ЛЕКЦІЯ 5
Тема: Основи штучних нейронних мереж

Питання лекції

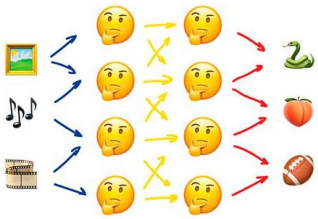
1. **Основні поняття штучних нейронних мереж**
2. **Деякі типи мереж прямого поширення**

ВСТУП

На попередніх лекціях ми вивчили що таке машинне навчання, його складові, провели класифікацію методів машинного навчання та розглянули методи класичного навчання.

Мета машинного навчання - передбачити результат за вхідними даними. Чим різноманітніші вхідні дані, тим простіше машині знайти закономірності і тим точніший результат.

Згадуйте, якщо ми хочемо навчити машину, нам потрібні три речі: **дані, ознаки, алгоритми.**



Neural Networks «У нас є мережа з тисячі шарів, десятки відеокарт, але ми все ще не придумали де це може бути корисним. Нехай малює котиків! »

Сьогодні використовують для:

Замість всіх перелічених вище алгоритмів

Визначення об'єктів на фото і відео

Розпізнавання і синтез мови

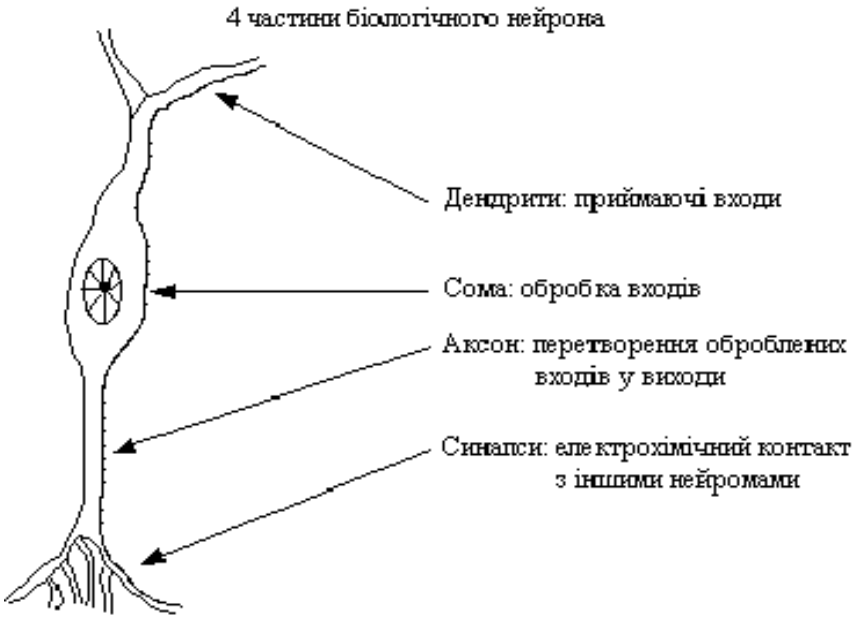
Обробка зображень, перенесення стилю

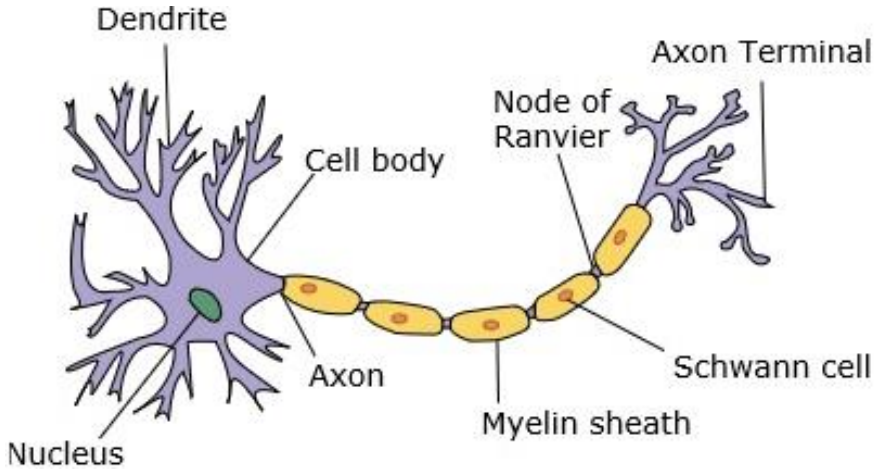
Машинний переклад

Популярні архітектури: Перцептрон, Згорткові Мережі (CNN), Рекурентні Мережі(RNN), Автоенкодери

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, кожний 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 нейроні мережі), що виконують функції, подібні до функцій мозку.

