



Машинне навчання

Лекція 1: Вступ до машинного навчання

Кочура Юрій Петрович
juriy.kochura@gmail.com
[@y_kochura](https://twitter.com/y_kochura)

Сьогодні

- Інтелект vs штучний інтелект
- Машинне навчання
- Області застосування та успіхи ШІ



Інтелект

VS

Штучний інтелект

Чи може машина 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](#)

In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components,

- a. The initial state of the mind, say at birth,*
- b. The education to which it has been subjected,*
- c. Other experience, not to be described as education, to which it has been subjected.*

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child-brain is something like a note-book as one buys it from the stationers. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed.

— Alan Turing, 1950

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

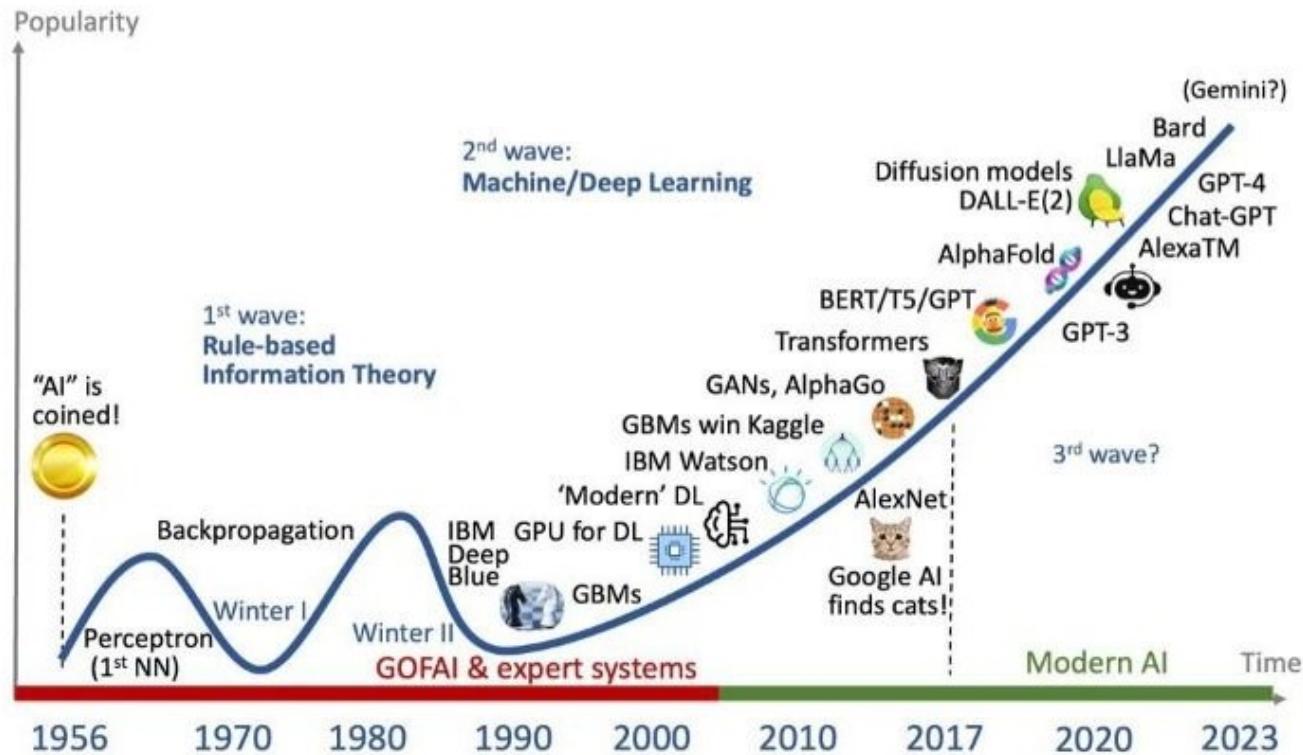
Що таке штучний інтелект?

- У вузькому сенсі

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

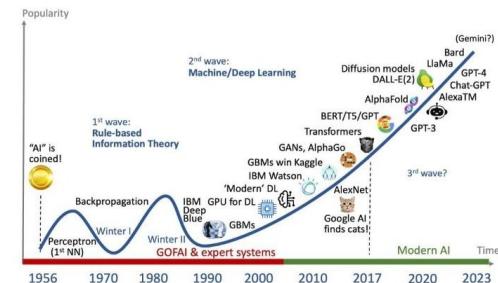
- **Знання** — це факти, інформація та навички, набуті через досвід або навчання.

Коротка історія

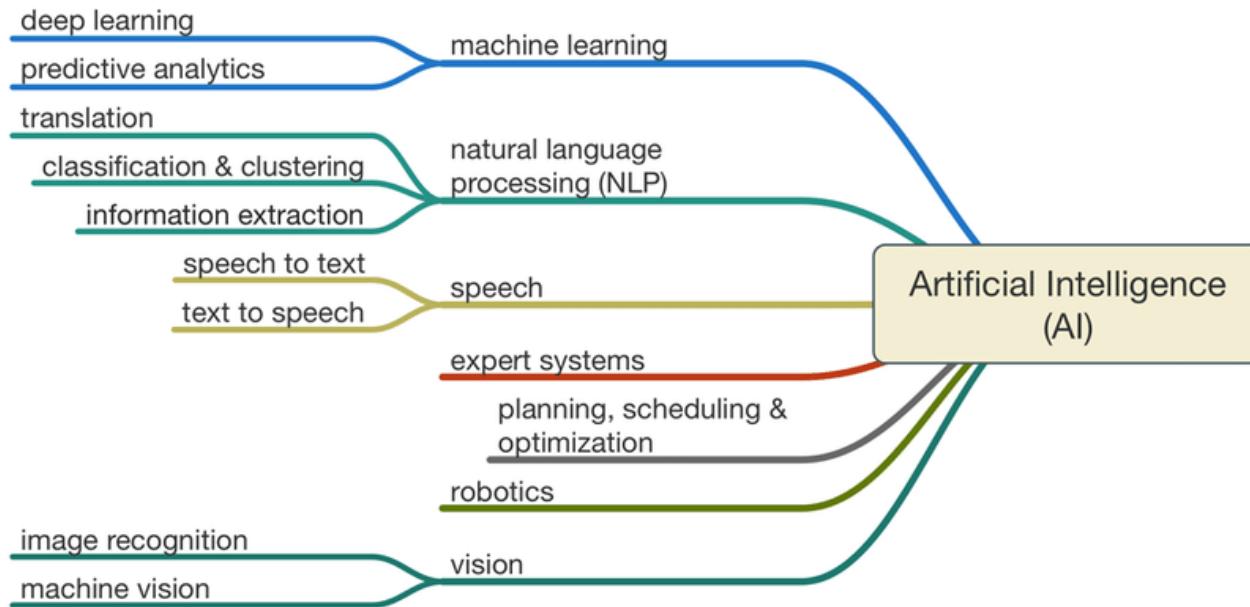


Коротка історія

- 1940–1952: Early days
 - 1943: McCulloch & Pitts: Boolean circuit model of brain
 - 1950: Turing's "Computing Machinery and Intelligence"
- 1952–1956: The birth of AI
 - 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
 - 1956: Dartmouth meeting: "Artificial Intelligence" adopted
- 1956–1974: The golden years
 - 1958: Frank Rosenblatt invented [perceptron](#) (simple neural network)
 - 1964: [Bobrow's program](#) that solves algebra word problems
 - 1965: Robinson's complete algorithm for logical reasoning
- 1974–1980: The first AI winter
- 1980–1987: Expert systems industry boom
- 1987–1993: Expert systems industry busts: the second AI winter
- 1993–2011: Statistical approaches
 - Resurgence of probability, focus on uncertainty
 - General increase in technical depth
 - Intelligent agents
- 2011–present: Deep Learning, Big Data and AI
 - Big data, big compute, neural networks
 - AI used in many industries



AI – багата галузь



SUMMARY OF ML/AI CAPABILITIES

USE CASES

CAPABILITIES

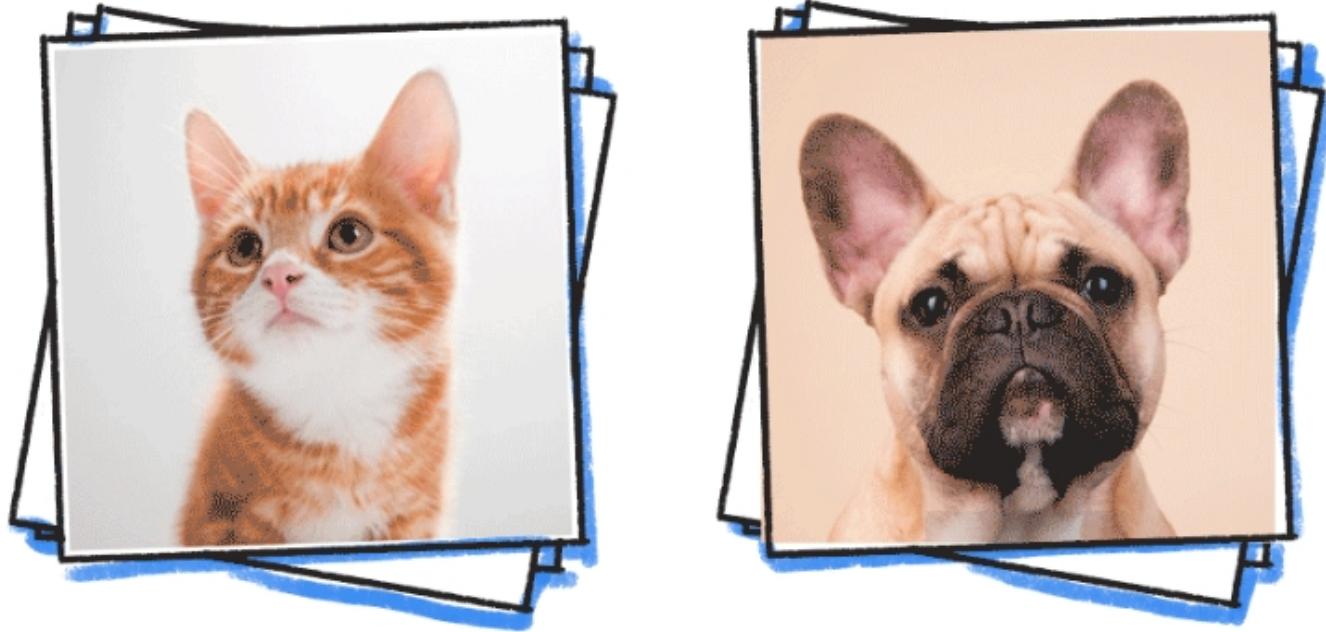
| PERCEPTION (interpreting the world) | VISION understanding images | AUDIO audio recognition | SPEECH • text-to-speech • speech-to-text conversions | NATURAL LANGUAGE understanding & generating text |
|---|---|---|--|---|
| COGNITION (reasoning on top of data) | REGRESSION • predicting a numerical value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



"Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI will transform in the next several years."

