НЕЙРОННА МЕРЕЖА

Штучна нейронна мережа (ANN) – це обчислювальна система, натхненна біологічними нейронними мережами (тобто, мозком живих організмів) і є підмножиною ML та основою DL. Також, штучні нейронні мережі називають: нейронна мережа (NN) або нейромережа.

ANN складається з [штучних нейронів](https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B8%D0%B9_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD), які концептуально походять від [біологічних](https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD). Кожен штучний нейрон має входи та видає єдиний вихід, який можливо надсилати багатьом іншим нейронам. Входи можуть бути значеннями ознак зразка зовнішніх даних, таких як зображення чи документи, або вони можуть бути виходами інших нейронів. Виходи кінцевих нейронів виходу нейронної мережі завершують завдання.

Щоби знайти вихід нейрона, береться зважена сума всіх входів, зважених за вагами з'єднань від входів до нейрона. До цієї суми додається зміщення ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) bias). Цю зважену суму іноді називають збудженням. Потім її пропускають крізь, зазвичай нелінійну, [передавальну функцію](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_%D1%88%D1%82%D1%83%D1%87%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%B0) для отримання виходу.

Нейрони зазвичай впорядковано в кілька шарів, особливо в [глибокому навчанні](https://uk.wikipedia.org/wiki/%D0%93%D0%BB%D0%B8%D0%B1%D0%BE%D0%BA%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F). Нейрони одного шару з'єднуються лише з нейронами безпосередньо попереднього й наступного шарів. Шар, який отримує зовнішні дані, це шар входу. Шар, який видає кінцевий результат, це шар виходу. Між ними є нуль або більше прихованих шарів.

ANN поділяють на багато типів, наприклад:

* Мережа прямого поширення – перший і найпростіший тип. У цій мережі інформація рухається лише від шару входу безпосередньо крізь будь-які приховані шари до шару виходу, без циклів/петель. Також, може називатись: перцептрон або багатошаровий перцептрон.

Використовується в: автокодувані (автокодувальна NN), класифікуванні та розпізнаванні образів (імовірнісна NN), розпізнаванні зображень та відео, рекомендаційних системах та обробці природньої мови (згорткова NN).

* Мережі з регуляторним зворотним зв'язком. У ній використовуються універсальні методи для сенсорного розпізнавання.
* Мережа радіально базисних функцій (RBF networks) у [математичному моделюванні](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C) — це [штучна нейронна мережа](https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), яка використовує [радіальні базисні функції](https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D1%96%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B8%D1%81%D0%BD%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F) у якості [функції активації](https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_%D1%88%D1%82%D1%83%D1%87%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%B0). Виходом мережі є [лінійна комбінація](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B0_%D0%BA%D0%BE%D0%BC%D0%B1%D1%96%D0%BD%D0%B0%D1%86%D1%96%D1%8F) радіальних базисних функцій входу та параметрів нейрона. Мережі радіальних базисних функцій мають багато застосувань, зокрема, такі як прогнозування [часових рядів](https://uk.wikipedia.org/wiki/%D0%A7%D0%B0%D1%81%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%8F%D0%B4), [задачі класифікації](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%97) та [керування системою](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F).

Радіальна базисна функція (RBF) — [дійснозначна функція](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B9%D1%81%D0%BD%D0%BE%D0%B7%D0%BD%D0%B0%D1%87%D0%BD%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F), чиє значення залежить від відстані до [початку системи координат](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D0%B0%D1%82%D0%BE%D0%BA_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82). Нормою зазвичай є [евклідова відстань](https://uk.wikipedia.org/wiki/%D0%95%D0%B2%D0%BA%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%B0_%D0%B2%D1%96%D0%B4%D1%81%D1%82%D0%B0%D0%BD%D1%8C), хоча можлива будь-яка [функція відстані](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)).

