ — вектор вхідних і бажаних вихідних сигналів мережі відповідно, вивчені реакції на них мережі й корекції відповідно до реакції вагових параметрів (елементів вагової матриці).

Перед початком навчання всім вагам присвоюють невеликі 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 навчальну пару $\left(\mathbf{x}(i), \mathbf{y}^*(i)\right),\ i = \overline{1,P}$ та подати на вхід мережі вхідний сигнал $\mathbf{x}(i)$.
2. Обчислити реакцію мережі $\mathbf{y}(i)$.
3. Визначити помилку $\mathbf{y}^*(i) - \mathbf{y}(i)$.
4. Скорегувати ваги так, щоб помилка була мінімальною.
5. Кроки 1-4 повторити для всієї множини пар $\left(\mathbf{x}(i), \mathbf{y}^*(i)\right)\ i = \overline{1,P}$ 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.

Алгоритм зворотного поширення застосовують також щодо мереж з будь-якою кількістю шарів: як мереж прямого поширення, так і таких, що містять зворотні зв’язки. Обмежимося розглядом випадку навчання двох шарів мережі прямого поширення. Фрагмент такої мережі зображено на рис. 3.4.

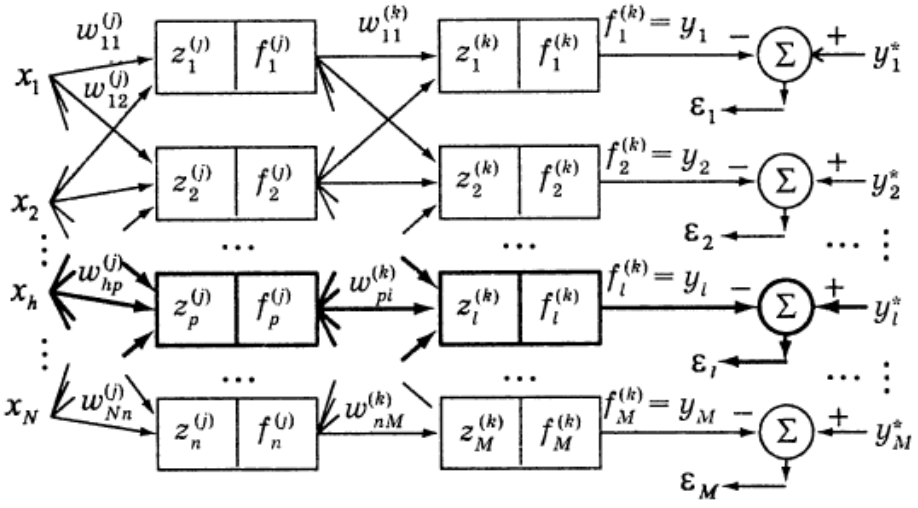


Рис. 3.4. Фрагмент мережі прямого поширення

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 j-го шару; $z_p^{(j)}$ — зважена сума вихідних сигналів попереднього (i-го) шару, що надходять на вхід p-го нейрона наступного (j-го) шару.

Обчислення ваг нейронів вихідного шару

Розглянемо l-й нейрон вихідного (k-го) шару. Вихід даного нейрона є виходом мережі, тому його сигнал $y_i = f_l^{(k)}$ порівнюється з необхідним y_l^* й обчислюється помилка

$$\varsigma_l = y_l^* - f_l^{(k)}.$$

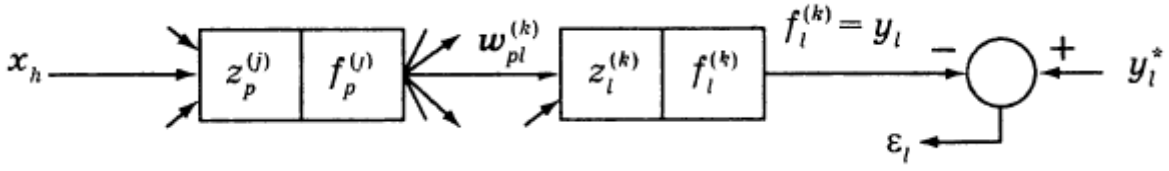


Рис. 3.5. Фрагмент багатошарової мережі

Використання квадратичного критерію якості навчання

$$\varsigma_l^2 = (y_l^* - f_l^{(k)})^2 \tag{3.1}$$

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

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

де γ_{pl} — коефіцієнт, що впливає на швидкість навчання. Обчислення складної частинної похідної здійснюється за правилом

$$\frac{\partial \varsigma_l^2}{\partial w_{pl}^{(k)}} = \frac{\partial \varsigma_l^2}{\partial f_l^{(k)}} \frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} \frac{\partial z_l^{(k)}}{\partial w_{pl}^{(k)}}. \tag{3.2}$$

З урахуванням (3.1), виду активаційних функцій і того, що $z_l^{(k)} = \sum_{p=1}^{n} w_{pl}^{(k)} f_p^{(j)}$, отримуємо

$$\frac{\partial \varsigma_l^2}{\partial f_l^{(k)}} = -2(y_l^* - y_l) = -2\varsigma_l; \tag{3.3}$$

$$\frac{\partial f_l^{(k)}}{\partial z_l^{(k)}} = \alpha f_l^{(k)}(1 - f_l^{(k)}); \tag{3.4}$$

$$\frac{\partial z_l^{(k)}}{\partial w_{pl}^{(k)}} = f_p^{(k)}. \tag{3.5}$$

Підстановка (3.3)-(3.5) у (3.2) дає

$$\Delta w_{pl}^{(k)} = 2\alpha\gamma_{pl}\varsigma_l f_l^{(k)}(1 - f_l^{(k)})f_p^{(k)} \tag{3.6}$$

або

$$w_{pl}^{(k)}(i+1) = w_{pl}^{(k)}(i) + \gamma_{pl}\delta_{pl}^{(k)} f_p^{(k)}, \tag{3.7}$$

де $\delta_{pl}^{(k)} = 2\alpha\varsigma_l f_l^{(k)}(1 - f_l^{(k)})$, i — номер ітерації.

Аналогічно обчислюються ваги інших нейронів вихідного шару. Всі величини, що входять у (3.7), є відомими. Присутня в (3.6), (3.7) помилка ς_l відмінна від нуля внаслідок того, що нейрони вихідного шару виробляють помилкові вихідні сигнали, тому що, по-перше, їм присвоєнні випадкові вагові коефіцієнти й, по-друге, нейрони прихованих шарів також виробляють помилкові сигнали. Помилка ς_l поширюється назад на попередні приховані шари й використовується для корекції ваг цих шарів.

Обчислення ваг нейронів прихованого шару

Фрагмент мережі для обчислення ваг нейронів прихованого шару наведено на рис. 3.6.

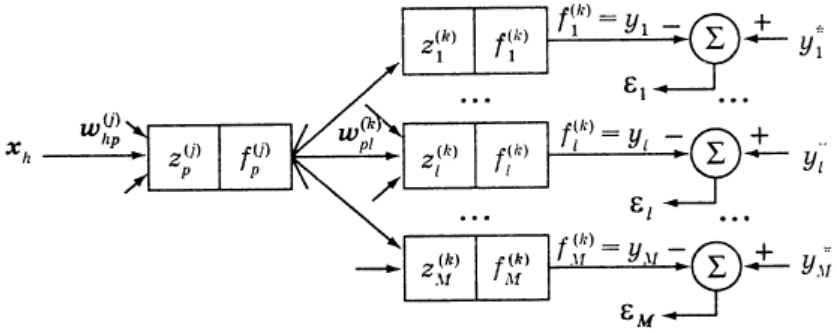


Рис. 3.6. Фрагмент мережі

Розглянемо p-й нейрон прихованого (j-го) шару. Оскільки вихідний сигнал цього нейрона надходить на виходи всіх нейронів вхідного шару, алгоритм настроювання записується в такий спосіб:

$$\Delta w_{hp}^{(j)} = -\gamma_{hp}\frac{\partial\varsigma^2}{\partial w_{hp}^{(j)}} = -\gamma_{hp}\sum_{i-1}^{M}\frac{\partial\varsigma_l^{(2)}}{\partial w_{hp}^{(j)}}. \tag{3.8}$$

Тут ς_l — помилка на i-му виході; γ_{hp} — параметр, що виконує ту саму роль, що й γ_{pl} у (3.7).

Оскільки квадратичний критерій якості в цьому випадку приймає вигляд

$$\varsigma = \sum_{l-1}^{M}[y_l^* - f_l^{(k)}]^2,$$

частинна похідна в (3.8), обчислюється так:

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

Для розглянутого випадку за аналогією з вищевикладеним отримуємо

$$\frac{\partial \varsigma_l^2}{\partial f_l^{(k)}} = -2\left[y_l^* - f_l^{(k)}\right] = -2\varsigma_l;$$

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

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

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

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

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,

$$\frac{\partial \varsigma^2}{\partial w_{hp}^{(j)}} = \sum_{i=1}^{m}(-2)\alpha\varsigma_l\left[f_l^{(k)}\left(1 - f_l^{(k)}\right)\right] w_{pl}^{(k)} \alpha\left[f_p^{(j)}\left(1 - f_p^{(j)}\right)\right] x_h =$$

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

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

Тоді

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

і алгоритм корекції ваг приймає вигляд

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

або

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

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

Опис бібліотек, у яких реалізовано різні модифікації методів навчання штучних нейромереж, наведено у [12, 13].