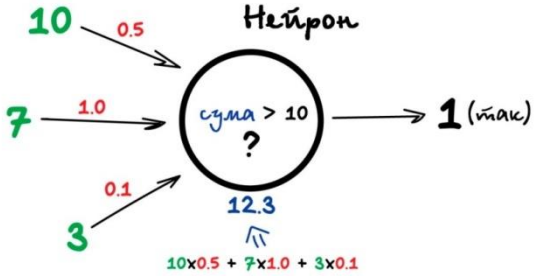
Штучні нейронні мережі є моделями нейронної структури мозку, який здатен сприймати, обробляти, зберігати та продукувати інформацію. Особливістю мозку також є навчання та самонавчання на власному досвіді. Адаптивні системи на основі штучних нейронних мереж дозволяють з успіхом вирішувати проблеми розпізнавання 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

Будь-яка нейромережа - це набір нейронів і зв'язків між ними. Нейрон найкраще уявляти собі просто як функцію з купою входів і одним виходом. Завдання нейрона - взяти цифри зі своїх входів, виконати над ними функцію і віддати результат на вихід. Простий приклад корисного нейрона: знайти суму всіх цифр зі входів, і якщо їх сума більше N - видати на вихід одиницю, інакше - нуль.

Зв'язки - це канали, через які нейрони шлють один одному цифри. Кожен зв'язок має свою вагу - її єдиний параметр, який можна умовно уявити як міцність зв'язку. Коли через зв'язок з вагою 0,5 проходить число 10, воно перетворюється в 5. Сам нейрон не розбирається, що до нього прийшло і сумує все підряд - ось ваги й потрібні, щоб керувати на які входи нейрон повинен реагувати, а на які - ні.

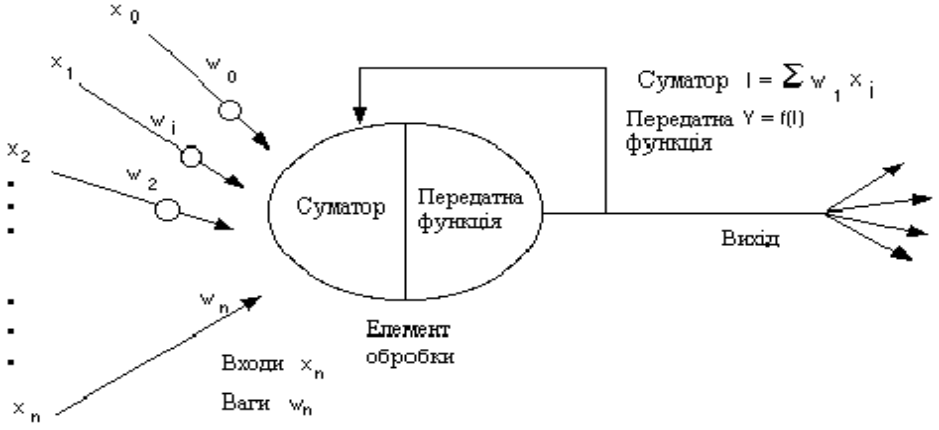


Щоб мережа не перетворилася в анархію, нейрони вирішили пов'язувати не як захочеться, а по шарах. Всередині одного шару нейрони ніяк не пов'язані, але з'єднані з нейронами наступного і попереднього шару. Дані в такій мережі йдуть строго в одному напрямку - від входів першого шару до виходів останнього.

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.



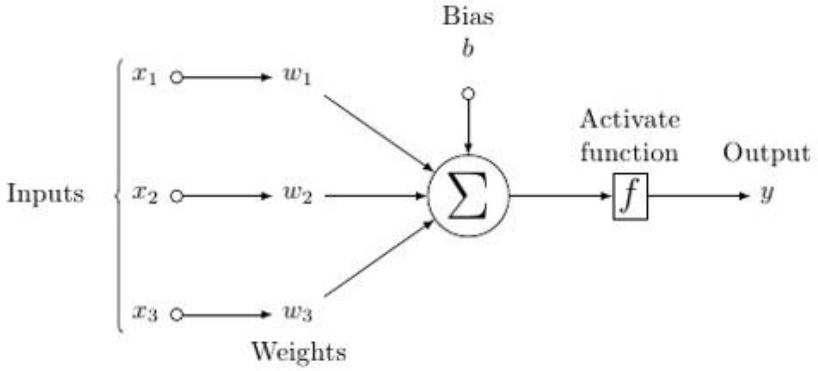


Рис. 2. Базовий штучний нейрон

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.

