深度學習
Deep Learning

Course Introduction

Instructor: 林英嘉 (Ying-Jia Lin)
2025/02/17

Course Website & Slido

113-2

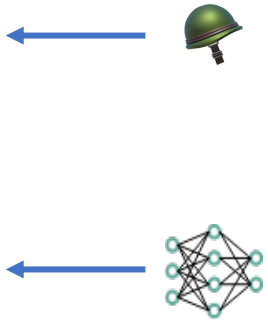
Deep Learning

https://github.com/mcps5601/CGUDL_2025_Spring

2875729
(Link)

About Me

Experience and Education (Time)	
Assistant Professor	Department of Artificial Intelligence, Chang Gung University (2025/02 -)
Postdoctoral Researcher	Department of Computer Science, National Tsing Hua University (2024/09 - 2025/01)
Ph.D.	Department of Computer Science and Information Engineering, National Cheng Kung University (2019/08 - 2024/02)
M.S.	Institute of Biomedical Informatics, National Yang-Ming University (2017/09 - 2019/07)
B.S.	Department of Biomedical Sciences, Chang Gung University (2013/09 - 2017/06)



Outline

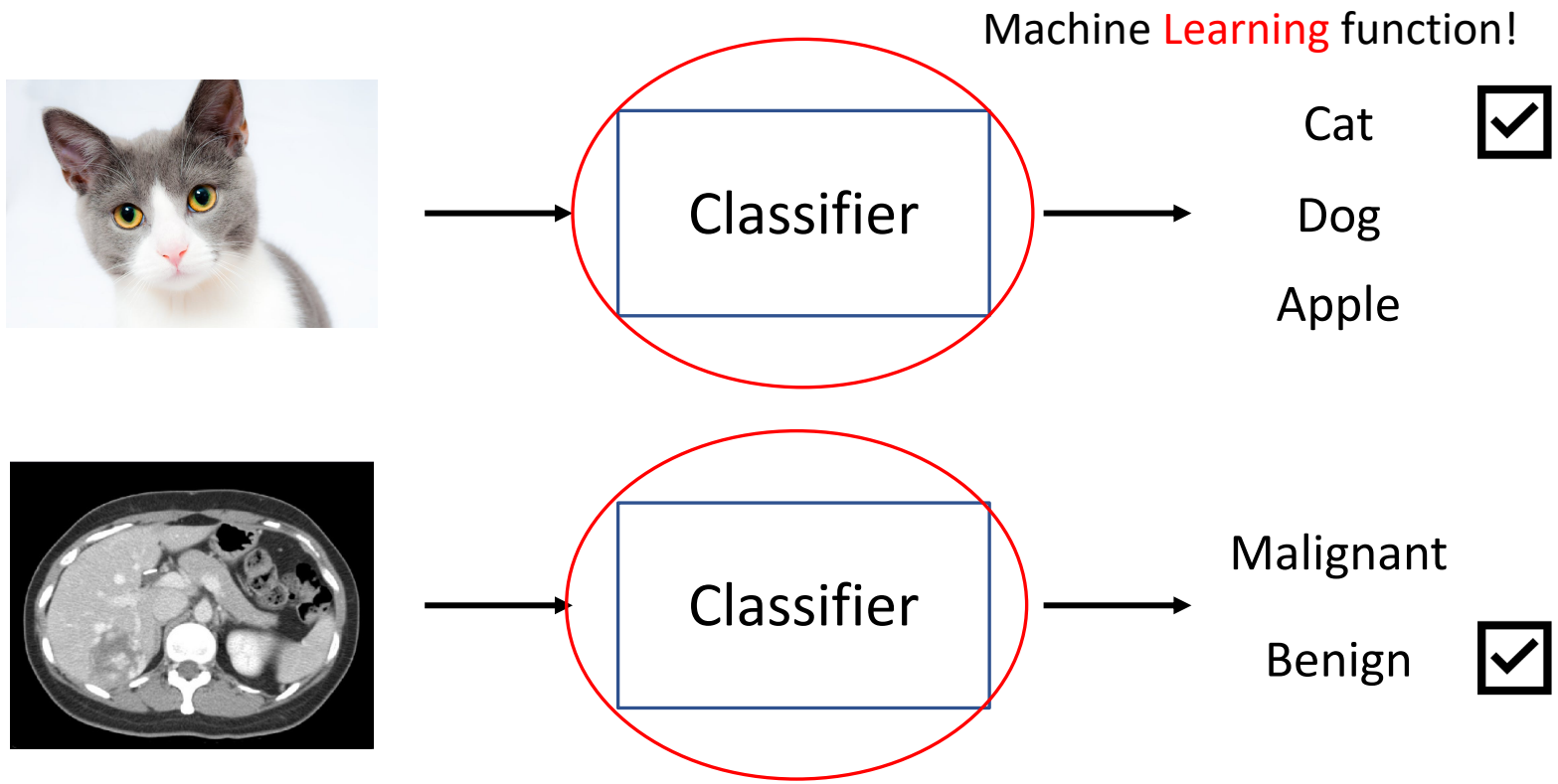
- Introduction to Machine Learning and Deep Learning
- History of Deep Learning
- Course Introduction and requirements

Outline

- AI / Deep Learning 到底是什麼？
- 深度學習為什麼現在變強了？
- 我們這學期要學什麼？

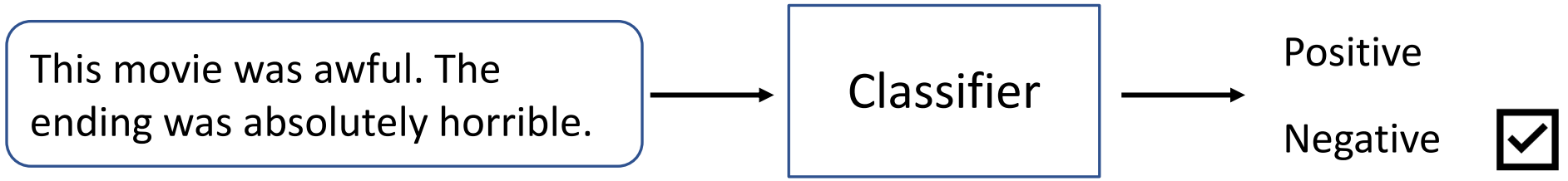
What is Deep Learning?

- We start the concepts from machine learning.

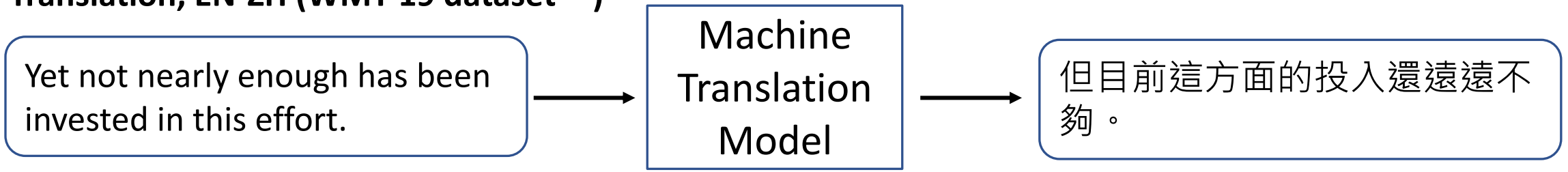


How about DL in Natural Language Processing?

Sentiment Analysis (IMDb dataset [1])



Translation, EN-ZH (WMT-19 dataset [2])



[1] https://huggingface.co/datasets/stanfordnlp/imdb
[2] https://huggingface.co/datasets/wmt/wmt19/viewer/zh-en

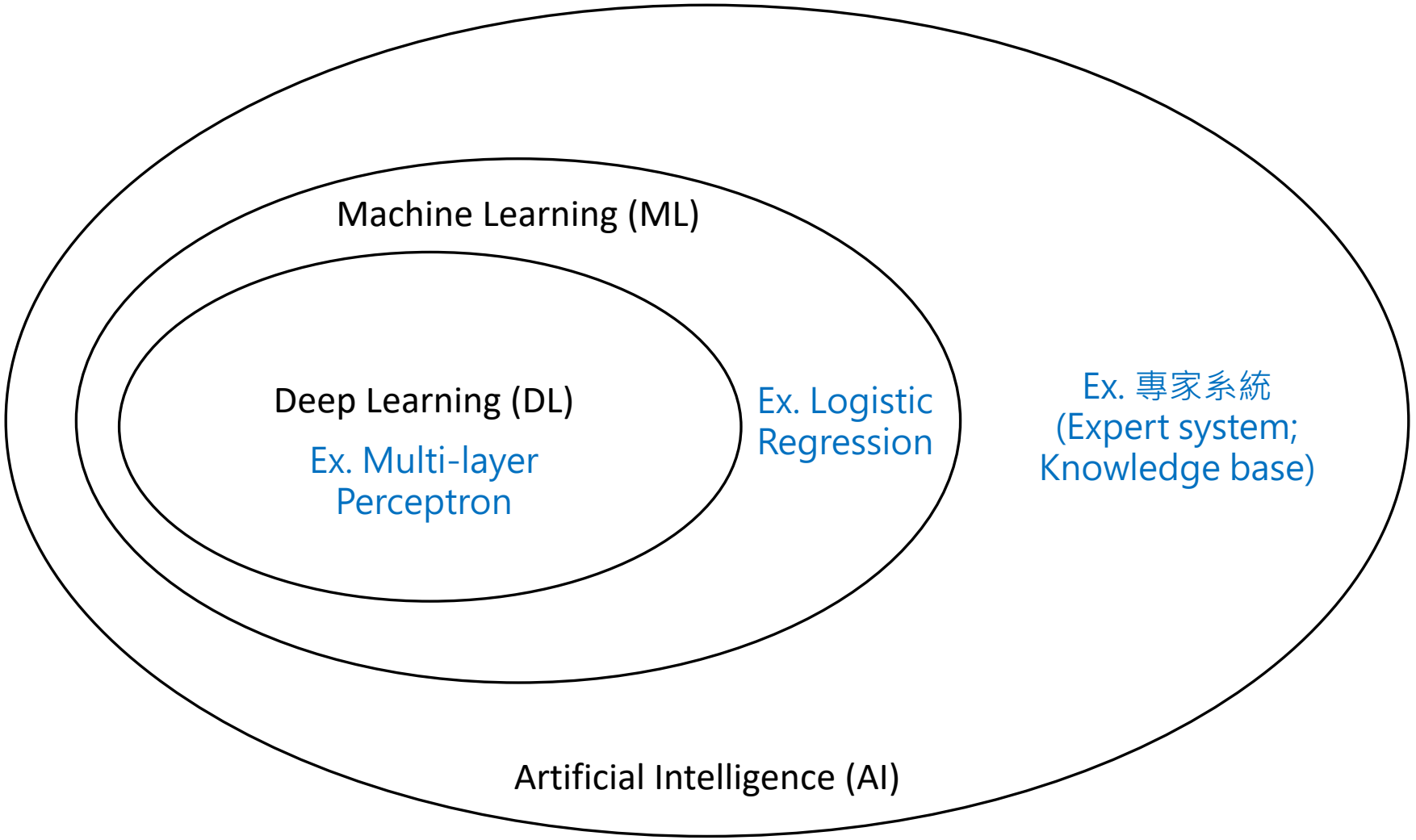
Training a model to play Super Mario

Using reinforcement learning.
Code: https://github.com/jiseongHAN/Super-Mario-RL

[Definition] What is Deep Learning?

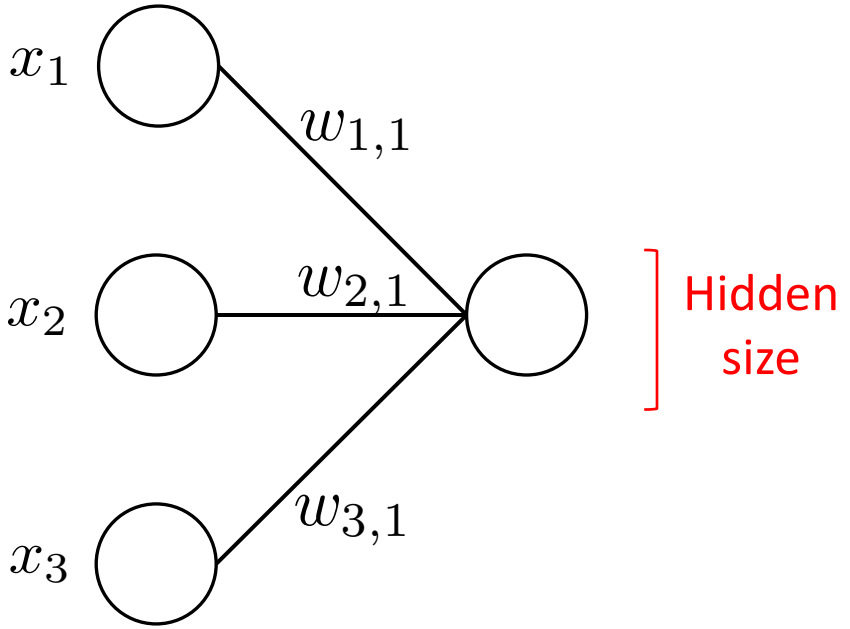
- Deep learning is a machine learning approach that focuses on utilizing artificial neural networks with multiple layers (很多層的類神經網路) to perform tasks, such as classification, generation, and so on.
- Usually, the number of parameters is big in a deep learning model, which results from the increased number of hidden layers and hidden size.