* Глибока мережа переконань (DBN) — це клас [глибоких](https://uk.wikipedia.org/wiki/%D0%93%D0%BB%D0%B8%D0%B1%D0%BE%D0%BA%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F) [нейронних мереж](https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), що складено з кількох шарів [латентних змінних](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B5%D0%BD%D1%82%D0%BD%D0%B0_%D0%B7%D0%BC%D1%96%D0%BD%D0%BD%D0%B0) («прихованих вузлів»), зі з'єднаннями між шарами, але не між вузлами всередині кожного шару. Спостереження щодо того, що DNB можливо тренувати [жадібно](https://uk.wikipedia.org/wiki/%D0%96%D0%B0%D0%B4%D1%96%D0%B1%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), по одному шару за раз, привело до створення одного з перших дієвих алгоритмів [глибокого навчання](https://uk.wikipedia.org/wiki/%D0%93%D0%BB%D0%B8%D0%B1%D0%BE%D0%BA%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F).

Використовується в: електроенцефалографії і пошуку нових ліків.

* Рекурентні нейронні мережі (RNN) — це клас [штучних нейронних мереж](https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), у якому з'єднання між вузлами утворюють [граф орієнтований](https://uk.wikipedia.org/wiki/%D0%9E%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9_%D0%B3%D1%80%D0%B0%D1%84) у часі. Це створює внутрішній стан мережі, що дозволяє їй проявляти динамічну поведінку в часі. На відміну від [нейронних мереж прямого поширення](https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D0%B3%D0%BE_%D0%BF%D0%BE%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8F), RNN можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей входів.

Використовується в: вивченні мови та розпізнаванні неперервного рукописного тексту (NN з довгою короткочасною пам’яттю).

* Модульна нейронна мережа — група [нейронних мереж](https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D1%96_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%96_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D1%96) (які в даному випадку називаються модулями), що керуються певним посередником. Кожна нейронна мережа слугує модулем і оперує окремими входами для вирішення певних підзавдань із групи завдань, які повинна виконати модульна нейронна мережа модулі якої не взаємодіють між собою.
* У фізичній нейронній мережі використовується електрично регульований опірний матеріал для імітування штучних зв’язків (синапсів).
* Динамічна нейронна мережа - розглядає нелінійну багатовимірну поведінку та включає навчання залежної від часу поведінки, такої як перехідні явища та ефекти затримки.
* Мережі з пам’яттю здатні навчатись з врахуванням попереднього досвіду та використовувати його для створення вихідного сигналу.

Загалом, можна виділити найпопулярніші сфери використання ANN:

* Розпізнавання обличчя за допомогою згорткової нейронної мережі (CNN).
* Прогнозування фондового ринку за допомогою багатошарового перцептрона (MLP).
* Соціальні медіа, зокрема видобуток даних із програм соціальних мереж за допомогою MLP.
* Аерокосмічна інженерія, зокрема моделювання нелінійних динамічних систем у часі за допомогою нейронних мереж із затримкою часу (TDNN) для розпізнавання ознак позиції.
* Захист, зокрема керування безпілотними дронами за допомогою CNN.
* Охорона здоров’я, наприклад: виявлення рентгенівських променів, комп’ютерна томографія та ультразвук. Тут, також, широко використовується згорткові (CNN) та повторювальні нейронні мережі (RNN).
* Перевірка підпису та аналіз почерку за допомогою CNN.
* Прогнозування погоди за допомогою MLP, CNN або RNN, також, можуть використовуватись комбіновані моделі з MLP або RNN в поєднанні з CNN, що дозволяє створити прогноз погоди до 15 днів на перед.

ANN поділяються на багато типів і мають не менше призначень, але і фреймворків для роботи з ними є достатньо для різного рівня та призначення, наприклад:

* TensorFlow – один із найпопулярніших фреймворків глибокого навчання, який підтримує такі мови програмування, як Python, C++ та R, для створення моделей глибокого навчання. Він доступний як на ПК, так і на смартфоні.

TensorFlow є одним із найбільш бажаних фреймворків глибокого навчання, оскільки він заснований на Python, підтримується Google і постачається з першокласною документацією та покроковими інструкціями.

* PyTorch – це фреймворк глибокого навчання на основі Lua, який широко використовується серед гігантів галузі, таких як Facebook, Twitter і Google.

Останніми роками PyTorch отримав високий рівень впровадження серед спільноти фреймворків глибокого навчання та вважається справжнім конкурентом TensorFlow. PyTorch — це, по суті, фреймворк глибокого навчання Port to Torch, який використовується для побудови глибоких нейронних мереж і виконання тензорних обчислень, які мають високу складність.

