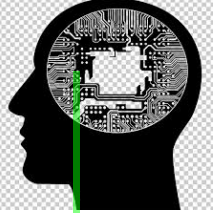


從人工智慧到深度學習

From AI to Deep Learning

陳木松 Mu-Song Chen

大葉大學電機系

Department of Electrical Engineering

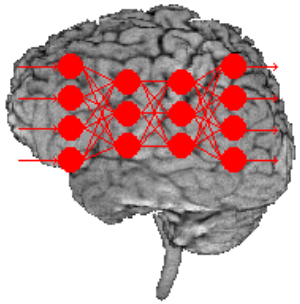
Da-Yeh University

chenms@mail.dyu.edu.tw

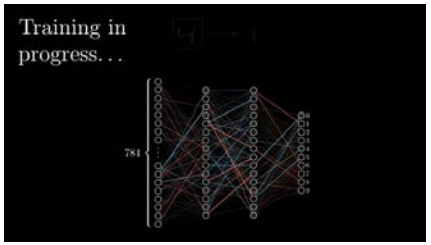
H335, 04-8511888-2167

Outlines

- **AI (人工智慧)**
  - **Why human can learn (為什麼人類可以學習)**



- **Machine Learning (機器學習), model driven v.s. data driven**
  - **Neural Network (神經網路)**



- **Deep Learning (深度學習)**
  - **Visual Cortex (視覺皮質)**
  - **Convolutional Neural Network (卷積神經網路)**

尤利烏斯·凱撒(Julius Caesar)

VENI VIDI VICI

我來、我看、我征服

I came, I saw, I conquered. [Latin: Veni, vidi, vici.]

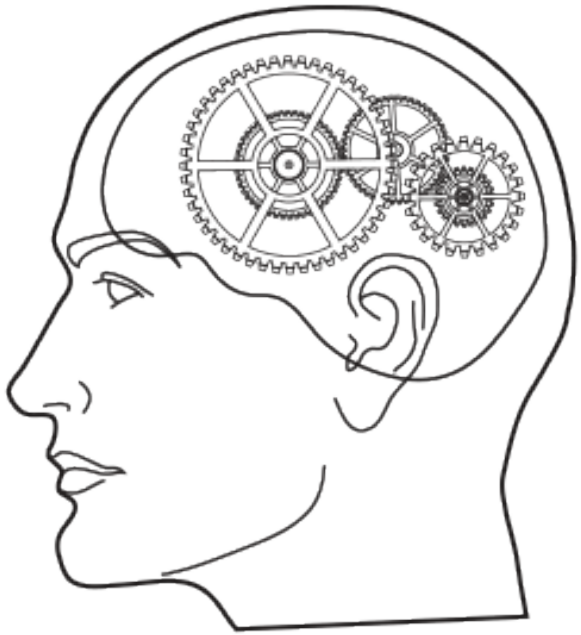
-Julius Caesar

AI comes

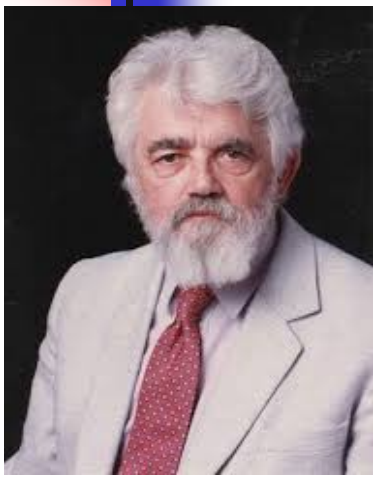
AI sees

AI conquers

Artificial Intelligence



Father of Artificial Intelligence

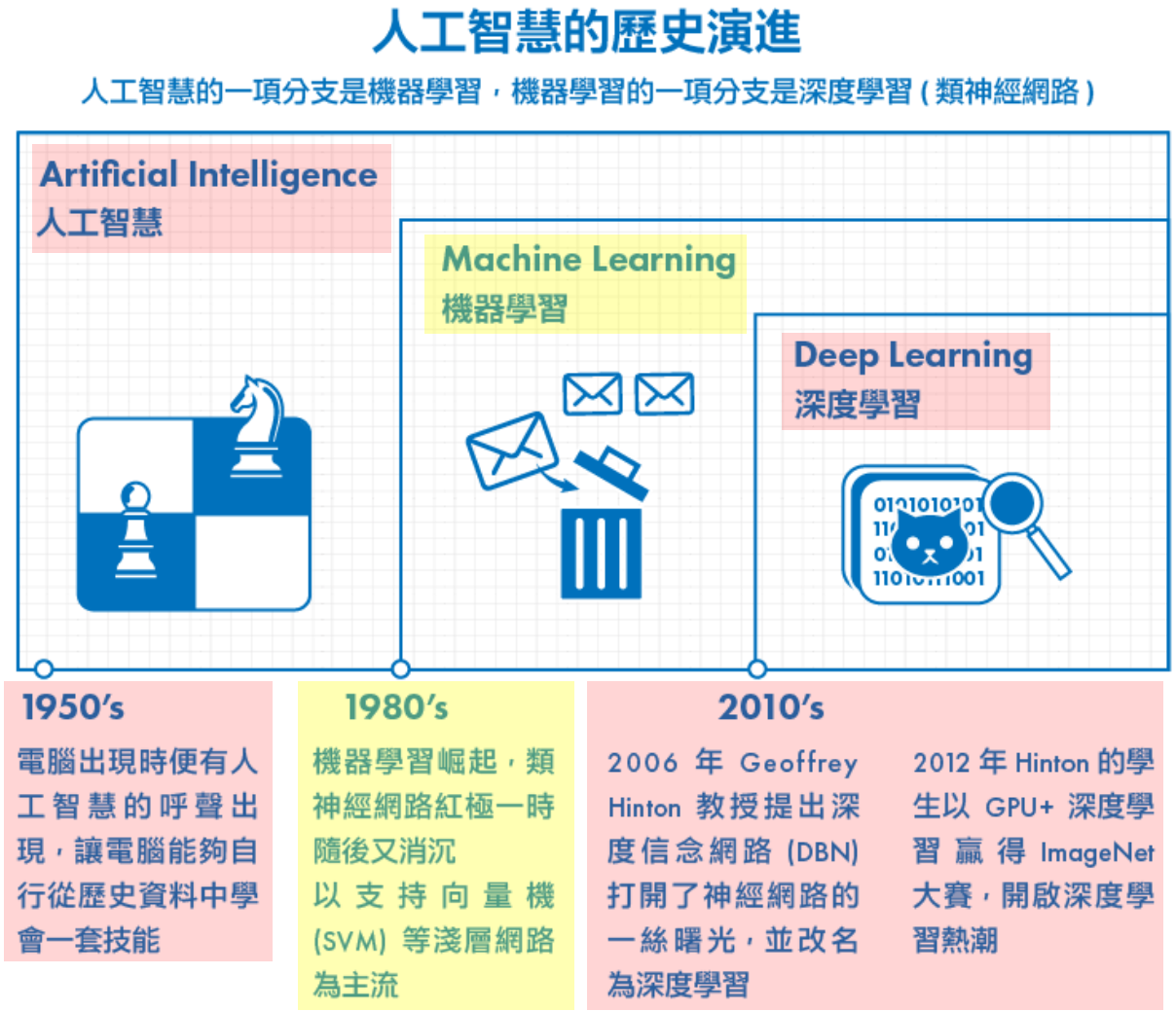


- 約翰·麥卡錫(John McCarthy)在1956年達特矛斯會議首次提出「人工智慧」(Artificial Intelligence)的概念。約翰·麥卡錫在人工智慧領域的貢獻，因而獲得1971年圖靈獎(Turing Award)。

- John McCarthy的AI定義是，「**未來機器可以像人類一樣思考、解決問題、以及自我成長， 甚至超越人類**」

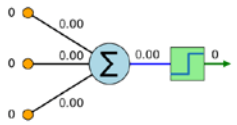
- *A Artificial Intelligence:* ***Intelligent Agents refers to the ability of the computer to simulate/model human thinking processes to imitate human ability or behavior.***

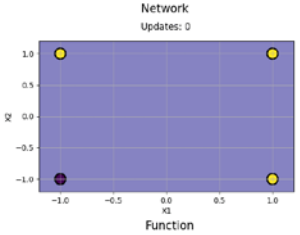
AI - Machine Learning - Deep Learning

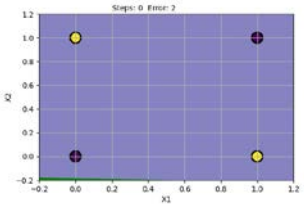


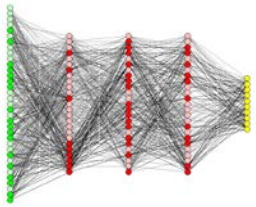
AI Evolution

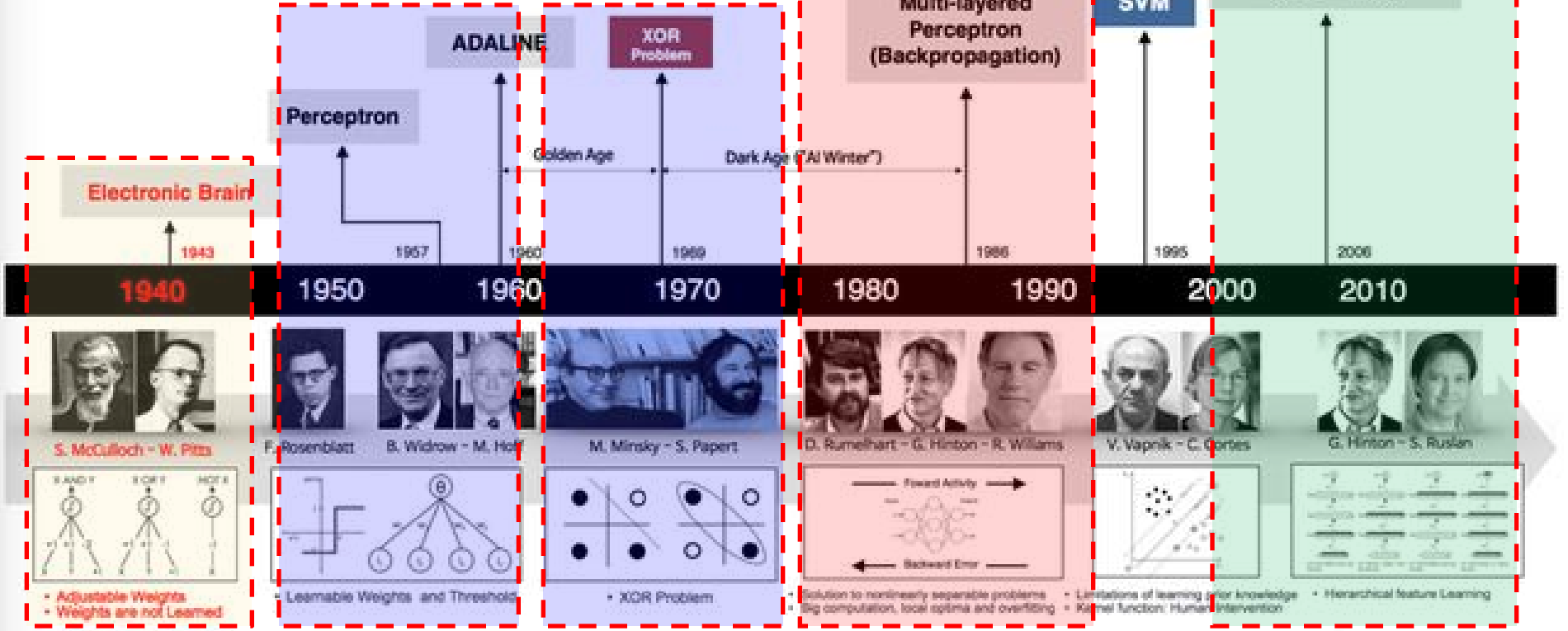
Big data, GPU











ENIAC, Mark I, II

From Theory to Application

弱人工智慧 v.s. 強人工智慧

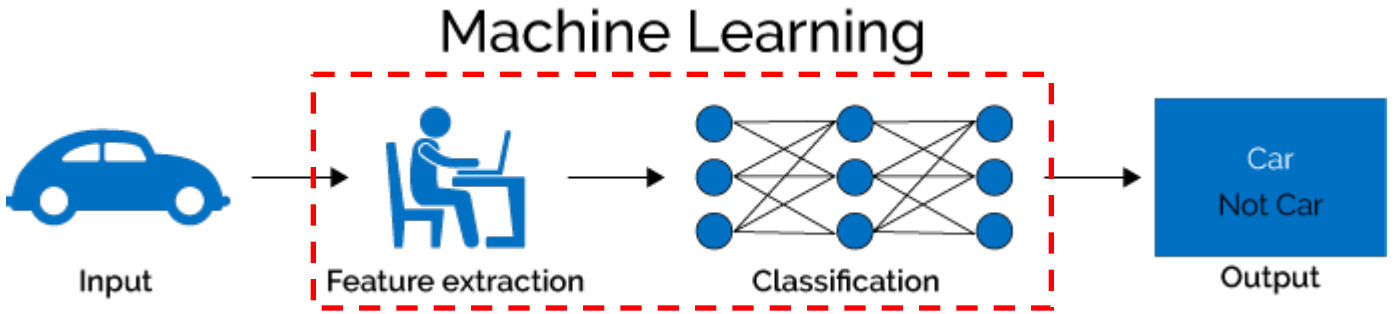
- **弱人工智慧**
  - 透過電腦軟體模擬「部份」的人類智慧與行為，例如下圍棋、下象棋、自動導航、影像判斷、醫療照護等特定功能。它們通常被應用在**解決特定領域問題**，因此也有人稱之為Narrow AI，但也是目前最常見的AI。
- **強人工智慧**
  - 期望電腦能執行所有人類的工作，因此除具有認知能力之外，還能夠推理、學習、溝通、與認知解決問題，甚至擁有自我意識，並綜合這些能力面對未知的情境，因應不同狀況作自適應調整，實踐特定目標。換言之強人工智慧就是能夠**執行「通用任務」**的人工智慧。

規則式(Rule-based)AI演算法

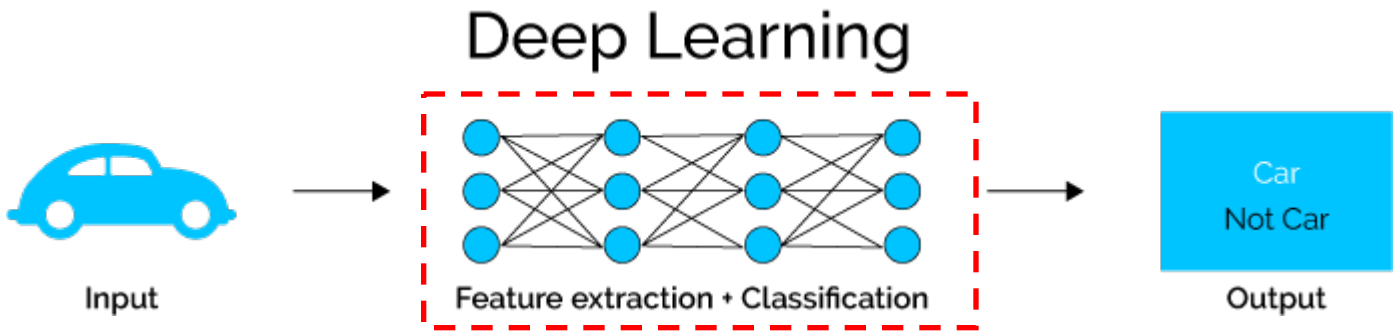
- 規則式(Rule-based)AI演算法指機器模仿人類，以邏輯推論的方式，並加入環境變數，推理判斷結果，主要代表為專家系統。**規則式AI演算法注重的是「推理」而非「學習」**。
- 規則式演算法僅適用於特定狹隘的問題領域。然而生活中所有事物不是都有規則可循，規則式AI難以應付所有事物的判斷與分析。
- AI規則只有對於外顯知識才能清楚描述其法則，但許多事物包括內隱知識，「只可意會，不可言傳」，因此難以清楚表達其內涵。AI規則必須明確、穩定、例外少，例如「貓都有花紋，狗沒有花紋」、「貓是喵喵叫，狗是汪汪叫」，這就是判斷貓狗的規則。但是「**貓沒有花紋且汪汪叫**」，如果用規則式演算法斷，貓就有可能會被分辨成狗。

學習式(Learning-based)AI演算法

- AI 學習式(Learning-based)演算法
  - **Machine learning**：資料→ **特徵擷取→建模(NN modeling)** →結果

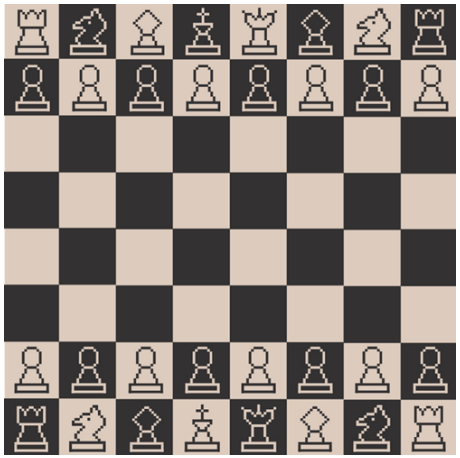


  - **Deep learning**：資料→ **建模(CNN modeling)** →結果

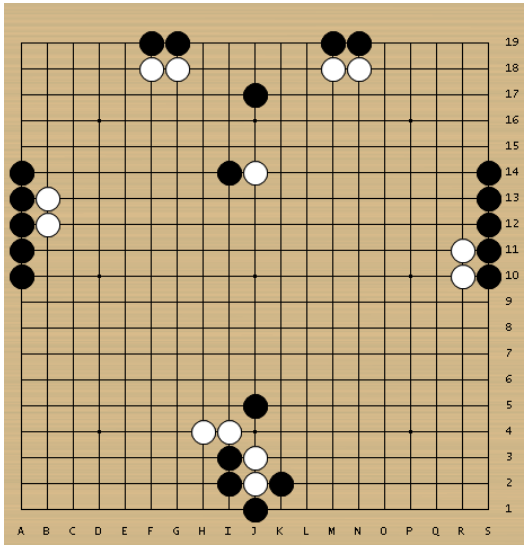


Applications of AI

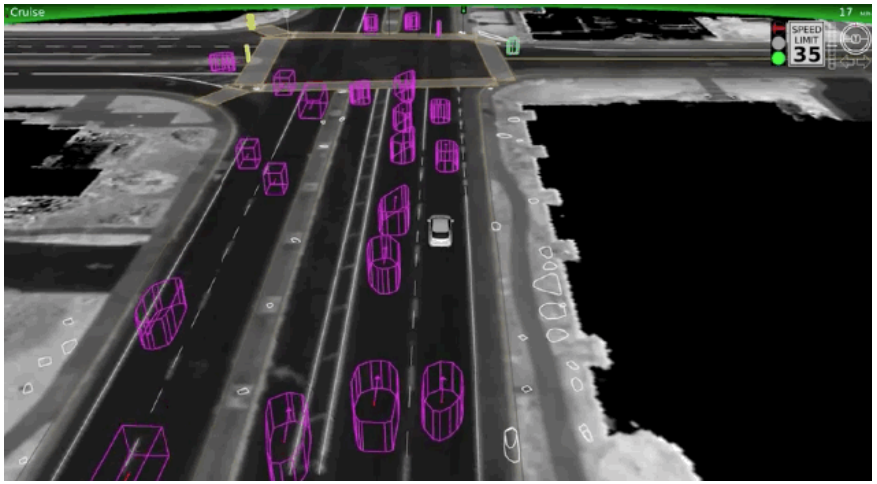
Character Recognition



Chess (IBM Deep Blue, 1997)

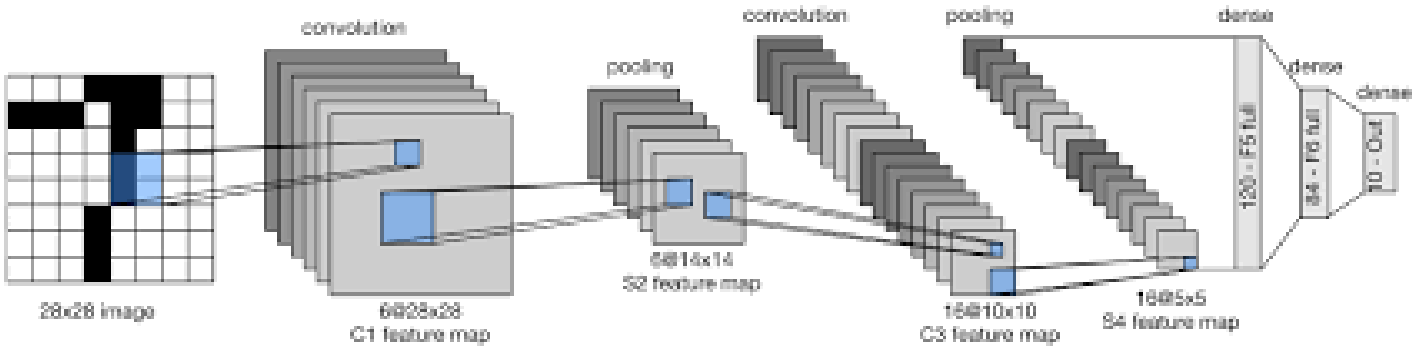


AlphaGo (AlphaGo Zero), 2015

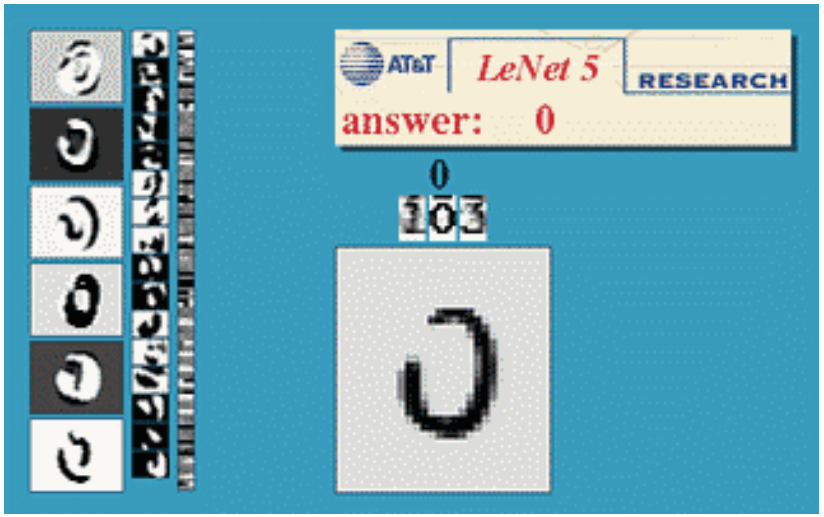


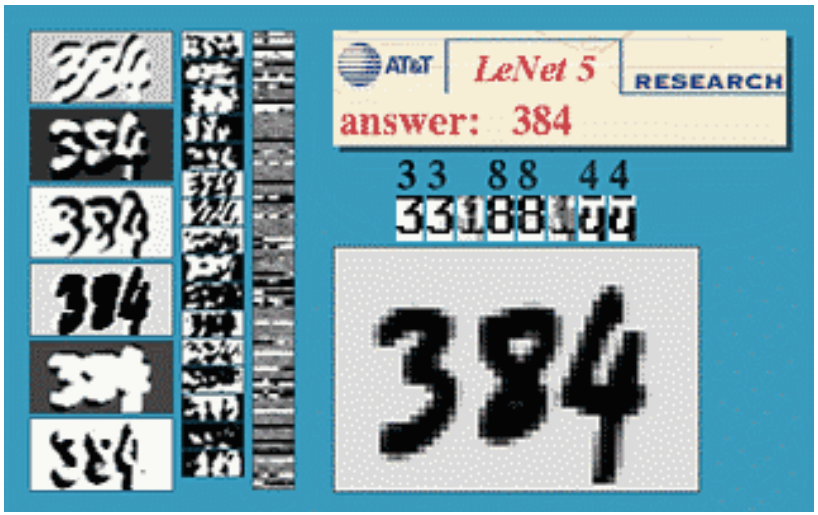
Self-driving Vehicles

LeNet-5 (Yann LeCun, 1989)



LeNet-5模型的總層數有6層(不含輸入與輸出層)，包括**C1卷積層**(Convolution)、**S2池化層**(Pooling/Sub Sampling)、**C3卷積層**、**S4池化層**、**C5卷積層**、**F6全連接層**(Fully connection)等。





Deep Blue (1997)

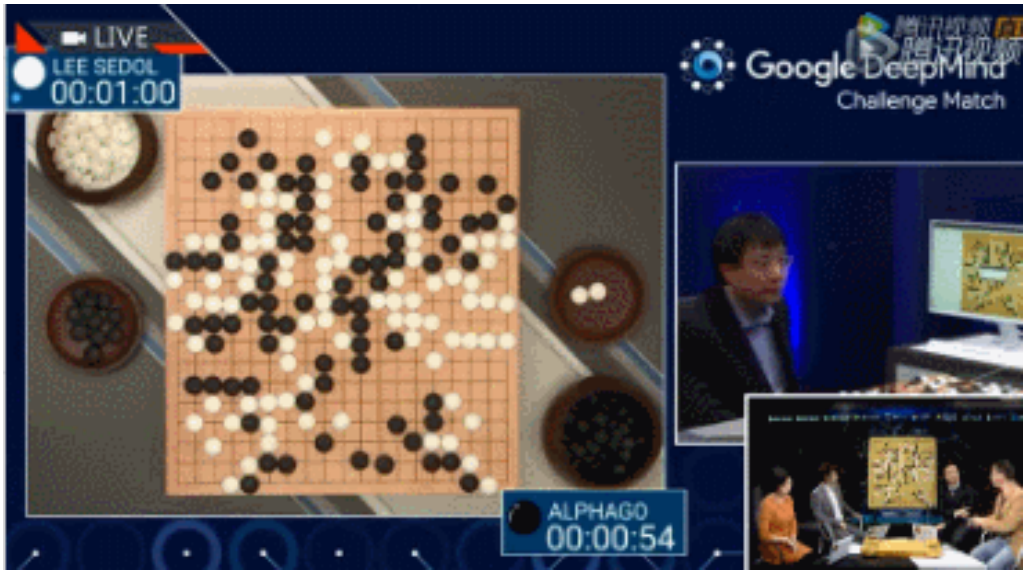
- **Deep Blue**, the IBM RS/6000 parallel supercomputer that defeated Chess Grandmaster Garry Kasparov in **May 1997**.

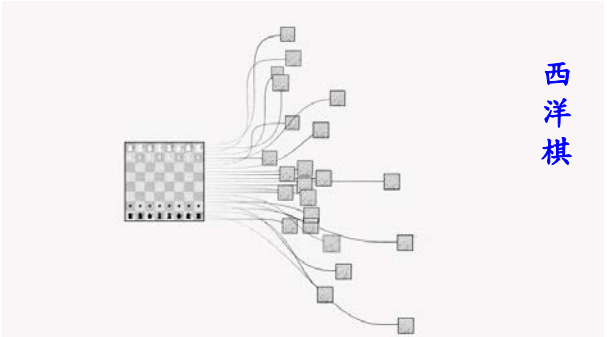
西洋棋的分支因數大概是40左右，這表示預測之後20步的動作需要計算40的20次方

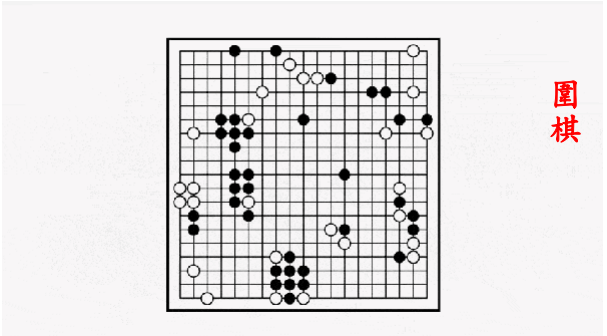
Deep Blue基本上是根據MinMax搜索算法與Alpha-Beta修剪法，縮減可能的計算範圍，找出最佳策略

AlphaGo (2015)

- AlphaGo is a computer program that plays the board game Go. It was developed by Alphabet Inc.'s Google DeepMind in London. AlphaGo has several versions including AlphaGo Lee, AlphaGo Master, AlphaGo Zero, etc.



