



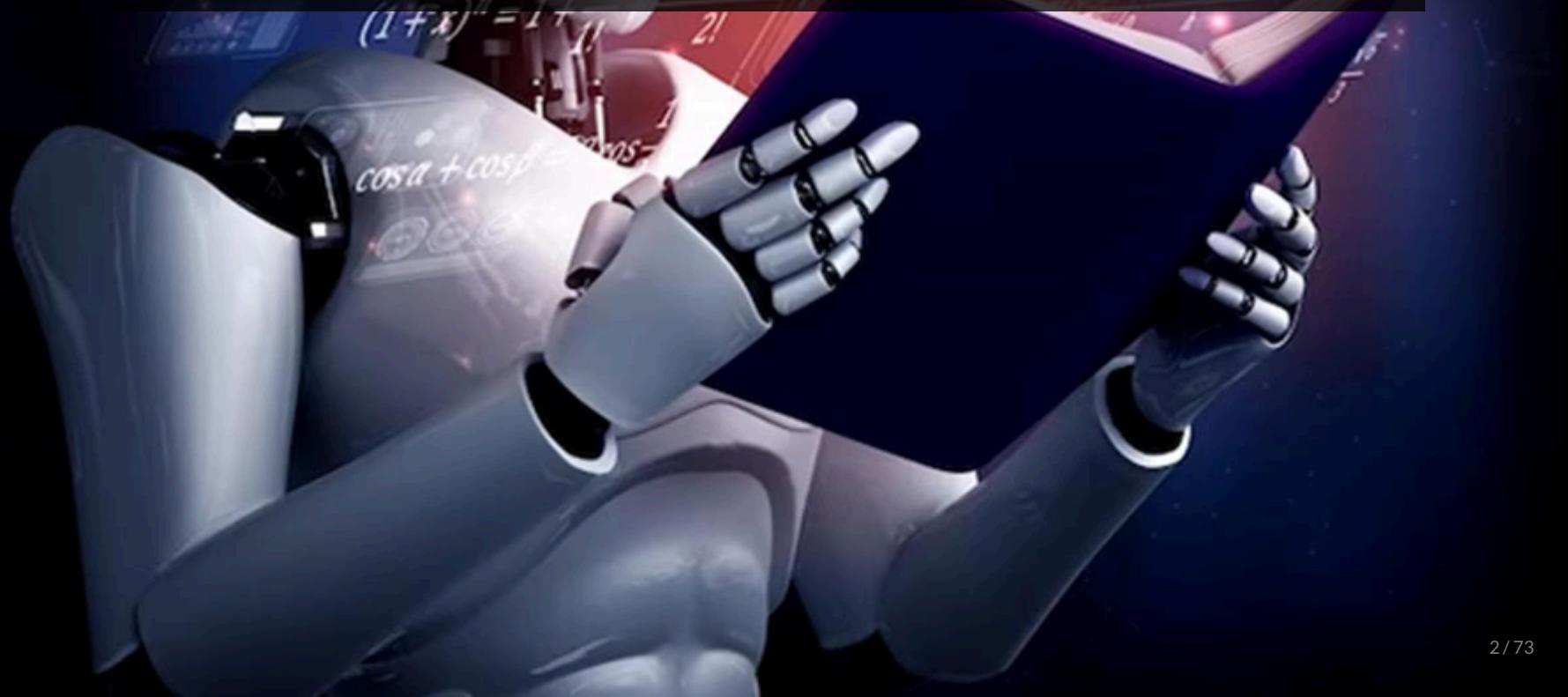
# Машинне навчання

Лекція 1: Вступ до машинного навчання

Кочура Юрій Петрович  
[iuriy.kochura@gmail.com](mailto:iuriy.kochura@gmail.com)  
[@y\\_kochura](https://twitter.com/y_kochura)

# Сьогодні

- Інтелект vs штучний інтелект (ШІ)
- Машинне навчання
- Сфери застосування та успіхи ШІ



Інтелект  
vs  
Штучний інтелект

# Що таке інтелект?

- Інтелект – це про здатність

навчатися приймати рішення для досягнення цілей

- Навчання, прийняття рішення, та цілі є ключовими

# Що таке штучний інтелект?

- У широкому сенсі

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

# Що таке штучний інтелект?

- У вузькому сенсі

Штучний інтелект — здатність інженерної системи обробляти, застосовувати та вдосконалювати здобуті знання та вміння.

- **Знання** — це факти, інформація та навички, набуті через досвід або навчання.

# Що таке штучний інтелект?



"Artificial intelligence is the science of making machines do things that would require intelligence if done by men." -- Marvin Minsky, 1968.

# Чи можуть машини думати?

## I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

### 1. *The Imitation Game.*

I PROPOSE to consider the question, ‘Can machines think?’

— Alan Turing, 1950



Image source: [biography](#)

*Намагаючись зімітувати розум дорослої людини, ми зобов'язані добре подумати про процес, який привів його до стану, в якому він перебуває. Ми можемо помітити три компоненти,*

*a. Початковий стан розуму, скажімо, при народженні,*

*b. Навчання, якому ми були піддані,*

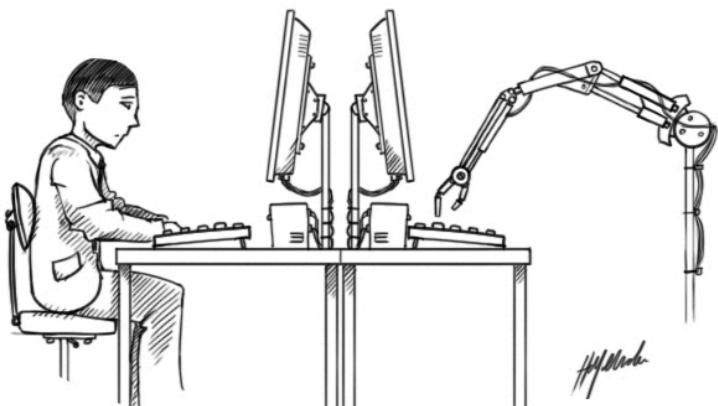
*c. Інший досвід, який не можна назвати навчанням, якому ми були піддані.*

*Замість того, щоб намагатися створити програму, яка моделює розум дорослої людини, чому б не спробувати створити таку, яка моделює розум дитини? Потім, якби це було піддано належному курсу навчання, можна було б отримати модель мозку дорослої людини. Ймовірно, дитячий мозок – це щось на зразок чистого зошита, який купуємо у канцтоварах. Досить маленький механізм і багато чистих аркушів. (Механізм і запис з нашої точки зору майже синоніми.) Ми сподіваємось, що в дитячому мозку настільки мало механізмів, що щось подібне можна легко запрограмувати.*

— Алан Тюрінг, 1950

## Тест Тюрінга

Вважається, що комп'ютер проходить тест Тюрінга («Гру в імітацію»), якщо людина, отримуючи письмові відповіді на свої запитання, не може встановити, хто їх автор — людина чи комп'ютер.



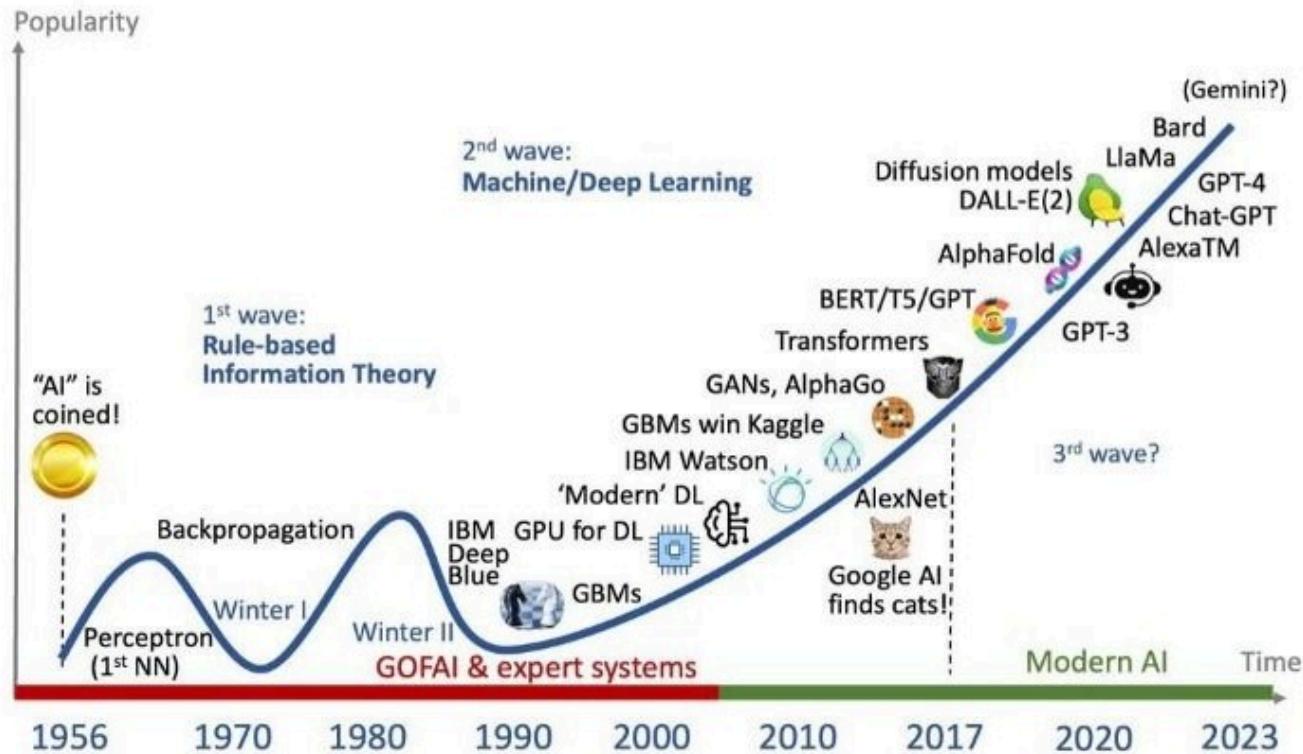
Чи можуть машини думати?  
(Алан Тюрінг, 1950)

Агент не зможе пройти тест Тюрінга без таких вимог:

- обробка природної мови
- представлення знань
- автоматизоване міркування
- машинне навчання
- комп'ютерний зір (для повного тесту Тюрінга)
- робототехніка (для повного тесту Тюрінга)

Незважаючи на те, що тест було запропоновано майже 75 років тому, він залишається актуальним і сьогодні.

