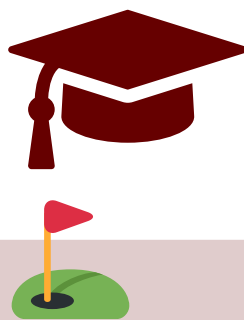




# Машинне навчання

11 лютого 2024 р.



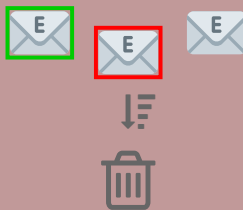
## Штучний інтелект

Будь-яка техніка, яка дозволяє комп'ютерам імітувати поведінку людини



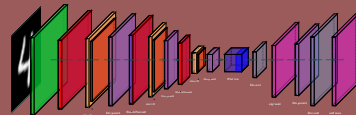
## Машинне навчання

Можливість комп'ютера навчитися не будучи явно запрограмованим



## Глибинне навчання

Пошук шаблону в даних за допомогою нейронних мереж



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Кочура Ю. П., Гордієнко Ю. Г.

# Машинне навчання

## Занурення в машинне навчання

Навчальний посібник

Рекомендовано для здобувачів ступеня “магістр”, які навчаються за освітніми програмами «Комп’ютерні системи та мережі» спеціальність 123 «Комп’ютерна інженерія» та «Інженерія програмного забезпечення комп’ютерних систем» спеціальність 121 «Інженерія програмного забезпечення»

Електронне видання

Навчальний посібник перебуває на етапі написання. Остання доступна версія цього документа за QR-кодом:



Ми були б дуже вдячні за надану допомогу та пропозиції щодо наповнення та покращення цього документа. Повідомлення про проблеми надсилайте через github: <https://github.com/dml-book/dml/issues>

Чернетка. *Версія 0.1.1*

# Зміст

<b>1</b>	<b>Вступ</b>	<b>3</b>
1.1	Що таке машинне навчання? . . . . .	3
1.1.1	Навчання з учителем . . . . .	4
1.2	Мотивація та інтуїція . . . . .	5
1.2.1	Інжиніринг ознак . . . . .	6

# Розділ 1

## Вступ

*“Теоретично між теорією та практикою нема ніякої різниці. Але на практиці вона є.”*

– Бенджамін Брюстер

### 1.1 Що таке машинне навчання?

**Машинне навчання (Machine Learning, ML)** – це галузь штучного інтелекту, що вивчає розробку та застосування алгоритмів, які дозволяють комп’ютеру вчитися робити прогнози чи приймати рішення на основі вхідних даних не будучи при цьому явно запрограмованим. Ці дані можуть бути отримані з реального середовища за допомогою сенсорів, створені вручну або згенеровані іншим алгоритмом.

Машинне навчання також можна розглядати як процес вирішення деякого практичного завдання шляхом алгоритмічного навчання **статистичної моделі** на наборі даних, що характеризує шаблон поставленого завдання.

Комп’ютери та обчислення допомагають нам досягати більш складних цілей і кращих результатів у вирішенні проблем, ніж ми могли б досягти самі. Однак, багато сучасних завдань вийшли за рамки обчислень через один основний обмежуючий фактор: традиційно, комп’ютери можуть дотримуватися лише конкретних вказівок/інструкцій, які їм дають.

Вирішення проблем з програмування вимагає написання конкретних покрокових інструкцій, які має виконувати комп’ютер. Ми називаємо ці кроки алгоритмами. У цьому випадку, комп’ютери можуть допомогти нам там, де ми:

1. Розуміємо як вирішити проблему.
2. Можемо описати проблему за допомогою чітких покрокових інструкцій, які комп’ютер може зрозуміти.

Методи контрольованого машинного навчання дозволяють комп’ютерам “учитися” на прикладах. Вирішення проблем із застосуванням машинного навчання вимагає виявлення деякого шаблону<sup>1</sup>, а потім, коли такий шаблон готовий, дозволяють, наприклад, нейронній мережі вивчити карту переходів між вхідними та вихідними даними. Ця особливість відкриває нові типи проблем, де комп’ютери можуть допомогти нам у їх розв’язанні, за умови, коли ми:

1. Визначили шаблон проблеми.
2. Маємо достатньо даних, що ілюструють даний шаблон.

---

<sup>1</sup>Пошук прикладів, що висвітлюють обидві сторони шаблону: вхід і вихід.