[PyTorch](https://pytorch.org/) працює на Python, що означає, що будь-хто, хто має базове розуміння Python, може розпочати створення своїх моделей глибокого навчання.

* Deeplearning4j – це бібліотека глибокого навчання для віртуальної машини Java (JVM). Він розроблений на Java та підтримує інші мови JVM, такі як Scala, Clojure та Kotlin.

Оскільки ця структура глибокого навчання реалізована на Java, вона набагато ефективніша порівняно з Python. Коли справа доходить до завдань [розпізнавання зображень](https://marutitech.com/working-image-recognition/) за допомогою кількох графічних процесорів (GPU), DL4J такий же швидкий, як Caffe. Ця структура демонструє незрівнянний потенціал для розпізнавання зображень, виявлення шахрайства, аналізу тексту, додавання тегів до частин мови та обробки природної мови.

* Microsoft Cognitive Toolkit (раніше відомий як CNTK) – це платформа глибокого навчання з відкритим кодом для навчання моделей глибокого навчання. Широко відомий легким навчанням і поєднанням популярних типів моделей на серверах.

Коли справа доходить до винаходу нових складних типів шарів, користувачам не потрібно реалізовувати їх мовою низького рівня завдяки дрібній деталізації будівельних блоків. Microsoft Cognitive Toolkit підтримує нейронні моделі типу RNN і CNN та здатний вирішувати проблеми з розпізнаванням зображення, рукописного тексту та мовлення.

* ONNX або Open Neural Network Exchange було розроблено як екосистему глибокого навчання з відкритим кодом. Розроблений Microsoft і Facebook, ONNX є системою глибокого навчання, яка дозволяє розробникам легко перемикатися між платформами.

ONNX набув популярності завдяки своїй гнучкості та сумісності. За допомогою ONNX можна легко перетворити свою попередньо навчену модель у файл, який потім можна об’єднати з додатком. ONNX — це потужний інструмент, який запобігає блокуванню фреймворку, надаючи легший доступ до оптимізації апаратного забезпечення та надаючи можливість спільного використання моделі.

ЕВКЛІДОВА ВІДСТАНЬ

У [математиці](https://en.wikipedia.org/wiki/Mathematics) евклідова відстань між двома точками в [евклідовому просторі](https://en.wikipedia.org/wiki/Euclidean_space)[–](https://en.wikipedia.org/wiki/Line_segment) це [довжина](https://en.wikipedia.org/wiki/Length) відрізка між двома [точками](https://en.wikipedia.org/wiki/Point_(geometry)). Її можна обчислити з [декартових координат](https://en.wikipedia.org/wiki/Cartesian_coordinate) точок за допомогою [теореми Піфагора](https://en.wikipedia.org/wiki/Pythagorean_theorem), тому її іноді називають відстанню Піфагора.

Відстань між двома об’єктами, які не є точками, зазвичай визначається як найменша відстань серед пар точок від двох об’єктів. Відомі формули для обчислення відстані між різними типами об’єктів, наприклад [відстані від точки до лінії](https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line). У розвиненій математиці поняття відстані було узагальнено для абстрактних [метричних просторів](https://en.wikipedia.org/wiki/Metric_space), і вивчалися інші відстані, крім евклідових. У деяких програмах [статистики](https://en.wikipedia.org/wiki/Statistics) та [оптимізації](https://en.wikipedia.org/wiki/Mathematical_optimization) квадрат евклідової відстані використовується замість самої відстані.

Евклідова відстань є прототипом відстані в [метричному просторі](https://en.wikipedia.org/wiki/Metric_space) і відповідає всім визначальним властивостям метричного простору:

* Вона є симетричною, тобто відстань між двома точками не залежить від того, яка з двох точок є початковою, а яка кінцевою.
* Вона додатна, тобто відстань між кожними двома різними точками є [додатним числом](https://en.wikipedia.org/wiki/Positive_number), тоді як відстань від будь-якої точки до неї самої дорівнює нулю.
* Вона підкоряється [нерівності трикутника](https://en.wikipedia.org/wiki/Triangle_inequality).