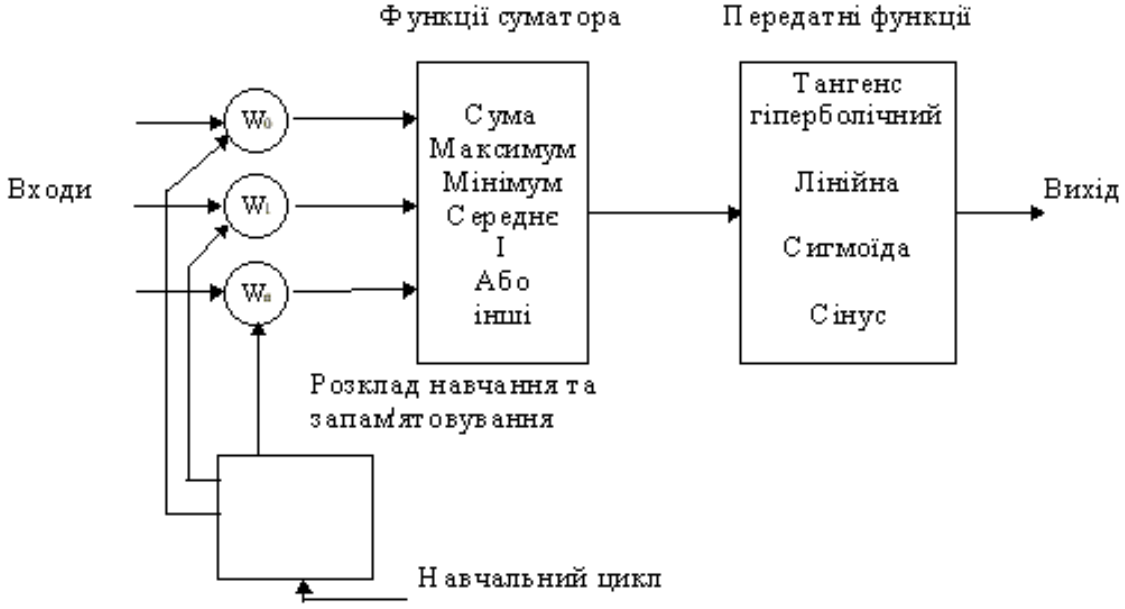


Рис. 3. Модель "елементу обробки"

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.

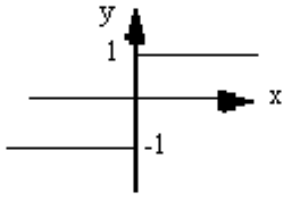
В деяких нейромережах суматор виконує додаткову обробку, так звану функцію активації, яка зміщує вихід функції суматора в часі. Цю функцію найкраще використовувати як компоненту мережі в цілому, ніж як компоненту окремого нейрона. Часто, ця функція є відсутньою.

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.

Переважно 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 "Перцептрони".

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

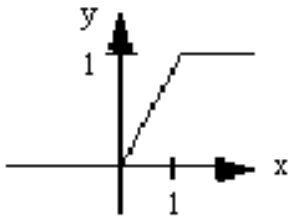
Жорстка порогова функція



x<0 y=-1
x>0 y= 1

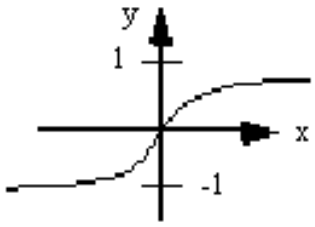
Для простої передатної функції нейромережа може видавати 0 чи 1, 1 чи -1 або інші числові комбінації. Передатна функція в таких випадках є пороговою або «жорстким обмежувачем» (рис. 4а).

Лінійна з насиченням



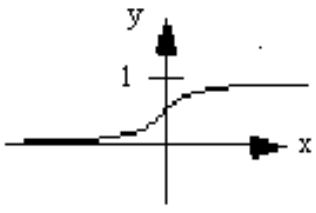
x<0 y=0
0<x<1 y=x
x>1 y=1

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).



x>0 y=1-1/(1+x)
x<0 y=-1+1/(1-x)

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.



$y=1/(1+e^{-x})$

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:

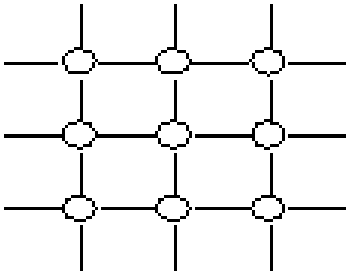
Структура нейромережі - спосіб зв'язків нейронів у нейромережі.

