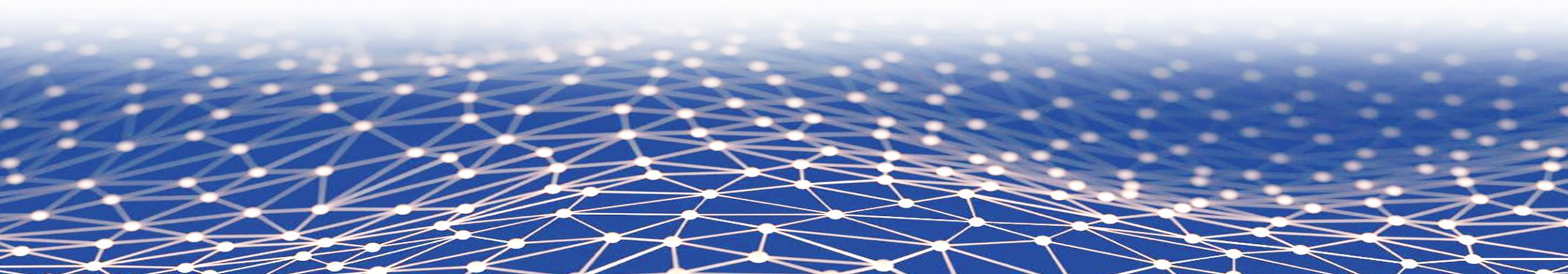


Початок роботи з нейронними мережами.

Структура нейронної мережі. Шари, моделі. Нейромережеві бібліотеки. Keras. TensorFlow. Theano. CNTK. Приклади застосування.



Що таке штучна нейронна мережа?

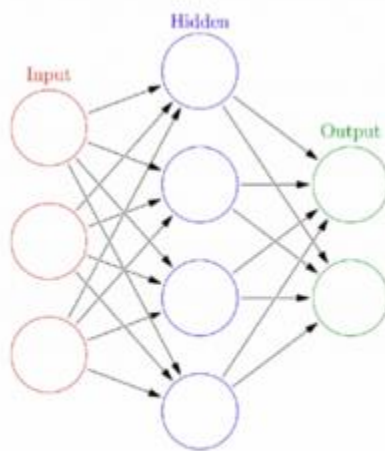
Штучні нейронні мережі (ШНМ) - це програмна імплементація нейронних структур нашого мозку.

Мозок містить *нейрони*, які є свого роду **органічними перемикачами**. Вони можуть змінювати тип переданих сигналів в залежності від електричних або хімічних сигналів, які в них передаються.

Нейронна мережа у людському мозку - величезна взаємопов'язана система нейронів, де сигнал, який передається одним нейроном, може передаватися у тисячі інших нейронів.



Биологическая нейронная сеть



Искусственная нейронная сеть

Навчання відбувається через повторну активацію деяких нейронних з'єднань. Чим більша імовірність виведення потрібного результату при відповідній вхідній інформації.

Такий вид навчання використовує *зворотний зв'язок* - при правильному результаті нейронні зв'язки, які виводять його, стають більш щільними.