У багатьох застосуваннях, і зокрема при порівнянні відстаней, може бути зручніше опускати кінцевий квадратний корінь при обчисленні евклідових відстаней. Значення, отримане в результаті цього пропуску, є [квадратом](https://en.wikipedia.org/wiki/Square_(algebra)) евклідової відстані та називається квадратом евклідової відстані. Наприклад, [евклідове мінімальне остовне дерево](https://en.wikipedia.org/wiki/Euclidean_minimum_spanning_tree) можна визначити, використовуючи лише порядок між відстанями, а не їх числові значення. Порівняння квадратів відстаней дає той самий результат, але дозволяє уникнути непотрібного обчислення квадратного кореня та обійти проблеми чисельної точності.

Евклідове мінімальне остовне дерево кінцевої [множини](https://en.wikipedia.org/wiki/Finite_set) точок на [евклідовій площині або](https://en.wikipedia.org/wiki/Euclidean_plane)[евклідовому просторі](https://en.wikipedia.org/wiki/Euclidean_space) вищих вимірів з’єднує точки системою [відрізків ліній](https://en.wikipedia.org/wiki/Line_segment) із точками як кінцевими точками, мінімізуючи загальну довжину відрізків. У ньому будь-які дві точки можуть досягати одна одної по шляху через відрізки. Його можна знайти як [мінімальне остовне дерево](https://en.wikipedia.org/wiki/Minimum_spanning_tree) повного [графа](https://en.wikipedia.org/wiki/Complete_graph) з точками як вершинами та [евклідовими відстанями](https://en.wikipedia.org/wiki/Euclidean_distance) між точками як вагами ребер.

У більш просунутих областях математики, коли розглядають евклідів простір як [векторний простір](https://en.wikipedia.org/wiki/Vector_space), його відстань пов’язана з [нормою](https://en.wikipedia.org/wiki/Norm_(mathematics)), яка називається [евклідовою нормою](https://en.wikipedia.org/wiki/Norm_(mathematics)#Euclidean_norm), яка визначається як відстань кожного вектора від [початку координат](https://en.wikipedia.org/wiki/Origin_(mathematics)). Однією з важливих властивостей цієї норми відносно інших норм є те, що вона залишається незмінною при довільних поворотах простору навколо початку координат. Згідно з [теоремою Дворецького](https://en.wikipedia.org/wiki/Dvoretzky%27s_theorem), кожен скінченновимірний [нормований векторний простір](https://en.wikipedia.org/wiki/Normed_vector_space) має високовимірний підпростір, на якому норма приблизно евклідова; евклідова норма є єдиною нормою з цією властивістю. Його можна розширити до нескінченновимірних векторних просторів як L2 норма або L2 відстань. Евклідова відстань надає евклідовому простору структуру [топологічного простору](https://en.wikipedia.org/wiki/Topological_space), [евклідової топології](https://en.wikipedia.org/wiki/Euclidean_topology), з [відкритими кулями](https://en.wikipedia.org/wiki/Open_ball), підмножинами точок, розташованих менше заданої відстані від даної точки, як його [околицями](https://en.wikipedia.org/wiki/Neighbourhood_(mathematics)).

Інші загальні відстані в евклідових просторах і векторних просторах низької розмірності включають:

* [Відстань Чебишева](https://en.wikipedia.org/wiki/Chebyshev_distance), яка вимірює відстань, припускаючи, що доречним є лише найбільш значущий розмір.
* [Манхеттенська відстань](https://en.wikipedia.org/wiki/Manhattan_distance), яка вимірює відстань лише за напрямками, вирівняними по осі.
* [Відстань Мінковського](https://en.wikipedia.org/wiki/Minkowski_distance), узагальнення, яке об’єднує евклідову відстань, відстань Манхеттена та відстань Чебишева.

У [математиці](https://en.wikipedia.org/wiki/Mathematics) та [фізиці](https://en.wikipedia.org/wiki/Physics)вектор — це термін, який у розмовній мові означає деякі [величини](https://en.wikipedia.org/wiki/Physical_quantity), які не можна виразити одним числом, або елементи деяких [векторних просторів](https://en.wikipedia.org/wiki/Vector_space).