## Коротка історія



# Коротка історія

- **1940–1952:** Ранні дні

- 1943: Мак-Каллок & Піттс: Булева модель нейрона
- 1950: Праця Тюрінга «Обчислювальні машини та інтелект»

- **1952–1956:** Народження ШІ

- 1950s: Ранні програми ШІ, включаючи програму Семюеля для гри в шашки
- 1956: Дартмутський семінар: запропоновано термін "Штучний інтелект"

- **1956–1974:** Золоті роки

- 1958: Френк Розенблatt винайшов **перцептрон** (проста нейронна мережа)
- 1964: Програма Боброу [STUDENT]  
[https://en.wikipedia.org/wiki/STUDENT\\_\(computer\\_program\)](https://en.wikipedia.org/wiki/STUDENT_(computer_program)), яка вирішує текстові задачі з алгебри

- **1974–1980:** Перша зима ШІ

- **1980–1987:** Бум експертних систем

- **1987–1993:** Провали експертних систем: друга зима ШІ

- **1993–2011:** Статистичні підходи

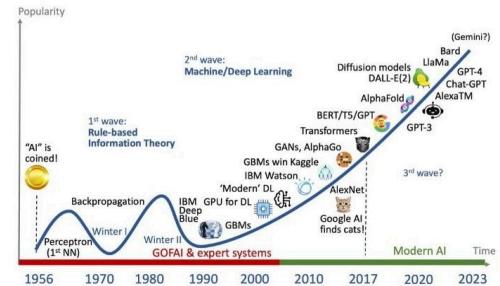
- Відродження ймовірнісних підходів, зосередженість на невизначеності
- Розумні агенти

- **2011– 2020:** Глибоке навчання, Великі дані

- Великі дані, велике обчислення, нейронні мережі

Credits: Wikipedia - History of artificial intelligence

- **2020– по теперішній час:** Ера ШІ, сильний штучний інтелект



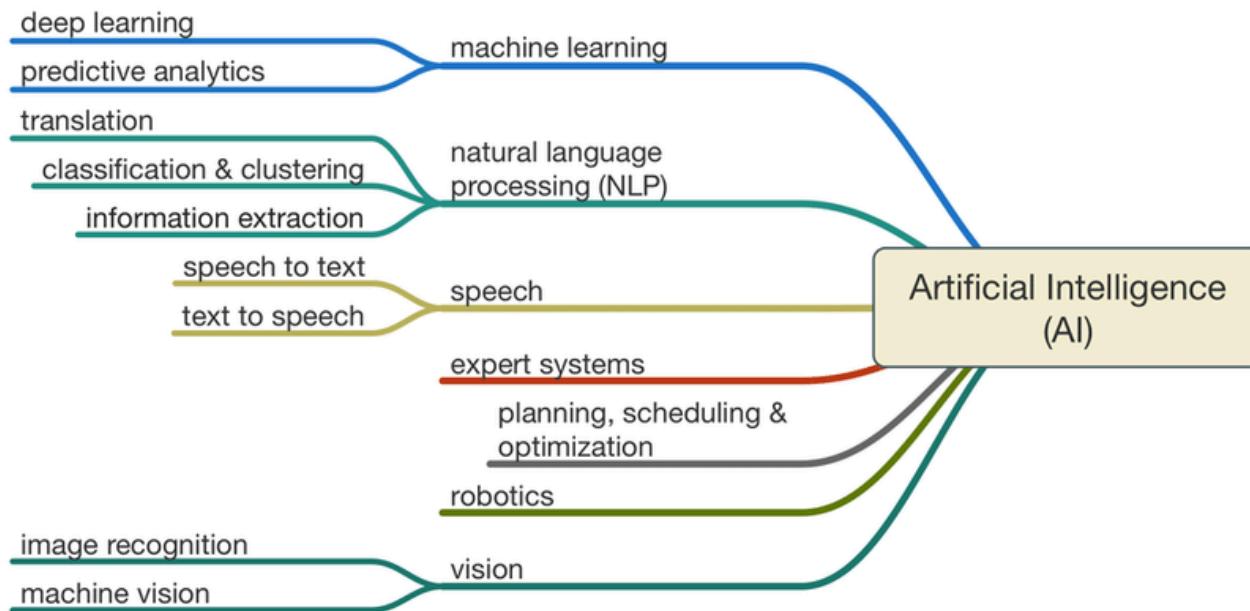


## Дартмутський семінар (1956)

Організували Марвін Мінський, Джон Маккарті, Клод Шенон та Натан Рочестер

*Будь-який аспект навчання або будь-яку іншу ознаку інтелекту можна описати так точно, що можливо буде зробити машину, яка її імітуватиме.*

# AI – багата галузь



# SUMMARY OF ML/AI CAPABILITIES

## USE CASES

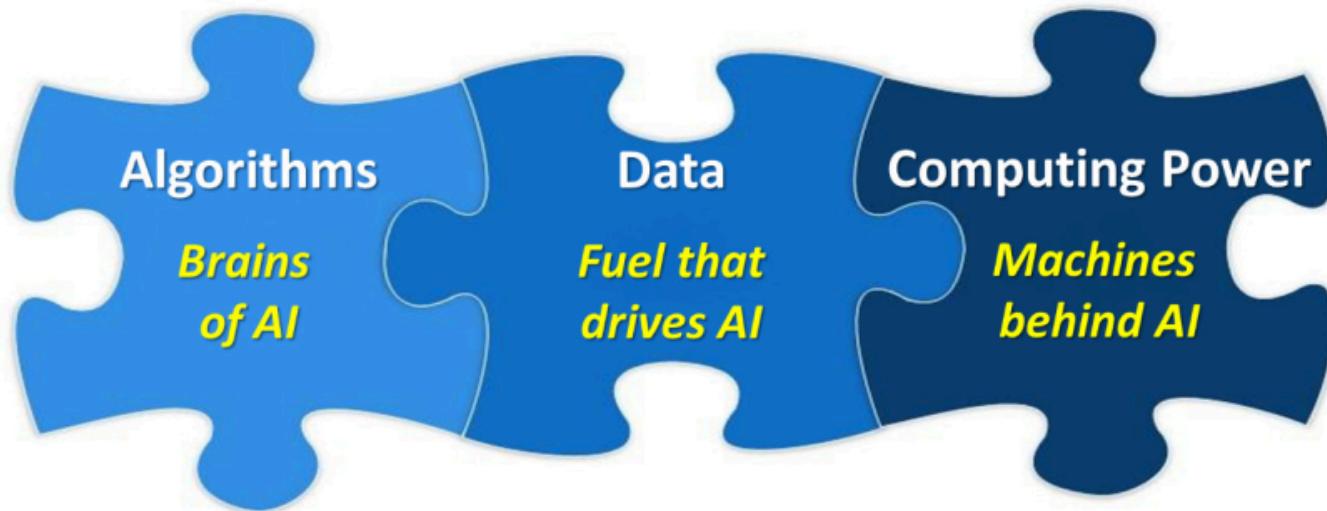
### CAPABILITIES

PERCEPTION (interpreting the world)	VISION understanding images	AUDIO audio recognition	SPEECH • text-to-speech • Speech-to-text conversions	NATURAL LANGUAGE understanding & generating text
COGNITION (reasoning on top of data)	REGRESSION • predicting a numerical value	CLASSIFICATION • predicting a category for a data point	PATTERN RECOGNITION • identifying relevant insights on data	
	PLANNING • determining the best sequence of steps for a goal	OPTIMISATION • identifying the most optimal parameters.	RECOMMENDATION • predicting user's preferences	
LEARNING (types of ML/AI)	SUPERVISED • learning on labelled data pairs: (input, output)	UNSUPERVISED • inferring hidden structures in an unlabelled data	REINFORCEMENT LEARNING • learning by experimenting • maximizing reward	



"Подібно до того, як електрика змінила майже все 100 років тому, сьогодні мені важко уявити галузь, яку, на мою думку, ІІ не змінить у найближчі кілька років."

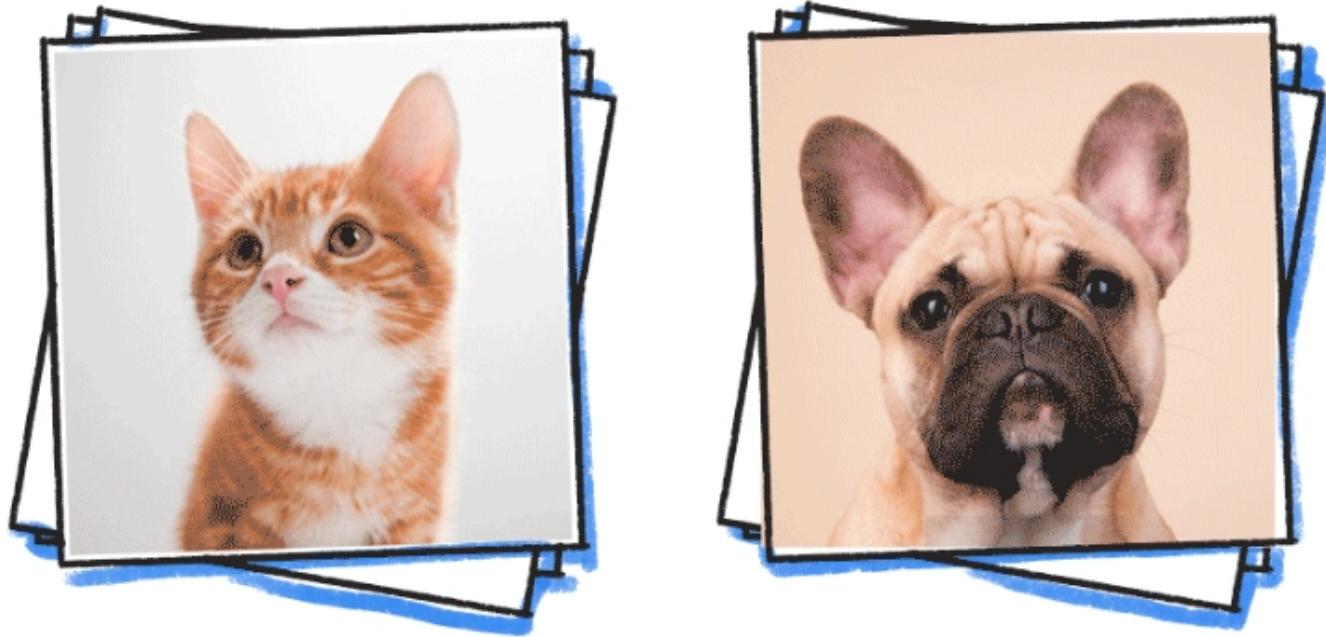
— Ендрю Ин, 2017



Однією з найважливіших особливостей штучного інтелекту є те, що це багатоцільова технологія. Подібно до електрики, її можна застосовувати у різних сферах та для різних завдань.

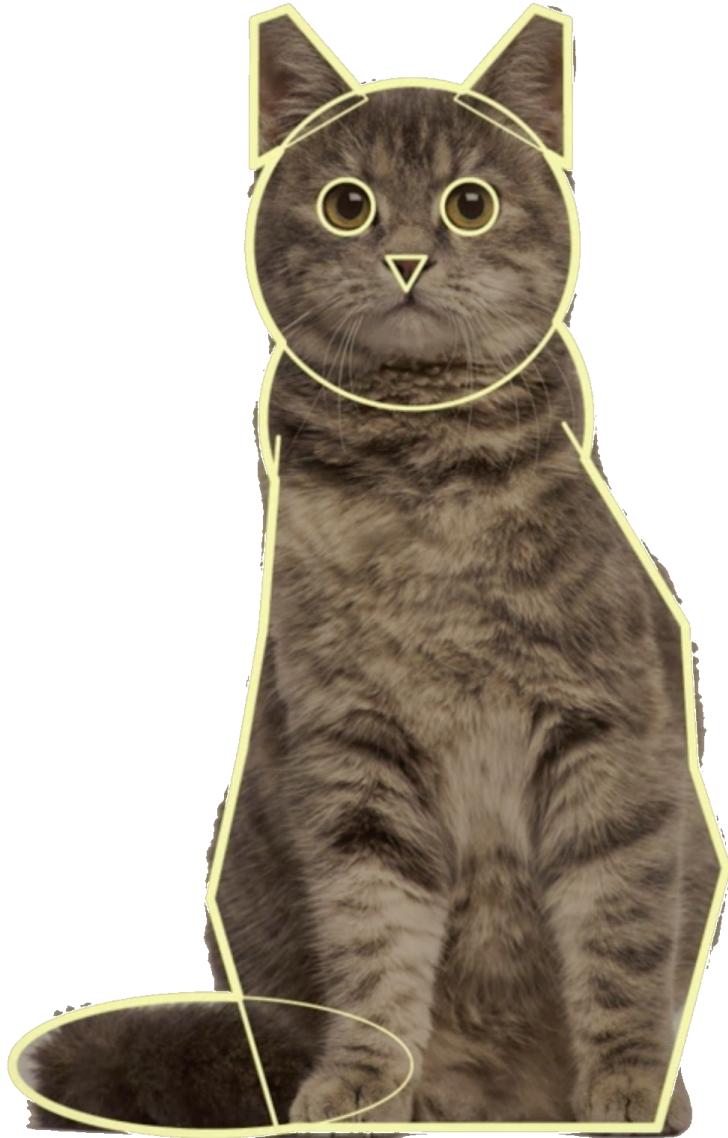
**Алгоритми** вказують комп’ютерам, що робити. **Дані** визначають, що комп’ютери мають вчити. **Обчислювальна потужність** надає машинам змогу навчатися та ухвалювати рішення.