4. МЕРЕЖА ХОПФІЛДА

4.1. Модель Хопфілда

Розроблена Хопфілдом модель асинхронної ШНМ має такі ознаки [11]:

1. Мережа є одношаровою й містить N нейронів, число яких є одночасно числом входів і виходів мережі.
2. Кожен нейрон мережі пов'язаний з усіма іншими нейронами, а також має один вхід, на який подається вхідний сигнал.
3. Кожен нейрон не має власного зворотного зв'язку ($w_{ii} = 0$).
4. Ваги мережі є симетричними, тобто вага зв'язку між i-им й j-им нейронами дорівнює вазі зв'язку між j-им й i-им нейронами $(w_{ij} = w_{ji})$.
5. Кожен нейрон має порогову функцію активації.
6. Вхідними є двійкові сигнали.

Слід зазначити, оскільки «мережева» архітектура мозку містить зворотні зв'язки, то функціонування мережі Хопфілда, яка задовольняє обмеженням 1–6, відповідає природному процесу обробки інформації.

Схема мережі Хопфілда наведена на рис. 4.1. Внаслідок симетрії мережі Хопфілда іноді використовують різні форми її зображення. Так, наприклад, мережі чотирма нейронами, структурна схема якої зображена на рис. 4.1 відповідає графічне зображення, наведене на рис. 4.2.

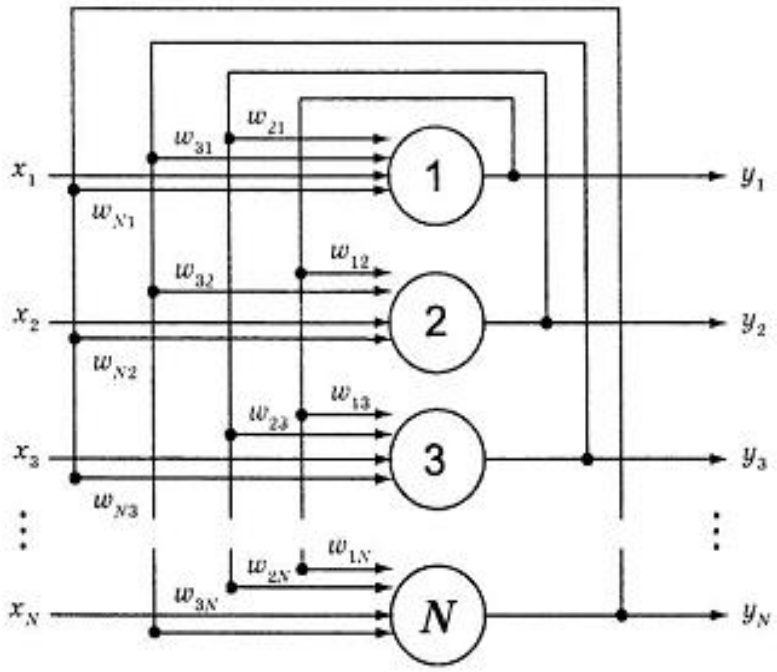


Рис. 4.1. Структурна схема мережі Хопфілда

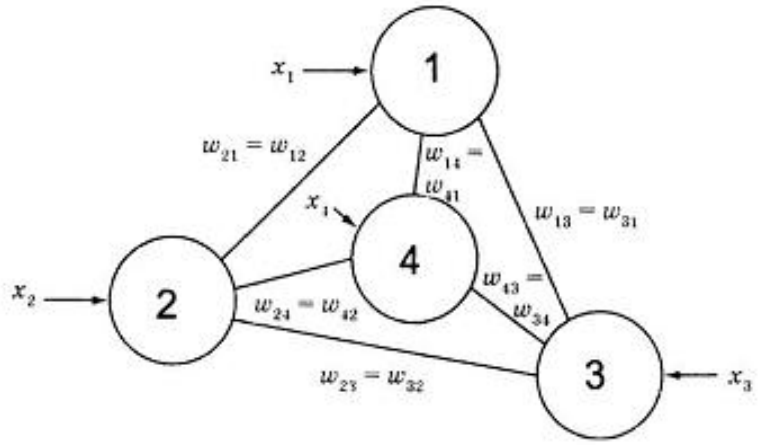


Рис. 4.2. Графічне зображення мережі Хопфілда

Дана мережа не використовує ані навчання з учителем, ані навчання без учителя. Вагові коефіцієнти в ній розраховуються тільки перед початком функціонування мережі на основі інформації про оброблювані дані, і все навчання мережі зводиться саме до цього розрахунку. З одного боку, подання апріорної інформації можна оцінювати як допомогу вчителя, а з іншого – мережа фактично запам'ятовує образи до того, як на її вхід надходять реальні дані, і не може змінювати свою поведінку, тому говорити про зворотний зв'язок із учителем не доводиться.

Слід зазначити, що в мережах зі зворотними зв'язками стан нейронів обчислюється доти, поки вони не виявляться сталими, не змінюваними згодом. Можна сказати, що мережа Хопфілда за певних умов збігається до сталого стану за скінченний час.

Мережі подаються P образів, які зображуються у вигляді N-вимірних векторів $x = (x_1, x_2, \dots, x_N)^T$, де N — кількість нейронів. Компоненти векторів приймають двійкові (0 та 1) або біполярні значення, наприклад (–1 та 1). Позначимо вектор вхідних сигналів, що описує p-й образ, як x^p $(p = \overline{1, P})$. Коли мережа розпізнає поданий образ, вона формує вектор вихідних сигналів, компоненти якого співпадають з відповідними компонентами образа. Тобто $y = x^p$, де $y = (y_1, y_2, \dots, y_N)$ — вектор вихідних сигналів мережі. В іншому випадку вихідний сигнал не збігатиметься з жодним із поданих.

Розрізняють *три стадії (фази) функціонування мережі* [11]:

- Ініціалізація;
- Подання вхідного образу;
- Обчислення стану нейронів.

На стадії *ініціалізація мережі* встановлюють значення вагових коефіцієнтів [11, 12]:

$$w_{ij} = \begin{cases} \sum_{p=0}^{P} x_i^p x_j^p, & i \neq j; \\ 0, & i = j \end{cases} \qquad (4.1)$$

Тут x_i^p й x_j^p — i-та j-та компоненти вектора x^p.

Цю стадію можна розглядати як стадію навчання мережі, після завершення якої вона здатна правильно відтворювати подані їй образи.

Подання мережі вхідного образу фактично здійснюється безпосереднім початковим (нульовим) присвоєнням компонент вихідних сигналів

$$y_i(0) = x_i, \quad i = \overline{1,N}, \tag{4.2}$$

Тому позначення на схемі мережі вхідних сигналів у явному вигляді носить суто умовний характер. Далі обчислюються величини

$$z_j(k+1) = \sum_{i=1}^{N} w_{ij} y_i(k) + x_j. \tag{4.3}$$

Визначаються послідовні стани нейронів, на підставі чого розраховуються відповідні значення вихідних сигналів

$$y_j(k+1) = f_a\left(z_j(k+1)\right) = \begin{cases} 1, & \text{якщо } z_j(k+1) > \theta_j; \\ 0, & \text{якщо } z_j(k+1) < \theta_j; \\ y_j(k), & \text{в інших випадках,} \end{cases} \tag{4.4}$$

де f_a — порогова активаційна функція; θ_j – заданий поріг.

Якщо вихідні значення змінилися, то за формулами (4.3) і (4.4) знову розраховуються стани й вихідні сигнали мережі, якщо не змінилися — робота мережі завершується (мережа перебуває в стійкому стані). У цьому випадку на виходах мережі сформувався вектор, що якнайкраще відповідає поданому образу.

Приклад 4.1. Розглянемо, до чого призведе порушення умов $w_{ii} = 0$. Припустимо, що є мережа, яка складається з двох нейронів, з матрицею ваг

$$w = \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix}$$

і порогами $\theta_l = -1 \quad (i = 1,2)$. Нехай у деякий момент часу k при подачі сигналу $x_i = 0 \ (i = 1,2)$ нейрони перебували в стані $z_i(k) = 1 \ (i = 1,2)$, тобто згідно з (4.4) на виході мережі було отримано сигнали $y_i = 1 \quad (i = 1,2)$. Відповідну мережу наведено на рис. 4.3.

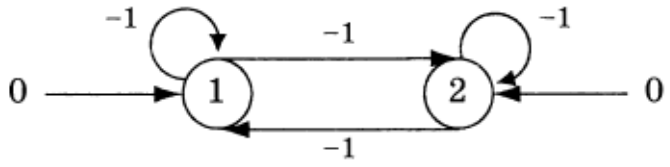


Рис. 4.3. Мережа, для якої порушується умова $w_{ii} = 0$

У наступний, (k+1)-й, момент часу, стани й вихідні сигнали нейронів дорівнюватимуть

$$z_i(k+1) = (-1) \cdot 1 + (-1) \cdot 1 + 0 = -2;$$

$$y_i(k+1) = 0, \quad i = 1,2.$$

Аналогічно для (k+2)-го моменти часу отримаємо

$$z_i(k+2) = (-1)\cdot 0 + (-1)\cdot 0 + 0 = 0;$$

$$y_i(k+1) = 1, \;\; i = 1{,}2.$$

Далі процес повторяється. Таким чином, при порушенні умови $w_{ii} = 0$ мережа осцилює або зациклюється, тобто не досягає стійкого стану.