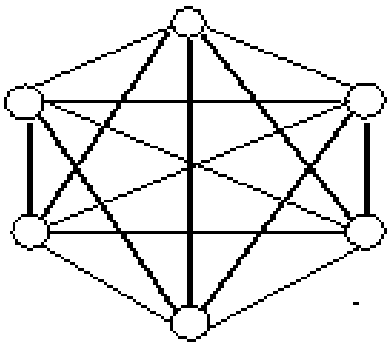
Архітектура нейромережі - структура нейромережі та типи нейронів.

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.





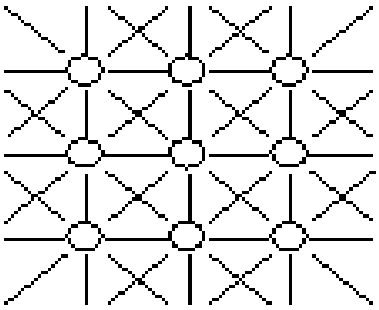
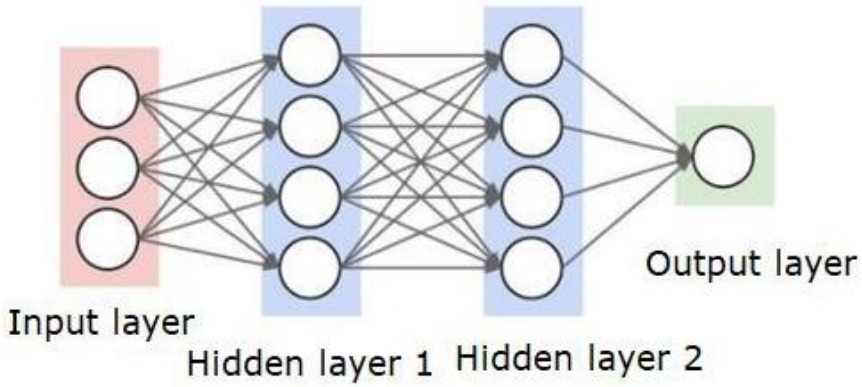


Рис. 5а. Слабозв'язані нейромережі

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.



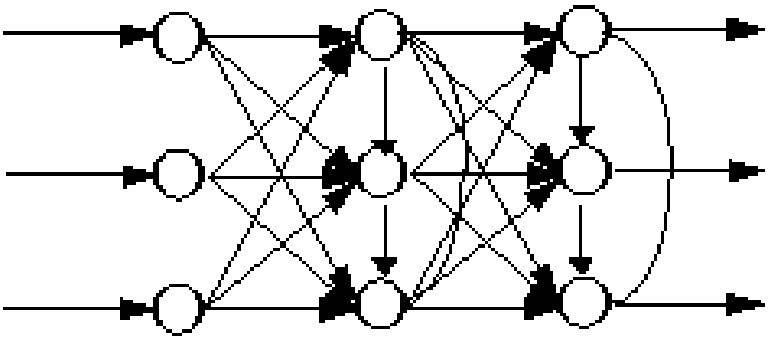


Рис. 6. Багатошаровий тип з'єднання нейронів

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).

