

НЕЙРОННІ МЕРЕЖІ ТА СУЧАСНІ ЗАСОБИ ЇХ ПРОЕКТУВАННЯ

В останні роки значно зріс інтерес до машинного навчання як складової галузі штучного інтелекту, яке досліджує та розробляє алгоритми, що наділяють комп'ютер здатністю навчатись на основі даних, «не будучи явно запрограмованими». Алгоритми будують модель з тренувального набору даних, яка здійснює передбачення на основі вхідних даних:

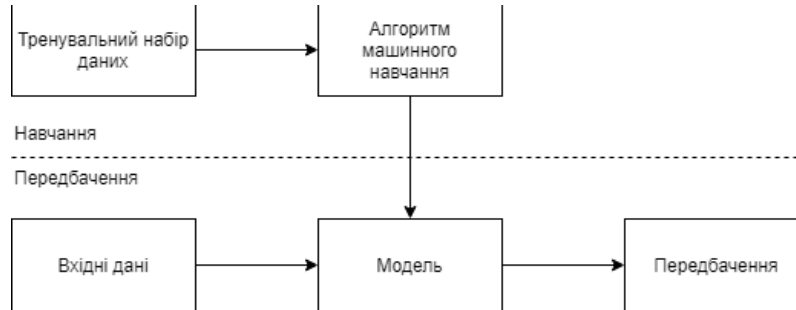


Рис. 1. Загальна схема роботи алгоритму машинного навчання

Виділяють три основні підходи машинного навчання:

- Навчання з учителем (supervised learning) – навчання з розміченим тренувальним набором:
 - Задачі регресії: вік людини за фото, ціна будинку за площею
 - Задачі класифікації: фільтр спаму, тип пухлини
- Навчання без учителя (unsupervised learning) – навчання з нерозміченим набором даних:
 - Задачі кластерування: групування генів за ознаками, які завідомо невідомі
 - Структуризація хаотичного середовища: розділення змішаних голосів, зображень
- Навчання з підкріпленням (reinforcement learning) – навчання на основі попередніх результатів для досягнення максимальної користі:

Задачі машинного навчання в свою чергу поділяють на такі типи:

- Задачі класифікації
- Задачі регресії
- Задачі кластерування
- Виявлення аномалій

Одним із розділів машинного навчання є глибоке навчання (Deep Learning). Глибоке навчання займається розробкою алгоритмів для навчання на основі ознак даних із використанням багатошарових графів, що отримали назву штучних нейронних мереж. Архітектури глибокого навчання використовуються в таких прикладних задачах:

- розпізнавання мовлення та обробка природної мови;
- комп'ютерний зір;
- механізми фільтрації контенту в соціальних мережах;
- машинний переклад;
- пошукові алгоритми;
- біоінформатика;
- розробка ліків тощо.

Розглянемо особливості використання штучних нейронних мереж та окреслимо сучасні проблеми в їх проектуванні. Штучна нейронна мережа – математична модель, принцип роботи якої нагадує роботу мережі біологічних нейронів.

Кожен шар представлений множиною вузлів – штучних нейронів, які видобувають ознаки все більшого рівня, поки останній шар не скомбінує ці ознаки, щоб зробити передбачення:

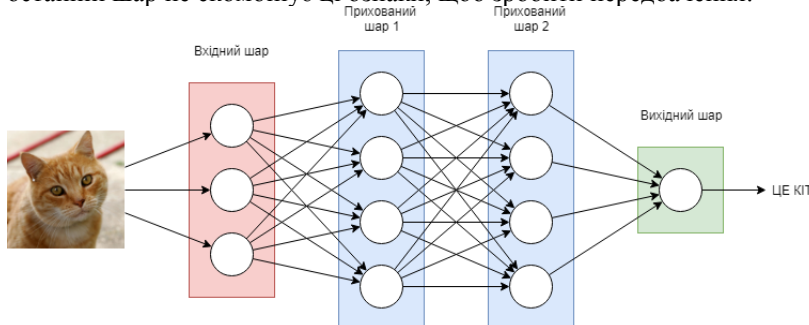


Рис. 2. Будова багатошарової нейронної мережі

Штучний нейрон – це модель, яка складається із шару вхідних сигналів $x_1, x_2, x_3, \dots, x_n$, кожен із яких має вагу $w_1, w_2, w_3, \dots, w_n$. Ваги відображають значимість кожного входу штучного нейрону. На основі вхідних активацій та їх ваг у вихідний шар обчислюється сумуюча функція, яка враховуючи вхідні сигнали та їх ваги обчислює деяке

значення. Далі результат сумуючої функції передається в активаційну функцію, яка в залежності від значення сумуючої функції обчислює вихідний результат нейрона.