Приклад 4.2. Розглянемо наслідки порушення умови $w_{ij} = w_{ji}$ на прикладі мережі, що складається із двох нейронів та має і нульовий поріг $(\theta_i = 0)$ і матрицю ваг

$$w = \begin{pmatrix} 0 & -2 \\ -2 & 0 \end{pmatrix}$$

Нехай у k-й момент часу при подачі на вхід нейронів сигналів $x_1 = 1, \; x_2 = -1$ на їхніх виходах були сигнали $y_1 = 1, \;\; y_1 = 0.$ Цю мережу зображено на рис. 4.4.

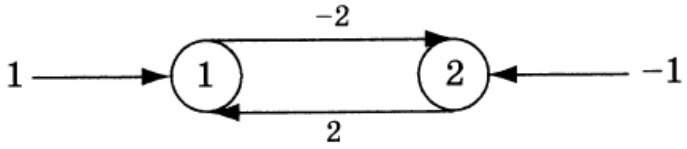


Рис. 4.4. Мережа, для якої порушується умова $w_{ij} = w_{ji}$

Скориставшись формулами (4.3) і (4.4), отримуємо, що в наступні моменти часу стану мережі й значення її вихідних сигналів зміняться у такий спосіб:
$k+1$:

$$z_1(k+1) = -2\cdot 0 + 1 = 1 \qquad y_1(k+1) = 1,$$
$$z_2(k+1) = 2\cdot 1 - 1 = \;\; 1 \qquad y_2(k+1) = 1;$$

$k+2$:

$$z_1(k+2) = -2\cdot 1 + 1 = -1 \qquad y_1(k+2) = 0,$$
$$z_2(k+2) = \;\; 2\cdot 1 - 1 = \;\; 1 \qquad y_2(k+2) = 1;$$

$k+3$:

$$z_1(k+3) = -2\cdot 1 + 1 = -1 \qquad y_1(k+3) = 0,$$
$$z_2(k+3) = \;\; 2\cdot 0 - 1 = -1 \qquad y_2(k+3) = 0;$$

$k+4$:

$$z_1(k+4) = -2\cdot 0 + 1 = 1 \qquad y_1(k+4) = 1,$$
$$z_2(k+4) = 2\cdot 0 - 1 = -1 \qquad y_2(k+4) = 0.$$

Мережа повернулася в початковий стан, і далі процес повторюється. Таким чином, при порушенні умови $w_{ij} = w_{ji}$ мережа також не досягає стійкого стану, тобто осцилює.

Активація мережі Хопфілда може здійснюватися асинхронно, коли на кожному такті змінює свій стан тільки одним випадковим способом обраний нейрон, і синхронно, коли всі нейрони змінюють свій стан одночасно.

4.2. Навчання в мережі Хопфілда

Робота мережі Хопфілда може бути пояснена термінами енергетичного ландшафту [11]. Є ландшафт, що являю собою гористу місцевість, на вершині якої перебуває куля. Потім куля котиться по схилу, поки не зупиниться в якій-небудь низині (западині). Ці низини відбивають стійкі стани мережі, і кожна з них відповідає певному поданому образу (образу навчання). У цій моделі куля, що має велику потенційну енергію, котиться вниз з меншою потенційною енергією, досягаючи локального мінімуму. Щоб знову опинитися в початковому стані, вона повинна здійснити у фізичному значенні роботу, тобто витратити енергію. Таким чином, робота мережі Хопфілда може бути охарактеризована деякою енергетичною функцією.

Якщо в процесі аналізу перцептрона вибір такої функції (функціонала) особливих ускладнень не викликає [11], оскільки відомі реальні й бажані значення виходів мережі, то в цьому випадку ситуація дещо інша. По-перше, структура мережі Хопфілда не дозволяє заздалегідь визначити бажану або необхідну послідовність станів нейронів. По-друге, наявність зворотних зв'язків призводить до того, що виходи мережі після кожного такту функціонування у свою чергу подаються на її входи. Крім того, має бути врахована відсутність у нейронів власних зворотних зв'язків. Зазначені особливості мережі відображаються в енергетичній функції вигляду

$$E = -\frac{1}{2}\sum_{i}\sum_{j\neq i} w_{ij} y_i y_j - \sum_{i} y_i x_i + \sum_{i} y_i \theta_i, \qquad (4.5)$$

де θ_i — значення порога i-го нейрона; w_{ij} – ваги зв'язку i-го й j-го нейронів; x_i — зовнішній вихідний сигнал i-го нейрона (постійна величина на протязі кожного моменту часу або такту); y_i — значення вихідного сигналу i-го нейрона.

Хопфілд довів [11], що при активації мережі функція (4.5) не зростає й досягає локального мінімуму в деякому сталому стані. А оскільки кількість таких стійких станів обмежена, мережа за певних умов досягає одного з них за скінченну кількість ітерацій. При цьому, як уже зазначалося, низини енергетичного ландшафту (енергетичні мінімуми) відповідають збереженим образам. Щоб мережа Хопфілда правильно класифікувала образи, ці низини не мають перекриватися.

4.2.1. Накопичення образів у мережі Хопфілда

Функціонал (4.5) можна розглянути як критерій помилки (чим далі від бажаного образу, тим більше значення він приймає). Щоб зберігати подані на вхід мережі образи, необхідно мінімізувати значення енергетичної функції для кожного з них, тобто визначати локальні мінімуми енергетичного ландшафту. Однак додавання нових образів не має знищувати вже наявну інформацію. А оскільки вся інформація щодо образів, які зберігаються, міститься у ваговій матриці, задача навчання зводиться до визначення значень ваг, що мінімізують енергетичний функціонал.

При мінімізації даного функціонала необхідно враховувати наступне. Як видно з (4.5), мінімум даного виразу перебуває у від'ємній області. Для виконання умови $\sum_i y_i\, \theta_i < 0$ необхідно, щоб компоненти y_i й θ_i постійно мали протилежні знаки, тому доцільно покласти $\theta_i = 0$, що призводить до критерію вигляду

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} y_i y_j - \sum_i x_i y_i . \quad (4.6)$$

Позначимо за аналогією з вищевикладеним x_m, y_m — компоненти m-го образу. Тоді матрицю ваг w_{ij} можна розбити на дві підматриці, перша з яких (w_{ij}^1) відображає зв'язки всіх образів, крім m-го, а друга (w_{im}) — тільки m-ий образ

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij}^1 x_i x_j - \sum_{i \neq m} x_i y_i - \frac{1}{2}\sum_{i \neq m} w_{im} y_i y_m - x_m y_m . \quad (4.7)$$

Перші два доданки (4.7) можна розглядати як збурення, що відповідає загальному впливу на функціонал усіх образів, крім m-го, а інші — як корисний сигнал. Отже, щоб мінімізувати функціонал (4.7) для заданого m-го образу, необхідно розглянути другий доданок, тобто

$$E_m = -\frac{1}{2}\sum_{i \neq m} w_{im} y_i y_m - x_m y_m . \quad (4.8)$$

Мінімізація (4.8) еквівалентна максимізації

$$\sum_{i \neq m} w_{im} y_i y_m + x_m y_m \quad (4.9)$$

Розглянемо перший доданок. Оскільки $y_m \in \{-1,1\}$, то завжди $(y_m)^2 > 0$. Якщо вибрати $w_{im} = y_i y_m$, то

$$\sum_{i \neq m} w_{im} y_i y_m = \sum_{i \neq m} (y_i)^2 (y_m)^2 . \quad (4.10)$$

Отже, вибір $w_{im} = y_i y_m$ забезпечує максимум функціоналу (4.9). Неважко побачити, що при нульових початкових умовах подача на вхід образу x призводить відповідно у (4.3) і (4.4) до появи на виході мережі вектора $y = x$. Цим пояснюється вибір значень вагових коефіцієнтів на стадії ініціалізації у вигляді (4.1).

4.2.2. Виклик образу

Після того, як усі подані образи будуть збережені мережею (побудовано енергетичний ландшафт), інформація про них може бути отримана шляхом мінімізації функціонала помилки на фазі навчання. Використання енергетичного ландшафту дозволяє визначити вплив окремої вершини на енергію, а мінімізація енергії — шлях переходу мережі в будь який стійкий стан.

Функціонування мережі Хопфілда полягає у наступному:

1. Навчання.

Подаються образи x^p $(p = \overline{1,P})$. За формулою (4.1) обчислюються елементи w_{ij} матриці ваг і для активних нейронів (що перебувають у стані «+1») на основі їх поточного й попереднього стану обчислюється енергія кожного образу

$$E = -\frac{1}{2}\sum_{i=1}^{N} x_i^p (k+1) \sum_{j=1}^{N} w_{ij} x_j^p (k).$$

2. Розпізнавання.

