

ЛЕКЦІЯ 4

Тема: Основи штучних нейронних мереж

Питання лекції

1. Основні поняття штучних нейронних мереж
2. Деякі типи нейронних мереж

ВСТУП

В останні роки разом з нечіткими системами великий інтерес викликає проблематика нейронних мереж і генетичних алгоритмів. Ці напрямки відносяться до наукової області, обумовленої в англійській літературі терміном Computational Intelligence. На рис. 1. видно, що задачі нейронних мереж, генетичних алгоритмів і нечітких систем можуть розглядатися поза зв'язком між собою, однак їхня взаємозалежність виявляється надзвичайно важливою. Зокрема, генетичні алгоритми можна застосовувати для підбора ваг і топології нейронної мережі, а також для формування бази правил і функцій приналежності нечіткої системи

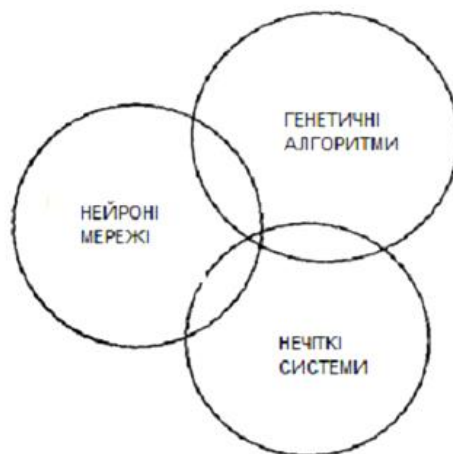
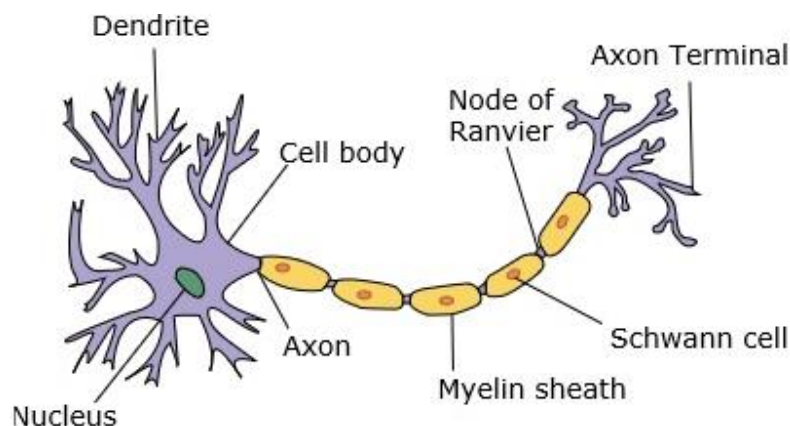
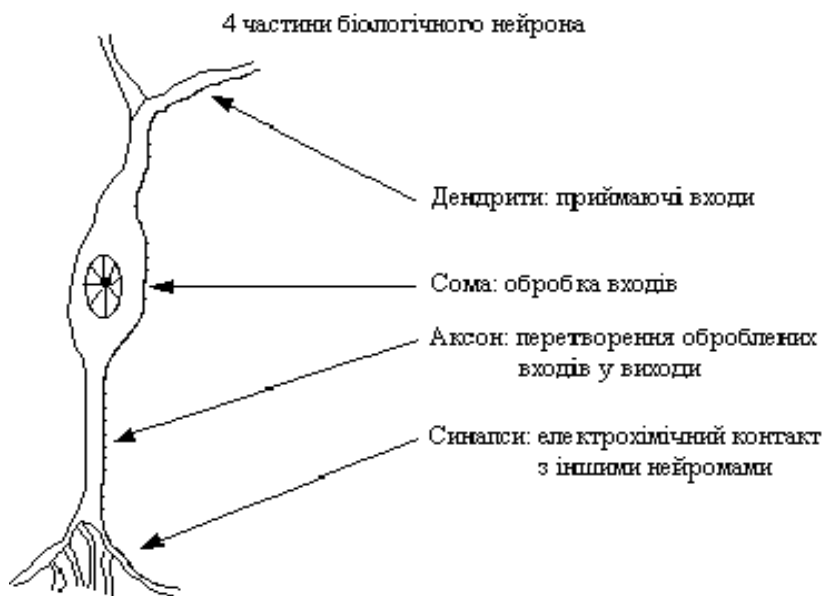


Рис. 1. Взаємозв'язки між нейронними мережами, генетичними алгоритмами і нечіткими системами

У свою чергу, нейронні мережі дозволяють вибирати відповідні параметри для самих генетичних алгоритмів (параметри схрещування і мутації); саму філософію нейронних мереж можна закласти у фундамент нечітких систем, що у результаті знаходять здатність до навчання.

Крім того, методи теорії нечітких множин дозволяють підбирати як згадані вище параметри генетичних алгоритмів, так і коефіцієнти, що визначають швидкість навчання нейронних мереж. Одним з популярних напрямків Artificial Intelligence є **теорія нейронних мереж (neuron nets)**.

Людей завжди цікавило їхнє власне мислення. Це самопитання, думання мозку самого про себе є, можливо, відмітною рисою людини. Нейробіологи і нейроанатоми досягли в цій області значного прогресу. Ретельно вивчаючи структуру і функції нервової системи людини, вони багато чого зрозуміли в «електропроводці» мозку, але мало довідалися про його функціонування. У процесі нагромадження ними знань з'ясувалося, що мозок має приголомшуючу складність. Сотні мільярдів нейронів, кожен з яких з'єднаний із сотнями або тисячами інших, утворюють систему, що далеко перевершує наші самі сміливі мрії про суперкомп'ютери.



На сьогоднішній день існують **дві** взаємно збагачуючі одна одну **мети нейронного моделювання**: *перша* – зрозуміти функціонування нервової системи людини на рівні фізіології і психології і *друга* – створити обчислювальні системи (штучні нейронні мережі), що виконують функції, подібні до функцій мозку.

Штучні нейронні мережі є моделями нейронної структури мозку, який здатен сприймати, обробляти, зберігати та продукувати інформацію. Особливістю мозку також є навчання та самонавчання на власному досвіді. Адаптивні системи на основі штучних нейронних мереж дозволяють з успіхом вирішувати проблеми розпізнавання образів, виконання прогнозів, оптимізації, асоціативної пам'яті і керування.

Механізм природного мислення базується на збереженні інформації у вигляді образів. Штучні нейронні мережі дозволяють створення паралельних мереж, їх навчання та вирішення інтелектуальних завдань, не використовуючи традиційного програмування. В лексиконі розробників та користувачів нейромереж присутні слова "поводити себе", "реагувати", "самоорганізовувати", "навчати", "узагальнювати" та "забувати".

1 Основні поняття штучних нейронних мереж

Штучний нейрон є базовим модулем нейронних мереж. Він моделює основні функції природного нейрона (рис. 2).

При функціонуванні нейрон одночасно отримує багато вхідних сигналів. Кожен вхід має свою власну синаптичну вагу, яка надає входу вплив, необхідний для функції суматора елемента обробки. Ваги є мірою сили вхідних зв'язків і моделюють

різноманітні синаптичні сили біологічних нейронів. Ваги суттєвого входу підсилюються і, навпаки, вага несуттєвого входу примусово зменшується, що визначає інтенсивність вхідного сигналу. Ваги можуть змінюватись відповідно до навчальних прикладів, топології мережі та навчальних правил.

Вхідні сигнали x_n зважені ваговими коефіцієнтами з'єднання w_n додаються, проходять через передатну функцію, генерують результат і виводяться.

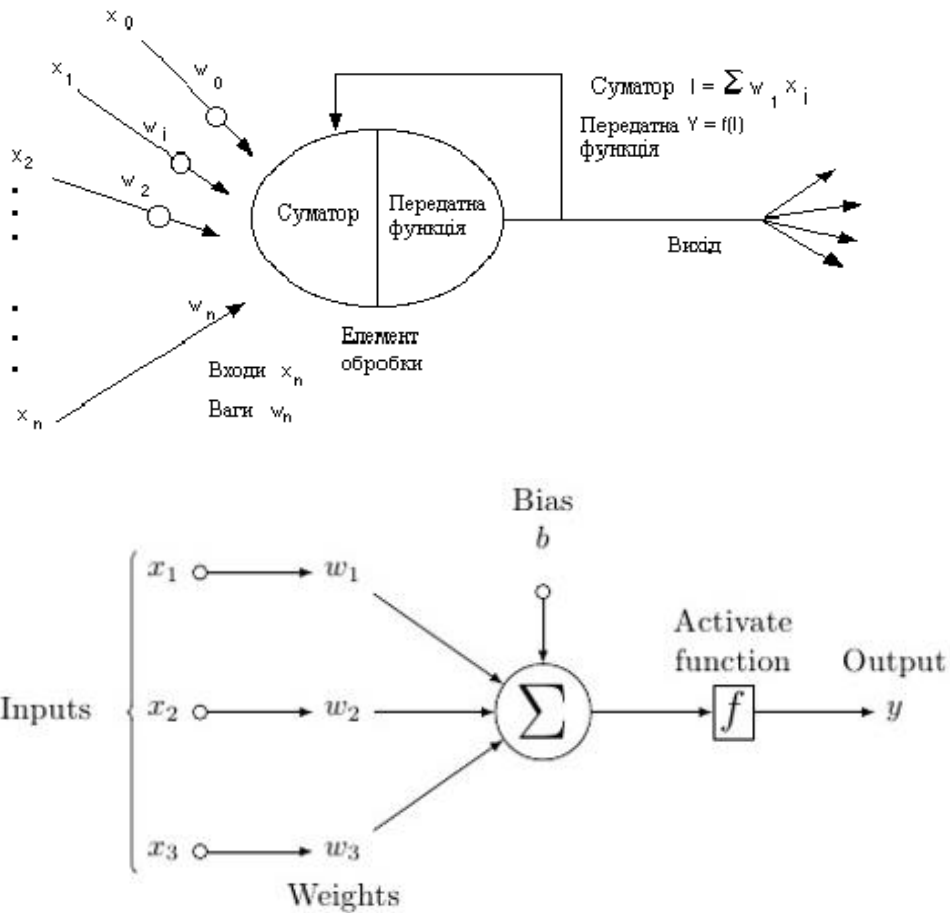


Рис. 2. Базовий штучний нейрон

В програмних реалізаціях штучні нейрони називають «елементами обробки» або «процесорами» і вкладають в них більше можливостей, ніж в базовому штучному нейроні, що описаний вище.