Одним із спрощених моделей штучного нейрона є перцептрон. Він складається із вхідного та вихідного шару. На вхідному шарі приймається вектор вхідних сигналів X , кожен з яких має відповідну вагу. Активаційна функція обчислює результуюче значення нейрона, що передається далі:

$$f(x, w, b) = \begin{cases} 1 & \text{якщо } \sum_{i=1}^n w_i x_i + b > 0 \\ 0 & \text{якщо } \sum_{i=1}^n w_i x_i + b \leq 0 \end{cases}$$

де x – вектор вхідних активацій, w – вектор ваг, b – зміщення (bias), $f(x, w, b)$ – активаційна функція.

Розглянемо навчальний алгоритм для перцептрона:

```
1. perfect = False
2. while NOT perfect:
3.     perfect = True
4.     for e in examples:
5.         if Predict(e) != Target(e):
6.             perfect = False
7.             if Predict(e) == 0:
8.                 w = w + e
9.             if Predict(e) == 1:
10.                w = w - e
```

Покладаємо, що спочатку класифікація не ідеальна та здійснюємо цикл поки виконується умова такої неідеальності (поки змінна `perfect` рівна `False`). Для всіх значень прикладів масиву `examples` перевіряємо, якщо передбачене значення для даного прикладу не рівне цільовому значенню, то класифікація не ідеальна і алгоритм продовжує свій цикл. В цьому ж блоці, якщо невірно передбачене значення рівне 0, то до ваги додаємо значення поточного прикладу. Якщо ж невірно передбачене значення рівне 1, то відповідно від ваги віднімаємо значення поточного прикладу.

Скорочено даний алгоритм можна записати у вигляді:

$$\Delta w_i = y - y \cdot x_i$$

Де Δw_i – зміна i -тої ваги після поточного прикладу, y – правильна вихідна активація перцептрона для даного прикладу (може бути рівна 0 або 1), y – поточна вихідна активація перцептрона для даного прикладу (може бути рівна 0 або 1), x_i – i -тий вхід на поточному прикладі.

Отже, завданням штучного нейрона є прийом вхідних сигналів, їх обробка та передача результату, що генерується активаційною функцією. Багаторівневі мережі, утворені зі штучних нейронів, здатні розв'язувати достатньо складні задачі.

Однією із необхідних умов ефективної роботи нейронної мережі є її паралельність обчислень. Послідовні обчислення центрального процесора персонального комп'ютера значно сповільнюють роботу нейронної мережі, особливо багатозарової з великою кількістю нейронів. Тому слід розглянути сучасні підходи при розробці нейронних мереж (як правило, дані підходи поєднуються):

- використання апаратних розподілених систем – нейрокомп'ютерів, графічних процесорів;
- створення паралельних обчислювальних систем на базі персональних комп'ютерів;
- використання хмарних сервісів

Широке використання графічних процесорів для паралельних обчислень та створення відповідних додатків стало можливе завдяки технології CUDA – платформі для паралельних обчислень на графічних процесорах від Nvidia. Для реалізації задач глибинного навчання розроблена бібліотека cuDNN, що забезпечує такі стандартні процедури нейронних мереж як пряме поширення, метод зворотного поширення помилки, об'єднуючий (pooling), нормуючий (normalization) та активаційний (activation) шари. Дана бібліотека підтримує такі фреймворки як TensorFlow, Caffe2, MATLAB, Microsoft Cognitive Toolkit, Theano, PyTorch.

Використання графічних процесорів з підтримкою технології CUDA для проведення досліджень з паралельних обчислень та зокрема глибинного навчання є хорошим рішенням. Проте доцільно звернути увагу на альтернативну можливість, наприклад, використання хмарних сервісів. Лідерами в напрямку хмарних технологій, зокрема і в проектуванні нейронних мереж є Google Cloud Platform, AWS, Nvidia GPU Cloud.

Окреслимо основні тенденції розвитку машинного навчання в найближчому майбутньому:

- прориви в підходах до навчання без учителя;
- складніші архітектурні рішення, що базуються на різних взаємозамінних модулях;
- глибші моделі, що навчаються на меншій кількості прикладів;
- вирішення більш складних задач, таких як розуміння відео та обробка природньої мови

Література

1. Andrew NG. Learn Machine Learning. <https://www.coursera.org/learn/machine-learning>
2. Pankaj Mehra, Benjamin W. Wah. Artificial Neural Networks: Concepts and Theory. IEEE Computer Soc. Press, 1992
3. Adit Deshpande. A Beginner's Guide To Understanding Convolutional Neural Networks. <https://adeshpande3.github.io/>
4. Krizhevsky, G. Hinton, I. Sutskever. Imagenet classification with deep convolutional neural networks.
5. Y. LeCun. Gradient-based learning applied to document recognition