Подаються образи, можливо спотворені, й за формулами (4.3), (4.4) обчислюються послідовні стани й значення вихідних сигналів нейронів до досягнення ними стійких станів.

Приклад 4.3. Розглянемо функціонування мережі Хопфілда, що складається з 7 нейронів і біполярної порогової функції активації з порогом $\theta = 0$, для навчання якої використовуються образи

$$x^1 = (1, -1, -1, 1, -1, 1, 1)^T \text{ і } x^2 = (-1, -1, 1, -1, 1, -1, 1)^T$$

і яка має розпізнавати створений образ $x^3 = \tilde{x}^2 = (-1, -1, \underline{-1}, -1, 1, -1, 1)^T$ (спотворений біт підкреслений).

На рис. 4.5 наведена ця мережа й позначені деякі її зв'язки. Елементи w_{ij} матриці ваг W розмірності 7×7, що має вигляд

$$W = W^1 + W^2 = \begin{pmatrix} w_{11} = 0 & w_{12} & w_{13} & \dots & w_{17} \\ w_{21} & w_{22} = 0 & w_{23} & \dots & w_{27} \\ \dots & \dots & \dots & \dots & \dots \\ w_{71} & w_{72} & w_{73} & \dots & w_{77} = 0 \end{pmatrix}$$

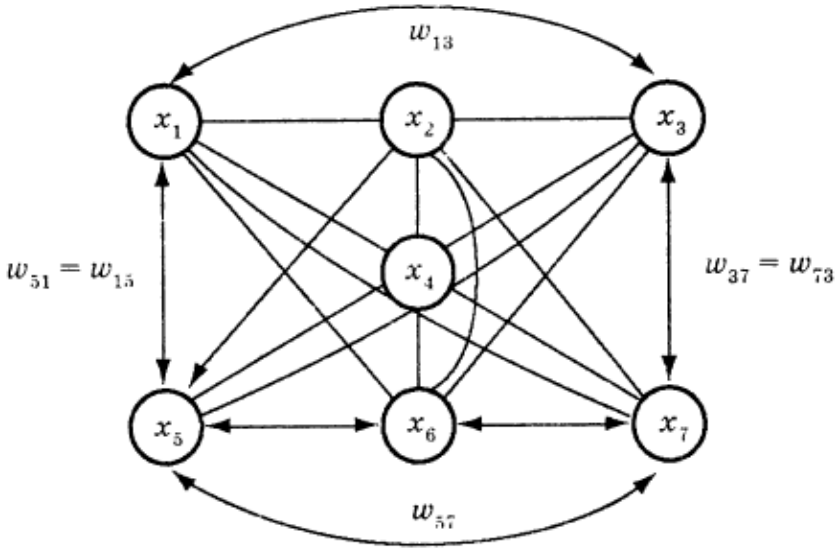


Рис .4.5. Мережа Хопфілда, що складається з 7 нейронів

де W^i — матриця ваг для *i*-го (*i*=1,2) образу, що навчає, визначають за формулою (2.3)

$$w_{12} = 1 \cdot (-1) + (-1) \cdot (-1) = 0; \quad w_{13} = 1 \cdot (-1) + (-1) \cdot 1 = -2;$$
$$w_{14} = 1 \cdot 1 + (-1) \cdot (-1) = 2; \quad w_{15} = 1 \cdot (-1) + (-1) \cdot 1 = -2;$$
$$w_{16} = 1 \cdot 1 + (-1) \cdot (-1) = 2; \quad w_{17} = 1 \cdot 1 + (-1) \cdot 1 = 0 \text{ і т. д.}$$

Після визначення всіх ваг матриця набуває вигляду

$$W = \begin{pmatrix} 0 & 0 & -2 & 2 & -2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 \\ -2 & 0 & 0 & -2 & 2 & -2 & 0 \\ 2 & 0 & -2 & 0 & -2 & 2 & 0 \\ -2 & 0 & 2 & -2 & 0 & -2 & 0 \\ 2 & 0 & -2 & 2 & -2 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Обчислимо енергію кожного образу, беручи до уваги тільки активні нейрони. Для першого образу

$$E^1 = \sum_{i=1}^{7} E_i^1 = E_1^1 + E_4^1 + E_6^1 + E_7^1,$$

$$E_1^1 = -\frac{1}{2}[x_1(w_{13}x_3 + w_{14}x_4 + w_{15}x_5 + w_{16}x_6)] =$$

$$= -\frac{1}{2}[(-2)\cdot(-1) + 2\cdot 1 + (-2)\cdot(-1) + 2\cdot 1] = -4;$$

$$E_4^1 = -\frac{1}{2}[x_4(w_{41}x_1 + w_{43}x_3 + w_{45}x_5 + w_{46}x_6)] =$$

$$= -\frac{1}{2}[2\cdot 1 + (-2)\cdot(-1) + (-2)\cdot(-1) + 2\cdot 1] = -4;$$

$$E_6^1 = -\frac{1}{2}[x_6(w_{61}x_1 + w_{63}x_3 + w_{64}x_4 + w_{65}x_5)] =$$

$$= -\frac{1}{2}[2\cdot 1 + (-2)\cdot(-1) + 2\cdot 1 + (-2)\cdot(-1)] = -4;$$

$$E_7^1 = -\frac{1}{2}[x_7(w_{72}x_2)] = -\frac{1}{2}[(-2)\cdot(-1)] = -1.$$

Тоді $E^1 = -13$.

Аналогічно отримуємо для другого образу

$$E^2 = \sum_{i=7}^{7} E_i^2 = E_3^2 + E_5^2 + E_7^2 = -9.$$

Розпізнавання образу $x^3 = \tilde{x}^2$ починаємо з визначення його енергії, що після проведення аналогічних обчислень дорівнюватиме

$$E^3 = \sum_{i=1}^{7} E_i^3 = E_5^3 + E_7^3 = -5.$$

За формулами (4.3) і (4.4) обчислюємо нові стани нейронів і відповідні значення вихідних сигналів, які з урахуванням біполярної активаційної функції дорівнюватимуть

$$z_j^3 = \sum_{i=1}^{7} w_{ij}x_i^3, \quad j = \overline{1,7}$$

$$z_1^3 = w_{31}x_3^3 + w_{41}x_4^3 + w_{51}x_5^3 + w_{61}x_6^3 + x_1^3 =$$

$$= (-2)\cdot(-1) + 2\cdot(-1) + (-2)\cdot 1 + 2\cdot(-1) - 1 = -5 < 0; \; y_1^3 = -1;$$

$$z_2^3 = w_{72}x_7^3 + x_2^3 = (-2)\cdot 1 - 1 = -3 < 0; \; y_2^3 = -1;$$

$$z_3^3 = w_{13}x_1^3 + w_{43}x_4^3 + w_{53}x_5^3 + w_{63}x_6^3 + x_3^3 =$$

$$= (-2)\cdot(-1) + (-2)\cdot(-1) + 2\cdot 1 + (-2)\cdot(-1) - 1 = 7 > 0; \; y_3^3 = -1;$$

$$z_4^3 = w_{14}x_1^3 + w_{34}x_3^3 + w_{54}x_5^3 + w_{64}x_6^3 + x_4^3 =$$

$$= 2 \cdot (-1) + (-2) \cdot (-1) + (-2) \cdot 1 + 2 \cdot (-1) - 1 = -5 < 0; \;\; y_4^3 = 1;$$
$$z_5^3 = w_{15}x_1^3 + w_{35}x_3^3 + w_{45}x_4^3 + w_{65}x_6^3 + x_5^3 =$$
$$= (-2) \cdot (-1) + 2 \cdot (-1) + (-2) \cdot (-1) + (-2) \cdot (-1) + 1 = 5 > 0; \;\; y_5^3 = -1;$$
$$z_6^3 = w_{16}x_1^3 + w_{36}x_3^3 + w_{46}x_4^3 + w_{56}x_5^3 + x_6^3 =$$
$$= 2 \cdot (-1) + (-2) \cdot (-1) + 2 \cdot (-1) + (-2) \cdot 1 - 1 = -5 < 0; \;\; y_6^3 = -1;$$
$$z_7^3 = w_{27}x_2^3 + x_7^3 = (-2) \cdot (-1) + 1 = 3 > 0; \;\; y_7^3 = 1.$$

Таким чином, після подання $x^3 = \widetilde{x^2}$ мережа перейде в стійкий стан

$$(-1, -1, 1, -1, 1, -1, 1)^T,$$

що відповідає образу x^2 з енергією, як неважко переконатися, $E = -9$. Отже, мережа правильно розпізнала (відновила створений образ).

5. НЕЙРОМЕРЕЖА КОХОНЕНА

У багатьох моделях ШНМ вирішальну роль відіграють зв'язки між нейронами, які визначаються ваговими коефіцієнтами й зазначають місце нейрона в мережі. Однак у біологічних системах, наприклад, у мозку, сусідні нейрони, отримуючи аналогічні вхідні сигнали, реагують на них подібним чином, тобто групуються, утворюючи деякі області. Оскільки під час обробки багатовимірного вхідного образу здійснюється його проектування на область меншої розмірності зі збереженням його топології, нерідко подібні мережі називають *мапами (self-organizing feature map).* У таких мережах істотним є врахування взаємного розташування нейронів одного шару.