На рис. 3 зображена детальна схема штучного нейрону.



Рис. 3. Модель "елементу обробки"

Функція суматора може бути складнішою, наприклад, вибір мінімуму, максимуму, середнього арифметичного, добутку або обчислюватися за іншим алгоритмом. Багато

програмних реалізацій використовують власні функції суматора, що запрограмовані на мові вищого рівня (C, C++).

Перед надходженням до передатної функції вхідні сигнали та вагові коефіцієнти можуть комбінуватись багатьма способами. Алгоритми для комбінування входів нейронів визначають відповідно до мережної архітектури та парадигми.

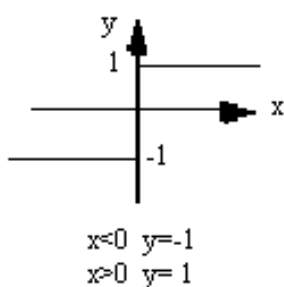
В деяких нейромережах суматор виконує додаткову обробку, так звану функцію активації, яка зміщує вихід функції суматора в часі. Цю функцію найкраще використовувати як компоненту мережі в цілому, ніж як компоненту окремого нейрона. Часто, ця функція є відсутньою.

Результат функції суматора перетворюється у вихідний сигнал через передатну функцію. В передатній функції для визначення виходу нейрона загальна сума порівнюється з деяким порогом (зазвичай, це діапазон $[0, 1]$ або $[-1, 1]$ або інше) за допомогою певного алгоритму.

Переважаючо застосовують нелінійну передатну функцію, оскільки лінійні (прямолінійні) функції є обмеженими і вихід є пропорційним до входу. Застосування лінійних передатних функцій було проблемою у ранніх моделях мереж, і їх обмеженість та недоцільність була доведена в книзі Мінскі та Пейперта "Перцептроні".

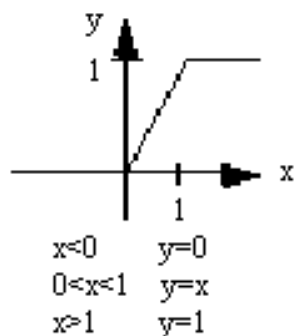
В існуючих нейромережах як передатну функцію використовують сигмоїду, синус, гіперболічний тангенс тощо. На рис. 4 зображені типові передатні функції.

Жорстка порогова функція

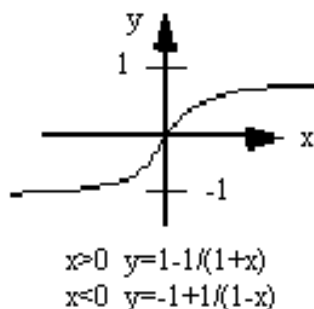


Для простої передатної функції нейромережа може видавати 0 чи 1, 1 чи -1 або інші числові комбінації. Передатна функція в таких випадках є пороговою або «жорстким обмежувачем» (рис. 4а).

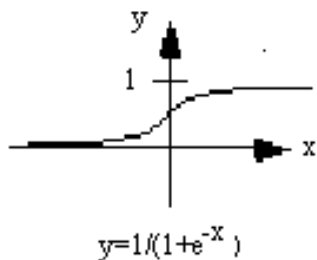
Лінійна з насиченням



Передатна функція лінійна з насиченням віддзеркалює вхід всередині заданого діапазону і діє як жорсткий обмежувач поза межами цього діапазону. Це лінійна функція, яка відсікається до мінімальних та максимальних значень, роблячи її нелінійною (рис. 4б).



Сигмоїда або S-подібна крива наближує мінімальне та максимальне значення у асимптотах. Вона називається сигмоїдою (рис. 4в), коли її діапазон $[0, 1]$, або гіперболічним тангенсом (рис. 4г), при діапазоні $[-1, 1]$. Важливою рисою сигмоїд є неперервність функцій та їх похідних. Застосування сигмоїдних функцій надає добрі результати і має широке застосування.



Для різних нейромереж можуть вибиратись інші передатні функції.

Після обробки сигналу, нейрон на виході має результат передатної функції, який надходить на входи інших нейронів або до зовнішнього з'єднання, як це передбачається структурою нейромережі.

Архітектура з'єднань штучних нейронів

Штучні нейромережі конструюються з базового блоку - **штучного нейрону**. Іншою властивістю нейромереж є величезна кількість зв'язків, які пов'язують окремі нейрони. Групування нейронів у мозку людини забезпечує обробку інформації динамічним, інтерактивним та самоорганізуючим шляхом.

Біологічні нейронні мережі з мікроскопічних компонентів існують у тривимірному просторі і здатні до різноманітних з'єднань. Але для реалізації штучних мереж присутні фізичні обмеження.

Об'єднуючись у мережі, штучні нейрони утворюють систему обробки інформації, яка забезпечує ефективну адаптацію моделі до постійних змін з боку зовнішнього середовища. В процесі функціонування мережі відбувається перетворення вхідного вектора сигналів у вихідний. Конкретний вид перетворення визначається архітектурою нейромережі, характеристиками нейронних елементів, засобами керування та синхронізації інформаційних потоків між нейронами.

Важливим фактором ефективності мережі є встановлення оптимальної кількості нейронів та типів зв'язків між ними.

Для опису нейромереж використовують кілька усталених термінів, які в різних джерелах можуть мати різне трактування, зокрема:

Структура нейромережі - спосіб зв'язків нейронів у нейромережі.

Архітектура нейромережі - структура нейромережі та типи нейронів.

Парадигма нейромережі - спосіб навчання та використання, іноді містить поняття архітектури.

На базі однієї архітектури може бути реалізовано різні парадигми нейромережі і навпаки.

Серед відомих архітектурних рішень виділяють групу **слабозв'язаних** нейронних мереж, у випадку, коли кожний нейрон мережі зв'язаний лише із сусідніми. В **повнозв'язаних нейромережах** входи кожного нейрона зв'язані з виходами всіх решти нейронів.

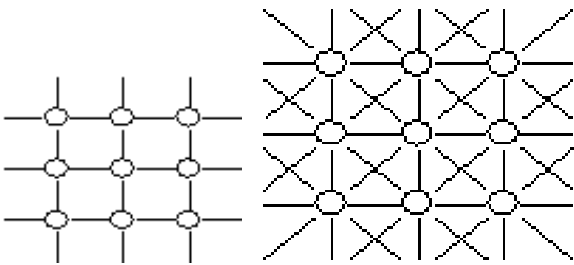


Рис. 5а. Слабозв'язані нейромережі

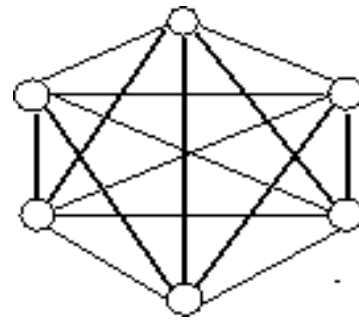


Рис. 5б. Повнозв'язані нейромережі

Самим поширеним варіантом архітектури є багатошарові мережі. Нейрони в даному випадку об'єднуються у прошарки з єдиним вектором вхідних сигналів. Зовнішній вхідний вектор подається на вхідний прошарок нейронної мережі (рецептори). Виходами нейронної мережі є вихідні сигнали останнього прошарку (ефектори). Окрім вхідного та вихідного прошарків, нейромережа має один або кілька прихованих прошарків нейронів, які не мають контактів із зовнішнім середовищем.

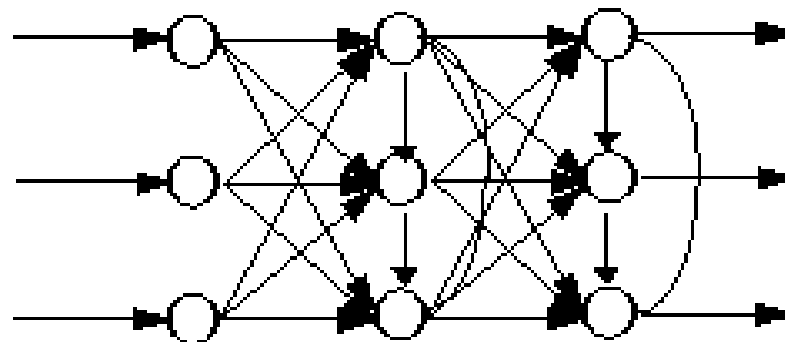
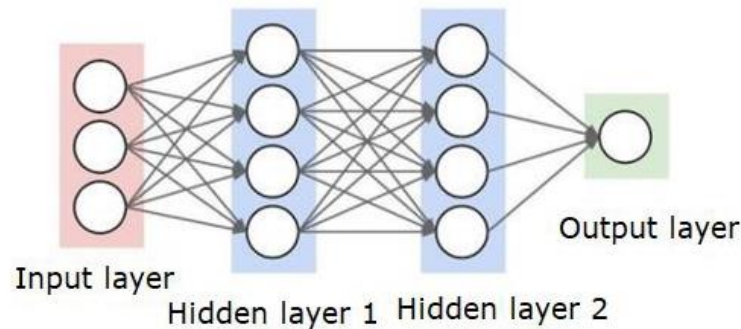


Рис. 6. Багатошаровий тип з'єднання нейронів

Зв'язки між нейронами різних прошарків називають *проективними*.

Зв'язки між нейронами одного прошарку називають *бічними (латеральними)*.

На рис. 6 показана типова структура штучних нейромереж. Хоча існують мережі, які містять лише один прошарок, або навіть один елемент, більшість застосувань вимагають мережі, які містять як мінімум три типи прошарків - вхідний, прихований та вихідний. Прошарок вхідних нейронів отримує дані або з вхідних файлів, або безпосередньо з електронних давачів. Вихідний прошарок пересилає інформацію безпосередньо до зовнішнього середовища, до вторинного комп'ютерного процесу, або до інших пристроїв. Між цими двома прошарками може бути багато прихованих прошарків, які містять багато нейронів в різноманітних зв'язаних структурах. Входи та виходи кожного з прихованих нейронів сполучені з іншими нейронами.