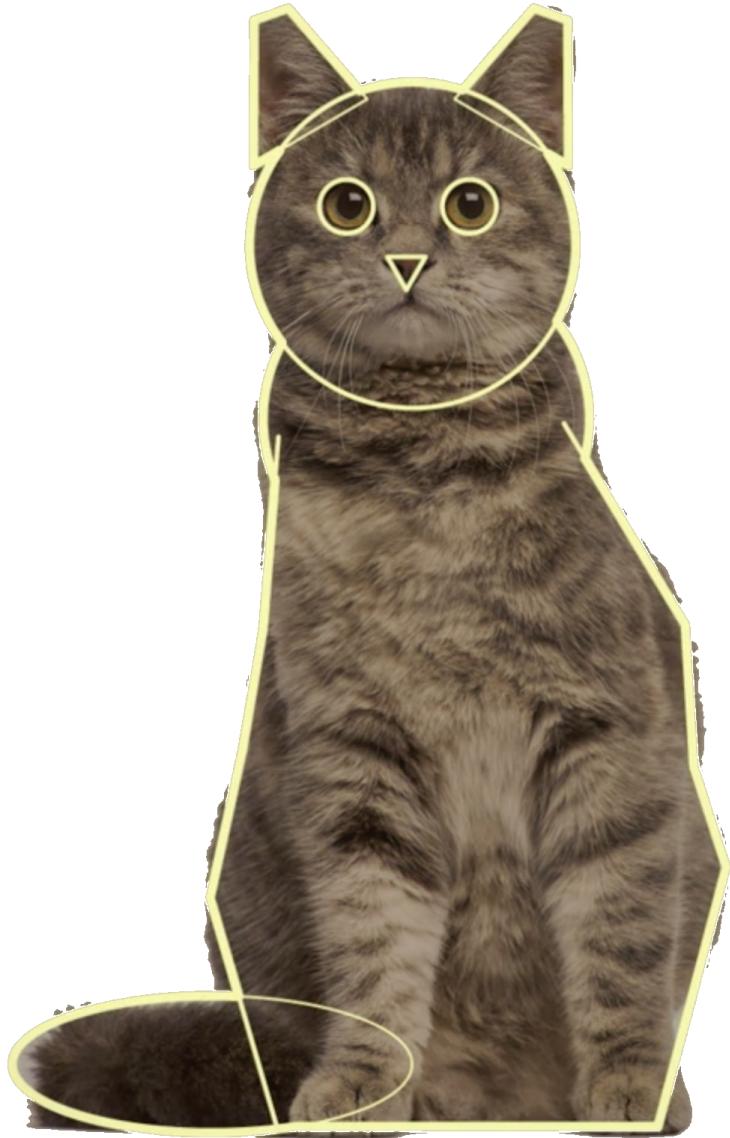
— Andrew Ng

Машинне навчання

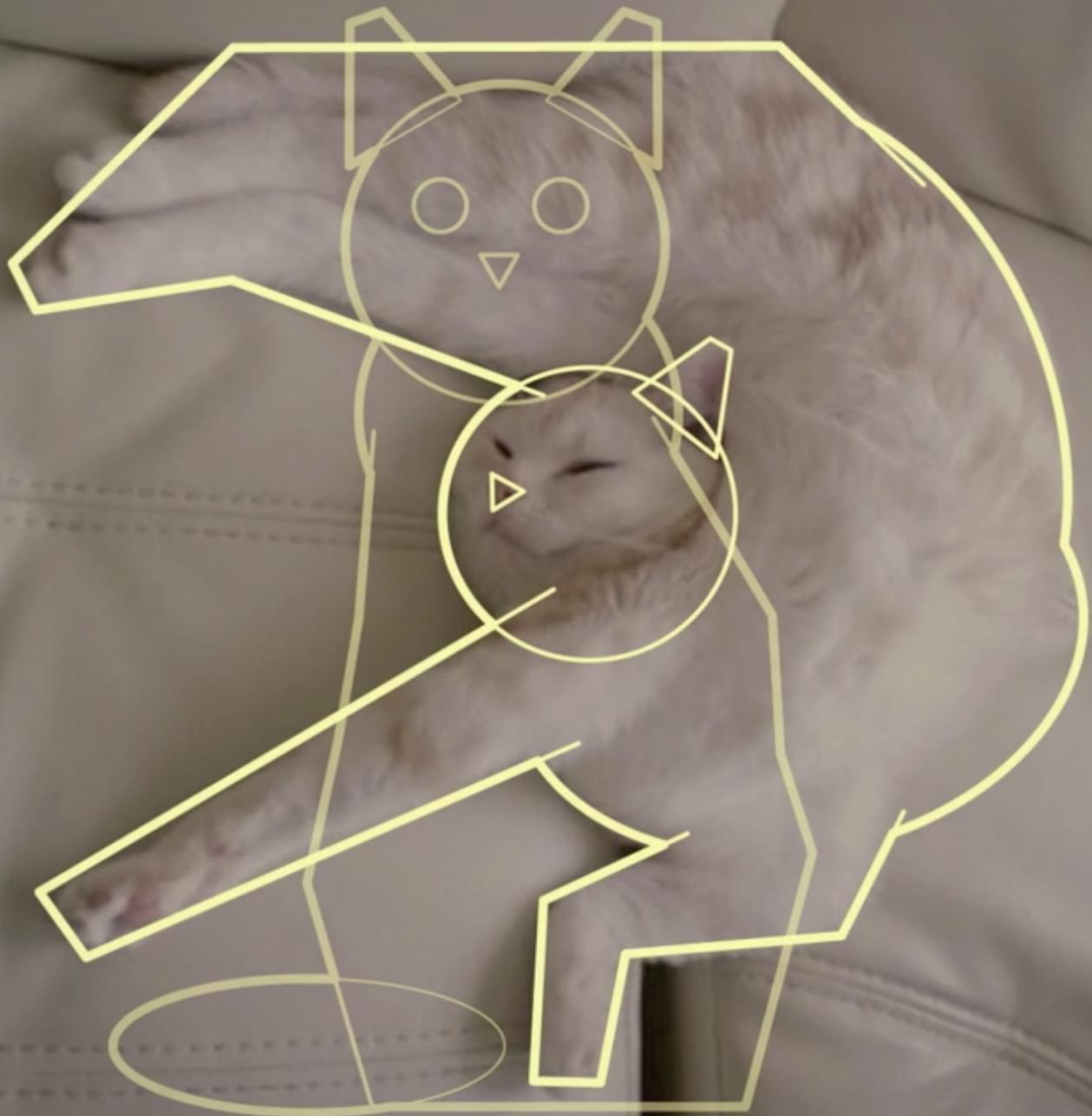


Чи могли б ви написали комп'ютерну програму, яка розпізнає **котів** від **собак**?







