# Тема 10. Класифікація та типи нейронних мереж

Нейронні мережі можна класифікувати за наступними показниками:

### За характером навчання:

* **Нейронні мережі, що використовують навчання з вчителем.** Навчання з вчителем передбачає наявність навчальної множини, що містить правильні приклади (пари «вхід» - «вихід»). Мережа навчається на деякій множині таких навчальних пар: пред'являються входи, обчислюється вихід мережі і порівнюється з відповідним правильним виходом. Для мінімізації похибки ваги змінюються відповідно до алгоритму навчання. Приклади навчальної множини пред'являються послідовно, обчислюються похибки і вагові коефіцієнти міжнейронних зв’язків корегуються для всіх прикладів використаної навчальної множини. Такий період називається епохою. Мережа ітеративно повторює кроки навчання для наступних епох. Вагові коефіцієнти коригуються доки похибка по всій навчальній множині не досягне прийнятного рівня. Після цього вагові коефіцієнти фіксуються і мережа готова до використання.
* **Нейронні мережі, що використовують навчання без вчителя.** Навчання без вчителя використовує навчальну множину, що складається лише з вхідних даних (вхідних векторів). Навчальний алгоритм підлаштовує вагові коефіцієнти мережі так, щоб пред'явлення достатньо близьких вхідних векторів надавало однакові виходи. Процес навчання виявляє статистичні властивості навчальної множини і групує подібні вхідні вектори у класи чи кластери.
* **Нейронні мережі, що використовують навчання з підкріпленням.** Навчання з підкріпленням - один із способів машинного навчання, в ході якого система навчається, взаємодіючи з певним середовищем. Відгуком середовища на прийняті рішення є сигнали підкріплення, тому, таке навчання є окремим випадком навчання з вчителем, але вчителем є середовище або його модель.

### За налаштуванням ваг

* **Мережі з фіксованими зв'язками** - вагові коефіцієнти нейронної мережі встановлюються певними значеннями відразу, виходячи з умов задачі.
* **Мережі з динамічними зв'язками** - для них налаштування вагових коефіцієнтів відбувається в процесі навчання.

### За типом вхідної інформації

* **Аналогова** - вхідну інформацію представлено в формі дійсних чисел.
* **Двійкова** - вся вхідна інформація в таких мережах представляється у вигляді нулів і одиниць.

### Тип нейронної мережі

* **Мережі прямого поширення** - всі зв'язки і потік обробки скеровані від вхідних нейронів до вихідних. До таких мереж відносяться, наприклад, багатошаровий перцептрон. Увага! В зворотному напрямку поширюється лише похибка для налаштування прямих зв’язків.
* **Мережі зворотного поширення (рекурентні)** - сигнал з вихідних нейронів або нейронів прихованого прошарку частково передається назад на входи нейронів попереднього прошарку. Цей сигнал буде поширюватися ітеративно, доки не налаштуються певним чином вагові коефіцієнти.

### Історія розвитку моделей нейронних мереж:

* **Передумови (1940-1950-ті).** Ідея штучних нейронних мереж виникла у 1940-х роках. Дослідники Уоррен Маккаллок та Уолтер Піттс розробили математичну модель штучного нейрона. Саме тоді створено першу модель нейронної мережі, але обчислювальні ресурси на той час були обмеженими.
* **Перша золота ера (1950-1960-ті).** В цей час активно досліджуються можливості нейронних мереж, з'являються перші нейронні мережі зі зворотними зв'язками. Однак виявлені обмежені можливості нейромереж та недостатні обчислювальні можливості пригальмували цей розвиток.
* **Перша зима (1970-1980-ті)**. У цей період інтерес до нейронних мереж згасає, дослідження зосереджувалися на інших (класичних) методах машинного навчання.
* **Друга золота ера (1980-1990-ті)**. У цей період завдяки збільшенню обчислювальної потужності та розробці нових алгоритмів навчання нейронні мережі знову стали популярними. У 1986 році представлено алгоритм зворотного поширення похибки. Дослідники розробляють різні алгоритми навчання багатошарових перцептронів та інші архітектури, що дозволяють вирішувати складні завдання.
* **Друга зима (1990-2000-ті)**. Повторне згасання інтересу, дослідження зосереджувалися на методах машинного навчання, що базуються на природніх алгоритмах (генетичні, ройові, мурашині тощо).
* **Сучасність (з 2010-х років)**. З розвитком глибокого навчання (Deep Learning) нейронні мережі досягли неймовірних успіхів. Прориви в галузі глибокого навчання та великі обсяги даних дозволили створювати потужні моделі, такі як згорткові, рекурентні, генеративні нейронні мережі для вирішення складних завдань.

Сучасні нейронні мережі, мають багато параметрів і здатні навчатися на великих обсягах даних. Вони використовуються у різних галузях: комп'ютерний зір, обробка природної мови, рекомендаційні системи, голосові помічники тощо.

Важливо відзначити, що розвиток апаратних обчислювальних засобів та методів навчання продовжує просувати область нейронних мереж вперед, і дослідження у цій галузі активно продовжуються.

## Класичні моделі нейронних мереж

Класичні моделі нейронних мереж розроблено в минулому столітті. Вони заклали основу для розвитку нейромоделювання.

### Перцептрон Розенбалата

Першою моделлю нейромереж вважають перцептрон Розенблата, який є основою для багатьох типів нейромереж і класикою для вивчення. Це одношарова нейронна мережа, що складається з вхідних нейронів, зв'язків з ваговими коефіцієнтами і порогової передатної функції.

Простий перцептрон складається з одного нейрону і здатний розпізнавати найпростіші образи. Нейрон представлено як функцію з багатьма входами і одним виходом. Завданням нейрону є взяти числа зі своїх входів, підсумувати, пропустити через передатну функцію і видати на вихід результат. Простий приклад корисного нейрона: підсумувати всі цифри зі входів, і якщо їх сума більше за N - видати на вихід одиницю, інакше – нуль (рис.10.1).