Важливим аспектом нейромереж є **напрямок зв'язку** від одного нейрону до іншого: Зв'язки скеровані від вхідних прошарків до вихідних називаються **аферентними**, Зв'язки в зворотному напрямку називаються **еферентними**.

В більшості мереж кожен нейрон прихованого прошарку отримує сигнали від всіх нейронів попереднього прошарку чи від нейронів вхідного прошарку. Після виконання операцій над сигналами, нейрон передає свій вихід до всіх нейронів наступних прошарків, забезпечуючи передачу вперед (feedforward) на вихід.

При зворотному зв'язку, вихід нейронів прошарку скеровується до нейронів попереднього прошарку (рис. 7).

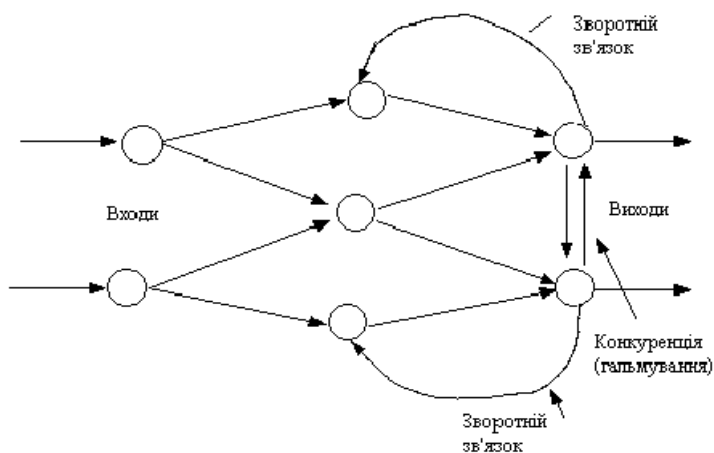
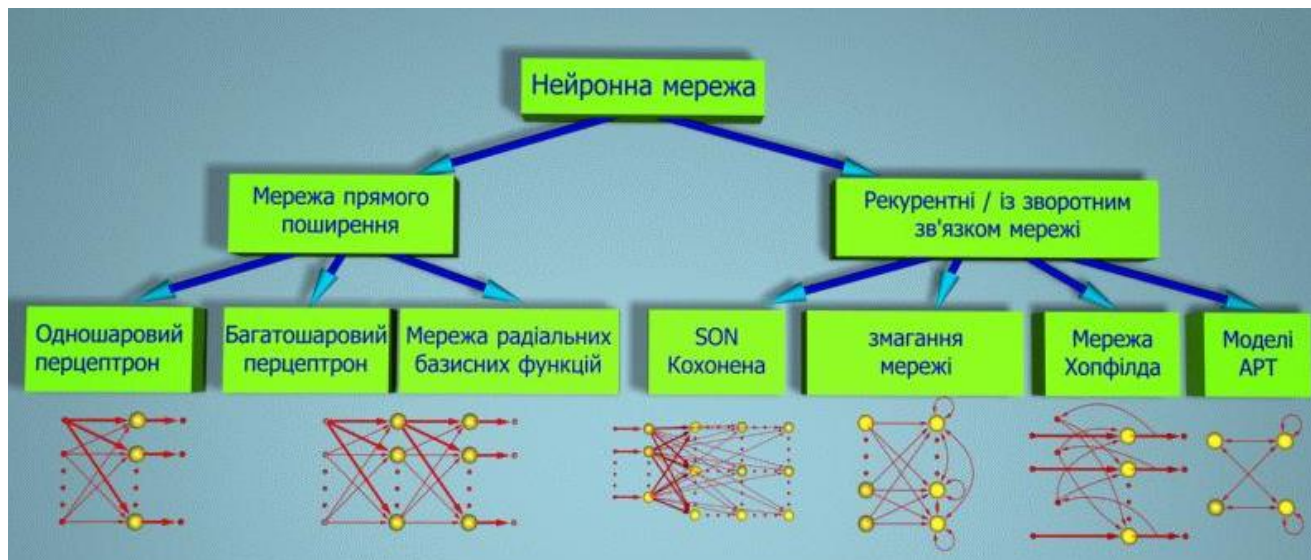


Рис.7

Напрямок зв'язків нейронів має значний вплив на роботу мережі. Більшість програмних нейромереж дозволяють користувачу додавати, вилучати та керувати з'єднаннями як завгодно. Корегуючи параметри, можна налаштувати зв'язки як на посилення так і на послаблення величини сигналів.

За архітектурою зв'язків, більшість відомих нейромереж можна згрупувати у два великих класи:



Мережі прямого поширення (з односторонніми послідовними зв'язками).

Мережі зворотного поширення (з рекурентними зв'язками).

Типові архітектури нейронних мереж

Мережі прямого поширення

- Перцептрони
- Мережа Back Propagation
- Мережа зустрічного поширення
- Карта Кохонена

Рекурентні мережі

- Мережа Хопфілда
- Мережа Хемінга
- Мережа адаптивної резонансної теорії
- Двостороння асоціативна пам'ять

Мережі *прямого поширення* відносять до *статичних*, тут на входи нейронів надходять вхідні сигнали, які не залежать від попереднього стану мережі.

Рекурентні мережі вважаються динамічними, оскільки за рахунок зворотних зв'язків (петель) входи нейронів модифікуються в часі, що призводить до зміни станів мережі.

Навчання штучної нейронної мережі

Оригінальність нейромереж, як аналога біологічного мозку, полягає у здібності до навчання за прикладами, що складають навчальну множину. Процес навчання нейромереж розглядається як налаштування архітектури та вагових коефіцієнтів синаптичних зв'язків відповідно до даних навчальної множини для ефективного вирішення поставленої задачі.

Для навчання нейромереж можливо:

Навчання з вчителем (контрольоване навчання)

Навчання без вчителя (неконтрольоване навчання)

Контрольоване навчання

Більшість реалізацій нейромереж використовують контрольоване навчання, де вихід, що змінюється, постійно порівнюється з бажаним виходом. Вагові коефіцієнти зв'язків на початку встановлюються випадково (ініціалізація мережі), але під час наступних ітерацій коректуються, щоб досягти близької відповідності між бажаним та поточним виходами. Такі методи навчання націлені на мінімізацію поточних похибок всіх елементів обробки, що відбувається завдяки неперервній зміні синаптичних ваг до досягнення прийнятної точності мережі.

Перед використанням, нейромережа з контрольованим навчанням повинна бути навченою. Фаза навчання займає певний час. Навчання вважається закінченим при досягненні нейромережею визначеного користувачем рівня ефективності і бажаної статистичної точності. Після навчання вагові коефіцієнти зв'язків фіксуються для подальшого застосування. Деякі типи мереж дозволяють під час використання продовжувати навчання, і це допомагає мережі адаптуватись до змінних умов.

Навчальні множини повинні бути достатньо великими, щоб містити всю необхідну інформацію для виявлення важливих особливостей і зв'язків. Навчальні приклади повинні містити широке різноманіття даних. Якщо мережа навчається лише для одного прикладу, вагові коефіцієнти, що старанно встановлено для цього прикладу, радикально змінюються у навчанні для наступного прикладу. Попередні приклади при навчанні наступних просто забуваються. В результаті система повинна навчатись всьому разом, знаходячи найкращі вагові коефіцієнти для загальної множини прикладів.

Наприклад, у навчанні системи розпізнавання піксельних образів для десяти цифр, які представлені двадцятьма прикладами кожної цифри, всі приклади цифри "сім" не доцільно представляти послідовно. Краще надати мережі спочатку один тип представлення всіх цифр, потім другий тип і так далі.

Головною компонентою для успішної роботи мережі є представлення і кодування вхідних і вихідних даних. **Штучні мережі працюють лише з числовими вхідними даними, отже, необроблені дані, що надходять із зовнішнього середовища повинні перетворюватись.** Важливою є **нормалізація даних**, тобто приведення всіх значень даних до єдиного діапазону. Нормалізація виконується шляхом ділення кожної компоненти вхідного вектора на довжину вектора, що перетворює вхідний вектор в

одиничний. Попередня обробка зовнішніх даних, отриманих за допомогою сенсорів, у машинний формат є спільною і легко доступною для стандартних комп'ютерів.

Якщо після контрольованого навчання нейромережа ефективно опрацьовує дані навчальної множини, важливим стає її ефективність при роботі з даними, які не використовувались для навчання. У випадку отримання незадовільних результатів для тестової множини, навчання продовжується. Тестування використовується для забезпечення запам'ятовування не лише даних заданої навчальної множини, але і створення загальних образів, що можуть міститись в даних.

Неконтрольоване навчання

Неконтрольоване навчання може бути великим надбанням у майбутньому. Воно проголошує, що комп'ютери можуть самонавчатись у справжньому роботизованому сенсі. На даний час, неконтрольоване навчання використовується в мережах відомих, як самоорганізовані карти (self organizing maps). Мережі не використовують зовнішніх впливів для коректування своїх ваг і внутрішньо контролюють свою ефективність, шукаючи регулярність або тенденції у вхідних сигналах та здійснюють адаптацію відповідно до навчальної функції. Навіть без повідомлення правильності чи неправильності дій, мережа повинна мати інформацію відносно власної організації, яка закладена у топологію мережі та навчальні правила.

Алгоритм неконтрольованого навчання скеровано на знаходження близькості між групами нейронів, які працюють разом. Якщо зовнішній сигнал активує будь-який вузол в групі нейронів, дія всієї групи в цілому збільшується. Аналогічно, якщо зовнішній сигнал в групі зменшується, це приводить до гальмуючого ефекту на всю групу.

Основу для навчання формує конкуренція між нейронами. Навчання конкуруючих нейронів підсилює відгуки певних груп на певні сигнали. Це пов'язує групи між собою та відгуком. При конкуренції змінюються ваги лише нейрона-переможця.

Оцінки навчання

Оцінка ефективності навчання нейромережі залежить від кількох керованих факторів, важливими з яких є: ємність, складність зразків і обчислювальна складність.

Ємність показує, скільки зразків може запам'ятати мережа, і які межі прийняття рішень можуть бути на ній сформовані.

Складність зразків визначає кількість навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення.