# Машинне навчання

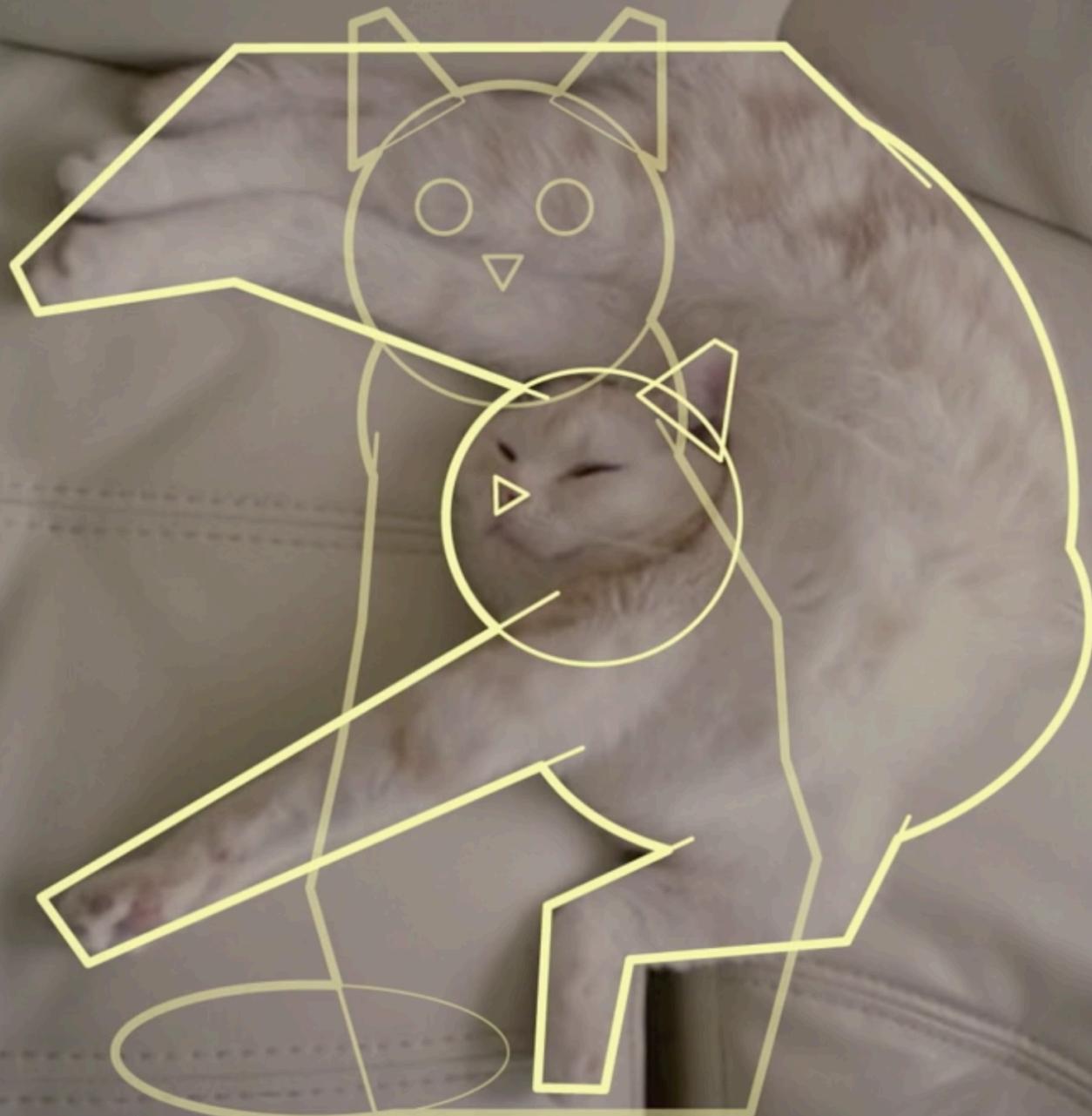


Чи могли б ви написати комп'ютерну програму, яка розпізнає **котів** та **собак**?







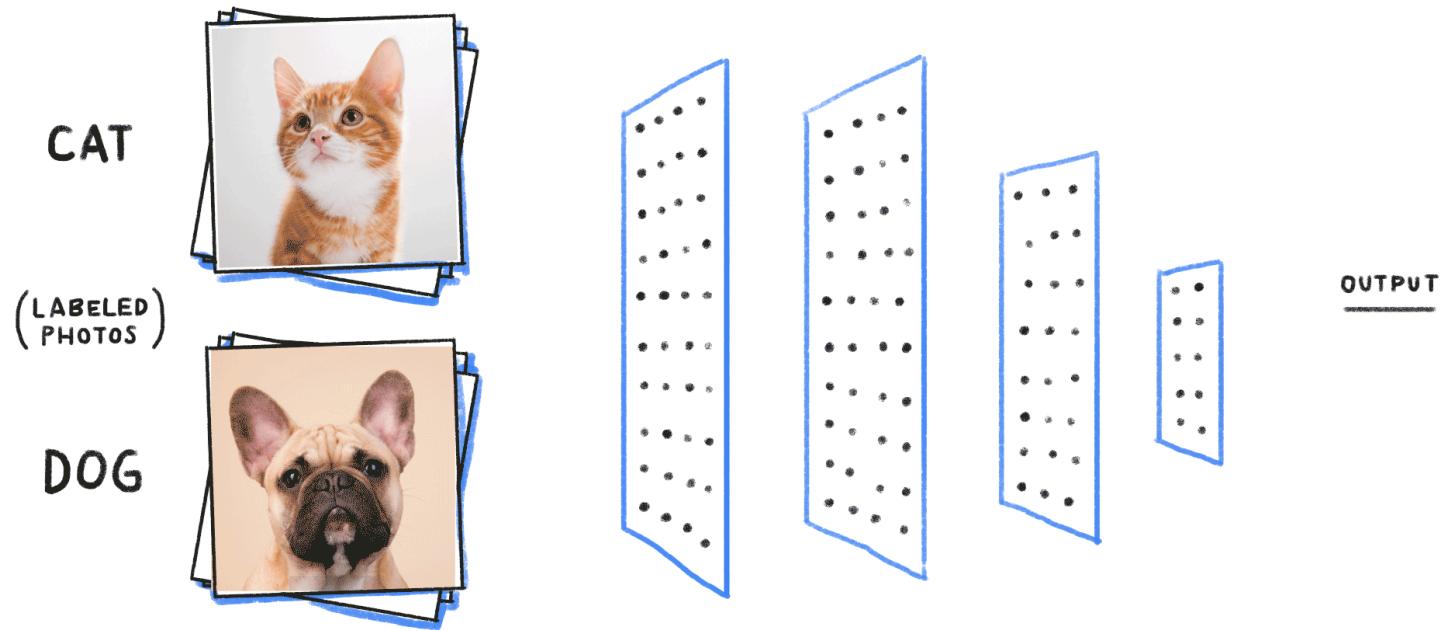




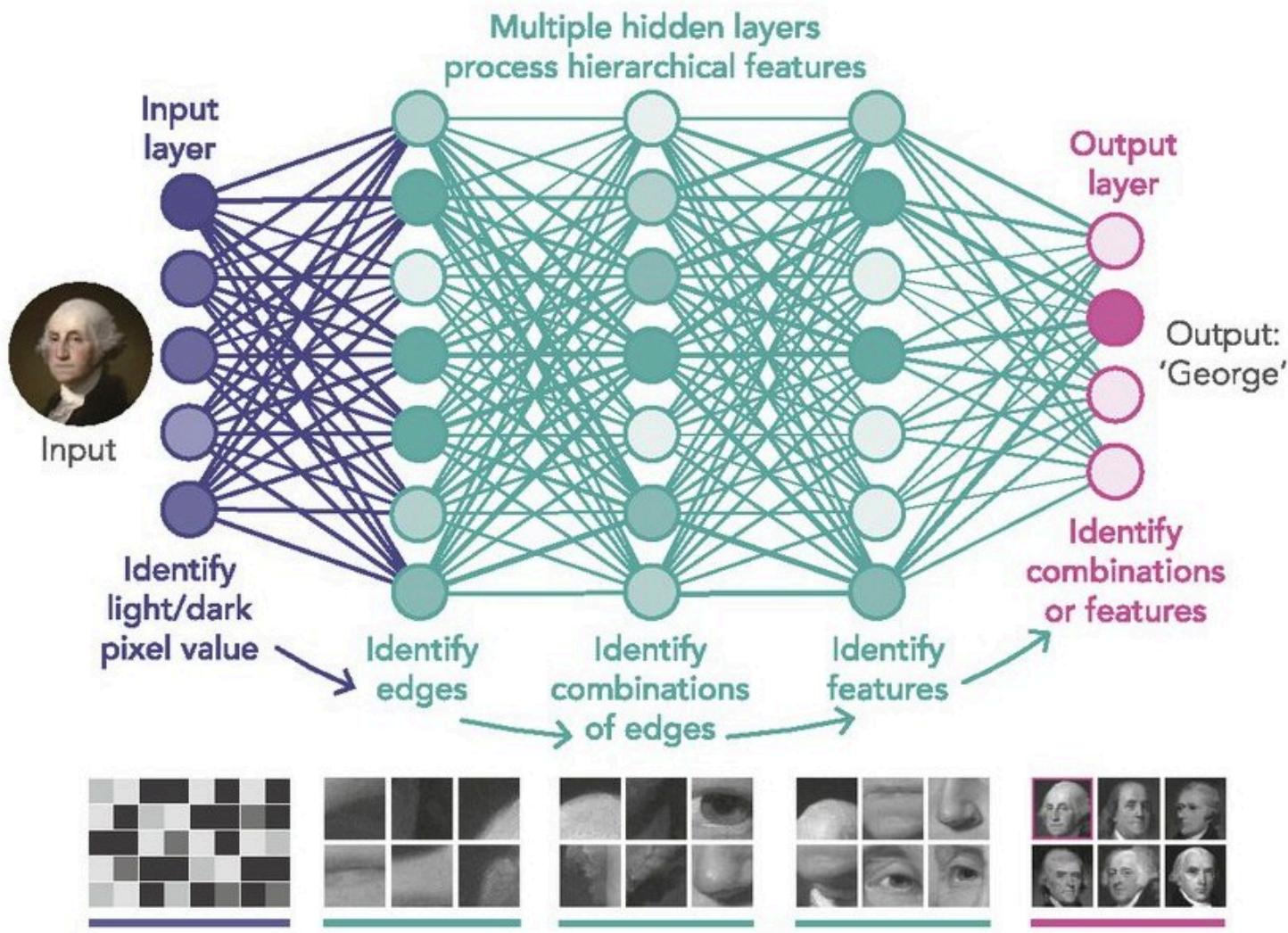


Для пошуку шаблону в даних (витягування семантичної інформації, ознак) потрібна побудова **складних моделей**, які б отримати вручну було б дуже складно.

Однак, можна використати алгоритми машинного навчання, які будуть **учитись** знаходити шаблон у даних самостійно.



Підхід глибокого навчання



# Що таке машинне навчання?

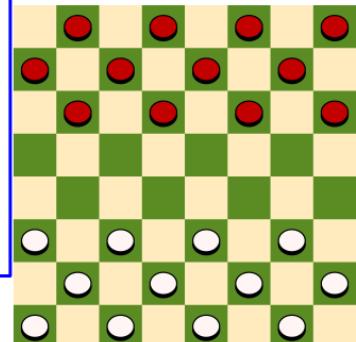
# Визначення за Артур Семюель

Артур Семюель (1959): Машинне навчання - це область навчання, яка надає комп'ютеру можливість вчитися не будучи явно запrogramованим.