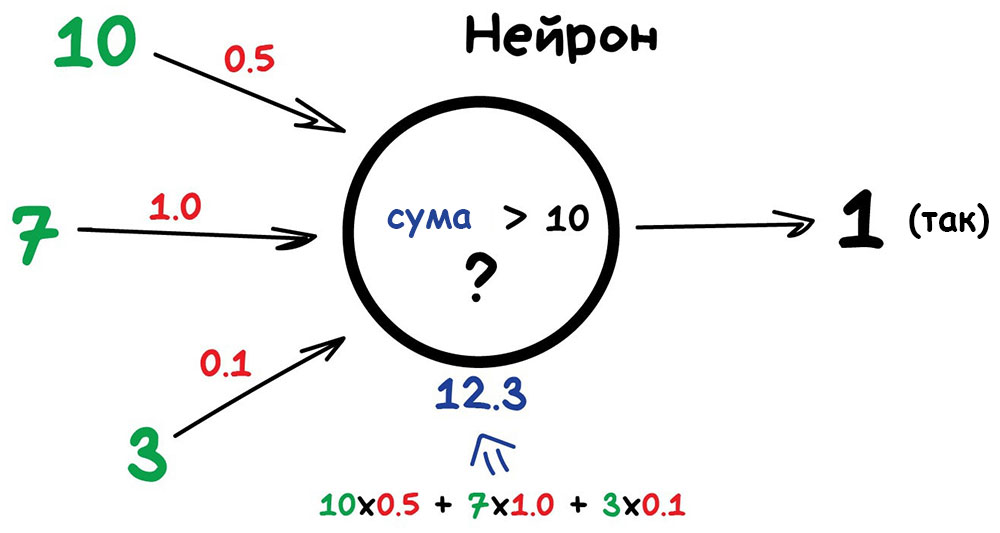


Рис. 10.1. Наочне представлення роботи перцептрону

В кожного входу є власний ваговий коефіцієнт. Якщо через зв'язок з ваговим коефіцієнтом 0.5 проходить число 10, воно перетворюється в 5. Окремий нейрон обчислює суму сигналів вхідних елементів і пропускає результат через жорстку порогову функцію, вихід якої дорівнює 1 чи 0.

В залежності від значення вихідного сигналу приймається рішення:

* 1 - вхідний сигнал належить до класу A,
* 0 - вхідний сигнал належить до класу B.

На рис.10.2. показано схему одношарового перцептрону та графік передатної функції.

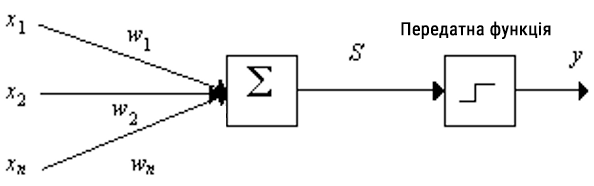


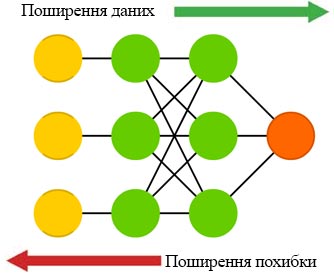
Рис.10.2. Схема одношарового перцептрону

## 10.2. Багатошарові перцептрони

Багатошарові нейронні мережі (Multilayer Perceptrons) складаються з кількох шарів нейронів. Вони здатні вирішувати складніші завдання класифікації та регресії. Прикладом є класичні архітектури прямого поширення.

Нейрони розподілені по прошарках: вхідний, приховані та вихідний. Всередині одного прошарку нейрони не пов'язані, але з'єднані з нейронами наступного і попереднього прошарків. Дані в такій мережі йдуть строго в одному напрямку - від входів першого шару до виходів останнього.

На базі архітектури багатошарового перцептрону реалізовано багато моделей нейронних мереж, що різняться за алгоритмом навчання, структурою зв’язків та функціоналом нейронів.

Класичним прикладом є ефективна парадигма навчання зворотного поширення похибки BackPropagation. Мережа може моделювати функцію практично будь якої складності та використовується у різних типах застосувань.

Типова мережа ВackРropagation має вхідний прошарок, вихідний прошарок та принаймні один прихований прошарок. Теоретично, обмежень відносно кількості прихованих прошарків не існує, але практично застосовують один або два.

Нейрони організовані в пошарову структуру з прямою передачею сигналу. Кожний нейрон мережі продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення.

Важливим при моделюванні мережі є визначення числа проміжних прошарків і числа нейронів в них. Існують загальні правила, зокрема:

1. Кількість входів та виходів мережі визначаються кількістю вхідних та вихідних параметрів навчальної множини. На відміну від зовнішніх прошарків, число нейронів прихованого прошарку nприх обирається емпіричним шляхом. В більшості випадків достатньою кількістю нейронів буде nприх ≈ nвх + nвих, де nвх, nвих - кількість нейронів у вхідному і, відповідно, у вихідному прошарках.
2. Якщо різниця (похибка) між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого прошарку повинна також збільшитись.
3. Якщо процес, що моделюється, може розділятись на багато етапів, потрібен додатковий прихований прошарок (прошарки). Якщо процес не розділяється на етапи, тоді додаткові прошарки можуть ускладнювати обчислення та продукувати невірне загальне рішення.

Після того, як визначено число прошарків і число нейронів в кожному з них, потрібно знайти значення для вагових коефіцієнтів зв’язків, які спроможні мінімізувати похибку спродукованого результату. Саме для цього існують алгоритми навчання, де відбувається підгонка моделі мережі до наявних навчальних даних.

Алгоритм діє ітеративне, його кроки називаються епохами. На кожній епосі на вхід мережі по черзі подаються всі приклади навчальної множини, вихідні значення мережі порівнюються з бажаними значеннями і обчислюються похибки для кожного прикладу, що зрештою перетворюється у загальну похибку для даної ітерації. Значення похибки використовують для корекції вагових коефіцієнтів, і дії повторюються. Процес навчання припиняється або коли пройдена визначена кількість епох, або коли похибка досягає визначеного рівня допустимості, або коли похибка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій припинення навчання).

Для прикладу, як працює мережа, що розпізнає цифри. Даними для мережі є приклади «входів» і правильних «виходів». Нейромережі показують зображення цифри 4 в різних написаннях і змушують її підлаштовувати ваги так, щоб на виході для цих прикладів завжди показувало четвірку (рис.10.3).