Обчислювальна складність напряму пов'язана з потужністю комп'ютера.

Основні етапи розв'язання задач за допомогою нейромереж

- Збір даних для навчання;
- Підготовка і нормалізація даних;
- Вибір топології мережі;
- Експериментальний підбір характеристик мережі;
- Експериментальний підбір параметрів навчання;
- Власне навчання;
- Перевірка адекватності навчання;
- Коректування параметрів, остаточне навчання;
- Вербалізація мережі з метою подальшого використання.

Розглянемо докладніше деякі з цих етапів.

Збір даних для навчання

Вибір даних для навчання мережі і їхня обробка є самим складним етапом розв'язання задачі. **Набір даних для навчання повинний задовольняти декільком критеріям:**

- **Репрезентативність** — дані повинні ілюструвати дійсне положення речей у предметній області;
- **Несуперечність** — суперечливі дані в навчальній вибірці призведуть до поганої якості навчання мережі;
- **Обсяг** — як правило, число записів у вибірці повинне на *кілька порядків перевершувати кількість зв'язків між нейронами в мережі*. У протилежному випадку мережа просто «запам'ятає» усю навчальну вибірку і не зможе виконати узагальнення.

Підготовка і нормалізація даних

Вихідні дані перетворюються до виду, у якому їх можна подати на входи мережі. Кожен запис у файлі даних називається навчальною парою або навчальним вектором. Навчальний вектор містить по одному значенню на кожен вхід мережі і, у залежності від типу навчання (із вчителем або без), по одному значенню для кожного виходу мережі. Навчання мережі на «сирому» наборі, як правило, не дає якісних результатів. Існує ряд способів поліпшити «сприйняття» мережі.

- **Нормування** виконується, коли на різні входи подаються дані різної розмірності. Наприклад, на перший вхід мережі подаються величини зі значеннями від нуля до одиниці, а на другий — від ста до тисячі. При відсутності нормування значення на другому вході будуть завжди робити істотно більший вплив на вихід мережі, чим значення на першому вході. При нормуванні розмірності усіх вхідних і вихідних даних зводяться воедино;

- **Квантування** виконується над безперервними величинами, для яких виділяється кінцевий набір дискретних значень. Наприклад, квантування використовують для завдання частот звукових сигналів при розпізнаванні мови;

- **Фільтрація** виконується для «зашумлених» даних.

Крім того, велику роль грає саме представлення як вхідних, так і вихідних даних. Припустимо, мережа навчається розпізнаванню букв на зображеннях і має один числовий вихід — номер букви в алфавіті. У цьому випадку мережа одержить неправильне уявлення про те, що букви з номерами 1 і 2 більш схожі, чим букви з номерами 1 і 3, що, загалом, невірно. Для того, щоб уникнути такої ситуації, використовують топологію мережі з великим числом виходів, коли кожен вихід має свій зміст. Чим більше виходів у мережі, тим більша відстань між класами і тим складніше їх поплутати.

Вибір топології мережі

Вибирати тип мережі необхідно виходячи з постановки задачі і наявних даних для навчання. Для навчання з учителем потрібна наявність для кожного елемента вибірки «експертної» оцінки. Іноді одержання такої оцінки для великого масиву даних просто неможливо. У цих випадках природним вибором є мережа, що навчається без учителя, наприклад, така як самоорганізуюча карта Кохонена або нейрона мережа Хопфілда. При розв'язанні інших задач, таких як прогнозування часових рядів, експертна оцінка вже утримується у вихідних даних і може бути виділена при їхній обробці. У цьому випадку можна використовувати багатошаровий перцептрон або мережу Ворда.

Експериментальний підбір характеристик мережі

Після вибору загальної структури потрібно експериментально підібрати параметри мережі. Для мереж, подібних перцептроні, це буде число шарів, число блоків у схованих шарах (для мереж Ворда), наявність або відсутність обхідних з'єднань, передатні функції нейронів. При виборі кількості шарів і нейронів у них варто виходити з того, що здатності мережі до узагальнення тим вище, чим більше сумарне число зв'язків між нейронами. З іншого боку, число зв'язків обмежене зверху кількістю записів у навчальних даних. Експериментальний підбір параметрів навчання Після вибору конкретної топології, необхідно вибрати параметри навчання нейронної мережі. Цей етап особливо важливий для мереж, які навчаються з учителем. Від правильного вибору параметрів залежать не тільки те, наскільки швидко відповіді мережі будуть сходитися до правильних відповідей. Наприклад, вибір низької швидкості навчання збільшить час сходження, однак іноді дозволяє уникнути паралічу мережі. Збільшення моменту навчання може привести як до збільшення, так і до зменшення часу збіжності, у залежності від форми поверхні похибки. Виходячи з такого суперечливого впливу параметрів, можна зробити висновок, що їх значення потрібно вибирати експериментально, керуючись при цьому критерієм завершення навчання (наприклад, мінімізація похибки або обмеження за часом навчання).

Власне навчання мережі

У процесі навчання мережа у визначеному порядку переглядає навчальну вибірку. Порядок перегляду може бути послідовним, випадковим і т.д. Мережі, які навчаються без учителя, переглядають вибірку тільки один раз. При навчанні з учителем мережа переглядає вибірку багато разів, при цьому один повний прохід по вибірці називається епохою навчання.

Зазвичай набір вихідних даних поділяють на дві частини — власне **навчальну вибірку і тестові дані**; принцип поділу може бути довільним. Навчальні дані подаються мережі для навчання, а перевіірочні використовуються для розрахунку похибки мережі (перевіірочні дані ніколи для навчання мережі не застосовуються). Таким чином, якщо на перевіірочних даних похибка зменшується, то мережа дійсно виконує узагальнення.

Якщо похибка на навчальних даних продовжує зменшуватися, а похибка на тестових даних збільшується, виходить, мережа перестала виконувати узагальнення і просто «запам'ятовує» навчальні дані. Це явище називається *перенавчанням мережі або оверфітінгом*. У таких випадках навчання зазвичай припиняють. У процесі навчання можуть проявитися інші проблеми, такі як *параліч або влучення мережі в локальний мінімум поверхні похибок*. Неможливо заздалегідь пророчити прояв тієї або іншої проблеми, так само як і дати однозначні рекомендації до їх розв'язання.

Перевірка адекватності навчання

Навіть у випадку успішного, на перший погляд, навчання мережа не завжди навчається саме тому, чого від неї хотів творець. Відомий випадок, коли мережа навчалася розпізнаванню зображень танків по фотографіях, однак пізніше з'ясувалося, що всі танки були сфотографовані на тому самому тлі. У результаті мережа «навчилася» розпізнавати цей тип ландшафту, замість того, щоб «навчитися» розпізнавати танки.

Таким чином, мережа «розуміє» не те, що від неї було потрібно, а те, що найпростіше узагальнити.

2. Деякі типи нейронних мереж

Перцептрон Розенблата

Першою моделлю нейромереж вважають перцептрон Розенблата. Теорія перцептронів є основою для багатьох типів штучних нейромереж прямого поширення і вони є класикою для вивчення.

Одношаровий перцептрон здатний розпізнавати найпростіші образи. Окремий нейрон обчислює зважену суму сигналів вхідних елементів, віднімає значення зсуву і пропускає результат через жорстку порогову функцію, вихід якої дорівнює +1 чи -1. В залежності від значення вихідного сигналу приймається рішення:

- +1 - вхідний сигнал належить до класу А,
- 1 - вхідний сигнал належить до класу В.

На рис. 5.1 показано схему одношарового перцептрона, графік передатної функції і схему вирішальних областей, створених у багатовимірному просторі вхідних сигналів. Вирішальні області визначають, які вхідні образи будуть віднесені до класу А, які - до класу В. Перцептрон, що складається з одного нейрона, формує дві вирішальні області, які розділено гіперплощиною.

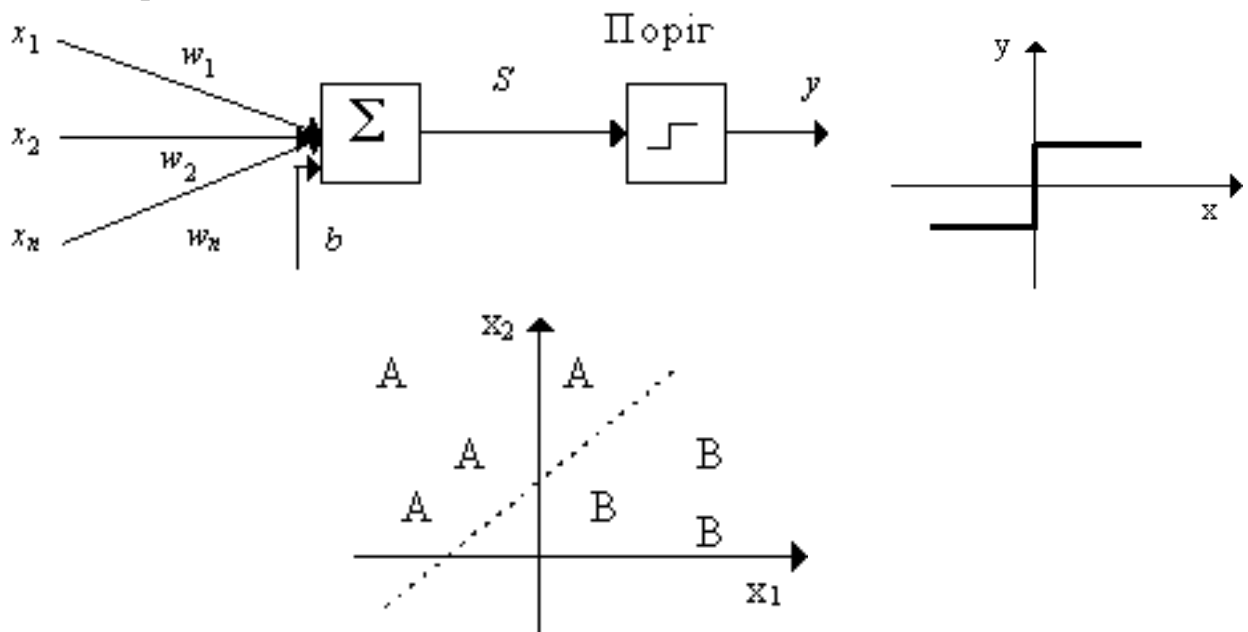


Рис. 5. 1. Схема нейрона, графік передатної функції і поділяюча поверхня

На рис.5.1 показано випадок з розмірністю вихідного сигналу - 2. Поділяюча поверхня є прямою лінією на площині. Рівняння, що задає поділяючу пряму, залежить від значень синаптичних ваг і зсуву.