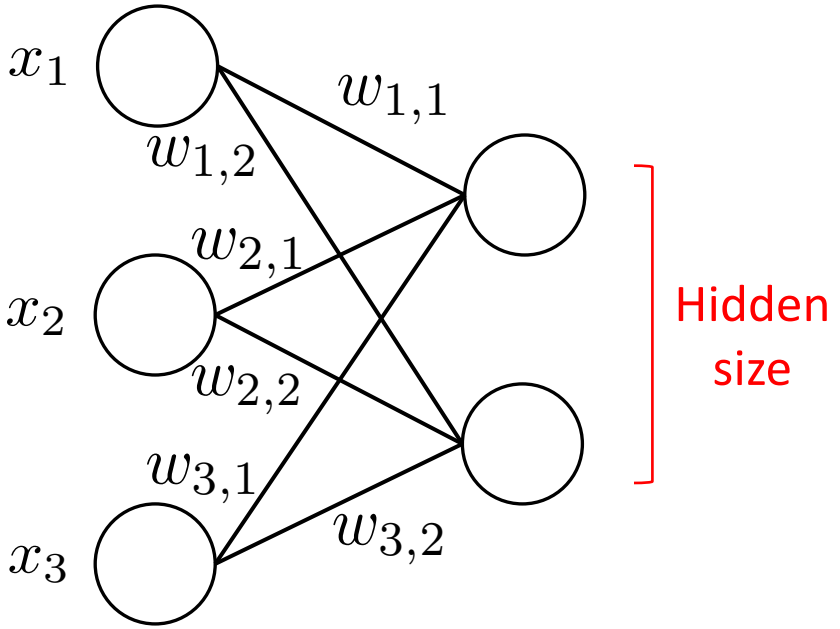
Venn Diagram about AI, ML, and DL



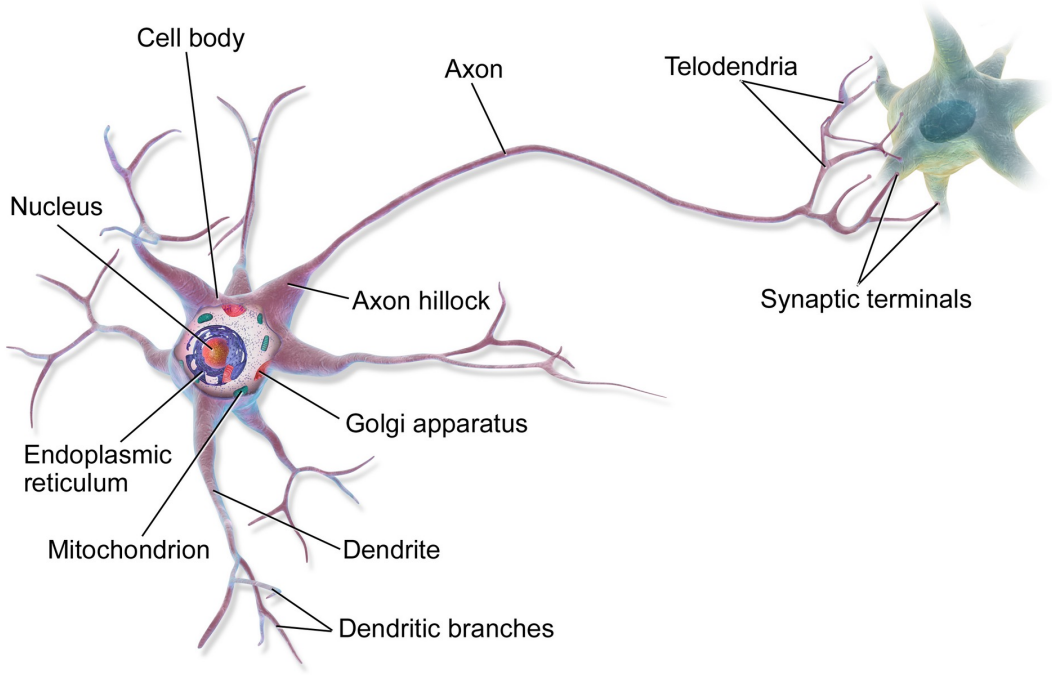
Perceptron (感知機)

Before we talk about multi-layer perceptron (MLP, 多層感知機)





A Biological Neuron vs. Perceptron



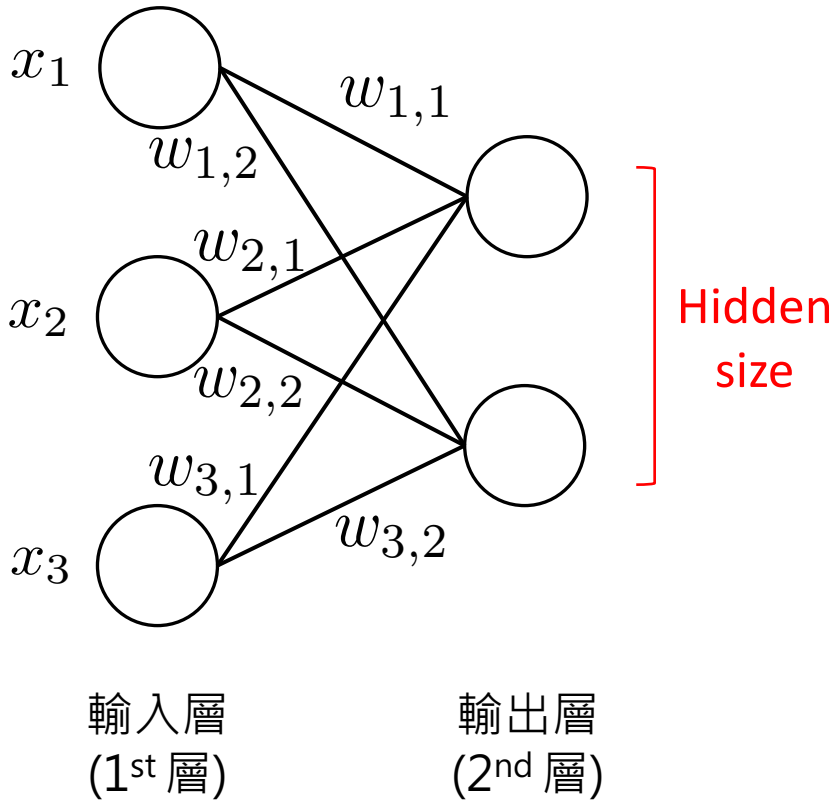
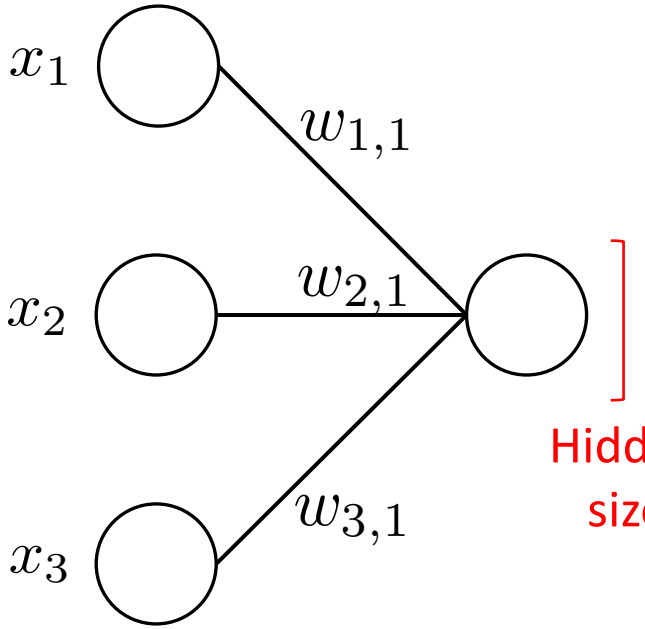


Figure source: https://zh-yue.wikipedia.org/wiki/神經細胞

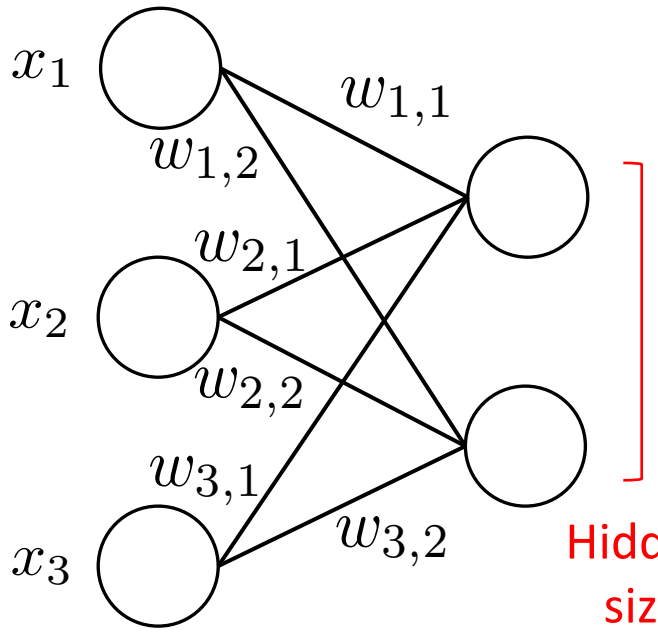
Math for Perceptron using a toy example (1)



$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_{1,1} \\ w_{2,1} \\ w_{3,1} \end{bmatrix} + \text{bias}$$

$\mathbf{R}^{1\times 3}$ $\mathbf{R}^{3\times 1}$ $\mathbf{R}^{1}$

Math for Perceptron using a toy example (2)



輸入層
(1st 層)

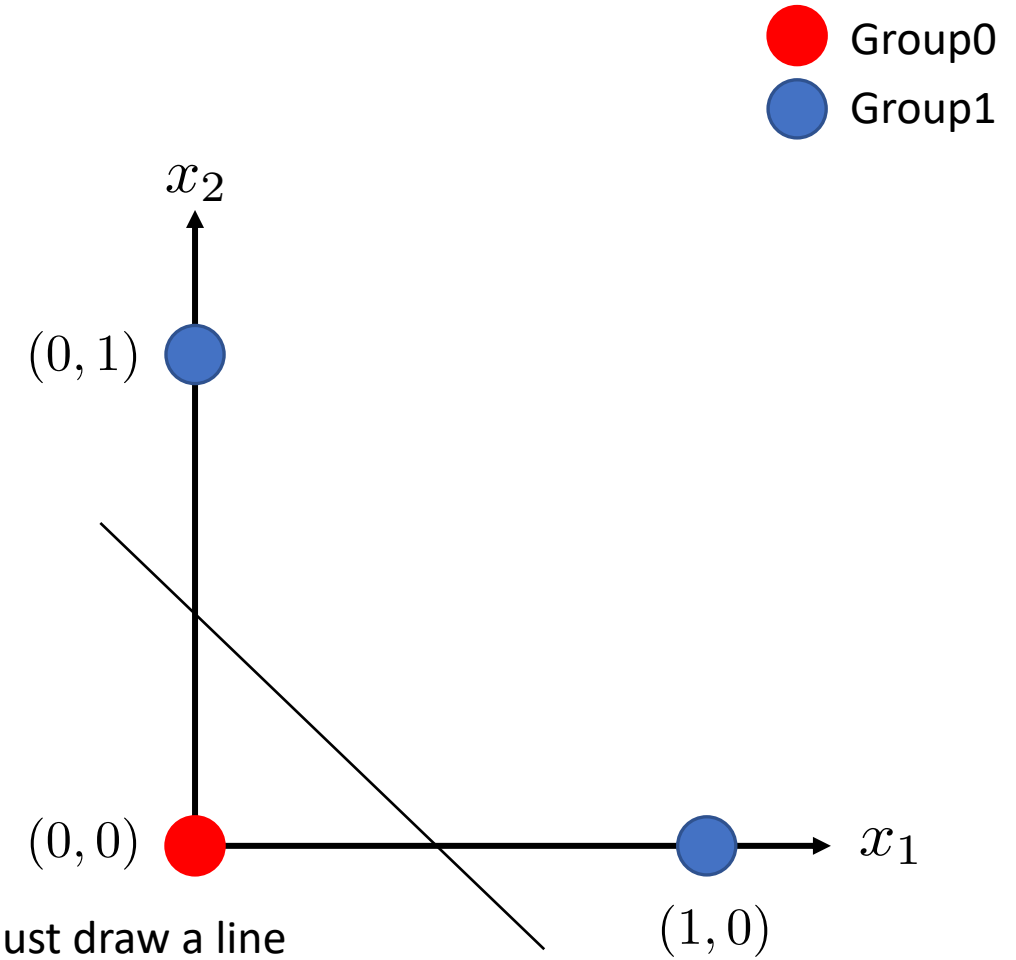
輸出層
(2nd 層)

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \\ w_{3,1} & w_{3,2} \end{bmatrix} + \text{bias}$$

$\mathbf{R}^{1\times 3}$ $\mathbf{R}^{3\times 2}$ $\mathbf{R}^{2}$

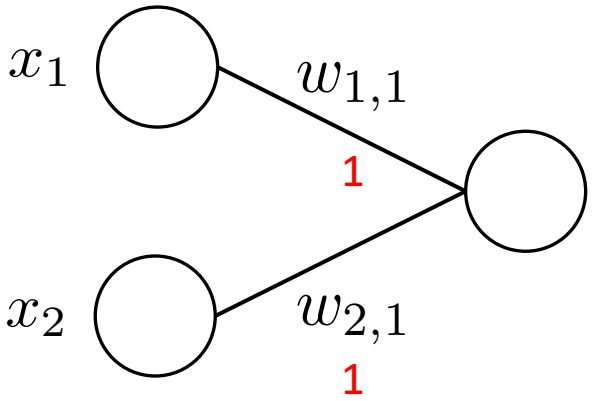
A toy example for classification

x1	x2	Output
0	0	0
0	1	1
1	0	1



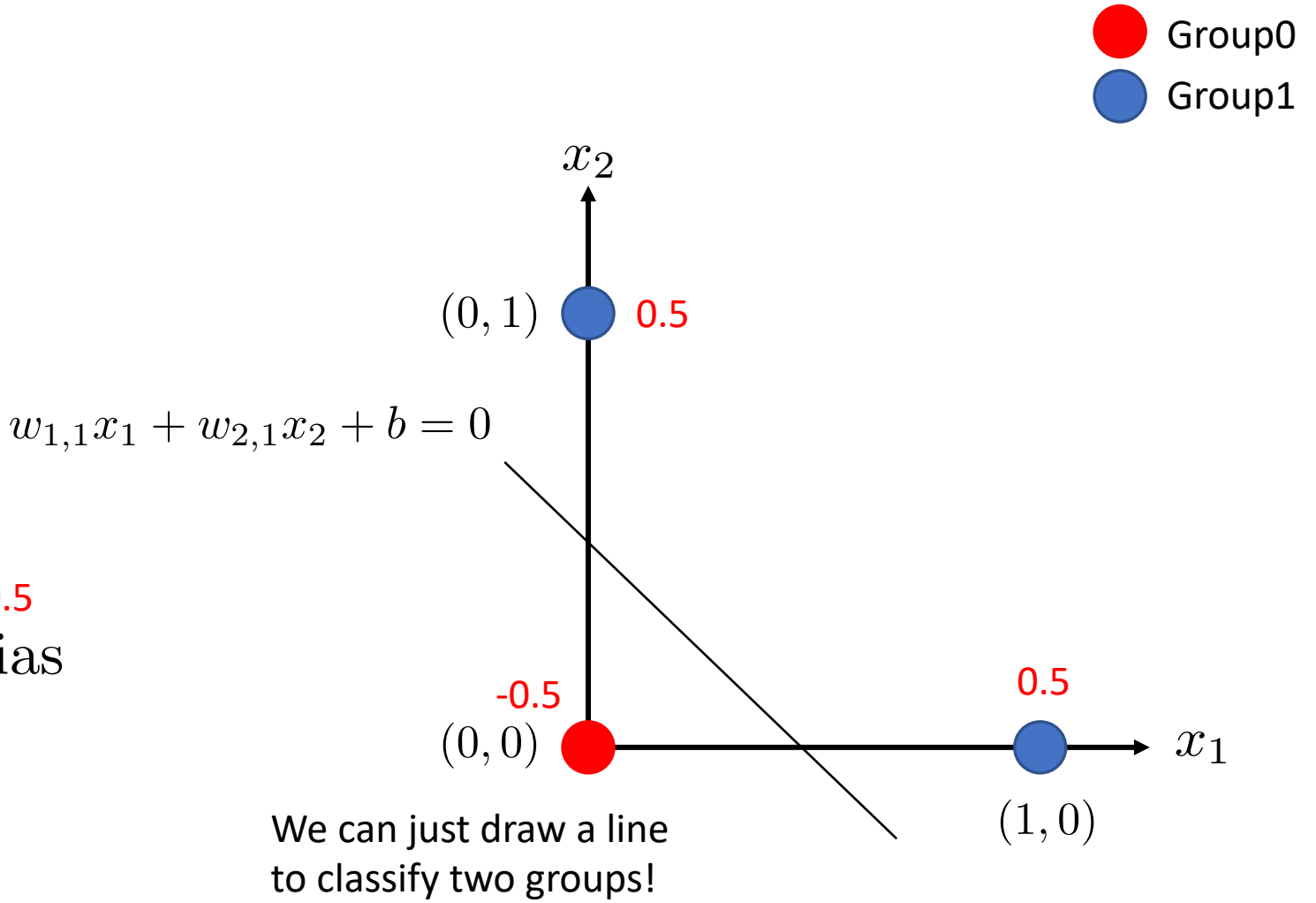
We can just draw a line
to classify two groups!