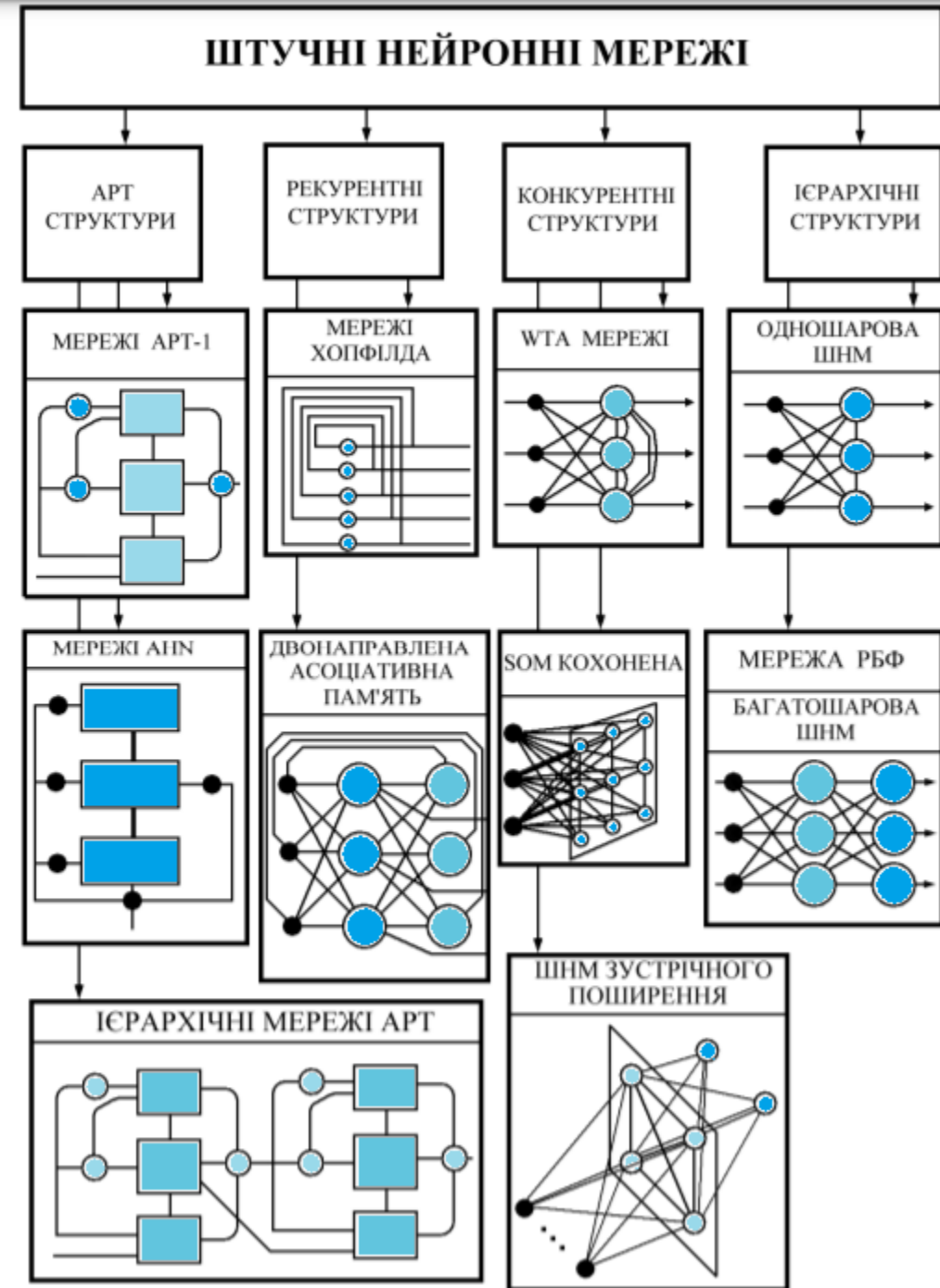
Основні функції штучних нейронних мереж

Функція	Зміст	Застосування
Класифікація образів	визначення приналежності вхідного образу, представленого вектором ознак до одного або декількох попередньо визначених класів;	розпізнавання літер, мови.
Кластеризація/ категоризація	віднесення схожих образів до одного кластеру;	для здобування знання, стиснення даних та дослідження їх властивостей.
Апроксимація функції	знаходження невідомої функції F , спотвореної шумом;	для задач моделювання.
Прогноз/ передбачення	передбачення значення в наступний період часу;	для прийняття рішень в бізнесі, науці та техніці.
Оптимізація	знаходження такого рішення, яке задовольняє системі обмежень і максимізує або мінімізує цільову функцію;	для оптимізації проблем в математиці, статистиці, економіці та ін.
Асоціативна пам'ять	знаходження необхідного документа по заданому змісту;	мультимедійні інформаційні бази даних.
Управління	розрахунок такого вхідного впливу, при якому система діє за бажаною траєкторією, заданою еталонною моделлю;	керівництво приладами.

Структура нейронної мережі.

Сьогодні відома велика кількість нейронних структур та їх модифікацій, що орієнтовані на вирішення конкретного типу задач.

Найбільш відомі типи таких структур



Основні властивості ШНМ:

– локальна обробка інформації в штучному нейроні, який є базовою структурною одиницею мережі;

– паралелізм, результатом якого є вирішення глобальної задачі шляхом представлення її у вигляді множини локальних задач, що тісно взаємодіють між собою;

– здатність до навчання, яке підвищує ефективність роботи мережі;

– здатність до розподіленого зберігання знань, які були одержані в ході навчання.

ШНМ задають у вигляді направлених графів, вершинами яких є нейрони, а ребрами позначені міжнейронні зв'язки.

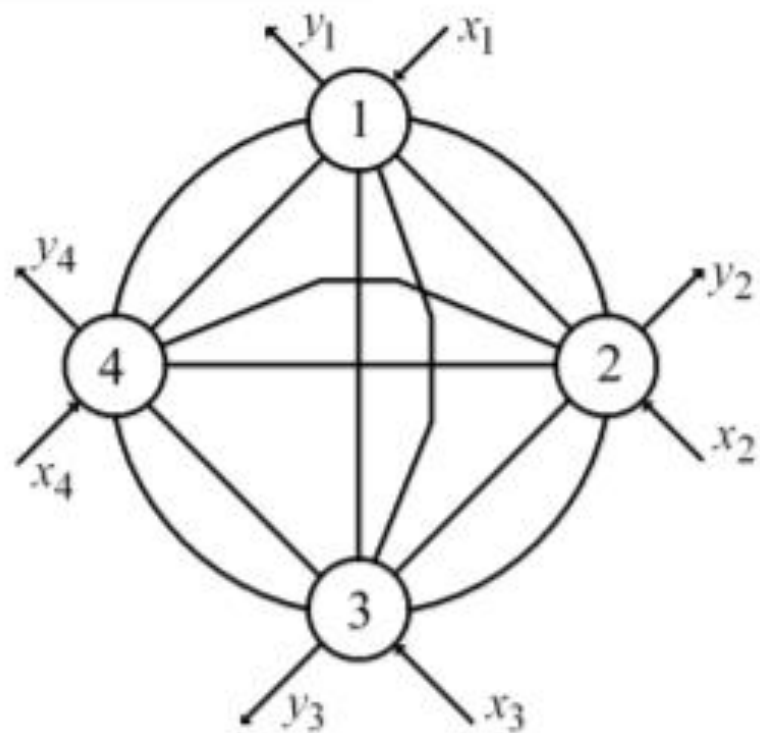
Архітектури поділяють:

– мережі з повним набором міжнейронних зв'язків;

– мережі з фіксованим індексом оточення;

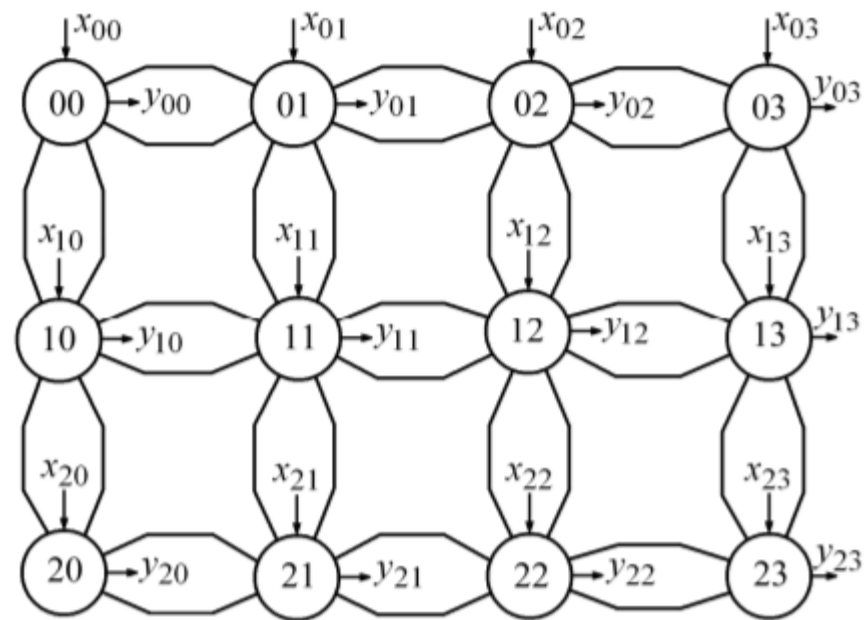
– мережі з пошаровою структурою.

У ШНМ із повним набором міжнейронних зв'язків забезпечується можливість взаємодії кожного нейрона мережі з будь-яким іншим.



Повне з'єднання чотирьох нейронів.