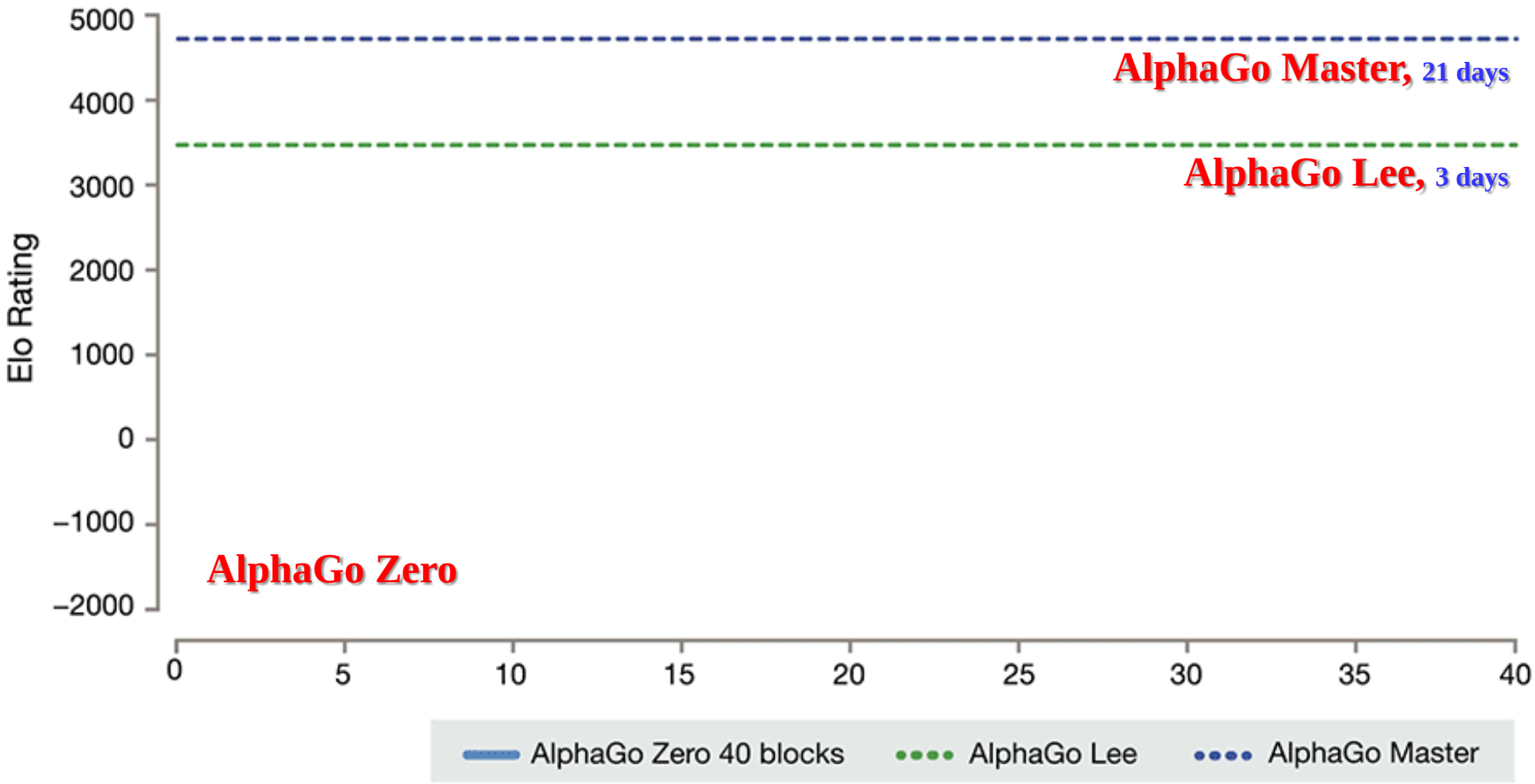




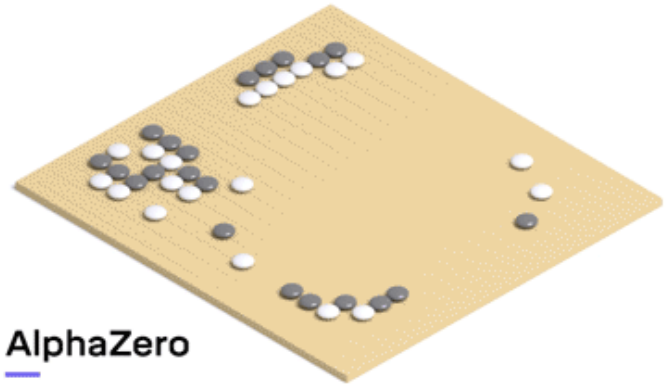
- 2016年3月9日，AlphaGo與韓國圍棋冠軍李世乭，進行5場對奕。最終比數4：1，人工智慧勝過人腦，AlphaGo也擠上世界排名第4名。

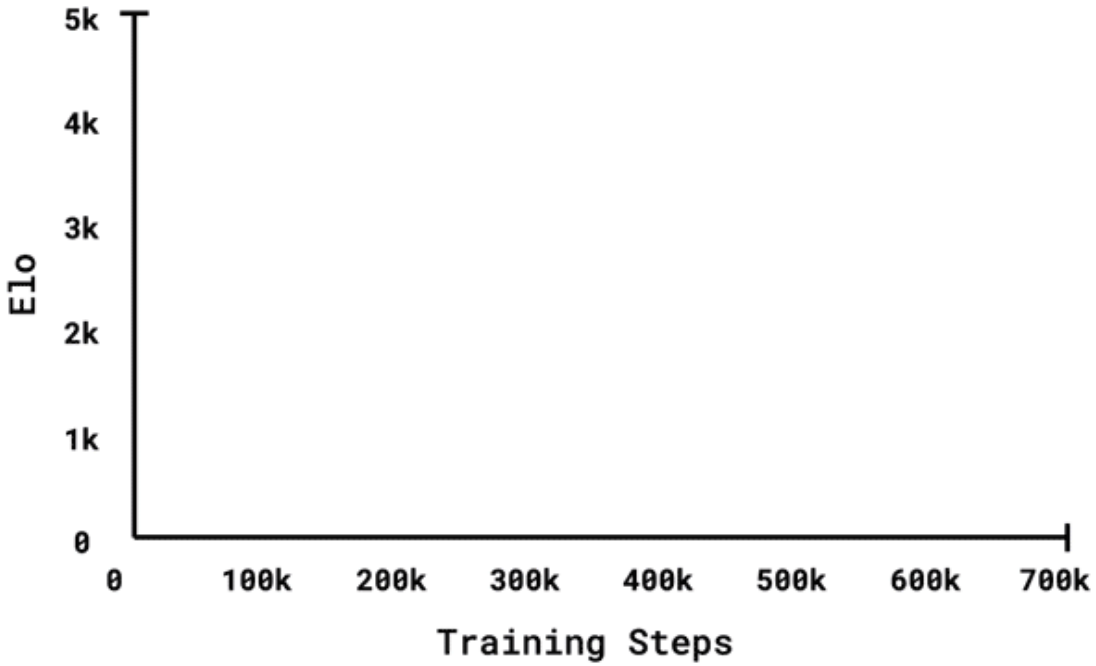
圍棋的分支因數是250，以圍棋19*19的方陣，共有361個落子點，所以整個圍棋棋局的總排列組合數高達10的171次方

AlphaGo Zero (2017)



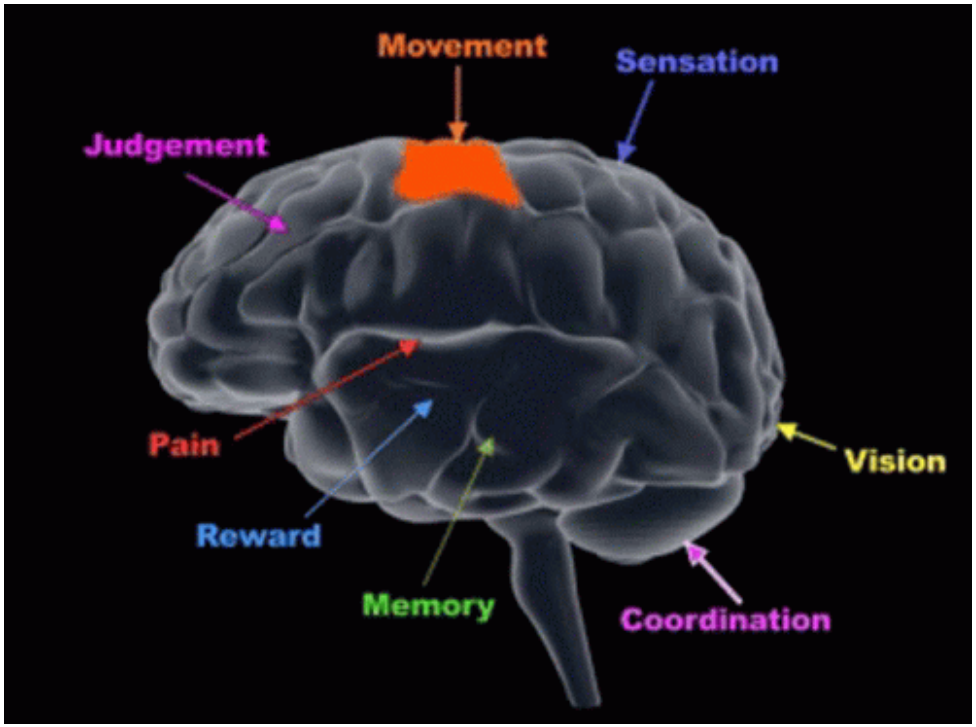
AlphaZero (2018) (reinforcement learning)





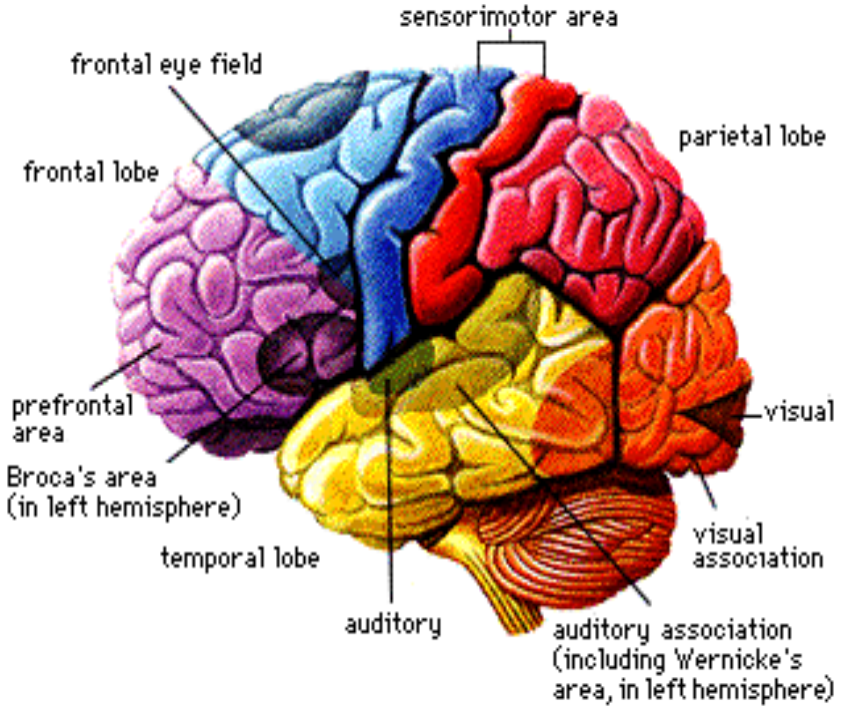
- In chess (西洋棋), AlphaZero first outperformed Stockfish after just **4 hours**.
- In shogi (將棋), AlphaZero first outperformed Elmo after **2 hours**.
- In Go (圍棋), AlphaZero first outperformed the version of AlphaGo that beat the legendary player Lee Sedol in 2016 after **30 hours**.

Why human can learn !



Human Brain Properties

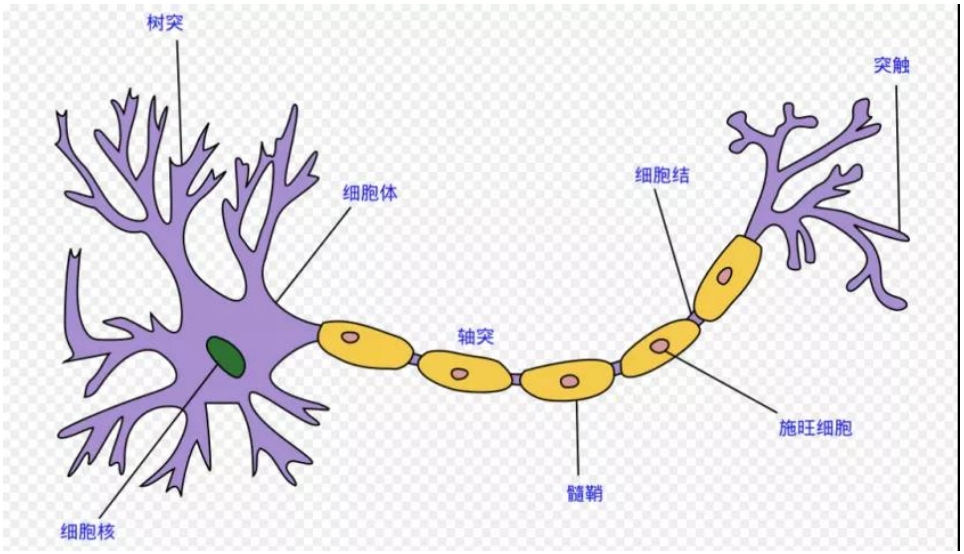
- Biological neural networks
  - 10^{11} neurons (1000 億個神經元)
  - Only a small portion of these cells are used
  - Main features
    - distributed nature parallel processing
    - each region of the brain controls specialized task(s)
    - no cell contains too much information: simple and small processors
    - information is saved mainly in the connections among neurons



Biological Neuron

- 大腦的神經網路是由「神經元」組成，神經元是用於感知環境的變化，再將信息傳遞給其他的神經元，基本構造包括**細胞體**(內含細胞核)、**樹突**(dendrites)、**軸突**(axon)組成。

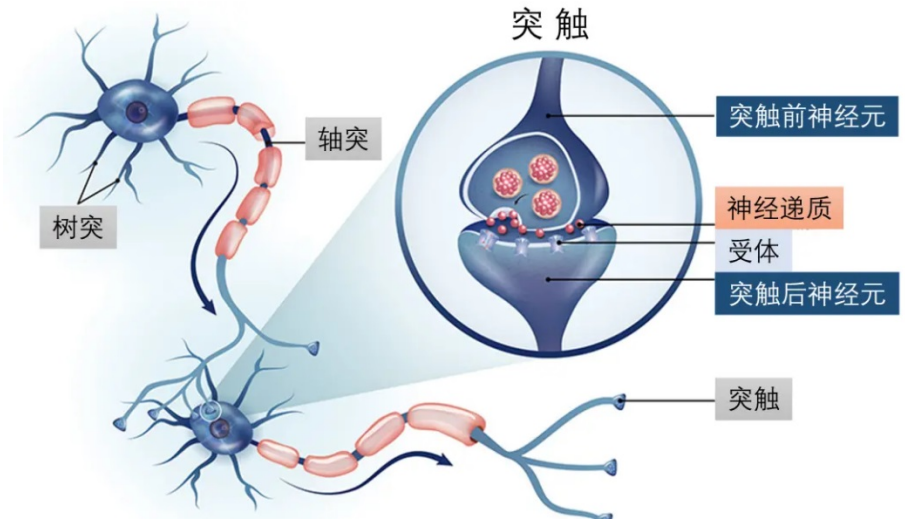
- **樹突**(Dendrites)是從神經元細胞體發出的多分支突起。樹突為神經元的輸入通道，其功能是將自其他神經元所接收的動作電位(電信號)傳送至細胞本體。其他神經元的動作電位藉由位於樹突分支上的多個**突觸**傳送至樹突。
- **軸突**(Axon)由神經元組成，即神經細胞之細胞體長出突起，功能為傳遞細胞本體之動作電位至**突觸**(Synapse)。神經系統中，軸突為主要神經信號傳遞渠道。
- **樹突負責將資訊帶回細胞(input)，而軸突則是負責將訊息傳遞出去(output)。**

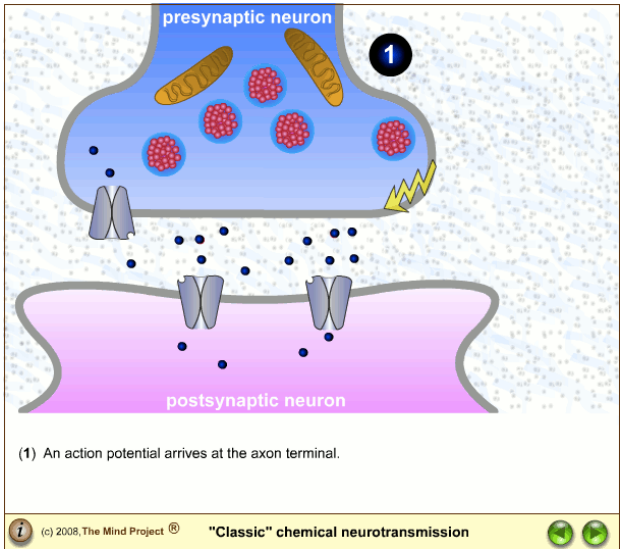


- **突觸**是神經元之間通信的特異性接頭(junction)。突觸可分為化學突觸與電突觸兩大類。
  - **化學突觸**的工作機制稱為神經遞質的信號分子的釋放與接收，兩個神經元之間沒有直接的電氣耦合。
  - **電突觸**是兩個神經元之間的直接電氣耦合。

Biological Neuron

- 神經元之間會互相連接形成複雜的網路，當神經元受到活化時，細胞內外離子的濃度會改變，產生非常微小的電位訊號及相對應的電流。這個電流沿著電線般長長的神經元纖維，傳遞至下一個神經元。

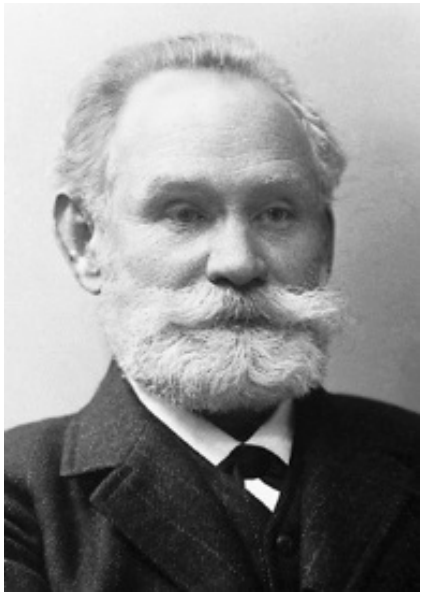




- 在神經元相接的地方(**突觸**)，因為電流無法直接跨越細胞，電訊號會活化神經末端釋放特定的化學分子，稱為神經傳導物質(neurotransmitter)，利用轉換後的化學訊號活化下一個神經元，進行神經電訊號的傳遞。

(1) Classical Conditioning (古典制約, 1897)

- 所謂古典制約(classical conditioning)就是藉由反覆的人為干預，使得原本不存在關聯的兩個事件(一為條件刺激S，一為生理反應R)之間建立起聯繫。
- **制約(conditioning)就是學習歷程**。



Pavlov(巴夫洛夫)

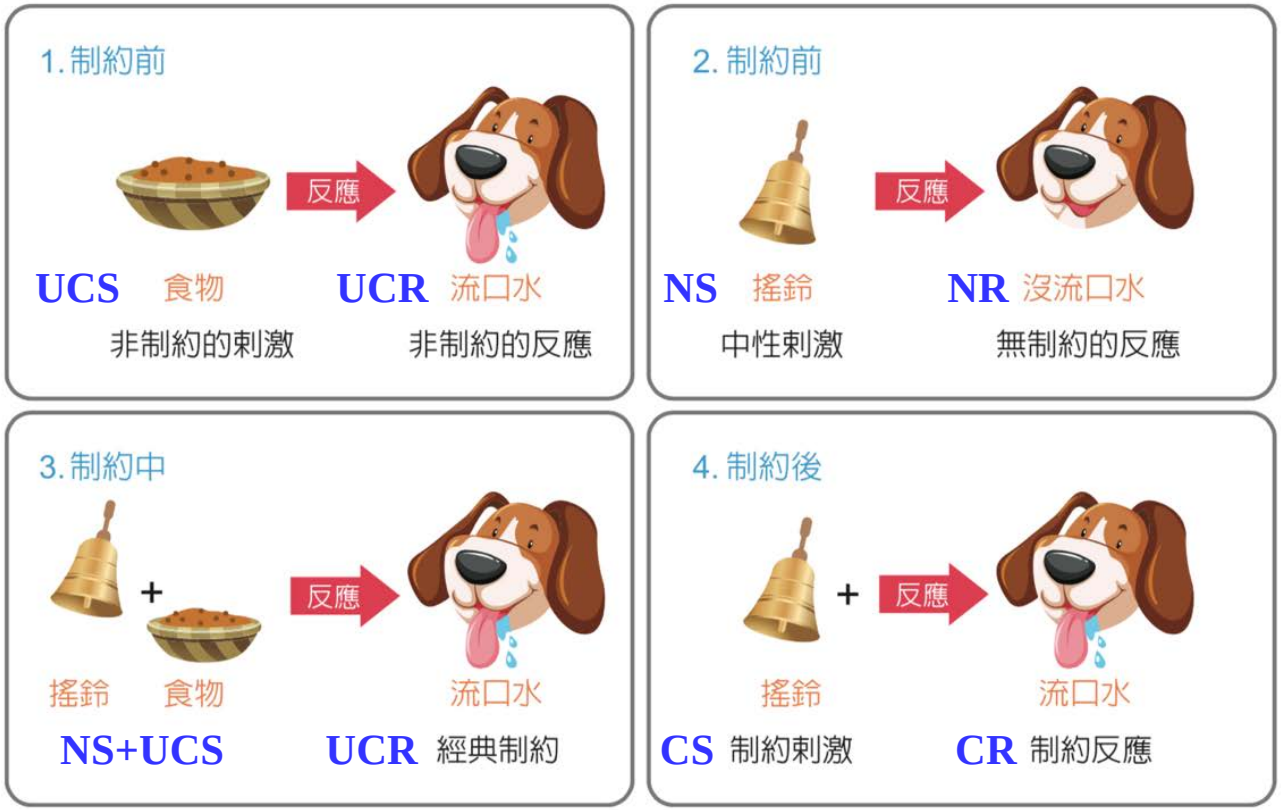


圖 9-2 巴夫洛夫的狗實驗示意圖

@帕夫洛夫狗狗制約實驗

(1) Classical Conditioning (cont.)

(1) 制約前

UCS

UCR

NS

(2) 制約前

NR

NS

(3) 制約中

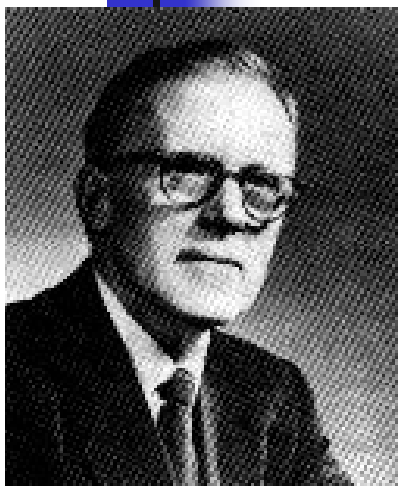
UCR

CS

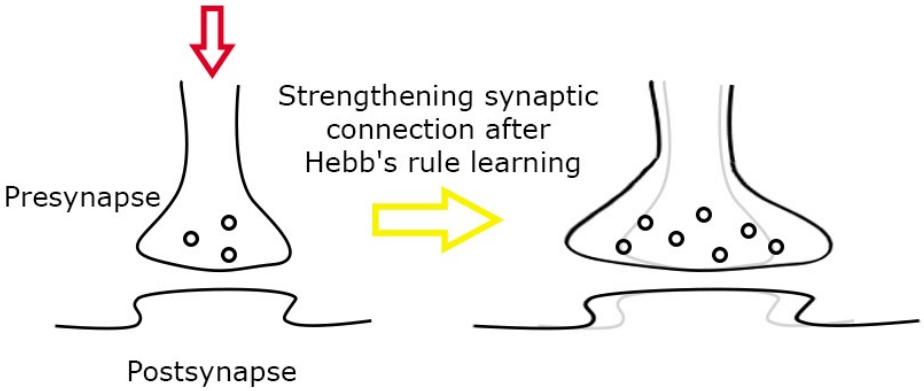
(4) 制約後

CR

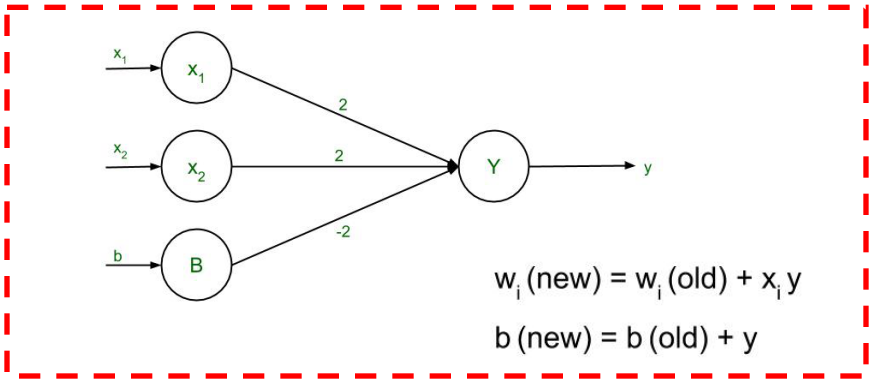
(2) Hebbian Theory (赫布理論, 1949)



- 赫布理論解釋突觸前(presynapse)神經元向突觸後(postsynapse)神經元的持續重複的刺激，可以導致突觸傳遞的效能增加。赫布理論解釋學習過程人腦的神經元發生的變化。

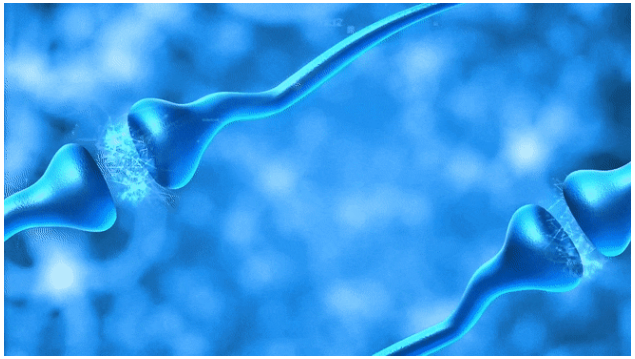


- 在人工神經網路中，突觸間傳遞作用的變化視為是神經元網路中**相對應權重(weight)的變化**。如果兩個神經元同步激發，則它們之間的權重增加。如果單獨激發，則權重減少 (**唯變所適、用進廢退**)。
- **赫布學習規是最簡單的神經元學習規則**。

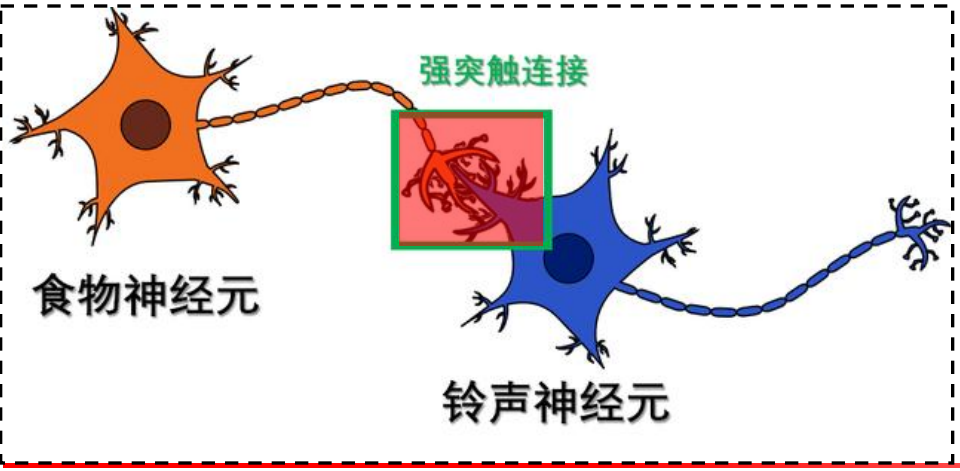


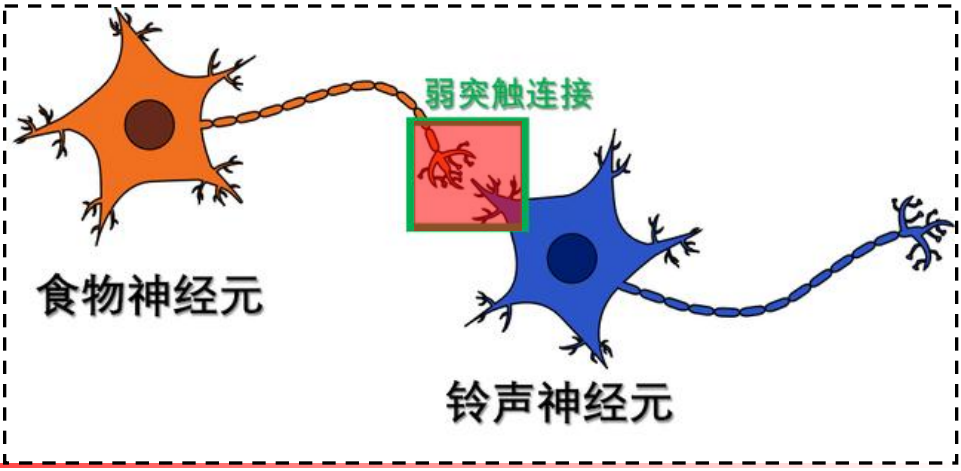
(3) Neuroplasticity (R. J. Davidson, 1992)

- Neuroplasticity(神經可塑性)是指重複性的經驗可以改變大腦的結構。神經元能夠在化學上、結構上、甚至是功能上改變自己，根據外部世界不斷優化大腦的神經網路。



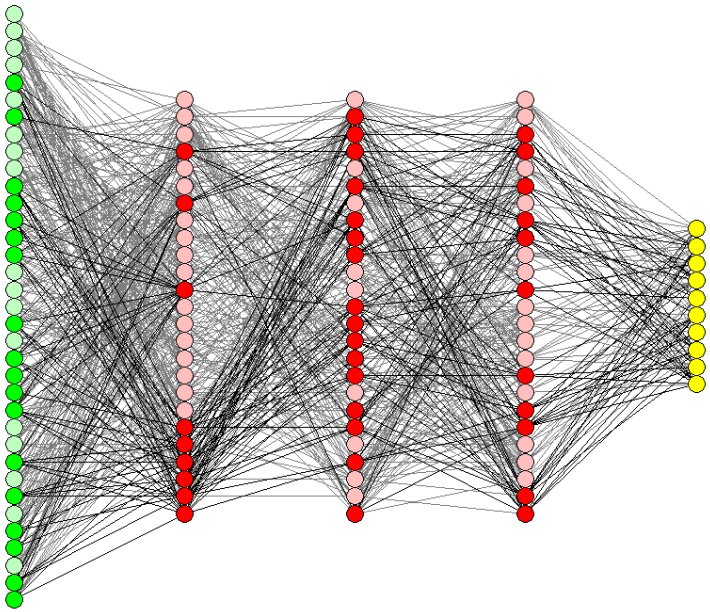
- **Left: As the neurons fire, they wire together to strengthen the Synapse (*the place where neurons connect with each other*).**
- **Right: As the neurons fall out of sync, the Synapse weakens and un-links.**



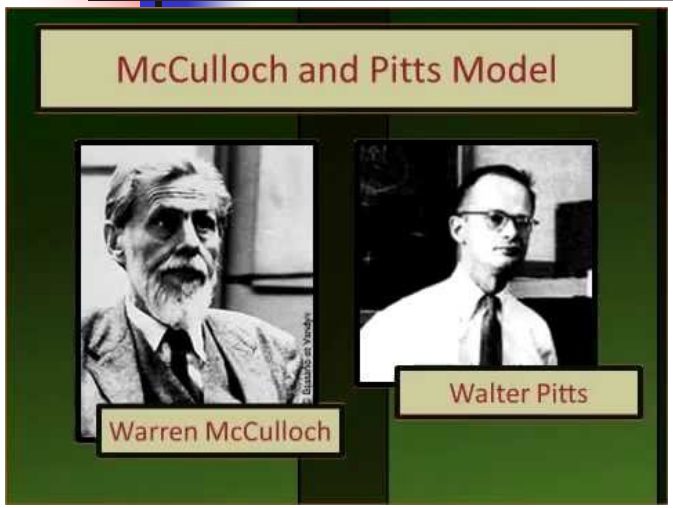


Machine Learning

learning the model from the data (data-driven model)



McCulloch and Pitts model (1943)



- 1943年McCulloch與Pitts仿造人腦的神經細胞結構，設計第一個軟體實現的人工神經元。
- 人工神經元如同數位邏輯電路的邏輯閘，輸入依照計算邏輯而輸出相對應的結果，若其值超過某一門限值(threshold) ，神經元輸出就會等於1 (on)，若沒有超過門限值，則為0 (off)。

https://articles.adxy.in

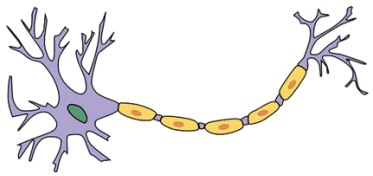


Fig: Biological Neuron

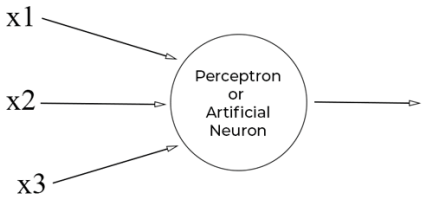
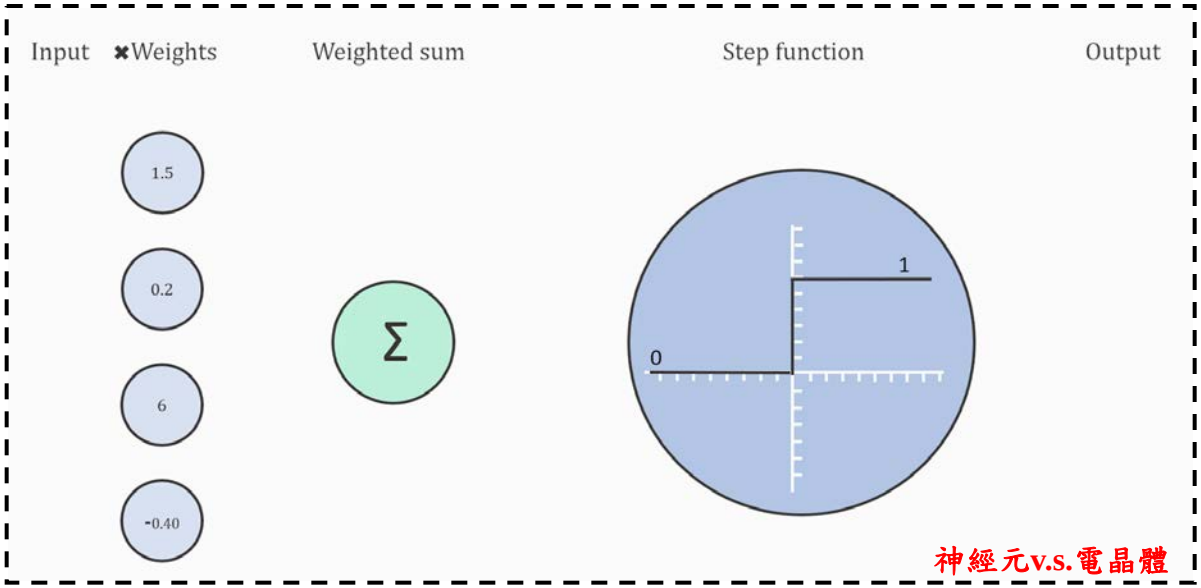
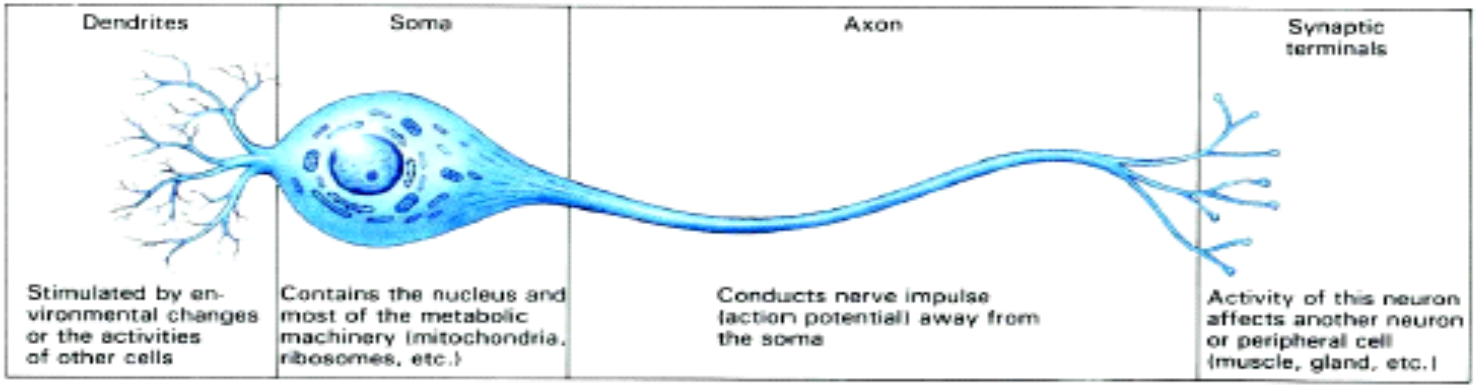