У [математиці](https://en.wikipedia.org/wiki/Mathematics) та [фізиці](https://en.wikipedia.org/wiki/Physics) векторний простір (також званий лінійним простором ) — це [набір](https://en.wikipedia.org/wiki/Set_(mathematics)), елементи якого, часто звані [векторами](https://en.wikipedia.org/wiki/Vector_(mathematics_and_physics)), можна додати разом і помножити на числа, які називаються [скалярами](https://en.wikipedia.org/wiki/Scalar_(mathematics)). Скаляри часто є [дійсними числами](https://en.wikipedia.org/wiki/Real_number), але можуть бути [комплексними числами](https://en.wikipedia.org/wiki/Complex_number) або, загалом, елементами будь-якого [поля](https://en.wikipedia.org/wiki/Field_(mathematics)). Операції додавання векторів і [скалярного множення](https://en.wikipedia.org/wiki/Scalar_multiplication) повинні задовольняти певним вимогам, які називаються векторними аксіомами. Терміни дійсний векторний простір і комплексний векторний простір часто використовуються для визначення природи скалярів:

* [Дійсний координатний простір](https://en.wikipedia.org/wiki/Real_coordinate_space) – множина [n -кортежів](https://en.wikipedia.org/wiki/Tuple) дійсних [чисел](https://en.wikipedia.org/wiki/Real_number), тобто множина всіх послідовностей n дійсних чисел
* [Комплексний координатний простір](https://en.wikipedia.org/wiki/Complex_coordinate_space) – це набір усіх упорядкованих n - [кортежів](https://en.wikipedia.org/wiki/Tuple) комплексних [чисел](https://en.wikipedia.org/wiki/Complex_number).

Векторні простори узагальнюють [евклідові вектори](https://en.wikipedia.org/wiki/Euclidean_vector), які дозволяють моделювати [фізичні величини](https://en.wikipedia.org/wiki/Physical_quantity), такі як [сила](https://en.wikipedia.org/wiki/Force) та [швидкість](https://en.wikipedia.org/wiki/Velocity), які мають не лише [величину](https://en.wikipedia.org/wiki/Magnitude_(mathematics)), але й [напрямок](https://en.wikipedia.org/wiki/Orientation_(geometry)).

У [математиці](https://en.wikipedia.org/wiki/Mathematics), [фізиці](https://en.wikipedia.org/wiki/Physics) та [інженерії](https://en.wikipedia.org/wiki/Engineering) евклідів вектор або просто вектор (іноді його називають геометричним вектором або просторовим вектором) — це геометричний об'єкт, який має [величину](https://en.wikipedia.org/wiki/Magnitude_(mathematics)) (або [довжину](https://en.wikipedia.org/wiki/Euclidean_norm)) і [напрямок](https://en.wikipedia.org/wiki/Direction_(geometry)). Вектори можна додавати до інших векторів відповідно до [векторної алгебри](https://en.wikipedia.org/wiki/Vector_algebra). Евклідів вектор часто зображають [напрямленим відрізком](https://en.wikipedia.org/wiki/Directed_line_segment) або графічно стрілкою, що сполучає початкову точку з кінцевою точкою.

Дескриптори — це вектори, які описують локальне оточення навколо ключових точок, присутніх на зображенні. Ці дескриптори використовуються для створення асоціацій між різними зображеннями.

Щоб отримати точний дескриптор, потрібно виключити обертання вектора ознак і залежність від освітлення. Залежність від обертання усувається шляхом визначення різниці між орієнтацією кожного градієнта та орієнтацією ключової точки. Подібним чином, залежність від освітлення усувається за допомогою порогового значення та нормалізації вектора дескриптора.

Створення дескрипторів є ключовим етапом в SIFT –  методі вилучення ознак,  який зменшує вміст зображення до набору точок, які використовуються для виявлення схожих моделей на інших зображеннях. Цей алгоритм зазвичай пов’язаний із програмами комп’ютерного зору, включаючи зіставлення зображень і виявлення об’єктів.

ЗГОРТКА

У [математиці](https://en.wikipedia.org/wiki/Mathematics) згортка — це [математична операція](https://en.wikipedia.org/wiki/Operation_(mathematics)) над двома [функціями](https://en.wikipedia.org/wiki/Function_(mathematics)), яка створює третю функцію, що вказує на те, як форма одного змінюється іншим. Термін згортка відноситься як до функції результату, так і до процесу її обчислення. Він визначається як [інтеграл](https://en.wikipedia.org/wiki/Integral) від добутку двох функцій після того, як одна з них відображена навколо осі Y і зміщена. Вибір того, яка функція відбивається та зсувається перед інтегралом, не змінює інтегральний результат. Інтеграл обчислюється для всіх значень зсуву, утворюючи функцію згортки.