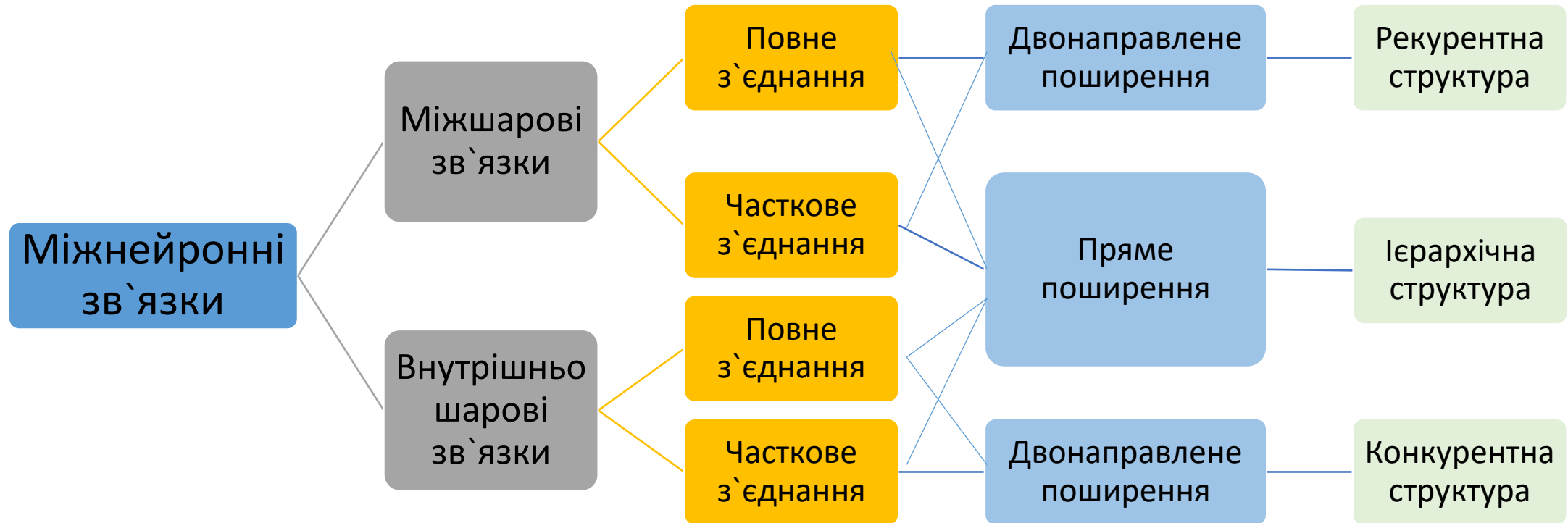
У випадку, коли необхідно використовувати структури з великою кількістю нейронів, застосовують кліткові структури з фіксованим індексом оточення.



Кліткова нейронна мережа з індексом оточення 4.



Класифікація видів міжнейронних зв'язків.



Найбільш поширеними є структуровані за шарами нейронні мережі, які, в залежності від свого функціонального призначення, можуть містити однотипні або різнотипні нейрони. Виходячи з шарової структури ШНМ, характер міжнейронних зв'язків має свої міжшарові та внутрішньшарові особливості.

Навчання нейронних мереж зосереджено на наступних об'єктах:

- шарах, які об'єднуються в мережу (або модель);
- вихідних даних і відповідність їх цілям;
- функції втрат, які визначають сигнал зворотного зв'язку, який використовується для навчання;
- оптимізаторі, який визначає, як відбувається навчання.

Мережа складається з шарів, які об'єднуються в ланцюжок і відображають вихідні дані в передбаченнях.

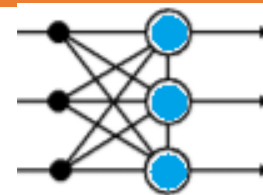
Потім функція втрат порівнює ці передбачення з цілями і повертає значення втрати: міру відповідності передбачення, виробленого мережею, очікуваного результату.

Оптимізатор використовує це значення втрати для зміни вагів мережі.

Класифікація мереж за структурою побудови.

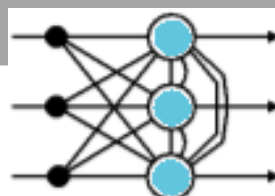
Ієрархічні структури

- Одношарові нейронні мережі прямого поширення
- Багатшарові нейронні мережі прямого поширення
- РБФ-мережі



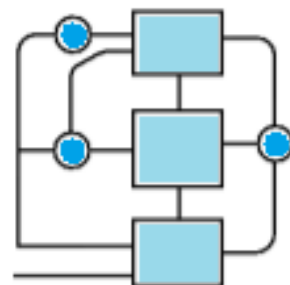
Конкурентні структури

- WTA структура Ліпмана–Хеммінга
- Нейронна мережа Кохонена

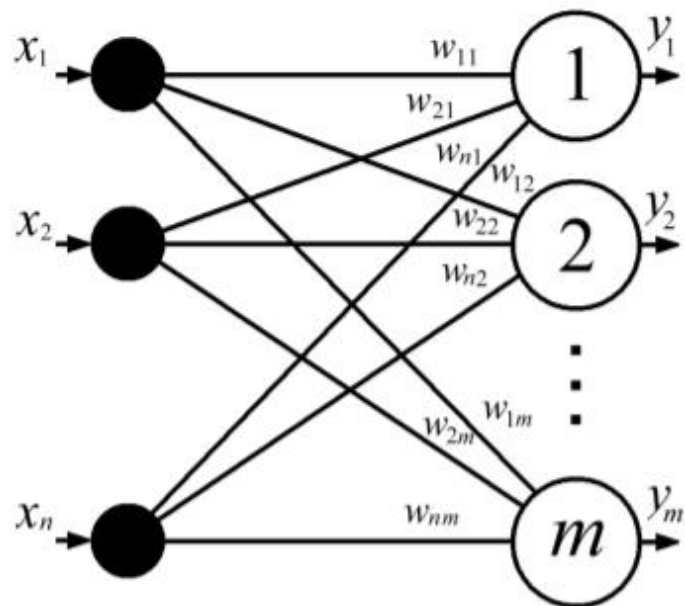
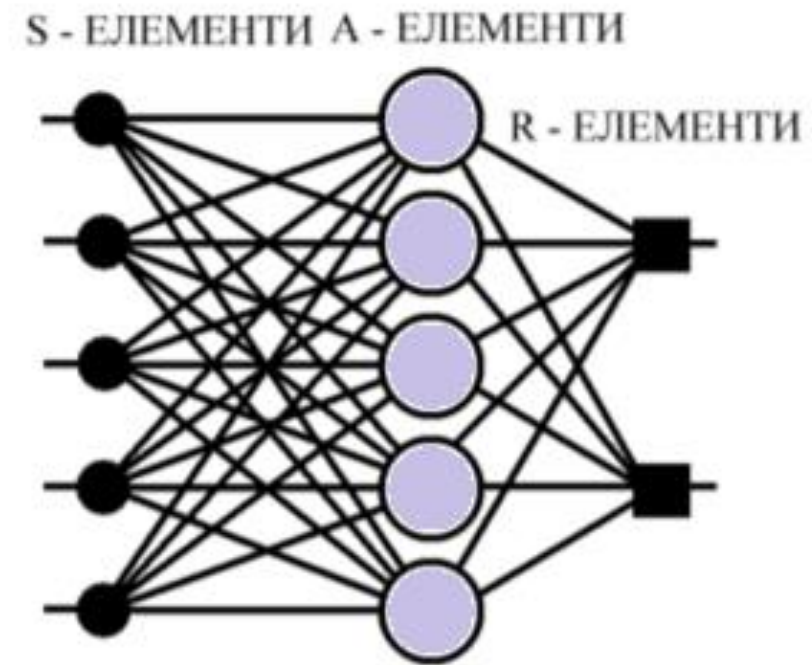