Алгоритм навчання одношарового перцептрона

1. Ініціалізація синаптичних ваг і зсуву: синаптичні ваги приймають малі випадкові значення.

2. Пред'явлення мережі нового вхідного і бажаного вихідного сигналів: вхідний сигнал $x=(x_1, x_2, \dots, x_n)$ пред'являється нейрону разом з бажаним вихідним сигналом d .

3. Обчислення вихідного сигналу нейрона:

$$y(t) = f\left(\sum_{i=1}^N w_i(t)x_i(t) - b\right)$$

4. Налаштування значень ваг:

$$w_i(t+1) = w_i(t) + r[d(t) - y(t)]x_i(t), i=1, \dots, N$$

$$d(t) = \begin{cases} +1, & \text{вихідний клас А} \\ -1, & \text{вихідний клас В} \end{cases}$$

де $w_i(t)$ - вага зв'язку від i -го елемента вхідного сигналу до нейрона в момент часу t ,
 r - швидкість навчання (менше 1);

$d(t)$ - бажаний вихідний сигнал.

Якщо мережа приймає правильне рішення, синаптичні ваги не модифікуються.

5. Перехід до кроку 2.

Тип вхідних сигналів: бінарні чи аналогові (дійсні).

Розмірності входу і виходу обмежені при програмній реалізації тільки можливостями обчислювальної системи, на якій моделюється нейронна мережа, при апаратній реалізації - технологічними можливостями.

Області застосування: розпізнавання образів, класифікація.

Недоліки. Примітивні поділяючі поверхні (гіперплощини) дають можливість вирішувати лише найпростіші задачі розпізнавання.

Переваги. Програмні та апаратні реалізації моделі прості. Простий і швидкий алгоритм навчання.

Модифікації. Багатошарові перцептрони дають можливість будувати складні поділяючі поверхні і є більш поширеними.

На рис.5.2 показана схема перцептрона з декількома входами та виходами.

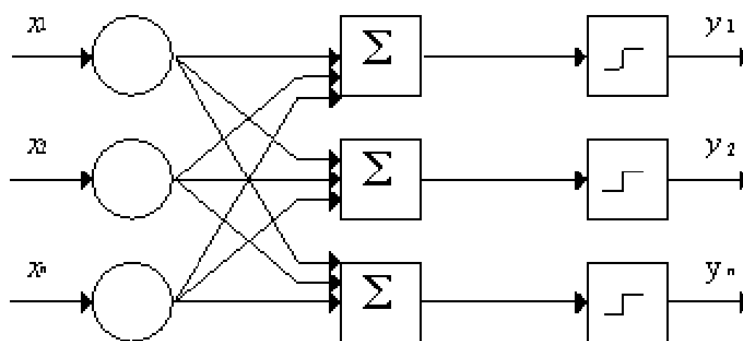


Рис. 5.2. Перцептрон з декількома входами та виходами

Нейромережа зворотного поширення похибки (Back Propagation)

Архітектура *FeedForward BackPropagation* була розроблена на початку 1970-х років декількома незалежними авторами: Вербор (*Werbos*); Паркер (*Parker*); Румельгарт (*Rumelhart*), Хінтон (*Hinton*) та Вільямс (*Williams*). На даний час, парадигма *BackPropagation* є популярною, ефективною та легкою моделлю навчання для складних, багатошарових мереж. Вона використовується в різних типах застосувань і породила великий клас нейромереж з різними структурами та методами навчання.

Типова мережа *BackPropagation* має вхідний прошарок, вихідний прошарок та принаймні один прихований прошарок. Теоретично, обмежень відносно числа прихованих прошарків не існує, але практично застосовують один або два. На рис. 5.3 представлена схема багатошарового перцептрона

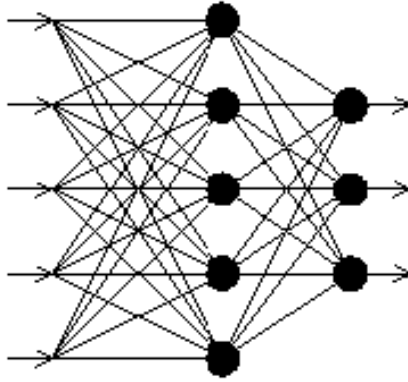


Рис. 5.3 Схеми багатошарового перцептрона

Нейрони організовано в пошарову структуру з прямою передачею (вперед) сигналу. Кожний нейрон мережі продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення. Мережа може моделювати функцію практично будь якої складності, причому число прошарків і число нейронів у кожному прошарку визначають складність функції.

Важливим при моделюванні мережі є **визначення числа проміжних прошарків і числа нейронів в них**. Більшість дослідників та інженерів використовують **загальні правила**, зокрема:

- Кількість входів та виходів мережі визначаються кількістю вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо. На відміну від зовнішніх прошарків, число нейронів прихованого прошарку $n_{\text{прих}}$ обирається емпіричним шляхом. В більшості випадків достатньою кількістю нейронів буде $n_{\text{прих}} \leq n_{\text{вх}} + n_{\text{вих}}$, де $n_{\text{вх}}$, $n_{\text{вих}}$ - кількість нейронів у вхідному і, відповідно, у вихідному прошарках.
- Якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого прошарку повинна також збільшитись.
- Якщо процес, що моделюється, може розділятися на багато етапів, потрібен додатковий прихований прошарок (прошарки). Якщо процес не розділяється на етапи, тоді додаткові прошарки можуть допустити перезапам'ятовування і, відповідно, невірне загальне рішення.

Після того, як визначено число прошарків і число нейронів в кожному з них, потрібно знайти значення для синаптичних ваг і порогів мережі, які спроможні мінімізувати похибку спродукованого результату. Саме для цього існують алгоритми навчання, де відбувається підгонка моделі мережі до наявних навчальних даних. Похибка для конкретної моделі мережі визначається шляхом проходження через мережу всіх навчальних прикладів і порівняння спродукованих вихідних значень з бажаними значеннями. Множина похибок створює функцію похибок, значення якої можна розглядати, як похибку мережі. В якості функції похибок найчастіше використовують суму квадратів похибок.

Для кращого розуміння алгоритму навчання мережі *Back Propagation* потрібно роз'яснити *поняття поверхні станів*. Кожному значенню синаптичних ваг і порогів мережі (вільних параметрів моделі кількістю N) відповідає один вимір в багатовимірному просторі. $N+1$ -ий вимір відповідає похибці мережі. Для різноманітних сполучень ваг відповідну похибку мережі можна зобразити точкою в $N+1$ -вимірному

просторі, всі ці точки утворюють деяку поверхню - поверхню станів. Мета навчання нейромережі полягає в знаходженні на багатовимірній поверхні найнижчої точки.

Поверхня станів має складну будову і досить неприємні властивості, зокрема, наявність локальних мінімумів (точки, найнижчі в своєму певному околі, але вищі від глобального мінімуму), пласкі ділянки, сідлові точки і довгі вузькі яри. Аналітичними засобами неможливо визначити розташування глобального мінімуму на поверхні станів, тому навчання нейромережі по суті полягає в дослідженні цієї поверхні.

Відштовхуючись від початкової конфігурації ваг і порогів (від випадково обраної точки на поверхні), алгоритм навчання поступово відшукує глобальний мінімум. Обчислюється вектор градієнту поверхні похибок, який вказує напрямок найкоротшого спуску по поверхні з заданої точки. Якщо трошки просунуться по ньому, похибка зменшиться. Зрештою алгоритм зупиняється в нижній точці, що може виявитись лише локальним мінімумом (в ідеальному випадку - глобальним мінімумом).

Складність полягає у виборі довжини кроків. При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути рішення, або піти в неправильному напрямку. При маленькому кроці, правильний напрямок буде виявлено, але зростає кількість ітерацій. На практиці розмір кроку береться пропорційним крутизни схилу з деякою константою - швидкістю навчання. Правильний вибір швидкості навчання залежить від конкретної задачі і здійснюється дослідним шляхом. Ця константа може також залежати від часу, зменшуючись по мірі просування алгоритму.

Алгоритм діє ітеративно, його кроки називаються епохами. На кожній епосі на вхід мережі по черзі подаються всі навчальні приклади, вихідні значення мережі порівнюються з бажаними значеннями і обчислюється похибка. Значення похибки, а також градієнту поверхні станів використовують для корекції ваг, і дії повторюються. Процес навчання припиняється або коли пройдена визначена кількість епох, або коли похибка досягає визначеного рівня малості, або коли похибка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій зупинки).

Алгоритм навчання мережі

1. Ініціалізація мережі: вагові коефіцієнти і зсуви мережі приймають малі випадкові значення.

2. Визначення елемента навчальної множини: (вхід - вихід). Входи $(x_1, x_2 \dots x_N)$, повинні розрізнятися для всіх прикладів навчальної множини.

3. Обчислення вихідного сигналу:

$$S_{im} = \sum_{j=1}^{N_{m-1}} w_{ij} y_{j,m-1} - b_{im}$$
$$y_{im} = f(S_{im})$$
$$i_m = 1, 2, \dots, N_m, m = 1, 2, \dots, L$$

де S - вихід суматора, w - вага зв'язку, y - вихід нейрона, b - зсув, i - номер нейрона, N - число нейронів у прошарку, m - номер прошарку, L - число прошарків, f - передатна функція.

4. Налаштування синаптичних ваг:

$$w_{ij}(t+1) = w_{ij}(t) + r g_j x_i$$

де w_{ij} - вага від нейрона i або від елемента вхідного сигналу i до нейрона j у момент часу t , x_i - вихід нейрона i , r - швидкість навчання, g_j - значення похибки для нейрона j .

Якщо нейрон з номером j належить останньому прошарку, тоді

$$g_j = y_j(1 - y_j)(d_j - y_j)$$

де d_j - бажаний вихід нейрона j , y_j - поточний вихід нейрона j .