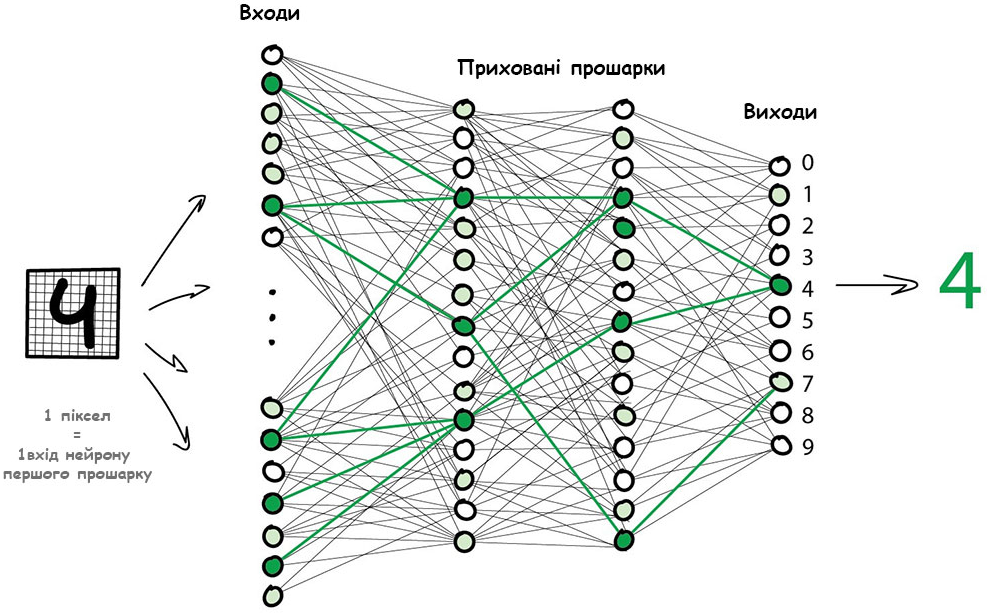


Рис. 10.3. Наочний приклад роботи мережі при розпізнаванні цифр

Спочатку всі ваги розставляються випадково, при пред’явленні цифри 4, вона видає випадкову відповідь (не налаштовані ваги), далі відбувається порівняння, наскільки результат відрізняється від потрібного. Отримане значення похибки поширюється по мережі в зворотному напрямку, від виходів до входів, і корегується вага зв’язків кожного нейрона.

Через значну кількість циклів «прогнали-перевірили-скорегували», вагові коефіцієнти в мережі налаштовуються так, як потрібно.

### Мережа Кохонена

Спеціальний тип нейронної мережі – самоорганізована карта Кохонена (Kohonen's self-organized map), що дозволяє здійснювати кластеризацію об'єктів. Мережа Кохонена складається з двох прошарків - вхідного і вихідного («прошарок Кохонена»). Кожен нейрон вхідного прошарку пов'язаний зі всіма нейронами вихідного (рис.10.4).

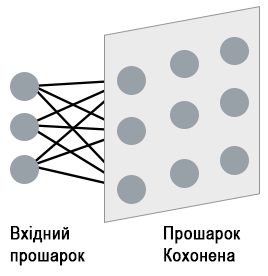


Рис. 10.4. Структура мережі Кохонена

Число вхідних нейронів дорівнює кількості ознак об'єкта. Вхідні нейрони не беруть участі в процесі навчання і обробки даних, а просто розподіляють вхідний сигнал по нейронах вихідного шару. Кількість вихідних нейронів мережі Кохонена дорівнює числу кластерів, яке повинно бути побудовано моделлю, і кожен нейрон асоційований з певним кластером.

Ваги на початку встановлюються випадковим чином. Подається перший приклад і обчислюються виходи. Алгоритм навчання діє за принципом «переможець забирає все», тобто нейрону з максимальним значенням виходу привласнюється одиниця, а всім іншим – 0. Після чого об'єкт відноситься до кластеру, асоційованого з даним нейроном-переможцем.

Основний ітераційний алгоритм Кохонена послідовно проходить ряд епох, на кожній з яких обробляється один приклад з навчальної вибірки. Після пред'явлення достатнього числа прикладів мережа підлаштовує вагові коефіцієнти під закономірності у вхідних даних та спроможна розподілити схожі приклади до відповідних кластерів. Вагові коефіцієнти встановлюються так, що кожен нейрон вихідного прошарку реагує на відповідну групу схожих вхідних сигналів (рис.10.5).

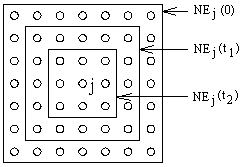


Рис.10.5. Ітераційне визначення нейрона-переможця

В результаті роботи алгоритму центр кластера встановлюється в певній позиції, яка задовольняє схожим прикладам, 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. Спочатку до околу належить велике число нейронів, далі її розмір поступово зменшується. Мережа формує топологічну структуру, в якій на схожі приклади реагують групи нейронів, які близько знаходяться на топологічної карті.

При використанні, навчена мережа Кохонена отримує нові дані і розподіляє їх до відповідних кластерів. Якщо мережа зустрічається з набором даних, несхожим з жодним відомим зразком, вона відносить його до нового кластеру або формується відповідь про не розпізнання даних.

Мережі Кохонена можна використовувати і в задачах, де класи є відомими. Якщо в даних містяться мітки класів, то мережа спроможна вирішувати задачі класифікації. - перевага буде у спроможності мережі виявляти подібність між різноманітними класами.

## Мережа Хопфілда

Нейронна мережа Хопфілда (Hopfield Network) - це повнозв’язна нейронна мережа, що вирішує завдання асоціативної пам’яті. Деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують якийсь об'єкти або характеристики процесів), вважають зразковим. Мережа повинна вміти з зашумленого сигналу, поданого на її вхід, виділити ("пригадати" по частковій інформації) відповідний зразок або "дати висновок" про те, що вхідні дані не відповідають жодному із зразків.