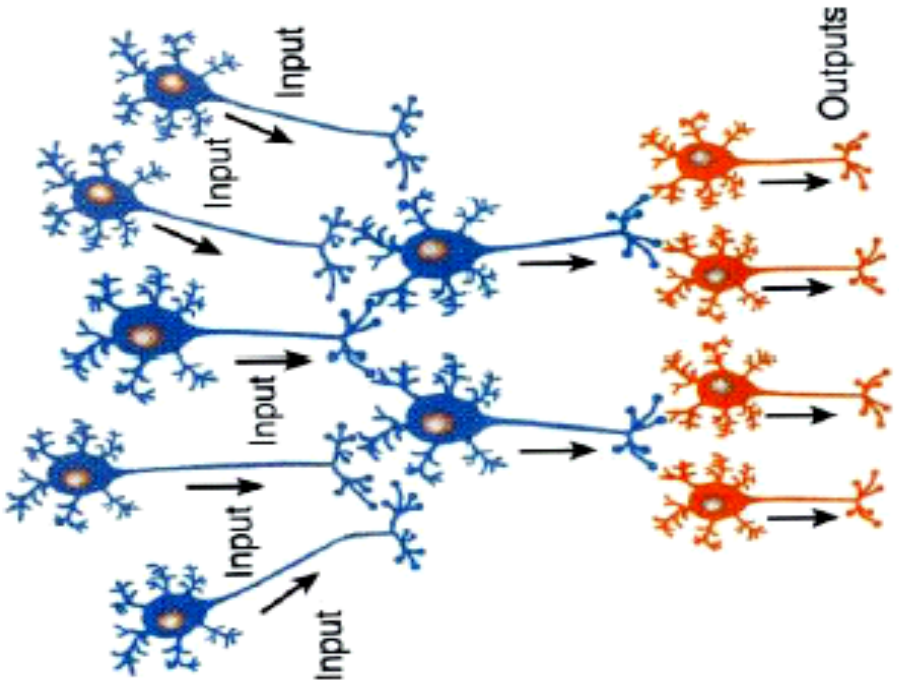
Fig: Artificial Neuron

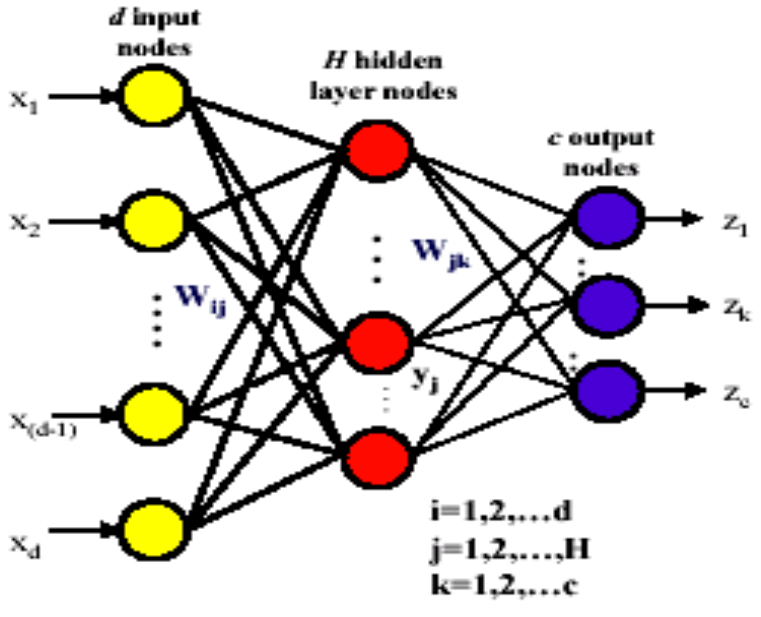


神經元v.s.電晶體

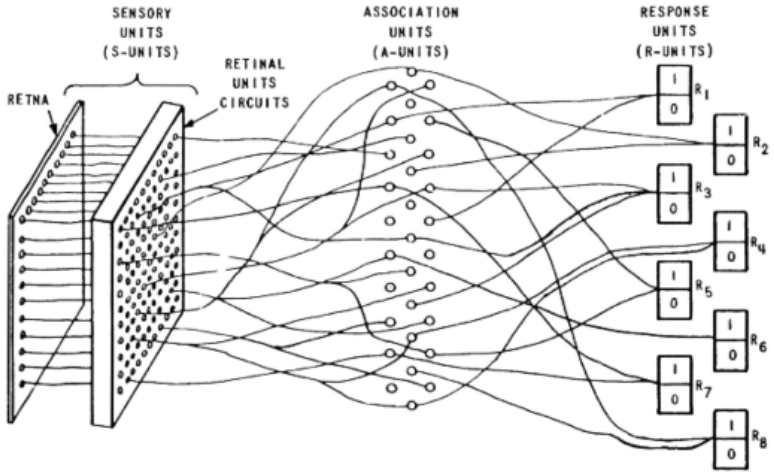
Biological Neuron to Artificial Neuron



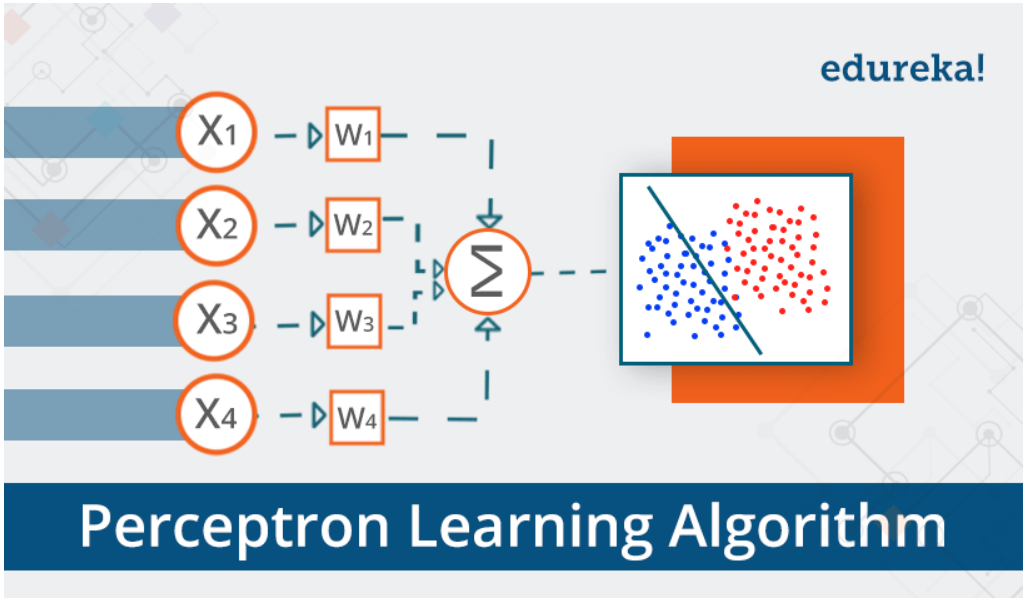


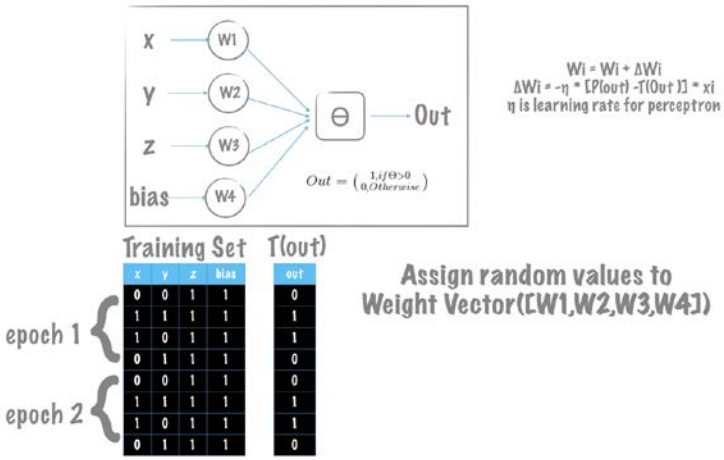


Perceptron (Rosenblatt, 1957)

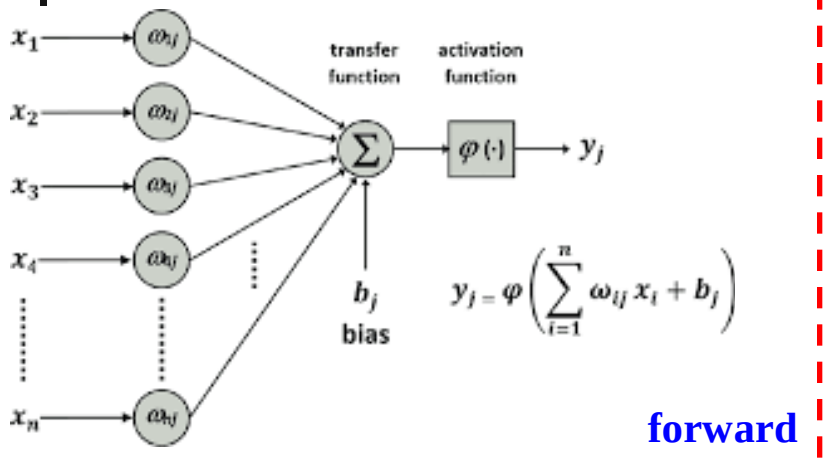


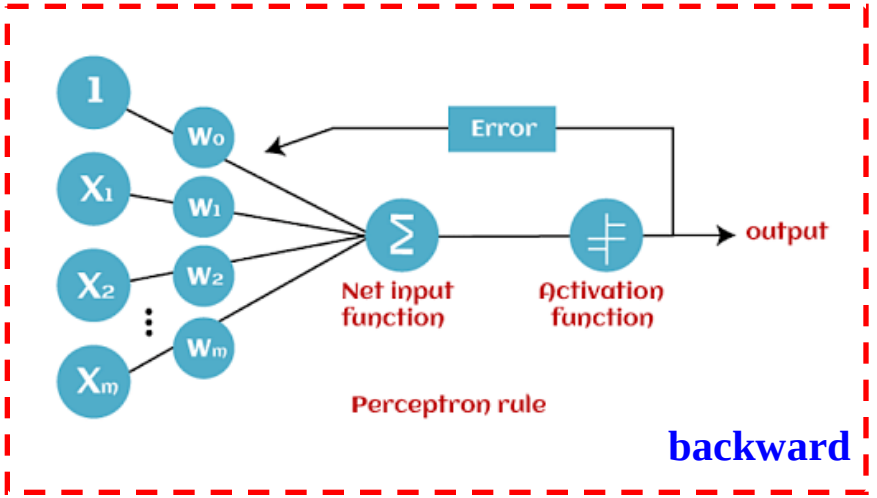
F. Rosenblatt

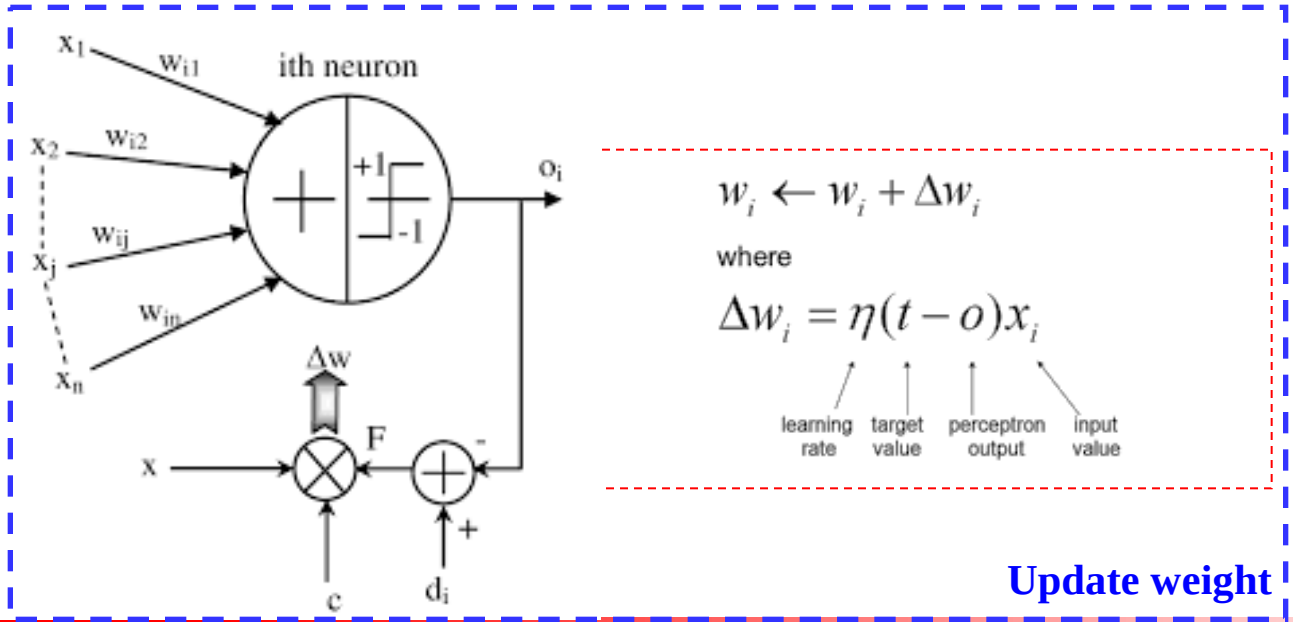




Perceptron Learning Algorithm







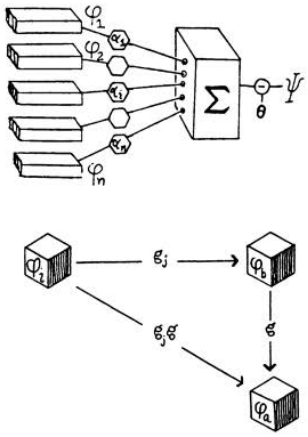
$$w_i \leftarrow w_i + \Delta w_i$$

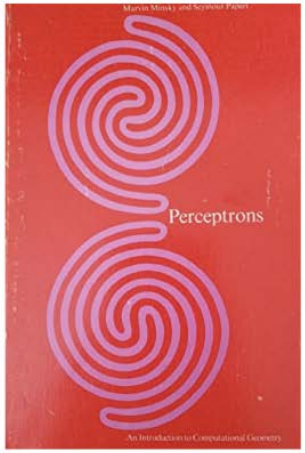
where

$$\Delta w_i = \eta (t - o) x_i$$

learning rate | target value | perceptron output | input value

Minsky and Papert (1969)

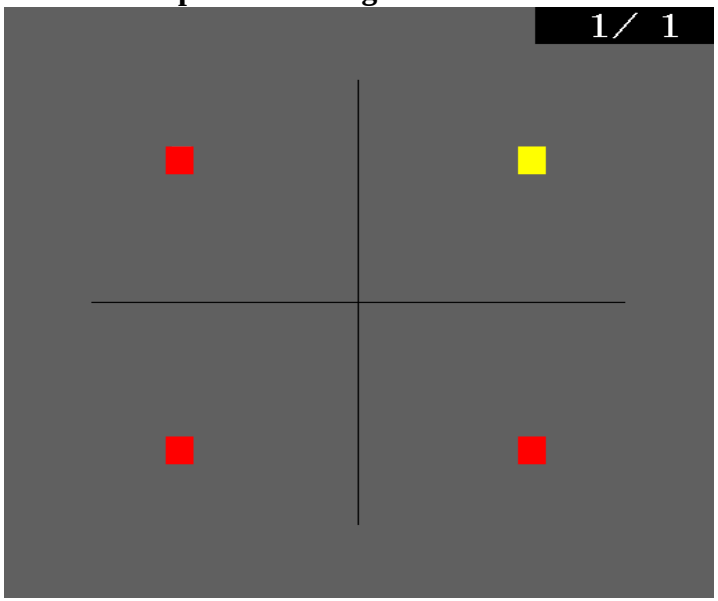




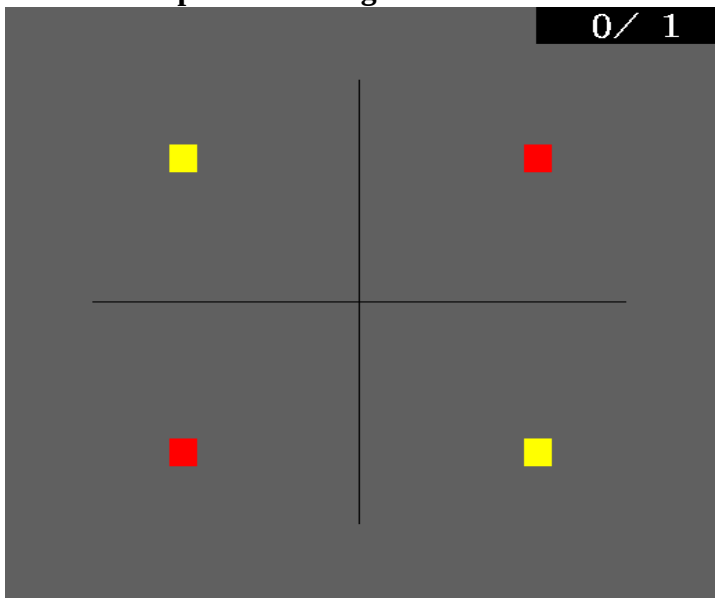
M. Minsky

S. Papert

Perceptron learning of AND function

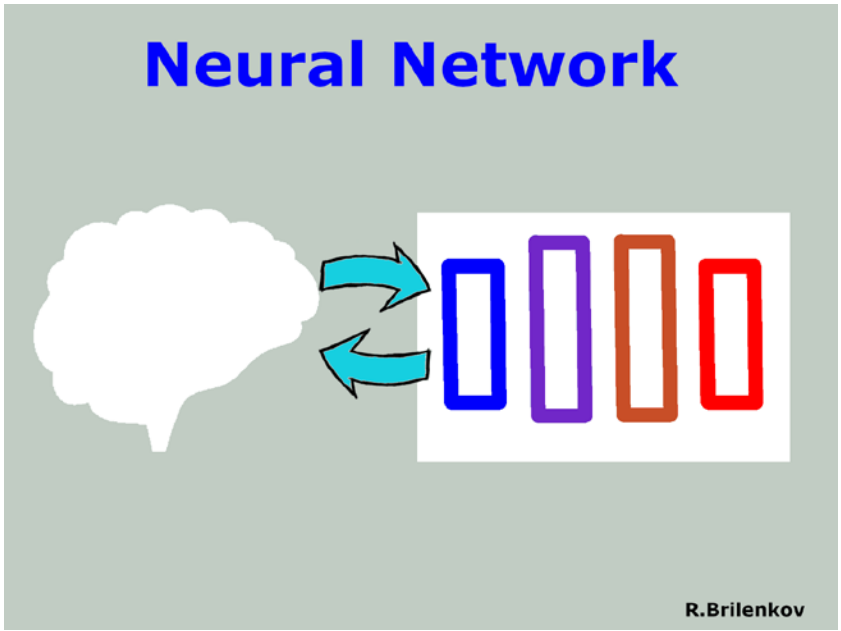


Perceptron learning of XOR function

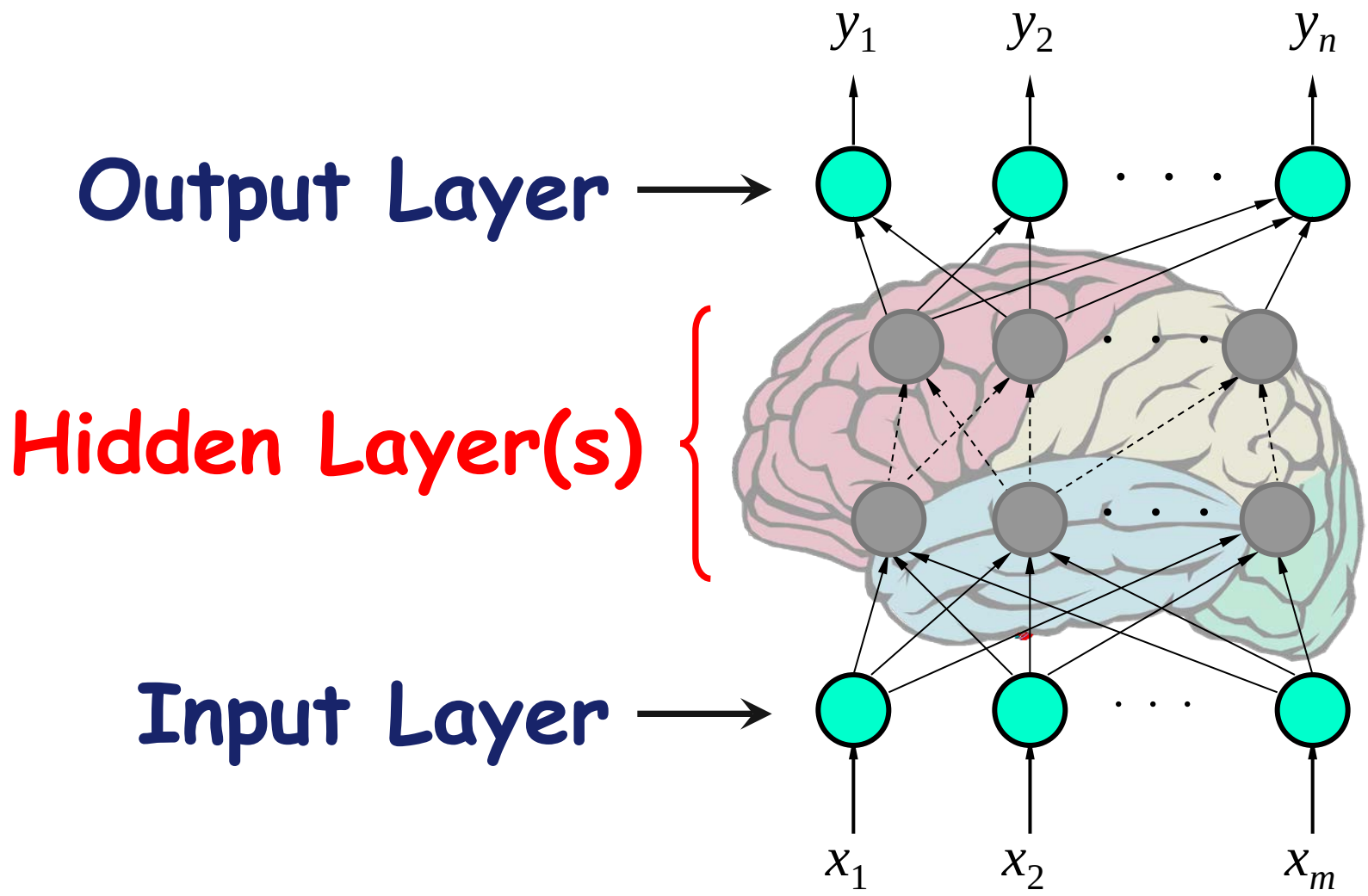


- **The exclusive or (XOR) problem cannot be solved by perceptron.**
- **The solution is adding more hidden layers.**