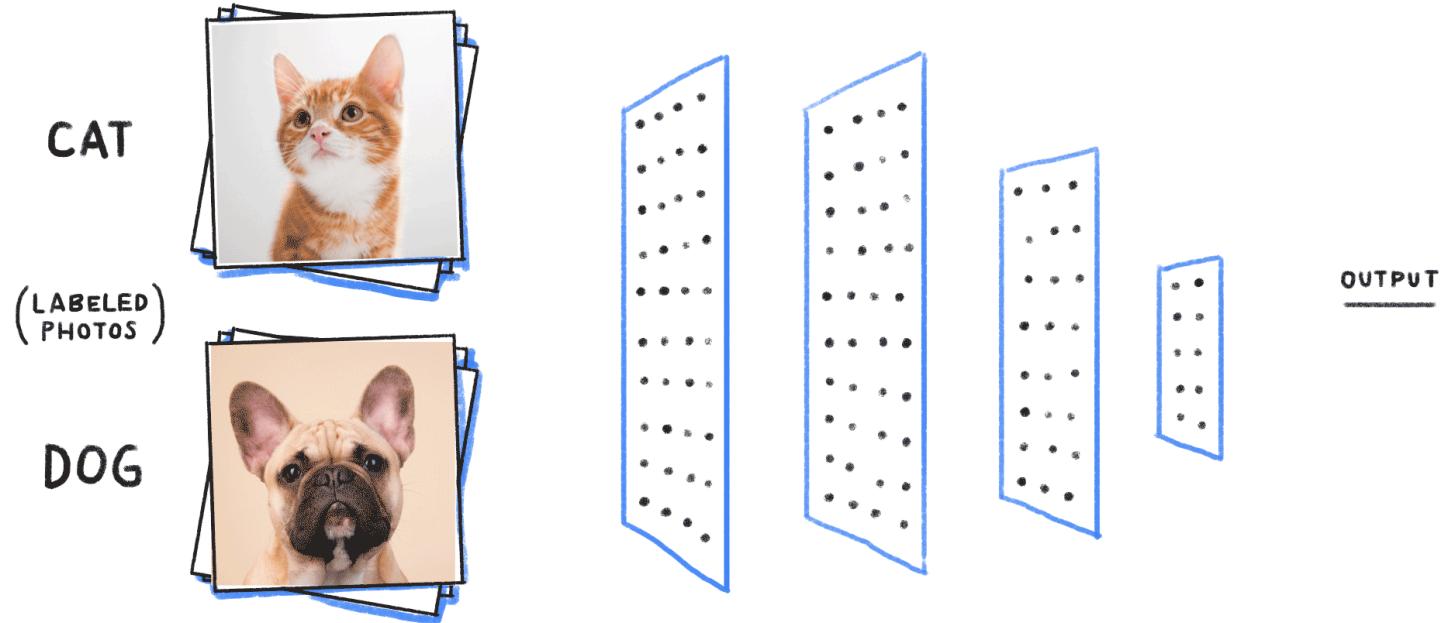




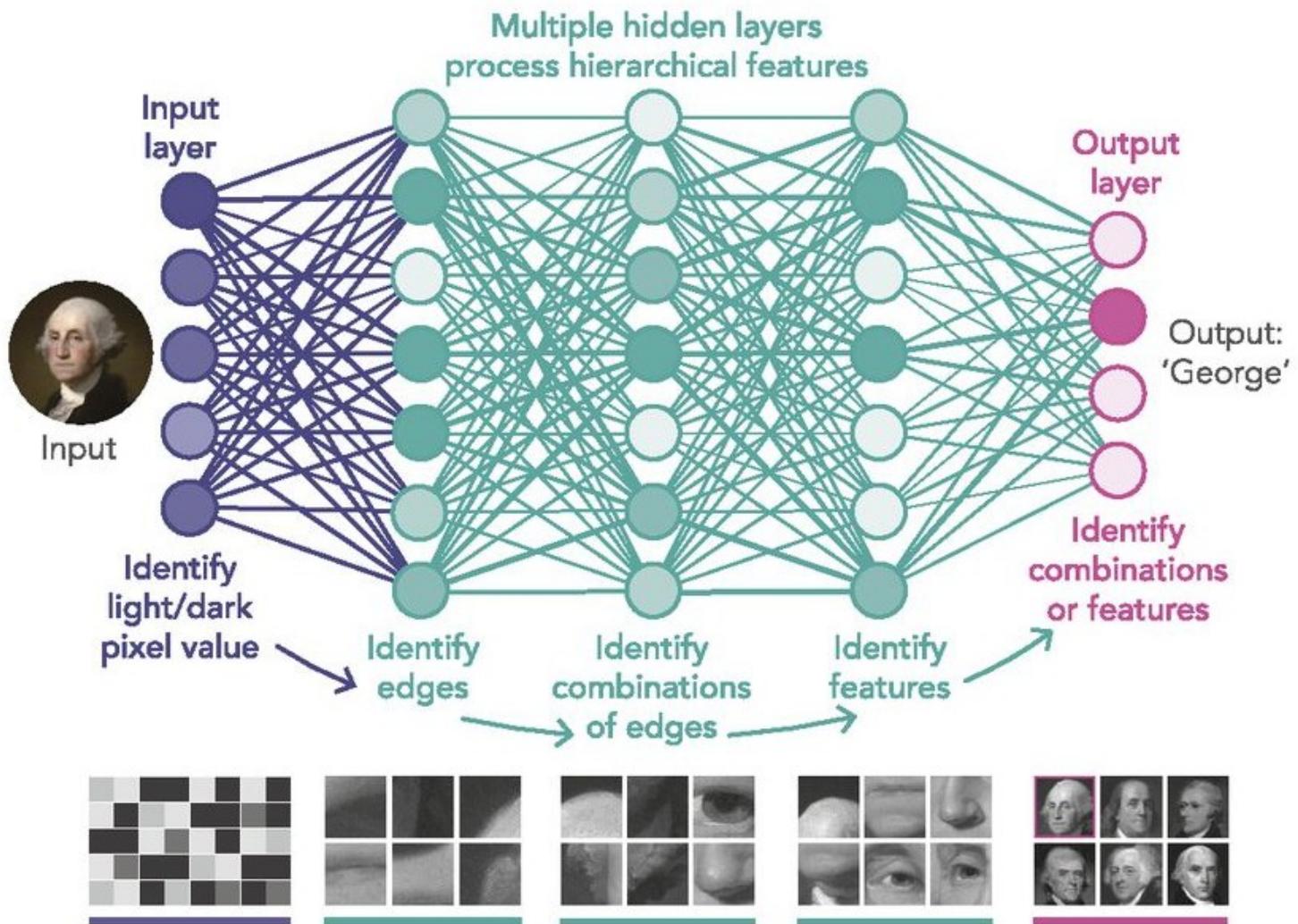


Для пошуку шаблону в даних (витягування семантичної інформації, ознак) потрібна побудова **складних моделей**, які б отримати вручну було б дуже складно.

Однак, можна використати алгоритм машинного навчання, який буде **вчитись** знаходити шаблон у даних самостійно.



Підхід глибокого навчання



Що таке машинне навчання?

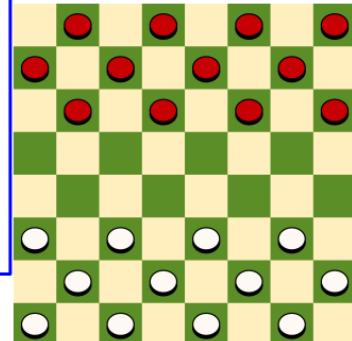
Визначення за Артур Семюель

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**



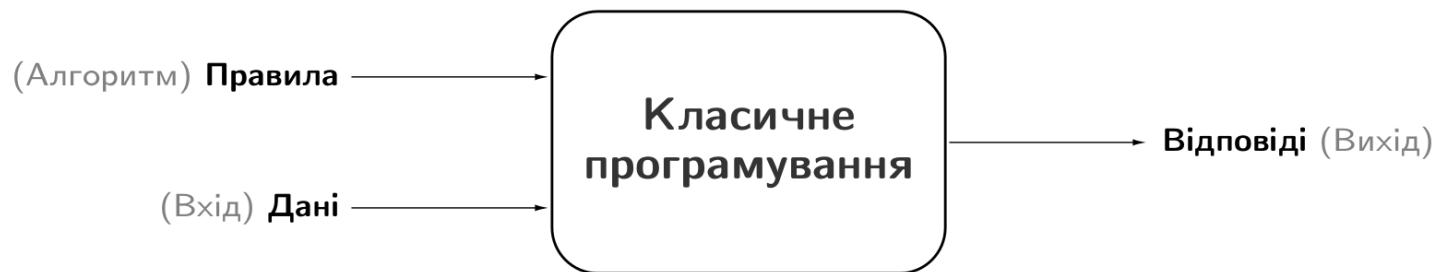
Визначення за Том Мітчелл

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 .

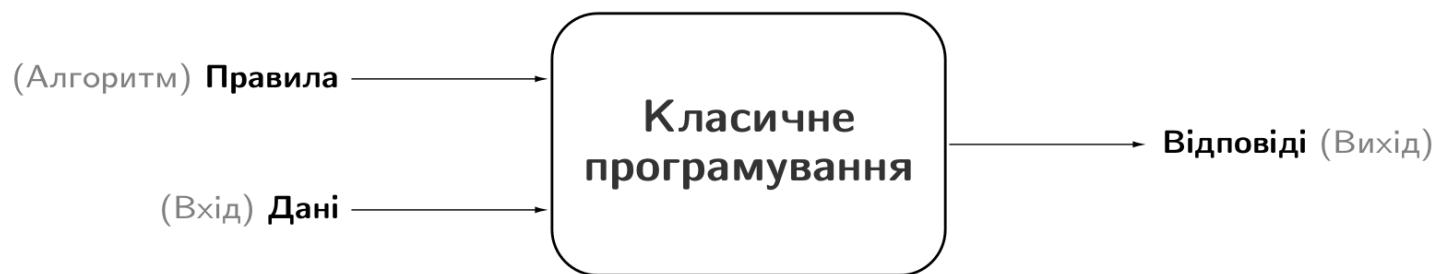


- Досвід (дані): ігри в які грає програма сама з собою
- Вимір продуктивності: коефіцієнт виграшу

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 Supervised learning | Напівконтрольоване навчання Semi-supervised learning | Неконтрольоване навчання Unsupervised learning | Навчання з підкріплення Reinforcement learning |
|---|--|--|--|
| Дани: (x, y) x – приклад, y – мітка | | | |
| Мета – знайти функцію відображення $x \rightarrow y$ | | | |
| Приклад  Це є яблуко. | | | |

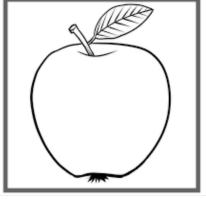
Типи навчання

| Контрольоване навчання Supervised learning | Напівконтрольоване навчання Semi-supervised learning | Неконтрольоване навчання Unsupervised learning | Навчання з підкріплення Reinforcement learning |
|--|--|---|---|
| Дани: (x, y) x – приклад, y – мітка | Дани: (x, y) та $x, (x, y) < x $ x – приклад, y – мітка | | |
| Мета – знайти функцію відображення $x \rightarrow y$ | Мета – знайти функцію відображення або категорію $x \rightarrow y$ | | |
| Приклад  Це є яблуко. | Приклад  Це є яблуко. | | |

Типи навчання

| Контрольоване навчання Supervised learning | Напівконтрольоване навчання Semi-supervised learning | Неконтрольоване навчання Unsupervised learning | Навчання з підкріплення Reinforcement learning |
|--|--|--|---|
| <p>Дани: (x, y) x – приклад, y – мітка</p> | <p>Дани: (x, y) та $x, (x, y) < x$ x – приклад, y – мітка</p> | <p>Дани: x x – приклад, немає міток!</p> | |
| <p>Мета – знайти функцію відображення $x \rightarrow y$</p> | <p>Мета – знайти функцію відображення або категорію $x \rightarrow y$</p> | <p>Мета – знайти правильну категорію.</p> | |
| <p>Приклад</p>  <p>Це є яблуко.</p> | <p>Приклад</p>  <p>Це є яблуко.</p> | <p>Приклад</p>  <p>Цей об'єкт схожий на інший.</p> | |

Типи навчання

| Контрольоване навчання Supervised learning | Напівконтрольоване навчання Semi-supervised learning | Неконтрольоване навчання Unsupervised learning | Навчання з підкріплення Reinforcement learning |
|---|---|---|--|
| Дани: (x, y) x – приклад, y – мітка | Дани: (x, y) та $x, (x, y) < x $ x – приклад, y – мітка | Дани: x x – приклад, немає міток! | Дани: 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 навколошнім середовищем. |
| Приклад  Це є яблуко. | Приклад  Це є яблуко. | Приклад  Цей об'єкт схожий на інший. | Приклад  Їжте це, бо це зробить вас сильнішим. |

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

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

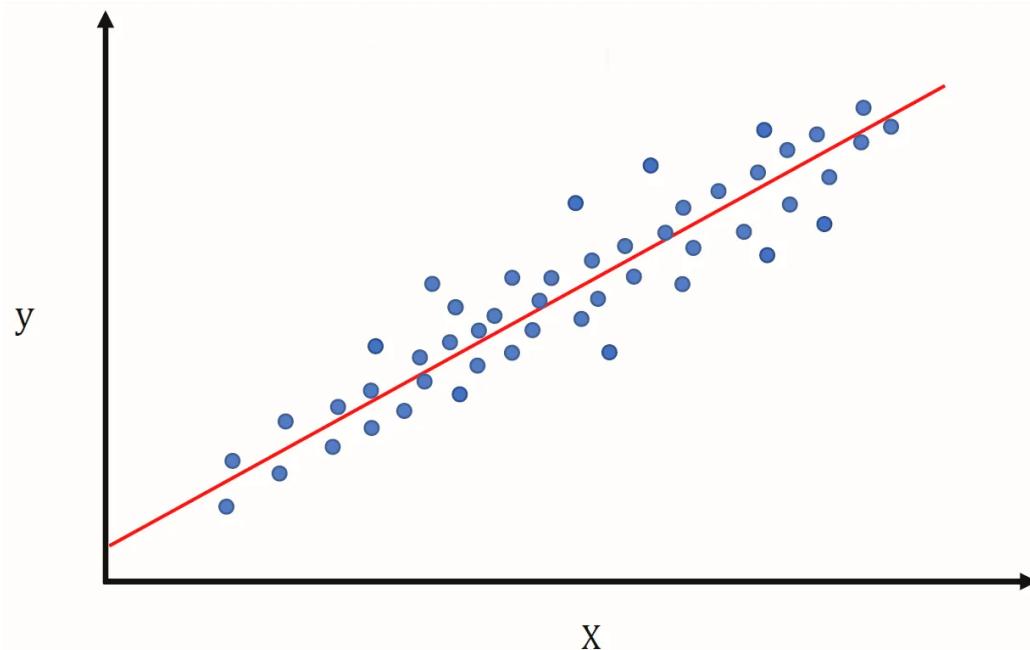
яка буде найкраще пояснювати дані.