A. L. Samuel\*

**Some Studies in Machine Learning  
Using the Game of Checkers. II—Recent Progress**



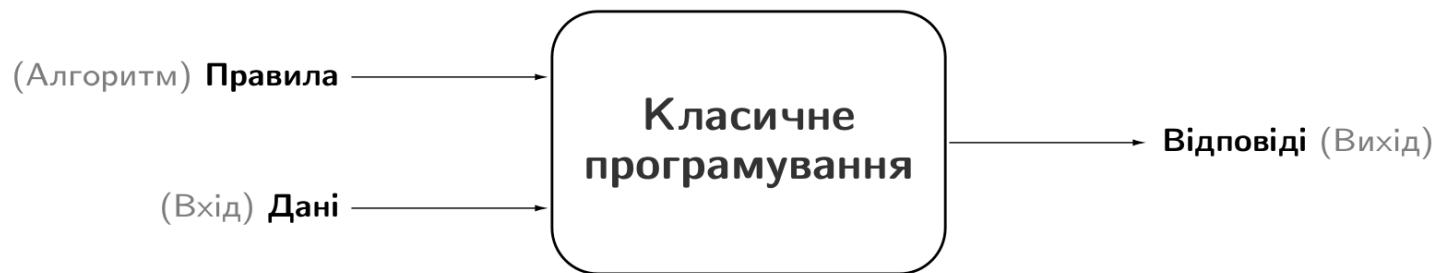
# Визначення за Том Мітчелл

Том Мітчелл (1998): Комп'ютерна програма, яка учається з досвіду  $E$  по відношенню до деякого класу задач  $T$  та міри продуктивності  $P$  називається машинним навчанням, якщо її продуктивність у задачах з  $T$ , що вимірюється за допомогою  $P$ , покращується з досвідом  $E$ .

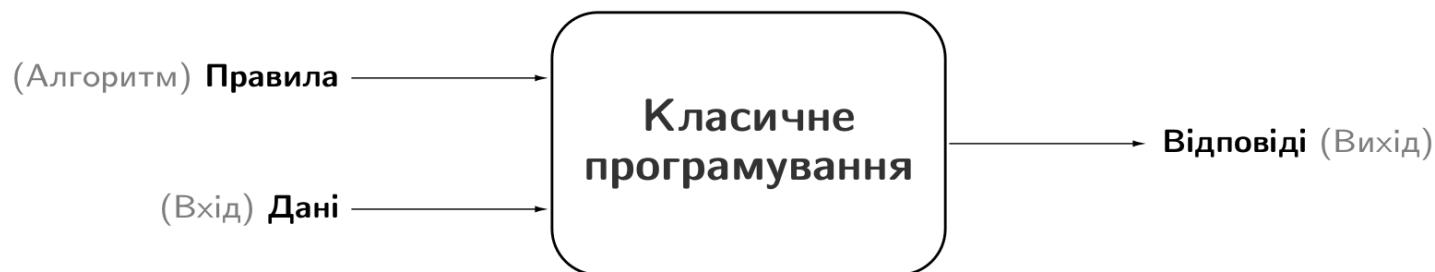


- Досвід (дані): ігри в які грає програма сама з собою
- Вимір продуктивності: коефіцієнт виграшу

# Класичне програмування vs машинне навчання



# Класичне програмування vs машинне навчання

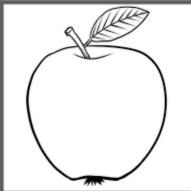
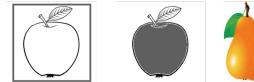


# ТИПИ НАВЧАННЯ

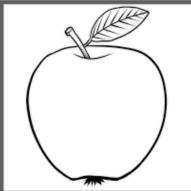
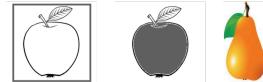
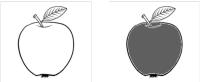
За характером навчальних даних (**досвіду**) машинне навчання поділяють на чотири типи: контролльоване (з учителем), напівконтрольоване, неконтрольоване (без учителя) та з підкріпленням.

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>
Даний: $(x, y)$ $x$ – приклад, $y$ – мітка			
Мета – знайти функцію відображення $x \rightarrow y$			
Приклад  Це є яблуко.			

# Типи навчання

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>
<p><b>Дани:</b> <math>(x, y)</math> <math>x</math> – приклад, <math>y</math> – мітка</p>	<p><b>Дани:</b> <math>(x, y)</math> та <math>x,  (x, y)  &lt;  x </math> <math>x</math> – приклад, <math>y</math> – мітка</p>		
<p><b>Мета</b> – знайти функцію відображення <math>x \rightarrow y</math></p>	<p><b>Мета</b> – знайти функцію відображення або категорію <math>x \rightarrow y</math></p>		
<p><b>Приклад</b></p>  <p>Це є яблуко.</p>	<p><b>Приклад</b></p>  <p>Це є яблуко.</p>		

# Типи навчання

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>
<p><b>Дани:</b> <math>(x, y)</math> <math>x</math> – приклад, <math>y</math> – мітка</p>	<p><b>Дани:</b> <math>(x, y)</math> та <math>x,  (x, y)  &lt;  x </math> <math>x</math> – приклад, <math>y</math> – мітка</p>	<p><b>Дани:</b> <math>x</math> <math>x</math> – приклад, немає мітки!</p>	
<p><b>Мета</b> – знайти функцію відображення <math>x \rightarrow y</math></p>	<p><b>Мета</b> – знайти функцію відображення або категорію <math>x \rightarrow y</math></p>	<p><b>Мета</b> – знайти правильну категорію.</p>	
<p><b>Приклад</b></p>  <p>Це є яблуко.</p>	<p><b>Приклад</b></p>  <p>Це є яблуко.</p>	<p><b>Приклад</b></p>  <p>Цей об'єкт схожий на інший.</p>	

# Типи навчання

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>
<p><b>Дани:</b> <math>(x, y)</math> <math>x</math> – приклад, <math>y</math> – мітка</p> <p><b>Мета</b> – знайти функцію відображення <math>x \rightarrow y</math></p> <p><b>Приклад</b> </p> <p>Це є яблуко.</p>	<p><b>Дани:</b> <math>(x, y)</math> та <math>x,  (x, y)  &lt;  x </math> <math>x</math> – приклад, <math>y</math> – мітка</p> <p><b>Мета</b> – знайти функцію відображення або категорію <math>x \rightarrow y</math></p> <p><b>Приклад</b> </p> <p>Це є яблуко.</p>	<p><b>Дани:</b> <math>x</math> <math>x</math> – приклад, немає мітки!</p> <p><b>Мета</b> – знайти правильну категорію.</p> <p><b>Приклад</b> </p> <p>Цей об'єкт схожий на інший.</p>	<p><b>Дани:</b> пари стан-дія</p> <p><b>Мета</b> – максимізація загальної винагороди, отриманої агентом при взаємодії з навколошнім середовищем.</p> <p><b>Приклад</b> </p> <p>Їжте це, бо це зробить вас сильнішим.</p>

# Навчання з учителем

Постановка задачі

Нехай  $\mathbf{d} \sim p(\mathbf{X}, y)$  – датасет з  $n$  пар прикладів вхід-виход

$$\mathbf{d} = \{(\mathbf{X}^{(1)}, y^{(1)}), (\mathbf{X}^{(2)}, y^{(2)}), \dots, (\mathbf{X}^{(n)}, y^{(n)})\},$$

де  $\mathbf{X}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_m^{(i)})$  – вхідний вектор ознак,  $y^{(i)}$  – мітка (виход), як правило,  $y^{(i)} \in \mathbb{R}$  або  $y^{(i)} \in \mathbb{N}$ .

На основі цих даних ми хочемо визначити статистичну модель

$$p(y|\mathbf{X}),$$

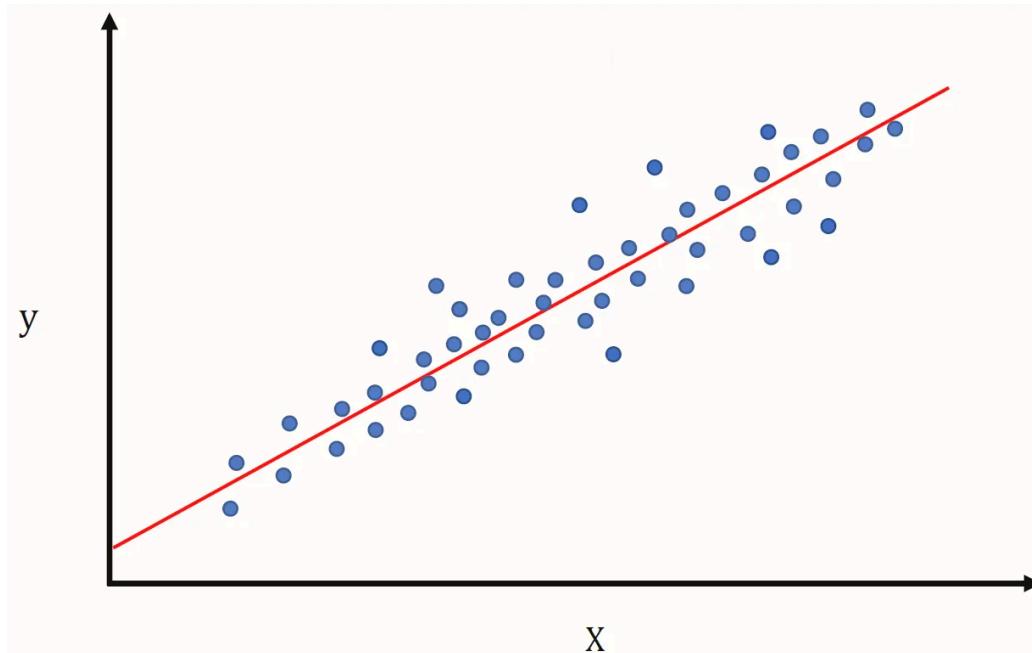
яка буде найкраще пояснювати дані.

# Вектор ознак

- Кожен вхідний приклад  $\mathbf{X} \in \mathbb{R}^m$  – вектор вхідних ознак, який складається з  $m$  атрибутів або ознак.
- Якщо дані спочатку не виражені як дійсні вектори, тоді їх потрібно підготувати та перетворити в цей формат.

# Лінійна регресія

Якщо  $y \in \mathbb{R}$ .



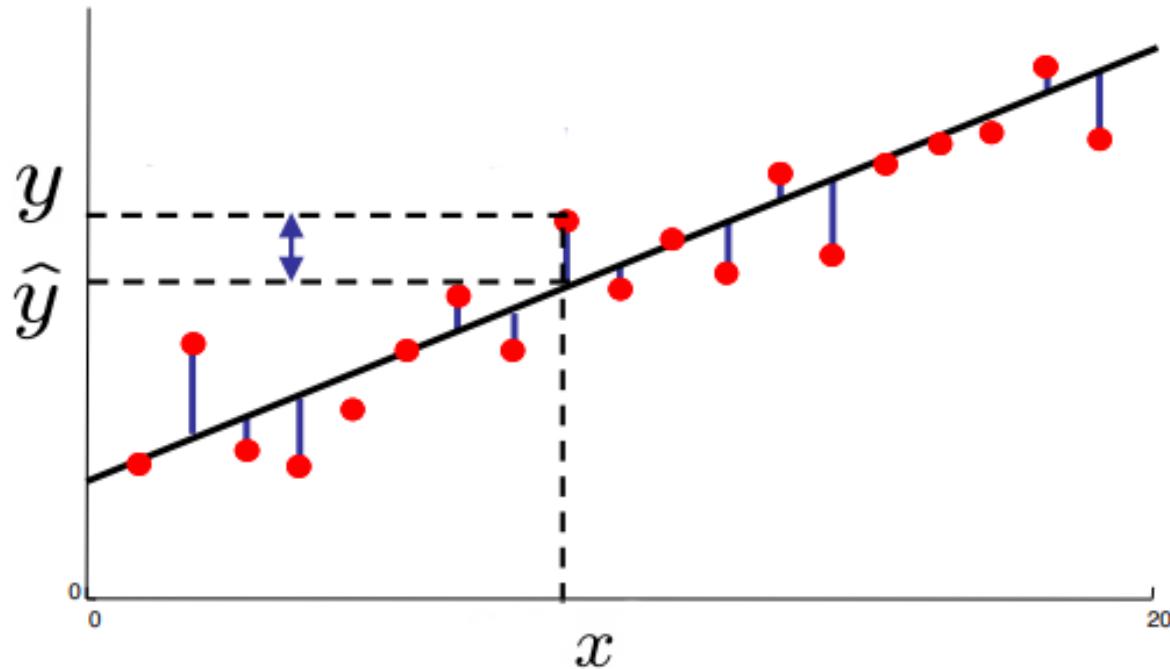
$$\hat{y} = W \cdot X + b$$

Алгоритм регресії прагне відшукати лінію чи гіперповерхню, яка найближче знаходиться до прикладів:  $(\mathbf{X}^{(i)}, y^{(i)})$ . Одновимірна регресія:  $\mathbf{X}^{(i)} = (x_1^{(i)})$