Якщо нейрон з номером j належить одному з прошарків з першого по передостанній, тоді

$$g_j = x'_j(1 - x'_j) \sum_k g_k W_{jk}$$

де k пробігає всі нейрони прошарку з номером на одиницю більше, ніж у того, котрому належить нейрон j .

Зовнішні зсуви нейронів b налаштовуються аналогічним образом.

Тип вхідних сигналів: цілі чи дійсні.

Тип вихідних сигналів: дійсні з інтервалу, заданого передатною функцією нейронів.

Тип передатної функції: сигмоїдальна. Сигмоїдальні функції є монотонно зростаючими і мають відмінні від нуля похідні по всій області визначення. Ці характеристики забезпечують правильне функціонування і навчання мережі.

Області застосування. Розпізнавання образів, класифікація, прогнозування.

Недоліки. Низька швидкість навчання.

Переваги. Ефективний та популярний алгоритм для вирішення численних практичних задач.

Модифікації. Модифікації алгоритму зворотного поширення зв'язані з використанням різних функцій похибки, різних процедур визначення напрямку і величини кроку.

Мережа Кохонена

Мережа розроблена Тойво Кохоненом на початку 1980-х рр. і принципово відрізняється від розглянутих вище мереж, оскільки використовує неконтрольоване навчання і навчальна множина складається лише із значень вхідних змінних.

Мережа розпізнає кластери в навчальних даних і розподіляє дані до відповідних кластерів. Якщо в подальшому мережа зустрічається з набором даних, несхожим ні з одним із відомих зразків, вона відносить його до нового кластеру. Якщо в даних містяться мітки класів, то мережа спроможна вирішувати задачі класифікації. Мережі Кохонена можна використовувати і в задачах, де класи є відомими - перевага буде у спроможності мережі виявляти подібність між різноманітними класами.

Мережа Кохонена має лише два прошарки: вхідний і вихідний (рис. 1.) її ще називають **самоорганізовуваною картою**. Елементи карти розташовуються в деякому просторі, як правило, двовимірному. Мережа Кохонена навчається методом послідовних наближень. У процесі навчання на входи подаються дані, але мережа при цьому підлаштовується не під еталонне значення виходу, а під закономірності у вхідних даних. Починається навчання з вибраного випадковим чином вихідного розташування центрів.

В процесі послідовної подачі на вхід мережі навчальних прикладів визначається *найбільш схожий нейрон (той, у якого скалярний добуток ваг і поданого на вхід вектора є мінімальним)*. Цей нейрон оголошується переможцем і є центром при підлаштуванні

ваг в сусідніх нейронів. Таке правило навчання передбачає "змагальне" (від «змагання») навчання з врахуванням відстані нейронів від "нейрона-переможця".

Навчання при цьому полягає не в мінімізації помилки, а в підлаштуванні ваг (внутрішніх параметрів нейронної мережі) для найбільшого збігу з вхідними даними.

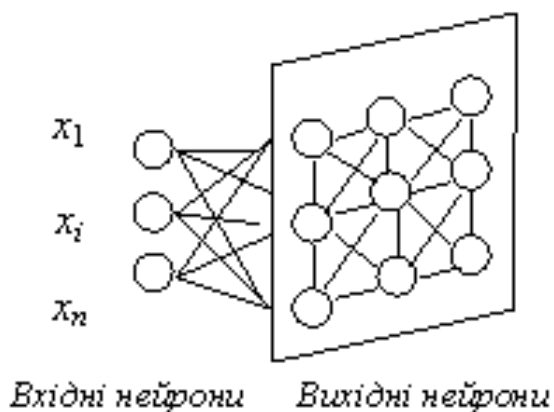


Рис. 1. Мережа Кохонена

Мережа Кохонена навчається методом послідовних наближень. Починаючи з випадковим чином обраного вихідного розташування центрів, алгоритм поступово поліпшується для кластеризації навчальних даних. Основний ітераційний алгоритм Кохонена послідовно проходить ряд епох, на кожній з яких обробляється один приклад з навчальної вибірки. Вхідні сигнали послідовно пред'являються мережі, при цьому бажані вихідні сигнали не визначаються. Після пред'явлення достатнього числа вхідних векторів синаптичні ваги мережі стають здатні визначити кластери. Ваги організуються так, що топологічно близькі вузли реагують до схожих вхідних сигналів.

В результаті роботи алгоритму центр кластера встановлюється в певній позиції, яка задовольняє кластеризовані приклади, для яких даний нейрон є "переможцем". В результаті навчання мережі необхідно визначити міру сусідства нейронів, тобто окіл нейрона-переможця, який представляє кілька нейронів, що оточують нейрон-переможець.

Для реалізації алгоритму необхідно визначити міру сусідства нейронів (околиця нейрона-переможця). На рис. 2 показані зони топологічного сусідства нейронів на карті ознак у різні моменти часу. $NE_j(t)$ - множина нейронів, що вважаються сусідами нейрона j у момент часу t . Зони сусідства зменшуються з часом.

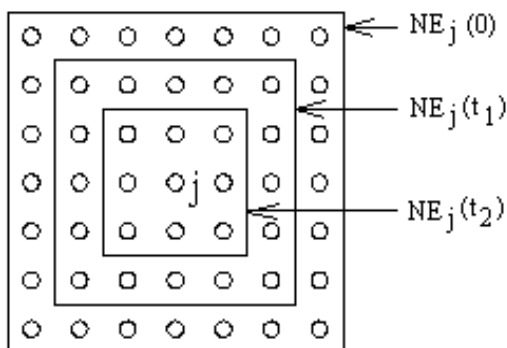
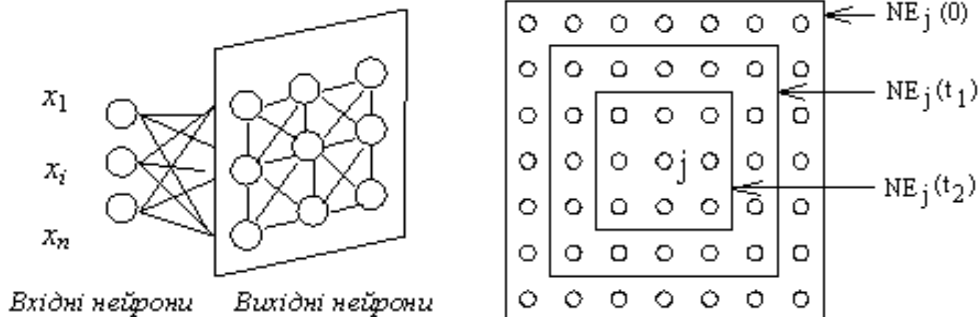


Рис. 2. Зони топологічного сусідства на карті ознак у різні моменти часу

Спочатку до околу належить велике число нейронів, далі її розмір поступово зменшується. Мережа формує топологічну структуру, в якій схожі приклади утворюють групи прикладів, які близько знаходяться на топологічній карті.

Алгоритм функціонування мережі Кохонена:

1. *Ініціалізація мережі.* Ваговим коефіцієнтам мережі надаються малі випадкові значення. Загальне число синаптичних ваг - $M \cdot N$ (див. рис. 1 Початкова зона сусідства показана на рис. 2).



2. *Пред'явлення мережі нового вхідного сигналу.*

3. *Обчислення відстані до всіх нейронів мережі:*

Відстані d_j від вхідного сигналу до кожного нейрона j визначаються за формулою:

$$d_j = \sum_{i=1}^N (x_i(t) \cdot w_{ij}(t))^2$$

де x_i - i -ий елемент вхідного сигналу в момент часу t , $w_{ij}(t)$ - вага зв'язку від i -го елемента вхідного сигналу до нейрона j у момент часу t .

4. *Вибір нейрона з найменшою відстанню:*

Вибирається нейрон-переможець j^* , для якого відстань d_j найменша.

5. *Налаштування ваг нейрона j^* і його сусідів:*

Робиться налаштування ваг для нейрона j^* і всіх нейронів з його околу NE. Нові значення ваг:

$$w_{ij}(t+1) = w_{ij}(t) + r(t)(x_i(t) - w_{ij}(t))$$

де $r(t)$ - швидкість навчання, що зменшується з часом (додатне число, менше одиниці).

6. *Повернення до кроку 2.*

В алгоритмі використовується коефіцієнт швидкості навчання, який поступово зменшується, для тонкішої корекції в новій епосі. В результаті позиція центру встановлюється в певній позиції. Вона задовільним чином кластеризує приклади, для яких даний нейрон є переможцем.

Властивість топологічної впорядкованості досягається в алгоритмі за допомогою використання поняття околу. **Окіл** - це декілька нейронів, що оточують нейрон-переможець. Відповідно до швидкості навчання, розмір околу поступово зменшується,

так, що спочатку до нього належить досить велике число нейронів (можливо вся карта), на самих останніх етапах окіл стає нульовим і складається лише з нейрона-переможця.

В алгоритмі навчання корекція застосовується не тільки до нейрона-переможця, але і до всіх нейронів з його поточного околу. В результаті такої зміни околу, початкові доволі великі ділянки мережі мігрують в бік навчальних прикладів. Мережа формує грубу структуру топологічного порядку, при якій схожі приклади активують групи нейронів, що близько знаходяться на топологічній карті.

З кожною новою епохою швидкість навчання і розмір околу зменшуються, тим самим всередині ділянок карти виявляються більш тонкі розходження, що зрештою призводить до точнішого налаштування кожного нейрона.

Часто навчання зумисно розбивають на дві фази: більш коротку, з великою швидкістю навчання і великих околів, і більш тривалу з малою швидкістю навчання і нульовими або майже нульовими околами.

Після того, як мережа навчена розпізнавати структури даних, її можна використовувати як засіб візуалізації при аналізі даних.

Області застосування. Кластерний аналіз, розпізнавання образів, класифікація.

Недоліки. Мережа може бути використана для кластерного аналізу маючи задалегідь відоме число кластерів.

Переваги. Мережа Кохонена здатна функціонувати в умовах завад, тому що число кластерів фіксоване, ваги модифікуються повільно, налаштування ваг закінчується після навчання.