Рекурентні структури

- Нейронна мережа Хопфілда
- Бінарна модель
- Двонаправлена асоціативна пам'ять
- Стохастичні рекурентні мережі



Першою штучною нейронною мережею, яку за сучасною термінологією відносять до одношарових, був персептрон Розенблатта. →



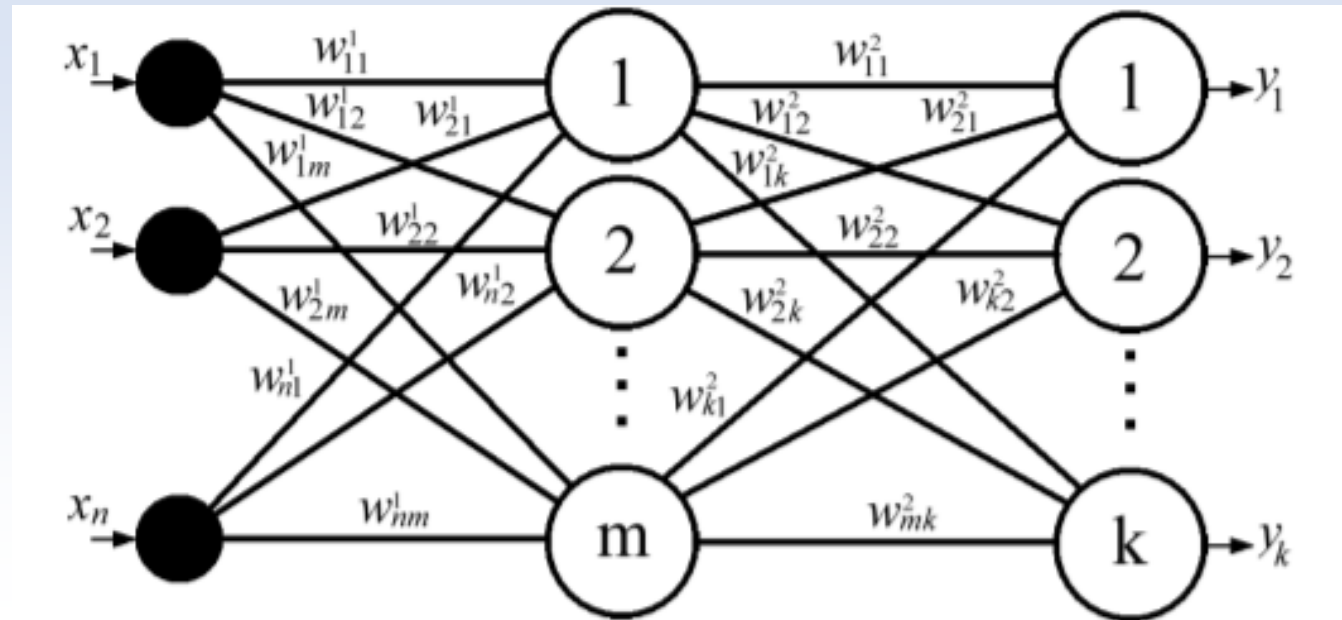
А-елемента. Сучасним прототипом персептрона Розенблатта є одношарова нейронна мережа прямого поширення.



Багатошарові нейронні мережі прямого поширення

Спроби застосувати одношарові нейронні мережі для розв'язування широкого кола задач наштовхнулися на ряд труднощів, пов'язаних з проблемою лінійної роздільності. Природним вирішенням цієї проблеми стало застосування багатошарових ШНМ, що нагадують багатошарові структури.

Розглянемо ієрархічну структуру, в якій нейрони структуровані за шарами. Вона складається з m нейронів першого (прихованого) шару, які одночасно здатні прийняти вхідний вектор сигналів, та k нейронів другого (вихідного) шару.



РБФ-мережі

Штучні нейронні мережі, які використовують радіальні базисні функції, називають РБФ-мережами. Вони є окремим випадком двошарових нейронних мереж прямого поширення, в яких прихований шар нейронів використовує радіальні базисні функції типу гаусової як активаційні. В просторі вхідних векторів вибирають вектор, який називають центром, і відповідно до нього задають вагові коефіцієнти прихованого шару. Аргумент активаційної функції для нейрона прихованого шару визначатиметься відстанню між вхідним вектором та вектором прихованого шару.