Мережа Кохонена відноситься до мереж, що самоорганізуються, які під час надходження вхідних сигналів, на відміну від мереж, що використовують навчання із учителем, не отримують інформацію про бажаний вихідний сигнал. У зв'язку з цим неможливо сформувати критерій настроювання, заснований на неузгодженості реальних і необхідних вихідних сигналів ШНМ, тому вагові параметри мережі корегують, виходячи з інших міркувань. Усі подані вхідні сигнали із заданої навчальної множини мережа у процесі навчання розділяє на класи, будуючи так звані топологічні мапи.

5.1. Структура мережі Кохонена

Мережа Кохонена використовує таку модель (рис. 5.1): мережа складається з *M* нейронів, що утворюють прямокутні гратки на площині — шар. До нейронів, розташованих в одному шарі, що є двовимірною площиною, підходять нервові волокна, по яких надходить *N*-вимірний вхідний сигнал. Кожен нейрон характеризується своїм розміщенням у шарі й ваговим коефіцієнтом. Розміщення нейронів, у свою чергу, характеризується деякою метрикою й визначається топологією шару, при якій сусідні нейрони під час навчання впливають один на одного сильніше, ніж розташовані далі. Кожен нейрон утворює зважену суму вхідних сигналів. Наявність зв'язків між нейронами призводить до того, що при збудженні одного з них можна обчислити збудження інших нейронів у шарі, причому це збудження зі збільшенням відстані від збудженого нейрона зменшується. Тому центр реакції шару, що виникає у відповідь на отримане подразнення, відповідає місцезнаходженню збудженого нейрона. Зміна вхідного сигналу, що навчає, призводить до максимального збудження іншого нейрона й відповідно — до іншої реакції шару.

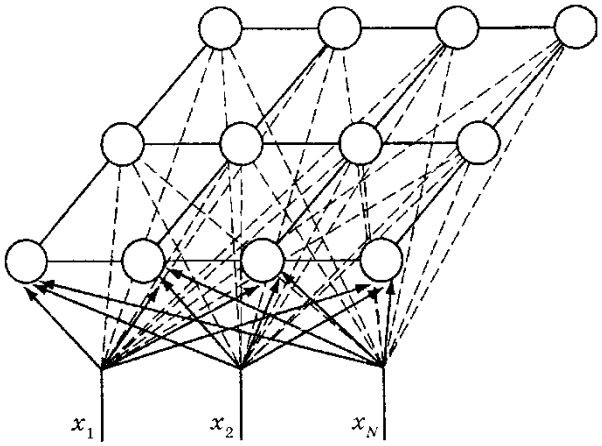


Рис. 5.1. Модель мережі Кохонена

Мережа Кохонена може розглядатися як подальший розвиток мережі векторного квантування. Відмінність їх полягає у способах навчання.

5.2. Навчання мережі Кохонена

Замість того, щоб шукати місцезнаходження нейрона шляхом розв'язання загальних рівнянь збудження, Кохонен істотно спростив розв'язання задачі, виділяючи зі всіх нейронів шару лише один *с-й* нейрон, для якого зважена сума вхідних сигналів максимальна

$$c = \arg\min_j \left(\mathbf{x}^T \mathbf{w}_j \right) \tag{5.1}$$

Зазначимо, що корисною операцією попередньої обробки вхідних векторів є їхня нормалізація $\overline{x}_i = \dfrac{x_i}{\|\mathbf{x}\|}$, $i = 1,...,N$, яка перетворює вектори вхідних сигналів в одиничні з тим же напрямком.

Отже, буде активований тільки той нейрон, вектор ваг якого ш найближчий до вхідного вектора *x.* А оскільки перед початком навчання невідомо, який саме нейрон активуватиметься при поданні мережі конкретного вхідного вектора, мережа навчається без учителя, тобто *самонавчається.*

Вводячи потенційну функцію — функцію відстані f_{ij} («сусідства») між *i-м* й *j*-м нейронами з місцезнаходження r_i й r_j відповідно, що монотонно спадає зі збільшенням відстані між цими нейронами, Кохонен запропонував такий алгоритм корекції ваг [11]:

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \alpha(k)\, f_{cj}\left(\mathbf{x}(k) - \mathbf{w}_j(k)\right), \tag{5.2}$$

де $\alpha(k) \in (0,1]$ — коефіцієнт підсилення, що змінюється в часі (звичайно вибирають $\alpha = 1$ на першій ітерації, поступово зменшуючи в процесі навчання до нуля); $f_{ij}(k)$ — монотонно спадна функція

$$f_{ij}(k) = f\left(\|r_i - r_j\|, k\right) = f(d,k) = f(d,\sigma),$$

де r_i та r_j — вектори, що визначають положення нейронів *i* й *j* у ґратках. При прийнятій метриці $d = \|r_i - r_j\|$ функція $f_{ij}(k)$ із зростанням часу *k* прямує до нуля. На практиці

замість параметра часу k використовують параметр відстані σ , що задає величину області «сусідства» і зменшується із часом до нуля.

Вибір функції $f_{i_i}(k)$ також впливає на величини ваг усіх нейронів у шарі. Очевидно, що для нейрона-переможця c $f_{cc}\left(\left\|r_c - r_c\right\|\right) = f_c(0) = 1$.

Зважаючи на те, що визначення нейронів-переможців у мережі Кохонена й у LVQ відбувається ідентично, то й корекція ваг цих нейронів здійснюється однаково.

На рис. 5.2 наведено приклад зміни двовимірних ваг мапи $w_j = \left(w_{j1}, w_{j2}\right)^T$, що утворює ланцюг. З появою вхідного образу **x** найбільш сильно змінюється ваговий вектор нейрона-переможця 5, менш сильно — ваги розташованих поруч із ним нейронів 3, 4, 6, 7. А оскільки нейрони 1,2,8,9 перебувають поза областю «сусідства», їхні вагові коефіцієнти не змінюються.

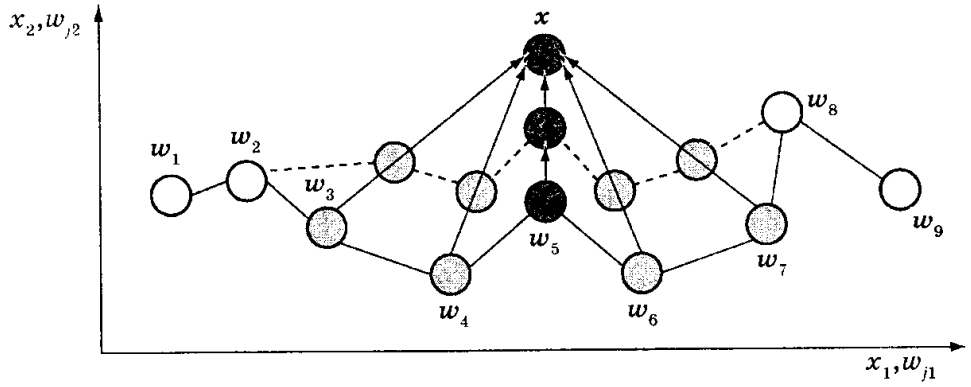


Рис. 5.2. Зміна ваг мапи Кохонена

Отже, алгоритм навчання мережі Кохонена може бути описаний так:

1. *Ініціалізація.*

Ваговим коефіцієнтам усіх нейронів привласнюються малі випадкові значення й здійснюється їхня нормалізація. Вибирається відповідна потенційна функція $f_{ij}(d)$ і призначається початкове значення коефіцієнта підсилення α_0.

2. *Вибір сигналу, що навчає.*

Із усієї множини векторів навчальних вхідних сигналів відповідно до функції розподілу $P(\mathbf{x})$ вибирається один вектор **x**, що представляє «сенсорний сигнал», поданий мережі.

3. *Аналіз відгуку (вибір нейрона)*

За формулою (1.4) визначається активований нейрон.

4. *Процес навчання.*

Відповідно до алгоритму (1.5) змінюються вагові коефіцієнти активованого й сусідніх з ним нейронів доти, поки не буде отримано необхідного значення критерію якості навчання або не буде подане задане число вхідних векторів, що навчають. Остаточне значення вагових коефіцієнтів збігається з нормалізованими векторами входів.

Оскільки мережа Кохонена здійснює проектування N-вимірного простору образів на M-вимірну мережу, аналіз збіжності алгоритму навчання є досить складною задачею. Однак властивості алгоритму можуть бути продемонстровані на простих прикладах.

Приклад 5.1. Розглянемо одновимірний випадок. Припустимо, що мережа складається з одного нейрона, а вхідний сигнал $x \in [a, b]$. Нехай початкове значення ваги дорівнює x_1 (рис. 5.3)

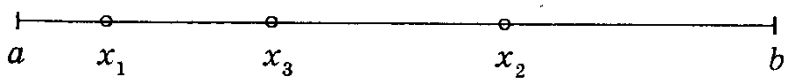


Рис. 5.3. Навчання мережі, що складається з одного нейрона

Кожна зміна x залежить тільки віддаленості від a й b, тобто

$$\frac{dx}{dt} = \frac{\alpha}{2}[(b-x)+(a-x)] = \alpha\left(\frac{a+b}{2} - x\right),$$

де $\alpha \in (0,1]$.