# Лінійна регресія

Втрати

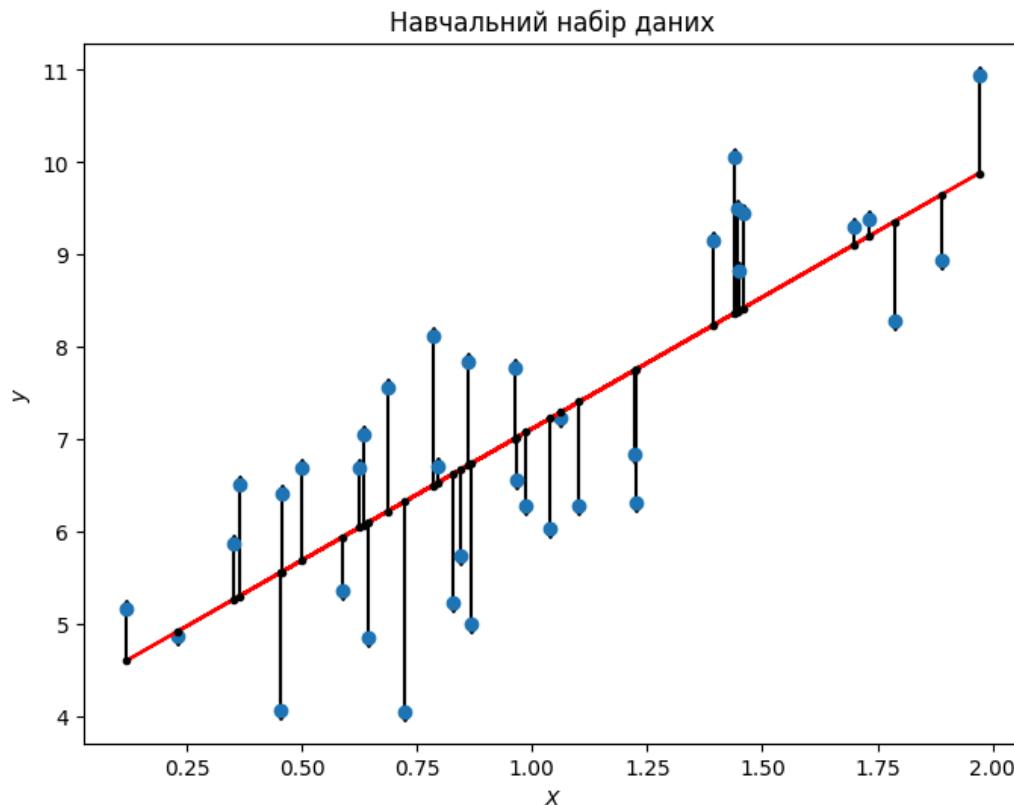
$$L^{(i)}(\hat{y}, y) = (\hat{y}^{(i)} - y^{(i)})^2$$



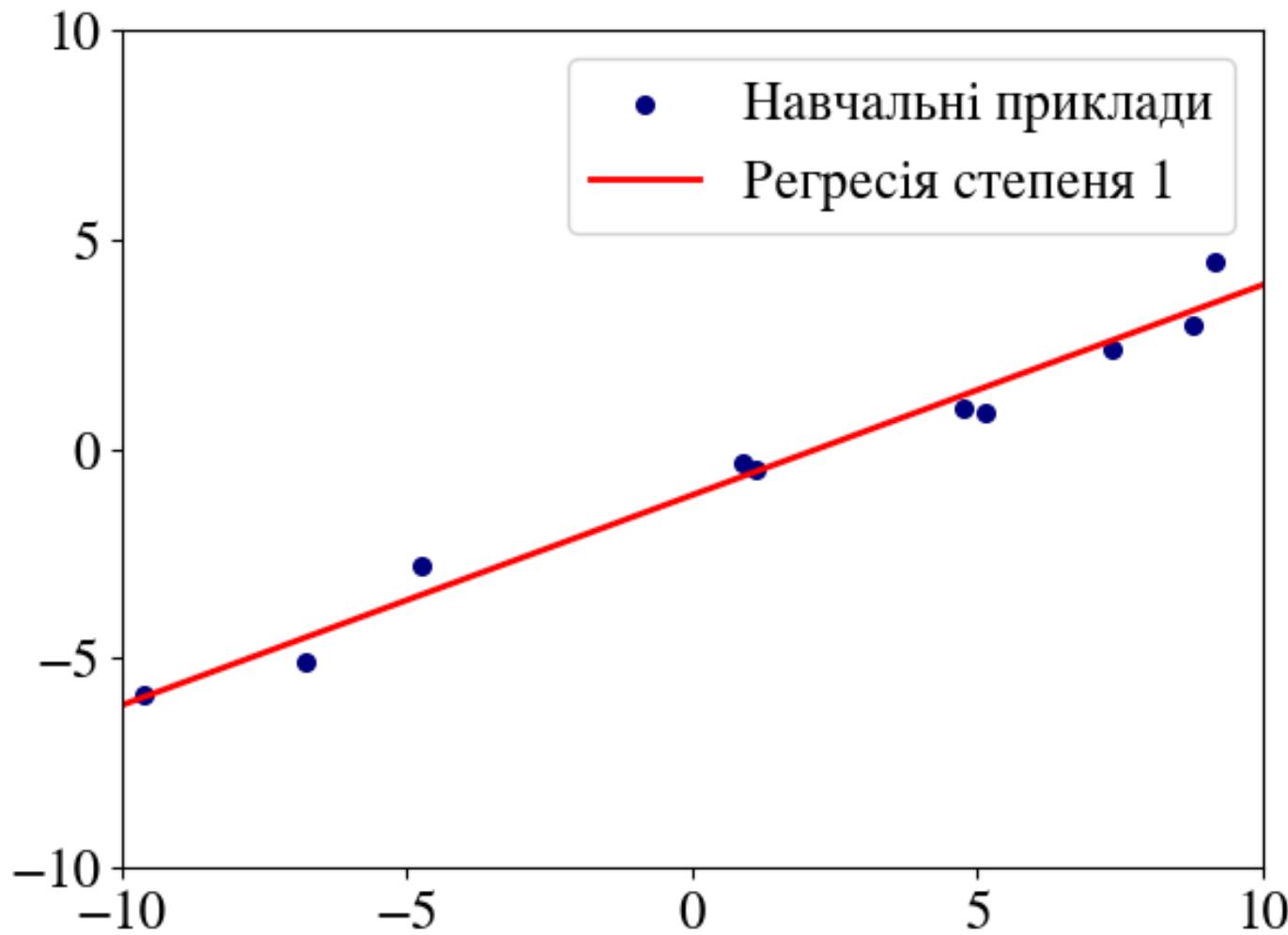
# Лінійна регресія

Цільова функція

$$J(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^n L^{(i)} = \frac{1}{n} \sum_{i=1}^n (\hat{y}^{(i)} - y^{(i)})^2$$

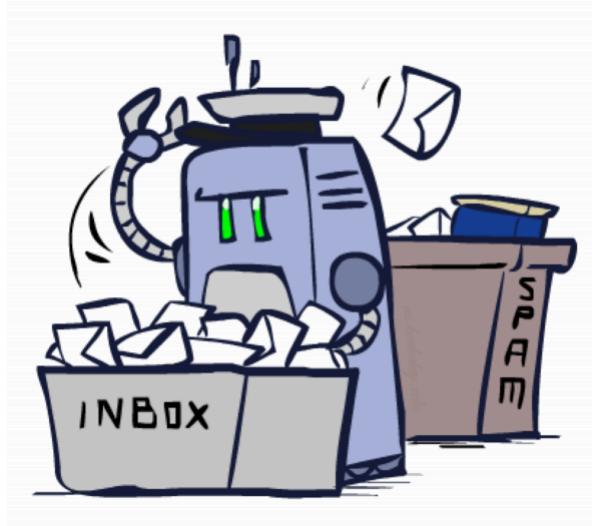


# Лінійна регресія



# Логістична регресія

Якщо  $y \in \{0, 1\}$ .



$$\hat{y} = \sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{1}{1 + \exp(-(W \cdot X + b))}$$

У разі класифікації алгоритм навчання шукає лінію (або, у загальному випадку, гіперповерхню), яка розділяє приклади різних класів.

# Логістична регресія

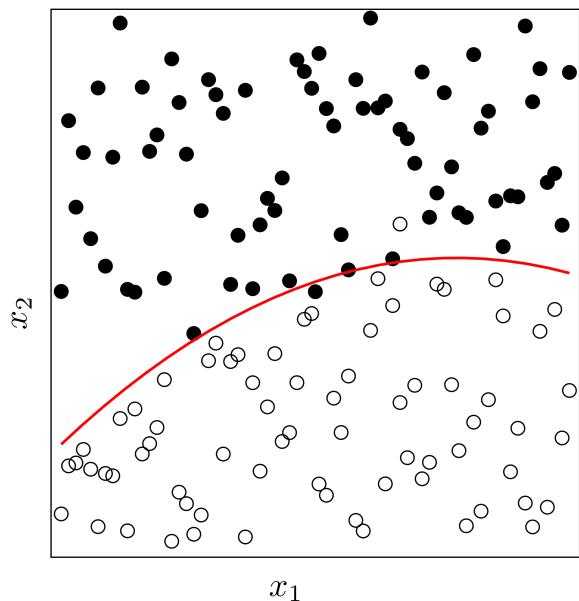
Втрати

$$L^{(i)}(\hat{y}, y) = -\left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

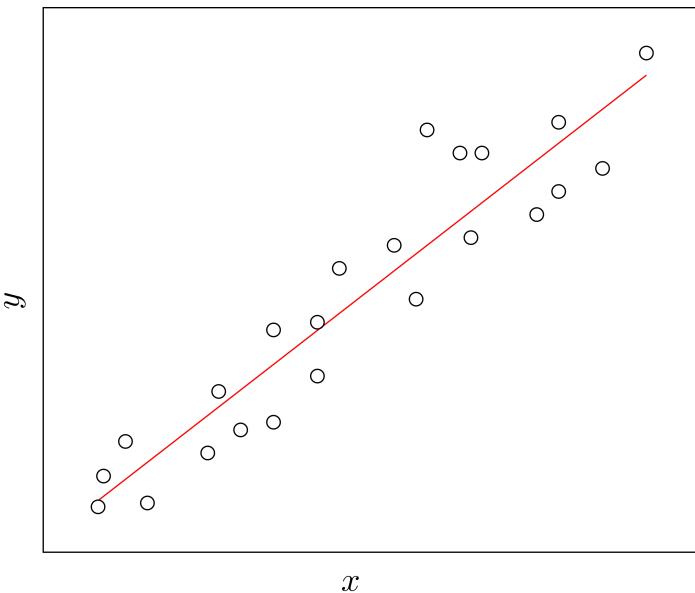
Цільова функція

$$J(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^n L^{(i)} = -\frac{1}{n} \sum_{i=1}^n \left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

# Класифікація vs регресія



Класифікація



Регресія

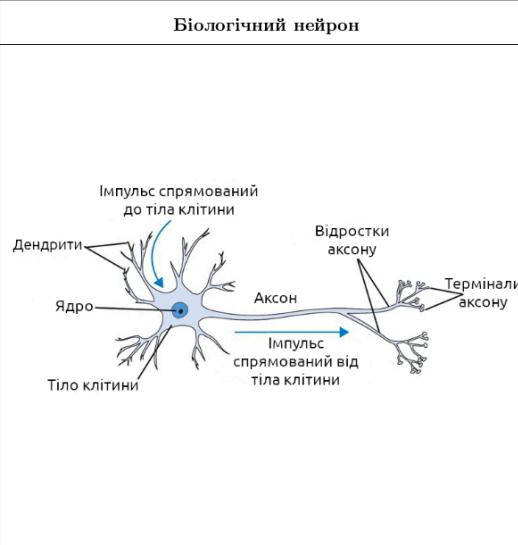
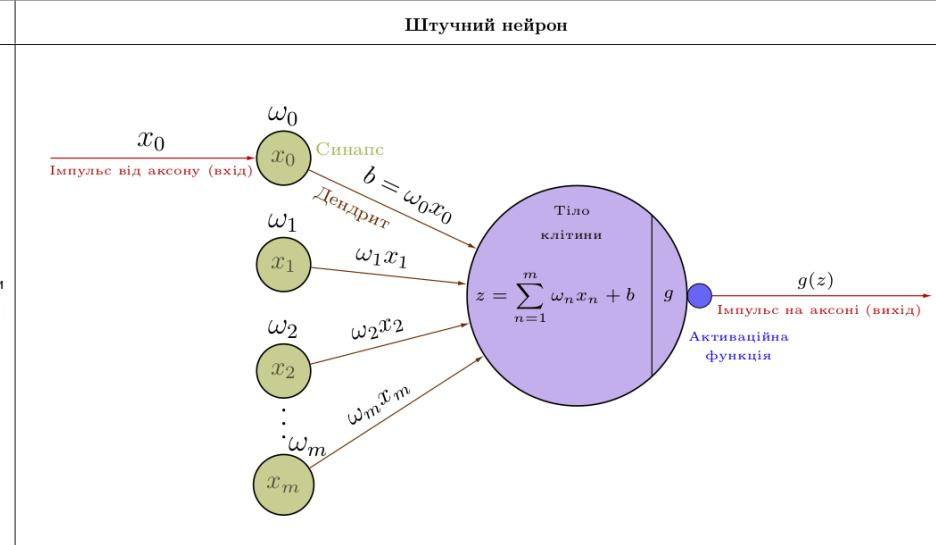
# Як вчиться людина?