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. 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 біжучий стан сигналів нейронів прошарку передається на вихідний прошарок. Цей стан відповідає образу, який буде запам’ятовано в мережі.

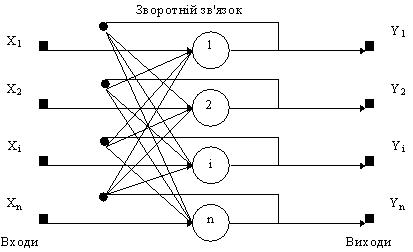


Рис. 10.6. Структура мережі Хопфілда

Навчання мережі Хопфілда вимагає, щоб навчальний образ було представлено 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 асоціативна пам'ять, вона має два головних обмеження.

1. 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 ємності пам'яті для мережі приблизно 15% від числа нейронів у прошарку Хопфілда.
2. Якщо навчальні приклади є занадто подібними, прошарок Хопфілда може стати нестабільним. Зразок образу вважається нестабільним, якщо мережа збігається до деякого іншого образу з навчальної множини. Ця проблема може бути вирішена вибором навчальних прикладів, що достатньо відрізняються між собою.

## Популярні сучасні нейронні мережі

Детальне пояснення роботи сучасних нейронних мереж можна подивитися у відео курсі «Нейронні мережі на Python» - <https://www.youtube.com/playlist?list=PLA0M1Bcd0w8yv0XGiF1wjerjSZVSrYbjh>

### Автокодувальник

Автокодувальник (Autoencoder) - спеціальна архітектура штучних нейронних мереж, що дозволяє застосовувати навчання без вчителя при використанні методу зворотного поширення похибки. Основною ідеєю є автоматичне стиснення (кодування) інформації. Використовується для зменшення розмірності даних чи вивчення внутрішнього складу даних. Автокодувальник складається з двох основних компонентів: кодувальника (Encoder) та декодувальника (Decoder).

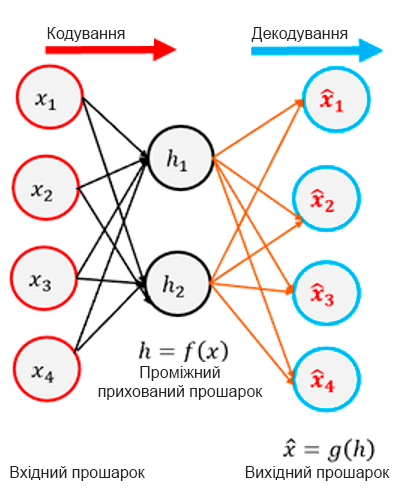


Рис.10.7. Структура мережі Автокодувальник

Кодувальник (Encoder) приймає вхідні дані і перетворює їх на передставлення з меншою розмірністю - кодоване подання. Кодувальник зазвичай складається з шарів нейронів, які зменшують розмірність даних у міру їх проходження через мережу. Зрештою кодувальник створює компактне представлення вихідних даних.

Декодувальник (Decoder) приймає кодоване представлення і відновлює з нього дані. Як і кодувальник, декодувальник складається з шарів нейронів, які поступово збільшують розмір даних до вхідного розміру. Мета декодувальника – отримати на виході дані, максимально схожі на вхідні.

Для навчання автокодувальника використовується функція втрат, яка вимірює різницю між вхідними даними та даними, отриманими на виході декодувальника. Популярною функцією втрат для автоенкодерів є середньоквадратична помилка, яку потрібно мінімізувати. У процесі навчання автокодувальник виявляє найважливіші ознаки даних у кодованому поданні та .навчається максимально точно відновлювати вхідні дані на виході.

Після навчання автокодувальник може використовуватися для вирішення різних завдань: зменшення розмірності даних, витягування ключових ознак, виявлення нестандартних зразків. Автокодувальники є важливими компонентами у галузі глибокого навчання та машинного навчання.

### Глибинна мережа

Глибинна мережа (Deep Belief Networks) - це тип архітектури, в якій мережа складається з кількох з'єднаних автокодувальників. Такі мережі навчаються по блоках, причому кожному блоку потрібно вміти закодувати попередній (рис.10.10).

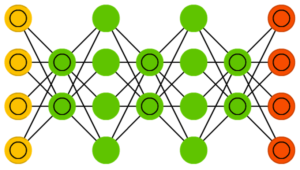


Рис.10.10. Архітектура глибинної мережі

Для попереднього навчання глибоких мереж застосовують каскадну структуру автокодувальників. Прошарки мережі навчаються послідовно, починаючи з перших. До кожного нового прошарку на час навчання надається додатковий вихідний шар, що доповнює мережу до автокодувальника, після чого на вхід мережі подаються набір даних для навчання. Ваги ненавченого та додаткового прошарків налаштовуються за методом зворотного поширення похибки.

Далі прошарок автокодувальника відключається і створюється новий, що відповідає наступному ненавченому прошарку мережі. На вхід мережі знов подається той же набір даних, навчені перші прошарки мережі залишаються без змін і працюють як вхідні для наступного прошарку автокодувальника. Навчання продовжується для всіх прошарків за винятком останніх. Останні прошарки мережі навчаються без використання авто кодувальника за методом зворотного поширення похибки та на відомих даних (навчання з учителем).

Така техніка називається "жадібним навчанням", яка полягає у виборі локальних оптимальних рішень, що не гарантують оптимальний кінцевий результат. Також мережу можна навчити (методом зворотного поширення помилки) відображати дані у вигляді ймовірнісної моделі. Якщо використовувати навчання без вчителя, модель можна використовувати для генерації нових даних.

### Згорткові нейромережі

Згорткові нейронні мережі (Convolutional Neural Networks) сильно відрізняються від інших видів мереж (рис.10.11). Вони використовуються для пошуку об'єктів на фото і відео, розпізнавання осіб, перенесення стилю, генерації і домальовування зображень, створення ефектів і покращення якості фотографій.