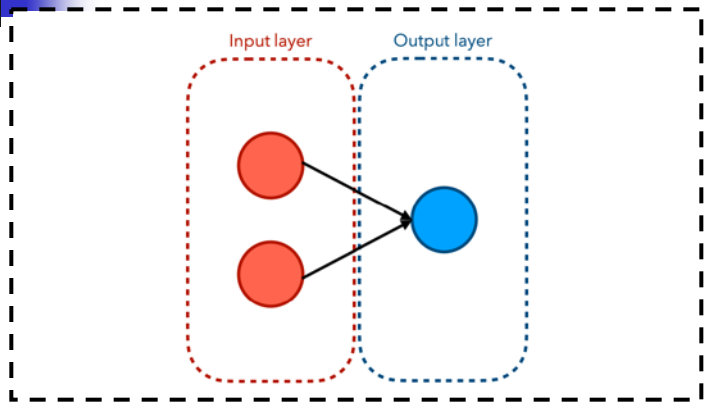
Neural Network

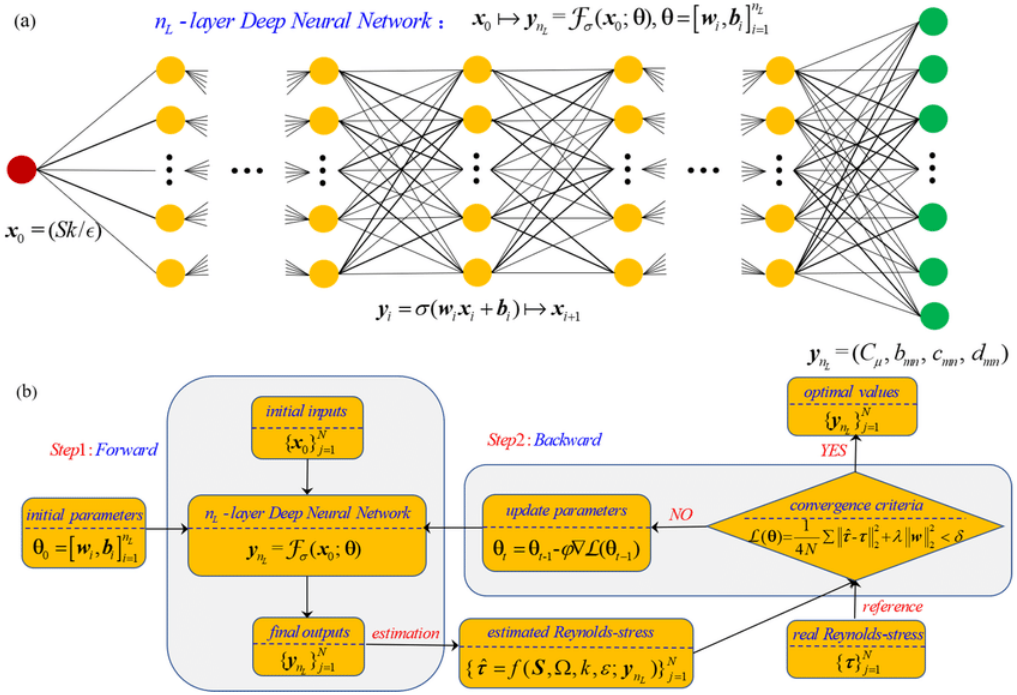


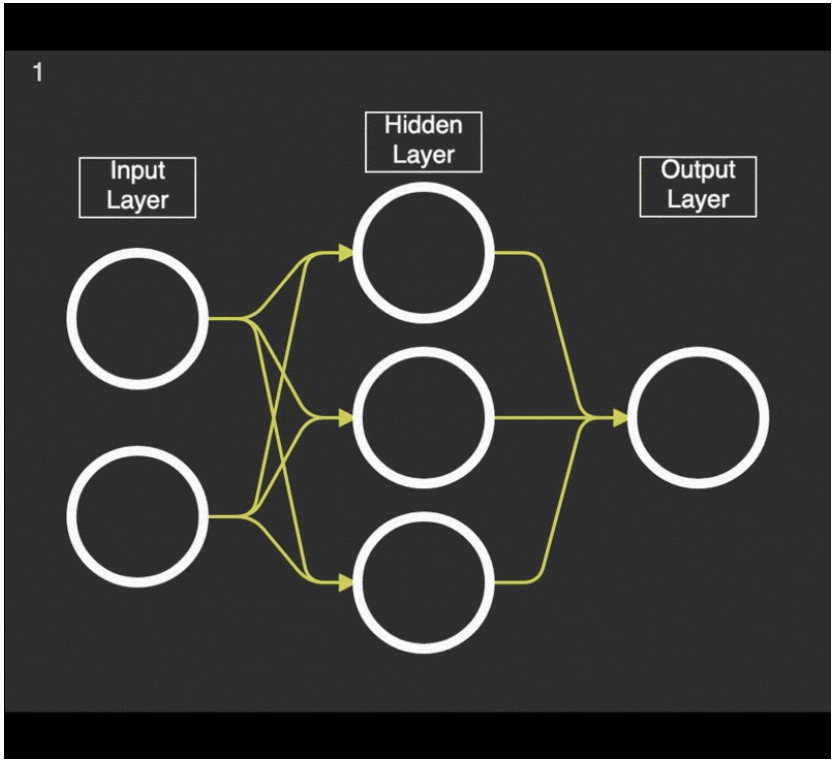
Multi-layered perceptron NNs



Multi-layered perceptron NNs

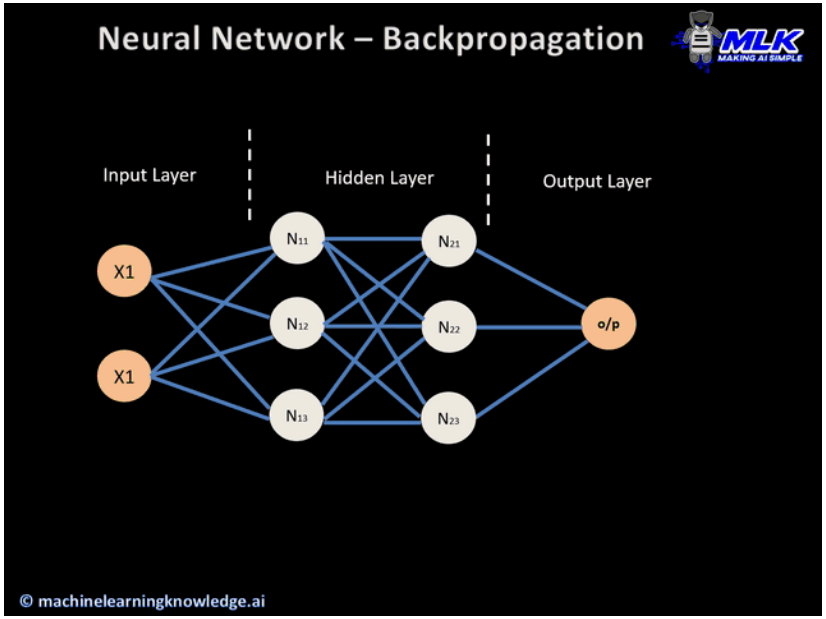


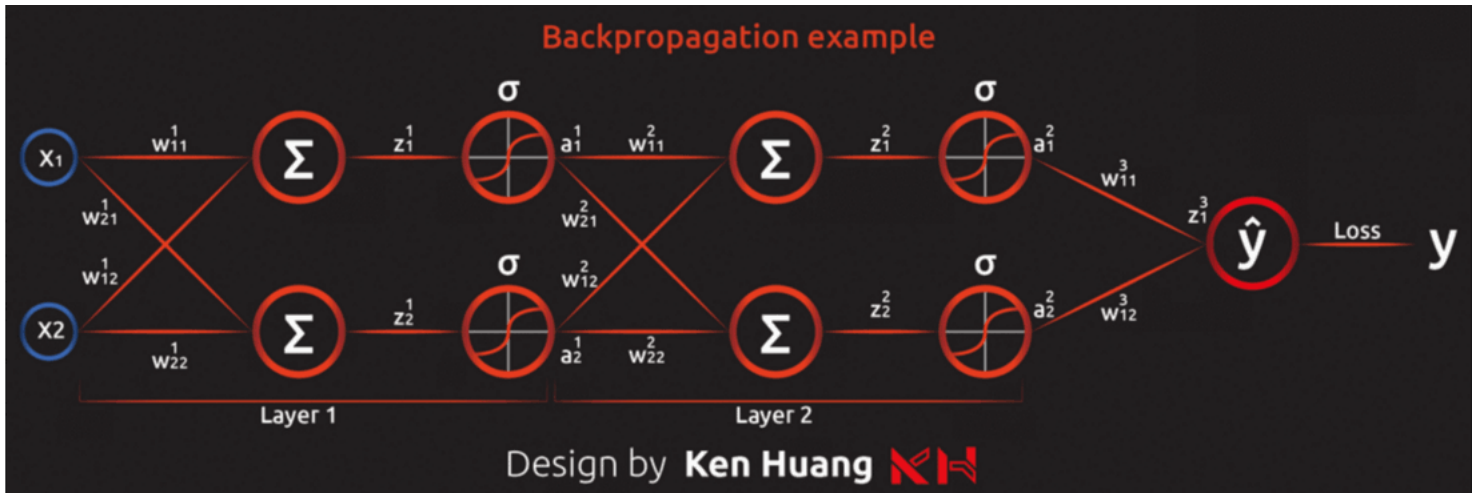




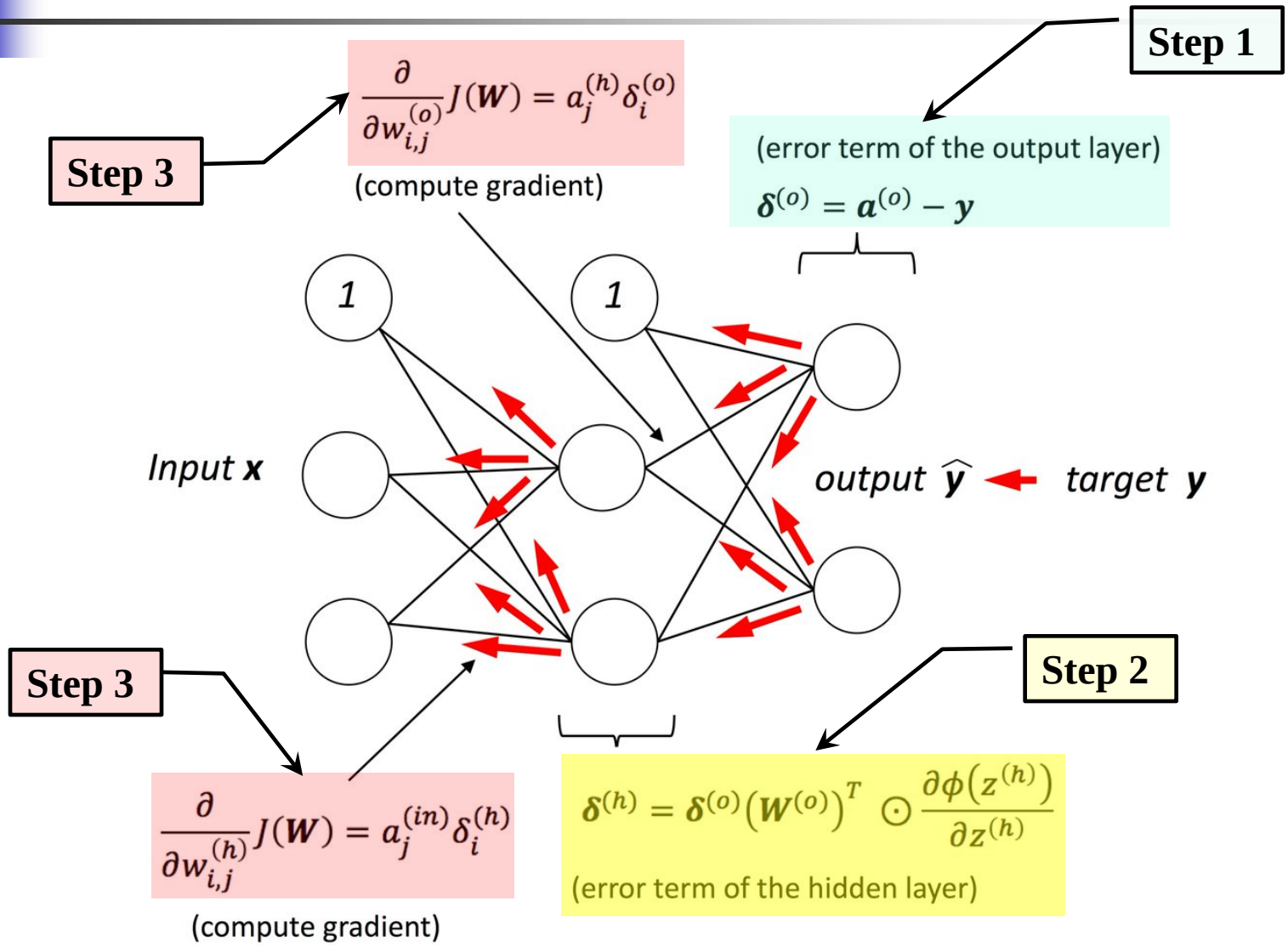
Feedfoward and backpropagation

Back Propagation, BP (多變數連鎖率) (Rumelhart, 1986)

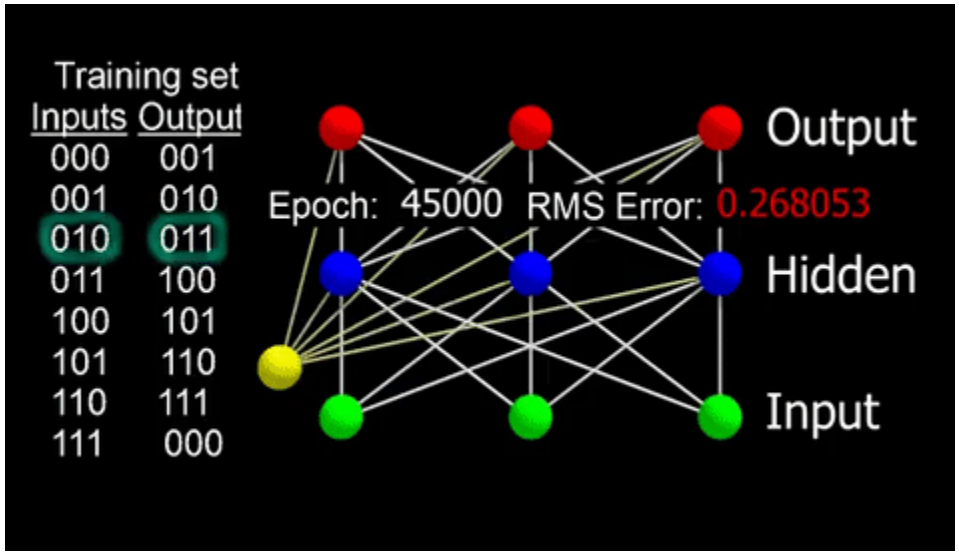


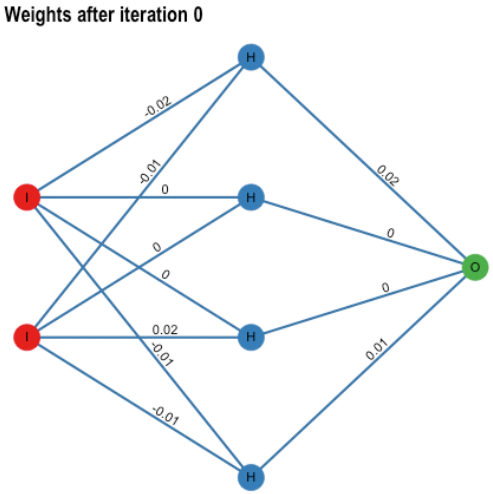


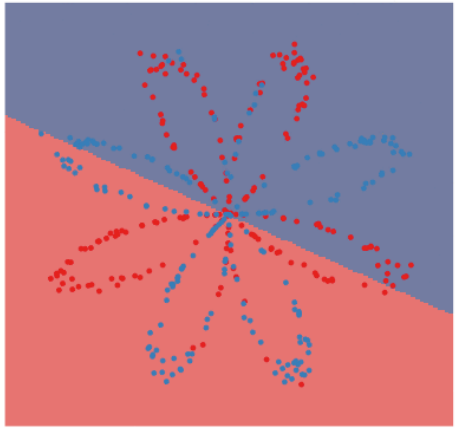
Back Propagation, BP



Simulation


Training set
Inputs Output
000 001
001 010
010 011
011 100
100 101
101 110
110 111
111 000
Epoch: 45000 RMS Error: 0.268053
Output
Hidden
Input


Weights after iteration 0
H
I
O
-0.02
-0.01
0
0.02
0
0
0
0
0.02
-0.01
0.01
-0.01

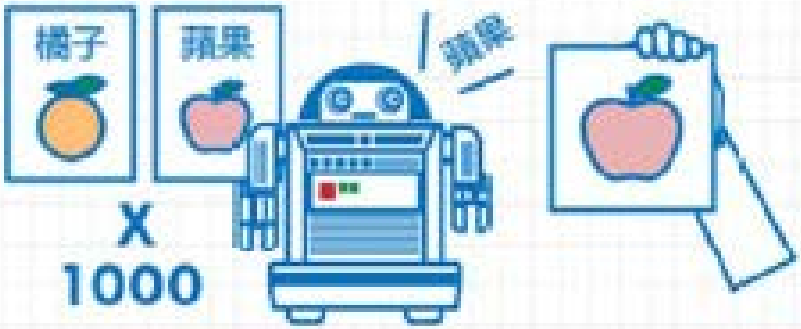

Decision boundaries after iteration 0


Cost after iteration 0

Supervised v.s. Unsupervised Learning

監督式學習
Supervised Learning

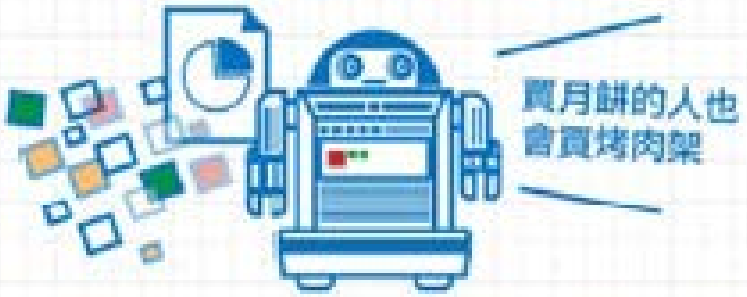
給予「有標籤」的資料



給機器各看了 1000 張標籤為蘋果和橘子的照片後、詢問機器新的一張照片中是蘋果還是橘子

非監督式學習
Unsupervised Learning

給予「無標籤」的資料，機器會自動中找出潛在的規則

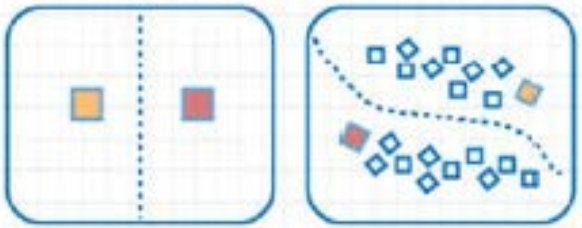


依據資料的分布、找到資料間的相似性；機器可能找出「買月餅的人也會買烤肉架」這個關聯

Semi-Supervised Learning

半監督學習
Semi-supervised learning

少部分資料有標籤，而大部分資料沒有標籤



先使用有標籤過的資料先切出一條分界線，再利用剩下無標籤資料的整體分布，調整出兩大類別的新分界。如此降低標籤資料的成本

增強學習
reinforcement learning

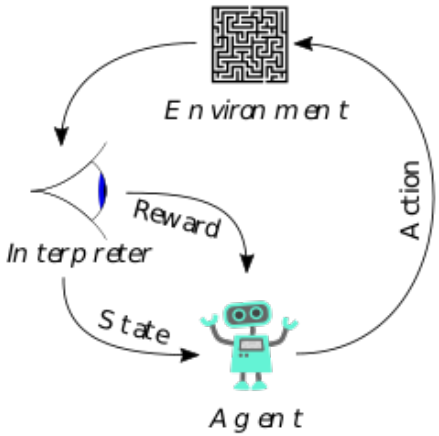
透過觀察環境而行動，並會隨時根據新進來的資料逐步修正、以獲得最大利益

機器人投籃，根據反饋的好壞，機器會自行逐步修正、最終得到正確的結果

資料科學家會依據資料量、資料類型與運算效能等現實情況，而選擇採用不同的模型。

Reinforcement Learning

- Reinforcement learning focuses on finding a balance between **exploration** (of uncharted territory) and **exploitation** (of current knowledge).

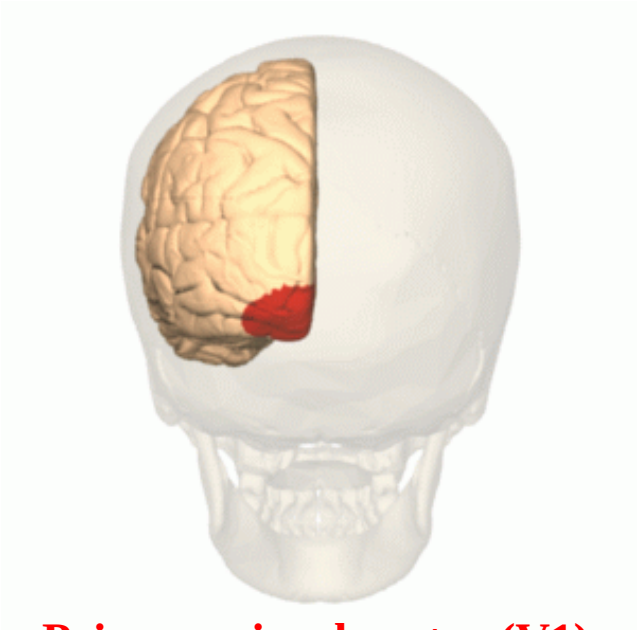


- The typical framing of a Reinforcement Learning (RL) scenario: an agent takes actions in an environment, which is interpreted into a reward and a representation of the state, which are fed back into the agent.

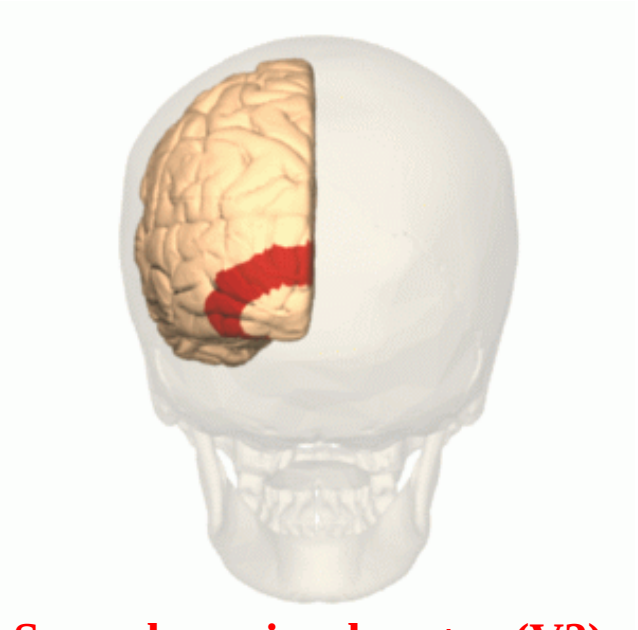
- 強化學習是以**制約刺激**為基礎，以**獎勵的刺激**鼓勵正向的行為，以**懲罰的刺激**減少負面的行為。

Deep Learning

Visual Cortex



Primary visual cortex (V1)



Secondary visual cortex (V2)

視覺系統

① 物體反射光線到並沿直線進入眼中

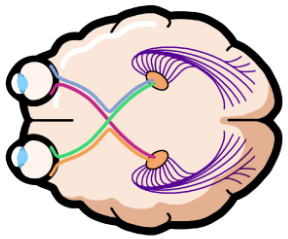
② 光線穿過角膜進入瞳孔並通過晶狀體

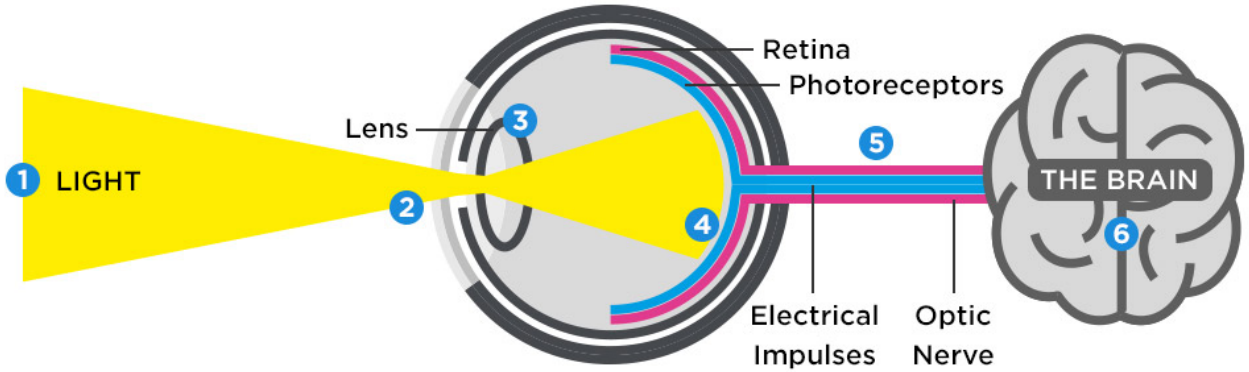
③ 光線經角膜及晶狀體彎曲(折射)後聚焦於視網膜

④ 視網膜上的感光細胞將光轉化成電脈衝

⑤ 電脈衝沿視神經傳遞到大腦

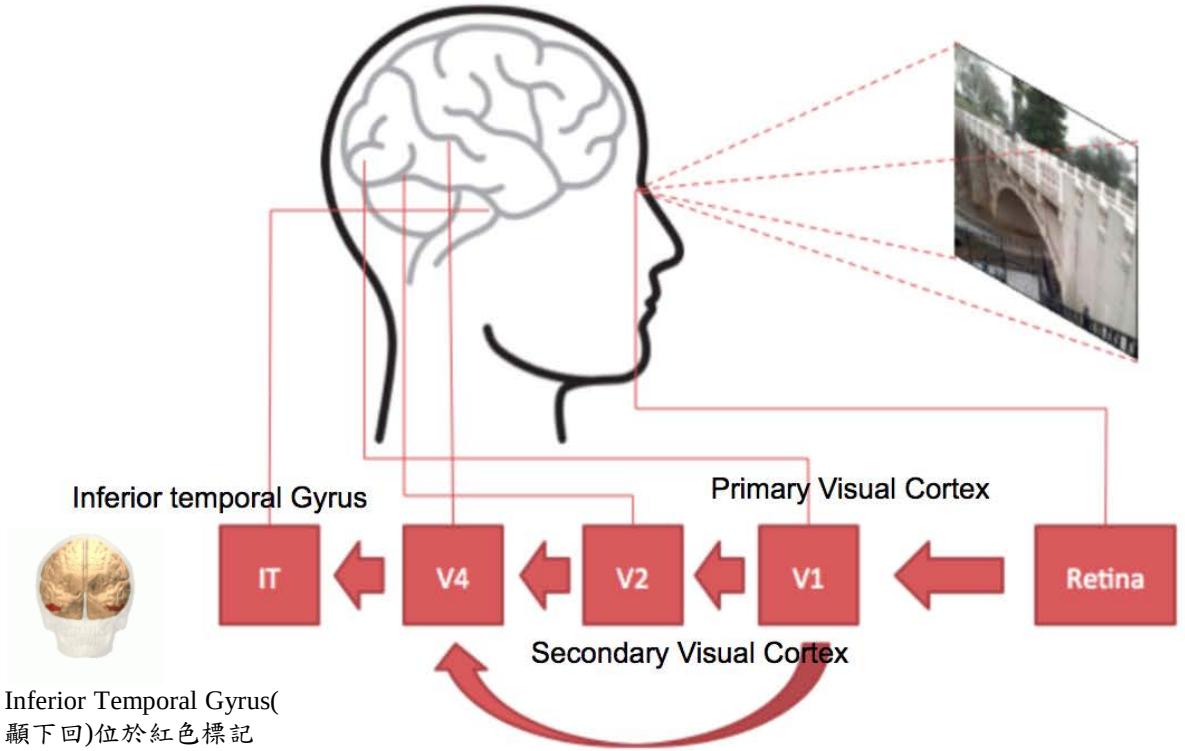
⑥ 大腦處理信號後形成圖像





視覺系統

- ✓ 當眼睛看外界物件時，資訊從視網膜流到**V1**(Primary Visual Cortex)，然後到**V2**(Secondary Visual Cortex)，**V4**，之後是**IT**(Inferior Temporal Gyrus，顳下回)。
- ✓ 人類的視覺系統是**分層遞進**，每一層級都比前一層級處理更高層次的概念。深度學習(或卷積神經網路)正是模仿人腦的分層架構產生，由底層**一層一層**的學習低階特徵，再慢慢地由**低階特徵組合出高階特徵**。



Inferior Temporal Gyrus(顳下回)位於紅色標記

V1: Edge detection, etc.

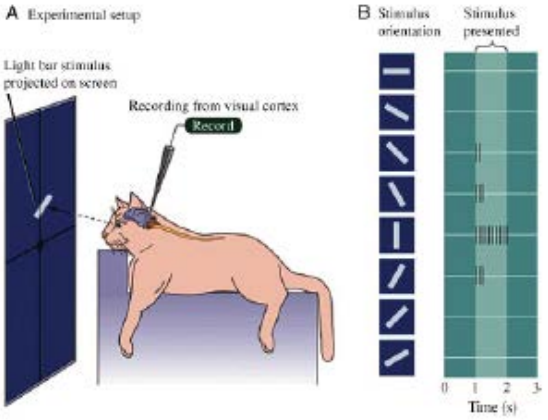
V2: Extract simple visual properties (orientation, spatial frequency, color, etc)

V4: Detect object features of intermediate complexity

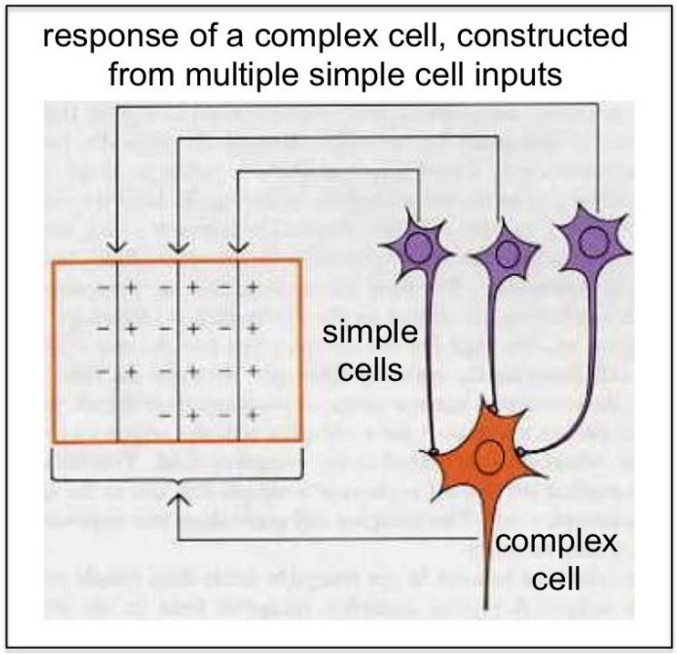
TI: Object recognition.

V1：邊緣檢測，V2：提取簡單的視覺要素(方向、空間、頻率、顏色等)，V4：監測物體的特徵，TI：物體識別

Hubel and Wiesel, 1959

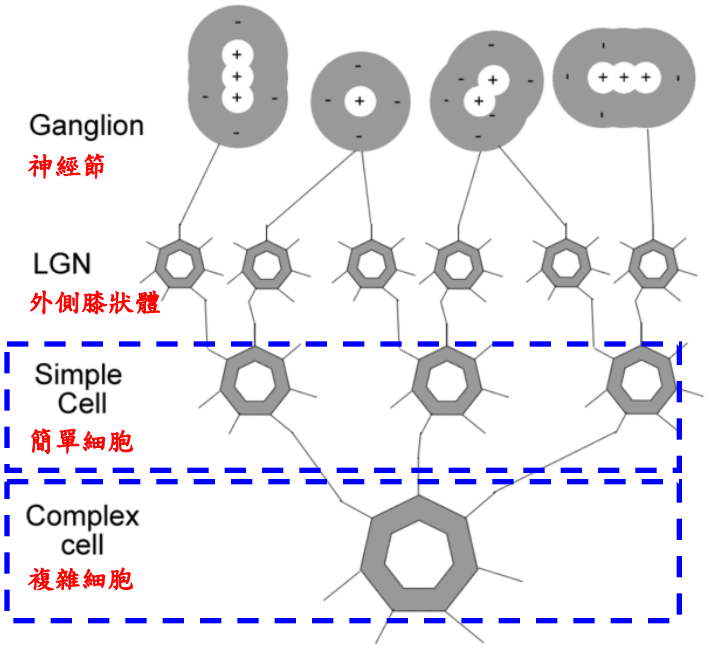


- 1950~1960年代，神經生理學家Hubel與Wiesel基於對貓與猴子視覺的研究，提出腦中兩種基本型態的視覺細胞，稱為**simple cells**與**complex cells**。這兩種細胞承擔不同層次的視覺感知功能。



- simple cells會被特定位置的鉛直邊緣形狀給激發，再將訊號傳給下一階段的complex cells。simple cell的感知野較小，會受圖形位置的影響激發與否
- complex cells在這樣的連結中卻不會受鉛直邊緣圖形所在位置，只要是在橘色長方形框中都可受到激發，所以感知野較大。
- **簡單細胞的感受野來自於將一系列的LGN細胞輸入的轉換，許多環狀的感受野就能組成一條線。**
- **複雜細胞的感受野來自於一系列空間傾向性特定排列的簡單細胞輸入的轉換。**

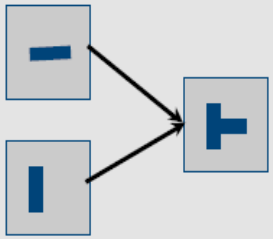
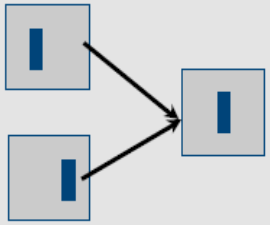
Hubel and Wiesel, 1959

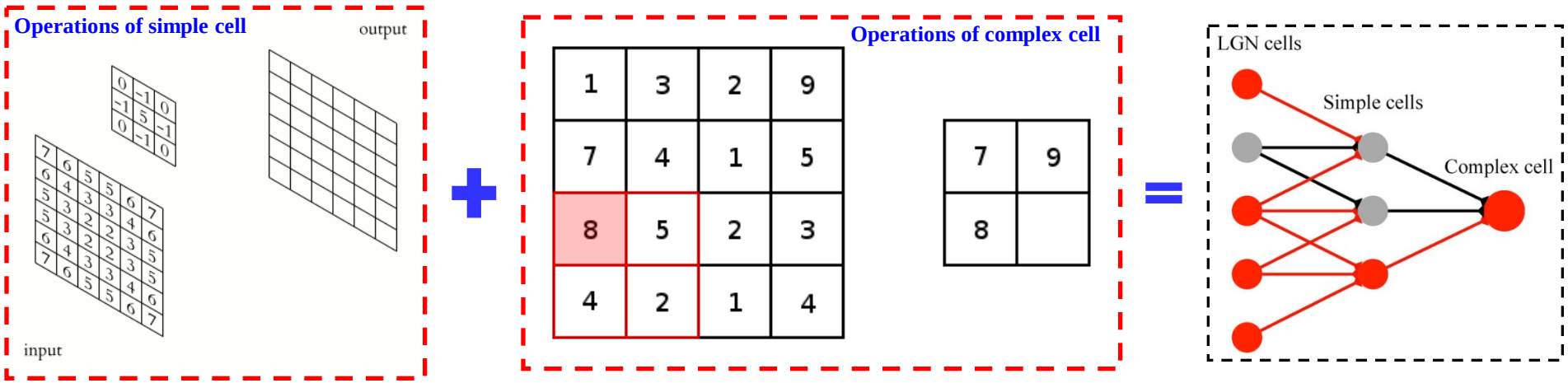


- V1中的**簡單細胞**(simple cell)的感受野是**狹長型**，每個簡單細胞只對感受野中**特定角度**(orientation)的光帶敏感。它們具有局部感受野，**卷積網路的卷積核據此設計**。
- V1中的**複雜細胞**(complex cell)對於感受野中以**特定方向**(direction)移動的**某種角度**的光帶敏感。它們用於回應簡單細胞檢測的特徵，且對於微小偏移具有不變形，這啟發**卷積網路的池化單元**。