- Ми та інші розумні істоти, вчимось завдяки **взаємодії** із своїм оточенням
- Взаємодії часто бувають **послідовними** - майбутні взаємодії можуть залежати від попередніх
- Ми направлені на **результат**
- Ми можемо вчитися **не маючи прикладів** оптимальної поведінки

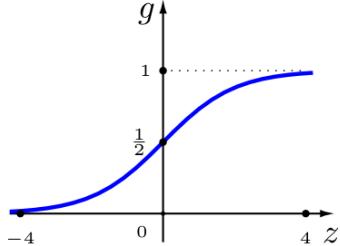
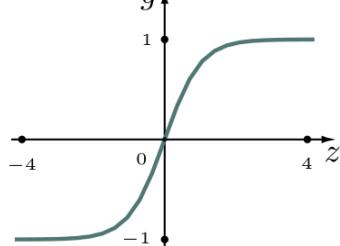
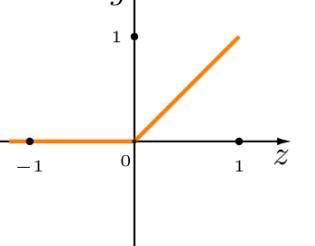
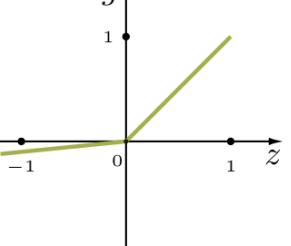
# Мозок людини

Базовою обчислювальною одиницею мозку є нейрон. Мозок дорослої людини складається з 86 мільярдів нейронів, які з'єднані між собою приблизно  $10^{14}$  –  $10^{15}$  синапсами.

# Біологічний та штучний нейрон

Біологічний нейрон	Штучний нейрон
 <p>Дендрити Ядро Тіло клітини Аксон Відростки аксону Термінали аксону Імпульс спрямований до тіла клітини Імпульс спрямований від тіла клітини</p>	 <p>Синапс Дендрит <math>b = \omega_0 x_0</math>  <math>x_0</math>  <math>\omega_0</math>  <math>x_1</math>  <math>\omega_1</math>  <math>x_2</math>  <math>\omega_2</math>  <math>\vdots</math>  <math>x_m</math>  <math>\omega_m</math>  <math>\omega_0 x_0</math>  <math>\omega_1 x_1</math>  <math>\omega_2 x_2</math>  <math>\omega_m x_m</math>  <math>z = \sum_{n=1}^m \omega_n x_n + b</math>  <math>g</math>  <math>g(z)</math>  <math>g(z)</math>      Активаційна функція      Тіло клітини      Імпульс від аксону (вхід)      Імпульс на аксоні (вихід)</p>

# Деякі функції активації

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ $\epsilon \ll 1$
			

Людина добре сприймати  
візуальну інформацію



Що Ви бачите?



Собака-вівця чи швабра?

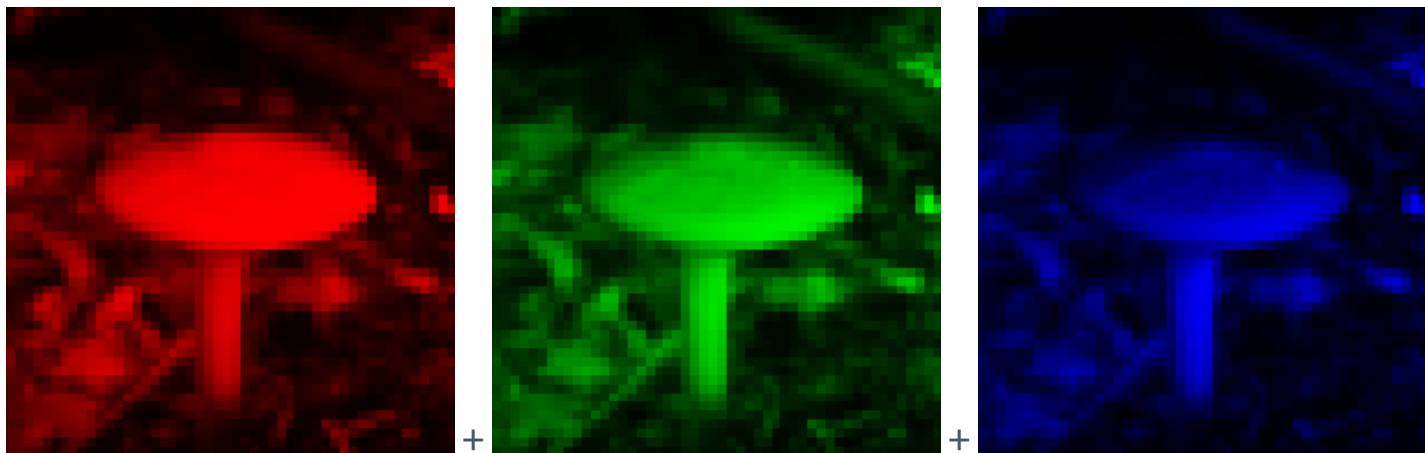


Кекс чи собака?

Людський мозок настільки добре інтерпретує візуальну інформацію, що **розвив** між зображенням та його семантичною інтерпретацією (пікселями) важко оцінити інтуїтивно:







```
array([[[0.03921569, 0.03529412, 0.02352941, 1.          ],
       [0.2509804 , 0.1882353 , 0.20392157, 1.          ],
       [0.4117647 , 0.34117648, 0.37254903, 1.          ],
       ...,
       [0.20392157, 0.23529412, 0.17254902, 1.          ],
       [0.16470589, 0.18039216, 0.12156863, 1.          ],
       [0.18039216, 0.18039216, 0.14117648, 1.          ]],

      [[0.1254902 , 0.11372549, 0.09411765, 1.          ],
       [0.2901961 , 0.2509804 , 0.24705882, 1.          ],
       [0.21176471, 0.2        , 0.20392157, 1.          ],
       ...,
       [0.1764706 , 0.24705882, 0.12156863, 1.          ],
       [0.10980392, 0.15686275, 0.07843138, 1.          ],
       [0.16470589, 0.20784314, 0.11764706, 1.          ]],

      [[0.14117648, 0.12941177, 0.10980392, 1.          ],
       [0.21176471, 0.1882353 , 0.16862746, 1.          ],
       [0.14117648, 0.13725491, 0.12941177, 1.          ],
       ...,
       [0.10980392, 0.15686275, 0.08627451, 1.          ],
       [0.0627451 , 0.08235294, 0.05098039, 1.          ],
       [0.14117648, 0.2        , 0.09803922, 1.          ]],

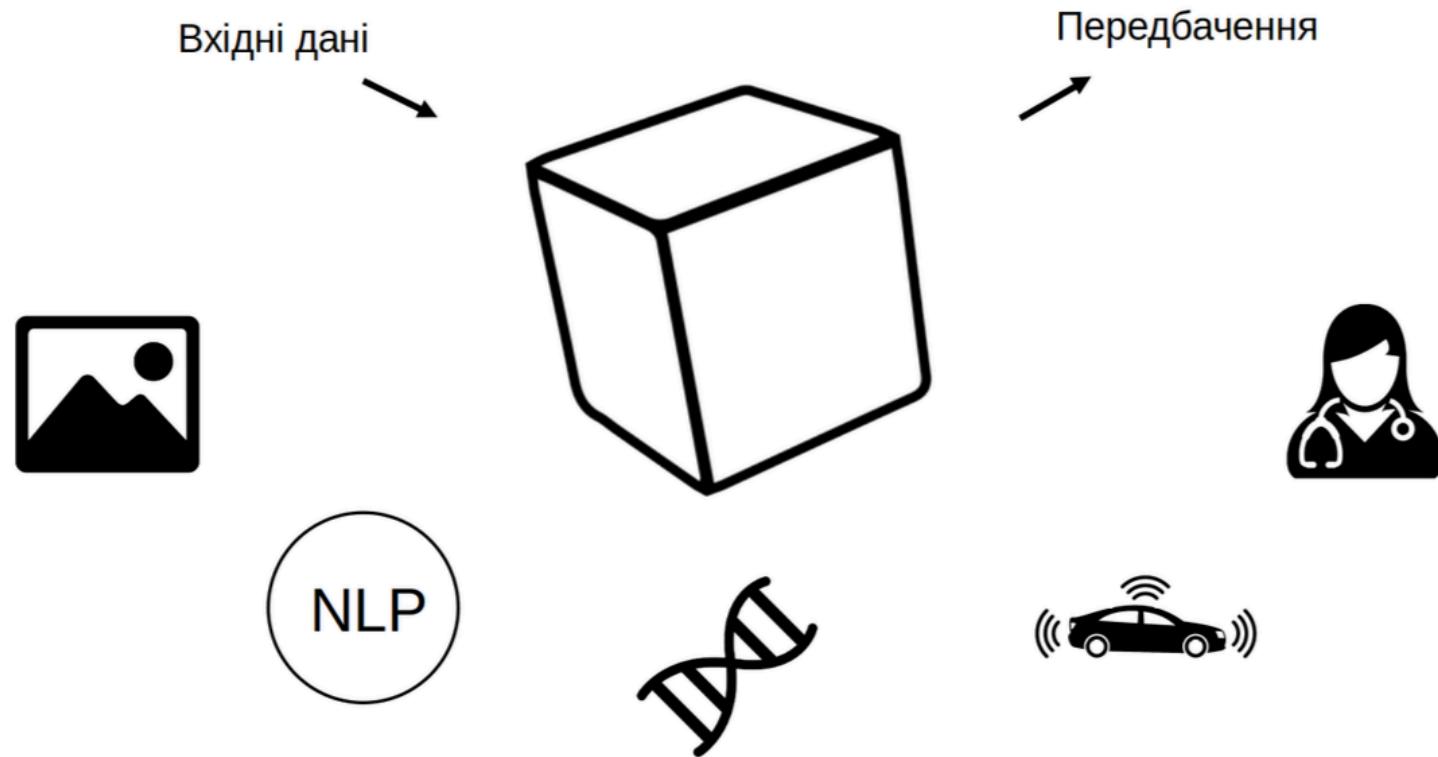
      ...]
```

Зображення гриба.