Формула для визначення вектору вхідного сигналу

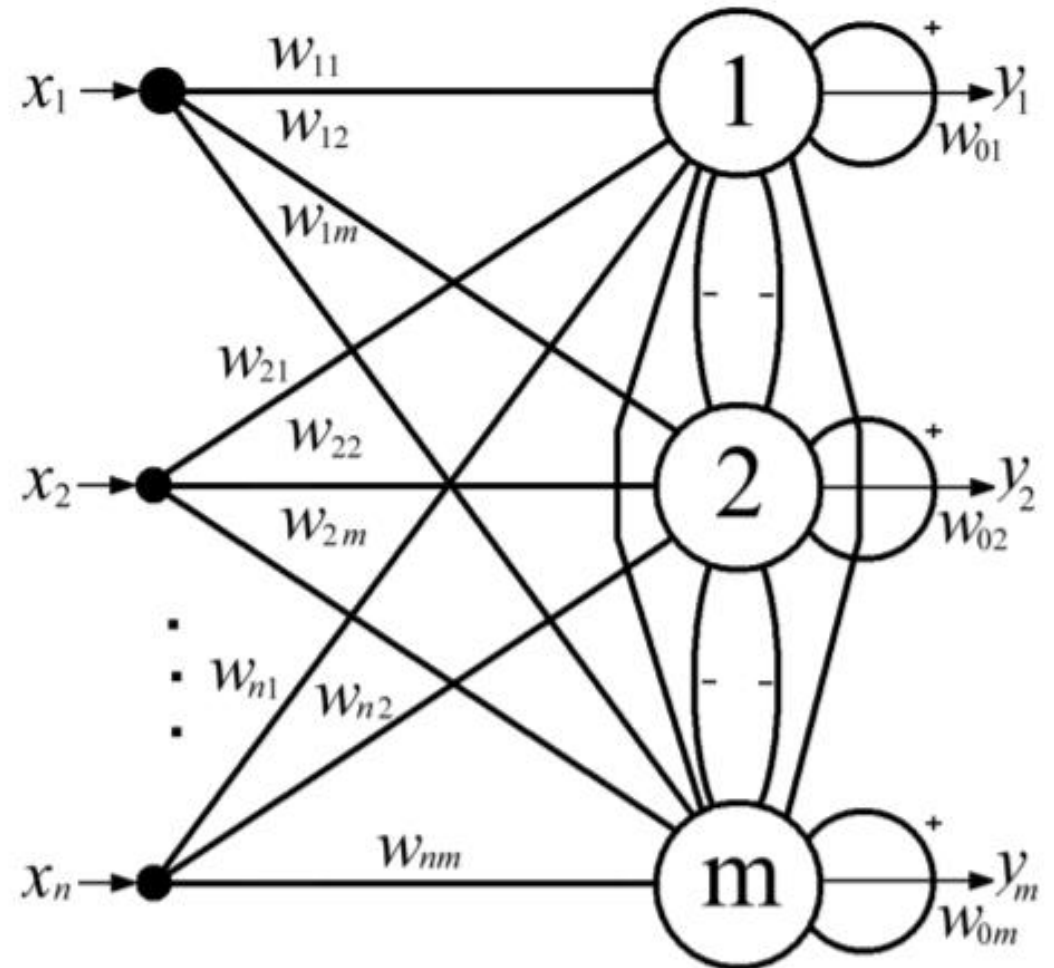
$$v_j = \|X - W_j\| = \sqrt{\sum_{i=1}^n (x_i - w_{ij})^2} \quad \longrightarrow \quad y_i = \exp\left[-\left(\frac{v_j}{\sigma_j}\right)^2\right] = \exp\left[-\left(\frac{\|X - W_j\|}{\sigma_j}\right)^2\right].$$

Формула для визначення вектору вихідного сигналу

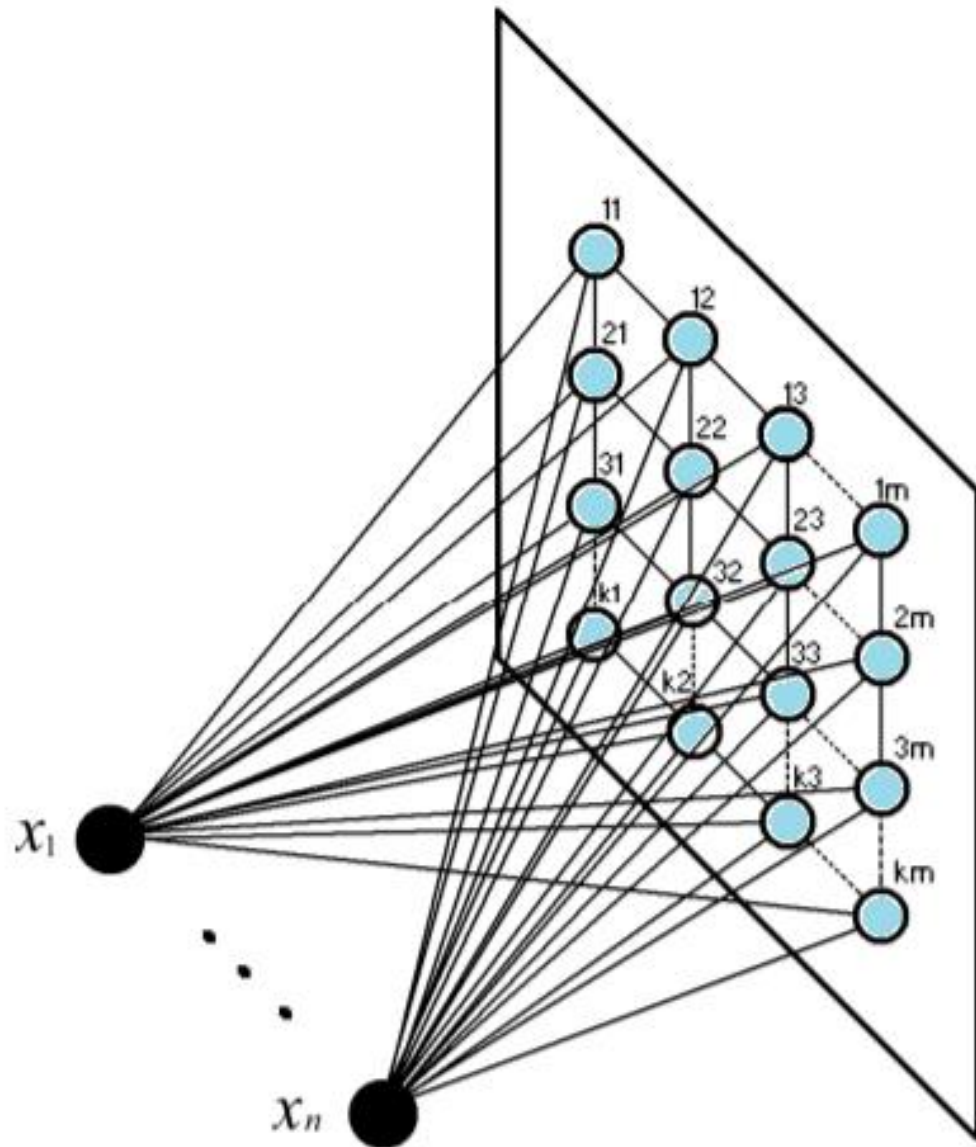
WTA структура Ліпмана–Хеммінга

Для розв'язання задач класифікації застосовують велику кількість структур, елементами яких є зірки Гроссберга.