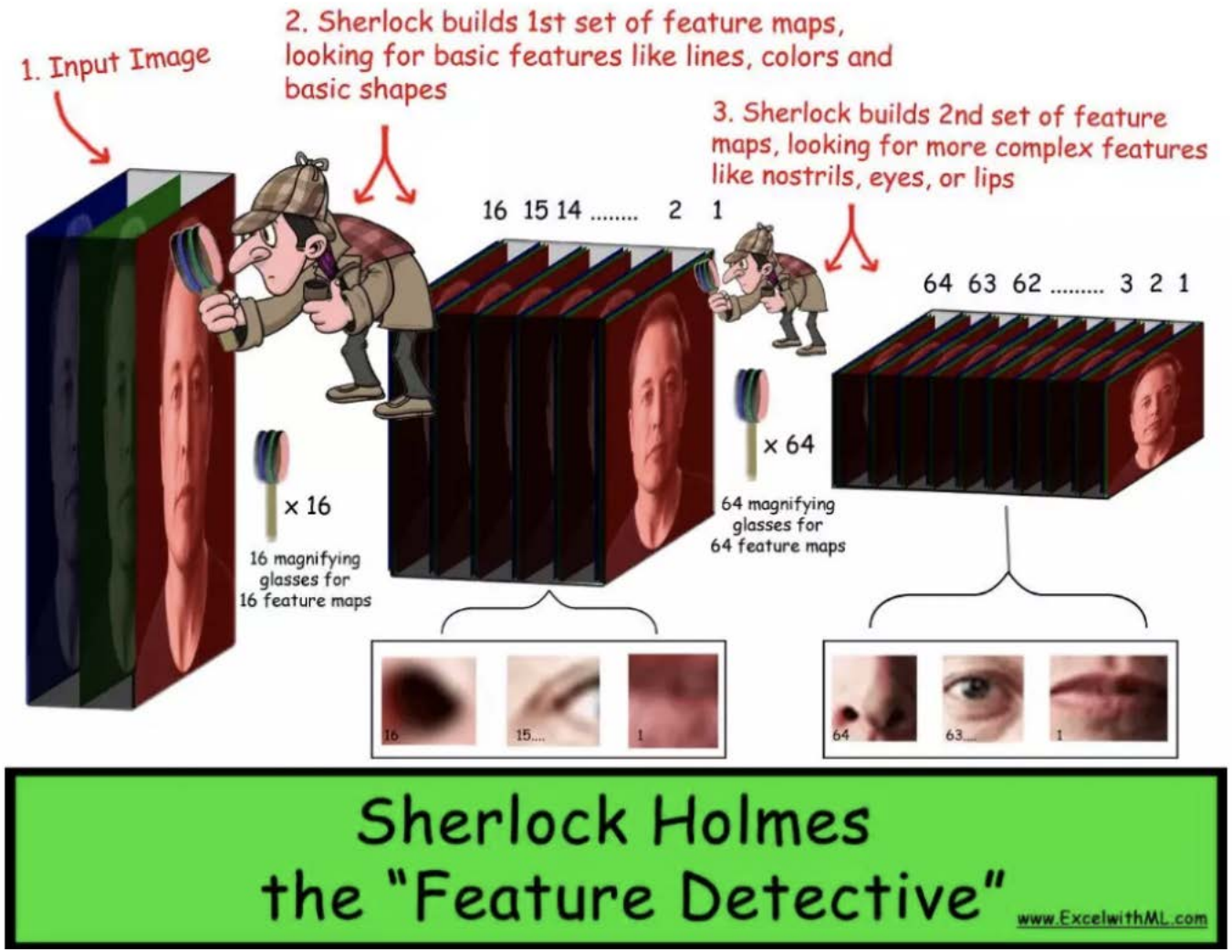
- 簡單細胞的感受野來自於一系列的背外側膝狀核(LGN)線性排列的神經元感受野的綜合，作為這類線性排列的神經元特定連接模式的結果。
- 複雜細胞有較大感受野，是由相連的一系列有特定朝向的簡單細胞所構建。
- **感受野**(Receptive Field)機制主要是指聽覺、視覺等神經系統中一些神經元的特性，即神經元只接受其所支配的刺激區域內的信號。

Simple cell and complex cell

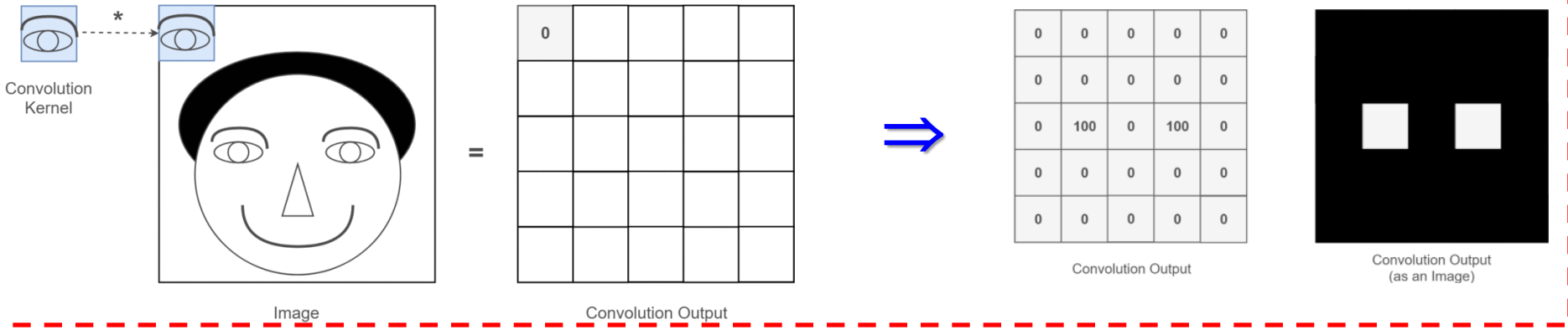
Unit	Pooling	Computation	Operation
Simple		Selectivity / template matching	Gaussian-tuning / AND-like
Complex		Invariance	Soft-max / or-like



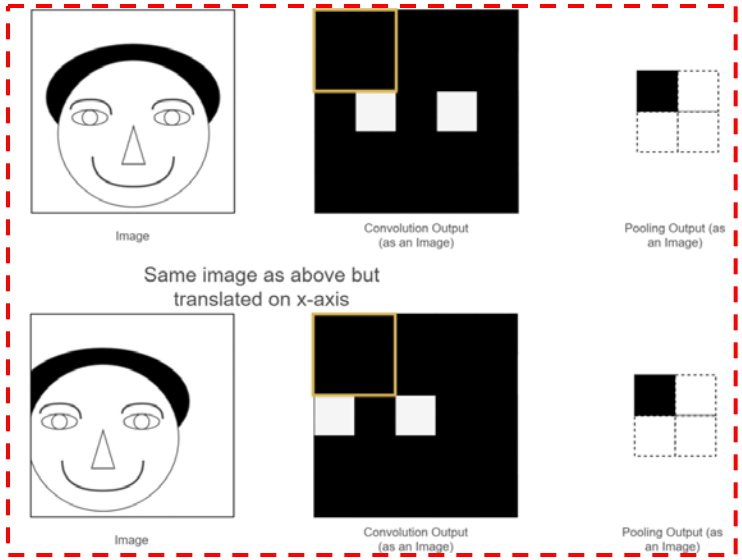
Feature Detective Holmes (福爾摩斯)



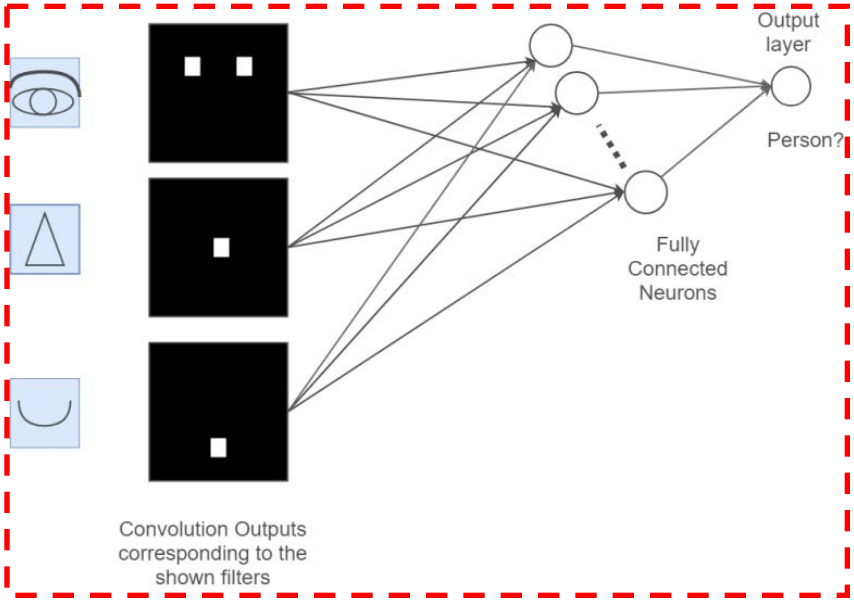
Simulation



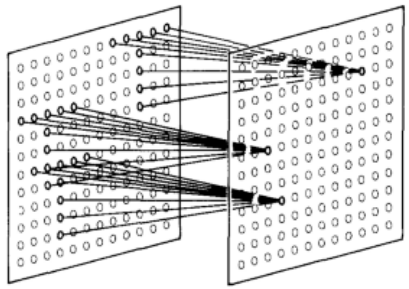
simple cell, convolution

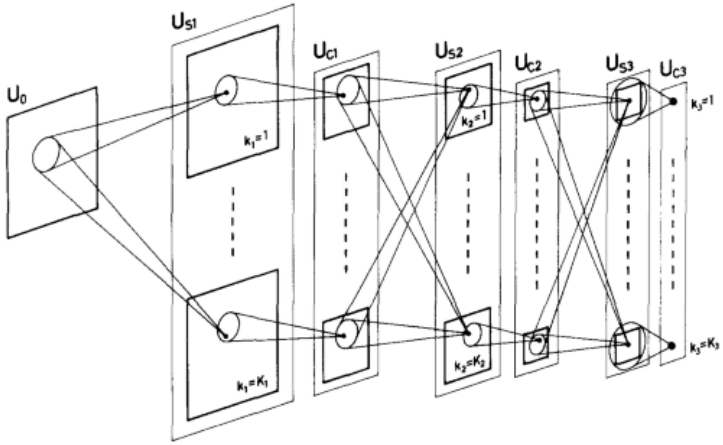


complex cell, Pooling (Translation invariant)

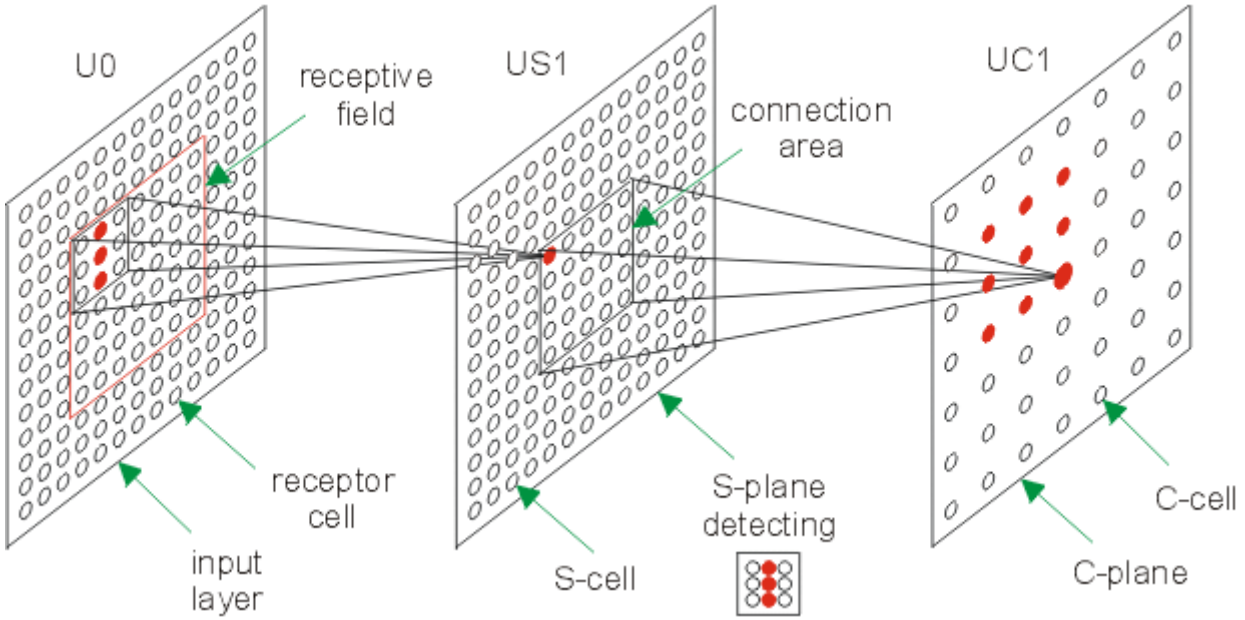


Neocognitron, Kunihiko Fukushima (福島邦彥, 1980)





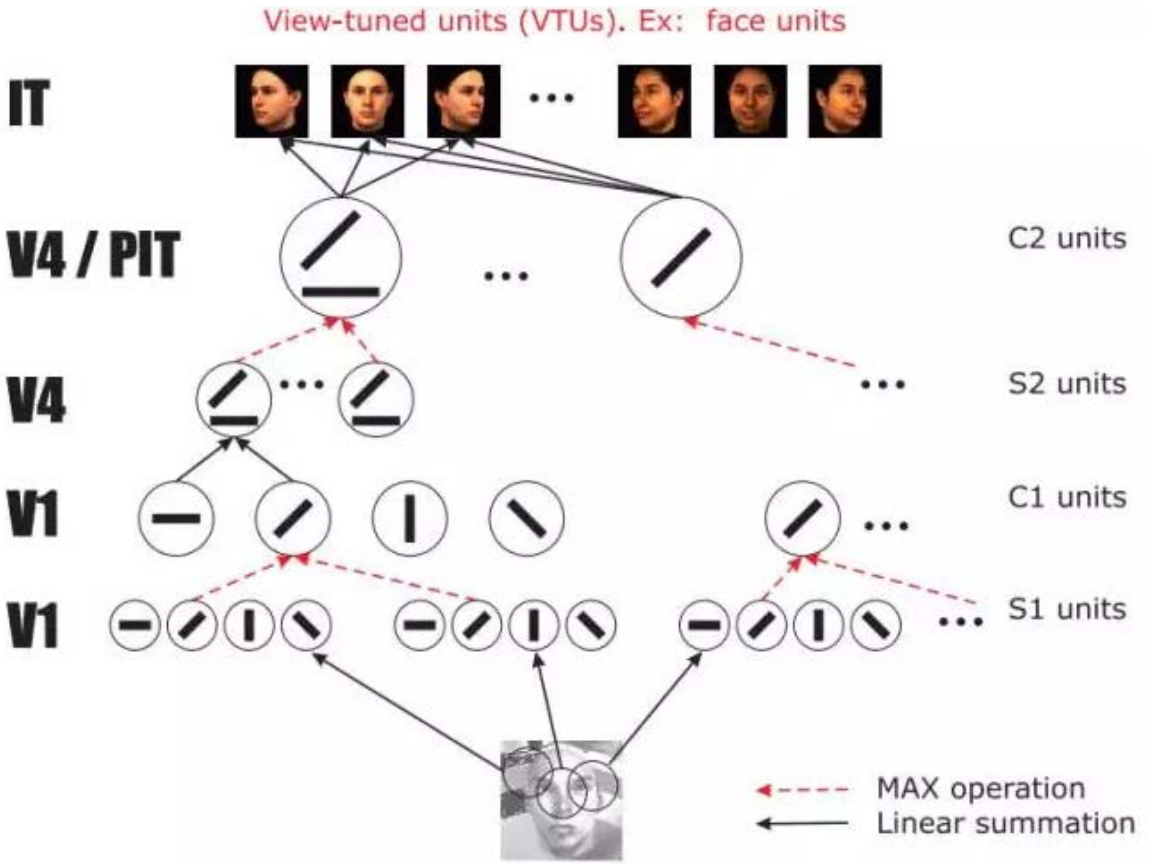
K. Fukushima



- 神經感知機(Neocognitron)主要是由兩種不同的cells分別稱作S-cells與C-cells交錯堆疊而成。
- **S-cell對應到大腦中的simple cells，其功用為feature extraction。**
- **C-cell對應到大腦中的complex cells，有較大的receptive field，具有downsampling的效果。**

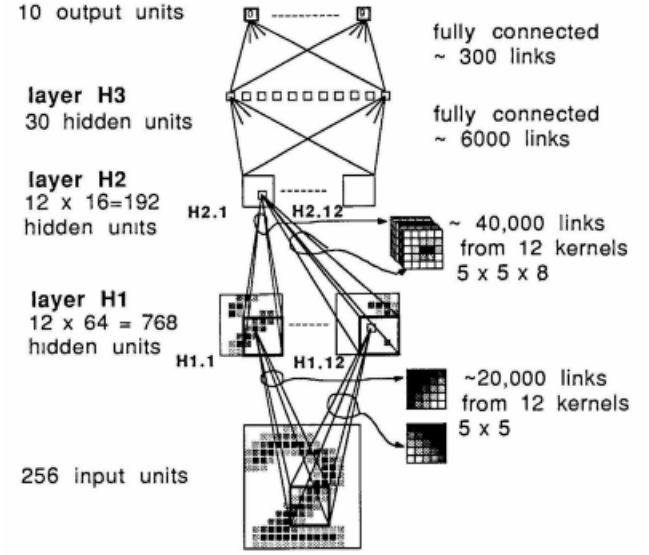
HMAX (Hierarchical Model And X), (Riesenhuber and Poggio, 1999)

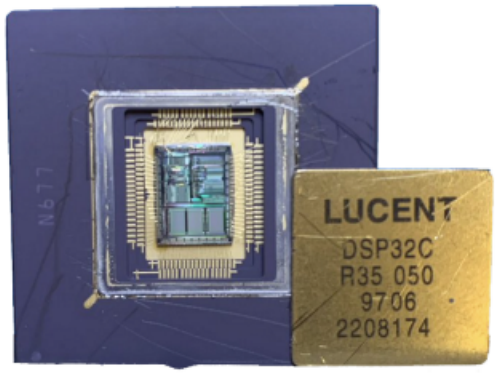
- 大腦皮層中對視覺資訊加工處理的操作主要有兩種
  - 一種是在簡單細胞中進行的線性操作
  - 一種是在複雜細胞中進行的非線性彙聚



- Poggio教授提出HMAX的腦視覺資訊處理流程，其中
  - 簡單單元「S1 unit」與「S2 unit」進行Gabor小波濾波的操作。
  - 複雜單元「C1 unit」與「C2 unit」進行Max Pooling的局部區域最大值的操作。
- 事實上除直接使用事先設定的Gabor濾波器，HMAX等價於一個四層的神經網路，已經初步具備現代深度模型的雛形。

LeNet-5, (LeCun, 1989)

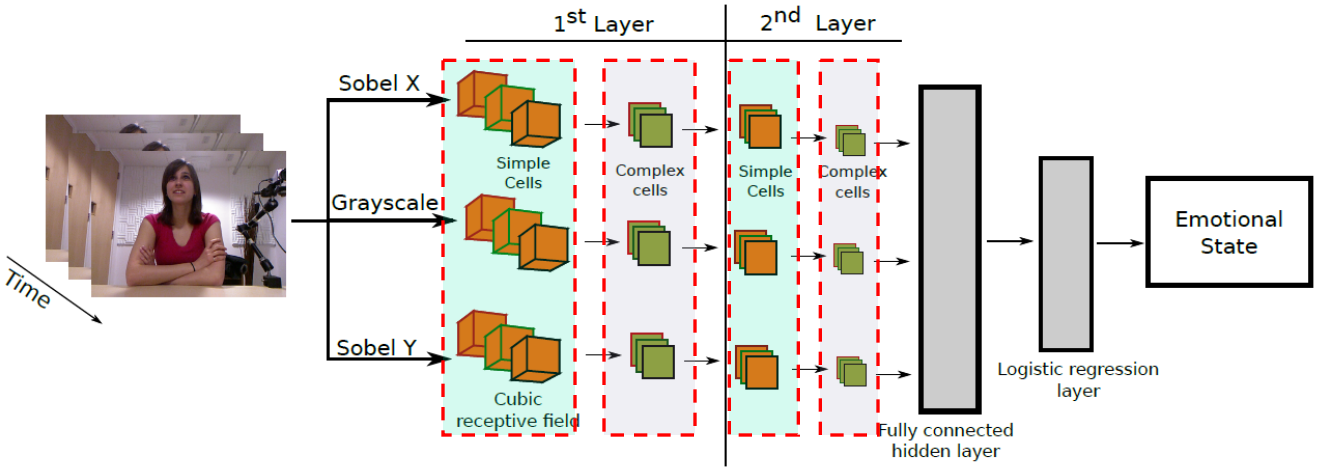




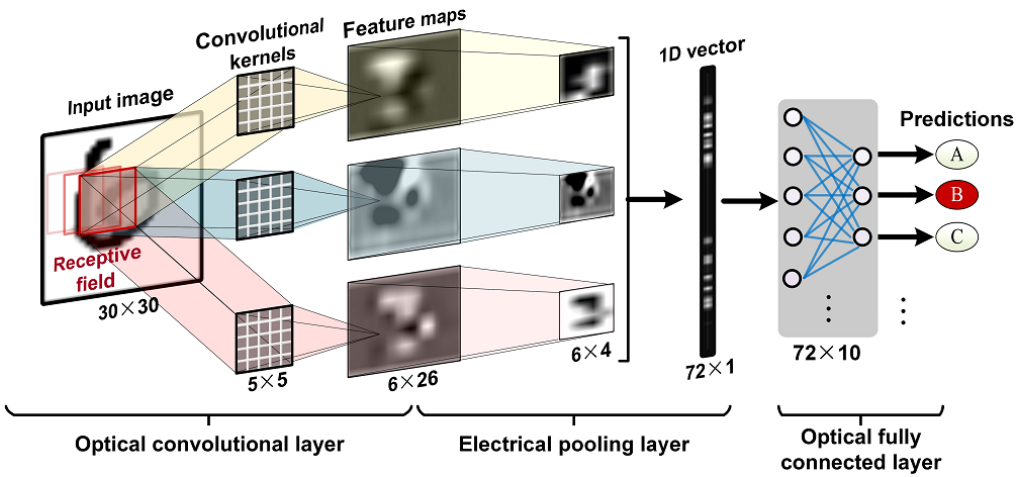
Y. LeCun

- 1989年貝爾實驗室Denker等人發表「手寫郵遞區號的神經網路辨識」方法，該方法運用卷積作特徵選取，不過此時卷積核(convolutional kernel)的參數都是人工設計，並非由神經網路自動學習獲得。
- 同年Y. LeCun 等人於貝爾實驗室亦發表針對「手寫郵遞區號數字辨識」的卷積神經網路，不同於DenKer等人的作法，這次模型卷積核參數的選取完全是自動學習。
- LeCun於**1998年提出LeNet-5**，其架構與現今的CNN已經別無二致，且在文字圖像辨識已經技壓群雄，但受限於當時的**計算能力**與**有限資料數據集合**，神經網路的優勢在當時並不十分顯著。

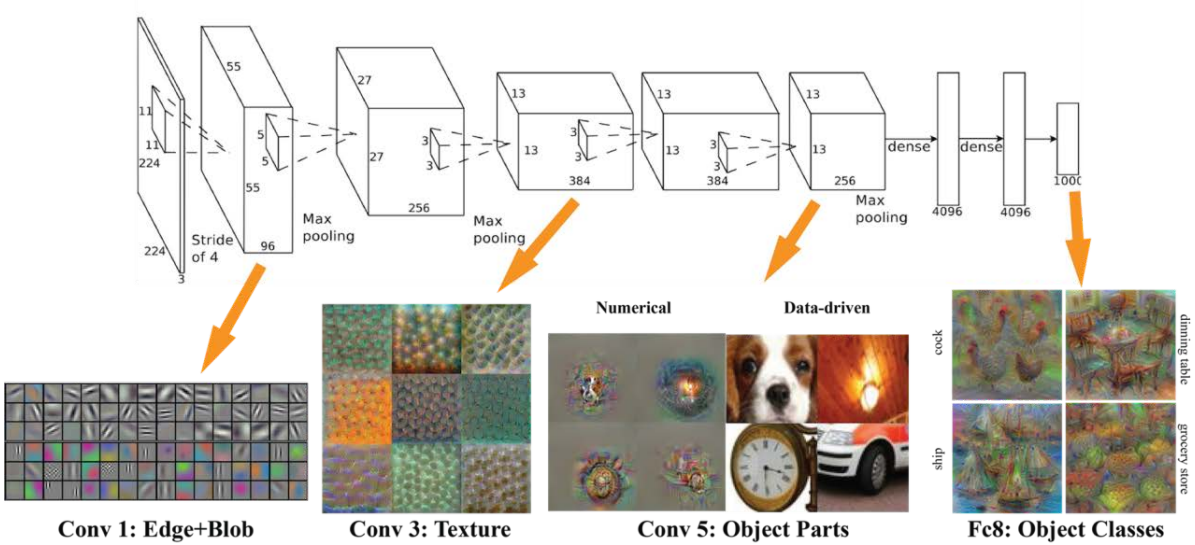
Prototype of Convolutional Neural Network



- Each layer of the CNN is composed of **simple cells** and **complex cells**, simulated using **convolution** and **pooling** operations.

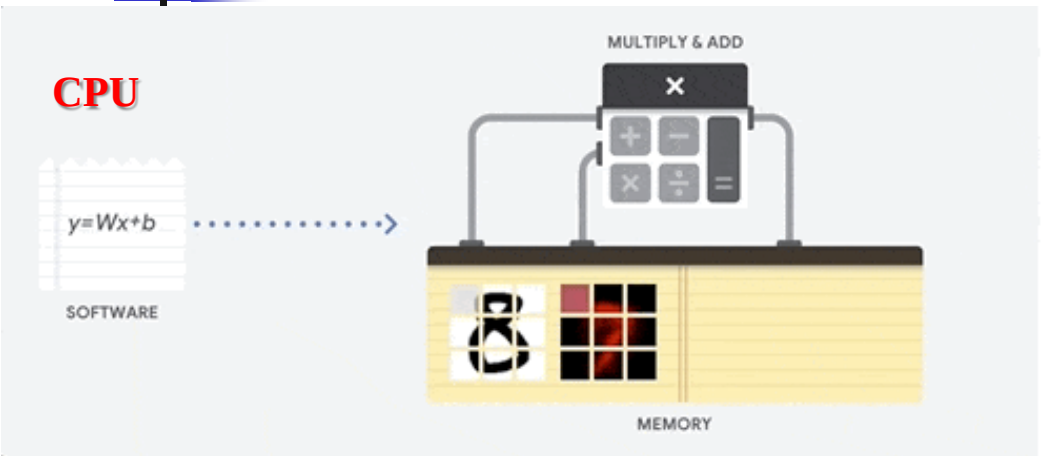


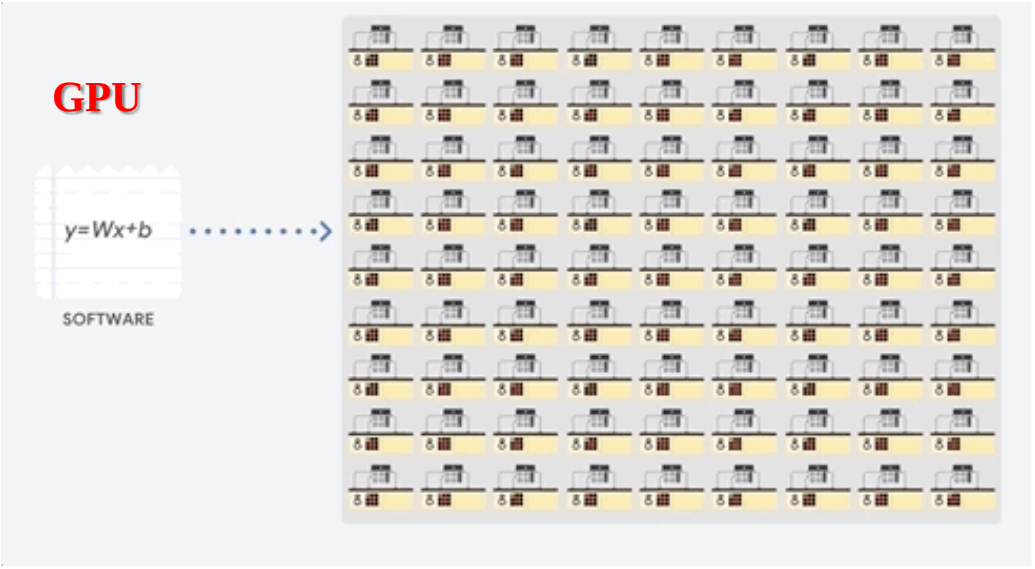
Convolutional Neural Network

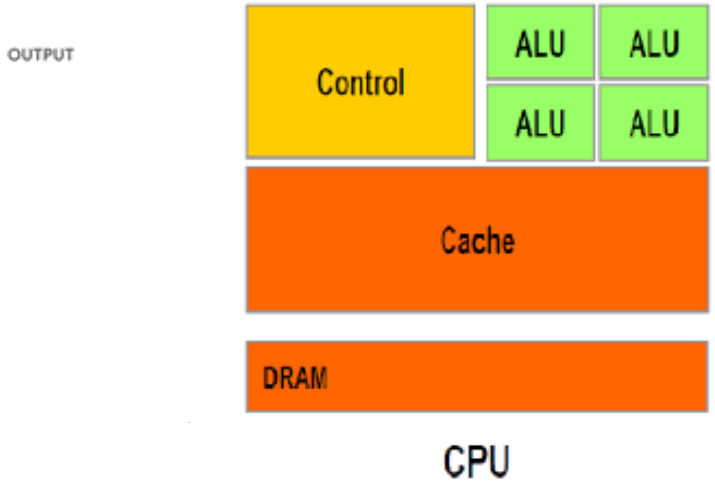


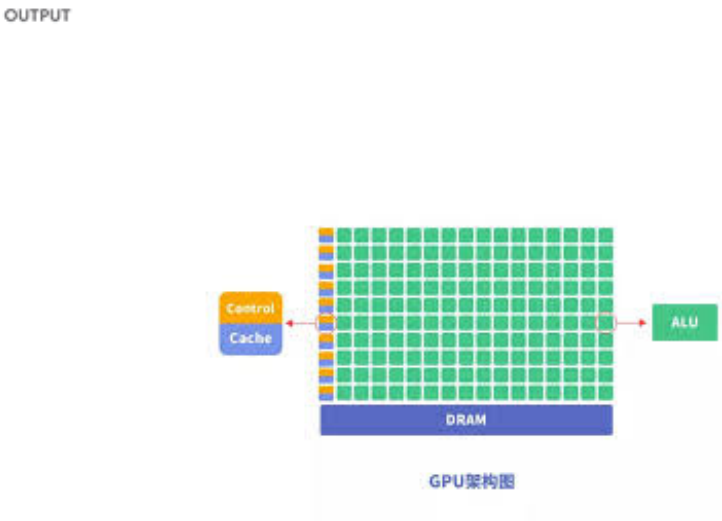
- 深度學習將特徵-特徵之間的轉換模式以層-層之間的轉換實現，而高維的特徵向量以層的形式呈現，所以越深的網路代表著經過多次的函數處理與萃取，所萃取資訊的抽象程度越高，也就越接近人類所想像。
- 這也代表越淺層的神經元是擷取不同的紋理，而深層的神經元會結合這些簡單的紋理特徵，產生更高複雜度的特徵，例如狗的臉，兔子的毛髮，貓的臉型等。