Згортка може бути визначена для функцій на [евклідовому просторі](https://en.wikipedia.org/wiki/Euclidean_space) та інших [групах](https://en.wikipedia.org/wiki/Group_(mathematics)). Наприклад, [періодичні функції](https://en.wikipedia.org/wiki/Periodic_function), такі як [перетворення Фур’є з дискретним часом](https://en.wikipedia.org/wiki/Discrete-time_Fourier_transform), можна визначити на [колі](https://en.wikipedia.org/wiki/Circle) та згорнути за допомогою [періодичної згортки](https://en.wikipedia.org/wiki/Periodic_convolution). Дискретну згортку можна визначити для функцій на множині [цілих чисел](https://en.wikipedia.org/wiki/Integers).

Узагальнення згортки мають застосування в галузі [чисельного аналізу](https://en.wikipedia.org/wiki/Numerical_analysis) та [чисельної лінійної алгебри](https://en.wikipedia.org/wiki/Numerical_linear_algebra), а також у розробці та реалізації фільтрів [із кінцевою імпульсною характеристикою](https://en.wikipedia.org/wiki/Finite_impulse_response) в обробці сигналів.

Обчислення операції, [зворотної](https://en.wikipedia.org/wiki/Inverse_function) до операції згортання, називається [деконволюцією](https://en.wikipedia.org/wiki/Deconvolution).

Згортка та пов’язані з нею операції знаходять багато застосувань у науці, інженерії та математиці. Наприклад в:

* Обробці зображень: обробка цифрових зображень, оптика (розфокусування), програми обробки зображень (розмиття).
* Цифровій обробці даних: статистика (ковзне середнє).
* Обробці звуку: акустична реверберація.
* [Згорткових нейронних мереж](https://en.wikipedia.org/wiki/Convolutional_neural_networks)ах застосовують кілька каскадних ядер згортки із застосуваннями в [машинному зорі](https://en.wikipedia.org/wiki/Machine_vision) та [штучному інтелекті](https://en.wikipedia.org/wiki/Artificial_intelligence).
* Дробовому численні: визначення дробового інтеграла та дробової похідної.

ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ

В нейронних мережах згортка – це просте застосування фільтра до вхідних даних, що призводить до активації. Повторне застосування того самого фільтра до вхідних даних призводить до створення карти активацій, яка називається картою функцій, із зазначенням розташування та потужності виявленої функції у вхідних даних, наприклад зображення.

Згорткова нейронна мережа (CNN) також відома, я штучна нейрона мережа з інваріативним зсувом або з просторовою інваріативністю (SIANN)— це спеціалізований тип моделі нейронної мережі, розроблений для роботи з двовимірними даними зображення, хоча їх можна використовувати з одновимірними та тривимірними даними.

У контексті згорткової нейронної мережі згортка — це лінійна операція, яка передбачає множення набору вагових коефіцієнтів на вхідні дані.

Результатом одноразового множення фільтра на вхідний масив є одне значення. Оскільки фільтр застосовується кілька разів до вхідного масиву, результатом є двовимірний масив вихідних значень, які представляють фільтрацію вхідних даних. Таким чином, двовимірний вихідний масив від цієї операції називається «картою функцій».

Історично склалося так, що фільтри розроблялися вручну фахівцями з комп’ютерного зору, які потім застосовувалися до зображення, щоб створити карту функцій або результат застосування фільтра, який певним чином полегшує аналіз зображення.

Інновація використання операції згортки в нейронній мережі полягає в тому, що значення фільтра є ваговими коефіцієнтами, які потрібно вивчати під час навчання мережі.

Мережа дізнається, які типи функцій витягувати з вхідних даних. Зокрема, під час навчання під час стохастичного градієнтного спуску мережа змушена навчитися витягувати із зображення ознаки, які мінімізують втрати для конкретного завдання, якому навчається розв’язувати мережу, наприклад, вилучати функції, які є найбільш корисними для класифікації зображень.

Кількість фільтрів

Згорткові нейронні мережі не вивчають один фільтр; вони, фактично, вивчають кілька функцій паралельно для певного вхідного сигналу.