Детальну інформацію можна подивитися у відео уроці «Як працюють згорткові нейронні мережі» <https://www.youtube.com/watch?v=CEUNTRdLhKk>

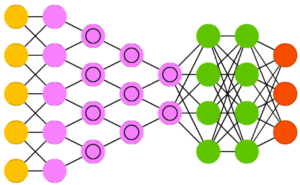


Рис.10.11. Архітектура згорткової мережі

Проблема з зображеннями завжди була в тому, що незрозуміло, як виділяти на них ознаки. Текст можна розділити за реченнями, взяти властивості слів зі словників. Картинки же доводилося розмічати власноруч, пояснюючи машині, де у котика на фотографії вуха, а де хвіст (рис.10.12).

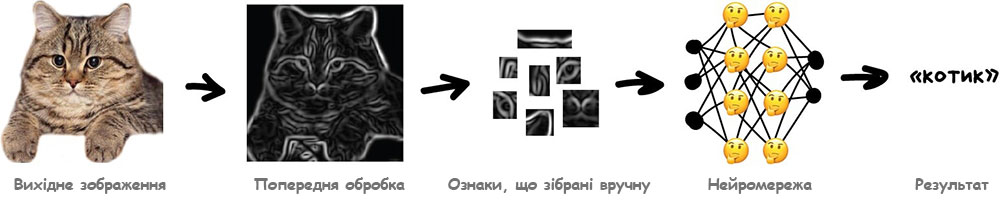


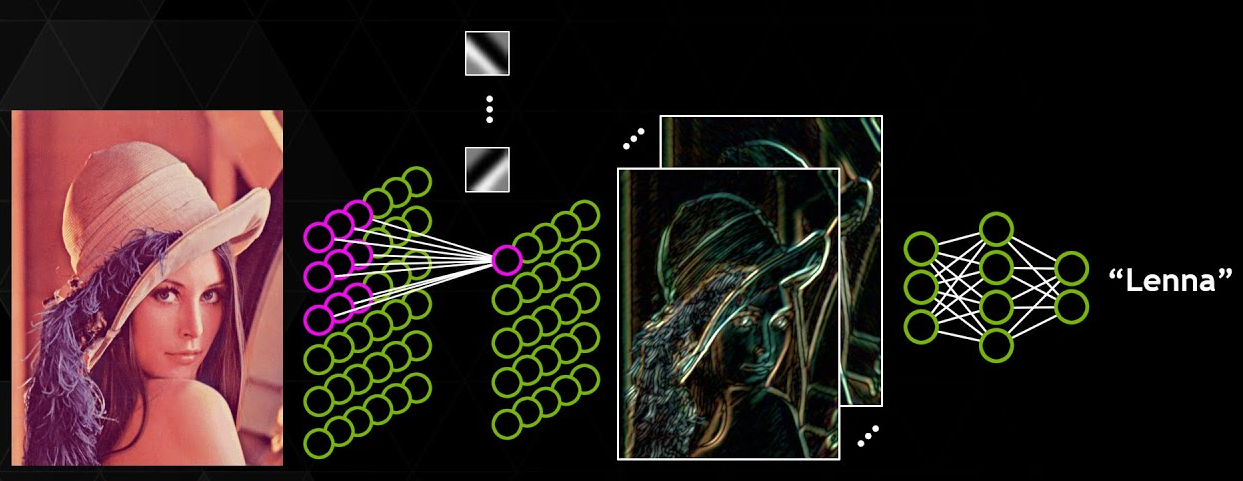
Рис.10.12. Процес навчання згорткової мережі

Проблем у ручного крафтінга багато.

1. По-перше, якщо котик на фотографії притиснув вушка або відвернувся - нейромережа нічого не побачить.
2. По-друге, важко сформувати хоча б десять характерних ознак, що відрізняють котиків від інших тварин. Однак людина, навіть краєм ока може розрізнити хто є котиком, а хто собакою. Людина не дивиться тільки на форму вух і кількість лап - вона оцінює об'єкт за множиною різних ознак, про які навіть не замислюється. А значить, не розуміє і не може пояснити машині.

Згорткова мережа - це прототип зорової кори мозку. Зорова кора має невеликі ділянки клітин, які чутливі до певних ділянок поля зору. Окремі мозкові нервові клітини реагують (або активуються) тільки при візуальному сприйнятті ліній певної орієнтації. Наприклад, деякі нейрони активуються, коли сприймають вертикальні лінії, а деякі - горизонтальні або діагональні. Ці нейрони зосереджені в вигляді стрижневої архітектури і разом формують візуальне сприйняття. Цю ідею спеціалізованих компонентів всередині системи, які вирішують конкретні завдання (як клітини зорової кори, які шукають специфічні характеристики) і використовують згорткові нейронні мережі.

Уявіть, що зображення - це аркуш паперу з рисунком. Згорткові нейронні мережі працюють, ніби накласти фільтр поверх аркушу і ковзати по зображенню, одночасно перевіряючи, що знаходиться під фільтром. Фільтр може бути налаштований на виявлення певних патернів, таких як грані, кути чи текстури.

Рис. 10.14. Робота фільтра в згортковій мережі

Мережі потрібно самій вчитися шукати ці ознаки, що складаються з базових ліній. Підхід буде таким: для початку зображення розділяється на блоки n x n пікселів і вибирається, яка лінія домінує в кожному - горизонтальна [-], вертикальна [|] або одна з діагональних [\] [/].

На виході отримується новий масив ліній, які є простими ознаками наявного образу об'єкта на зображенні. Для цього масиву знов вибираються блоки n x n і з’ясовується, як ці лінії поєднуються одна з одною. І таким чином на кожному кроці відбувається згортка масивів. Згортку можна уявити як прошарок нейромережі.

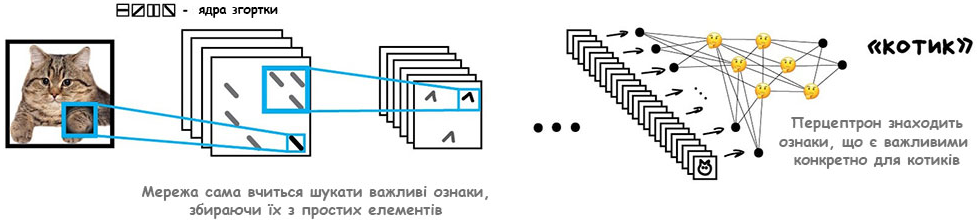


Рис.10.13. Схема функціонування згорткової мережі

Коли через нейромережу проходить багато фотографій котів, вона автоматично розставляє великі ваги тим сполученням з ліній, які побачила найчастіше. Причому неважливо, це пряма лінія спини або складний геометричний об'єкт типу мордочки - щось обов'язково буде яскраво активуватися.