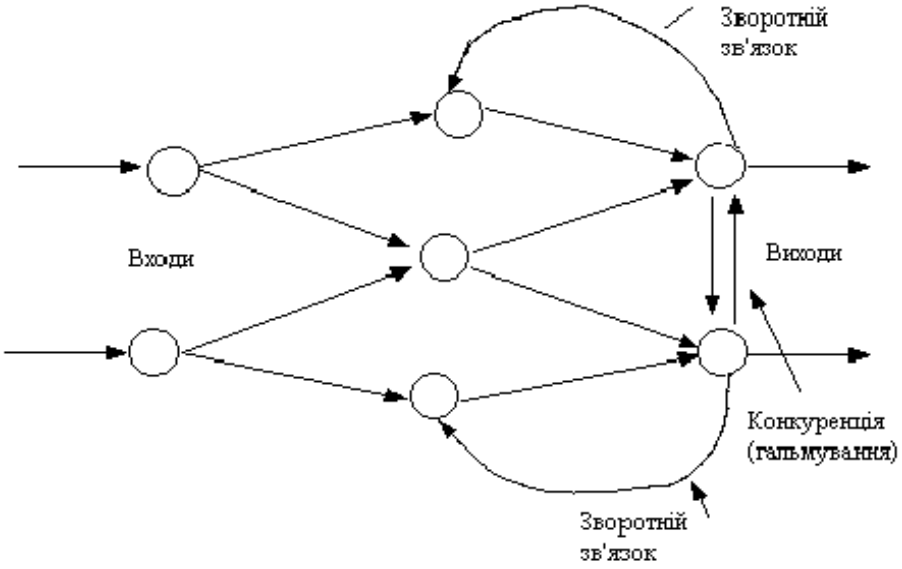
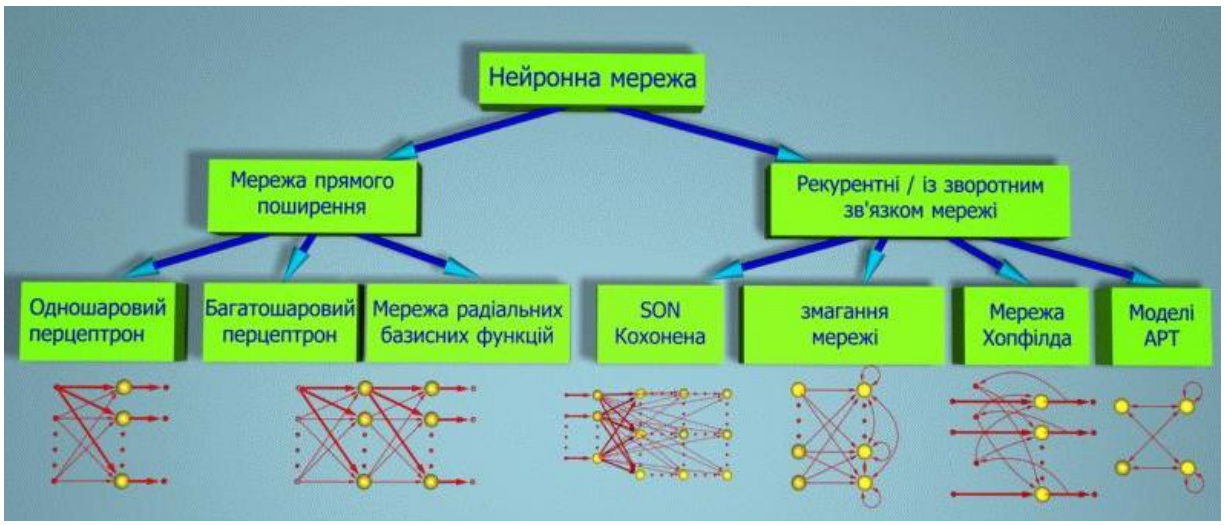


Рис.7

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 односкерованими послідовними зв'язками).
Мережі зворотного поширення (з рекурентними зв'язками).

Типові архітектури нейронних мереж

Мережі прямого поширення	**Рекурентні мережі**
Перцептрони	Мережа Хопфілда
Мережа Back Propagation	Мережа Хемінга
Мережа зустрічного поширення	Мережа адаптивної резонансної теорії
Карта Кохонена	Двоскерована 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 мінімізації 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 повинен 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 противному випадку

мережа просто «запам'ятає» усю навчальну вибірку і не зможе виконати узагальнення.

Підготовка і нормалізація даних

Вихідні дані перетворяться 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 воєдино;

• *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 перцептрону, 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 нейроної мережі. Цей етап особливо важливий для мереж, які навчаються с 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 перевірочні використовуються для розрахунку похибки мережі (перевірочні дані ніколи для навчання мережі не застосовуються). Таким чином, якщо на перевірочних 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 оверфіттінгом*. 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’язанню.

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 мереж прямого поширення

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.