Елементи нейронної мережі Ліпмана–Хеммінга мають властивості вхідної та вихідної зірок Гроссберга. Структура, реалізує принцип “Winner Take All” (WTA). Вона містить один шар з m нейронів, кожен з яких відповідає за свій клас об'єктів, які потрібно класифікувати. У загальному випадку кожний елемент вхідного вектора зв'язаний з кожним класифікуючим нейроном.



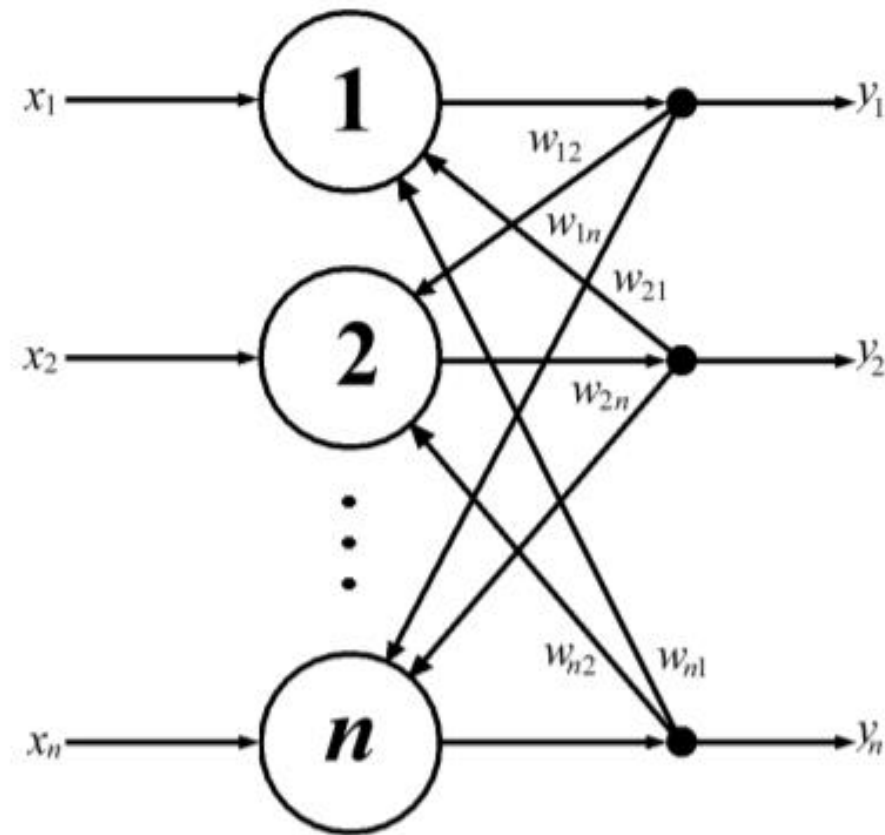
Нейронна мережа Кохонена



На відміну від мережі Ліпмана–Хеммінга модель нейронної мережі Кохонена характеризується структурою розподіленої пам'яті. Така структура дозволяє уникнути катастрофічної деградації у випадку відмови одного з нейронів. Ефект підвищеної живучості досягається саме завдяки розподіленій пам'яті, дія якої проявляється за рахунок того, що за класифікацію вхідного вектора відповідає не один нейрон, а кластер нейронів.

Нейронна мережа Хопфілда

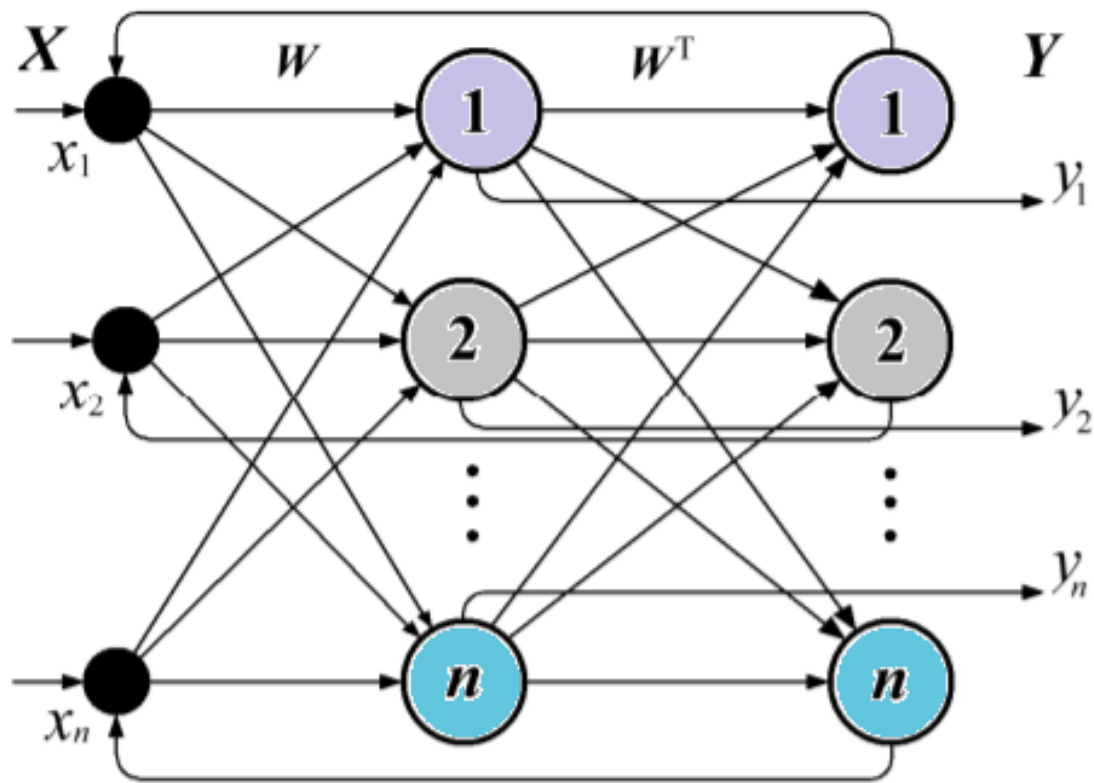
Структура рекурентної нейронної мережі, яка показана на рис., складається з одного шару нейронів, виходи яких через спеціальні пристрої з'єднані зі входами всіх нейронів цього ж шару, крім тих зв'язків, що з'єднують вихід нейрона з його власним входом.