Graphics Processing Unit, GPU







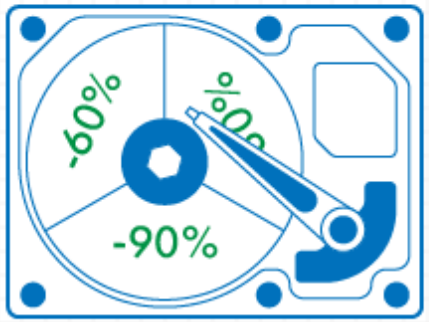


Big Data

大數據的興起

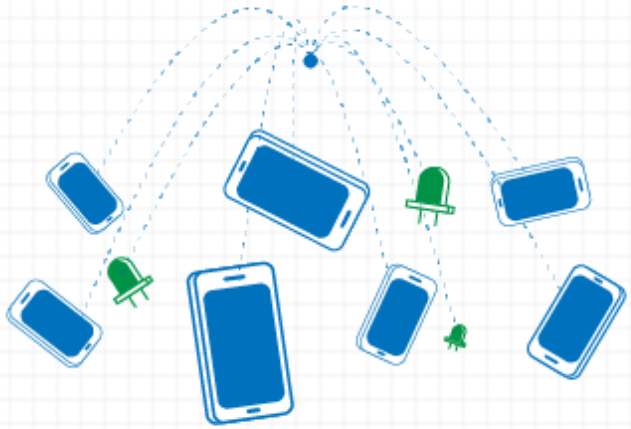
儲存成本下降

硬碟價格下跌



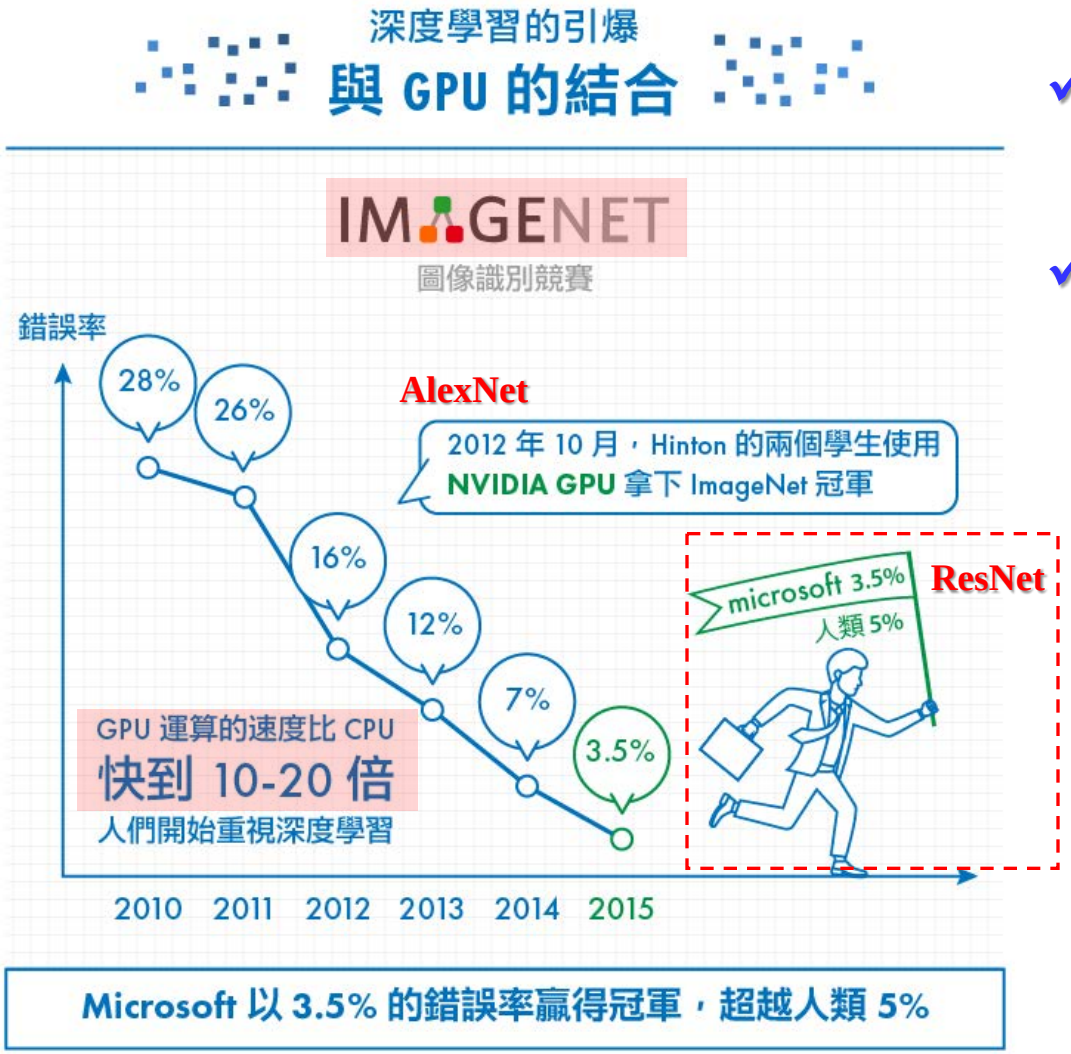
資料取得成本下降

感測器、智慧型手機



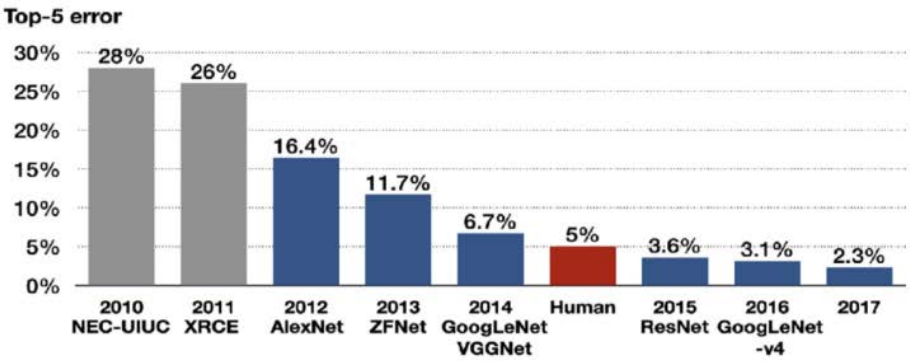
如何「儲存」、「分析」海量數據
並成功地「溝通」分析結果，成為新的瓶頸與研究重點

GPU + Big Data



- ✓ **Big Data (IoT, Cloud Computing)**
- ✓ **Hardware improvement (GPU)**

ImageNet, 2006

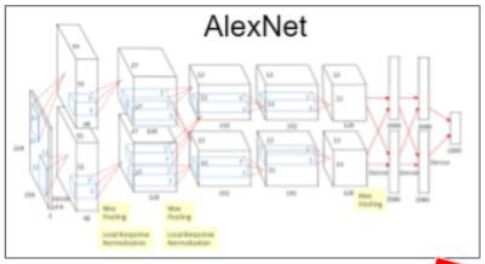


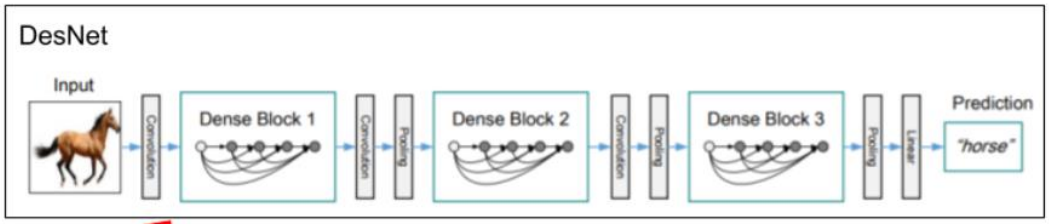
F.-F. Li

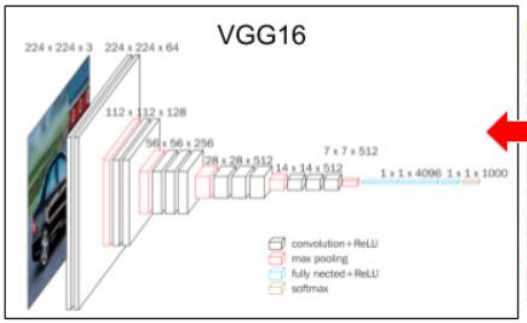
- **ImageNet** dataset是由史丹佛大學電腦科學教授李飛飛於2006年所開創，其目的是希望能擴大及增進訓練AI所能使用的資料，ILSVRC (ImageNet Large Scale Visual Recognition Challenge)競賽所用的資料庫即為ImageNet。

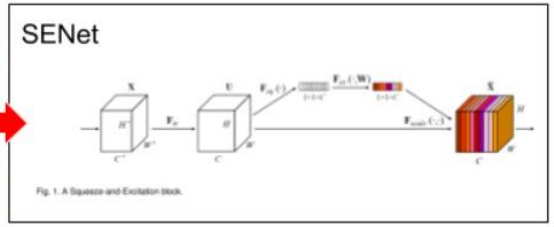
- **ImageNet**是大型視覺資料庫，用於視覺目標辨識軟體研究。ImageNet包含**2萬多個典型類別**，每一類包含**數百張圖像**。資料庫已手動注釋1400多萬張圖像，以指出圖片中的物件，並在至少100萬張圖像中提供邊框。
- ILSVRC競賽是從ImageNet dataset取出一部份作為比賽用資料，訓練資料共包含1000個類別，每類別約1000張照片，總計訓練資料1200萬張照片，其中validation set與testing set照片數量各為5萬及10萬張。

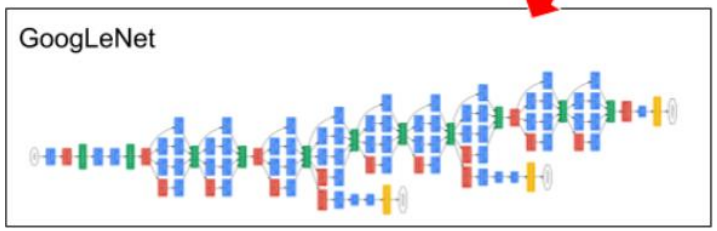
ImageNet for ILSVRC

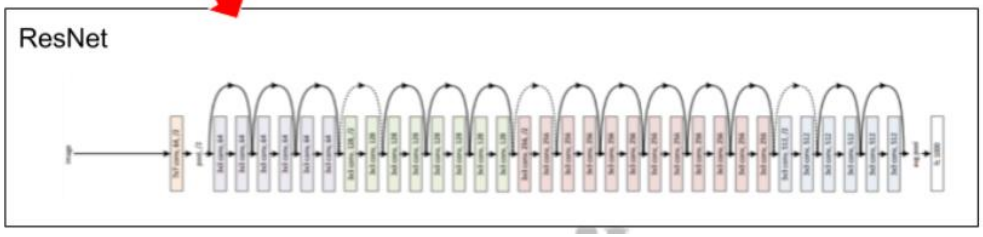












KITTI資料集

- KITTI資料集由德國卡爾斯魯厄理工學院與豐田美國技術研究院聯合創辦，是目前國際上最大的自動駕駛場景下的計算機視覺演算法評測資料集。該資料集用於評測立體影像(stereo)，光流(optical flow)，視覺測距(visual odometry)，3D物體檢測(object detection)和3D跟蹤(tracking)等計算機視覺技術在車載環境下的效能。KITTI包含市區、鄉村與高速公路等場景採集的真實影像資料，每張影像中最多達15輛車與30個行人，還有各種程度的遮擋與截斷。整個資料集由389對立體影像與光流圖，39.2 km視覺測距序列以及超過200k 3D標註物體的影像組成，以10Hz的頻率取樣及同步。

- KITTI資料集的平臺裝配有2個灰度攝像機，2個彩色攝像機，一個Velodyne 64線3D鐳射雷達，4個光學鏡頭，以及1個GPS導航系統。感測器引數如下
  - 2 × PointGray Flea2 **grayscale cameras** (FL2-14S3M-C), 1.4 Megapixels, 1/2" Sony ICX267 CCD, global shutter
  - 2 × PointGray Flea2 **color camera**s (FL2-14S3C-C), 1.4 Megapixels, 1/2" Sony ICX267 CCD, global shutter
  - 4 × Edmund Optics lenses, 4mm, opening angle ~ 90◦, vertical opening angle of region of interest (ROI) ~ 35◦
  - 1 × Velodyne HDL-64E rotating 3D laser scanner, 10 Hz, 64 beams, 0.09 degree angular resolution, 2 cm distance accuracy, collecting ~ 1.3 million points/second, field of view: 360◦ horizontal, 26.8◦ vertical, range: 120 m
  - 1 × OXTS RT3003 inertial and GPS navigation system, 6 axis, 100 Hz, L1/L2 RTK, resolution: 0.02m / 0.1◦

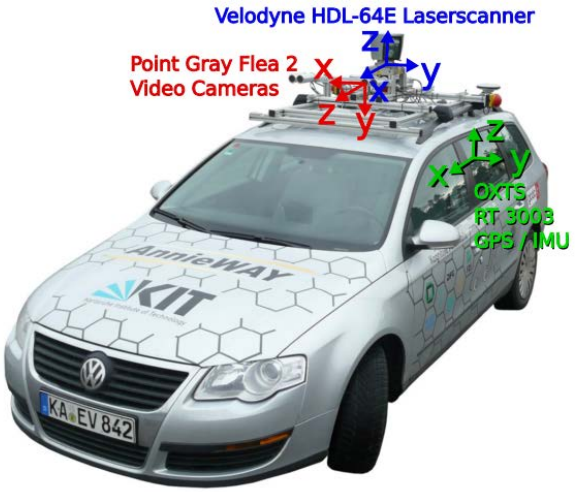
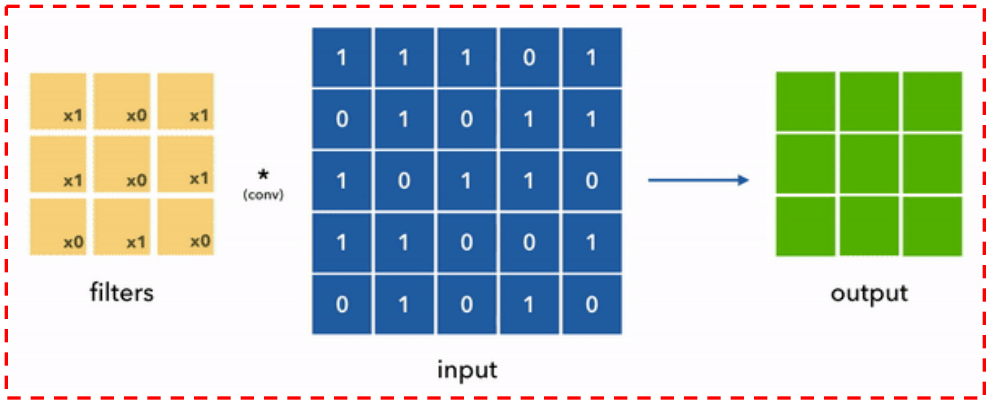


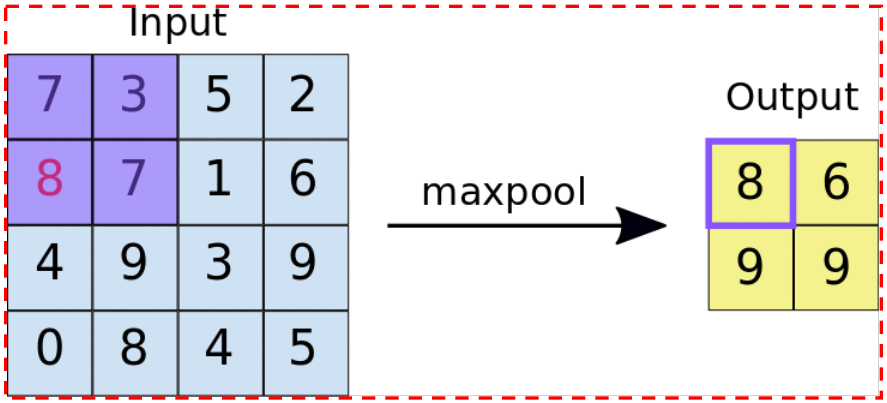
Fig. 1. **Recording Platform.** Our VW Passat station wagon is equipped with four video cameras (two color and two grayscale cameras), a rotating 3D laser scanner and a combined GPS/IMU inertial navigation system.

CNN architecture

- Three main properties of the CNN architecture
  - Local Receptive Field (sparse connectivity) (稀疏連結)
  - Weight Sharing (parameter sharing) (權重共用) (權重共用意味著卷積核在檢測影像不同位置的同一個特徵)
  - Spatial Sub-sampling (pooling) (下採樣)



Convolution



Max pooling

- 全連接的神經網路中，每個神經元的值取決於網路的整個輸入，而卷積神經網路中特徵圖(feature map)的神經元，僅取決於輸入的一個區域，這個區域就是該神經元的感受野。

(1) 稀疏連結

- 稀疏連結的靈感源自動物的視覺皮層結構，意指視覺的神經元在感知外界物體的過程中，只有部份神經元(稀疏連結)會起作用。
  - 例如CNN對於單通道的8x8輸入影像以3x3的卷積核進行卷積，卷積核於輸入影像選取3x3區域作卷積運算，然後以滑動視窗的方式進行整體影像的卷積運算。
- 相較於神經網路的全連結方式，稀疏連結極大程度的減少參數的數量，同時也避免模型的過擬合。

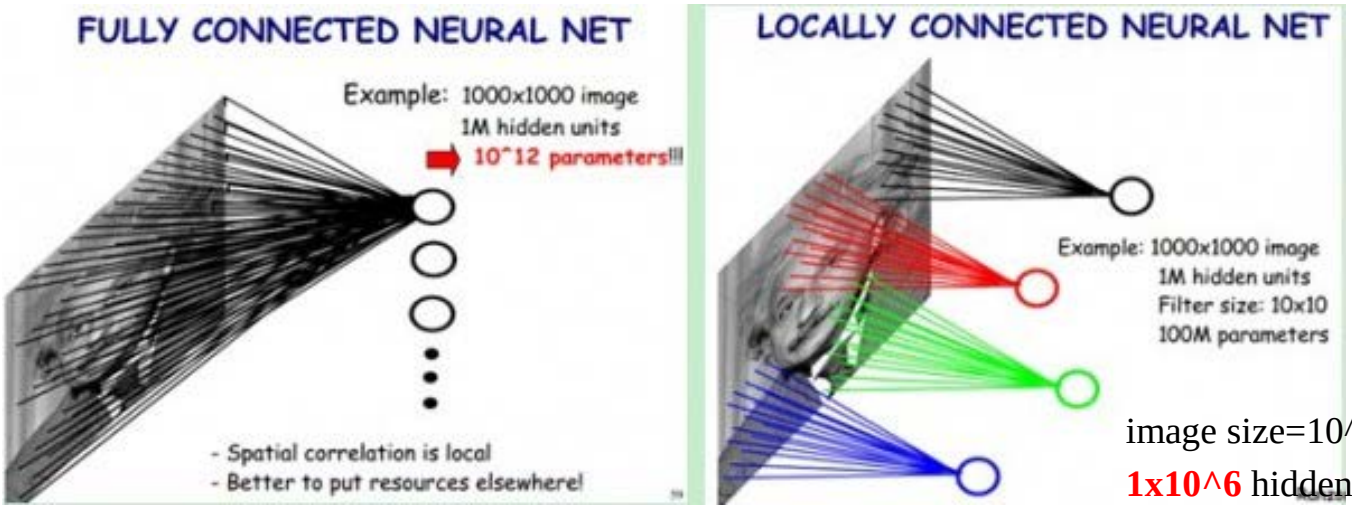
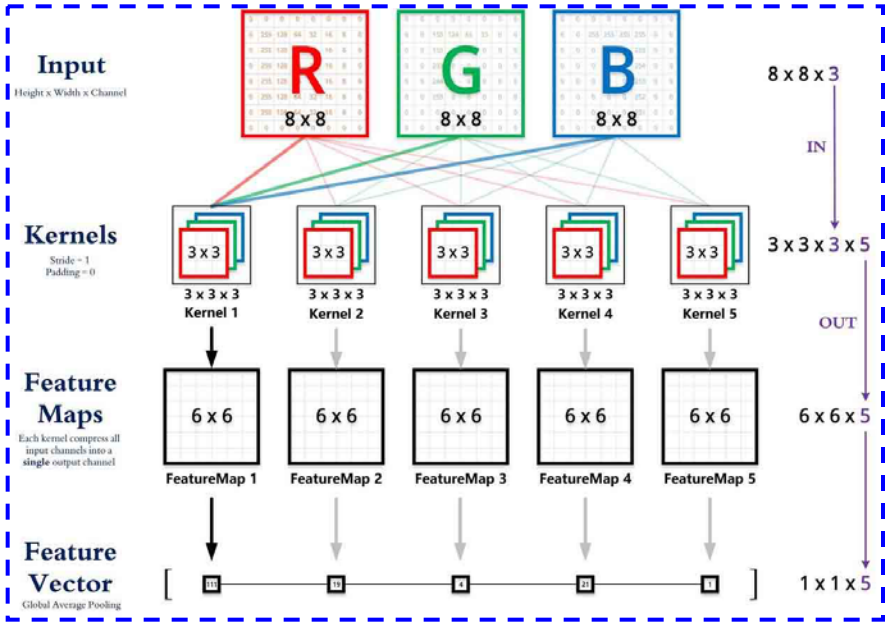


image size=10^3x10^3
1x10^6 hidden units
# of paras=10^3x10^3 x **1x10^6**=10^12

image size=10^3x10^3
1x10^6 hidden units
filter size=**10x10**
# of paras=**1x10^6** x **10x10**=10^8
(**此時不考慮權重共用**)

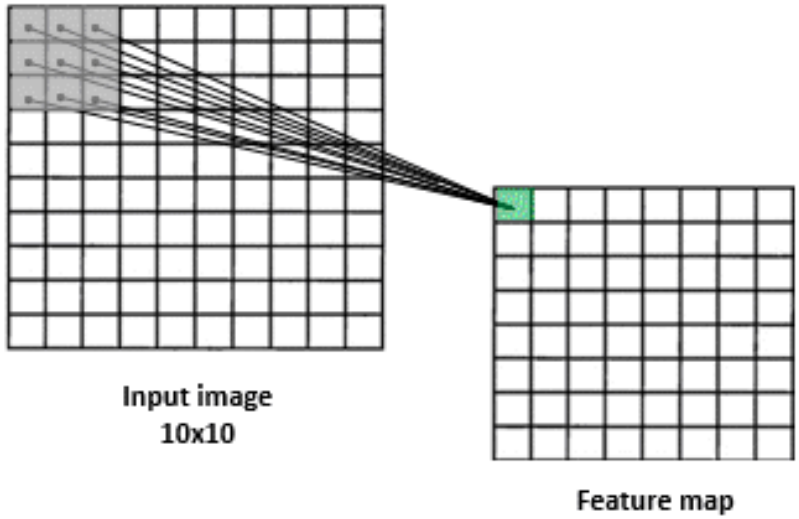
(2) 權重共用

- 輸入大小為8x8x3的彩色影像，並預設卷積核的個數為5(步長為1，填充為0)。
- 卷積操作得到三個6x6的特徵圖(因為RGB)，然後再將三個6x6的特徵圖(依照Element-wise)相加進行通道融合，最終得到5個(因為5個卷積核) 6x6的特徵圖。

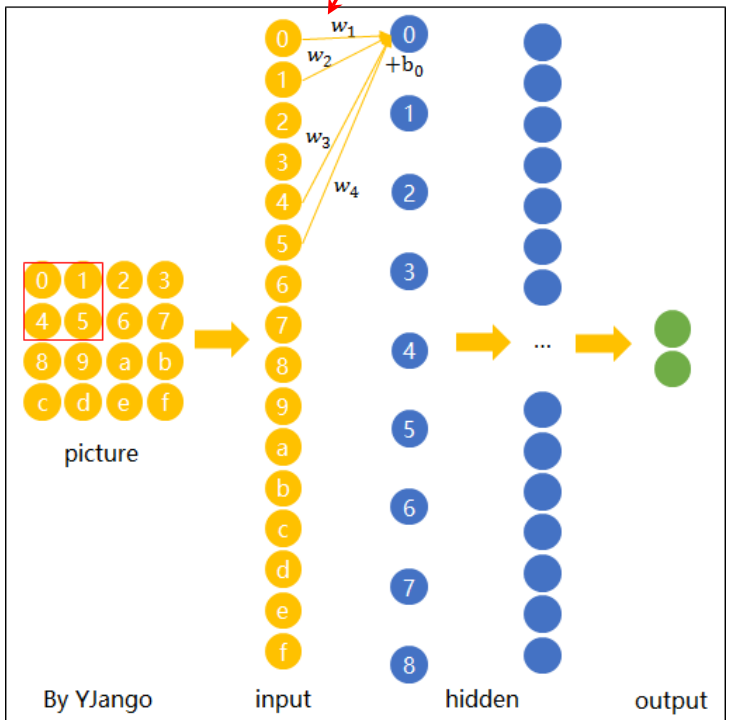


- 當第一個卷積核(3x3)與輸入影像的第一個通道(8x8)進行卷積操作
  - 按照全連結神經網路的架構，權重矩陣是9x64 (卷積核是3x3所有9個神經元)，權重參數個數是9x64=8x8x3x3(不考慮bias)=**576**
  - 但如果**權重共用**，權重矩陣是3x3，權重參數個數是**9**
- 綜合而言(考慮五個卷積核)
  - 全連結神經網路的權重參數個數是8x8x3x3x3x5=**8640**
  - 使用**權重共用**，權重個數為3x3x3(三個通道)x5(五個卷積核)=**135**

稀疏連結及權重共用

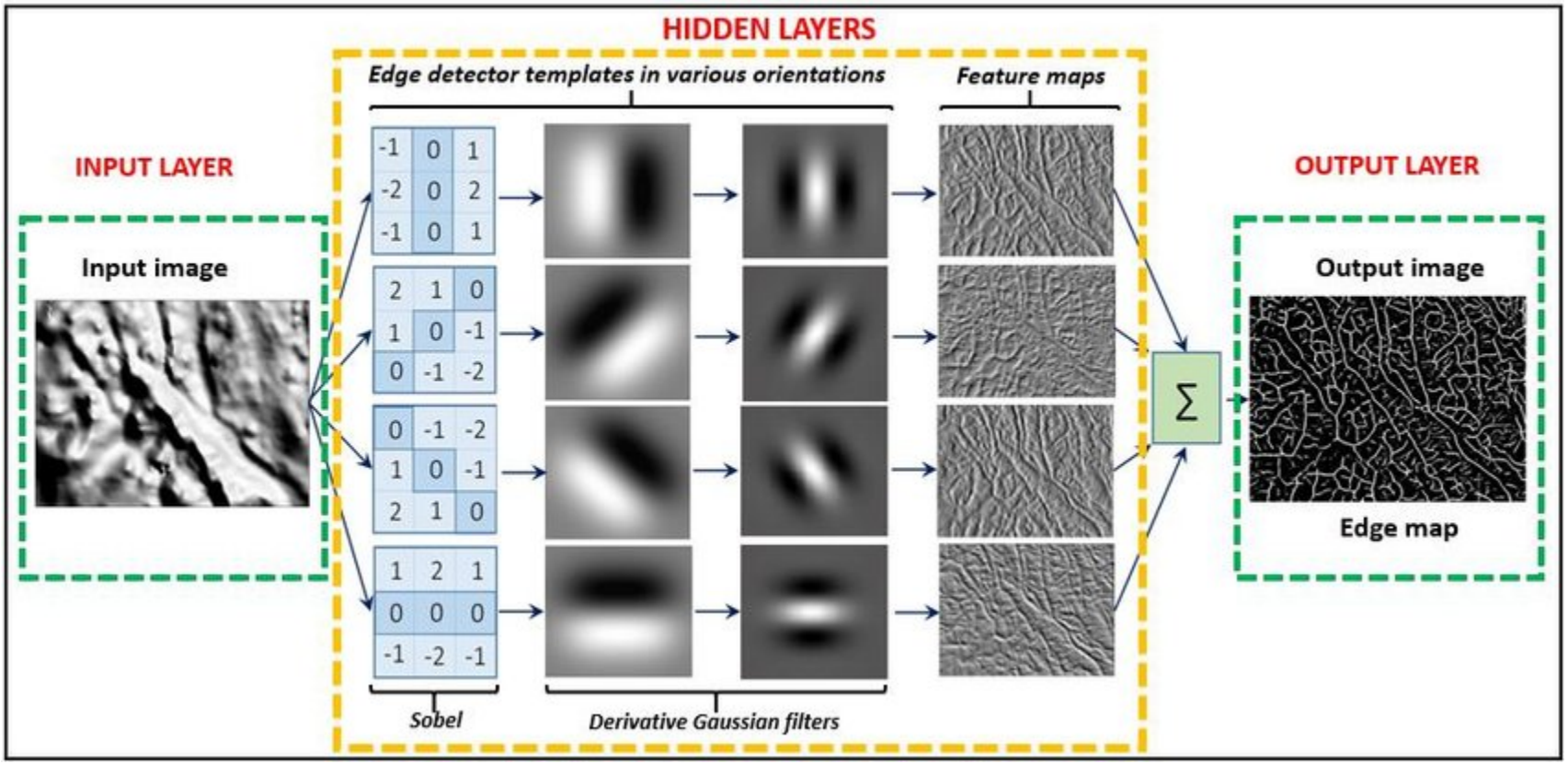


稀疏連結及權重共用

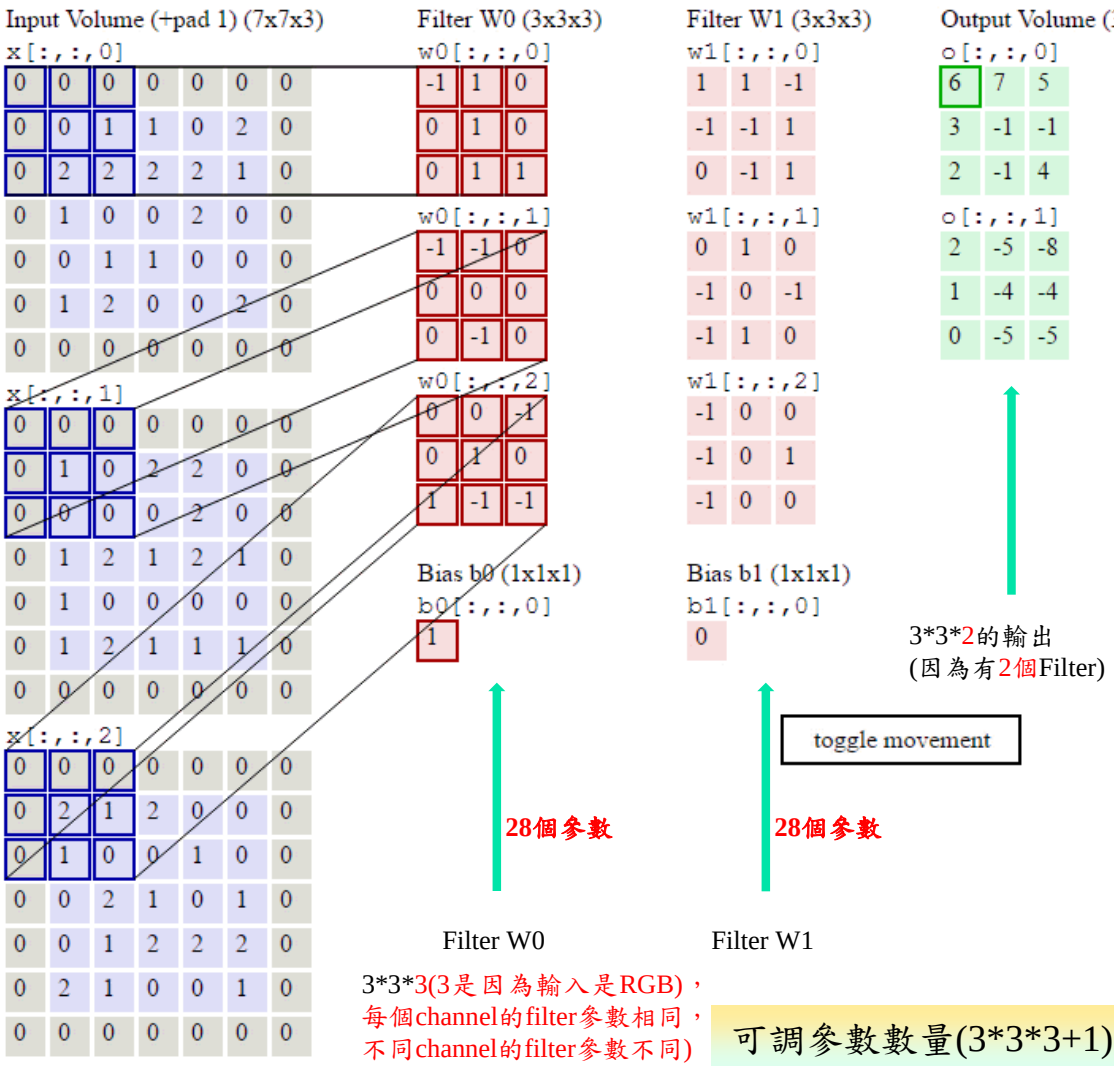


2D Conv ⇒ 1D Conv

Convolution



Multi-Channel Convolution



- 7*7*3輸入(原圖5*5, zero padding為7*7)
- 經過兩個3*3*3filter的卷積(步幅為2)，得到3*3*2的輸出

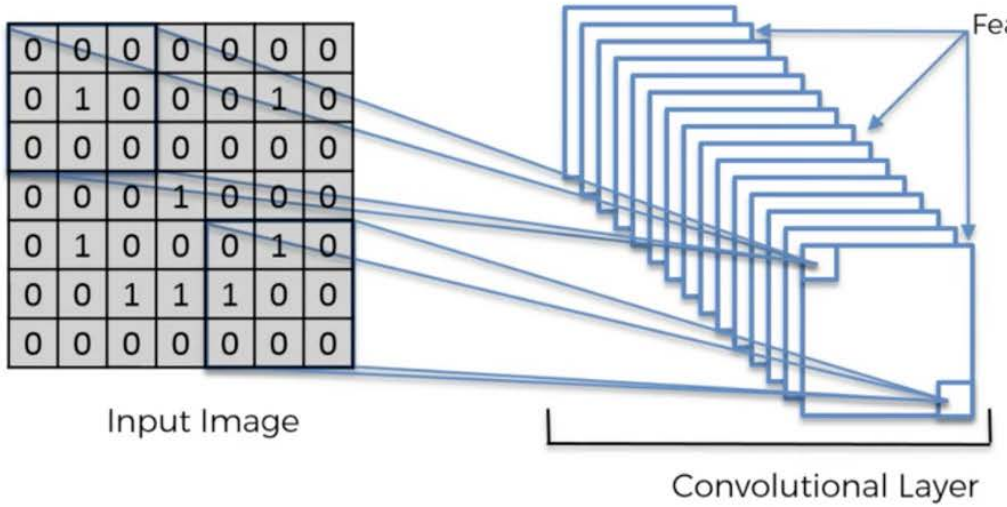
- Feature Map是經卷積變換提取的圖像特徵
- 兩個Filter就對原始圖像提取出兩組不同的特徵(即兩個Feature Map)
- 每個filter與原始圖像(或前一層)進行卷積後，都可以得到一個Feature Map。所以卷積後Feature Map的個數(或深度)與卷積層的filter個數是相同