# Що входить до задачі машинного навчання?

- Постановка проблеми + збір даних
- Навчання моделі
- Визначення функції втрат
- Вибір алгоритму оптимізації

# Які дані використовуються?



# Ознаки у машинному навчанні

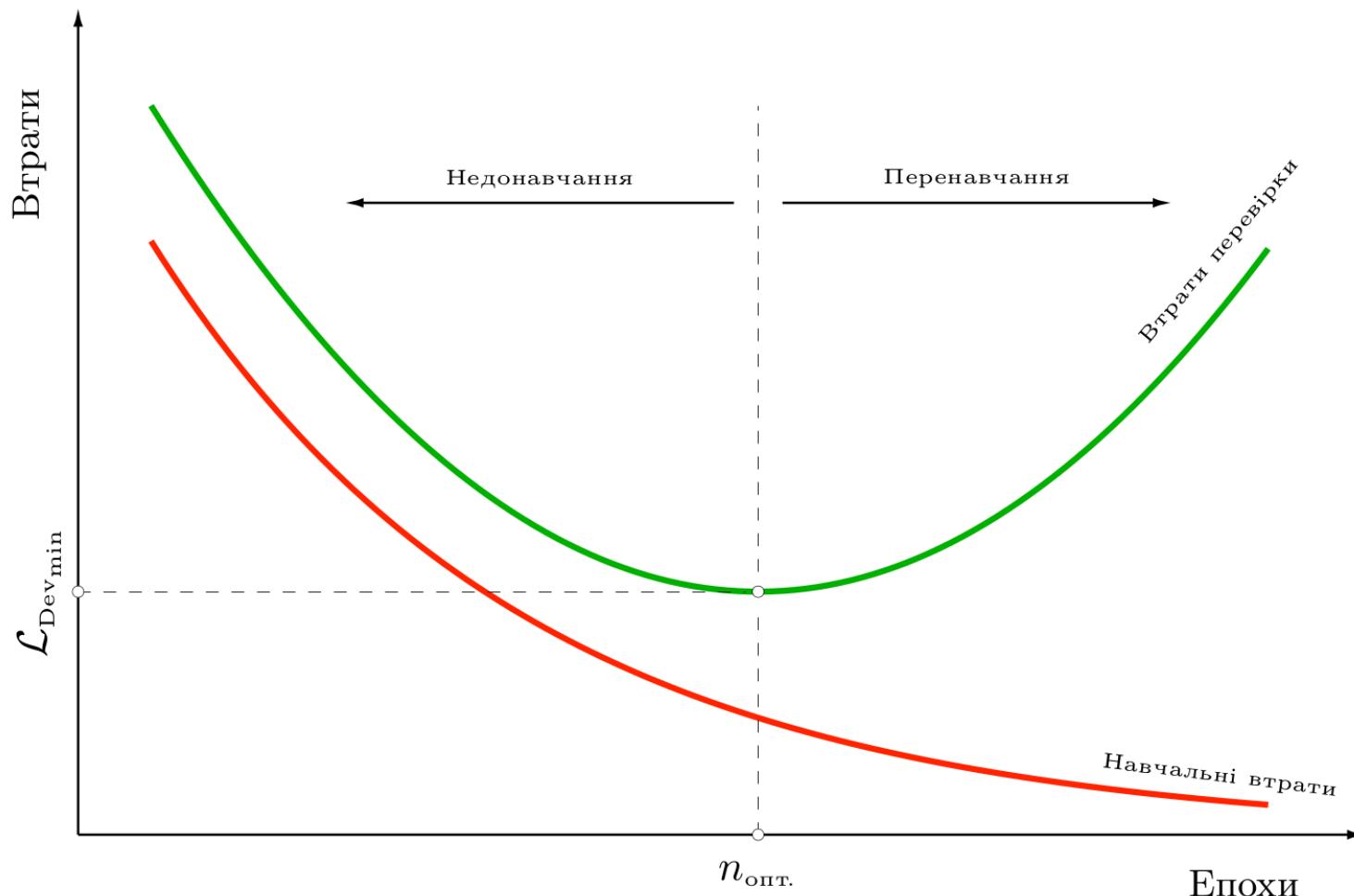
Ознаки - це спостереження, які використовуються для прийняття рішень моделлю.

- Для класифікації зображень  **кожен** піксель є ознакою
- Для розпізнавання голосу, **частота** та **гучність** є ознаками
- Для безпілотних автомобілів дані з **камер, радарів і GPS** є ознаками

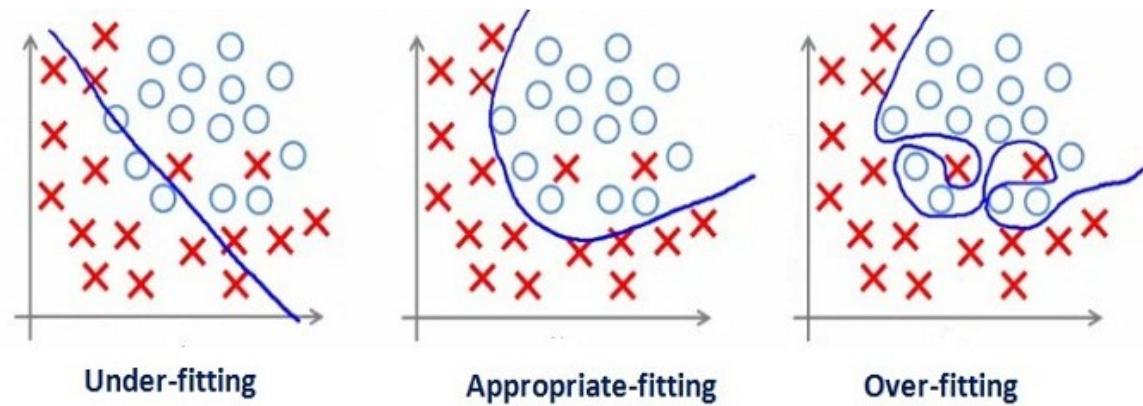
# Типи ознак у робототехніці

- Пікселі (RGB дані)
- Глибина (сонар, лазерні далекоміри)
- Орієнтація або прискорення (гіроскоп, акселерометр, компас)

# Недонавчання vs перенавчання

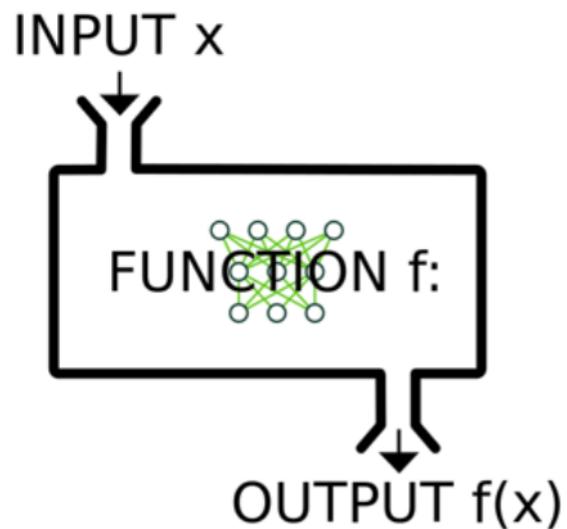


# Недонавчання vs перенавчання



# Що таке модель?

Хоча те, що знаходиться всередині глибинної нейронної мережі, може бути складним, за своєю суттю це просто функції. Вони беруть певні вхідні дані: **INPUT x** і генерують деякі вихідні дані: **OUTPUT f(x)**



# Типи моделей

Дискримінтивні та генеративні моделі

Machine Learning

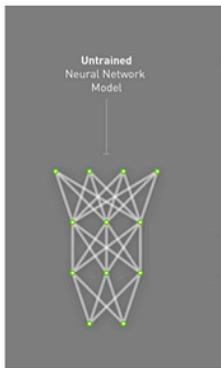
Discriminative

Generative

# З чого складається модель?

## Компоненти моделі

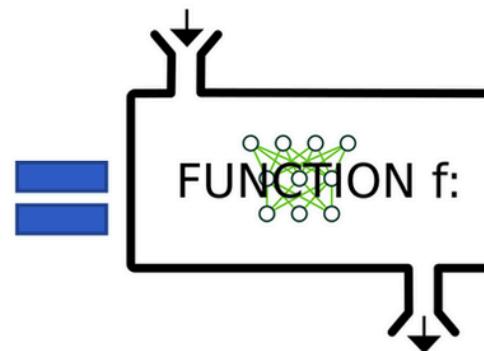
Архітектура мережі = [deploy.prototxt](#)



Навчені ваги = \*\*\*.caffemodel



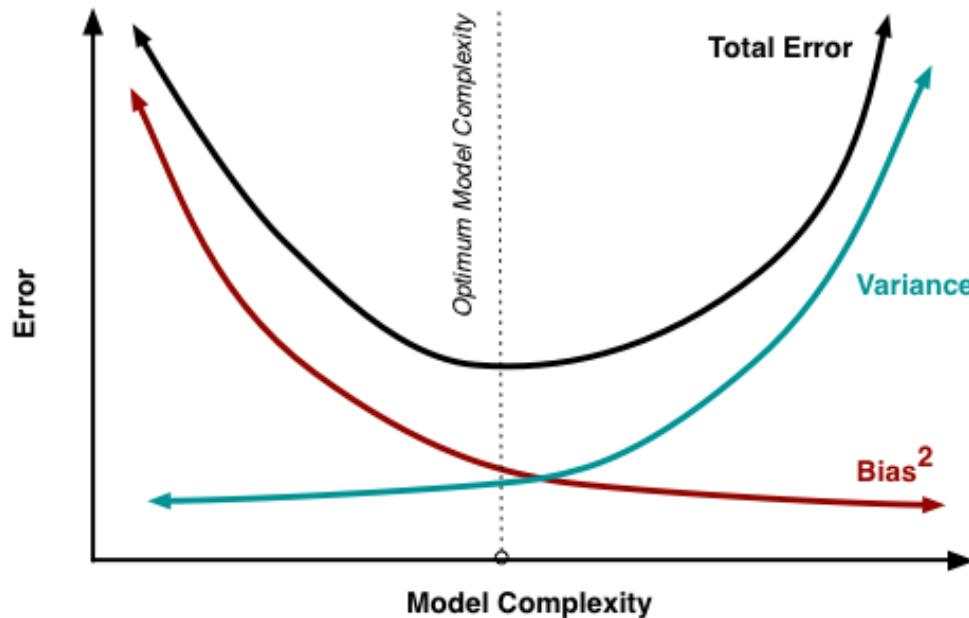
Модель



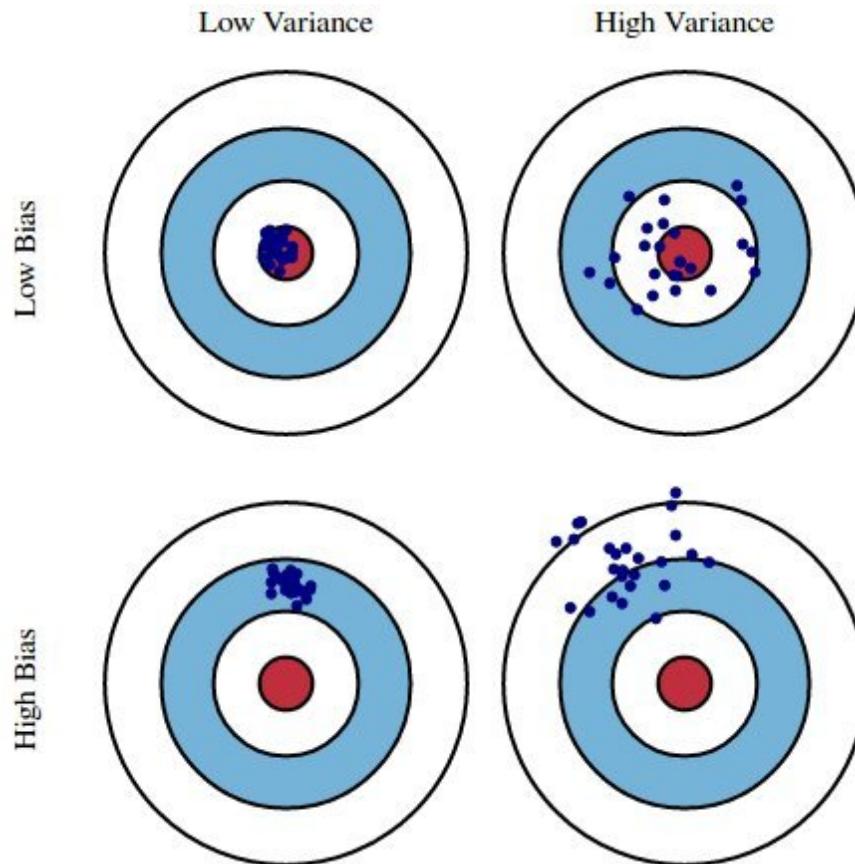
# Джерела помилок моделі

- Зсув (Bias)
- Розкид (Variance)
- Шум (Irreducible error)