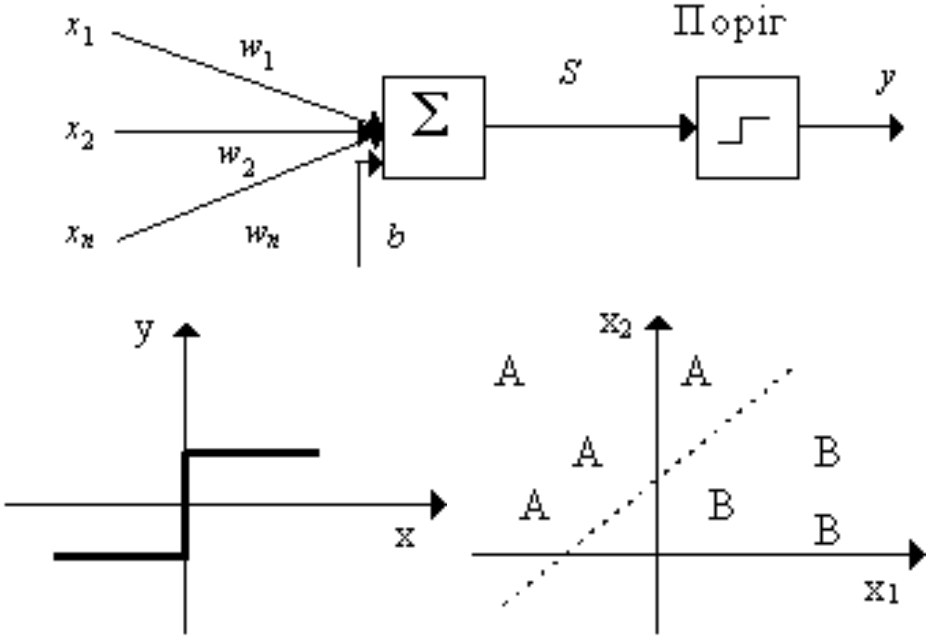


Рис. 5. 1. Схема нейрона, графік передатної функції і поділяюча поверхня

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 $x=(x_1, x_2, ..., x_n)$ 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:

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

$$d(t) = \begin{cases} +1, \text{вихідний клас A} \\ -1, \text{вихідний клас B} \end{cases}$$

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.

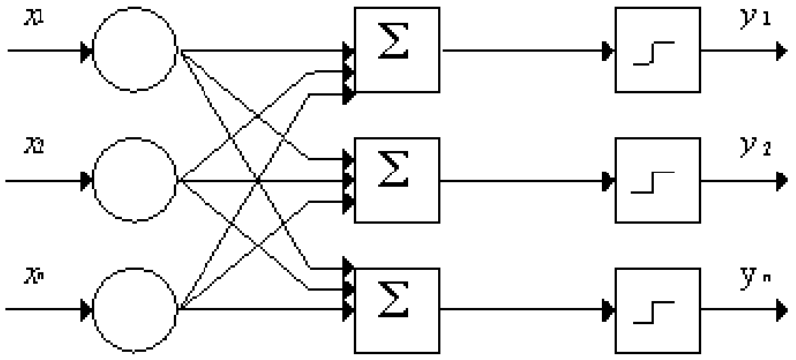


Рис. 5.2. Перцептрон з декількома входами та виходами

Нейромережа зворотного поширення похибки

(Back Propagation)

Архітектура *FeedForward BackPropagation* була розроблена на початку 1970-х років декількома незалежними авторами: Вербор (*Werbor*); 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

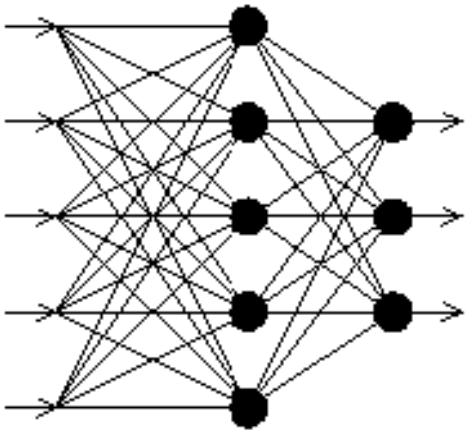


Рис. 5.3 Схема багатошарового перцептрона

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 $n_{прих}$ обирається емпіричним шляхом. В більшості випадків достатньою кількість нейронів буде $n_{прих} <= n_{вх} + n_{вих}$, де $n_{вх}$, $n_{вих}$ - 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 розділятись 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 просунутись 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 крутизні 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... x_N), повинні розрізнятися для всіх прикладів навчальної множини.

3. Обчислення вихідного сигналу:

$$S_{i_m} = \sum_{i_{m-1}}^{N_{m-1}} w_{i_m j_{m-1}} y_{i_{m-1}} - b_{i_m}$$

$$y_{im} = f(S_{jm})$$

$$i_m = 1, 2, ..., N_m,\ m = 1, 2, ..., L$$

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.

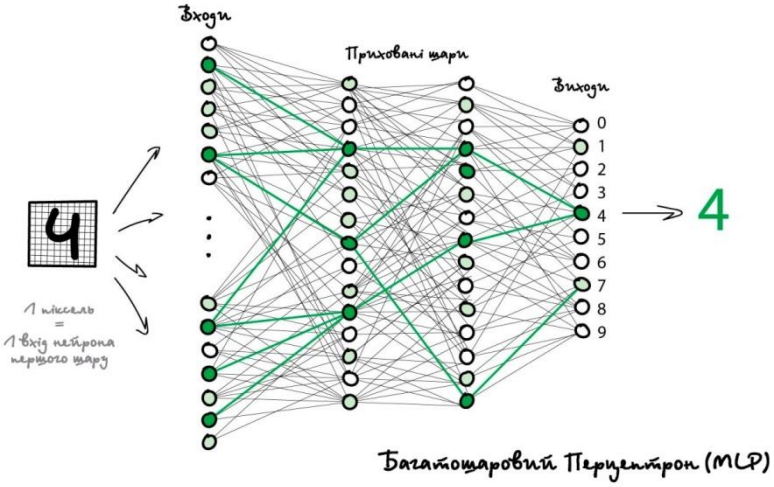
Недоліки. Низька швидкість навчання.