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 $y \in \mathbb{R}$.



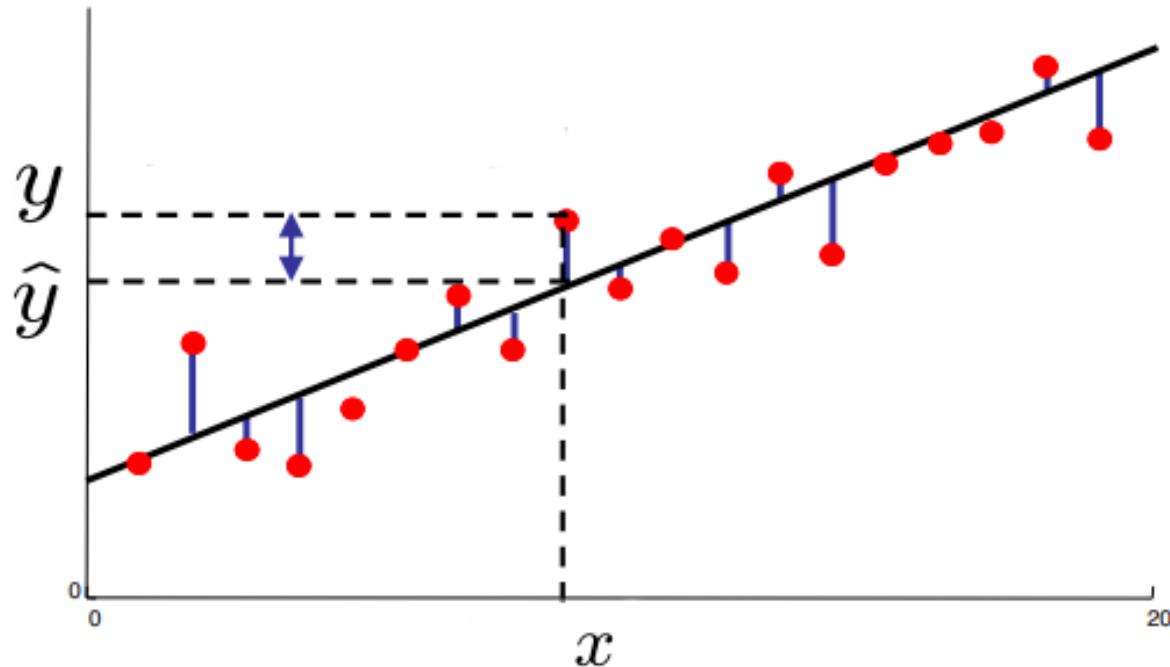
$$\hat{y} = W \cdot X + b$$

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

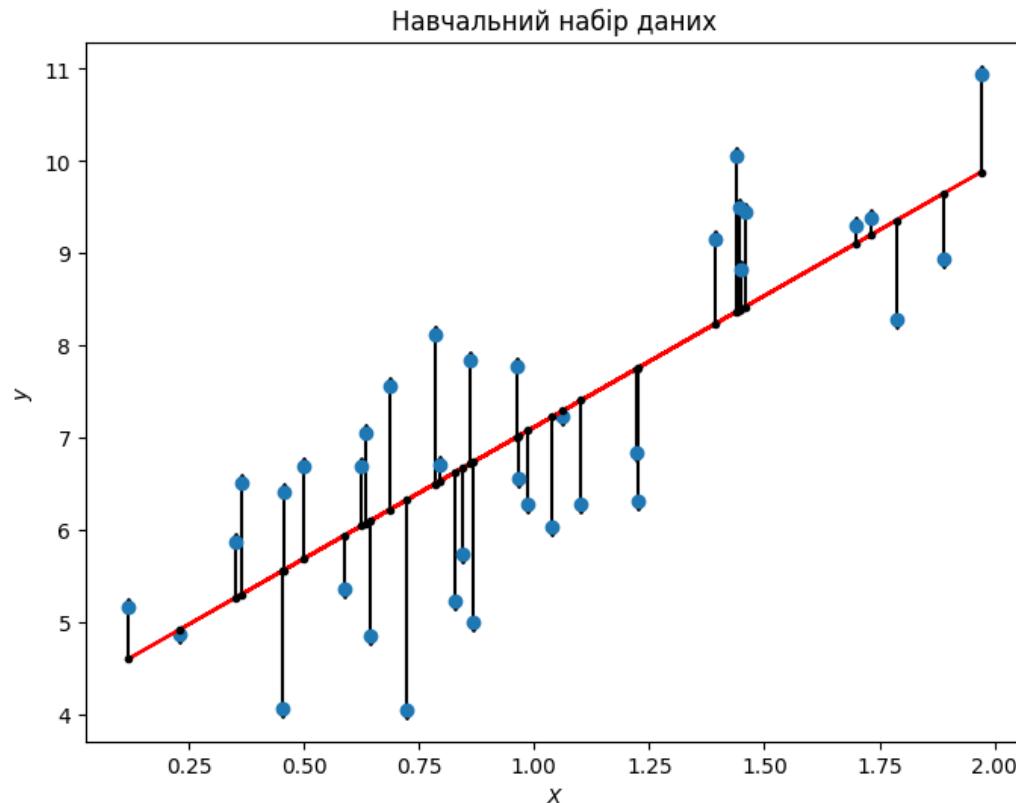
$$L^{(i)}(\hat{y}, y) = (\hat{y}^{(i)} - y^{(i)})^2$$



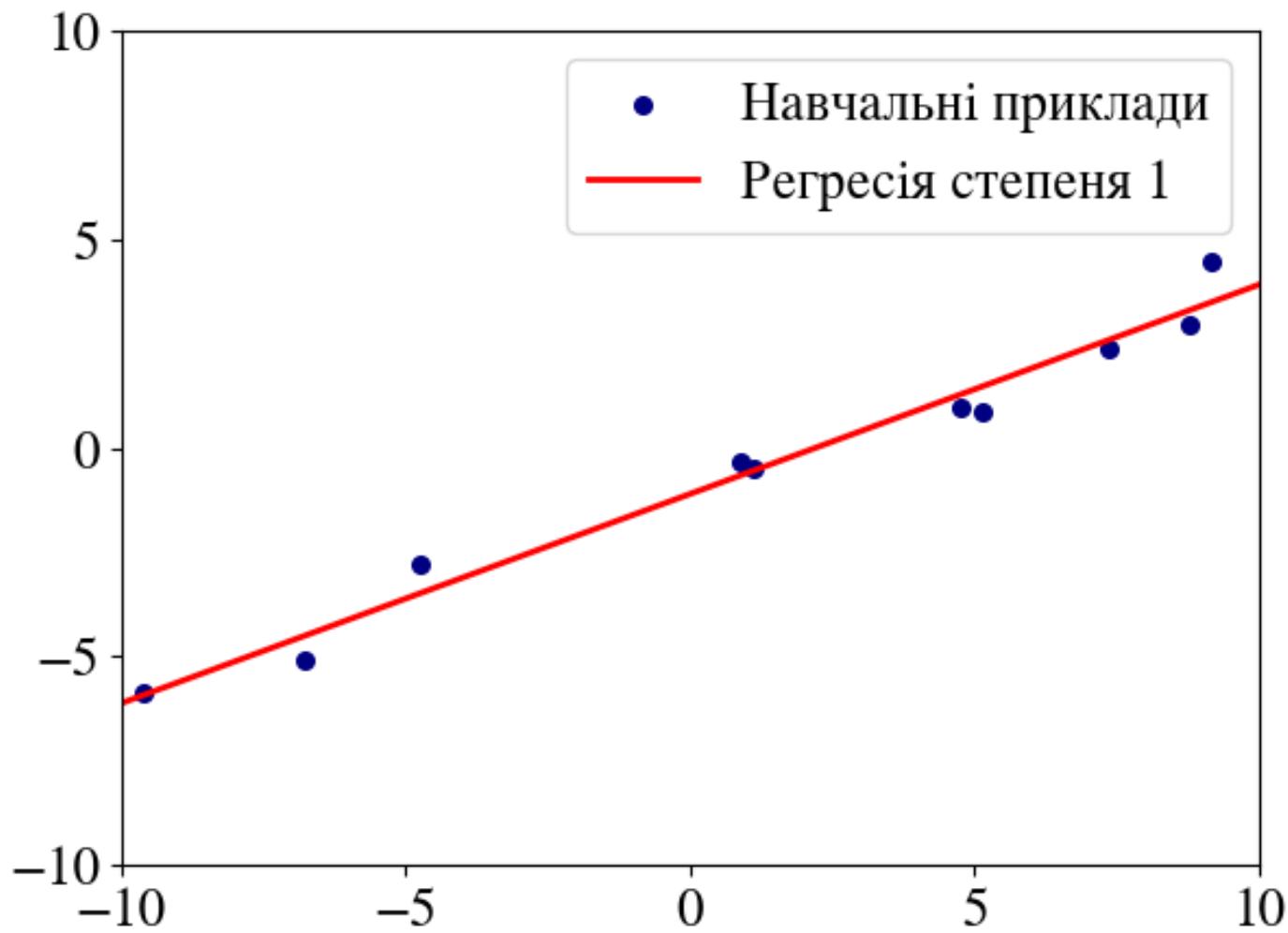
Лінійна регресія

Цільова функція

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

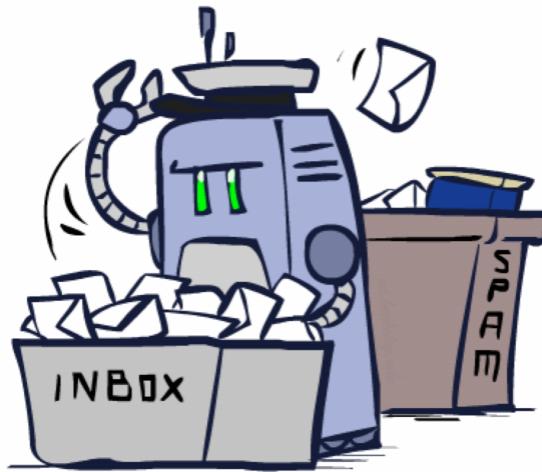


Лінійна регресія



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 поділяє приклади різних класів.