Мережа зустрічного поширення (CounterPropagation)

Роберт Хехт-Нильсен (Robert Hecht-Nielsen) розробив мережу Counterpropagation як засіб для об'єднання неконтрольованого шару Кохонена з контрольованим вихідним шаром. Мережа призначена для розв'язання складних класифікацій, при мінімізації числа нейронів і часу навчання. Приклад мережі зображено на рис. 4. Односпрямована мережа CounterPropagation має три шари: вхідний шар, самоорганізована карта Кохонена і вихідний шар, що використовує правило "дельта" для зміни вхідних ваг з'єднань. Цей шар називають шаром Гроссберга.

Перша мережа Counterpropagation складалася з двонаправленого відображення між вхідним і вихідним шарами. Дані надходять на вхідний шар для генерації класифікації на вихідному шарі, вихідний шар по черзі приймає додатковий вхідний вектор і генерує вихідну класифікацію на вхідному шарі мережі. Через такий зустрічно-расповсюджений потік інформації мережа одержала свою назву. Багато розроблювачів використовують односпрямований варіант Counterpropagation, коли існує лише один шлях прямого поширення від вхідного до вихідного шару. У мережі зустрічного поширення об'єднані два алгоритми: самоорганізована карта Кохонена і зірка Гроссберга (Grossberg Outstar). Нейрони мережі, що поєднують різні нейропарадигми як будівельні блоки, більш близькі до мозку по архітектурі, чим однорідні структури. Вважається, що в мозку саме каскадні з'єднання модулів різної спеціалізації дозволяють виконувати необхідні обчислення. Кожен елемент вхідного сигналу подається на всі нейрони шару Кохонена. Ваги зв'язків (w_{mn}) утворюють матрицю W . Кожен нейрон шару Кохонена з'єднаний із усіма нейронами шару Гроссберга. Ваги зв'язків (v_{np}) утворюють матрицю ваг V . У процесі навчання мережі зустрічного поширення вхідні вектори асоціюються з відповідними вихідними векторами (двійковими або аналоговими).

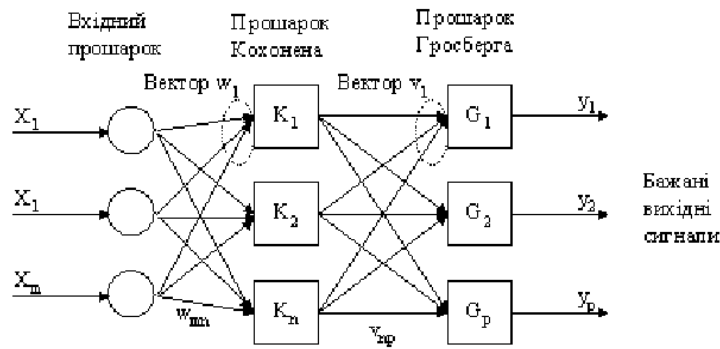


Рис. 4. Мережа зустрічного поширення без зворотних зв'язків

Після навчання мережа формує вихідні сигнали, що відповідають вхідним сигналам. Узагальнююча здатність мережі дає можливість одержувати правильний вихід, коли вхідний вектор неповний або перевернутий.

Навчання мережі

У результаті самонавчання шар здобуває здатність розділяти несхожі вхідні вектори. Який саме нейрон буде активуватися при пред'явленні конкретного вхідного сигналу, заздалегідь важко передбачити. При навчанні шару Кохонена на вхід подається вхідний вектор і обчислюються його скалярні добутки з векторами ваг усіх нейронів. Скалярний добуток є мірою подібності між вхідним вектором і вектором ваг. Нейрон з максимальним значенням скалярного добутку вважається "переможцем" і його ваги підсилюються (ваговий вектор наближається до вхідного).

$$w_n = w_c + r(x - w_c)$$

де w_n - нове значення ваги, що з'єднує вхідний компонент x з нейроном, що виграв, w_c - попереднє значення ваги, r - коефіцієнт швидкості навчання, що спочатку зазвичай дорівнює 0,7 і може поступово зменшуватися в процесі навчання. Це дозволяє робити великі початкові кроки для швидкого грубого навчання і менші кроки при підході до остаточної величини.

Кожна вага, зв'язана з нейроном-переможцем Кохонена, змінюється пропорційно різниці між його величиною і величиною входу, до якого він приєднаний. Напрямок зміни мінімізує різниця між вагою і відповідним елементом вхідного шару. Навчальна множина може містити багато схожих між собою вхідних векторів, і мережа повинна бути навчена активувати один нейрон Кохонена для кожного з них. Ваги цього нейрона є усередненням вхідних векторів, що його активують. Виходи шару Кохонена подаються на входи нейронів шару Гроссберга. Входи нейронів обчислюються як зважена сума виходів шару Кохонена.

Кожна вага коректується лише в тому випадку, якщо він з'єднаний з нейроном Кохонена, що має ненульовий вихід. Величина корекції ваг пропорційна різниці між вагою і необхідним виходом нейрона Гроссберга. Навчання шару Гроссберга - це навчання "із учителем", алгоритм використовує задані бажані виходи.

Функціонування мережі.

У своїй найпростішій формі шар Кохонена функціонує за правилом "переможець одержує все". Для даного вхідного вектора один і тільки один нейрон Кохонена видає логічну одиницю, всі інші видають нуль. Шар Гроссберга функціонує в схожій манері.

Його вихід є зваженою сумою виходів шару Кохонена. Якщо шар Кохонена функціонує так, що лише один вихід дорівнює одиниці, а інші дорівнюють нулеві, то кожен нейрон шару Гроссберга видає величину ваги, що зв'язує цей нейрон з єдиним нейроном Кохонена, чий вихід відмінний від нуля. У повній моделі мережі зустрічного поширення є можливість одержувати вихідні сигнали по вхідним і навпаки. Цим двом діям відповідають пряме і зворотне поширення сигналів.

Області застосування. Розпізнавання образів, відновлення образів (асоціативна пам'ять), стиск даних (із утратами).

Недоліки. Мережа не дає можливості будувати точні апроксимації (точні відображення). У цьому мережа значно уступає мережам зі зворотним поширенням похибки. До недоліків моделі також варто віднести слабкий теоретичний базис модифікацій мережі зустрічного поширення.

Переваги

- Мережа зустрічного поширення проста. Вона дає можливість одержувати статистичні властивості з множини вхідних сигналів. Кохонен довів, що для навченої мережі ймовірність того, що випадково обраний вхідний вектор буде найближчим до будь-якого заданого вагового вектора, дорівнює $1/k$, k - число нейронів Кохонена.

- Мережа піддається швидкому навчанню. Час навчання в порівнянні зі зворотним поширенням може бути в 100 разів менше.

- По своїх можливостях будувати відображення мережа зустрічного поширення значно перевершує одношарові перцептрони.

- Мережа корисна для додатків, де потрібна швидка початкова апроксимація.

- Мережа дає можливість будувати функцію і зворотну до неї, що знаходить застосування при рішенні практичних задач.

Модифікації. Мережі зустрічного поширення можуть розрізнятися способами визначення початкових значень синаптичних ваг.

- Для підвищення ефективності навчання застосовується додавання шуму до вхідних векторів.

- Ще один метод підвищення ефективності навчання - надання кожному нейрону "почуття справедливості". Якщо нейрон стає переможцем частіше, ніж $1/k$ (k - число нейронів Кохонена), то йому тимчасово збільшують поріг, даючи тим самим учитися й іншим нейронам.

- Крім "методу акредитації", при якому для кожного вхідного вектора активується лише один нейрон Кохонена, може бути використаний "метод інтерполяції", при використанні якого ціла група нейронів Кохонена, що мають найбільші виходи, може передавати свої вихідні сигнали в шар Гроссберга. Цей метод підвищує точність відображень, реалізованих мережею.

Мережа Хопфілда

Джон Хопфілд вперше представив свою асоціативну мережу у 1982 р. у Національній Академії Наук. На честь Хопфілда та нового підходу до моделювання, ця мережна парадигма згадується як мережа Хопфілда. Мережа базується на аналогії фізики динамічних систем. Початкові застосування мережі включали асоціативну, або адресовану за змістом пам'ять та вирішували задачі оптимізації.

Мережа Хопфілда використовує три прошарки: вхідний, прошарок Хопфілда та вихідний прошарок. Кожен прошарок має однакову кількість нейронів. Виходи нейронів вхідного прошарку надходять до входів відповідних нейронів прошарку Хопфілда. Тут,

зв'язки мають фіксовані вагові коефіцієнти. Виходи прошарку Хопфілда під'єднуються до входів всіх нейронів прошарку Хопфілда, за винятком самого себе, а також до відповідних елементів у вихідному прошарку. Під час навчання, мережа скеровує дані з вхідного прошарку до прошарку Хопфілда. Прошарок Хопфілда коливається, поки не буде завершена певна кількість циклів, і поточний стан сигналів нейронів прошарку передається на вихідний прошарок. Цей стан відповідає образу, який буде запам'ятовано в мережі.

Навчання мережі Хопфілда вимагає, щоб навчальний образ був представлений на вхідному та вихідному прошарках одночасно. Рекурсивний характер прошарку Хопфілда забезпечує засоби корекції всіх ваг з'єднань. Для правильного навчання мережі відповідні пари "вхід-вихід" мають відрізнятися між собою.

Якщо мережа Хопфілда використовується як пам'ять, що адресується за змістом вона має два головних обмеження.

Число образів, що можна зберегти та точно відтворити є строго обмеженим. Якщо зберігається занадто багато образів, мережа може збігтись до нового неіснуючого образу, відмінному від всіх запрограмованих образів, або не збігтись взагалі. Межа має ємність пам'яті приблизно 15% від числа нейронів у прошарку Хопфілда.

Якщо навчальні приклади є занадто подібними, прошарок Хопфілда може стати нестабільним. Зразок образу вважається нестабільним, якщо він застосовується за нульовий час і мережа збігається до деякого іншого образу з навчальної множини. Ця проблема може бути вирішена вибором навчальних прикладів більш ортогональних між собою.