Формування аргументу активаційної функції довільного нейрона j відбувається за формулою:

$$v_j = \sum_{\substack{i=1 \\ i \neq j}}^n w_{ij} y_i + x_j .$$

Двонаправлена асоціативна пам'ять



Основною властивістю асоціативної пам'яті, що реалізована на мережі Хопфілда, є здатність відтворювати образи шляхом їх асоціації з частково помилковим запитом.

Ця структура дозволяє додатково встановлювати асоціативні зв'язки між різними образами.

Бінарна модель

Бінарна модель базується на використанні порогової активаційної функції, яку задають виразом:

$$y_j = f(v_j) = \begin{cases} 1 & \text{при } v_j > T_j, \\ 0 & \text{при } v_j < T_j, \\ y_j & \text{при } v_j = T_j. \end{cases}$$

Стохастичні рекурентні мережі

Еволюція стійких детермінованих рекурентних мереж завжди відбувається у напрямку локального мінімуму енергії. Ця властивість є корисною у випадку, коли нейромережа моделює функції асоціативної пам'яті. Коли ж ставиться задача пошуку глобального мінімуму, стійкий стан мережі у локальному мінімумі є небажаним явищем. Для розв'язування подібних задач використовують стохастичні рекурентні мережі.

Принциповою відмінністю стохастичних алгоритмів є відмова від заборони переходу мережі у стан з більш високим рівнем енергії. Модель, що описує роботу стохастичної мережі, нагадує процес застигання металу, кожен атом якого знаходиться в неупорядкованому русі і має різний запас енергії, але загальна температура прямує до нуля. Імовірність переходу між енергетичними рівнями визначається за формулою Больцмана:

$$P(E) = \frac{1}{1 + \exp\left\{-\frac{E_2 - E_1}{T}\right\}},$$

де $P(E)$ — імовірність переходу системи зі стану з енергією E_1 у стан з енергією E_2 ; T — температура за шкалою Кельвіна.

При великих значеннях температури величина імовірності $P(E) = \frac{1}{2}$, тобто наступний стан системи, не залежить від її попереднього стану, що відповідає хаотичній поведінці. Зі

Будівельні блоки глибокого навчання – Шари.

Шари, є фундаментальною структурою даних в нейронних мережах.

Шар - це модуль обробки даних, що приймає на вході і повертає на виході один або кілька тензорів.

Деякі верстви незберігають стан, але частіше це не так: ваги шару, один або кілька тензорів, навчаючись із застосуванням алгоритму стохастичного градієнтного спуску, які разом зберігають знання мережі.

Моделі: мережі шарів

Модель глибокого навчання є орієнтованим, ациклічним графом шарів. Найчастіше на практиці використовується лінійний стек шарів, що відображають єдиний вхід в єдиний вихід. Однак під час руху вперед вам зустрінеться набагато ширший спектр топологій мереж.

Приклади топології мереж:

- мережі з двома гілками (two-branch networks);
- багатоголові мережі (multihead networks);
- вхідні блоки (inception blocks).

Функції втрат і оптимізатори.

Після того як ви визначилися з архітектурою мережі, потрібно також вибрати ще два параметри:

Функцію втрат (цільову функцію), яка повертає кількісну оцінку, яка буде мінімізуватися в процесі навчання. Являє собою міру успіху у вирішенні текучого завдання.

Оптимізатор, який визначає, як буде змінюватися мережа під впливом функції втрат. Реалізує конкретний варіант стохастичного градієнтного спуску (Stochastic Gradient Descent, SGD).

Вибір правильної цільової функції для вирішення конкретного завдання грає дуже важливу роль: ваша мережа буде використовувати будь-яку можливість, щоб мінімізувати втрати; тому, якщо цільова функція в повному обсязі корелює з успішним вирішенням завдання, ваша мережа в кінцевому підсумку зробить результат, можливо, зовсім не той, що вам потрібен

Після вибору опції втрат. Наприклад, для класифікації в дві категорії можна використовувати функцію бінарної перехресної ентропії, для класифікації в не стільки категорій - багатозначною перехресною ентропією, для задач регресії - середньоквадратичної помилки, для навчання на послідовностях - асоціатив тимчасової класифікації (Connectionist Temporal Classification, CTC) і т. д.

Тільки стикаючись з дійсно новими дослідницькими завданнями, вам доведеться розробляти свої цільові функції. У наступних декількох розділах ми докладно пояснимо, які функції втрат слід вибирати для широкого кола типових завдань.