На виході ставиться багатошаровий перцептрон, який визначає, які поєднання активувалися і для кого вони є більш характерними – для котів чи собак.

Згорткова нейромережа знаходить характерні ознаки об'єктів, складає карти ознак і навчається визначати що завгодно. З цього приводу є жарт: «Дай нейромережі рибу - вона зможе визначати рибу до кінця життя. Дай нейромережі вудку - вона зможе визначати і вудку до кінця життя ...»

Такі мережі зазвичай використовують "сканер", який не обробляє всі дані за один раз. Наприклад, якщо є зображення 200 × 200 пікселів, він не буде відразу обробляти всі 40 тисяч пікселів. Замість цього мережа зчитує квадрат розміру 20 x 20 (зазвичай з лівого верхнього кута), потім зсувається на 1 піксель і зчитує новий квадрат, і так далі. Ці вхідні дані далі передаються через згорткові прошарки. Ці шари мають властивість стискуватися з глибиною, причому часто використовуються ступені двійки: 32, 16, 8, 4, 2, 1. Для подальшої обробки дані надходять до багатошарового перцептрону.

Кожен шар нейронної мережі використовує власне перетворення. Якщо на перших шарах мережу оперує такими поняттями як "ребра", "межі" і подібне, то далі використовуються поняття "текстура", "частини об'єктів". В результаті такого опрацювання можна правильно класифікувати картинку або виділити на кінцевому етапі шуканий об'єкт на зображенні.

Згорткова нейронна мережа за рахунок застосування згортки дозволяє зменшити кількість інформації, що зберігається в пам'яті. Вона краще справляється з картинками більш високої роздільної здатності, і виділяє опорні ознаки зображення, такі як ребра, контури або межі. На наступному рівні обробки з цих ребер і граней можна розпізнати повторні фрагменти текстур, які далі можуть скластися в фрагменти зображення.

Згорткові нейромережі зробили революцію в комп'ютерному зорі і розпізнаванні образів. Це основний інструмент для класифікації і розпізнавання об'єктів, осіб на фотографіях, розпізнавання мови. В першу чергу CNN використовують для зображень, але "згортка" - універсальна операція. Її можна застосувати для будь-якого сигналу, будь то дані з давачів, аудіосигнал або картинка.

Є багато варіантів модифікацій згорткових мереж, такі як Deep Convolutional Neural Network (DCNN), Region-CNN (R-CNN), Fully Convolutional Neural Networks (FCNN), Mask R-CNN і інші. Їх використовують для розпізнавання мови, обробки аудіосигналів, обробки часових рядів, для аналізу змісту текстів. На даний момент це найуспішніша модель, найуспішніша інновація в рамках того, що називають глибоким або глибинним навчанням.

### Розгорткові нейромережі

Розгорткові нейронні мережі (Deconvolutional Networks) є зворотними до згорткових нейронних мереж. Наприклад, якщо передати мережі слово "кіт", то вона генерує картинки з котами, що подібні на реальні зображення котів (рис.10.15).

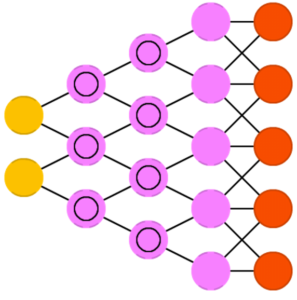


Рис.10.15. Архітектура розгорткової мережі

Варто зауважити, що в більшості випадків мережі передається не рядок, а який бінарний вектор: наприклад, <0, 1> - це кіт, <1, 0> - собака, а <1, 1> - і кіт, і собака.

## 10.8. Рекурентні нейромережі

Рекурентні нейронні мережі (Recurrent Neural Networks) - це вид нейронних мереж, які добре вирішують всі завдання, пов'язані з послідовностями - голосові, текстові або музичні. Вони працюють інакше, ніж згорткові нейронні мережі. Завдяки рекурентним мереж виконується [машинний переклад текстів](https://vas3k.ru/blog/machine_translation/) і комп'ютерний синтез мови. На них

Перші голосові синтезатори смішно вимовляли слова по літерах, намагаючись якось склеїти їх між собою. Сучасні програми не просто вимовляють слова без помилок, вони навіть розставляють акценти в реченні!

Сучасні голосові помічники навчають говорити не буквами, а фразами. Відразу змусити нейромережу цілком видавати фрази не вийде, оскільки тоді їй треба буде запам'ятати всі фрази в мові, цей розмір буде велетенським. Однак текст, мова або музика - це послідовності. Кожне слово або звук - це самостійна одиниця, але яка залежить від попередніх.

Досить легко навчити мережу вимовляти окремі слова або букви. Береться множина аудіо файлів, розмічених на слова і навчається за вхідним словом видавати послідовність сигналів, схожих на його вимову. Здійснюється порівняння з оригіналом від диктора і намагання максимально наблизитися до ідеалу. Для таких дій можна застосувати навіть перцептрон.

Втім, перцептрон не запам'ятовує, що він генерував раніше, для нього кожен запуск як в перший раз. З'явилася ідея додати до кожного нейрона пам'ять. Так були винайдені рекурентні мережі, в яких кожен нейрон запам'ятовував всі свої попередні відповіді і при наступному запуску використовував їх як додатковий вхід. Тобто, нейрон міг сказати самому собі в майбутньому - наступний звук повинен звучати вище, голосна бути довшою (дуже спрощений приклад).