A toy example for classification with Perceptron



$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} w_{1,1} \\ w_{2,1} \end{bmatrix} + \text{bias}$$

-0.5



We can just draw a line to classify two groups!

XOR Problem

x1	x2	OR
0	0	0
0	1	0
1	0	0
1	1	1

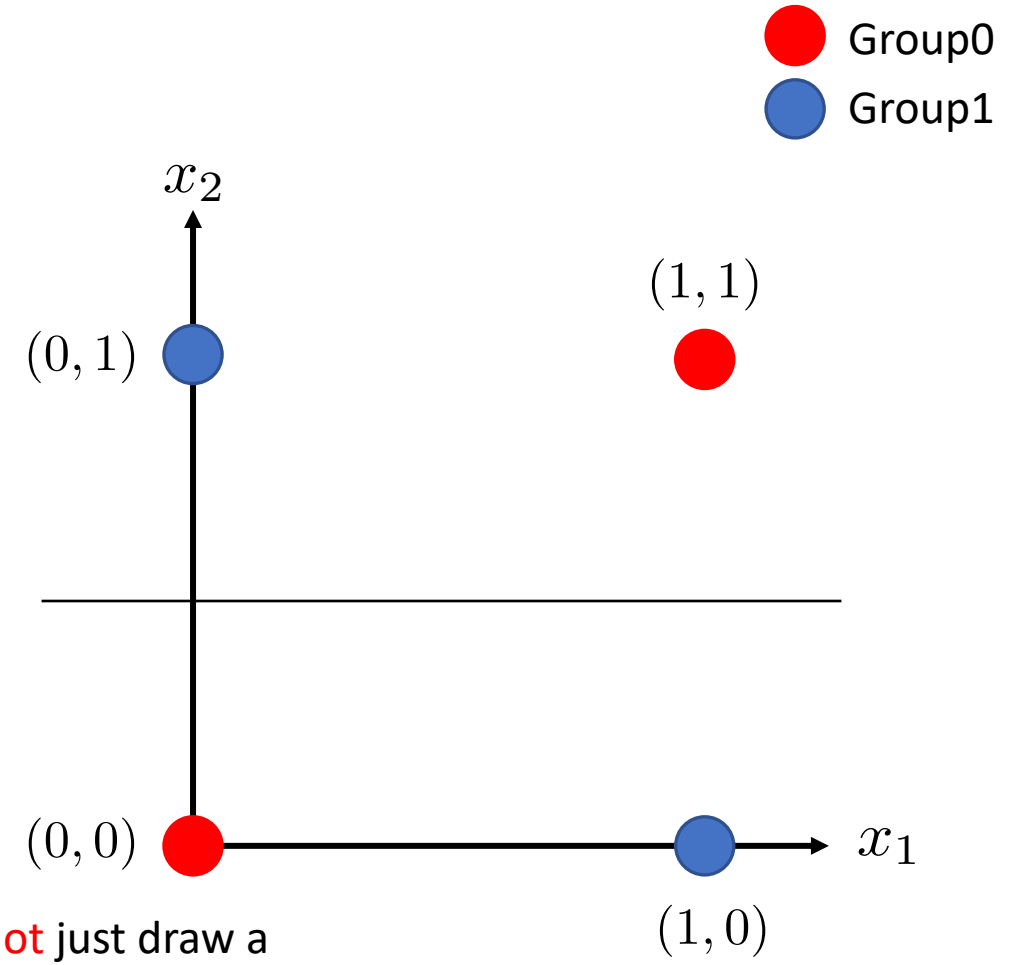
x1	x2	OR
0	0	0
0	1	1
1	0	1
1	1	1

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

XOR Problem

XOR: Exclusive-or

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

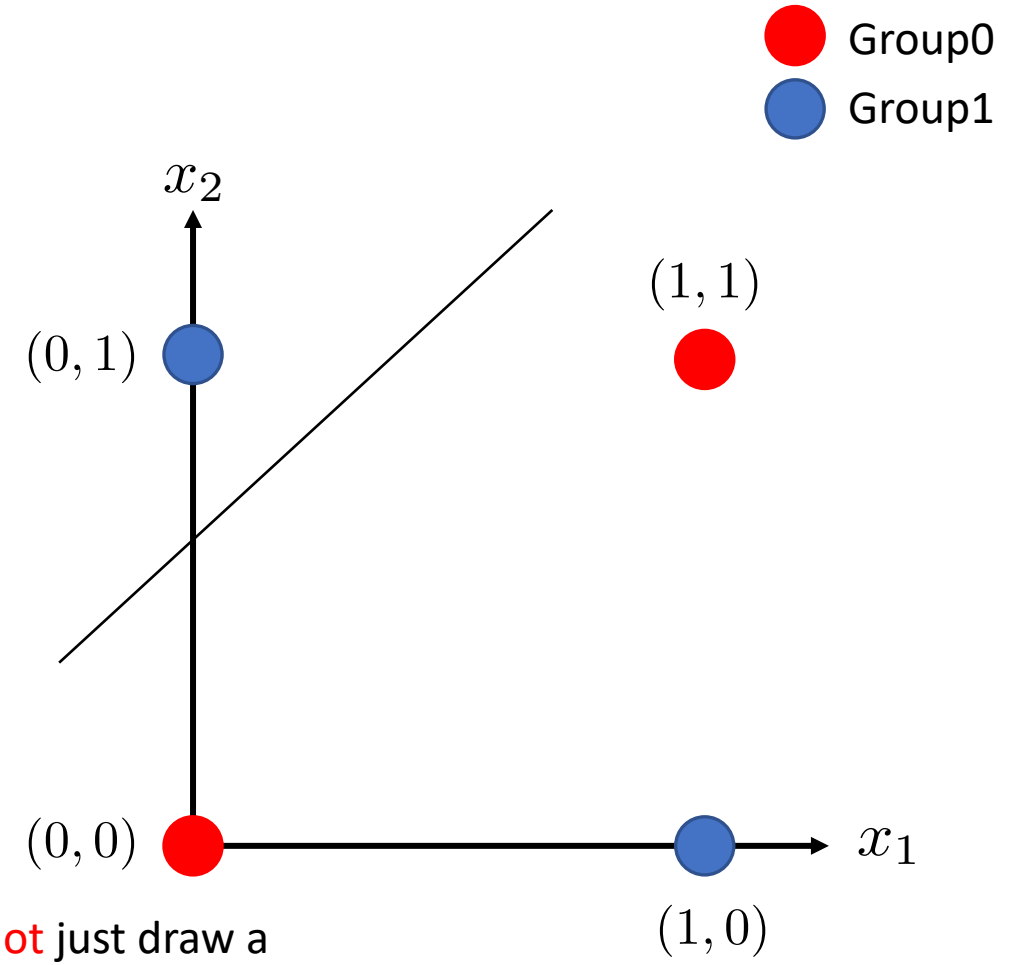


We cannot just draw a line to classify two groups!

XOR Problem

XOR: Exclusive-or

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

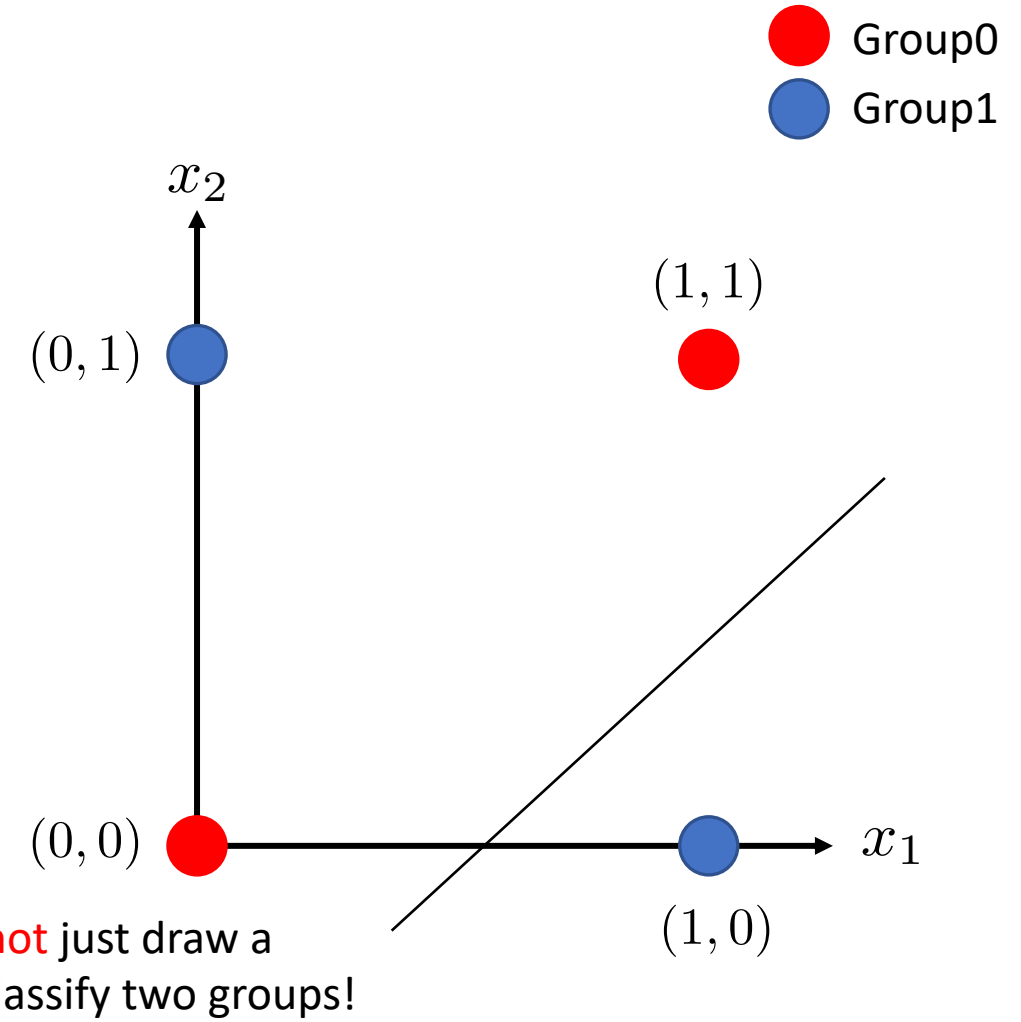


We cannot just draw a line to classify two groups!

XOR Problem

XOR: Exclusive-or

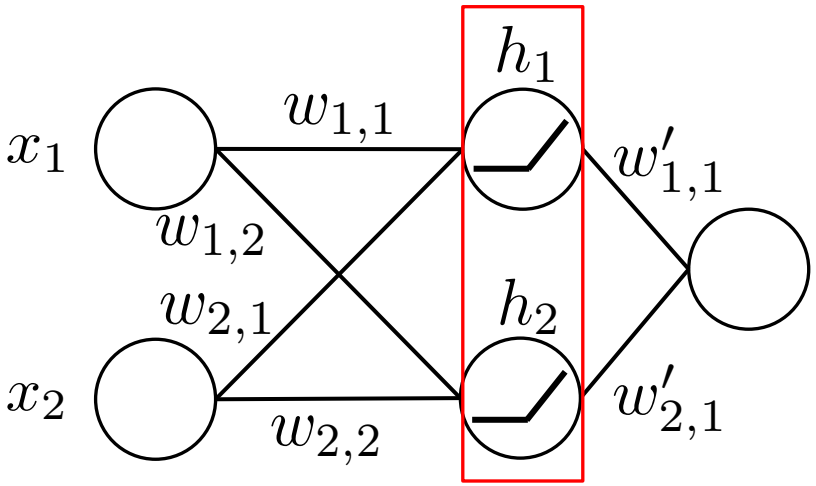
x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

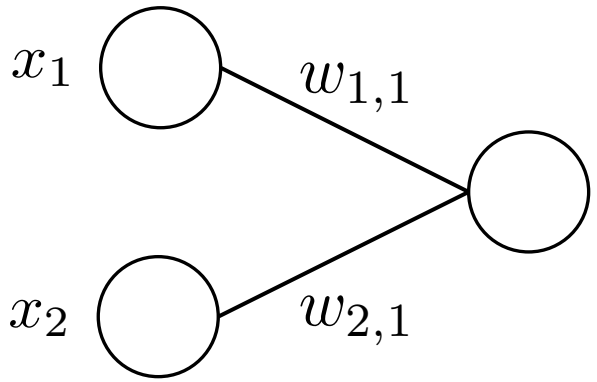


We cannot just draw a line to classify two groups!

MLP for the XOR Problem

ReLU (Rectified Linear Unit)





輸入層
(1st 層)

隱藏層
(2nd 層)

輸出層
(3rd 層)

Perceptron通常沒有隱藏層

ReLU (Rectified Linear Unit)[1][2]

- Non-linear transformation
- Negative values will be transformed to zeros.
- Positive values remain their original ones.