Принципова відмінність парадигм: класичного програмування (символьний штучний інтелект) та машинного навчання, показана на рисунку 1.1. Таким чином, для вирішення задачі за допомогою класичного програмування нам потрібно прописати чіткі покрокові інструкції нашого алгоритму. Далі, коли алгоритм реалізовано, ми можемо подати йому на вхід деякі дані, що характеризують окремий стан поставленої задачі, а на виході він згенерує нам деякі вихідні дані (відповіді). Коли мова йде про контрольоване машинне навчання, тут на вхід комп'ютерній програмі подаються приклади, що висвітлюють обидві сторони шаблону поставленої задачі: вхідні та очікувані вихідні дані (відповіді), а на виході отримуємо правило (алгоритм) за яким буде здійснюватись перехід між вхідними та вихідними даними.



**Рис. 1.1:** Відмінність класичного програмування від машинного навчання.

Для стислості, далі будуть використовуватись терміни **навчання** та **машинне навчання** як синоніми. З тієї ж причини часто буде використовуватись термін **модель**, під яким будемо мати на увазі **статистичну модель**.

За характером навчальних даних навчання буває з учителем (контрольоване), без учителя (неконтрольоване), з частковим залученням учителя (напівконтрольоване) та з підкріпленням.

### 1.1.1 Навчання з учителем

У цьому випадку ми працюємо з множиною даних, яку представлено наступним чином:

$$\{(\mathbf{X}^{(1)}, y^{(1)}), (\mathbf{X}^{(2)}, y^{(2)}), \dots, (\mathbf{X}^{(n)}, y^{(n)})\}, \quad (1.1)$$

де  $n$  – загальна кількість прикладів у наборі,  $y^{(i)}$  – мітка  $i$ -го прикладу, як правило,  $y^{(i)} \in \mathbb{R}$  або  $y^{(i)} \in \mathbb{N}$ .

Кожен елемент  $\mathbf{X}^{(i)}$  множини (1.1) називається вектором вхідних ознак або прикладом. Вектор – це одновимірний масив. Одновимірний масив, у свою чергу, є упорядкованою та проіндексованою послідовністю значень. Довжина цієї послідовності,  $D$ , називається розмірністю вектора:  $\mathbf{X}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_D^{(i)})$ .

Вектор ознак – це вектор, у якому кожен елемент  $j = 1, \dots, D$  містить значення, що характеризує приклад. Кожне таке значення називається ознакою та позначається  $x_j$ . Запис  $x_j^{(i)}$  означає, що ми розглядаємо ознаку  $j$  для  $i$ -го прикладу.





Контрольоване навчання <b>Supervised learning</b>	Напівконтрольоване навчання <b>Semi-supervised learning</b>	Неконтрольоване навчання <b>Unsupervised learning</b>	Навчання з підкріплення <b>Reinforcement learning</b>
Дані: $(X, y)$ $X$ – приклад, $y$ – мітка	Дані: $(X, y)$ та $X,  X, y  <  X $ $X$ – приклад, $y$ – мітка	Дані: $X$ $X$ – приклад, немає міток!	Дані: пари стан-дія
Мета – знайти функцію відображення $X \rightarrow y$	Мета – знайти функцію відображення або категорію $X \rightarrow y$	Мета – знайти правильну категорію.	Мета – максимізація загальної винагороди, отриманої агентом при взаємодії з навколишнім середовищем.
Приклад  Це є яблуко.	Приклад  Це є яблуко.	Приклад  Цей об'єкт схожий на інший.	Приклад  Їжте це, бо це зробить вас сильнішим.

Рис. 1.2: Види навчання.

Мітка  $y^{(i)}$  може бути або елементом кінцевої множини класів  $\{1, 2, \dots, C\}$  або дійсним числом, або складнішою структурою, такою як вектор, матриця, дерево чи граф. У задачах класифікації, клас – категорія даних, до якої належить розглянутий приклад. Скажімо, якщо прикладами є повідомлення електронної пошти, а наше завдання полягає у виявленні спаму, тобто усі повідомлення потрібно розділити на два класи: спам, не спам.

Завдання передбачення класу називається класифікацією, а завдання передбачення дійсного числа називається регресією.

## 1.2 Мотивація та інтуїція

Для більшості задач машинного навчання вибір хороших ознак<sup>2</sup> має першочергове значення. Навіть найкращий алгоритм не зможе продемонструвати хороших результатів, якщо для його навчання було використано погані ознаки. З іншого боку, вибір ознак часто є вкрай нетривіальним завданням. Наприклад, розглянемо кольорове зображення обличчя людини. Нехай у цьому випадку зображення обличчя людини буде представлено інтенсивністю пікселя для трьох кольорів: червоного, зеленого та синього (якщо використовується RGB-кодування). Тому, 1М піксельне кольорове зображення буде мати 3М ознак за якими ми можемо навчати нашу модель. З іншого боку, вираз обличчя людини, ймовірно, можна охарактеризувати  $\leq 56$  ознаками (на обличчі людини є  $\sim 56$  м'язів). Отже, для ідентифікації конкретних виразів обличчя, дані великої розмірності можуть бути представлені відповідними ознаками меншої розмірності.