Логістична регресія

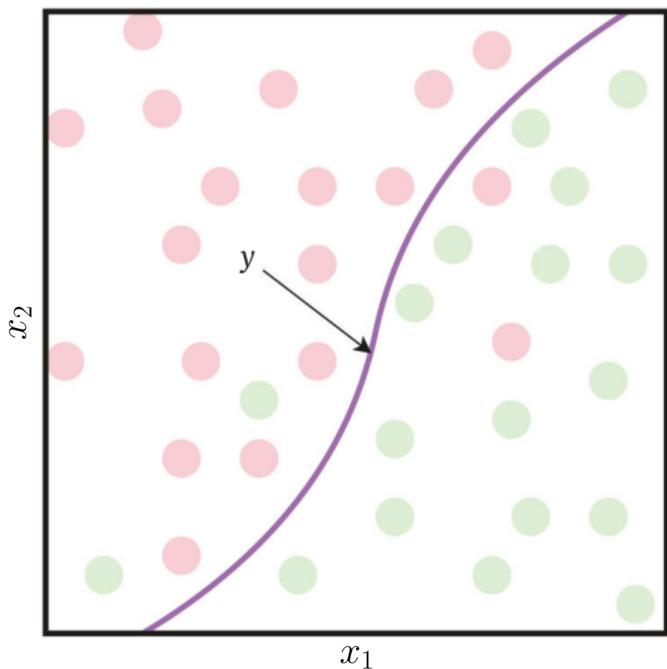
Втрати

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

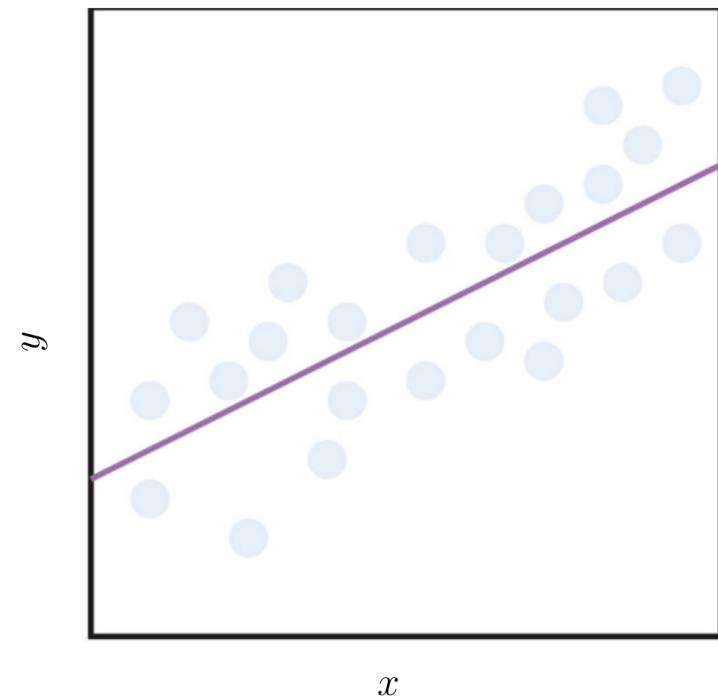
Цільова функція

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

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>Імпульс від аксону (вхід) x_0 Синапс Дендрит ω_0 x_0 $b = \omega_0 x_0$ ω_1 x_1 ω_2 x_2 \vdots ω_m x_m $z = \sum_{n=1}^m \omega_n x_n + b$ Тіло клітини g $g(z)$ Активаційна функція 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?

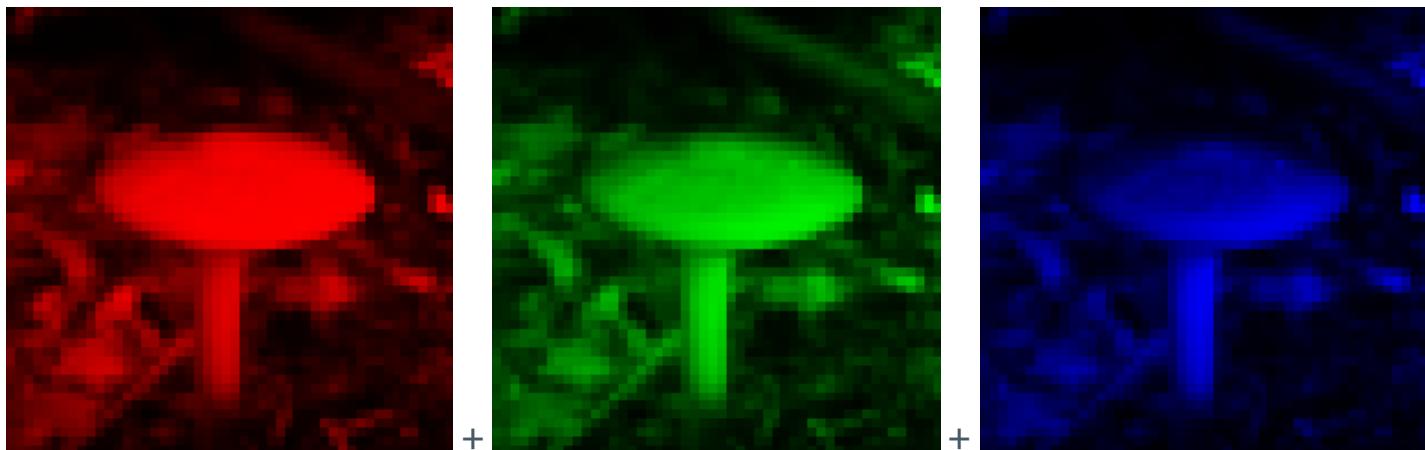
Людський мозок настільки добре інтерпретує візуальну інформацію, що **розврив** між зображенням та його семантичною інтерпретацією (пікселями) важко оцінити інтуїтивно:



Це мухомор.



Це мухомор.



Це мухомор.

```
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.          ]],

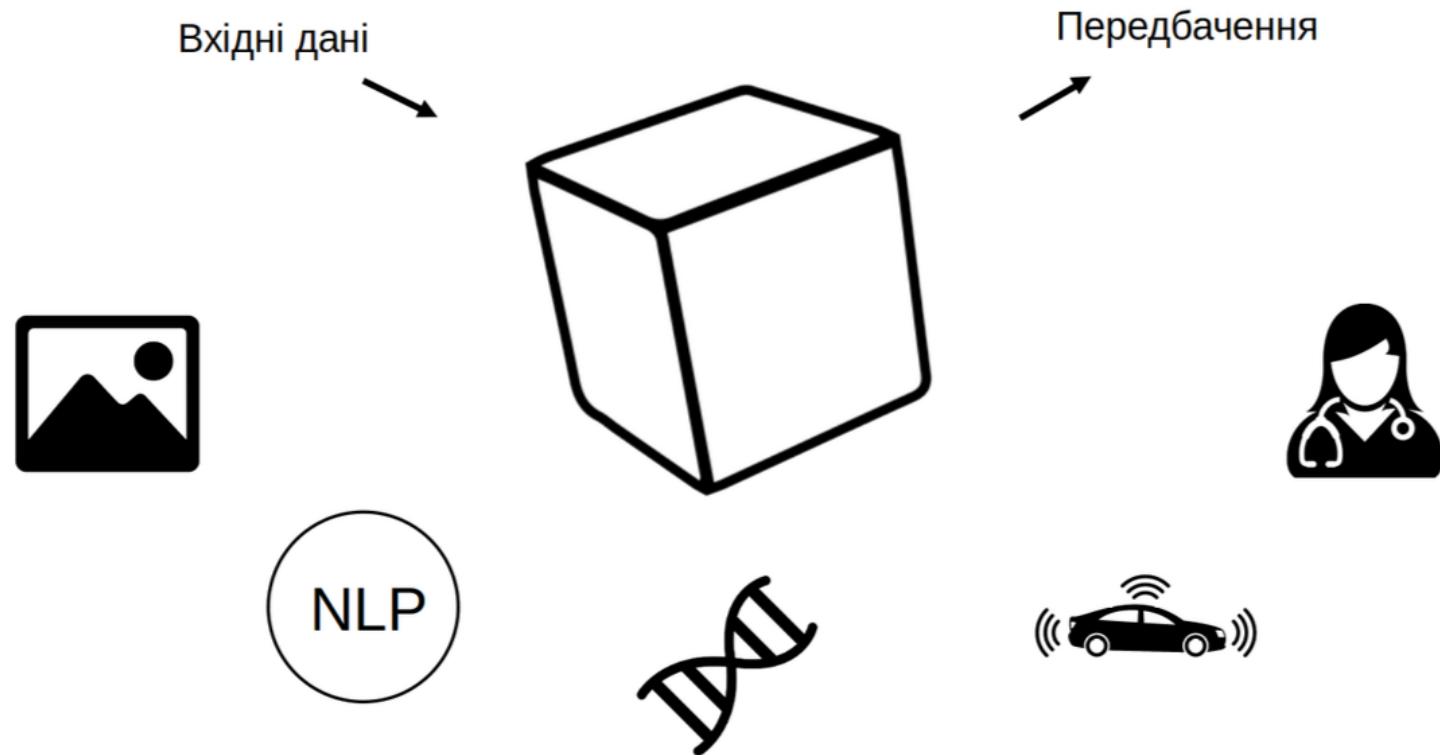
      ...]
```

Це мухомор.

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

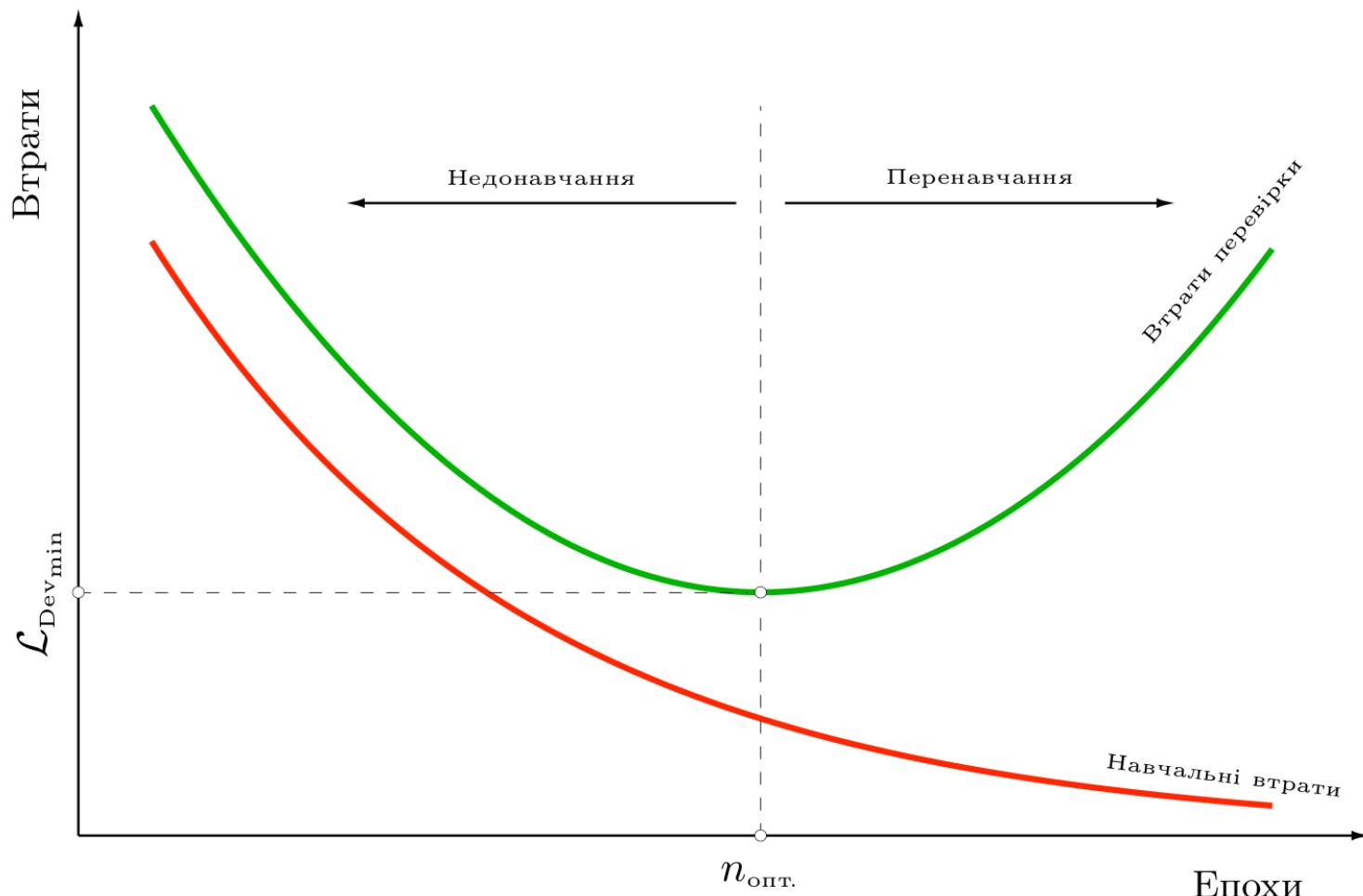
Ознаки - це спостереження, які використовуються для прийняття рішень моделлю.