Структурна схема мережі Хопфілда приведена на рис. 6

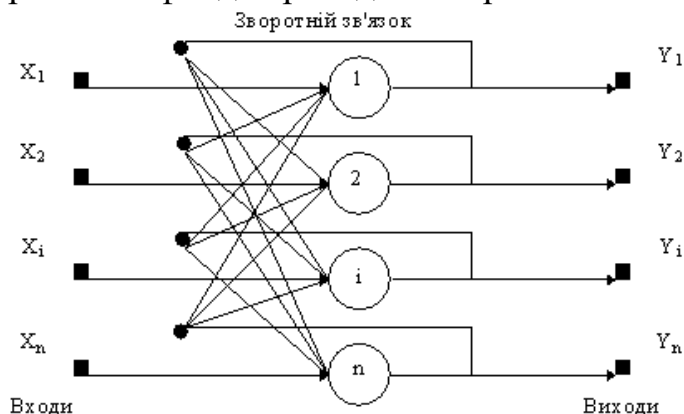


Рис.6. Структурна схема мережі Хопфілда

Для вирішення завдання асоціативної пам'яті є деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують об'єкти або характеристики процесів), який вважається зразковим. Мережа повинна вміти з зашумленого сигналу, поданого на її вхід, виділити ("пригадати" за частковою інформацією) відповідний зразок або "дати висновок" про те, що вхідні дані не відповідають жодному із зразків.

В загальному випадку, будь-який сигнал може бути описаний вектором x_1, x_i, x_n, \dots, n - число нейронів у мережі і величина вхідних і вихідних векторів. Кожний елемент x_i дорівнює або +1, або -1. Позначимо вектор, що описує k -ий зразок, через X_k , а його компоненти, відповідно, - $x_{ik}, k=0, \dots, m-1, m$ - число зразків. Якщо мережа розпізнає (або "пригадає") якийсь зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто $Y = X_k$, де Y - вектор вихідних значень мережі: y_1, y_i, y_n . У протилежному випадку, вихідний вектор не збігається з жодним зразковим.

Якщо, наприклад, сигнали є певним образом зображення, то, відобразивши у графічному виді дані з виходу мережі, можна буде побачити картинку, що цілком збігається з однієї зі зразкових (у випадку успіху) або ж "вільну імпровізацію" мережі (у випадку невдачі).

Алгоритм функціонування мережі

1. На стадії ініціалізації мережі вагові коефіцієнти синапсів встановлюються таким чином:

$$w_{ij} = \begin{cases} \frac{\sum_{k=1}^m x_i^k x_j^k, i \neq j}{0,} & i = j \end{cases}$$

Тут i і j - індекси, відповідно, передсинаптичного і постсинаптичного нейронів; x_i^k , x_{jk} - i -ий і j -ий елементи вектора k -ого зразка.

2. На вході мережі подається невідомий сигнал (t - номер ітерації). Його поширення безпосередньо встановлює значення виходів:

$$y_i(0) = x_i, i = 0 \dots n-1,$$

тому позначення на схемі мережі вхідних сигналів у явному виді носить чисто умовний характер. Нуль у дужках справа від y_i означає нульову ітерацію в циклі роботи мережі.

3. Обчислюється новий стан нейронів

$$S(t + 1) = \sum_{i=1}^n w_{ij} y_j(t), j=0 \dots n-1$$

і нові значення виходів

$$y_j(t + 1) = f |S_j(t + 1)|$$

де f - передатна функція у виді порогової, приведена на рис.7

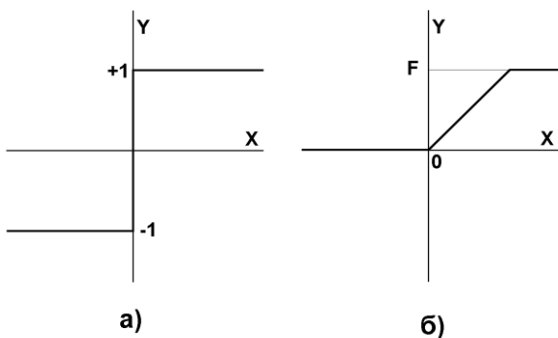


Рис.7. Порогові функції

4. Перевіряємо чи змінилися вихідні значення виходів за останню ітерацію. Якщо так - перехід до пункту 2, інакше (якщо виходи стабілізувались) - кінець. При цьому вихідний вектор являє собою зразок, що найкраще відповідає вхідним даним.

Іноді мережа не може провести розпізнавання і видає на виході неіснуючий образ. Це пов'язано з проблемою обмеженості можливостей мережі. Для мережі Хопфілда число збережених образів m не повинно перевищувати $0.15 * n$ (n - кількість нейронів вихідного прошарку) Крім того, якщо два образи А и Б сильно схожі, вони, можливо, будуть викликати в мережі перехресні асоціації, тобто пред'явлення на вході мережі вектора А призведе до появи на її виходах вектори Б и навпаки.

Мережа Хемінга

Мережа Хемінга (*Hamming*) є розширенням мережі Хопфілда. Ця мережа була розроблена Річардом Ліппманом (*Richard Lippman*) у середині 80-х рр. Мережа Хемінга реалізує класифікатор, що базується на найменшій похибці для векторів двійкових входів, де похибка визначається відстанню Хемінга. Відстань Хемінга визначається як число бітів, які відрізняються між двома відповідними входними векторами фіксованої довжини. Один входний вектор є незашумленим прикладом образу, інший є спотвореним образом. Вектор виходів навчальної множини є вектором класів, до яких належать образи. У режимі навчання входні вектори розподіляються до категорій для яких відстань між зразковими входними векторами та біжучим входним вектором є мінімальною.

Мережа Хемінга має три прошарки: входний прошарок з кількістю вузлів, скільки є окремих двійкових ознак; прошарок категорій (прошарок Хопфілда), з кількістю вузлів, скільки є категорій або класів; вихідний прошарок, який відповідає числу вузлів у прошарку категорій.

Мережа є простою архітектурою прямого поширення з входним рівнем, повністю під'єднаним до прошарку категорій. Кожен нейрон у прошарку категорій є зворотно під'єднаним до кожного нейрона у тому ж самому прошарку і прямо під'єднаним до вихідного нейрону. Вихід з прошарку категорій до вихідного прошарку формується через конкуренцію.

Навчання мережі Хемінга є подібним до методології Хопфілда. На входний прошарок надходить бажаний навчальний образ, а на виході вихідного прошарку надходить значення бажаного класу, до якого належить вектор. Вихід містить лише значення класу до якої належить входний вектор. Рекурсивний характер прошарку Хопфілда забезпечує засоби корекції всіх ваг з'єднань.

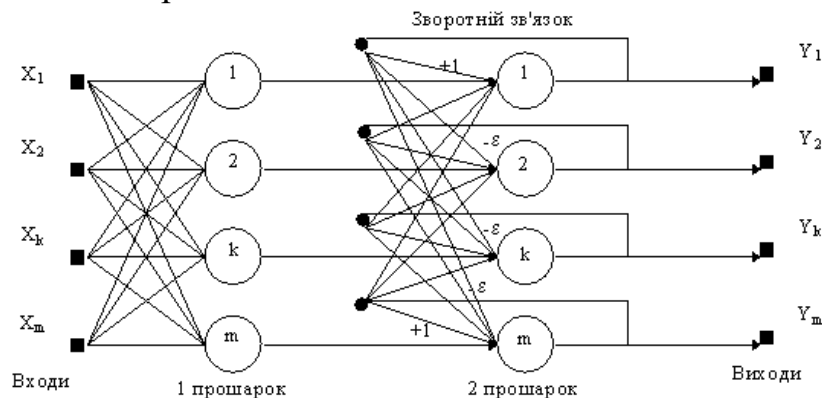


Рис.8.

Алгоритм функціонування мережі Хемінга

1. На стадії ініціалізації ваговим коефіцієнтам першого прошарку і порогу передатної функції присвоюються такі значення:

$$W_{ik} = x_i^k / 2, \quad i=0 \dots n-1, \quad k=0 \dots m-1$$

$$b_k = n / 2, \quad k = 0 \dots m-1$$

Тут x_i^k - i -ий елемент k -ого зразка.

Вагові коефіцієнти гальмуючих синапсів у другому прошарку беруть рівними деякій величині $0 < v < 1/m$. Синапс нейрона, пов'язаний із його ж виходом має вагу $+1$.

2. На входи мережі подається невідомий вектор $x_1, x_i, x_n \dots$. Розраховуються стани нейронів першого прошарку (верхній індекс у дужках указує номер прошарку):

$$y_j^{(1)} = S_j^{(1)} = \sum_{i=1}^n W_{ij} x_i + b_j, \quad j=0 \dots m-1$$

Після цього отримання значення ініціалізують значення виходів другого прошарку:

$$y_j^{(2)} = y_j^{(1)}, j = 0 \dots m-1$$

3. Обчислюються нові стани нейронів другого прошарку:

$$S_j^{(2)}(t+1) = y_j(t) - \varepsilon \sum_{k=1}^m y_k^{(2)}(t), k \neq j, j = 1 \dots m$$

і значення їх виходів:

$$y_j^{(1)}(t+1) = f \left| S_j^{(2)}(t+1) \right|$$

Передатна функція f має вид порога, причому величина b повинна бути достатньо великою, щоб будь-які можливі значення аргументу не призводили до насичення.

4. Перевіряється, чи змінилися виходи нейронів другого прошарку за останню ітерацію. Якщо так - перейти до кроку 3. Інакше - кінець.

Роль першого прошарку є умовною: скориставшись один раз на першому кроці значеннями його вагових коефіцієнтів, мережа більше не вертається до нього, тому перший прошарок може бути взагалі виключений із мережі.

Мережа Хемінга має ряд переваг над мережею Хопфілда. Вона реалізує оптимальний класифікатор мінімуму похибки, якщо похибки вхідних бітів є випадковими та незалежними. Для функціонування мережі Хемінга потрібна менша кількість нейронів, оскільки середній прошарок вимагає лише один нейрон на клас, замість нейрону на кожен вхідний вузол. І, нарешті, мережа Хемінга не страждає від неправильних класифікацій, які можуть трапитись у мережі Хопфілда. В цілому, мережа Хемінга є як швидшою, так і точнішою за мережу Хопфілда.

Висновки

Серед усіх різновидів і типів нейронних мереж необхідно чітко розрізняти мережі з прямими зв'язками та рекурентні мережі. Також слід розрізняти мережі що навчаються з учителем, та такі що здатні навчатися самі.