Тоді переміщення x_1 у x_2 відбуватиметься за правилом

$$x_2 = x_1 + \alpha\left(\frac{a+b}{2} - x_1\right) = \frac{a+b}{2} + (1-\alpha)\left(x_1 - \frac{a+b}{2}\right).$$

Аналогічно

$$x_3 = x_2 + \alpha\left(\frac{a+b}{2} - x_2\right) = \frac{a+b}{2} + (1-\alpha)^2\left(x_1 - \frac{a+b}{2}\right).$$

Після n-ї ітерації отримуємо

$$x_n = \frac{a+b}{2} + (1-\alpha)^{n-1}\left(x_1 - \frac{a+b}{2}\right),$$

звідки випливає, що

$$\lim_{n\to\infty} x_n = \frac{a+b}{2},$$

тобто навчання завершується, коли значення ваги ділитиме інтервал $[a,b]$ навпіл. Отже, x_n займе стійке розташування в середині інтервалу $[a,b]$.

Приклад 5.2. Розглянемо одновимірний шар, що складається з нейронів, які ділять заданий інтервал $[a,b]$ на N рівних частин. Нехай ваги нейронів упорядковані, тобто $a < x_1 < x_2 < \cdots < x = b$ (рис. 5.4).

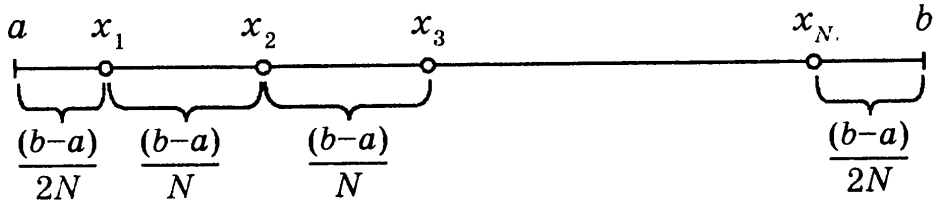


Рис. 5.4. Навчання мережі

Тоді кожна вага мережі може бути обчислена за формулою

$$x_i = a + (2i-1)\frac{b-a}{2N}, \qquad i = \overline{1,N}.$$

Оскільки ці ваги ділять інтервал $[a, b]$ на N рівних частин, то вони займають на цьому інтервалі стійкі розміщення.

Приклад 5.3. Розглянемо двовимірний шар, що складається *з* $N \times N$ нейронів. Вхідний вектор складатиметься із двох компонентів $x = (x_1 x_2)^T$, $x_1 \in [a, b]$, $x_2 \in [c, d]$. Відповідно вектор вагових коефіцієнтів ij-нейрона має вигляд $\omega^{ij} = \left(\omega_1^{ij}, \omega_2^{ij}\right)^T$. Припустимо, що всі компоненти вагового вектора впорядковані $\omega_1^{ij} < \omega_1^{ik}$, $j < k$, $\omega_2^{ij} < \omega_2^{mj}$, $i < m$. Визначаючи для кожного стовпця нашої матриці ваг середнє значення

$$\omega_1^j = \frac{1}{N}\sum_{i=1}^{N} \omega_1^{ij}, \quad j = \overline{1, N},$$

з урахуванням упорядкованості компонентів отримуємо

$$a < \omega_1^1 < \omega_1^2 < \cdots < \omega_1^N < b.$$

Аналогічно отримуємо для кожного рядка матриці ваг

$$c < \omega_2^1 < \omega_2^2 < \cdots < \omega_2^N < d.$$

Слід зазначити, що в процесі навчання вагові коефіцієнти розіб'ють інтервали $[a, b], [c, d]$ на рівні частини, тобто нейрони розташуються у вузлах ґраток.

Якби з кожним нейроном шару асоціювався один вхідний вектор, то вектор ваг будь-якого нейрона шару Кохонена міг би бути навчений за допомогою одного обчислення, оскільки вага нейрона-переможця корегувалися б з $\alpha = 1$ (для одновимірного випадку вага відразу б потрапляла в центр відрізка $[a, b]$). Однак звичайно навчальна множина містить багато подібних між собою вхідних векторів, і мережа Кохонена має бути навчена активувати той самий нейрон для кожного з них. Це досягається усередненням вхідних векторів шляхом зменшення величини α при поданні кожного наступного вхідного сигналу. Отже, ваги, асоційовані з нейроном, усередняться й приймуть значення поблизу «центра» вхідних сигналів, для яких даний нейрон є «переможцем».

Приклад 5.4. Нехай мережа, що складається із трьох нейронів, має класифікувати такі пропоновані їй нормалізовані вектори:

$$x(1) = (0{,}8;\ 0{,}6)^T; \qquad x(2) = (0{,}7071;\ 0{,}7071)^T;$$
$$x(3) = (0{,}6;\ -0{,}8)^T; \qquad x(4) = (0{,}1961;\ -0{,}9806)^T;$$
$$x(5) = (-0{,}9806;\ -0{,}1961)^T; \qquad x(6) = (-0{,}9806; 0{,}1961)^T.$$

Випадковим способом обрані нормалізовані початкові значення ваг дорівнюють
$w_1(0) = (1{,}000; 0{,}000)^T$; $w_2(0) = (0; -1)^T$; $w_3(0) = (-0{,}707; 0{,}707)^T$.
Розміщення векторів x й $w_i(0)$ зображено на рис. 5.5

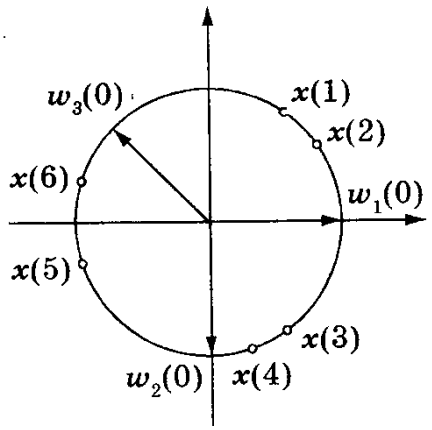


Рис. 5.5. Розміщення векторів x і $\omega(0)$

Подання мережі вектора $x(0)$ дає

$$w_1^T(0)x(1) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ -0,7 & 0,7 \end{bmatrix}^T \begin{bmatrix} 0,8 \\ 0,6 \end{bmatrix} = \begin{bmatrix} 0,8 \\ -0,6 \\ -0,14 \end{bmatrix}.$$

Оскільки $max[w_1^T(0)x(1)] = 0,8$, переможцем буде перший нейрон. Скорегуємо його ваги відповідно до алгоритму (1.5), прийнявши $\alpha = 0,5$,

$$w_1(1) = w_1(0) + 0,5\big(x(1) - w_1(0)\big) = \begin{bmatrix} 1,0 \\ 0 \end{bmatrix} + 0,5\left(\begin{bmatrix} 0,8 \\ 0,6 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0,9 \\ 0,3 \end{bmatrix}.$$

Ваги змінилися таким чином, що вектор перемістився до вхідного вектора $x(1)$ *(див.* рис. 5.7). Оскільки порядок подання вхідних векторів довільний, то при поданні $x(4)$ отримуємо

$$w_1^T(1)x(4) = -0,11769;\ w_2^T(0)x(4) = 0,9806;\ w_3^T(0)x(4) = -0,8319.$$

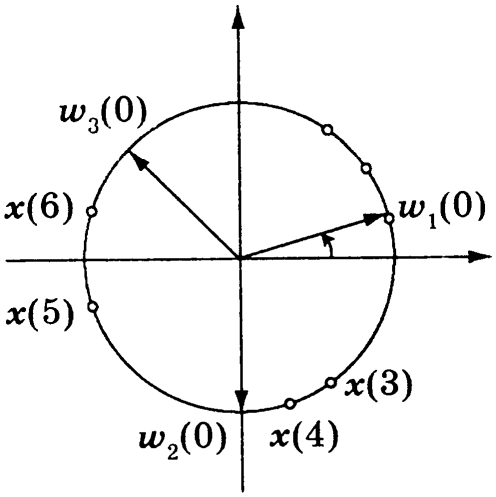


Рис. 5.6. Корекція вагового вектора w_1

Оскільки переможцем виявився другий нейрон, скорегуємо його вагові коефіцієнти, міняючи його розміщення (рис. 1.14) переміщенням до $x(4)$

$$w_2(1) = w_2(0) + 0,5\big(x(4) - w_2(0)\big) =$$

$$= \begin{bmatrix} 0 \\ -1 \end{bmatrix} + 0{,}5\left(\begin{bmatrix} 0{,}1961 \\ -0{,}9806 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0{,}1 \\ -0{,}99 \end{bmatrix}.$$

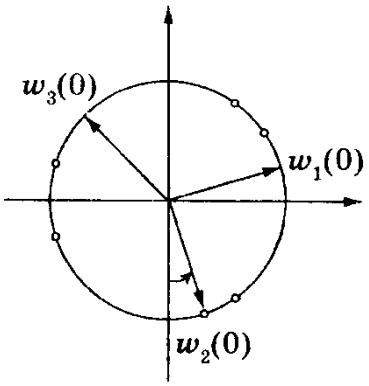


Рис. 5.7. Корекція вагового вектора w_2

Процес навчання завершиться, коли кожен ваговий вектор стане прототипом різних поданих мережі вхідних сигналів, тобто відповідатиме своїй групі вхідних сигналів. В остаточному підсумку вектори ваг займуть положення приблизно такі ж, які наведено на рис. 5.5.