Наприклад, для згорткового рівня зазвичай вивчається від 32 до 512 фільтрів паралельно для одного входу.

Це дає моделі 32 або навіть 512 різних способів вилучення функцій із вхідних даних або багато різних способів як «навчання бачити», так і після навчання багато різних способів «бачити» вхідні дані.

Ця різноманітність дозволяє спеціалізувати, наприклад, не лише рядки, а конкретні лінії, які можна побачити в конкретних тренувальних даних.

Кількість каналів

Кольорові зображення мають кілька каналів, як правило, по одному для кожного колірного каналу, наприклад червоного, зеленого та синього.

З точки зору даних це означає, що одне зображення, надане як вхідні дані для моделі, насправді складається з трьох зображень.

Фільтр завжди повинен мати таку ж кількість каналів, як і вхід, що часто називають «глибиною». Якщо вхідне зображення має 3 канали, то фільтр, застосований до цього зображення, також повинен мати 3 канали. У цьому випадку фільтр 3×3 насправді буде 3x3x3 або [3, 3, 3] для рядків, стовпців і глибини. Незалежно від глибини входу та глибини фільтра, фільтр застосовується до входу за допомогою операції скалярного добутку, що призводить до єдиного значення.

Це означає, що якщо згортковий шар має 32 фільтри, ці 32 фільтри є не лише двовимірними для вхідного двовимірного зображення, але також є тривимірними, маючи певну вагу фільтра для кожного з трьох каналів. Тим не менш, кожен фільтр призводить до однієї карти функцій. Це означає, що глибина результату застосування згорткового шару з 32 фільтрами становить 32 для 32 створених карт функцій.

Кількість шарів

Згорткові шари застосовуються не лише до вхідних даних, наприклад необроблених значень пікселів, але також можуть застосовуватися до вихідних даних інших шарів.

Накопичування згорткових шарів дозволяє ієрархічно розкласти вхідні дані.

Наприклад, фільтри, які працюють безпосередньо з необробленими значеннями пікселів, навчаться виділяти низькорівневі функції, такі як лінії.

Фільтри, які діють на вихідних даних перших рядкових шарів, можуть витягувати об’єкти, які є комбінаціями об’єктів нижчого рівня, наприклад об’єкти, що містять кілька рядків для вираження форм. Цей процес триває до тих пір, поки дуже глибокі шари не вилучають обличчя, тварин або будинки.

Отже, суть шарів CNN – це абстракція ознак до вищих і вищих порядків у міру збільшення глибини мережі.

CNN використовують для:

* Розпізнавання зображень і відео.
* Рекомендаційних систем.
* Класифікації зображень.
* Сегментації зображень.
* Аналізу медичного зображення.
* Обробки природньої мови.
* Інтерфейсів типу мозок-комп’ютер.
* Фінансових часових рядів.

Окрім описаних вище фреймворків для роботи з нейронними мережами, існує бібліотека Caffe, яка розроблена спеціально для роботи з CNN.

ВИСНОВКИ

Отже, нейронна мережа, як і машинне навчання, є частиною концепції штучного інтелекту. Проте, машинне навчання є множиною, яка складається з різних типів, методів і підходів до навчання моделей, зокрема штучних нейронних мереж.

Тобто, штучна нейронна мережа і є тим, що зазвичай називають штучним інтелектом; виконавцем поставлених завдань.

Самі нейронні мережі, як і підходи до їх навчання, поділяються на багато типів і здатні виконувати не менше різноманітних завдань.

Проте, одним із найпопулярніших застосувань є робота з фото та відео. Стрімкий розвиток згорткових нейронних мереж спричинений розвитком технологій створення та збереження візуальних даних, що збільшує об’єм можливостей для застосування цих нейронних мереж для виконання індивідуальних і комерційних завдань.

Про популярність роботи з нейронними мережами, також, свідчить велика кількість фреймворків для їх навчання з використанням різних мов програмування. Найпопулярнішими мовами програмування для роботи з нейронними мережами є Python, через його простоту до вивчення, та Java, через можливості, які ця мова надає.

Підсумовуючи, робота з нейронними мережами і штучним інтелектом, в цілому, є популярною в сучасному світі і досі несе за собою велику кількість проблем, що означає необхідність в висококваліфікованих спеціалістах.