- 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

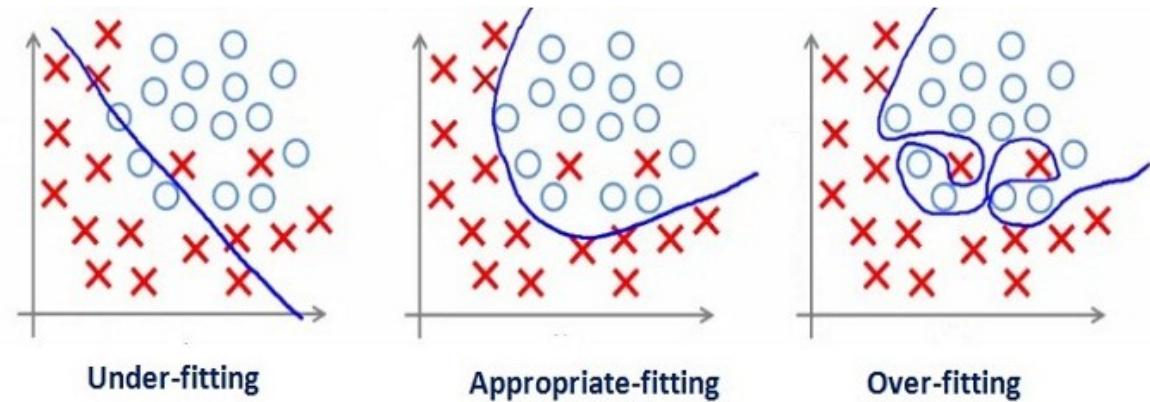
Типи ознак у робототехніці

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

Недонавчання vs перенавчання

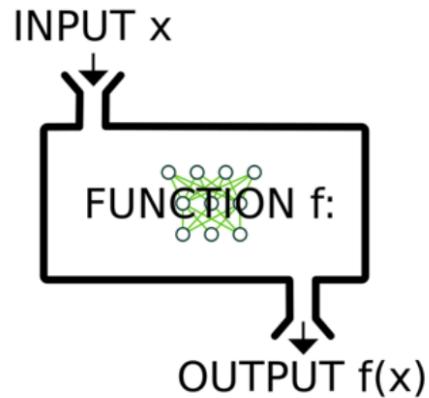


Недонавчання vs перенавчання



Що таке модель?

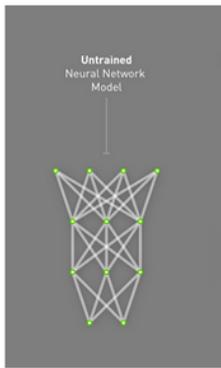
Хоча те, що знаходиться всередині глибинної нейронної мережі, може бути складним, за своєю суттю це просто функції. Вони беруть певні вхідні дані: **INPUT x** і генерують деякі вихідні дані: **OUTPUT f(x)**



З чого складається модель?

Компоненти моделі

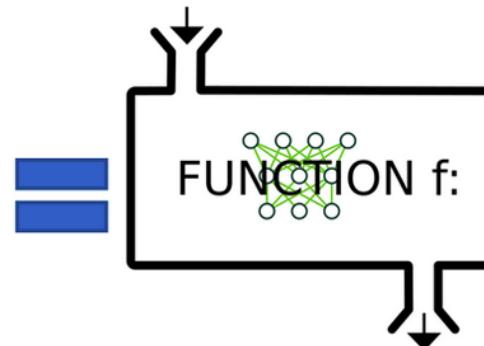
Архітектура мережі = [deploy.prototxt](#)



Навчені ваги = ***.caffemodel



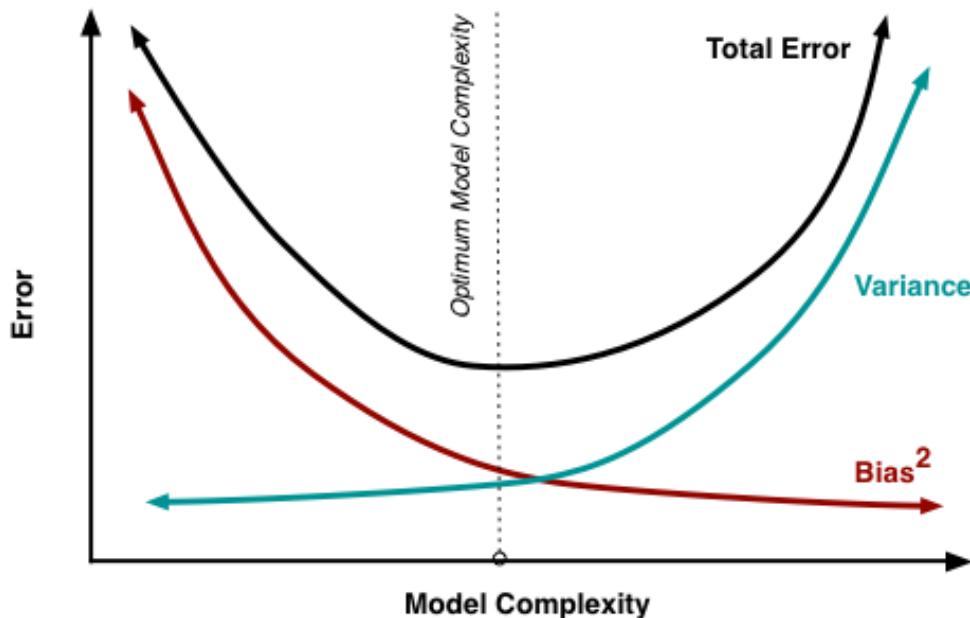
Модель



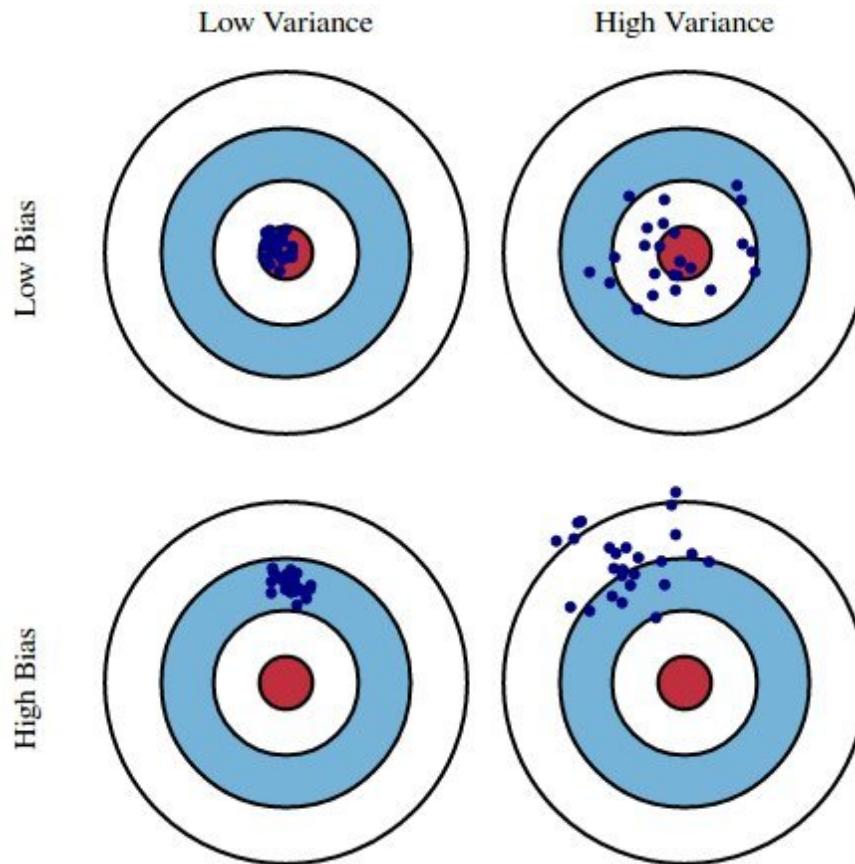
Джерела помилок моделі

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

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

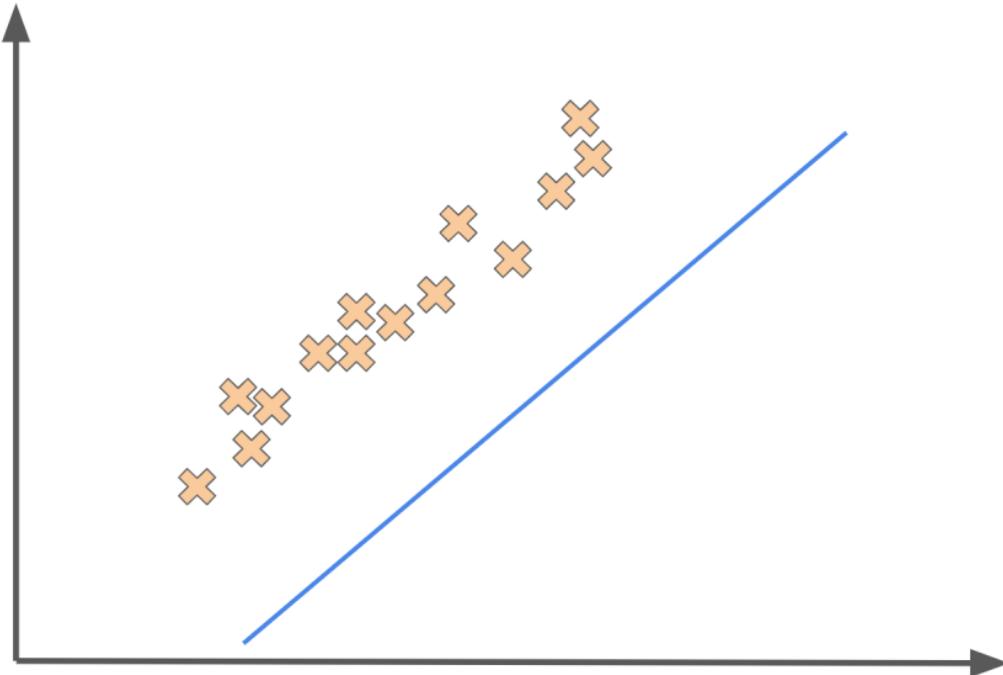


Інтуїція



Інтуїція

Великий зсув



Області застосування та успіхи ШІ



Detectron2: A PyTorch-based modular object detection li...