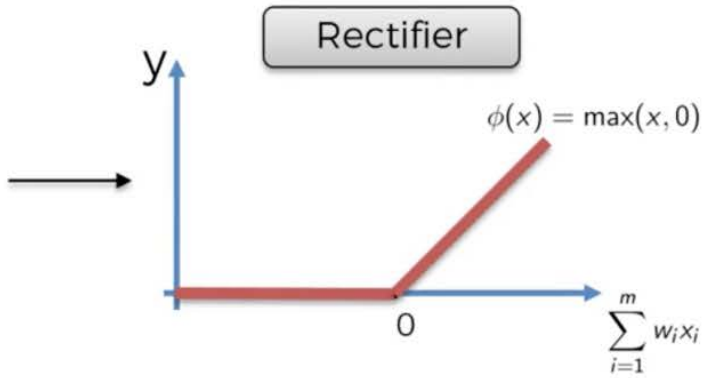
- n=5, f=3, s=2, p=1
- (input(n)=5x5, filter(f)=3x3, Stride(s)=2, zero_padding(p)=1)
- Feature map 大小 =(n-f+2*p)/s +1 =(5-3+2*1)/2 + 1 = 3

$$a_{ij} = f\left(\sum_{d=0}^{D-1}\sum_{m=0}^{F-1}\sum_{n=0}^{F-1} w_{d,m,n}x_{d,i+m,j+n} + w_b\right),$$

本例D=deep=3,F=3

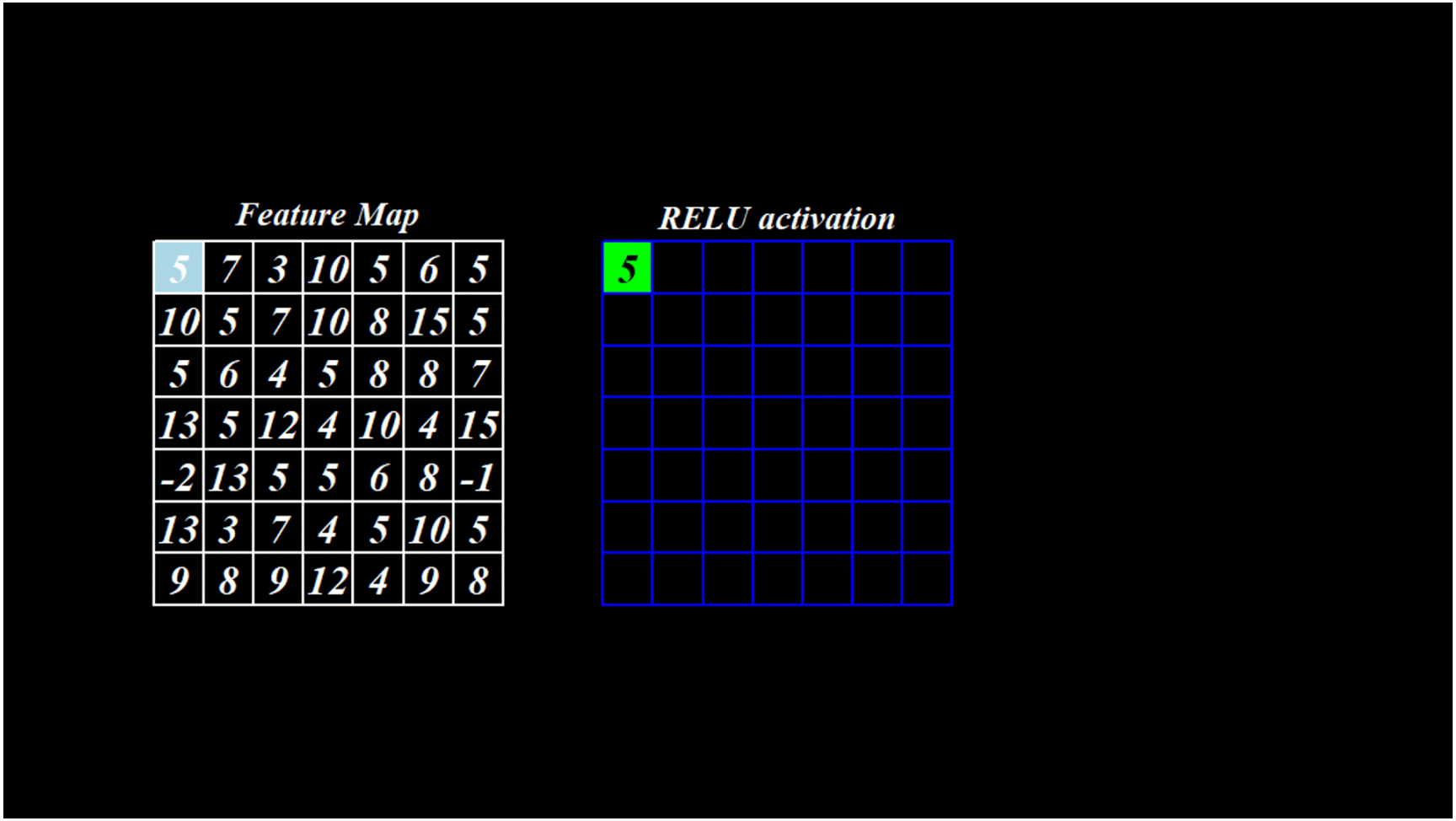
Convolution + ReLu





ReLu

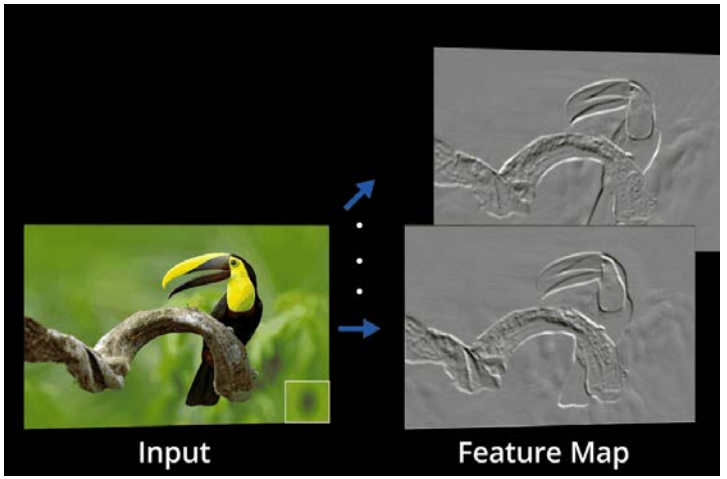
ReLu + Pooling



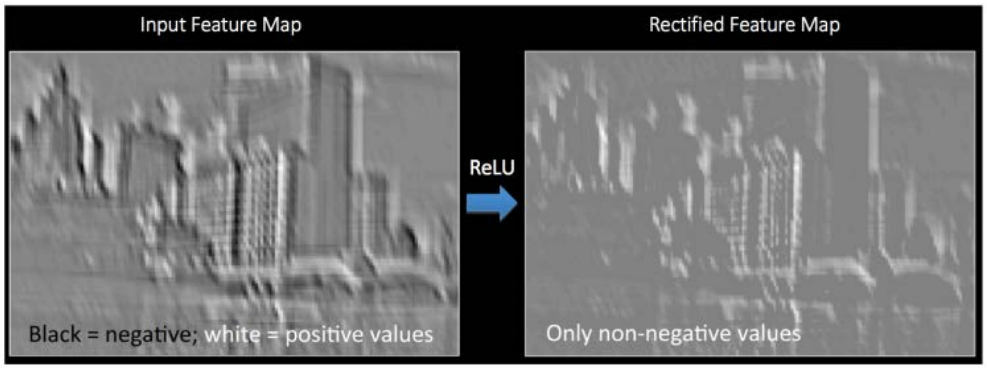
Convolution + ReLu + Pooling

Convolution

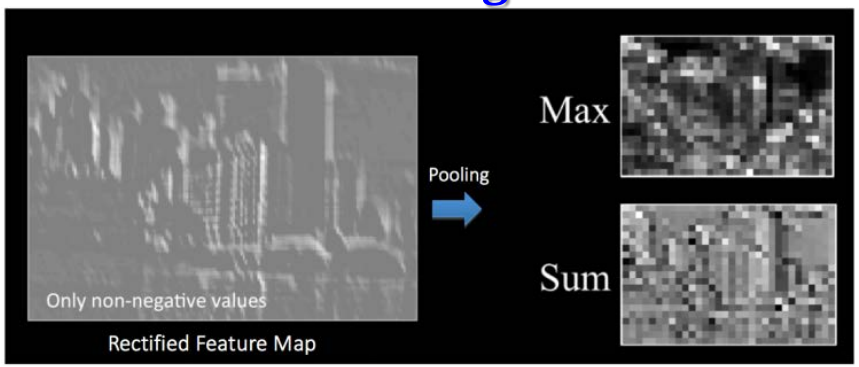
(RGB) Convolution



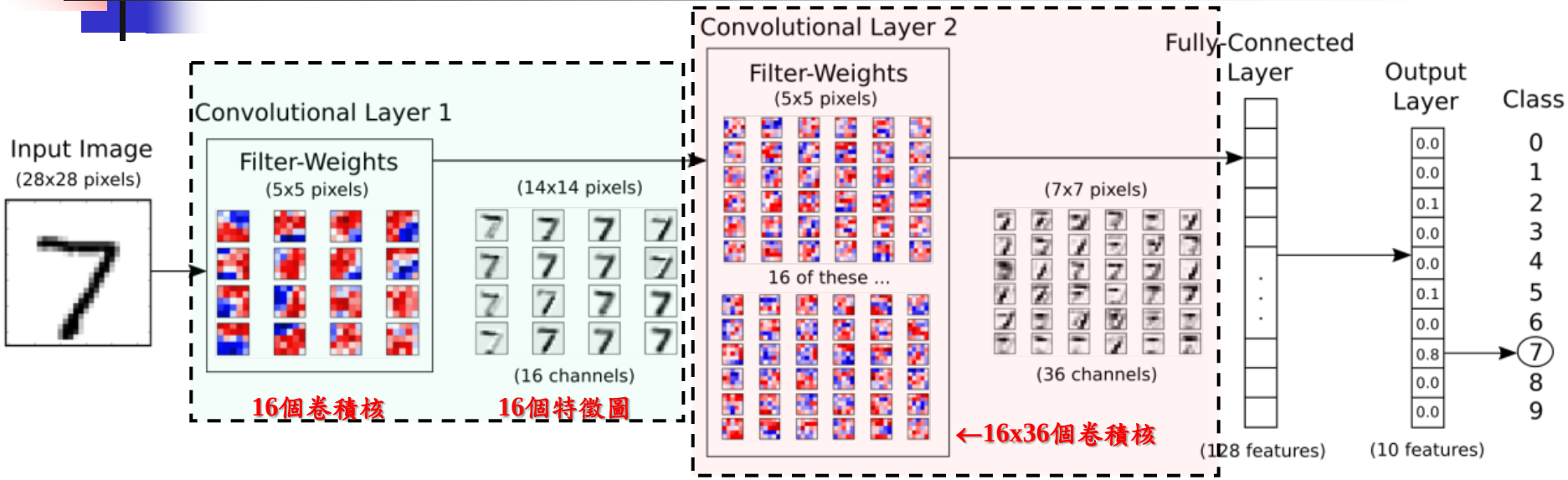
ReLu



Pooling



Simulation



The input image is processed in the first convolutional layer using the filter-weights. This results in 16 new images, one for each filter in the convolutional layer. The images are also down-sampled so the image resolution is decreased from 28x28 to 14x14. These 16 smaller images are then processed in the second convolutional layer.

輸入影像(28x28)經由16個5x5的卷積核操作，產生16個14x14的特徵圖。

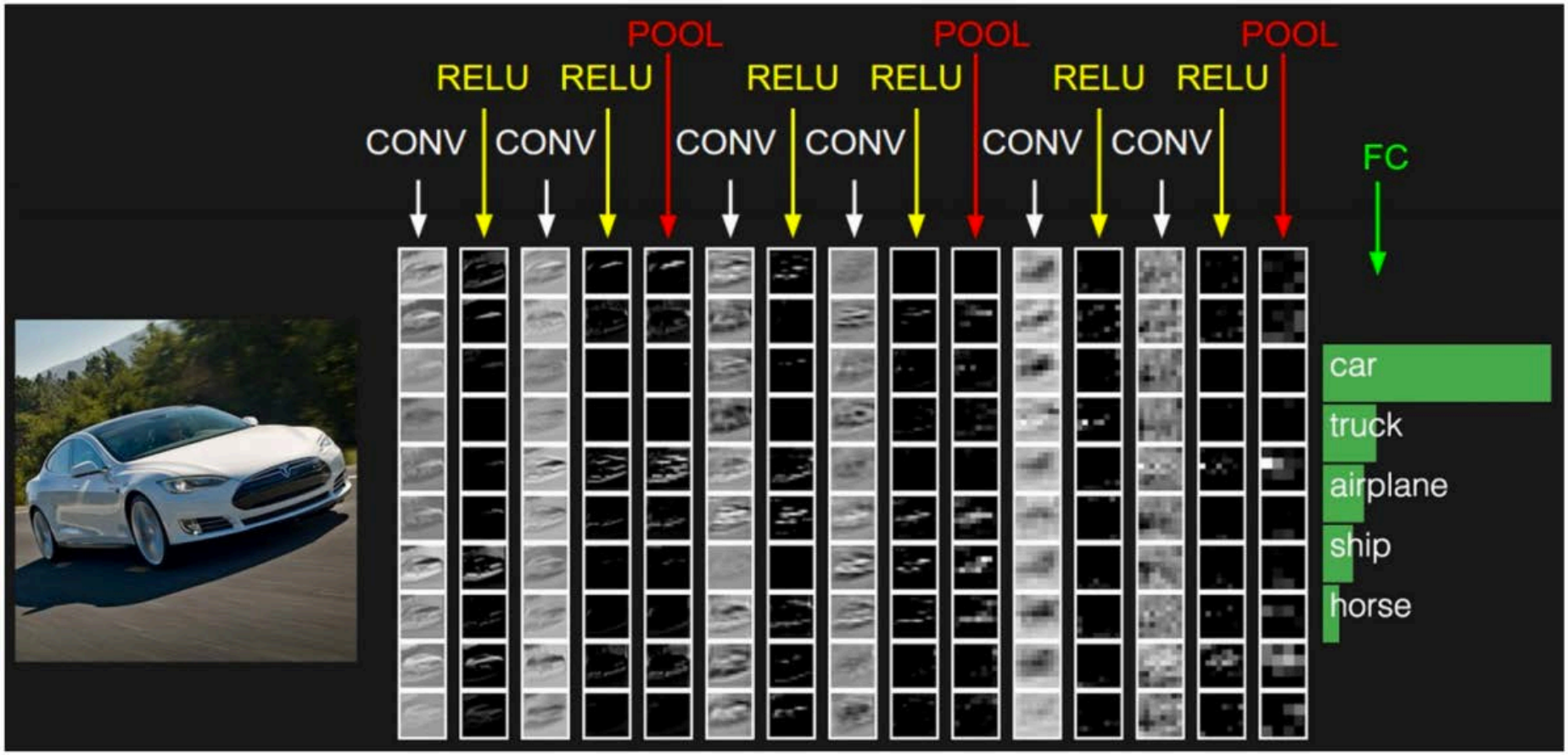
These 16 smaller images from (the first convolutional layer) are then processed in the second convolutional layer. We need filter-weights for each of these 16 channels, and we need filter-weights for each output channel of this layer. There are 36 output channels so there are a total of 16x36=576 filters in the second convolutional layer. The resulting images are down-sampled again to 7x7 pixels.

將前一層的16個特徵圖，經由36組卷積操作(每組有16個卷積核)，產生36個7x7的特徵圖。

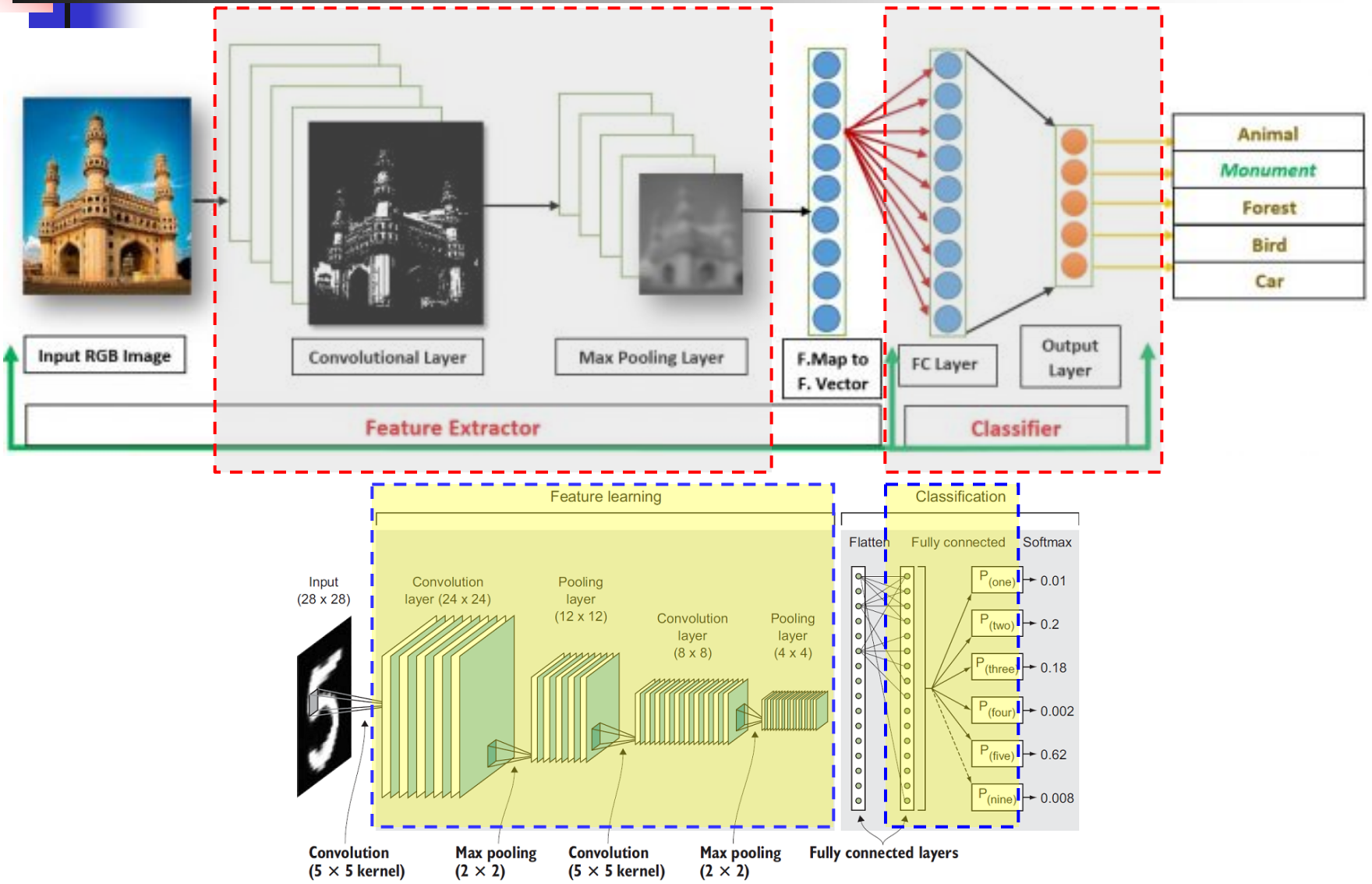
The output of the second convolutional layer is 36 images of 7x7 pixels each. These are then flattened to a single vector of length 7x7x36=1764, which is used as the input to a fully-connected layer with 128 neurons (or elements). This feeds into another fully-connected layer with 10 neurons, one for each of the classes, which is used to determine the class of the image, that is, which number is depicted in the image.

1764-128-10

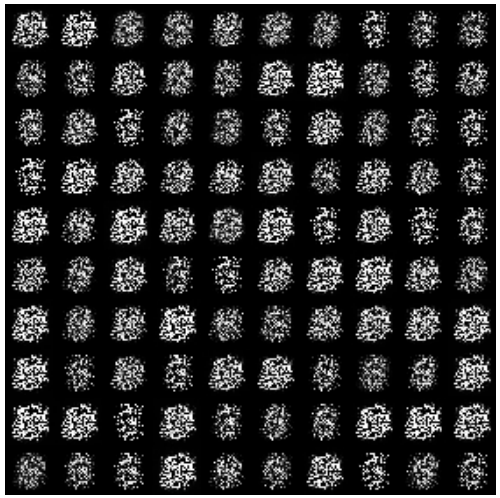
Convolutional NN

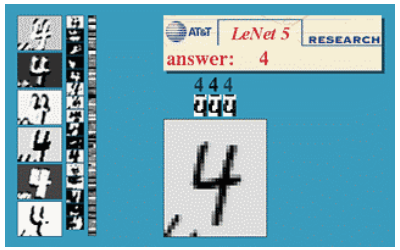


Convolutional NN

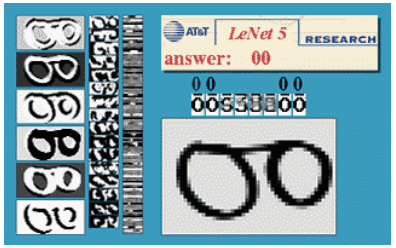


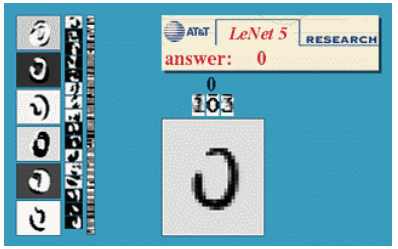
Simulation




LeNet 5
RESEARCH
answer: 4

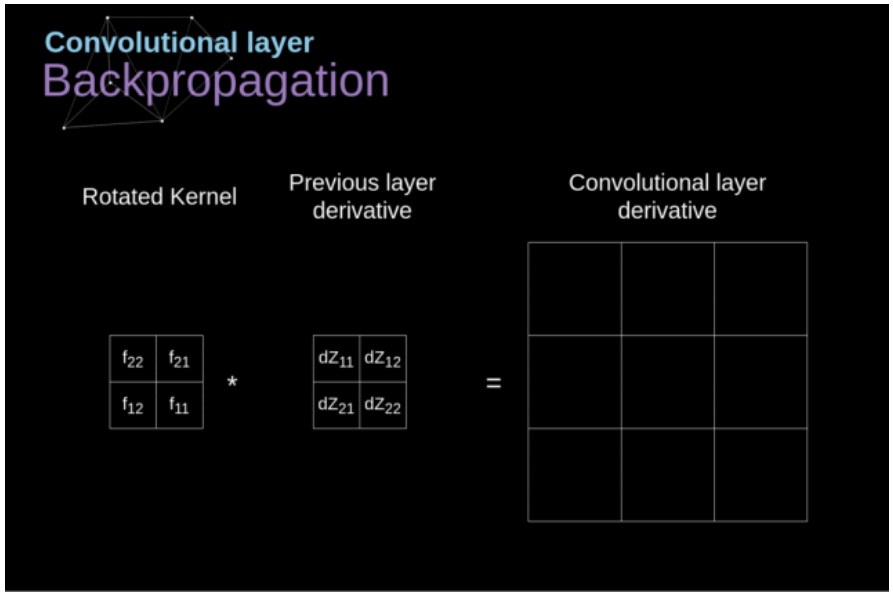

LeNet 5
RESEARCH
answer: 384

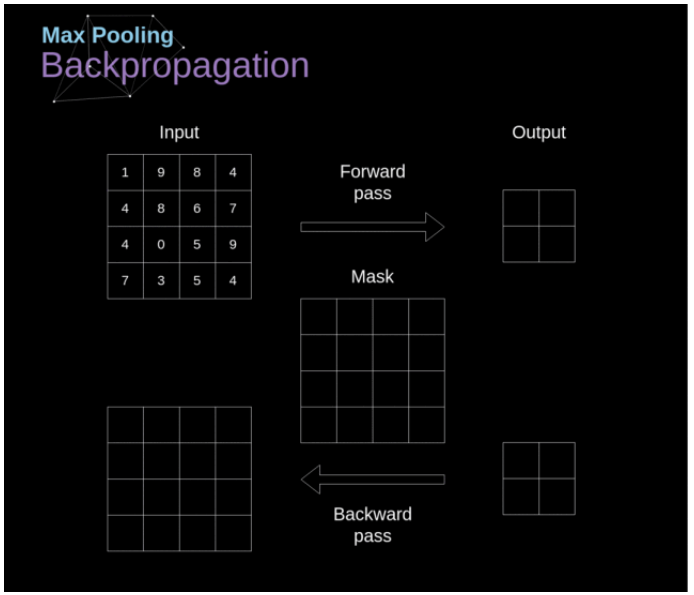

LeNet 5
RESEARCH
answer: 00


LeNet 5
RESEARCH
answer: 0

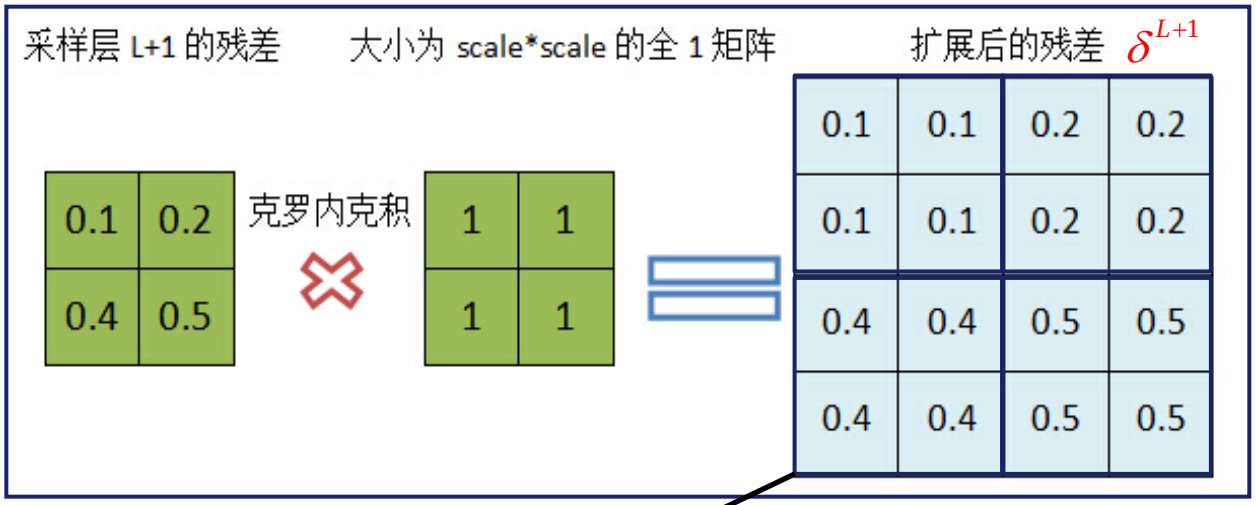
Back propagation of CNN

- CNN的反向傳播包括兩部分
  - 全連接層的反向傳播
  - 卷積層的反向傳播
  - 池化層反向傳播



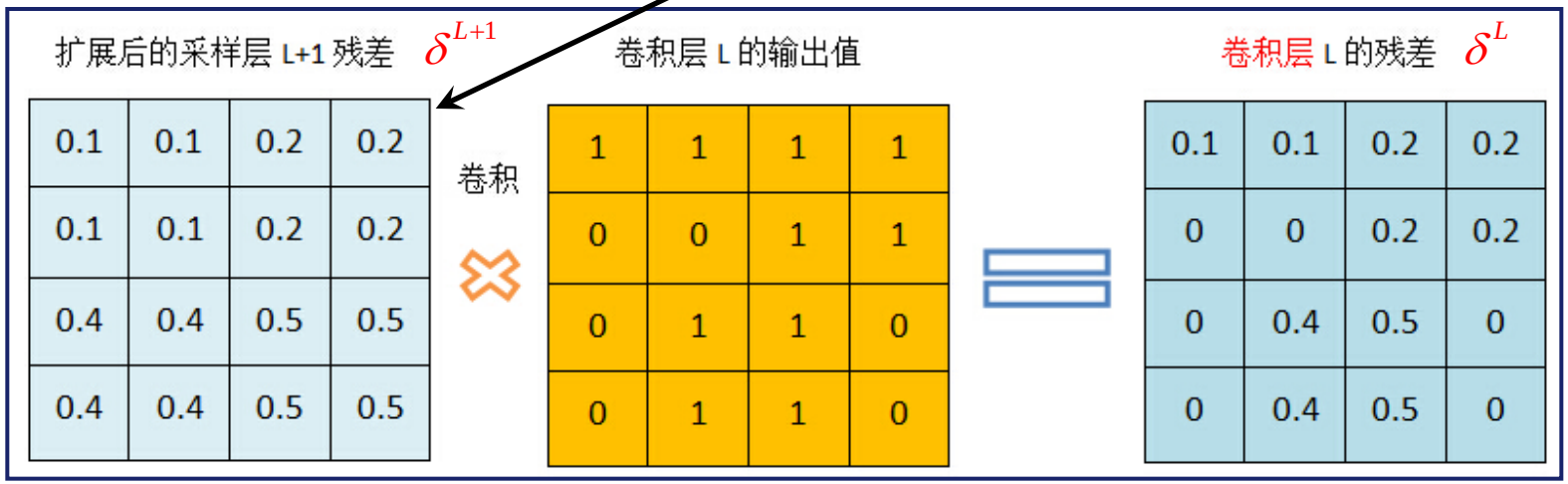


卷積層(L)的敏感度 (由池化層遞迴推導求解) 卷積層(L)←池化層(L+1)