5.3. Вибір функції «сусідства»

На рис. 5.9 зображено шари нейронів з нейроном-переможцем, позначеним чорним кільцем. Оскільки ваги всіх затемнених нейронів змінюються по-різному, залежно від їхньої далекості від нейрона-переможця, найбільш простим є вибір як f_{ij} деякої величини, що дорівнює одиниці при $j = i$, меншої одиниці для затемнених нейронів, тобто нейронів, що лежать у безпосередній близькості від активованого нейрона, і нуль для інших, позначених світлими кружками.

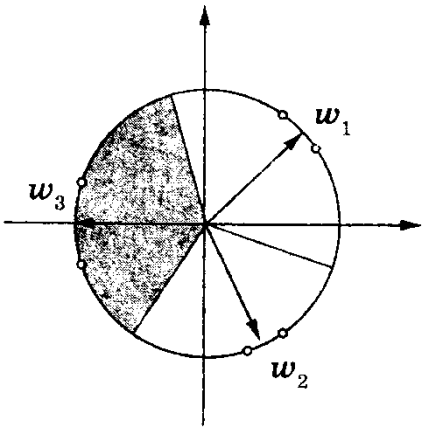


Рис. 5.8. Остаточне розміщення вагових векторів

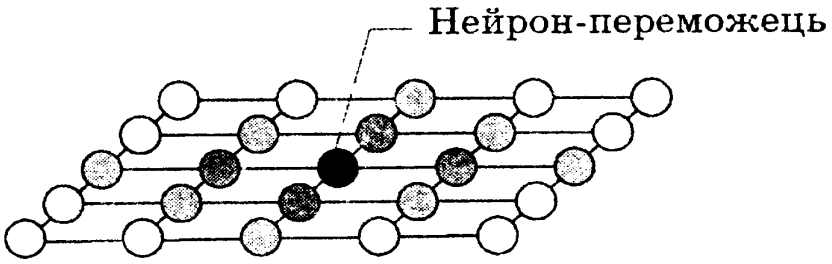


Рис. 5.9. Шар нейронів Кохонена

На практиці ж як f_{ij} вибирають функції, що використовують евклідову метрику

$$d = \sum_k (r_{ik} - r_{jk})^2,$$

де r_{ik}, r_{jk} – координати i-го й j-го нейронів.

До найбільш поширених потенційних функцій відносяться:

а) функція Гаусса

$$f_{\text{gauss}}(d, \sigma) = e^{-\frac{a^2}{2\sigma^2}},$$

де σ^2— дисперсія відхилення;

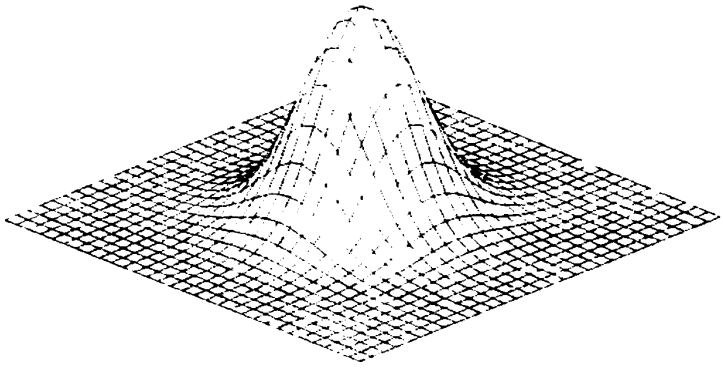


Рис. 5.10. Функція Гаусса

б) функція «Мексиканський капелюх»

$$f_{gauss2}(d, \sigma) = \left(1 - \left(\frac{d}{\sigma}\right)^2\right) e^{-\left(\frac{d}{\sigma}\right)^2},$$

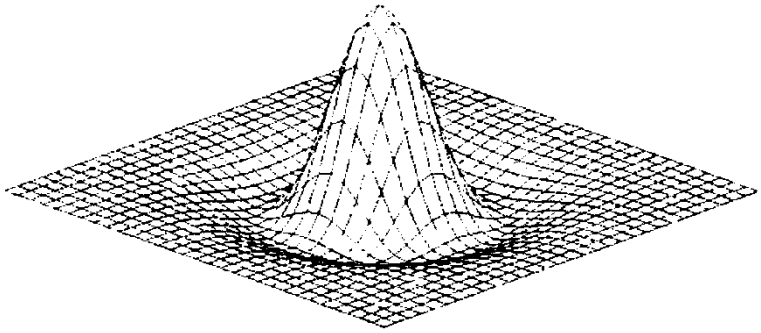


Рис. 5.11. Функція «Мексиканський капелюх»

в) косинусоїдна функція

$$f_{cos}(d,\sigma) = \begin{cases} cos\left(\frac{d\pi}{2\sigma}\right), d < \sigma; \\ 0, \qquad\qquad d \geq \sigma; \end{cases}$$

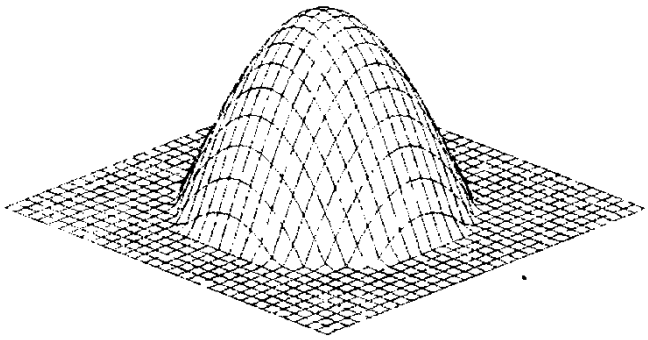


Рис. 5.12. Косинусоїдна функція

г) конусоподібна функція

$$f_{cone}(d,\sigma) = \begin{cases} 1 - \frac{d}{\sigma}, \qquad d < \sigma; \\ 0, \qquad\qquad d \geq \sigma; \end{cases}$$

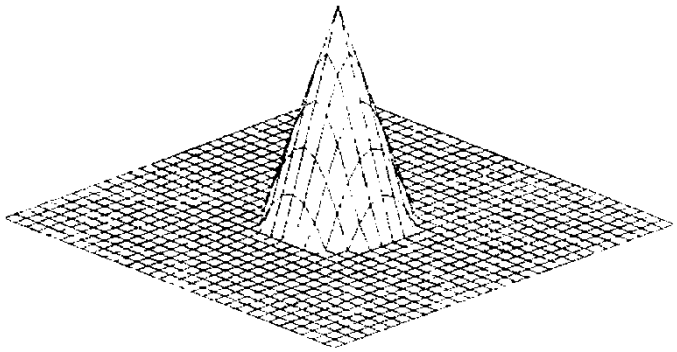


Рис. 5.13. Конусоподібна функція

д) циліндрична функція

$$f_{cylinder}(d,\sigma)=\begin{cases}1, & \text{при } d<\sigma;\\ 0, & \text{при } d\geq\sigma;\end{cases}$$

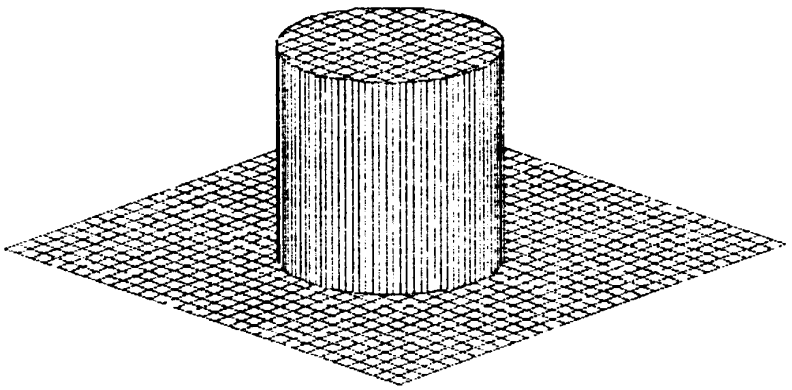


Рис. 5.14. Циліндрична функція

5.4. Побудова мапи Кохонена

Як правило, спочатку будують досить грубу мапу (модель розбиття), поступово уточнюючи її в процесі навчання. Для цього необхідно повільно змінювати не тільки параметр α, але й, наприклад, параметр σ у формулі (5.2). Одним з ефективних способів зміни цих параметрів є такий:

$$\alpha(k)=\alpha(0)\left[\frac{\alpha_{min}}{\alpha(0)}\right]^{\frac{k}{k_{max}}},$$

$$\sigma(k)=\sigma(0)\left[\frac{\sigma_{min}}{\sigma(0)}\right]^{\frac{k}{k_{max}}},$$

де $\alpha(0)\approx 0{,}8;\ \alpha_{min}\ll 1;\ \sigma(0)=0{,}2;\ \sigma_{min}=0{,}5$ — параметри, що визначають крутизну функції $f_{ij};\ k_{max}$ — кількість ітерацій, що задаються.