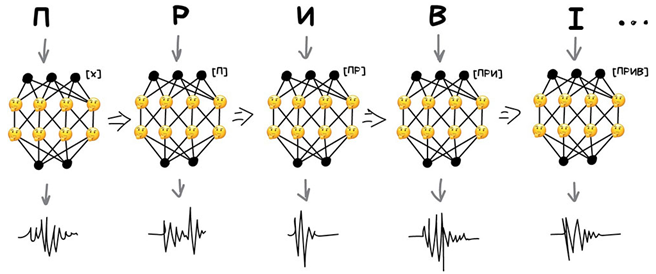


Рис. 10.16. Наочне представлення роботи рекурентної мережі для обробки голосу

Рекурентна мережа працює по черзі з кожним словом речення, і для кожного слова вона робить наступні дії:

* Приймає поточне слово (або символ) як вхідні дані і також враховує інформацію, яку вона бачила на попередніх кроках.
* Мережа має внутрішній прихований стан (Hidden State), який є своєю "пам'яттю" про попередні кроки. Цей прихований стан оновлюється на кожному кроці відповідно до поточного входу та попереднього стану.
* На кожному кроці мережа створює вихід, який може бути використаний для будь-якого завдання. Наприклад, у задачі передбачення наступного слова у реченні вихід може бути ймовірністю кожного слова.

Головна ідея рекурентної мережі полягає в тому, що вона враховує контекст та залежність між елементами послідовності. Наприклад, під час аналізу тексту вона може враховувати попередні слова, щоб зрозуміти поточне слово у його контексті. Це робить рекурентні мережі потужними інструментами обробки послідовних даних.

Була лише одна проблема - коли кожен нейрон запам'ятовував всі минулі результати, в мережі утворювалася надвелика кількість входів, що навчити таку кількість зв'язків стало нереально. Коли нейромережа не вміє забувати - її не можна навчити.

Спочатку проблему вирішили просто – відімкнули для кожного нейрона пам'ять. Але потім придумали як для цієї «пам'яті» використовувати спеціальні комірки, що схожі на пам'ять комп'ютера або регістри процесора. Кожна комірка дозволяла записати в себе числа, прочитати або скинути - їх назвали комірками довгої і короткотермінової пам'яті LSTM (Long Short-Term Memory).

Ці комірки пам'яті захищені від забування за допомогою трьох воріт:

* **Ворота входу.** Визначають, яка інформація із вхідного сигналу буде збережена в комірці пам'яті.
* **Ворота забування.** Визначають, яка інформація з комірки пам'яті буде забута.
* **Ворота виходу.** Визначають, яка інформація з комірки пам'яті буде передана до наступного шару нейронів.

LSTM значно покращили продуктивність рекурентної мережі у завданнях, які потребують тривалого зберігання інформації. Вони використовуються у широкому спектрі додатків, включаючи розпізнавання мови, машинний переклад та обробку природної мови.

Нижче наведено прості аналогії, які можуть допомогти зрозуміти принцип роботи RNN та LSTM:

* **RNN.** Уявіть собі ланцюжок людей, кожен із яких передає інформацію наступному. Кожна людина може обробляти інформацію, що надходить від попередньої людини, та додавати власну інформацію. Це схоже на те, як RNN обробляє вхідну послідовність.
* **LSTM.** Уявіть собі ланцюжок людей, кожен з яких має коробку для зберігання інформації. Кожна людина може додавати інформацію до своєї коробки та отримувати інформацію з коробки попередньої людини. Однак люди можуть також забувати інформацію зі своєї коробки. LSTM використовує три ворота, щоб контролювати, яка інформація буде збережена в коробці, яка інформація буде забута і яка інформація буде передана наступній людині.

Коли нейрону потрібно було поставити собі нагадування на майбутнє - він заносив це в комірку, коли навпаки вся історія ставала непотрібною (наприклад, речення закінчилося) - комірки скидалися, залишаючи тільки «довготермінові» зв'язки. Мережа навчалася не тільки встановлювати поточні зв'язки, а й ставити нагадування.

Озвучені тексти для навчання почали брати звідки завгодно. На прикладі багатьох відеозаписів з виступами Обами непогано навчили нейромережу розмовляти його голосом (<https://www.youtube.com/watch?time_continue=4&v=cQ54GDm1eL0>). На цьому прикладі видно, що імітувати голос - досить просте завдання для сьогоднішніх машин.

## Генеративно змагальна нейронна мережа

## Генеративно змагальна мережа (Generative Adversarial Network) - це клас нейронних мереж, що містить дві моделі: генератор і дискримінатор. Мережі призначені для генерації нових даних, які можуть бути схожими на деякий заданий набір навчальних даних.

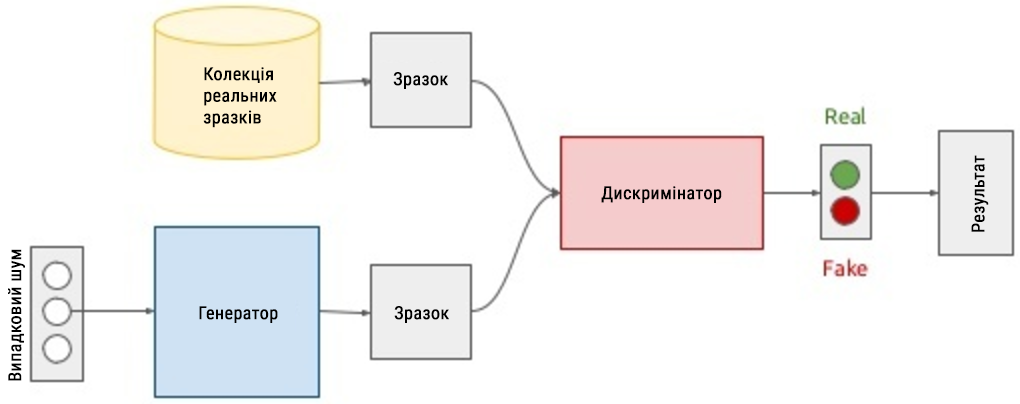
## Робота генеративно змагальної мережі заснована на ідеї змагання між двома моделями: генератором та дискримінатором.

## Генератор приймає на вхід випадковий вектор шуму і генерує з нього нові зразки даних. Наприклад, якщо мережа навчається на зображеннях обличь, генератор може приймати на вхід випадковий вектор і генерувати зображення, яке виглядає як обличчя. Спочатку генератор створює випадкові зразки з невисокою якістю.