Share



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



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



Share



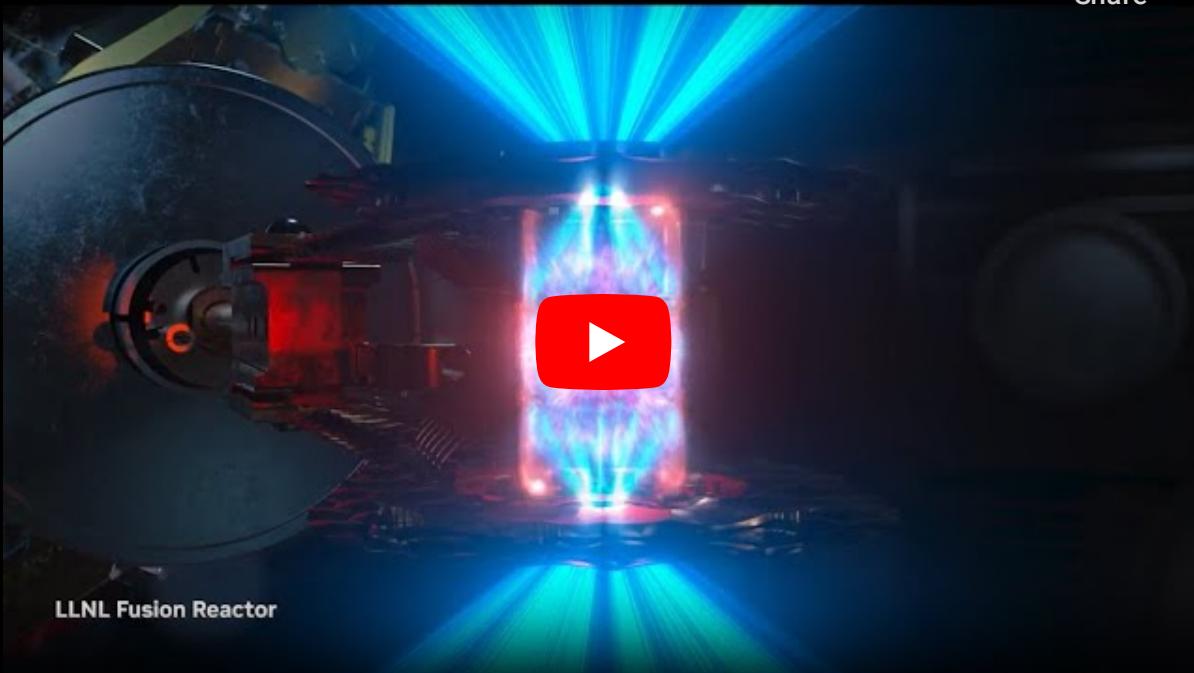
Створення автономних автомобілів (Waymo, 2022)



Powering the Future of Clean Energy | I AM AI



Share



Рушій для розвитку чистої енергії (NVIDIA, 2023)



Camels, Code & Lab Coats: How AI Is Advancing Scienc...



Share



Як ШІ розвиває медицину (Google, 2023)

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



Findings:

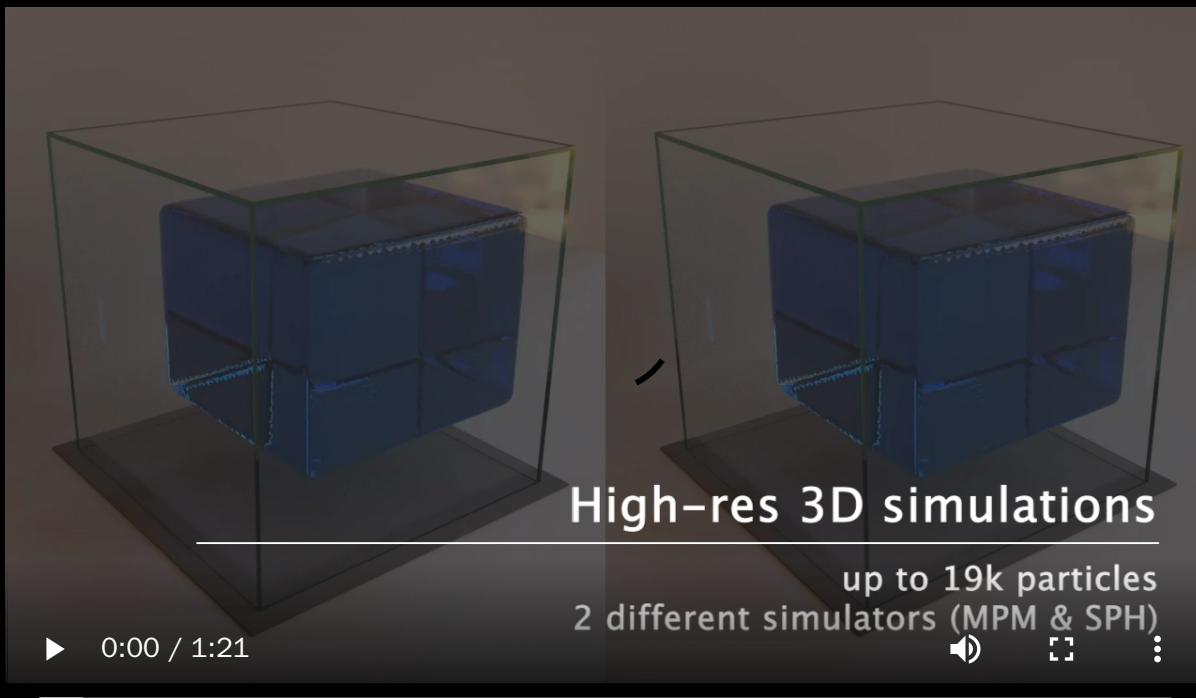
- Devices: None.
- Lungs: No pneumothorax. No substantial pleural effusion. Lungs appear clear.
- Cardiomediastinal: Normal heart size. Mediastinal contours within normal limits.
- Other: No acute skeletal abnormality.

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).



Physics simulation (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

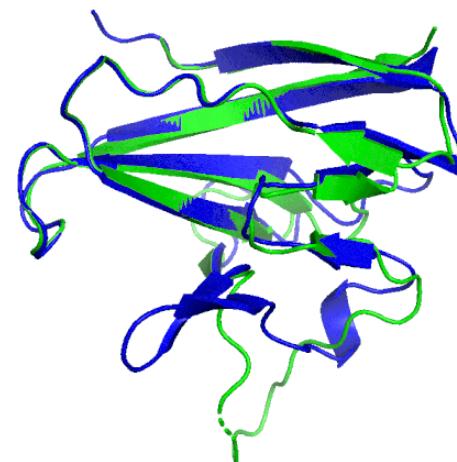
John Jumper , Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishabh Jain, 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^{1,2,3,4}, the structures of around 100,000 unique proteins have been determined⁵, but this represents a small fraction of the billions of known protein sequences^{6,7}. 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’⁸—has been an important open research problem for more than 50 years⁹. Despite recent progress^{10,11,12,13,14}, existing methods fall far short of atomic accuracy, especially when no homologous structure is available. Here we provide the

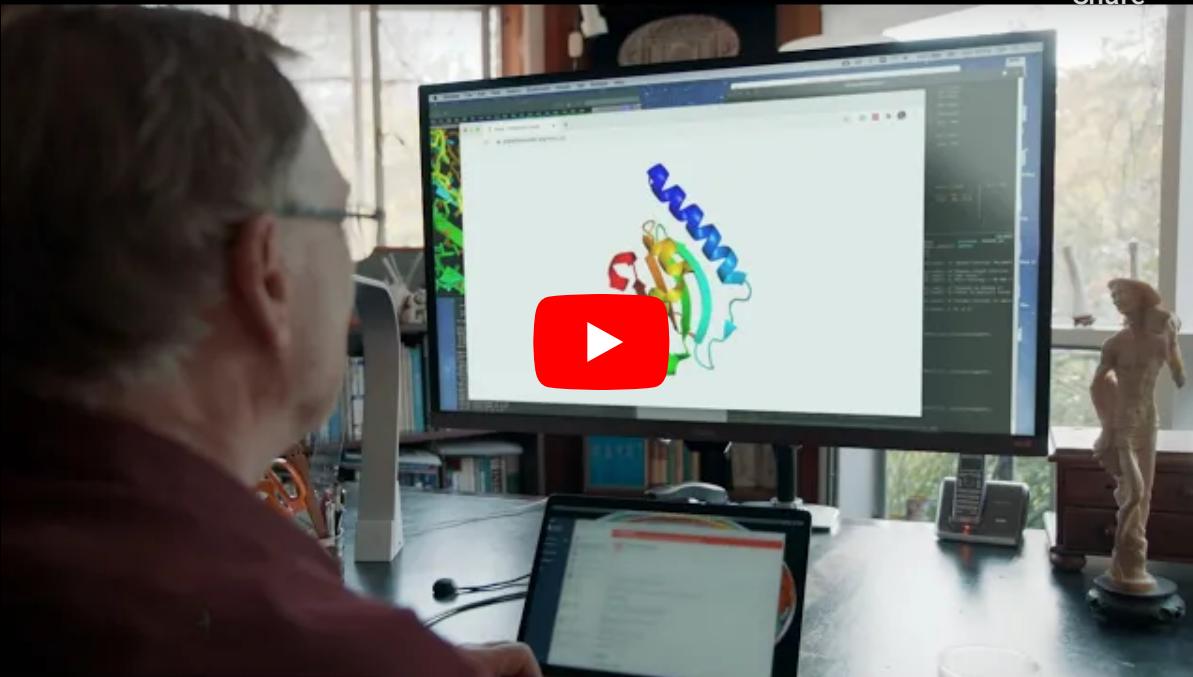




AlphaFold: The making of a scientific breakthrough



Share



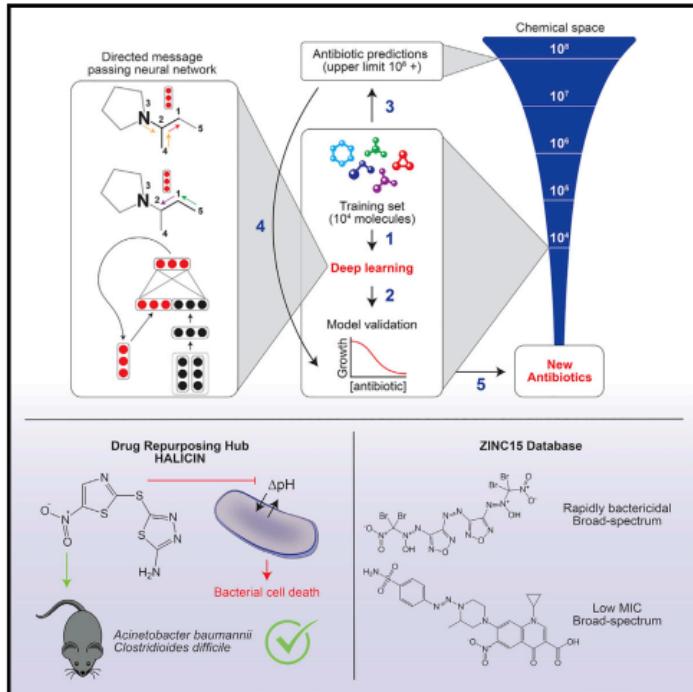
AI for Science (Deepmind, AlphaFold, 2020)