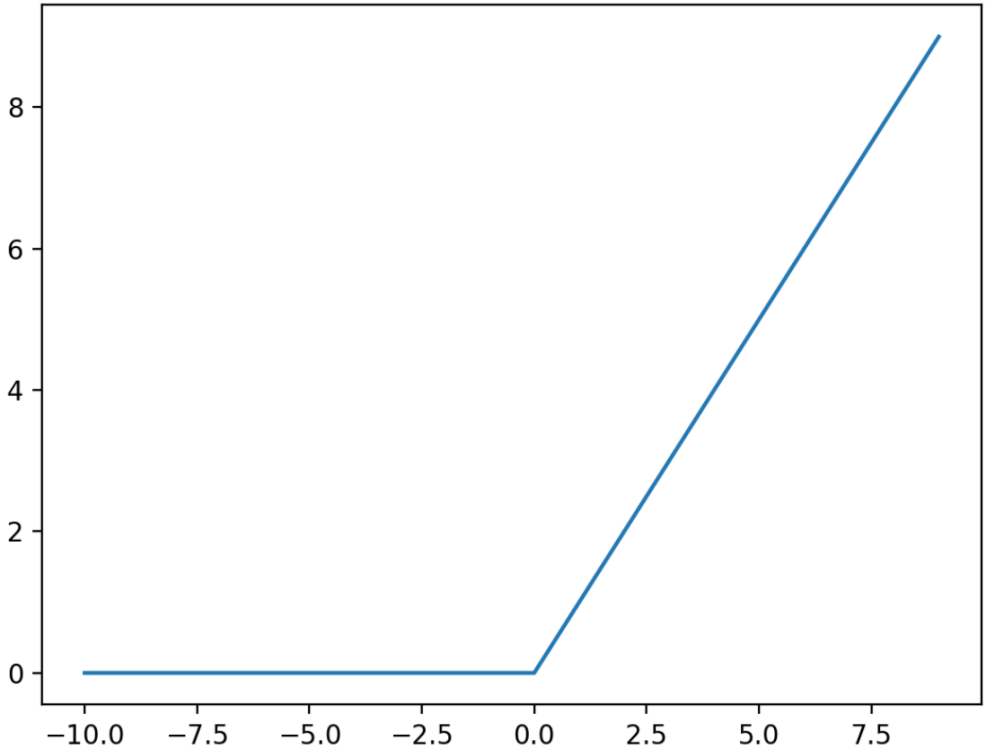
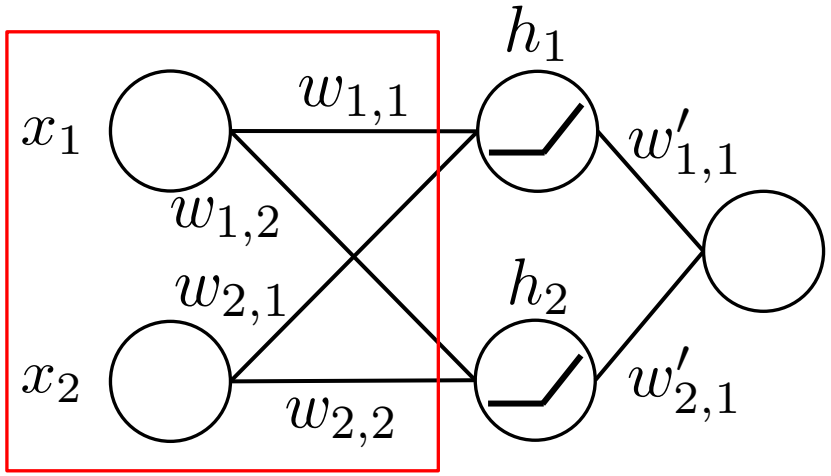


Figure source: https://paperswithcode.com/method/relu

[1] Jarrett, Kevin, et al. "What is the best multi-stage architecture for object recognition?." ICCV 2009.
[2] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." ICML 2010.

Math of MLP for the XOR Problem (1)



x1	x2
0	0
0	1
1	0
1	1

任一筆資料

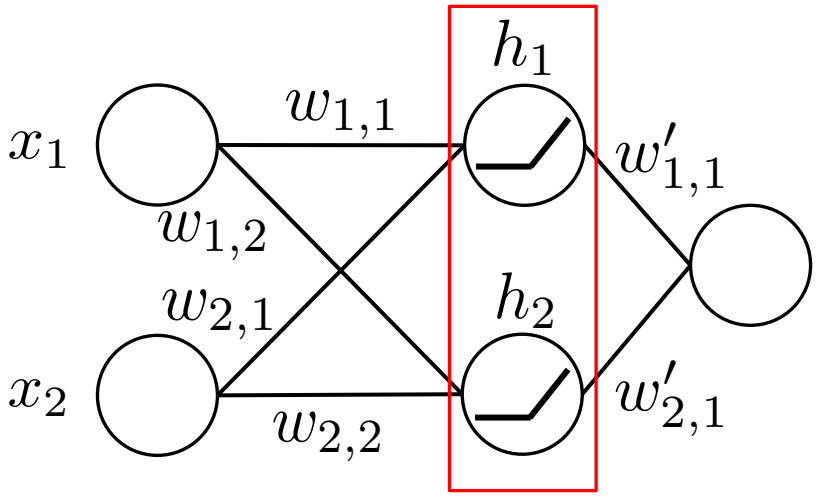
$$[x_1 \quad x_2] \times \begin{bmatrix} \overset{1}{w_{1,1}} & \overset{1}{w_{1,2}} \\ \underset{1}{w_{2,1}} & \underset{1}{w_{2,2}} \end{bmatrix} + [\overset{0}{b_1} \quad \overset{-1}{b_2}]$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

Element-wise summation with bias

Math of MLP for the XOR Problem (2)



$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \xrightarrow{\text{ReLU}} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

ReLU (Rectified Linear Unit)

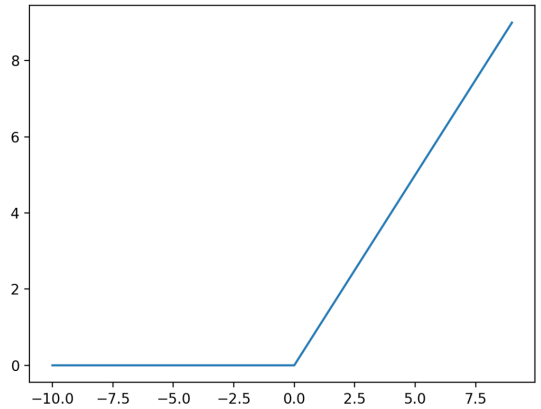
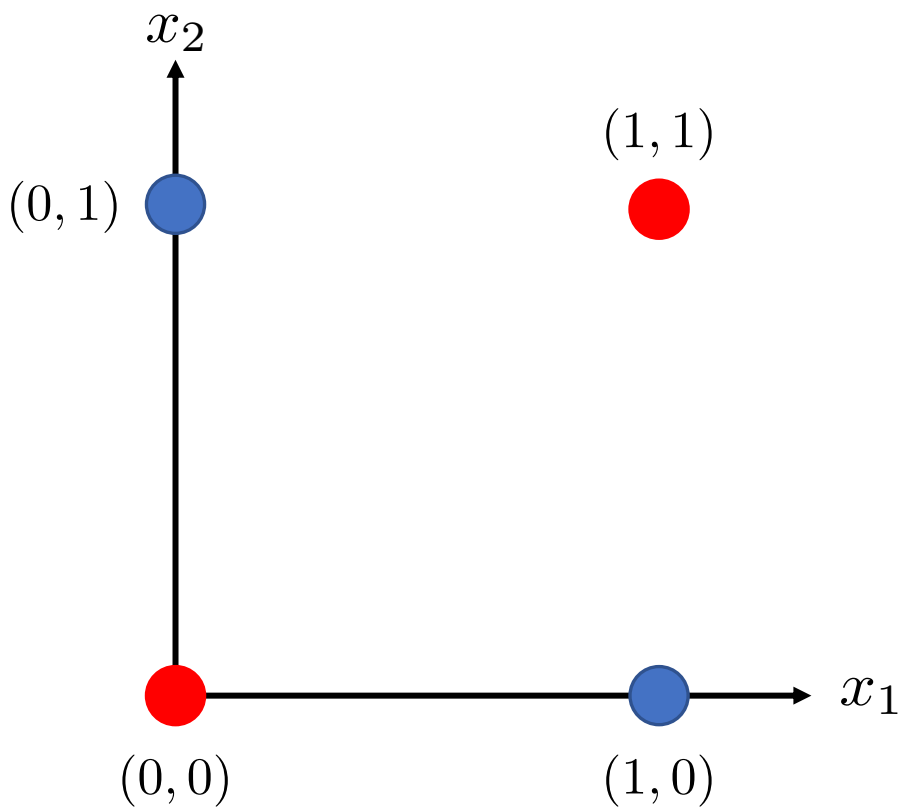


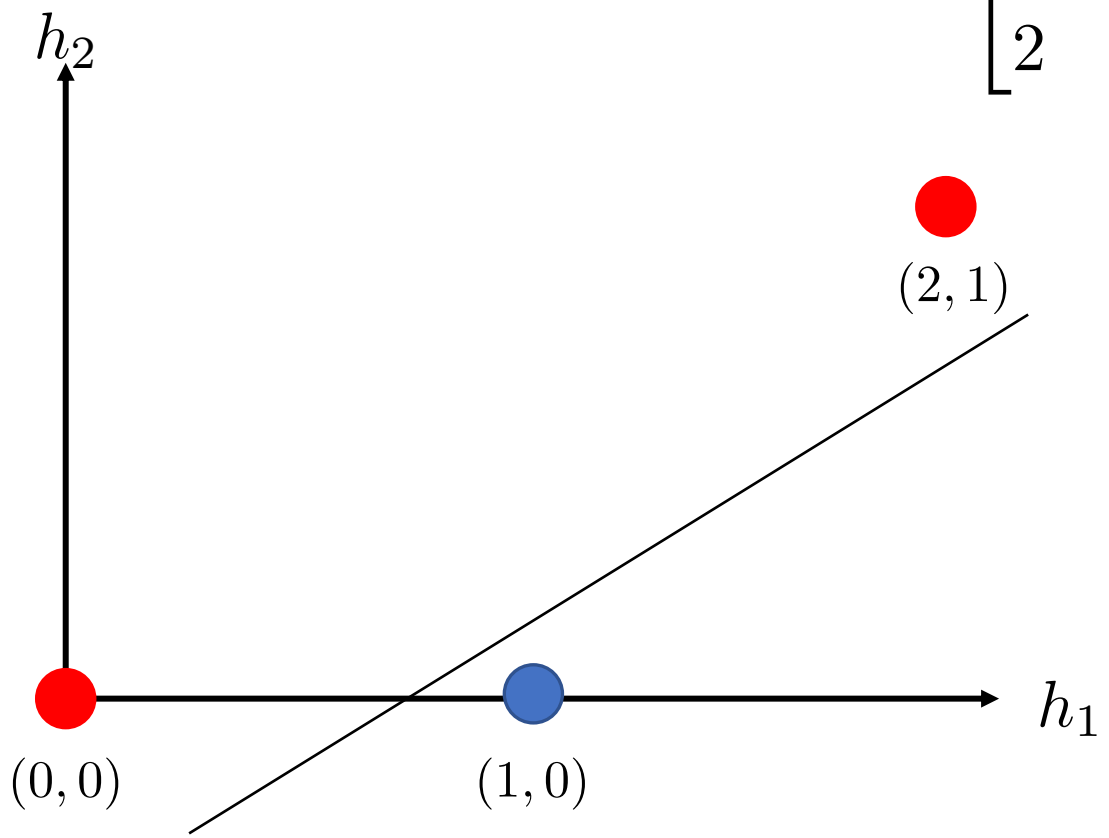
Figure source:
https://paperswithcode.com/method/relu

Math of MLP for the XOR Problem (2)

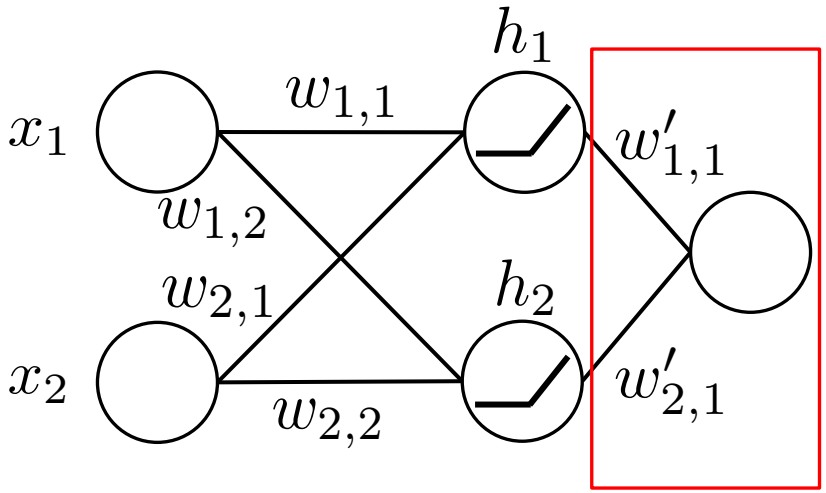
Current values

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$





Math of MLP for the XOR Problem (3)



任一筆資料

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} \overset{1}{w'_{1,1}} \\ \underset{-2}{w'_{2,1}} \end{bmatrix} + \begin{bmatrix} \overset{0}{b'_1} & \overset{0}{b'_2} \end{bmatrix}$$

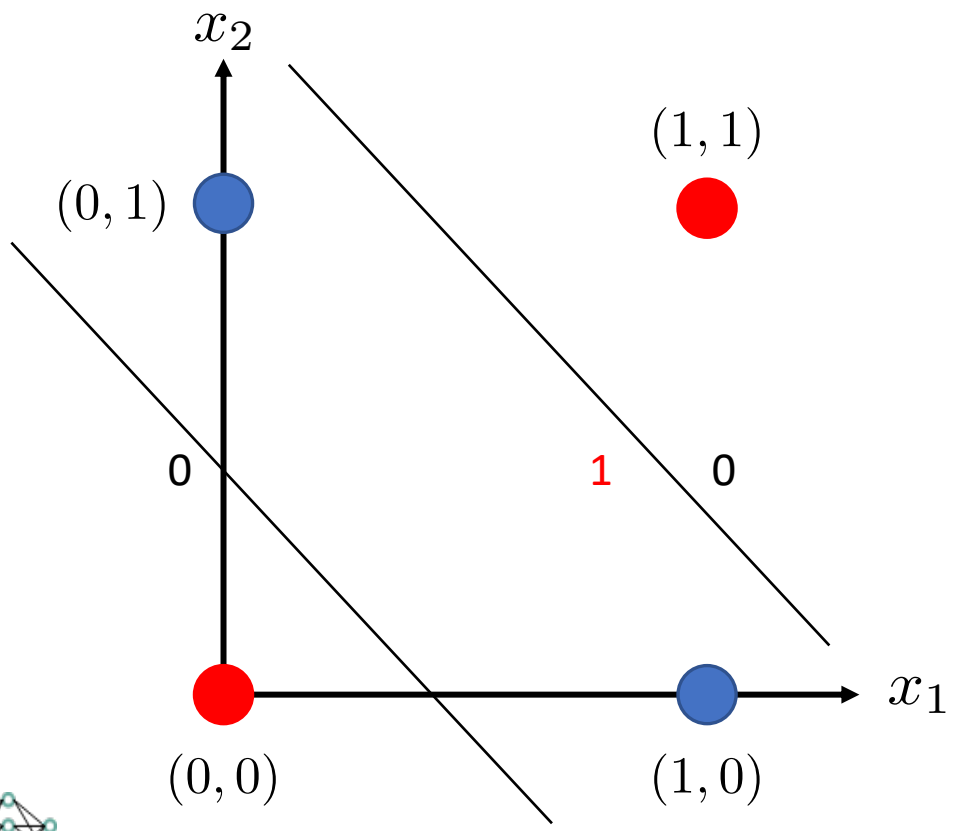
$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