Таким чином, ідеальний екстрактор (видобувач) ознак буде приймати на вхід зображення обличчя, а на виході видавати видобуті ознаки, що характеризують вираз обличчя людини. Однак, на теперішній час, не існує ідеальних способів як це зробити. У традиційних методах розпізнавання образів, які були розроблені починаючи з 50-х років, ці екстрактори ознак були жорстко закодовані на основі суб'єктивної інтуїції дослідників. Основна ідея, яка прийшла до нас разом з нейронними мережами, полягає в тому, що хороші ознаки можуть бути вивчені мережею безпосередньо з даних, таким чином більше непотрібно досліднику видобувати їх вручну.

З іншого боку, базова модель слабо розвинулася з часу її створення починаючи з 1950-х років. Перша машина («Марк-1», Френк Розенблат в 1957 р.), яка реалізовувала алгоритм перцептрона (нейрон Маккалока-Пітса) була лінійним класифікатором, побудованим поверх простого жорстко прописаного екстрактора ознак. До сьогодення на практиці для вирішення прикладних задач машинного навчання використовують ручне видобування ознак.

<sup>2</sup>Ознака (фіча) – це окрема властивість чи характеристика у даних, від якої безпосередньо залежить вихідний результат передбачення моделі.

### 1.2.1 Інжиніринг ознак

Інжиніринг (конструювання) ознак є дуже важливим етапом для створення моделі. Він передбачає видобування та вибір ознак. Під час видобування ознак витягуються з даних усі ознаки, які характеризують поставлену задачу. Під час вибору – визначаються усі найбільш важливі ознаки з метою покращення продуктивності моделі.

На рисунку 1.3 розглянуто приклад класифікації зображень. Ручне вилучення ознак з даних вимагає глибоких знань як задачі, яка вирішується, так і предметної галузі. Крім того, цей спосіб є трудомістким. Ми можемо автоматизувати процес конструювання ознак за допомогою глибокого навчання!

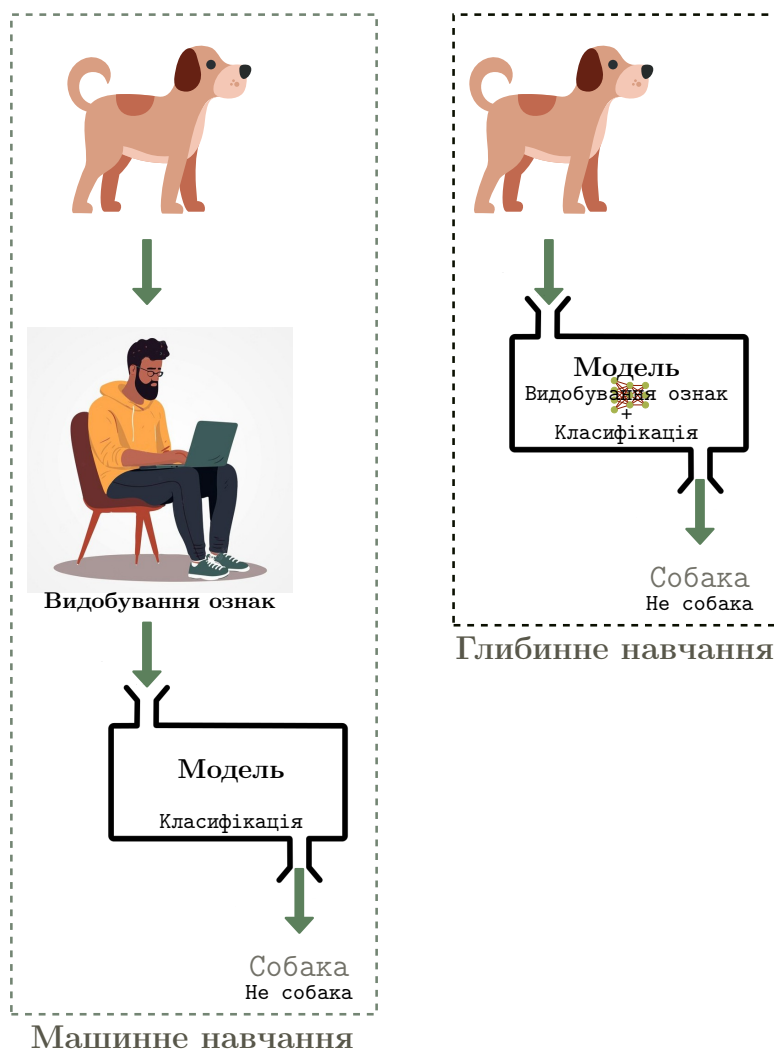


Рис. 1.3: Інжиніринг ознак в машинному навчанні та глибинному навчанні