На рис. 5.15 зображено процес побудови мапи Кохонена, що являє собою двовимірну ґратку, утворену шляхом з'єднання сусідніх нейронів, які перебувають у вузлах ґраток. Виходячи з початкових умов і використовуючи алгоритм навчання , мережа в міру збільшення кількості навчальних вхідних образів розвивається й набуває вигляду ґраток. Внизу кожного рисунка зображено кількість образів, на основі яких отримана відповідна мережа [1].

При реалізації мережі Кохонена виникають такі проблеми:

1. Вибір коефіцієнта навчання α.

Цей вибір впливає як на швидкість навчання, так і на стійкість отримуваного розв'язку. Очевидно, що процес навчання прискорюється (швидкість збіжності алгоритму навчання збільшується) при виборі α близьким до одиниці. Однак у цьому випадку подання мережі різних вхідних векторів, що відносяться до одного класу, призведе до змін відповідного вектора вагових коефіцієнтів. Навпаки, при $\alpha\to 0$ швидкість навчання буде повільною, однак вектор вагових коефіцієнтів за умови досягнення

центра класу при подачі на вхід мережі різних сигналів, що відносяться до одного класу, залишатиметься поблизу цього центра.

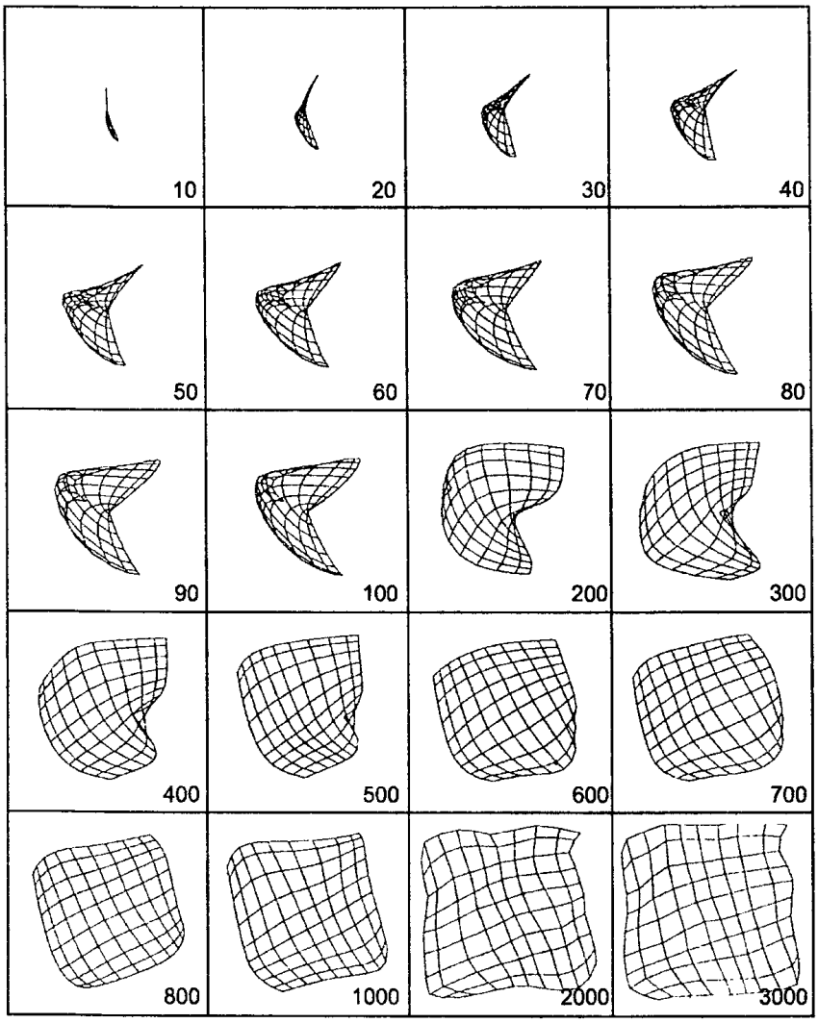


Рис. 5.15. Побудова мапи Кохонена

Тому одним зі шляхів прискорення процесу навчання при одночасному забезпеченні отримання стійкого розв'язку є вибір змінного α з $\alpha \to 1$ на початкових етапах навчання й $\alpha \to 0$ — на завершальних. На жаль, такий підхід не може бути застосований у тих випадках, коли мережа має безупинно підлаштовувати-ся до поданих їй нових вхідних сигналів.

2. Рандомізація ваг.

Рандомізація ваг шару Кохонена може породити серйозні проблеми під час навчання, оскільки в результаті цієї операції вагові вектори розподіляються рівномірно по поверхні гіперсфери. Як правило, вхідні вектори розподілені нерівномірно й групуються на відносно малій частині поверхні гіперсфери. Тому більшість вагових векторів виявляться настільки віддаленими від будь-якого вхідного вектора, що не будуть активовані й стануть марними. Більш того, активованих нейронів, які залишилися, може виявитися занадто мало, щоб розбити близько розташовані вхідні вектори на кластери.

3. Вибір початкових значень векторів вагових коефіцієнтів і нейронів.

Якщо початкові значення обрані невдало, тобто розташованими далеко від поданих вхідних векторів, то нейрон не виявиться переможцем ні при яких вхідних сиг-

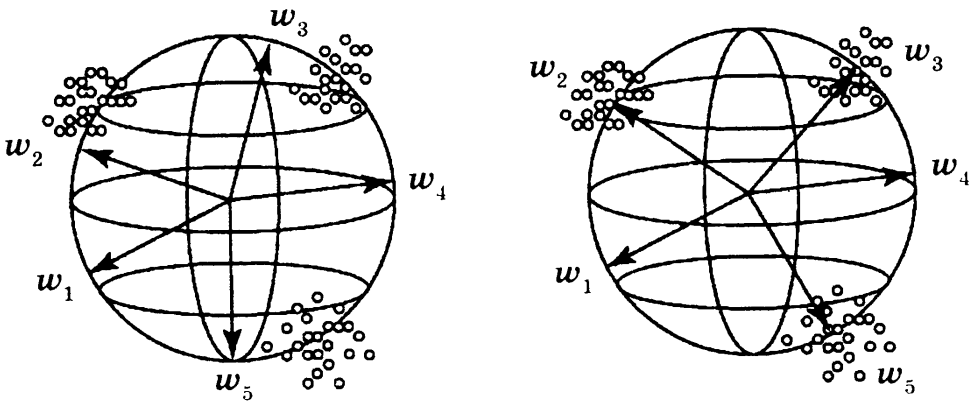
налах, а, отже, не навчиться. На рис. 5.16 наведено випадок, коли початкове розташування ваг w_1 й w_4 призвело до того, що після навчання мережі всі пропоновані образи класифікують нейрони з вагами w_2, w_3 і w_5. Ваги ж w_1 й w_4 не змінилися й можуть бути вилучені з мережі.

4. Кількість нейронів у шарі.

Кількість нейронів у шарі Кохонена має відповідати кількості класів вхідних сигналів. Це може бути неприпустимо в задачах, коли кількість класів заздалегідь невідома.

5. Класи вхідних сигналів.

Шар Кохонена може формувати тільки класи, що являють собою опуклі області вхідного простору.



а б

Рис. 5.16.

ЛІТЕРАТУРА

1. Глибовець М.М., Олецький О.В. Системи штучного інтелекту. — К.: КМ Академія, 2002. — 366 с.
2. Рассел С., Норвиг П. Искусственный интеллект. Современный поход. — М.: Вильямс, 2006. — 1408 с.
3. Люгер Дж. Искусственный интеллект. Стратегии и методы решения сложных проблем. — М.: Вильямс, 2003. — 864 с.
4. Смолин Д.В. Введение в искусственный интеллект: конспект лекций. — М.: ФИЗМАТЛИТ, 2004. — 208 с.
5. Братко И. Алгоритмы искусственного интеллекта на языке Пролог. — М.: Вильямс, 2004. — 640 с.
6. Девятков В.В. Системы искусственного интеллекта. — М.: Изд-во МГТУ им. Баумана, 2001. — 352 с.
7. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя: ЗНТУ, 2008. — 341 с.
8. Представление и использование знаний / Под ред. Уэно Х., Исидзука М. — М.: Мир, 1989. — 220 с.
9. Искусственный интеллект: Справочник: В 3-х т. — М.: Радио и связь, 1990.
10. Нильсон Н. Принципы искусственного интеллекта. — М.: Радио и связь, 1985. — 376 с.
11. Руденко О. Г., Бодянський Є. В. Штучні нейронні мережі: Навчальний посібник. — Харків: ТОВ "Компанія СМІТ", 2006. — 404 с.
12. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. — СПб.: Питер, 2018. — 480 с.
13. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. — М.: Вильямс, 2018. — 688 с.
14. Коцовський В. М. Спектральний аналіз дискретних нейрофункцій. Частина І: Методичний посібник / В. М. Коцовський. — Ужгород: Видавництво УжНУ "Говерла", 2017. — 44 с.