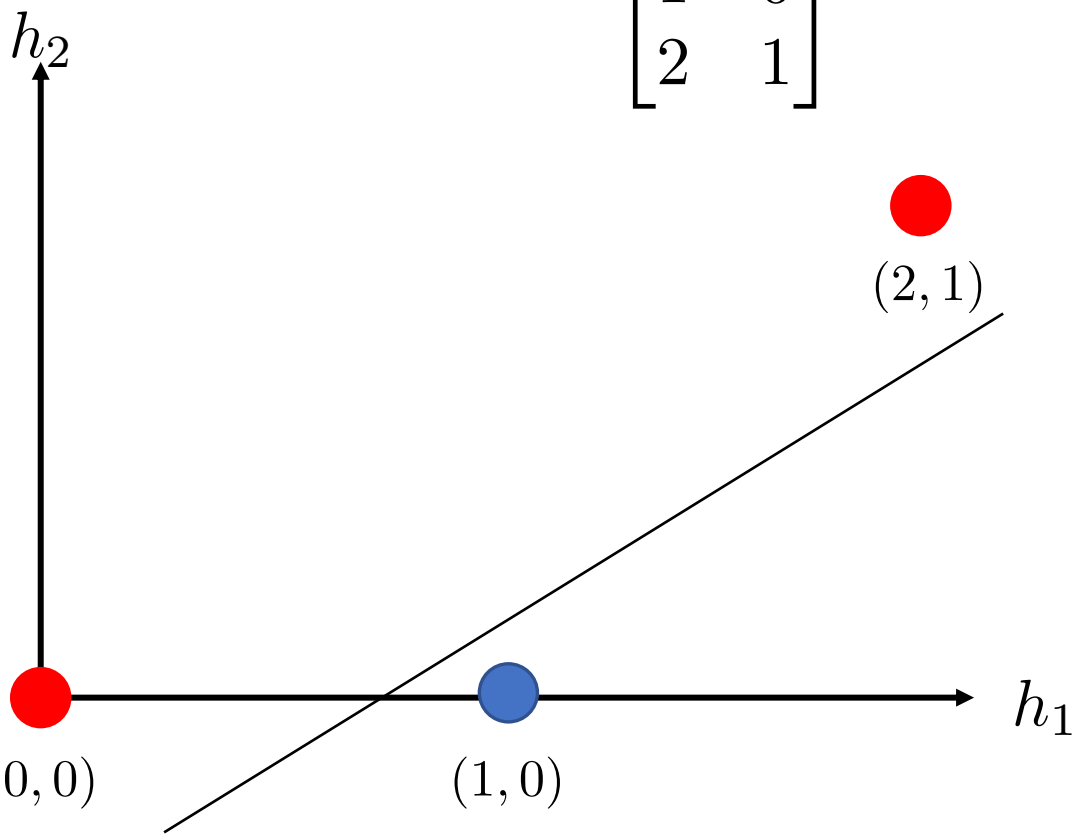
可視為分類層

Visualization of MLP for the XOR Problem

Before the last layer $\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$

Final values $\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$





MLP is also one of the ML methods!

機器學習（英語：machine learning）是人工智慧的一個分支。機器學習理論主要是設計和分析一些讓電腦可以自動「學習」的演算法。機器學習演算法是一類從資料中自動分析獲得規律，並利用規律對未知資料進行預測的演算法。因為學習演算法中涉及了大量的統計學理論，機器學習與推論統計學聯絡尤為密切，也被稱為**統計學習理論**。演算法設計方面，機器學習理論關注可以實現的，行之有效的學習演算法（要防止錯誤累積）。很多推論問題屬於非程式化決策，所以部分的機器學習研究是開發容易處理的近似演算法。

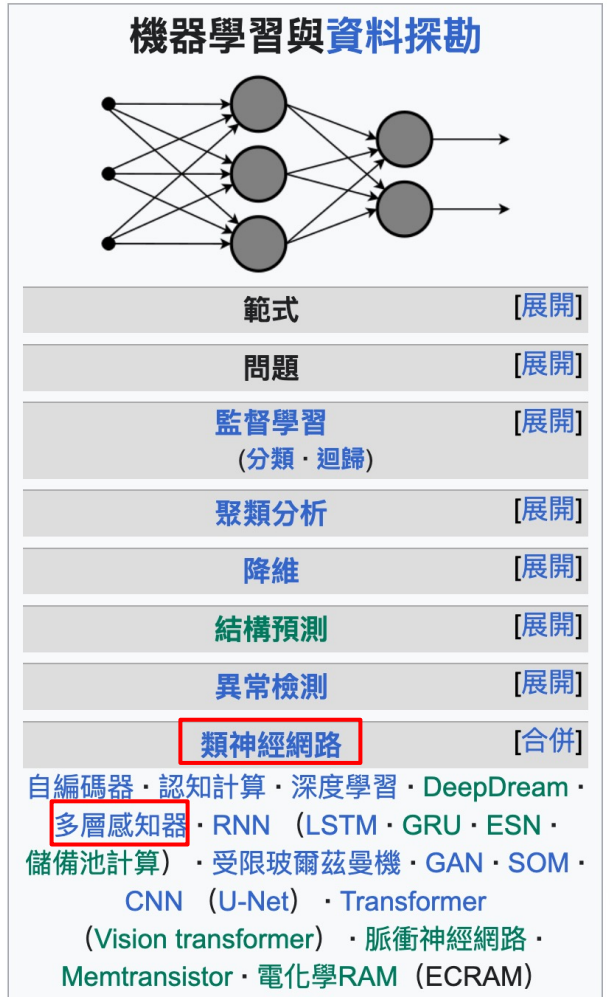
機器學習已廣泛應用於資料探勘、電腦視覺、自然語言處理、生物特徵辨識、搜尋引擎、醫學診斷、檢測信用卡詐騙、證券市場分析、DNA序列定序、語音和手寫辨識、遊戲和機器人等領域。機器學習在近30多年已發展為一門多領域科際整合，涉及機率論、統計學、逼近論、凸分析、計算複雜性理論等多門學科。

定義 [編輯]

機器學習有下面幾種定義：

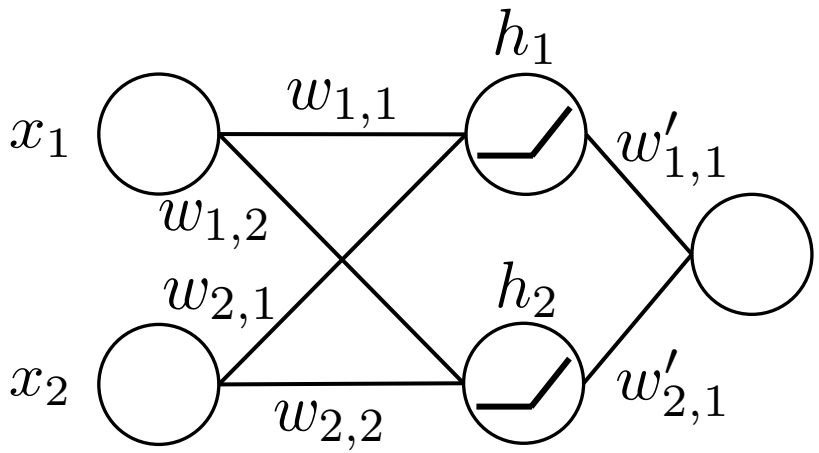
- 機器學習是一門人工智慧的科學，該領域的主要研究對象是人工智慧，特別是如何在經驗學習中改善具體演算法的效能。
- 機器學習是對能通過經驗自動改進的計算機演算法的研究。
- 機器學習是用數據或以往的經驗，以此優化計算機程式的效能標準。

另外，電腦科學家湯姆·米切爾在其著作的Machine Learning一書中定義的機器學習為[1]：



https://zh.wikipedia.org/zh-tw/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92

Deeper and Wider (加深、加寬)



Larger hidden size

More hidden layers

An example of Semantic Net (語義網路)

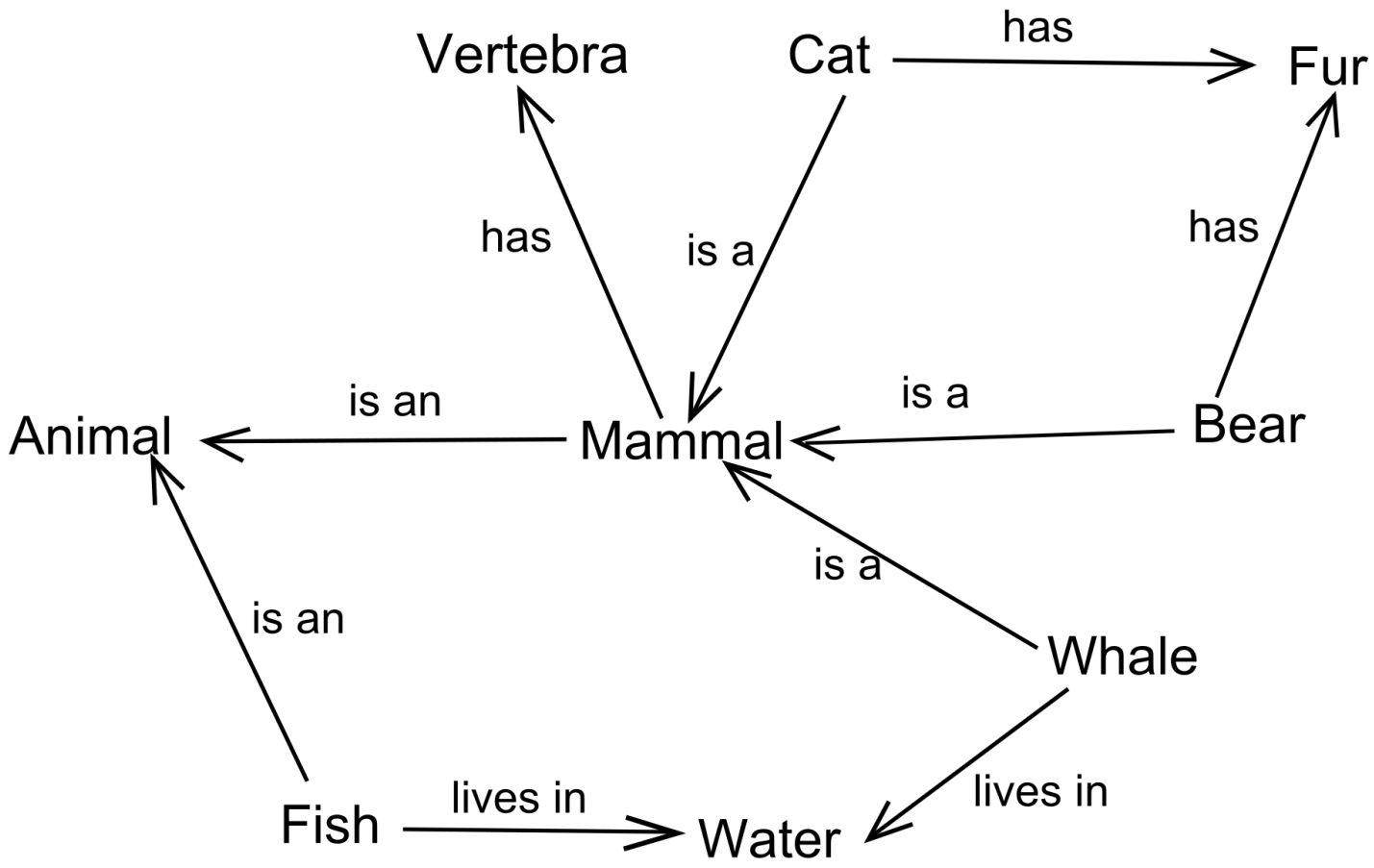


Figure source: https://zh.wikipedia.org/zh-tw/人工智慧史

深度學習為什麼現在變強了？

Difference between AI, ML, and DL

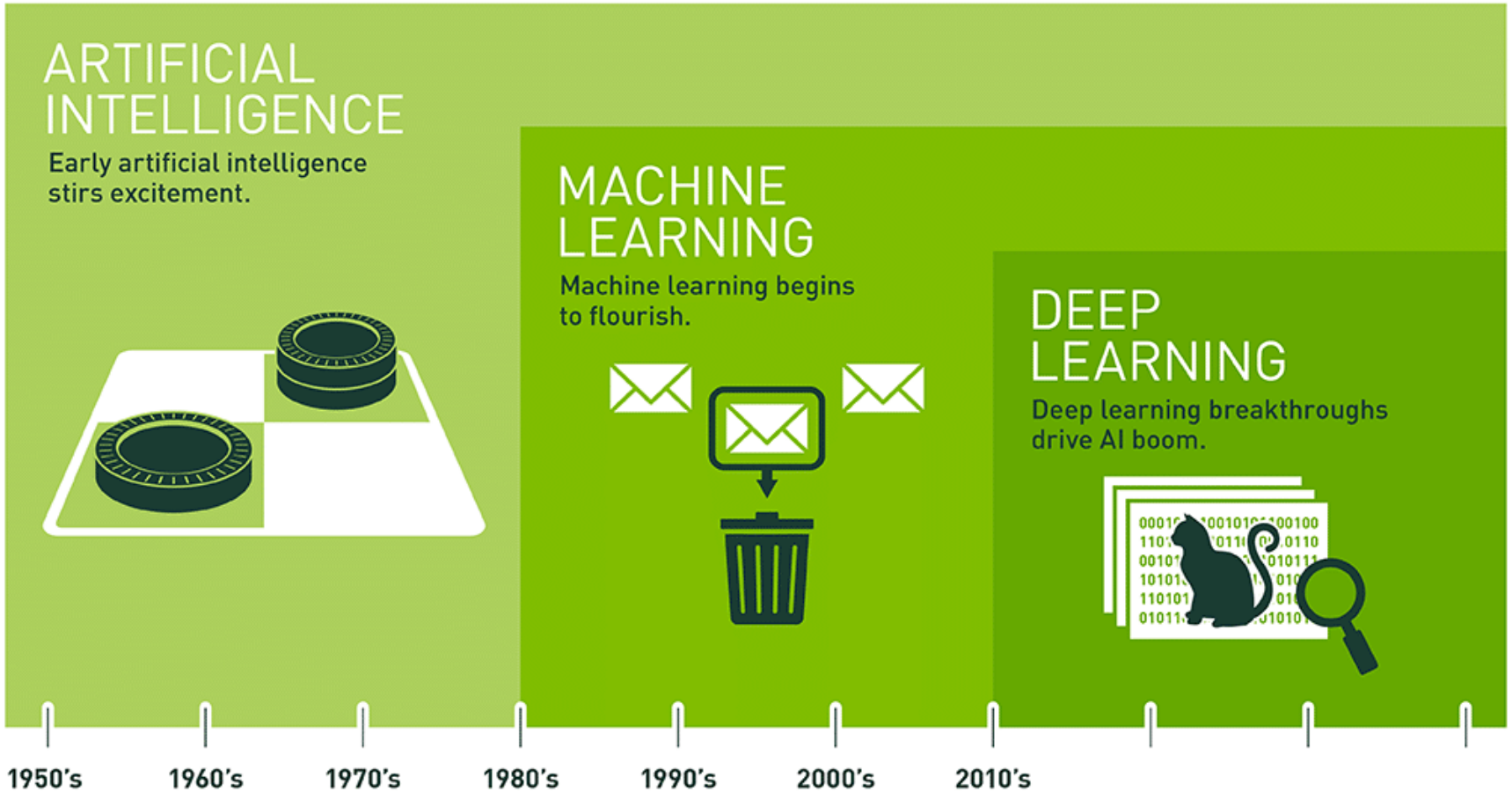
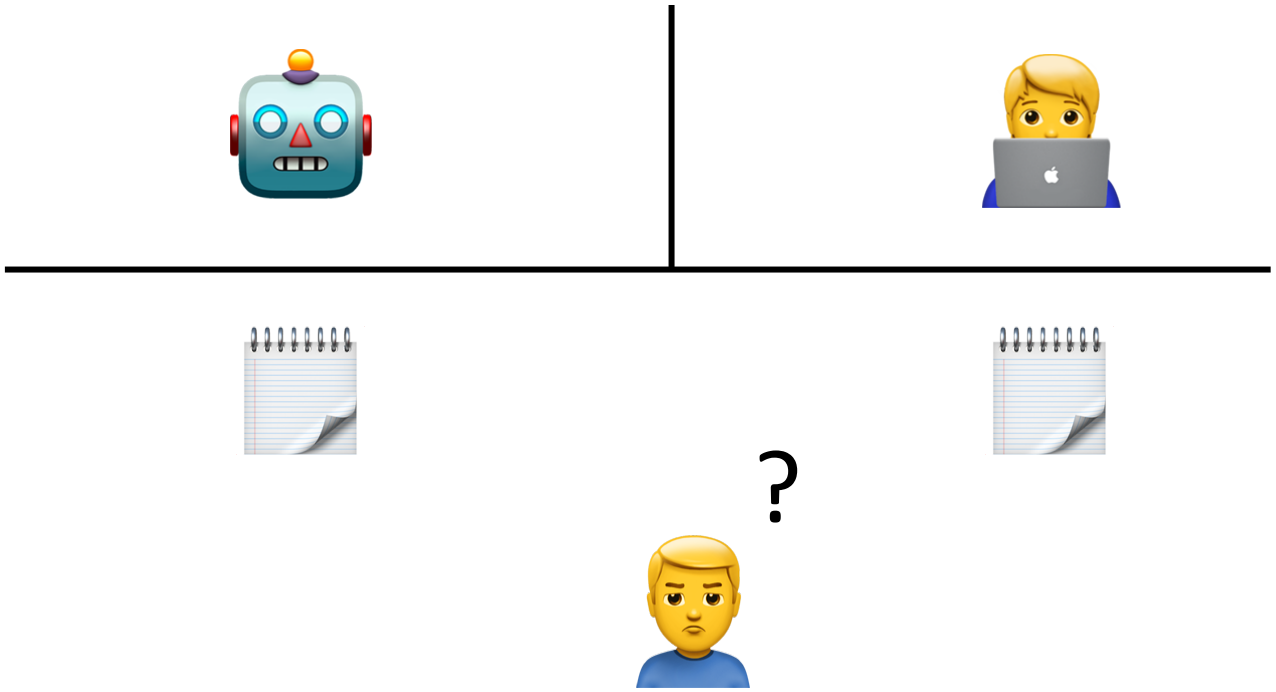