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.

ПРИКЛАД

Якщо створити достатню кількість шарів і правильно розставити ваги в такій мережі, виходить наступне - подавши на вхід, скажімо, зображення написаної від руки цифри 4, чорні пікселі активують пов'язані з ними нейрони, ті активують наступні шари, і так далі і далі, поки в результаті не висвітлиться той самий вихід, який відповідає за четвірку. Результат досягнуто.



У реальному програмуванні, очевидно, жодних нейронів і зв'язків не пишуть, все зображають матрицями, тому що потрібна швидкість.

Така мережа, де є декілька шарів і між ними пов'язані всі нейрони, називається перцептроном (MLP) і вважається найпростішою архітектурою для новачків. У бойових задачах особисто я ніколи її не зустрічав.

Коли ми побудували мережу, наша задача так правильно розставити ваги, щоб нейрони реагували на потрібні сигнали. Тут потрібно згадати, що у нас же є дані - приклади «входів» і правильних «виходів». Будемо показувати нейромережі малюнок тієї ж цифри 4 і говорити «побудуй свої ваги так, щоб при такому вході на твоєму виході завжди спалахувала четвірка».

Спочатку всі ваги просто розставлені випадково, ми показуємо мережі цифру, вона видає якусь випадкову відповідь (ваг же немає), а ми порівнюємо, наскільки результат відрізняється від потрібного нам. Потім йдемо по мережі в зворотному напрямку, від виходів до входів, і говоримо кожному нейрону - так, ти ось тут

чогось активувався, через тебе все пішло не так, давай ти будеш трохи менше реагувати на ось цей зв'язок і трохи більше на отой, ок?

Через тисяч сто таких циклів «прогнали-перевірили-покарали» є надія, що ваги в мережі відкоригуються так, як би ми хотіли. Науково цей підхід називається Backpropagation або «Метод зворотного поширення помилки». Цікаво те, що щоб відкрити цей метод знадобилося двадцять років. До нього нейромережі навчали як могли.

Мережа Delta Bar Delta

Мережа Delta bar Delta була розроблена Робертом Джекобсом (Robert Jacobs), для поліпшення оцінки навчання стандартних мереж Feed Forward і є модифікацією мережі Back Propagation.

Процедура Back Propagation базується на підході крутого спуску, що мінімізує похибку мережі під час процесу зміни синаптичних ваг. Стандартні оцінки навчання застосовуються на базисі "шар за шаром" і значення моменту призначаються глобально. Моментом вважається фактор, що використовується для згладжування оцінки навчання. Момент додається до стандартної зміни ваги і пропорційний до попередньої зміни ваги.

Хоча цей метод успішний у розв’язанні багатьох задач, збіжність процедури занадто повільна для використання. *Delta bar Delta має "неформальний" підхід до навчання штучних мереж, при якому кожна вага має свій власний самоадаптований фактор навчання і минулі значення похибки використовуються для обчислення майбутніх значень.* Знання ймовірних похибок дозволяє мережі робити інтелектуальні кроки при зміні ваг, але процес ускладнюється тим, що кожна вага може мати зовсім різний вплив на загальну похибку. Джекобс запропонував поняття "здорового глузду", коли кожна вага з'єднання мережі повинна мати власну оцінку навчання, а розмір кроку, що призначений одній вазі з'єднання не застосовується для усіх ваг у шарі.

Оцінка навчання будь-якої ваги з'єднання змінюється на основі інформації про поточну похибку, знайденої зі стандартної Backpropagation. Якщо локальна похибка має однаковий знак для декількох послідовних часових кроків, оцінка навчання для цього з'єднання лінійно збільшується. Якщо локальна похибка часто змінює знак, оцінка навчання зменшується геометрично і це гарантує, що оцінки навчання з'єднання будуть завжди додатними. Оцінкам навчання дозволено мінятися в часі.

Призначення оцінки навчання до кожного з'єднання, дозвіл цій оцінці навчання неперервно мінятися з часом обумовлюють зменшення часу збіжності.