$$Err = Bias^2 + Variance + Irreducible\ error$$

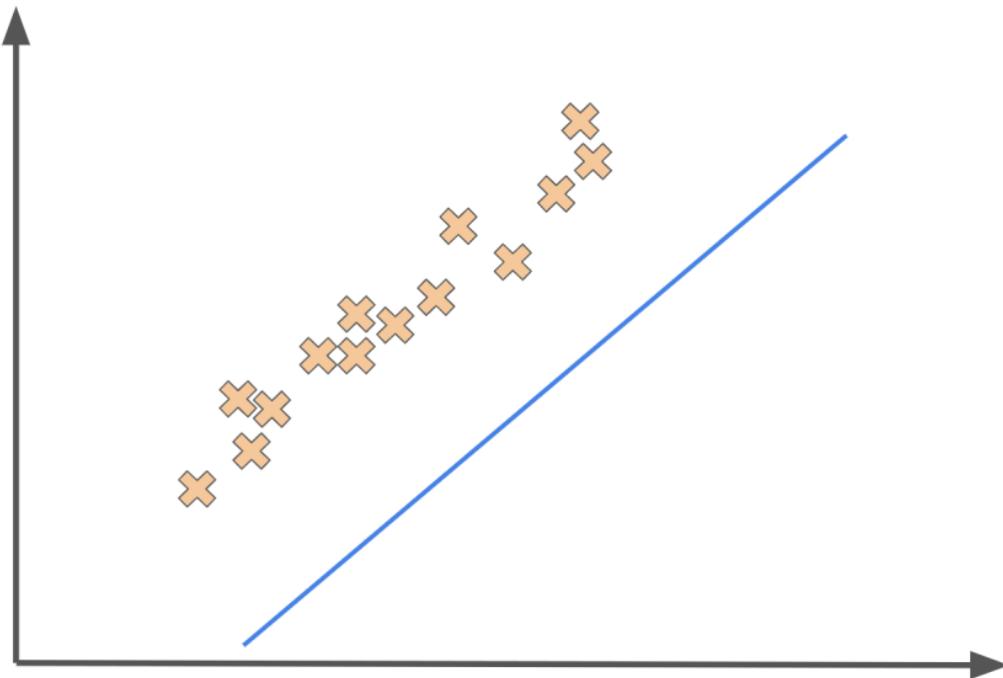


# Інтуїція



# Інтуїція

Великий зсув



# Сфери застосування та успіхи ШІ



## Detectron2: A PyTorch-based modular object detection |

AI at Meta



Watch on

Виявлення об'єктів, визначення положення людини, сегментація (2019)

Sense, Solve, and Go: The Magic of the Waymo Driver



Створення автономних автомобілів (Waymo, 2022)

Powering the Future of Clean Energy | I AM AI



Рушій для розвитку чистої енергії (NVIDIA, 2023)

Camels, Code & Lab Coats: How AI Is Advancing Science and M...



Як ШІ розвиває медицину (Google, 2023)

Can you write me a report analyzing this chest X-ray?



**Findings:**

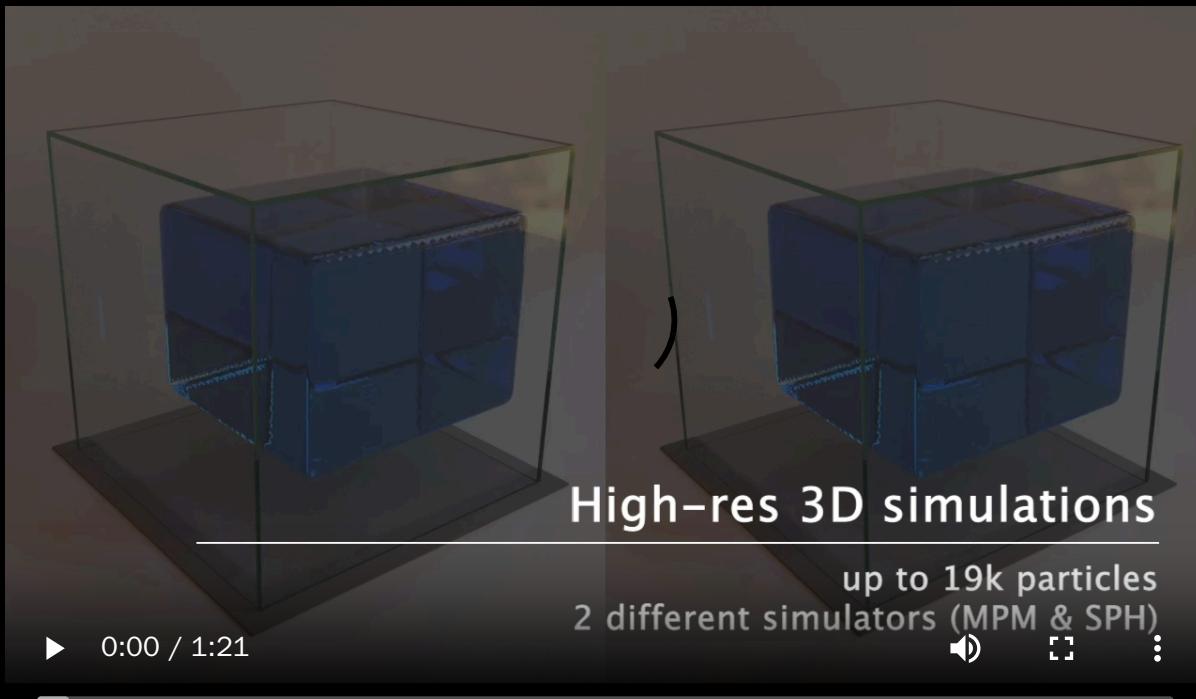
- Devices: None.
- Lungs: No pneumothorax. No substantial pleural effusion. Lungs appear clear.
- Cardiomediastinal: Normal heart size. Mediastinal contours within normal limits.
- Other: No acute skeletal abnormality.

**Impression:**

No active disease seen in chest.

Enter a question here

Med-PaLM 2 (Google) – це велика мовна модель, налаштована для сфери медицини. Досягає 85%+ точності для запитань у стилі експертизи медичного професійного, три-етапного іспиту (USMLE).



Симуляція фізики явищ (Sanchez-Gonzalez et al, 2020)

# AlphaFold: Від послідовності амінокислот до 3D структури

nature

Explore content ▾ About the journal ▾ Publish with us ▾

nature > articles > article

Article | Open access | Published: 15 July 2021

## Highly accurate protein structure prediction with AlphaFold

John Jumper ✉, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishabh Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, ... Demis Hassabis ✉  
+ Show authors

Nature 596, 583–589 (2021) | Cite this article

1.42m Accesses | 12k Citations | 3493 Altmetric | Metrics

### Abstract

Proteins are essential to life, and understanding their structure can facilitate a mechanistic understanding of their function. Through an enormous experimental effort<sup>1,2,3,4</sup>, the structures of around 100,000 unique proteins have been determined<sup>5</sup>, but this represents a small fraction of the billions of known protein sequences<sup>6,7</sup>. Structural coverage is bottlenecked by the months to years of painstaking effort required to determine a single protein structure. Accurate computational approaches are needed to address this gap and to enable large-scale structural bioinformatics. Predicting the three-dimensional structure that a protein will adopt based solely on its amino acid sequence—the structure prediction component of the ‘protein folding problem’<sup>8</sup>—has been an important open research problem for more than 50 years<sup>9</sup>. Despite recent progress<sup>10,11,12,13,14</sup>, existing methods fall far short of atomic accuracy, especially when no homologous structure is available. Here we provide the



John Jumper

(Нобелівська премія з хімії  
2024)



AlphaFold: The making of a scientific breakthrough



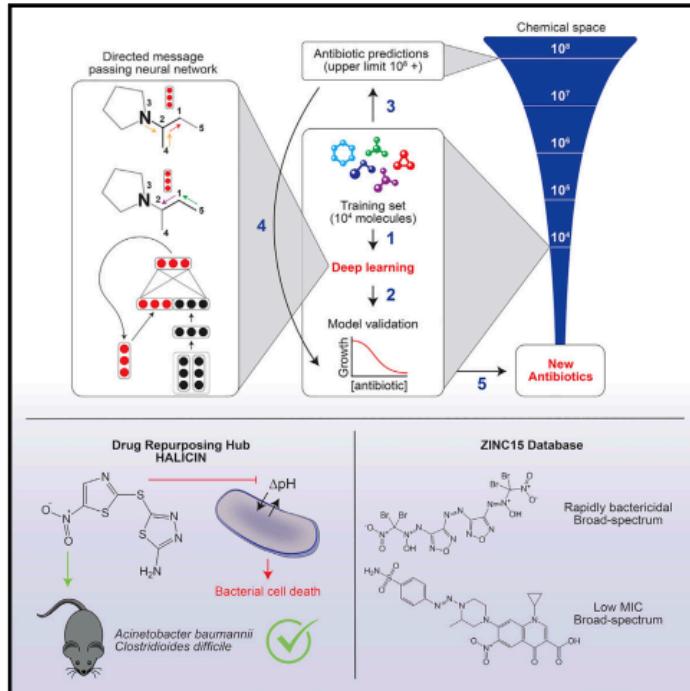
ШІ для науки (Deepmind, AlphaFold, 2020)

# Відкриття ліків за допомогою графових нейронних мереж

Cell

## A Deep Learning Approach to Antibiotic Discovery

### Graphical Abstract



### Authors

Jonathan M. Stokes, Kevin Yang,  
Kyle Swanson, ..., Tommi S. Jaakkola,  
Regina Barzilay, James J. Collins

### Correspondence

regina@csail.mit.edu (R.B.),  
jimjc@mit.edu (J.J.C.)

### In Brief

A trained deep neural network predicts antibiotic activity in molecules that are structurally different from known antibiotics, among which Halicin exhibits efficacy against broad-spectrum bacterial infections in mice.

Google Assistant will soon be able to call restaurants and make ...



Синтез мовлення та відповіді на питання (Google, 2018)

A Style-Based Generator Architecture for Generative Adversarial ...



Генерація зображень (Karras et al, 2018)

DALL·E 2 Explained



Генерація зображень і мистецтво ШІ (OpenAI, 2022)

Reface оживив відомі київські муралі до Дня Києва



Reface оживив відомі київські муралі до Дня Києва (2021)

Creating a Space Game with OpenAI Codex



Написання комп'ютерного коду (OpenAI, 2021)

+ New Thread

# ChatGPT



## Examples

"Explain quantum computing in simple terms" →



## Capabilities

"Got any creative ideas for a 10 year old's birthday?" →



## Limitations

May occasionally generate incorrect information



Allows user to provide follow-up corrections

May occasionally produce harmful instructions or biased content

"How do I make an HTTP request in Javascript?" →

Trained to decline inappropriate requests

Limited knowledge of world and events after 2021

Light Mode

OpenAI Discord

Updates & FAQ

Log out

Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them safer.

Відповісти на всі ваші запитання (OpenAI, 2022)



"Протягом останніх сорока років ми програмували комп'ютери;  
протягом наступних сорока років ми будемо їх навчати."

— Крістофер Бішоп, 2020.

# Штучний інтелект

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



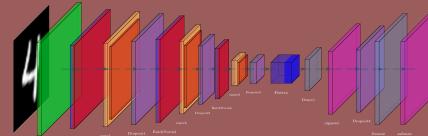
## Машинне навчання

Можливість комп'ютера учитися не будучи явно запрограмованим



## Глибинне навчання

Пошук шаблону в даних за допомогою нейронних мереж





# Література

- Andriy Burkov (2020). [Machine Learning Engineering](#). Chapter 1: Introduction.