Figure source: https://blogs.nvidia.com/blog/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

圖靈測試 (Turing Test)

- Proposed by Alan Turing in 1950
- One of the earliest concepts of artificial intelligence.



The first Perceptron (1958)

Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.

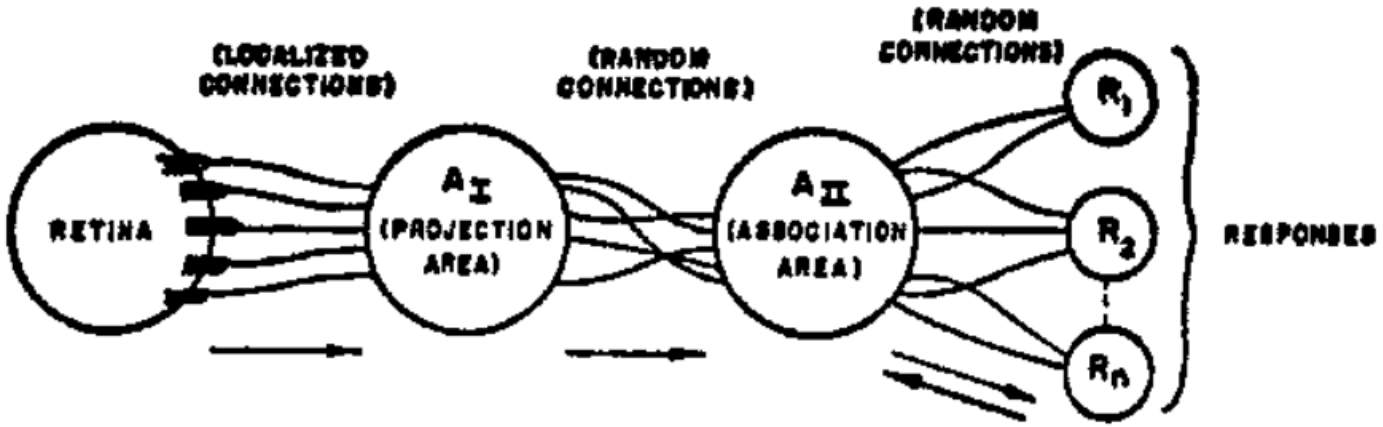


FIG. 1. Organization of a perceptron.

The First Convolutional Neural Networks: Neocognitron (1980)

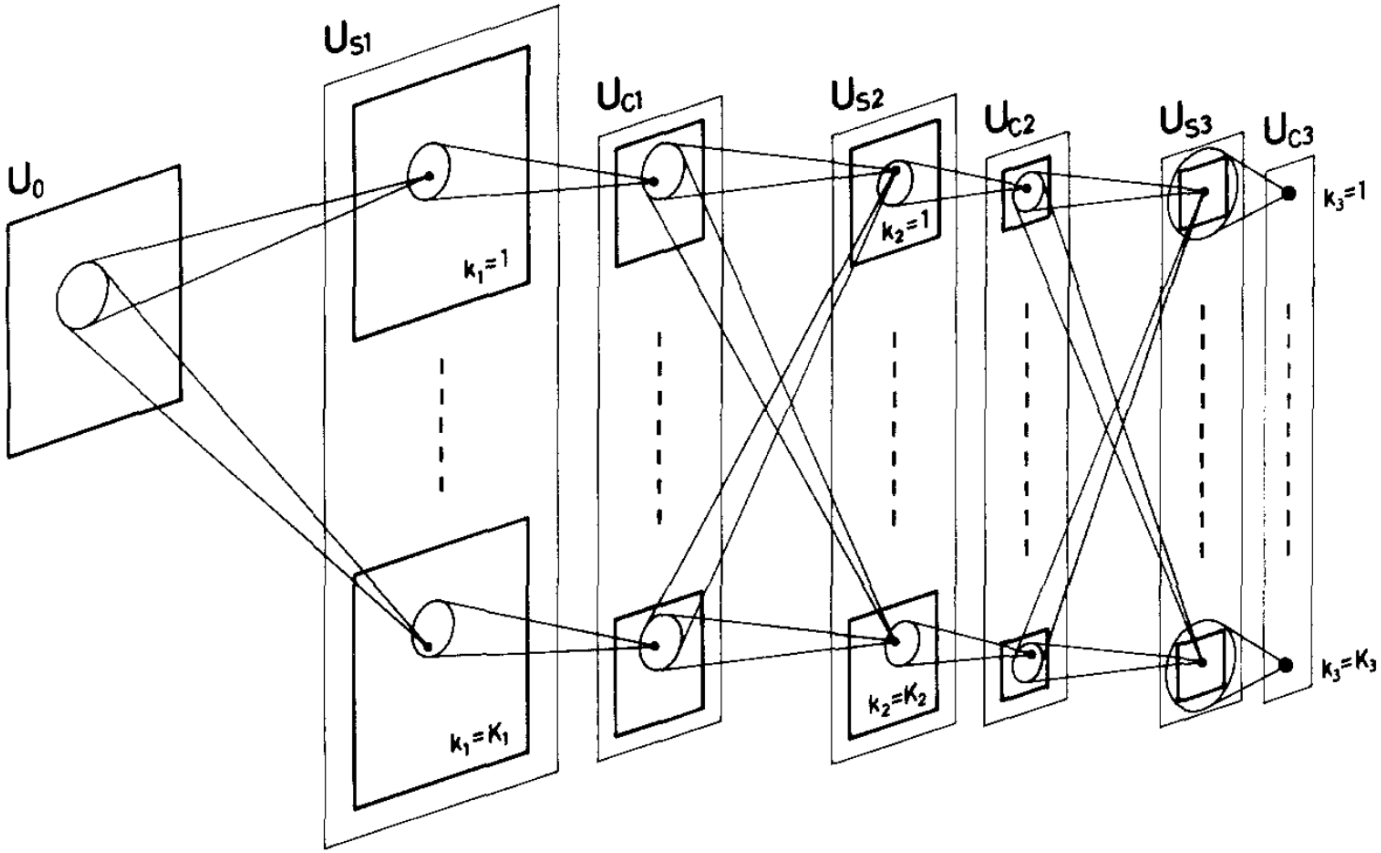


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

Fukushima, Kunihiko. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." Biological cybernetics 36.4 (1980): 193-202.

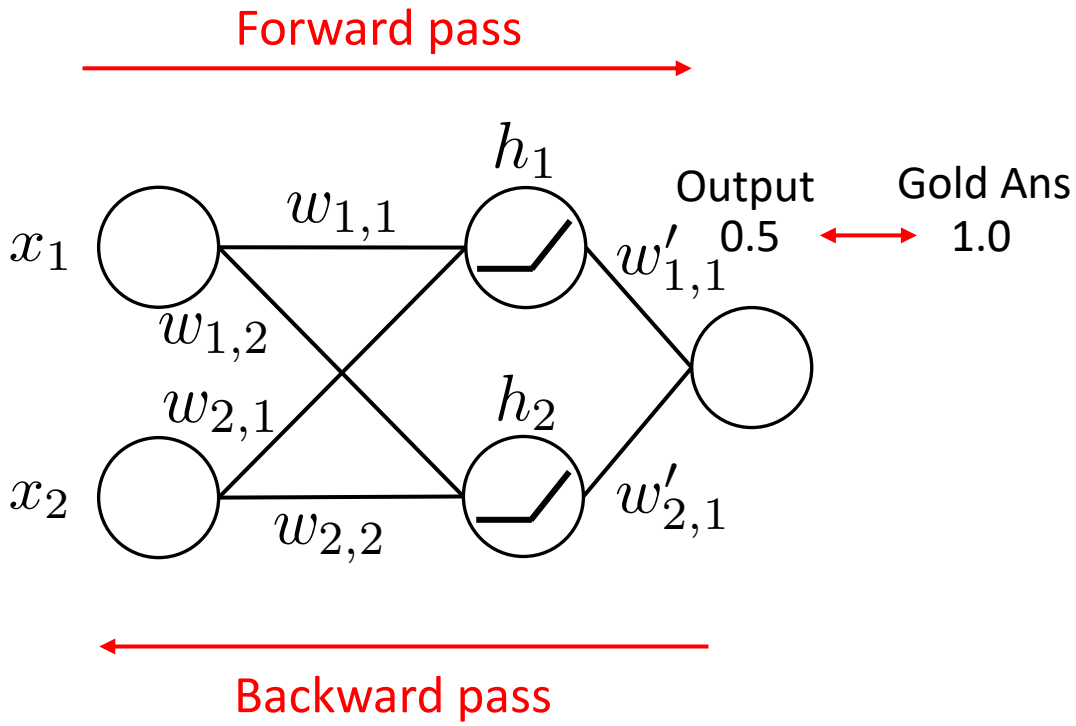
The Back-Propagation Algorithm (1986)

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].



Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533-536.

LeNet-5: Early Convolutional Neural Networks

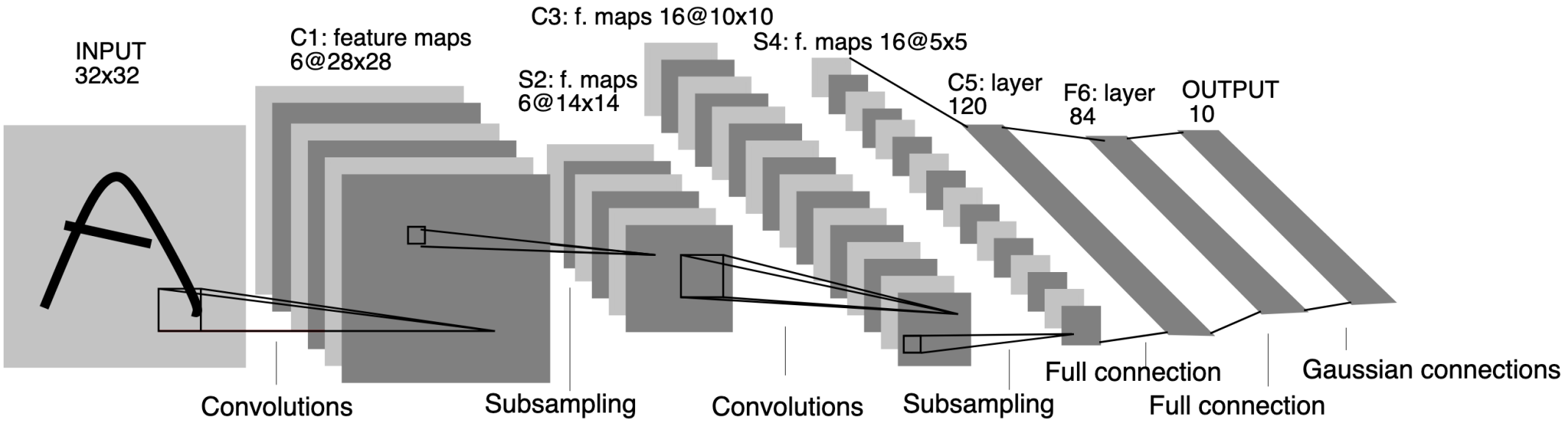
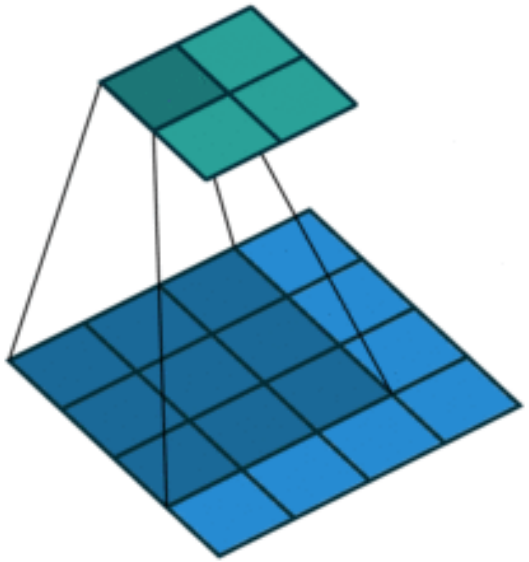