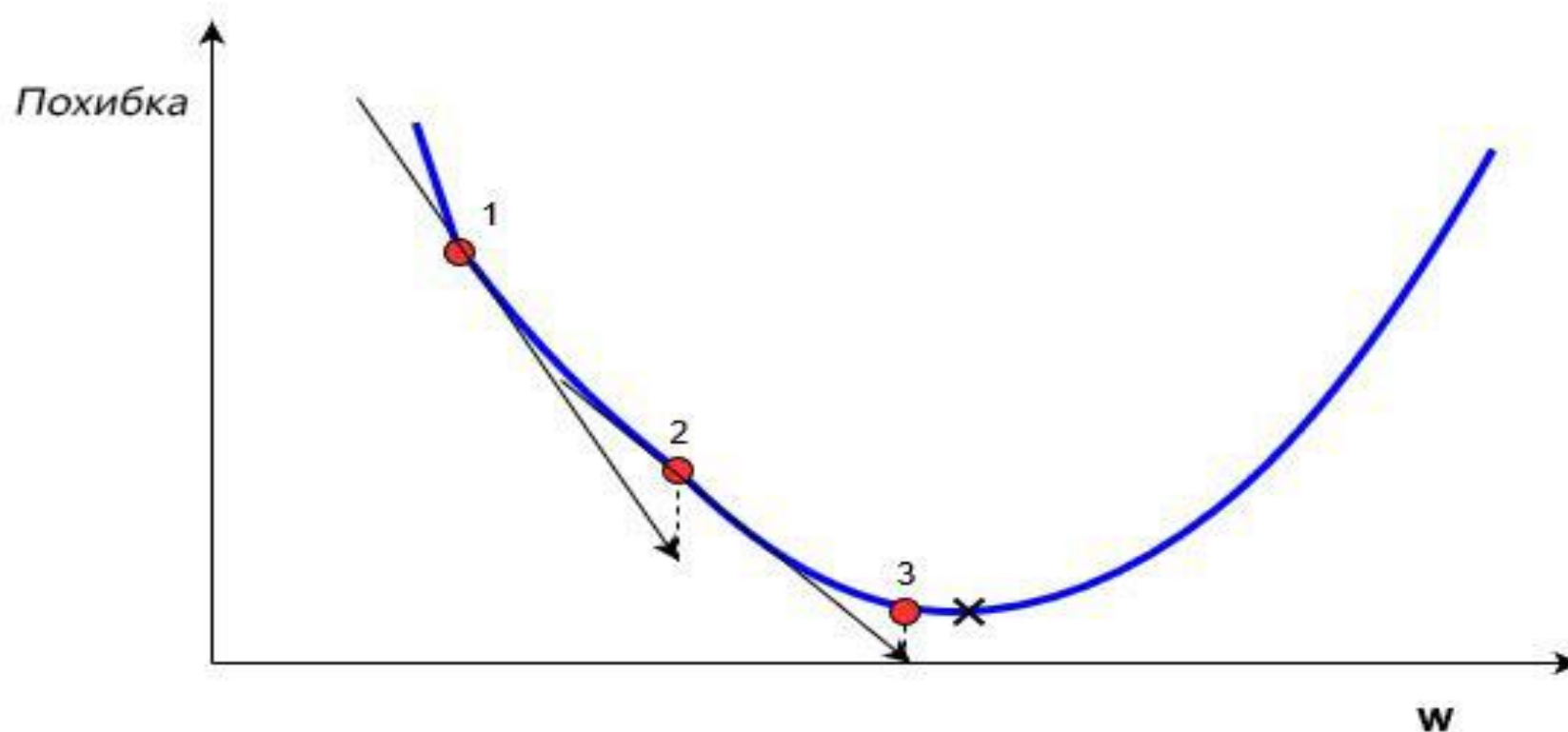
Градiєнтний спуск та оптимiзацiя

Розрахунки значень ваг, якi з'єднують шари у мережi, i є тим, що ми називаємо навчанням системи. У контрольованому навчаннi iдея полягає у тому, щоб зменшити похибку мiж входом та потрiбним виходом. Якщо ми маємо нейронну мережу з одним вихiдним шаром та деякий вхiд i ми хочемо, щоб на виходi було число 2, але мережа видає 5.

Говорячи мовою математики, можна знайти норму помилки.

У навчаннi мережi, використовуючи (x, y) , метою є покращувати її у знаходженнi правильного y при вiдомому x . Це робиться через змiну значень ваг, щоб мiнiмiзувати похибку. Як тодi змiнювати їх значення? Для цього нам i знадобиться **градiєнтний спуск**. Розглянемо наступний графiк:

На цьому графіку зображено похибку, залежну від скалярного значення ваги, w . Мінімально можлива похибка позначена чорним хрестом, але ми не знаємо яке саме значення w дає нам це мінімальне значення. Підрахунок починається з рандомного значення змінної w , яка дає похибку, позначену червоною крапкою під номером "1" на кривій. Нам потрібно змінити w таким чином, щоб досягнути мінімальної похибки, чорного хреста. Одним з найпоширеніших способів є градієнтний спуск.



Моделювання нейронних мереж засобами бібліотек мови Python

Мова Python є однією з найбільш популярних мов програмування в останнє десятиріччя. Завдяки безкоштовним засобам розробки та потужним бібліотекам Python знаходить широке використання при розробці прикладного програмного забезпечення та вирішенні наукових задач..

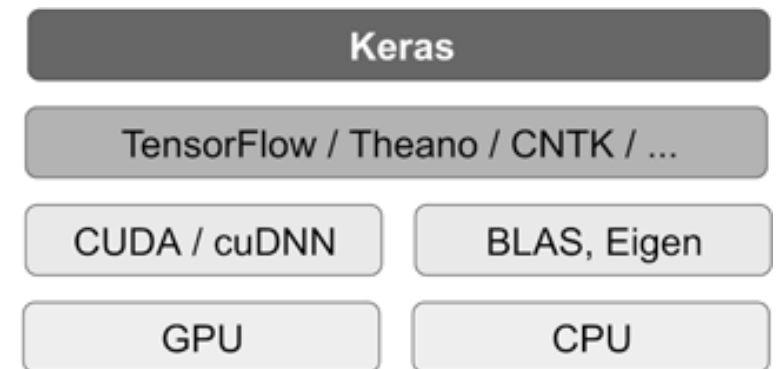
Тип мережі	Функція створення	Кількість шарів	Підтримка функцій навчання	Функція помилки
Одношаровий персептрон	newp	1	train_delta	SSE
Багатошаровий персептрон (БНМ)	newff	≥ 1	train_gd, train_gdm, train_gda, train_gdx, train_rprop, train_bfgs, train_cg	SSE
Конкуруючий шар (SOM)	newc	1	train_wta, train_cwta*	SAE
LVQ	newlvq	2	train_lvq	MSE
Мережа Елмана	newelm	≥ 1	train_gdx	MSE
Мережа Хопфілда	newhop	1	—	—
Мережа Хеммінга	newhem	2	—	—

Keras, TensorFlow, Theano і CNTK

Keras - це бібліотека рівня моделі, що надає високо рівневі будівельні блоки для конструювання моделей глибокого навчання. Вона не реалізує низькорівневі операції, такі як маніпуляції з тензорами і диференціювання, - для цього використовується спеціалізована і оптимізована бібліотека підтримки тензорів.

При цьому Keras не покладається на якусь одну бібліотеку підтримки тензорів, а використовує модульний підхід; тобто до фреймворку Keras можна підключити декілька різних низькорівневих бібліотек.

В даний час підтримуються три такі бібліотеки: TensorFlow, Theano і Microsoft Cognitive Toolkit (CNTK). В майбутньому Keras, швидше за все, буде розширений ще декількома низькорівневими механізмами підтримки глибокого навчання.



TensorFlow, CNTK і Theano - це одні з провідних платформ глибокого навчання в даний час.

- Theano розроблена в лабораторії MILA Монреальського університету;
- TensorFlow розроблена в Google, а CNTK розроблена в Microsoft.

Будь-код, який використовує Keras, можна запускати з будь з цих бібліотек без необхідності змінювати щось в коді: ви можете легко перемикатися між ними в процесі розробки, що часто виявляється корисно, наприклад, якщо одна з бібліотек показує більш високу продуктивність при вирішенні даного конкретного завдання.