З розрішенням різних оцінок навчання для будь-якої ваги з'єднання в мережі, пошук крутого спуску може не виконуватися. Замість цього, ваги з'єднань змінюються на основі часткових похідних похибок щодо самої ваги й оцінки "кривизни поверхні похибки" поблизу поточної точки. Зміни ваг відповідають обмеженню місцевості і вимагають інформацію від нейронів, з якими вони з'єднані.

Переваги. Парадигма Delta Bar Delta є спробою прискорити процес збіжності алгоритму зворотного поширення за рахунок використання додаткової інформації про зміну параметрів і ваг під час навчання.

Недоліки

• Навіть невелике лінійне збільшення коефіцієнта може привести до значного росту швидкості навчання, що викликає стрибки в просторі ваг.

• Геометричне зменшення коефіцієнта іноді буває недостатньо швидким.

Мережа спрямованого випадкового пошуку

Мережа спрямованого випадкового пошуку (Directed Random Search), використовує стандартну архітектуру FeedForward, що не базується на алгоритмі BackProragation і коректує ваги випадковим чином. Для забезпечення порядку в такому процесі, до випадкового кроку додається компонент напрямку, що гарантує напрямок ваг до попередньо успішного напрямку пошуку. Вплив на нейрони здійснюється окремо. Для збереження ваг усередині компактної області, де алгоритм працює добре, установлюють верхню границю величини ваг. Установлюючи границі ваг великими, мережа може продовжувати працювати, оскільки дійсний глобальний оптимум залишається невідомим.

Іншою особливістю правила навчання є початкова відмінність у випадковому розподілі ваг. У більшості комерційних пакетів існує рекомендоване розроблювачем число для параметра початкової відмінності. *Парадигма випадкового пошуку має кілька важливих рис*. Вона швидка і легка у використанні, найкращі результати виходять, коли початкові ваги знаходяться близько до найкращих ваг. Швидкою парадигма є завдяки тому, що для проміжних нейронів похибки не обчислюються, а обчислюється лише вихідна похибка. Алгоритм дає ефект лише в невеликій мережі, оскільки при збільшенні числа з'єднань, процес навчання стає довгим і важким. Існує *чотири ключових компоненти мережі* з випадковим пошуком. Це - **випадковий крок, крок реверсування, спрямований компонент і самокоригувальна відмінність**.

Випадковий крок. До будь-якої ваги додається випадкова величина. Уся навчальна множина пропускається через мережу, створюючи "похибку пророкування". Якщо нова загальна похибка навчальної множини менша ніж попередня найкраща похибка пророкування, поточні значення ваг, що включають випадковий крок, стають новою множиною "найкращих" ваг.

Поточна похибка пророкування зберігається як нова, найкраща похибка пророкування.

Крок реверсування. Якщо результати випадкового кроку гірші попередньої найкращої похибки, випадкова величина віднімається від початкового значення ваги. Це створює множину ваг, що знаходяться в протилежному напрямку до попереднього випадкового кроку. Якщо загальна "похибка пророкування" менше попередньої найкращої похибки, поточне значення ваг і поточна похибка пророкування зберігаються як найкращі. Якщо і прямий і зворотний кроки не поліпшують результат, до найкращих ваг додається цілком нова множина випадкових значень і процес починається спочатку.

Спрямований компонент. Для збіжності мережі створюється множина спрямованих компонентів, отриманих за результатами прямого і зворотного кроків. Спрямовані компоненти, що відображають ланцюг успіхів або невдач попередніх випадкових кроків, додаються до випадкових компонентів на кожнім кроці процедури і забезпечують елемент "здорового глузду" у пошуку.

Доведено, що додавання спрямованих компонентів забезпечує різке підвищення ефективності алгоритму.

Самокоригувальна відмінність. Визначається параметр початкової відмінності для керування початковим розміром випадкових кроків, що додається до ваг. Адаптивний механізм змінює параметр відмінності, що базується на поточній оцінці успіху або невдачі. Правило навчання припускає, що поточний розмір кроків для ваг у правильному напрямку збільшується для випадку декількох послідовних успіхів. Навпаки, якщо відбувається кілька послідовних невдач, відмінність зменшується для зменшення розміру кроку.

Переваги. Для невеликих і середніх нейромереж, спрямований випадковий пошук дає гарні результати за короткий час. Навчання автоматичне, вимагає невеликої взаємодії з користувачем.

Недоліки. Кількість ваг з'єднань накладає практичні обмеження на розмір задачі. Якщо мережа має більше чим 200 ваг з'єднань, спрямований випадковий пошук може вимагати збільшення часу навчання, але продукувати прийнятні рішення.