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Examples of Convolutions

Padding = 0, stride = 1



Padding = 0, stride = 2

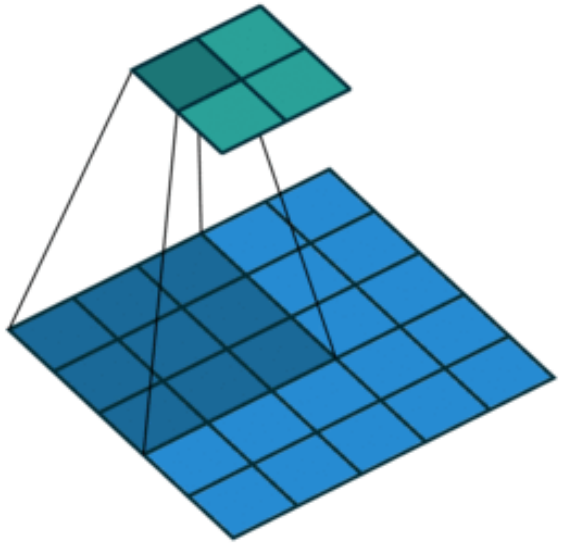


Figure source: https://hannibunny.github.io/mlbook/neuralnetworks/convolutionDemos.html

ImageNet Competition

Complete name: ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

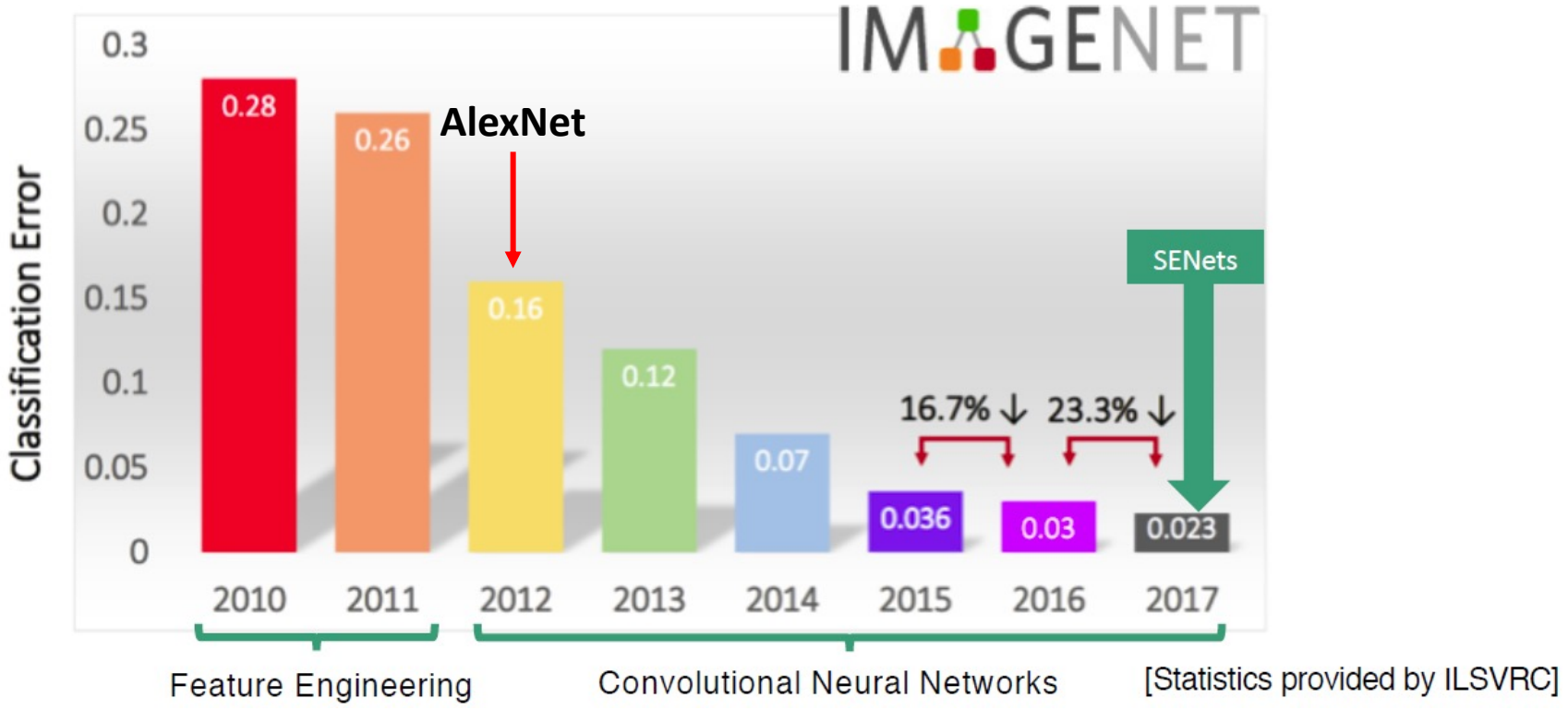


Figure source: https://www.kaggle.com/discussions/getting-started/149448

Growing Trend of GPU Usage during the ImageNet Competition

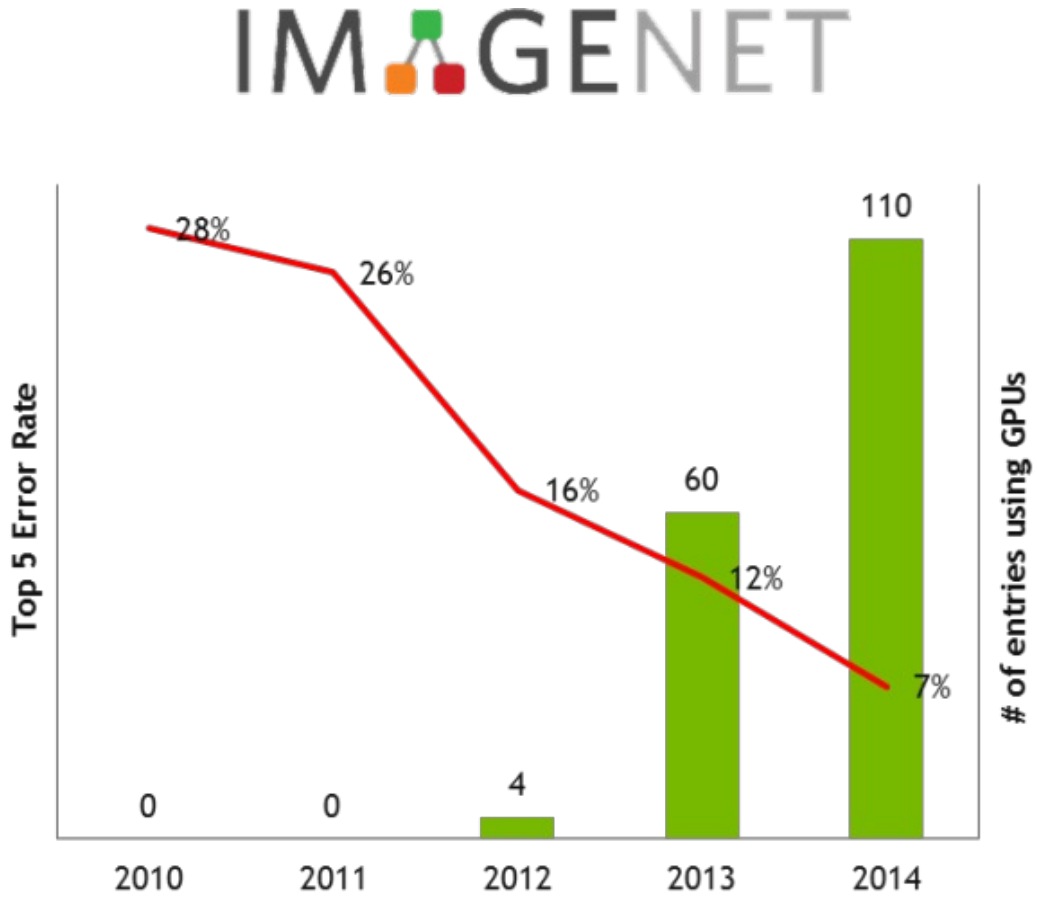
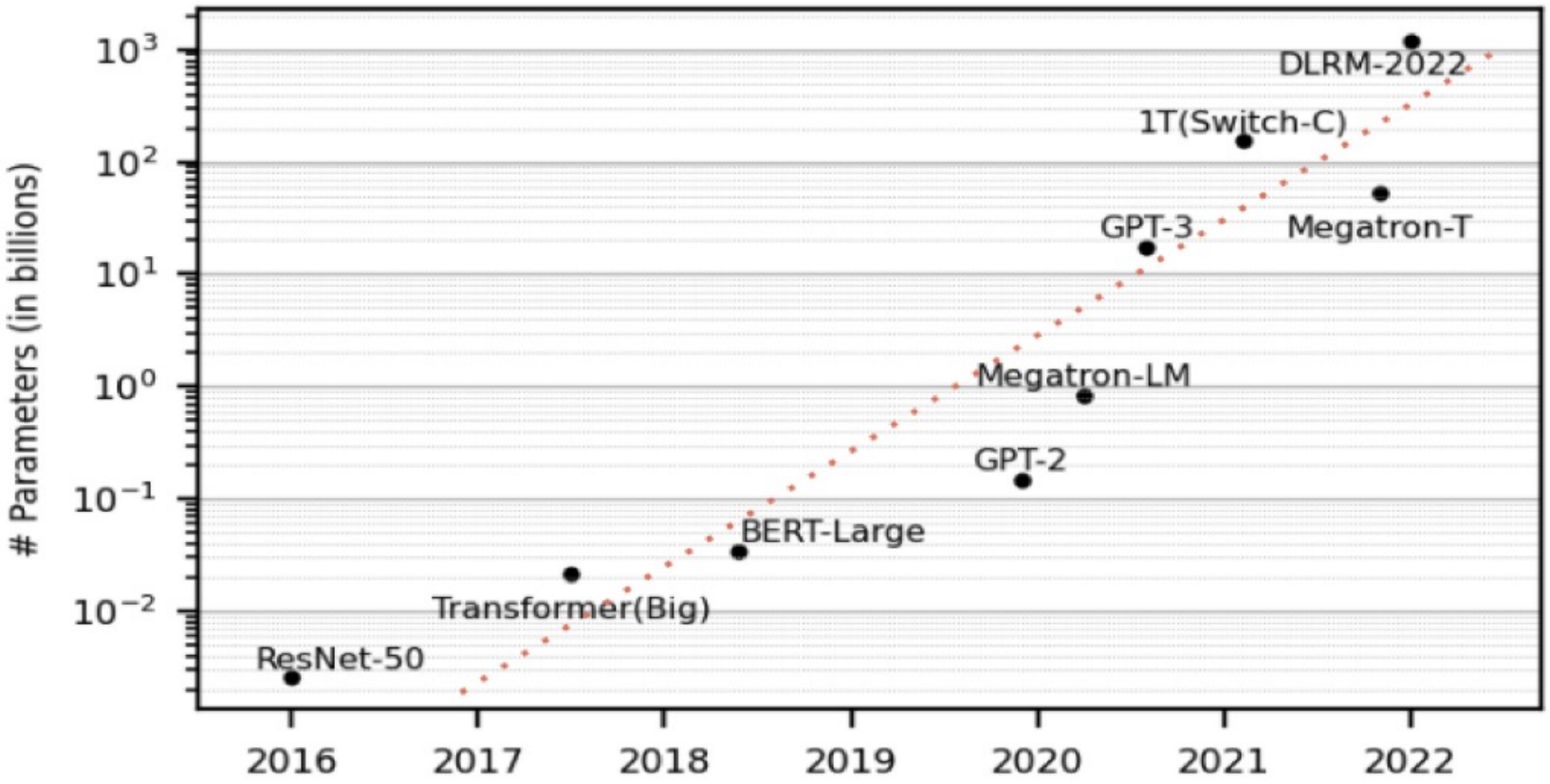


Figure source: https://developer.nvidia.com/blog/nvidia-ibm-cloud-support-imagenet-large-scale-visual-recognition-challenge/

Growing Sizes of Deep Learning Models



Wu, Z., Dai, L. Y., Novick, A., Glick, M., Zhu, Z., Rumley, S., ... & Bergman, K. (2023). Peta-scale embedded photonics architecture for distributed deep learning applications. Journal of Lightwave Technology, 41(12), 3737-3749.

The Revolution of ChatGPT

ChatGPT came out in November, 2022.

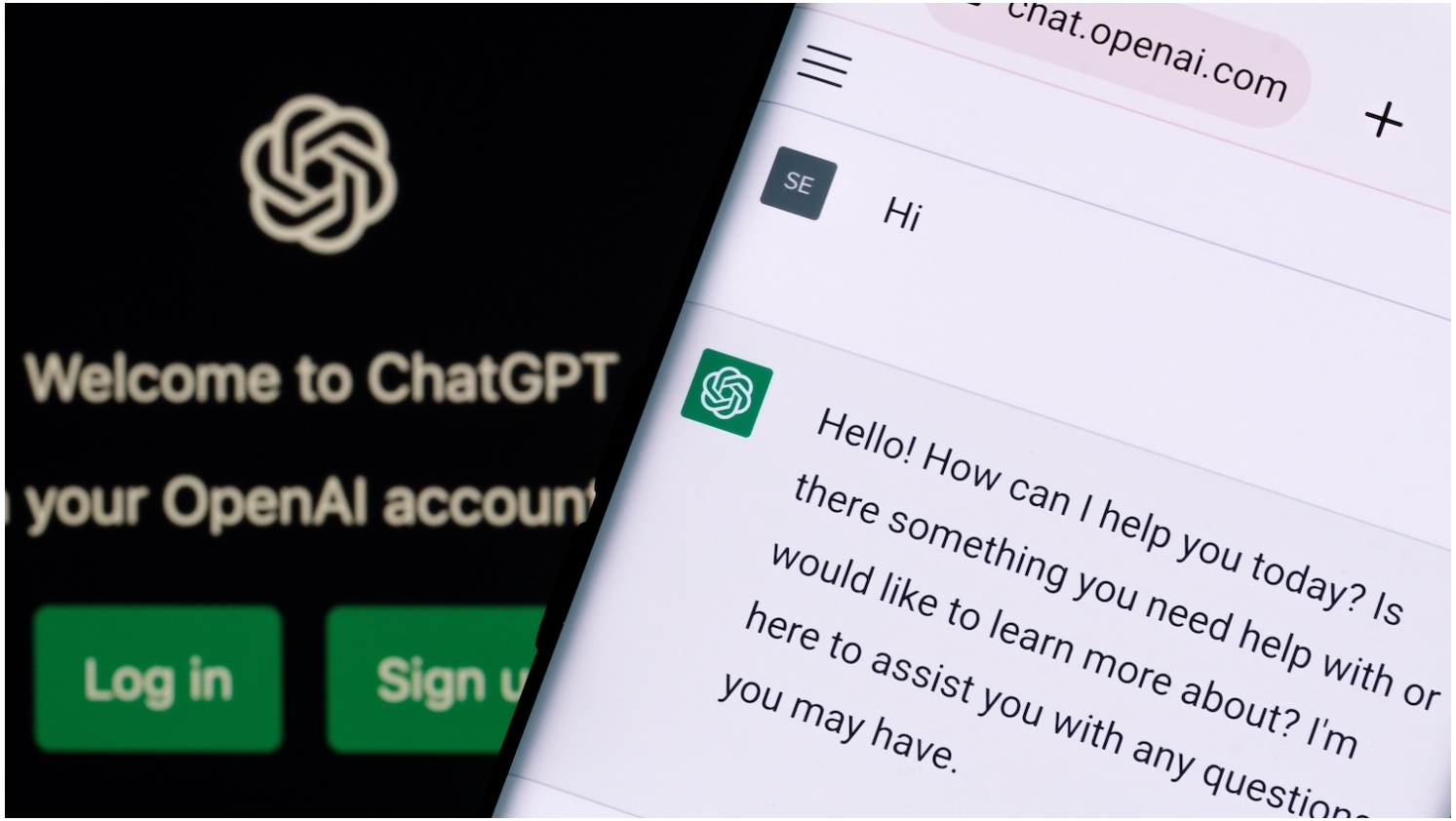


Figure source: https://poole.ncsu.edu/thought-leadership/article/lets-chat-about-chatgpt/
https://openai.com/index/chatgpt/