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

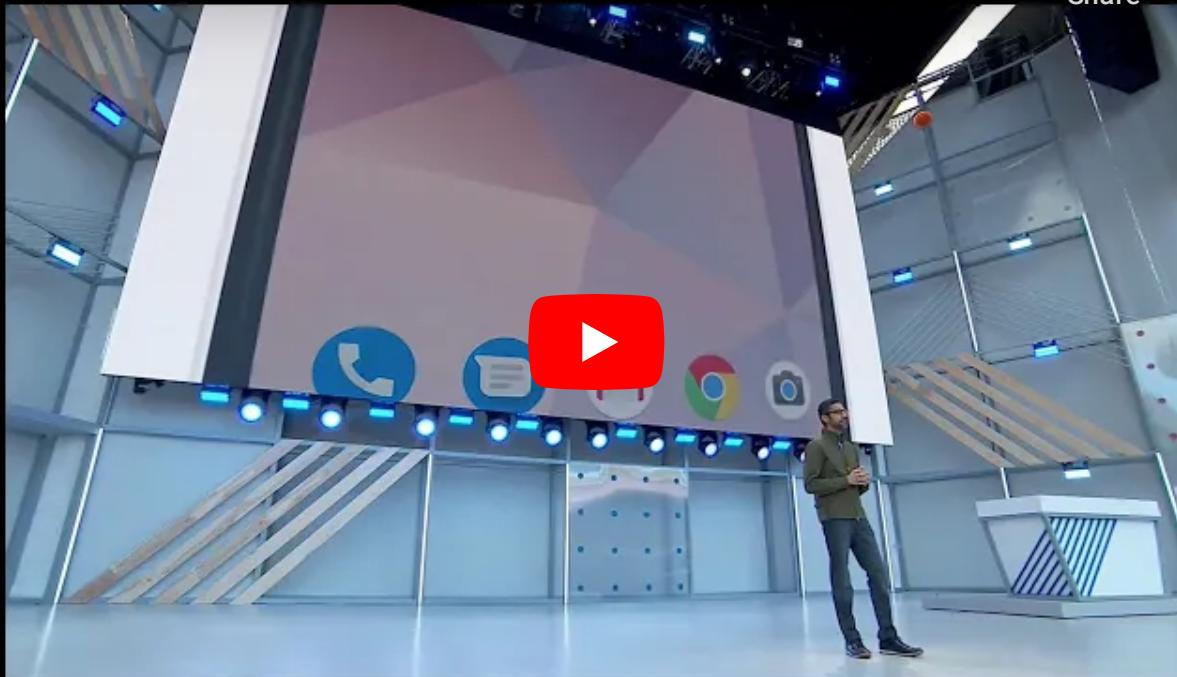
A trained deep neural network predicts antibiotic activity in molecules that are structurally different from known antibiotics, among which Halicin exhibits efficacy against broad-spectrum bacterial infections in mice.



Google Assistant will soon be able to call restaurants an...



Share



Speech synthesis and question answering (Google, 2018)

T

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



Share

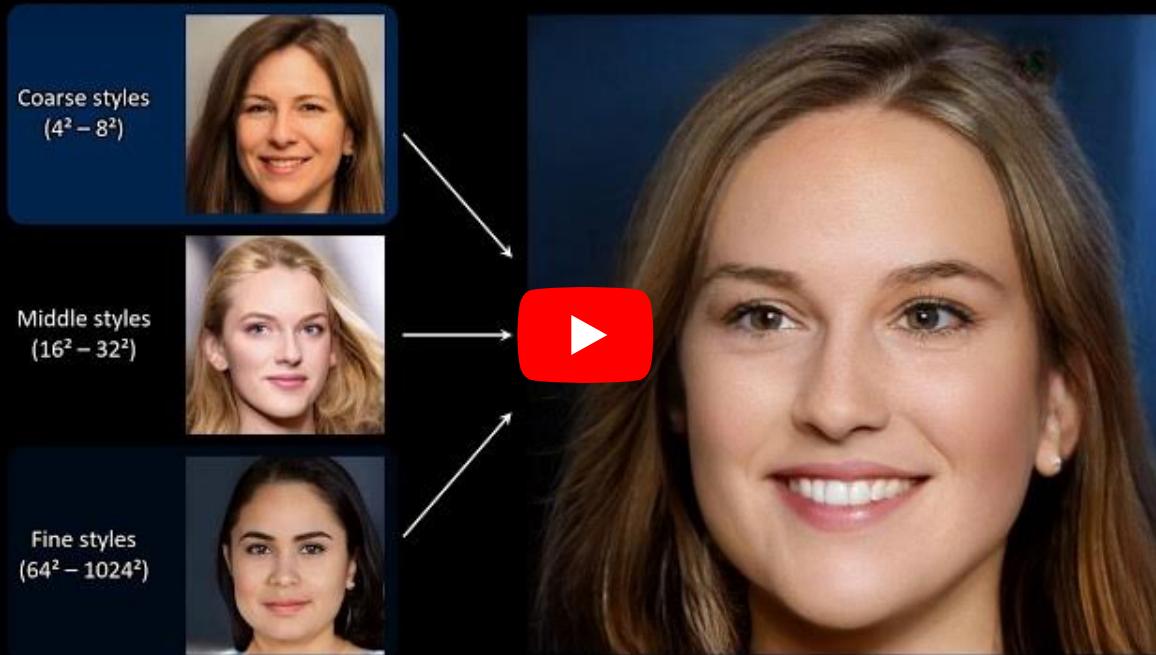


Image generation (Karras et al, 2018)



DALL-E 2 Explained



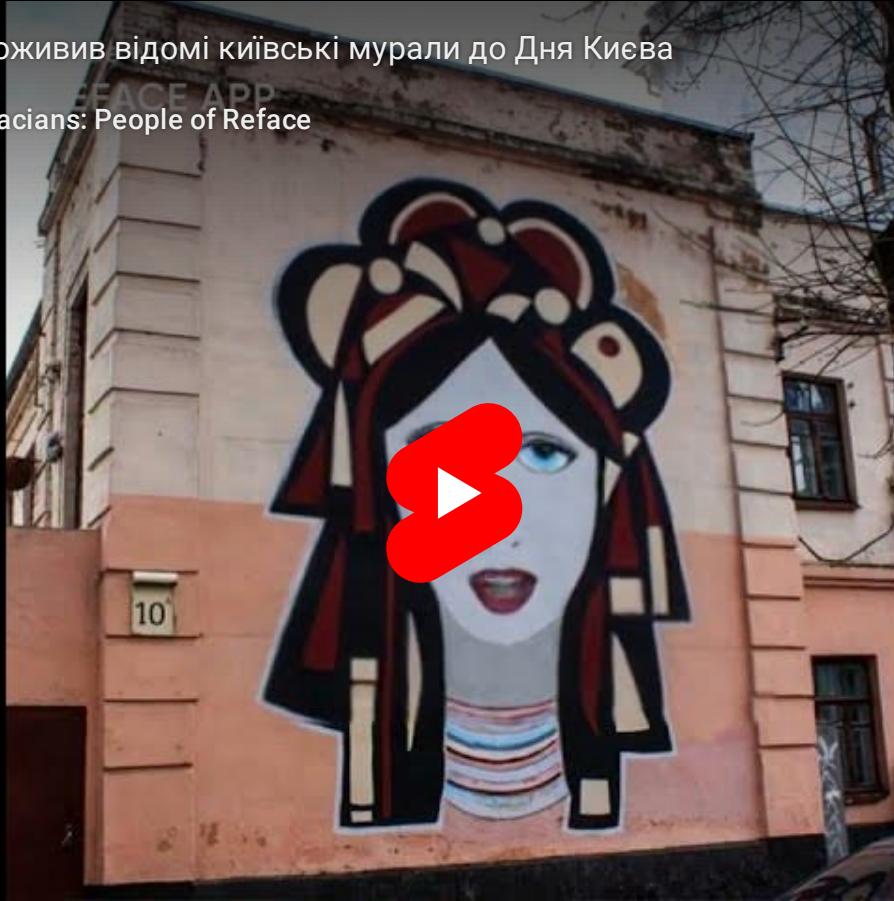
Share



Image generation and AI art (OpenAI, 2022)

Reface оживив відомі київські мурали до Дня Києва

⊕ Refacians: People of Reface



Reface оживив відомі київські мурали до Дня Києва (2021)



Creating a Space Game with OpenAI Codex



Share



Make it be the size of the rocketship times 0.75



```
text.style.left =  
rocketship.offsetLeft + 'px';  
text.style.top =  
rocketship.offsetTop + 'px';  
  
document.body.appendChild(text);  
xSpeed = 20;  
setInterval(function() {  
    xSpeed = 5;  
  
    document.body.removeChild(text);  
}, 250);  
};  
  
/* Now add an image of an  
asteroid:  
https://d.newsweek.com/en/full/1721338/asteroid.jpg?w=1600&h=1600&q=88&f=9d82d35c9de96a82b3fcacf7705eb325b */  
var asteroid =  
document.createElement('img');  
asteroid.src =  
https://d.newsweek.com/en/full/1721338/asteroid.jpg?w=1600&h=1600&q=88&f=9d82d35c9de96a82b3fcacf7705eb325b;  
document.body.appendChild(asteroid);
```



Write computer code (OpenAI, 2021)

+ New Thread

ChatGPT

| Examples | Capabilities | Limitations |
|--|--|---|
| "Explain quantum computing in simple terms" → | Remembers what user said earlier in the conversation | May occasionally generate incorrect information |
| "Got any creative ideas for a 10 year old's birthday?" → | 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)



"For the last forty years we have programmed computers; for the next forty years we will train them."

Chris Bishop, 2020.

Кінець

Література

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