## Дискримінатор приймає на вхід зразки даних: справжні (з навчального набору) або згенеровані генератором, і намагається відрізнити справжні зразки від згенерованих. Він видає ймовірність того, що цей зразок є реальним.

## Навчання мережі відбувається через ітеративний процес, в якому генератор і дискримінатор конкурують один з одним. Генератор намагається виробляти зразки, які дискримінатор зможе відрізнити від реальних, а дискримінатор намагається стати точнішим у класифікації зразків. Навчання проводиться шляхом мінімізації функції втрат обох моделей.

## У міру навчання генератор та дискримінатор стають дедалі досвідченішими. Ціль полягає в тому, щоб досягти рівноваги, коли генератор створює зразки, які дискримінатор не може відрізнити від реальних даних.



Можна навести аналогію, щоб зрозуміти принцип роботи GAN:

* **Генератор.** Це художник, що намагається намалювати картину, яка схожа на картину з музею.
* **Дискримінатор.** Це експерт з мистецтва, який намагається відрізнити справжню картину від підробки.

У цьому випадку GAN – це змагання між художником та експертом. Художник намагається створити картину, яка обдурить експерта, а експерт намагається відрізнити справжню картину від підробки.

## Головна ідея GAN полягає в тому, щоб навчити генератор створювати зразки, які стають все більш реалістичними та невідмінними від вихідних даних.

## Генеративно змагальні мережі – це складний та ефективний метод машинного навчання. Вони мають потенціал революціонізувати багато галузей, включаючи комп'ютерний зір, обробку природної мови та штучний інтелект.

## Висновок

Нейронні мережі є потужним інструментом машинного навчання, який має здатність виявляти складні закономірності в даних і приймати рішення на основі цих закономірностей. Вони показали вражаючі результати у багатьох сферах і мають високий потенціал для подальшого розвитку.

* Нейронні мережі ефективно працюють із високорозмірними та складними даними, такими як зображення, звук, текст та відео. Вони можуть автоматично розпізнавати ознаки з цих даних та використовувати їх для класифікації, розпізнавання образів, синтезу та інших завдань.
* Глибокі нейронні мережі, такі як згорткові та рекурентні нейронні мережі, дозволяють створювати моделі з великою кількістю шарів, що дозволяє їм навчатися на більш абстрактних і складних рівнях. Глибоке навчання дозволяє досягти високої точності у різних завданнях, таких як комп'ютерний зір, обробка природної мови та мови.
* Нейронні мережі можуть автоматично налаштовувати свої параметри на основі навчальних даних. Це дозволяє автоматизувати процес аналізу даних та прийняття рішень, що може заощадити час та зусилля.
* Нейронні мережі також можуть застосовуватися для оптимізації різних процесів, наприклад, в галузі управління та оптимізації ресурсів.
* Нейронні мережі мають здатність навчатися на різних типах даних і пристосовуватися до нових ситуацій. Вони здатні виявляти приховані залежності та адаптуватися до змін у даних. Це робить їх корисними для вирішення завдань, де потрібна обробка динамічних та неструктурованих даних.

### Перспективи розвитку нейронних мереж

Перспективи розвитку нейронних мереж дуже широкі. У міру розвитку технологій нейронних мереж вони стануть більш ефективними та доступними. Це призведе до їх широкого поширення у різних галузях.

* Розвиток ефективних і спеціалізованих архітектур нейронних мереж для конкретних завдань.
* Створення методів, що дозволяють нейронним мережам вчитися з мінімальною участю людини.
* Проникнення нейронних мереж у нові області, такі як сільське господарство, екологія, робототехніка та інші.
* Розвиток методів, здатних забезпечувати надійне узагальнення навчених нейронних мереж на різні дані та умови.
* Розвиток потужніших та енергоефективних чіпів та пристроїв для обробки нейронних мереж.

З врахуванням швидкого розвитку технологій та активного дослідження в цій галузі, нейронні мережі очікують подальшого зростання в ефективності та розширення своїх можливостей.

## Контрольні питання

1. Провести класифікацію нейронних мереж за типом навчання. Навести приклади.
2. Назвати базові типи неромереж за напрямком потоку обробки даних. В чому полягає різниця?
3. Пояснити принцип роботи простого перцептрону, які завдання він спроможний вирішити.
4. Назвати особливість алгоритму навчання Back Propagation.
5. Назвати загальні правила формування структури нейромережі Back Propagation щодо кількості шарів та нейронів.
6. Які особливості закладено в мережі Кохонена? Для яких завдань ці мережі є ефективними?
7. Назвати особливості, тип і призначення мережі Хопфілда.
8. Яку основну ідею покладено в архітектуру нейромережі Автокодувальник.
9. Назвати основні завдання, вирішення яких покладаються на згорткові мережі
10. Які завдання успішно вирішують рекурентні мережі?

## Використані джерела

1. Фундаментальні відмінності нейромереж. Частина 1 - <https://blog.heyml.com/разновидности-нейронных-сетей-часть-1-12c4f7da8e32>
2. Фундаментальні відмінності нейромереж. Частина 2 - <https://blog.heyml.com/нейрон-персептрон-кошка-фундаментальные-отличия-нейросетей-часть-2-efafc7c089f6>
3. Нейромережі и глибоке навчання, глава 1 - <https://habr.com/ru/post/456738/>
4. Нейромережі и глибоке навчання, глава 2 - <https://habr.com/ru/post/457980/>
5. Нейромережі и глибоке навчання, глава 3 - <https://habr.com/ru/post/458724/>
6. Нейромережі и глибоке навчання, глава 4 - <https://habr.com/ru/post/461659/>
7. Нейромережі и глибоке навчання, глава 5 - <https://habr.com/ru/post/462381/>
8. Нейромережі и глибоке навчання, глава 5 - <https://habr.com/ru/post/463171/>
9. Стенфордський курс: «Згорткові нейронні мережі для візуального розпізнавання» <https://www.reg.ru/blog/stenfordskij-kurs-lekciya-1-vvedenie/>
10. Словник по нейронних мережах <http://textbot.org/Dictionary>
11. Що можуть нейронні мережі <https://neural-university.ru/neural-networks-basics>