InstructGPT: the precursor of ChatGPT

Model size: 175B (same as GPT-3)

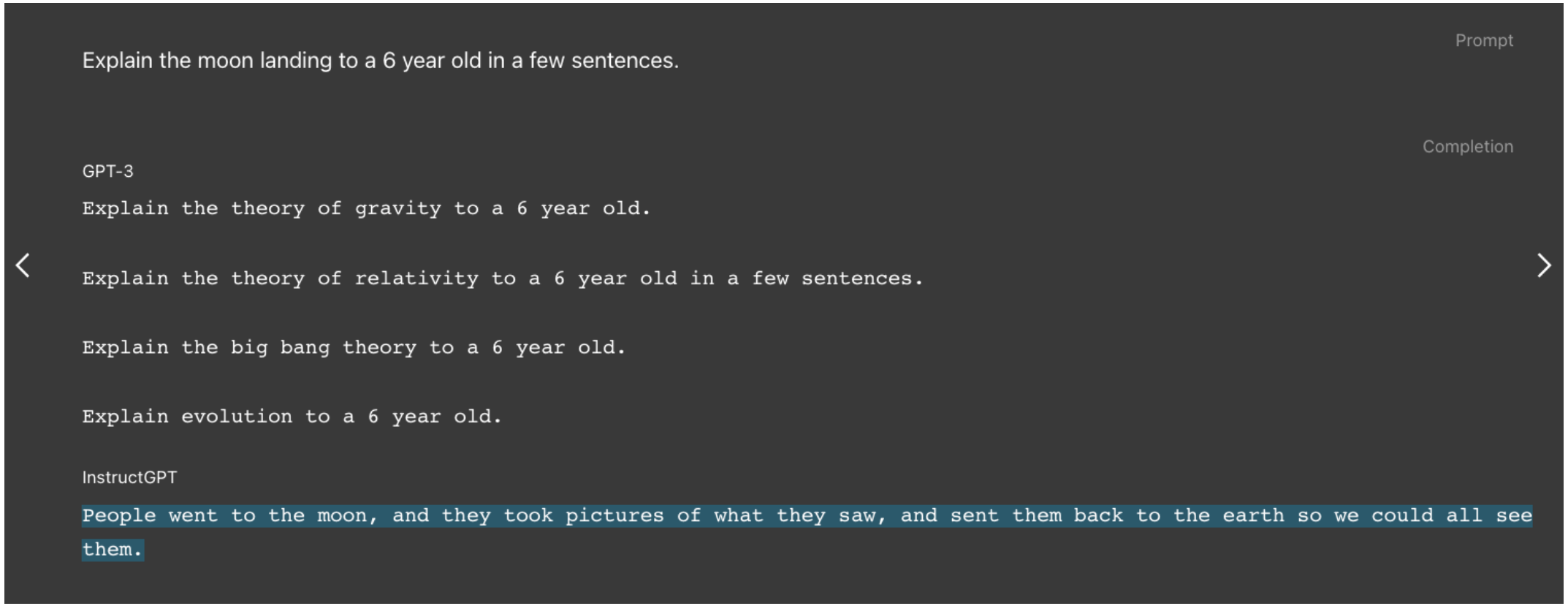
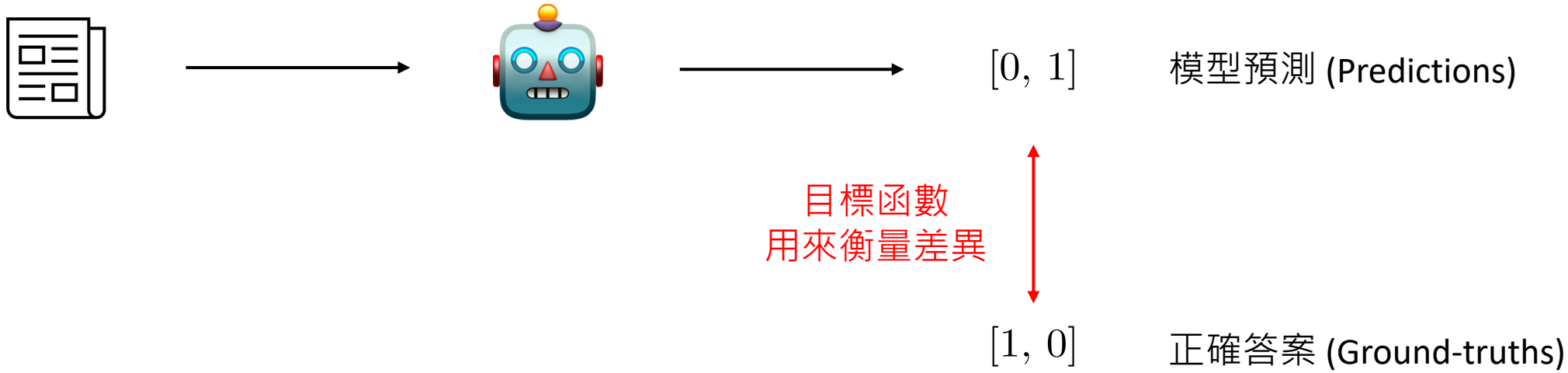


Figure source: https://openai.com/index/instruction-following/
Ouyang, Long, et al. “Training language models to follow instructions with human feedback.” NeurIPS 2022.

模型訓練師新手村

所有的機器學習都是在做最佳化



最小化Predictions和Ground-truths之間的差異即為機器學習的目標

目標函數的別名

- 目標函數 (Objective Function)
- 損失函數 (Loss Function)
- 誤差值 (Loss)
- 可能採用「交叉熵」 (Cross entropy)

如何訓練模型：資料集

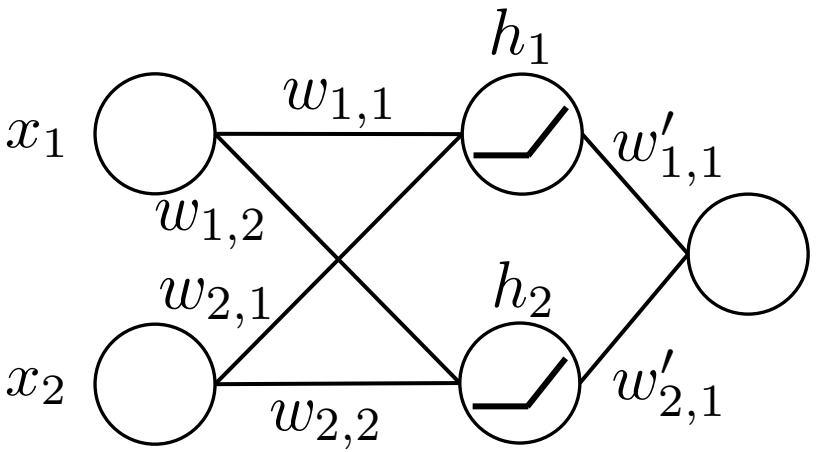
Training Set
(訓練資料集)

Test Set
(測試資料集)

Training Set
(訓練資料集)

Validation Set
(驗證資料集)

如何訓練模型：演算法



模型預測 (Predictions) ⟷ 正確答案 (Ground-truths)

- 透過 Gradient Descent (梯度下降) 尋找最佳解
- 透過 Back-Propagation 更新權重值 (w跟b)

TensorFlow Playground

- https://playground.tensorflow.org

我們這學期要學什麼？

In this course, we will learn:

- [Deep Learning Fundamentals]
  - PyTorch
  - Gradient Descent
  - Neural Networks and Back-propagation
  - Building a Deep Learning Framework from Scratch

In this course, we will learn:

- [Deep Learning Applications]
  - Computer Vision (CNNs)
  - Natural Language Processing (RNNs)
  - Modern Architectures (Transformers, Vision Transformers)
  - Self-supervised Learning
  - Reinforcement Learning

In this course, we will learn:

- [Advanced Topics in Deep Learning]
  - Model Compression
  - Big Models and how to train them efficiently
  - Explainable AI

Scoring of this course

- Assignments 7.5% x 4 times (30% in total)
- Quizzes: 1% x 10 times
- Mid-term exam 30%
- Final project 30%

About the Assignments

- 作業一定都會有範例程式碼 (Sample Code)

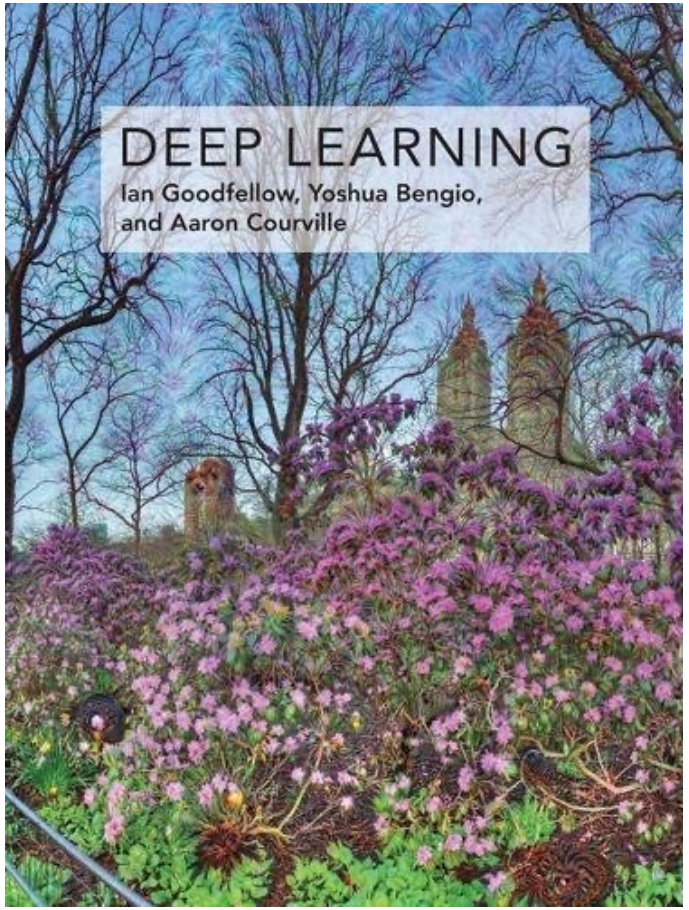
Final Project 是什麼？

- Final Project 是一種完整的訓練
- 你將自行從頭到尾做完一個題目，包含：
  - 資料前處理 (Data Pre-processing)
  - 模型訓練 (Model Training)
  - 模型評估 (Evaluations) -->了解問題並改進模型
- 每組人數待修課同學總數訂定，原則上1-3人/組

Syllabus

Week	Topic	Note
1	深度學習介紹以及本課程綱要 (Introduction to Deep Learning / Syllabus)	
2	PyTorch教學 / 梯度下降 (I)	(最佳化)
3	梯度下降 (II)	
4	神經網路與反向傳播法	HW1
5	~~手把手實作~~深度學習框架細節探索	
6	卷積神經網路 (with LAB)	
7	循環神經網路 (with LAB)	HW2
8	自注意力模型 (with LAB)	
9	期中考	
10	自監督式模型方法	HW3
11	強化學習 (Reinforcement Learning)	
12	模型壓縮 (Model Compression)	
13	大模型時代如何有效率訓練模型？	HW4
14	可解釋性人工智慧	
15	小組實作成果報告 (1)	
16	小組實作成果報告 (2)	

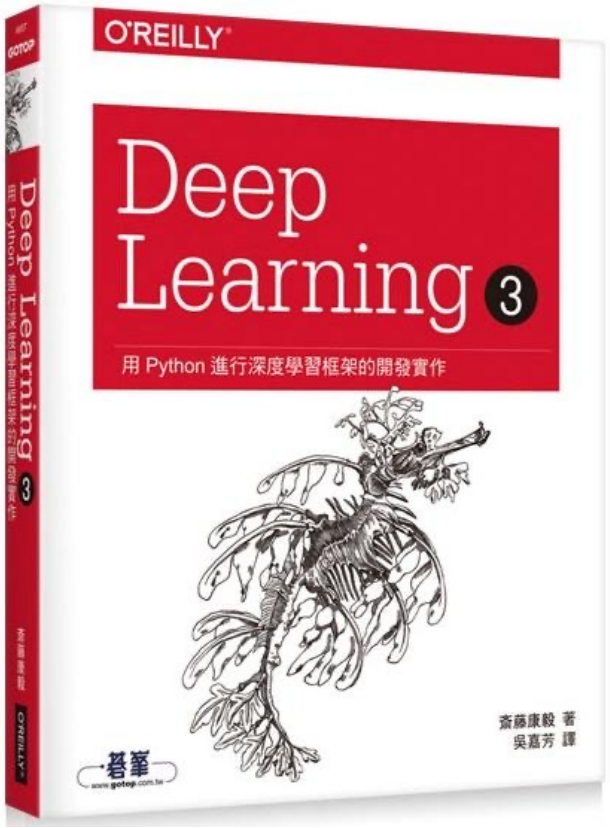
Textbook



Ian Goodfellow, Yoshua Bengio, & Aaron Courville (2016). Deep Learning. MIT Press.

Free Online book:
https://www.deeplearningbook.org/

Reference book



Deep Learning 3 | 用Python進行深度學習框架的開發實作，歐萊禮 (ISBN: 9789865027346)

GitHub of this book: https://github.com/oreilly-japan/deep-learning-from-scratch-3

More resources for XOR

- Community blog post
  - https://dev.to/jbahire/demystifying-the-xor-problem-1blk
  - https://chih-sheng-huang821.medium.com/機器學習-神經網路-多層感知機-multilayer-perceptron-mlp-運作方式-f0e108e8b9af
- Stanford course:
  - https://youtu.be/s7nRWh_3BtA?si=ZEGRN5pmPchM0NxM
- Also, in Deep Learning book (Chapter 6.1)

Contact Information

Instructor: 林英嘉

✉ yjlin@cgu.edu.tw

🕘 Office Hour: 歡迎來信預約，一定沒問題

TA: 林君襄

✉ becky890926@gmail.com

寄信前請在主旨加註記 [深度學習]

Thank you!

Instructor: 林英嘉

✉ yjlin@cgu.edu.tw