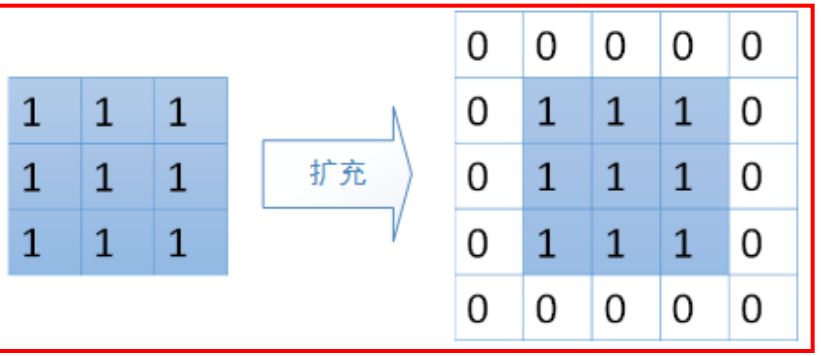
池化層(L+1)的敏感度

卷積層L的敏感度



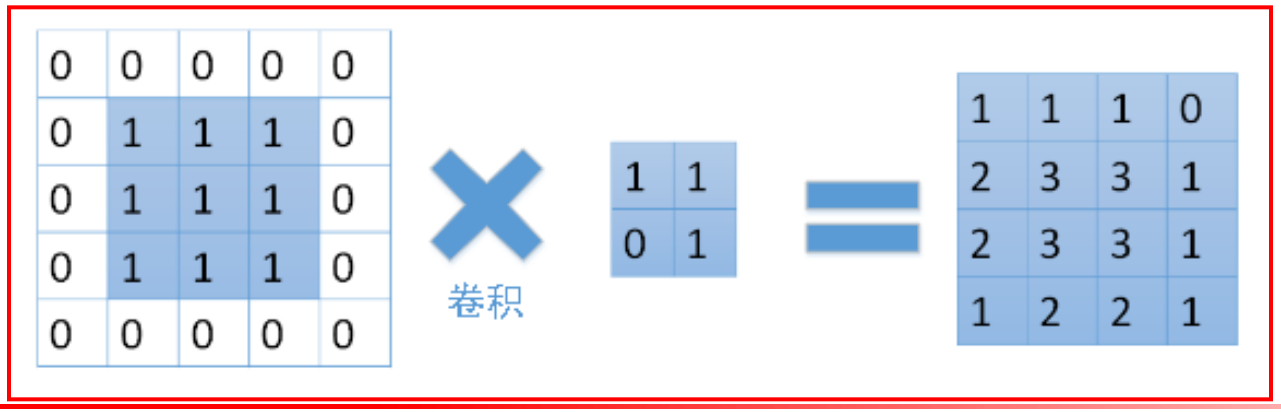
池化層的敏感度 (由卷積層遞迴推導求解) (池化層←卷積層) (example 1)

已知卷積層的梯度矩陣D (內含δ) (3*3)



梯度矩陣D擴充至
n+2*(k-1)*n+2*(k-1)
其中n是梯度矩陣D的大小(亦是卷積層的大小)，k是卷積核的大小
所以 (3+2*(2-1))*(3+2*(2-1))=5*5矩陣

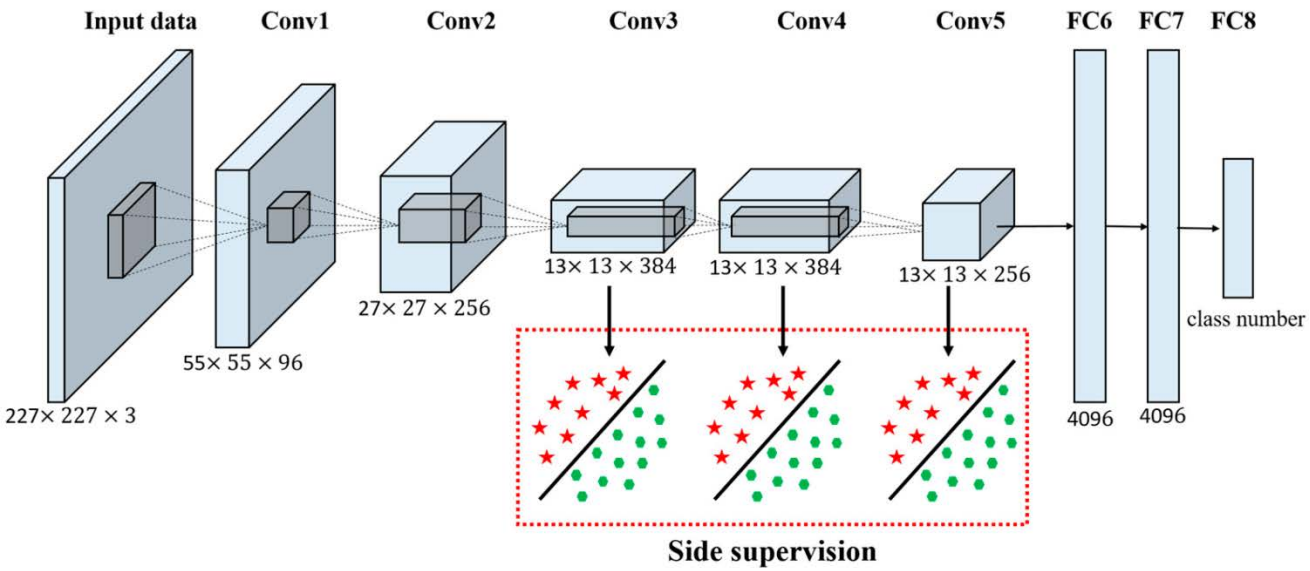
卷積核(k=2，2x2)
旋轉卷積核180度

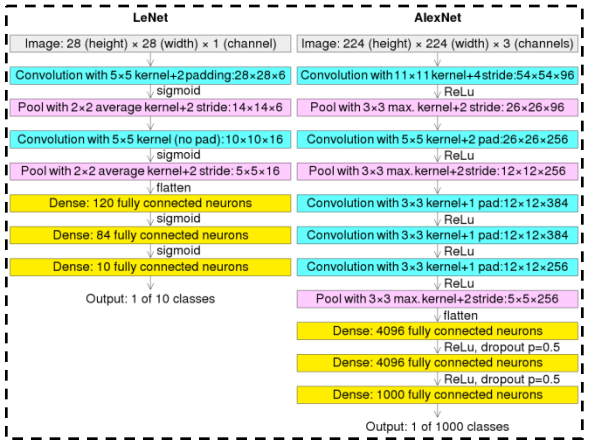


反向傳遞得到池化層的梯度矩陣D (內含δ)

AlexNet, 2012

- AlexNet是Hinton學生Alex Krizhevsky所設計的CNN架構，2012年參與ILSVRC競賽，在圖像分類任務獲得優勝，其Top 5 error rate領先第2名近10個百分點。
- AlexNet是8層(含卷積層與全連接層)的神經網路，輸入圖像大小是224*224*3。AlexNet的輸入維度與堆疊的層數，都較LeNet-5多且深。

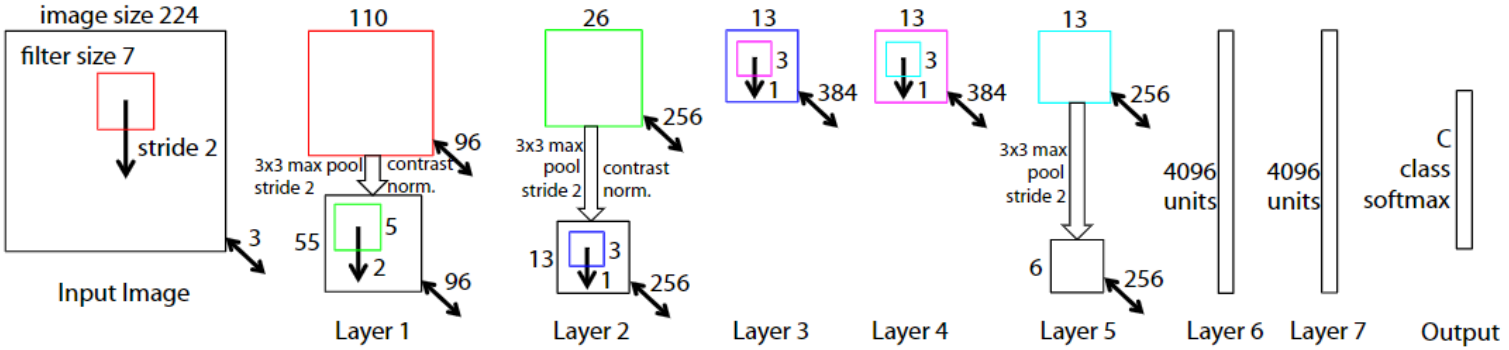


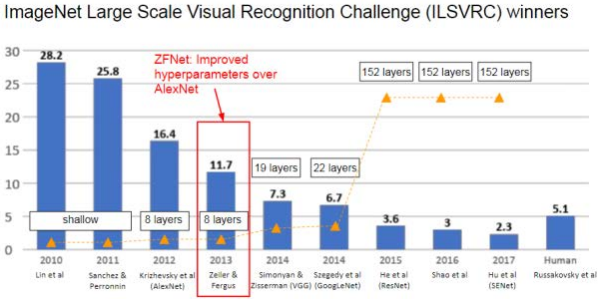


5層　　8層

ZFNet (Zeiler and Fergus), 2013

- ZFNet的網路架構是以AlexNet為基礎，經過對中間層的**特徵視覺化**，觀察模型中間層的特徵，**調整模型的卷積核大小與步長**。
  - AlexNet的第2層出現aliasing(aliasing是指在取樣頻率過低時，出現不同訊號混淆的現象)。作者認為這是第1個卷積層stride過大引起(為解決這個問題，可以提高取樣頻率)，所以將stride從4調整為2，與之相應的將kernel size也縮小(可視為stride變小，kernel沒有必要看那麼大範圍)。
- 第1個卷積層，kernel size從11減小為**7**，將stride從4減小為**2**

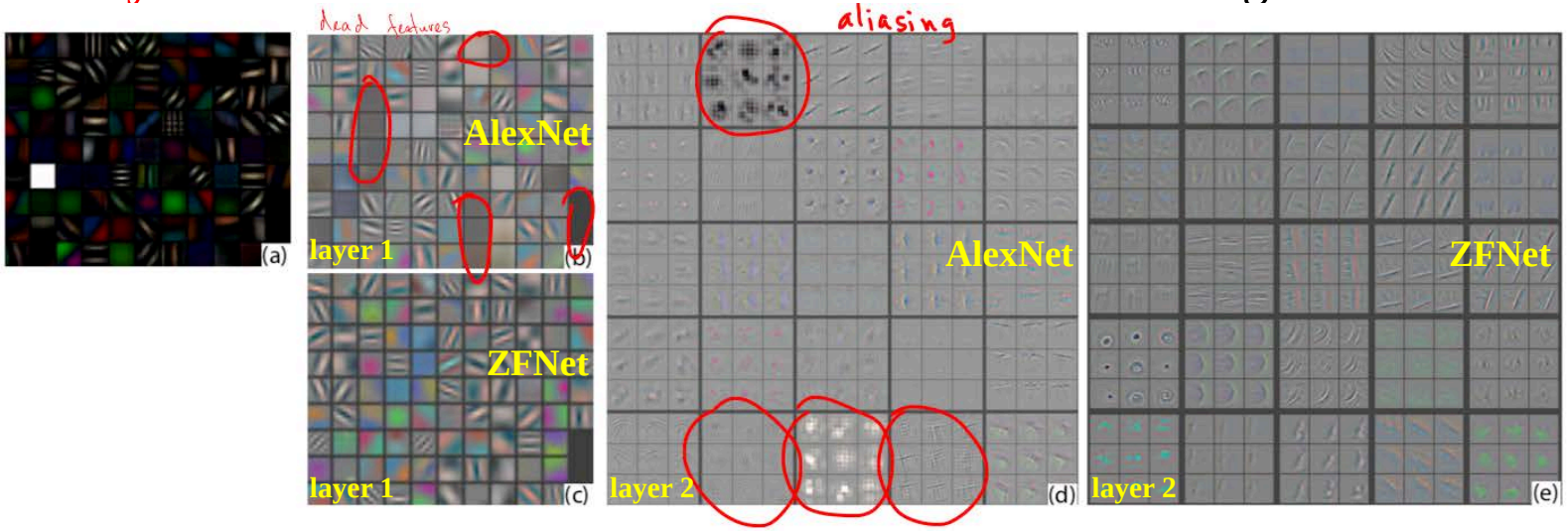




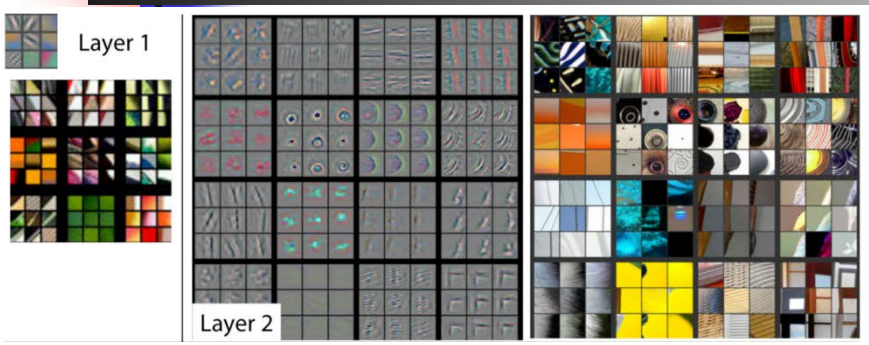
- CNN視覺化主要有以下幾類：
  - **視覺化隱藏層特徵**：理解每個CNN filter將輸入變成什麼輸出。亦即將隱藏層特徵進行逆向得到輸入圖像，可以知道究竟隱藏層的特徵對應的原始圖是什麼。
  - **視覺化CNN的Filters**：理解每個filter代表什麼模式。
  - **視覺化啟動的類別**：理解圖片那部分決定最後的類別。

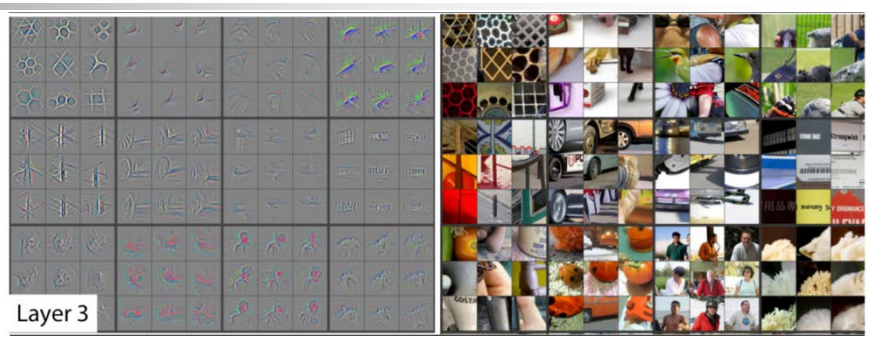
ZFNet v.s. AlexNet

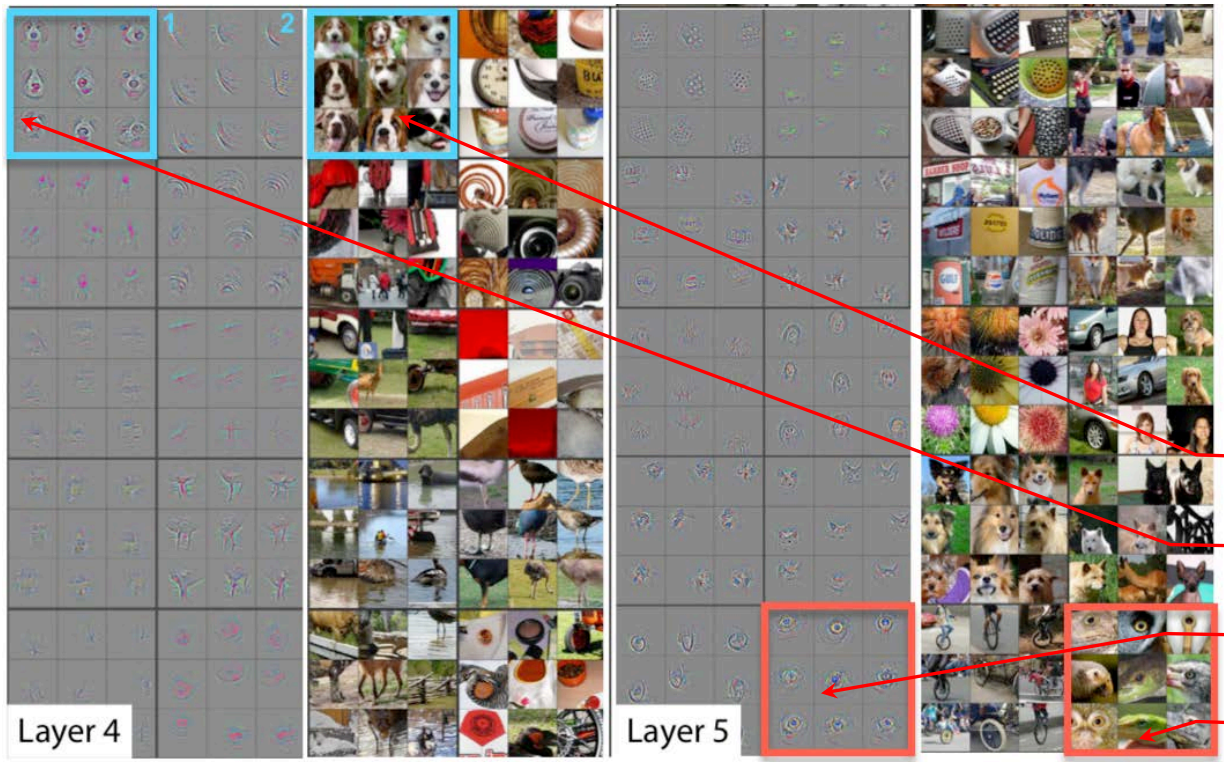
- (a) **1st layer** ZFNet features without feature scale clipping.
- (b) **1st layer** features from AlexNet. Note that there are a lot of **dead features** - ***ones where the network did not learn any patterns***.
- (c) **1st layer** features for ZFNet. Note that are only a few dead features.
- (d) **2nd layer** features from AlexNet. The grid-like patterns are so-called **aliasing artifacts(混疊)**. *They appear when **receptive fields of convolutional neurons overlap** and **neighboring neurons learn similar structure***.
- (e) **2nd layer** features for ZFNet. Note that there are no aliasing artifacts.



Convnet Visualization (Unpooling and Deconvolution)







ZFNet嘗試將每層中activation值最大的部分，還原在**輸入圖像的那個區塊**(特徵圖反向映射回原始輸入所在的圖像空間)，並發現較為淺層的捲積層捕捉到的只是單純的線條與顏色，然而越深層的卷積層捕捉到的會越趨完整的物件。

patch of the input image

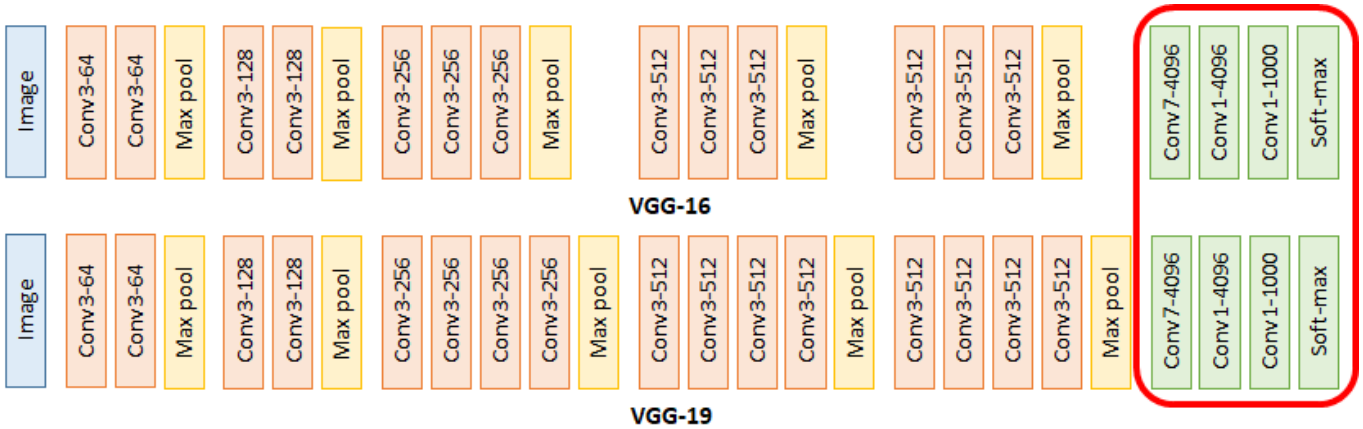
相對應的特徵圖

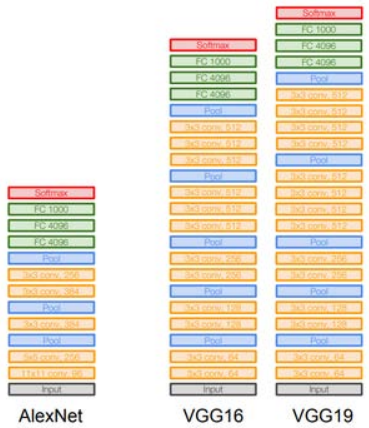
相對應的特徵圖

patch of the input image

VGGNet, 2014(亞軍)

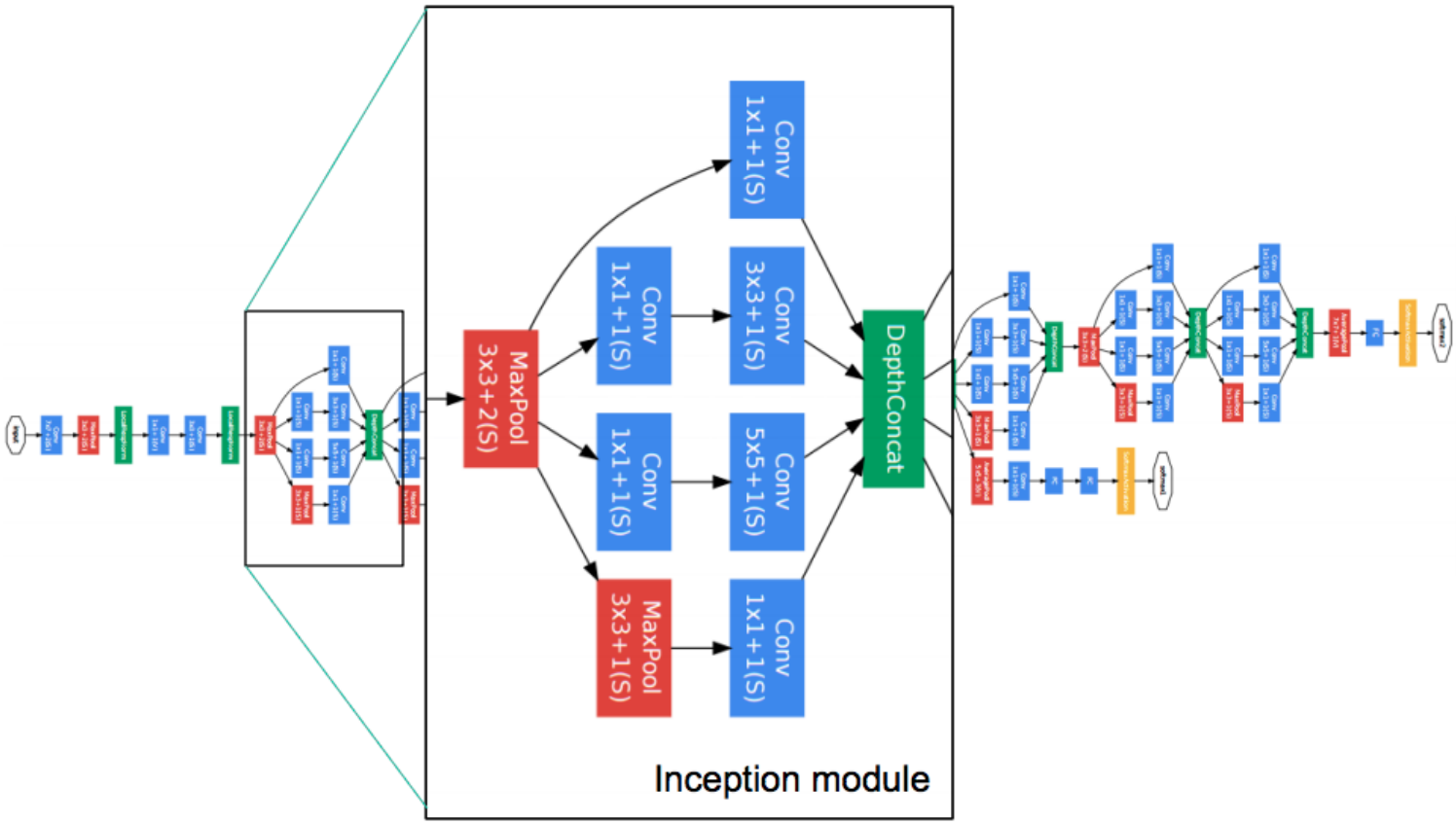
- 2014年牛津大學的VGG以19層(VGG19)的CNN，將Top 5 error rate從AlexNet 的16.4%下降到7.3% (2014 ILSVRC競賽亞軍)。





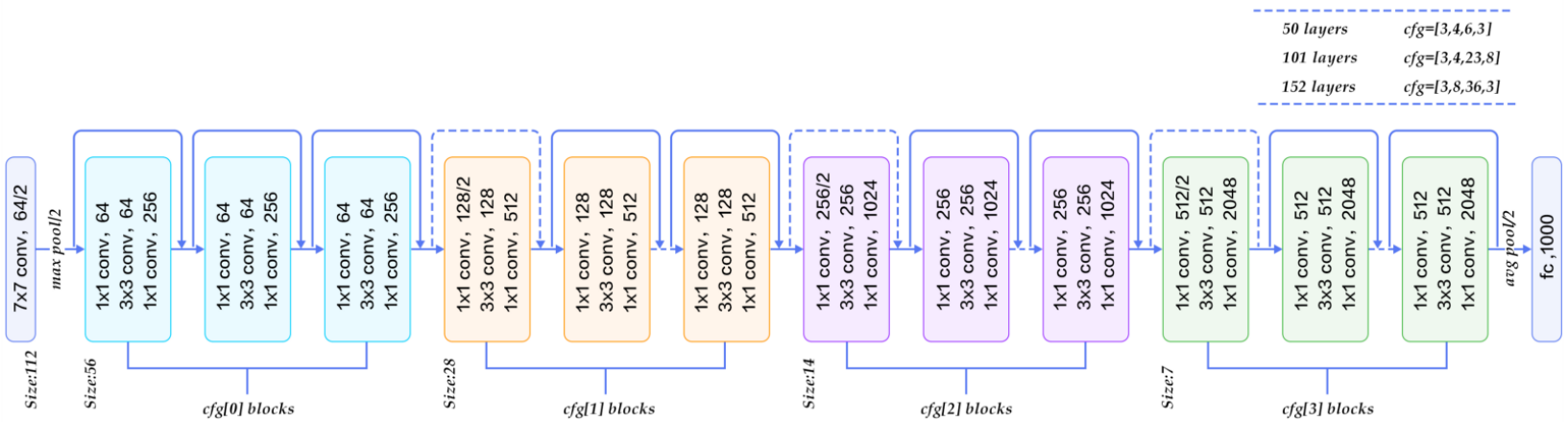
GoogleNet, 2014(冠軍)

- 2014年GoogleNet(**四個並行卷積的層，這四個卷積並行的層就是inception 模組**)以22 層的CNN將Top 5 error rate 下降到 6.7% (2014 ILSVRC競賽冠軍)。**此時距離人類的 Top 5 error rate: 5.1%** 已經不遠。



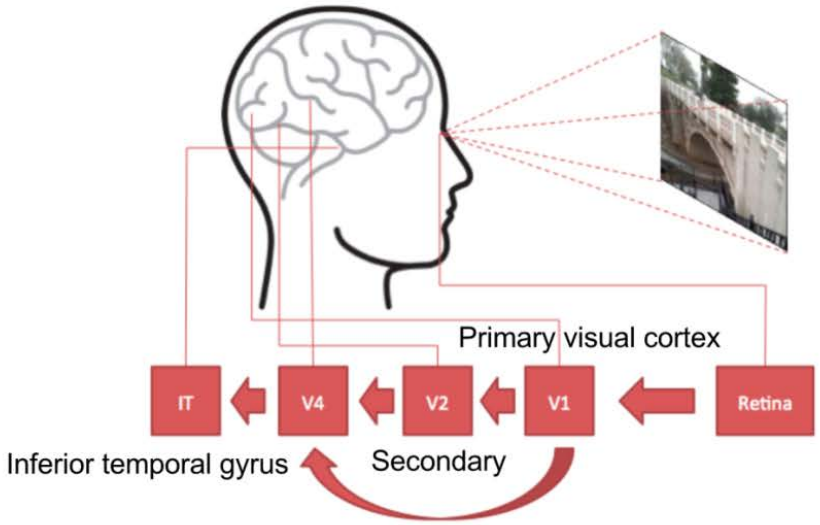
ResNet, (Kaimei He, 2015)

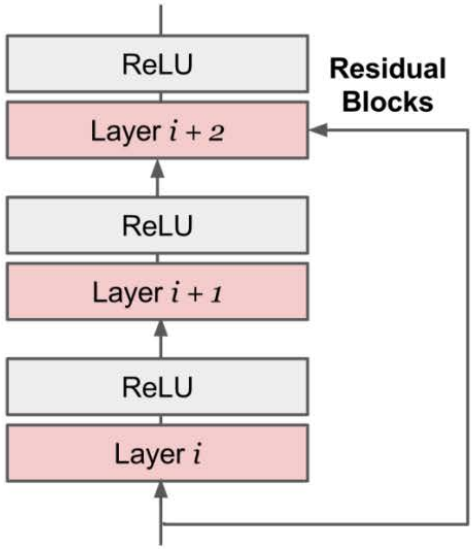
- 隨著 AlexNet, VGG, GoogleNet層數的加深，模型的表現也越來越好，**是否The deeper, the better ?** 但答案並非那麼的顯然，甚至人們發現在層數加更深的情況下模型表現反而下降。**(Vanishing Gradient Problem and Degradation Problem)**
- 2015年在微軟研究院的何凱明(Kaimei He)設計帶有**殘差結構(residual block)**的神經網路，並發表152層的ResNet。ResNet在2015年的ILSVRC競賽獲得冠軍，Top 5 error rate達到3.57%，準確率已經超越人類的 5.1%。
- In **ResNet**, **identity mapping** is proposed to promote the gradient propagation. Element-wise addition is used. It can be viewed as algorithms with a state passed from one ResNet module to another one.
- Resnet helps in building large neural networks by **ensembling short Resnet blocks**.

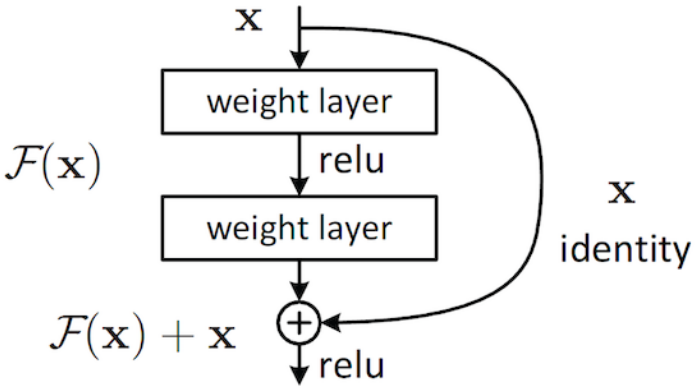


ResNet

- ResNet殘差塊的設計靈感來自人類視覺皮層系統中V4直接從V1獲取輸入。



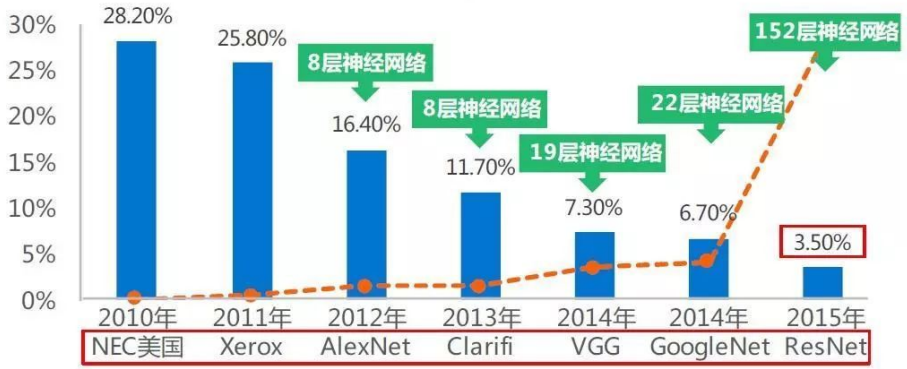


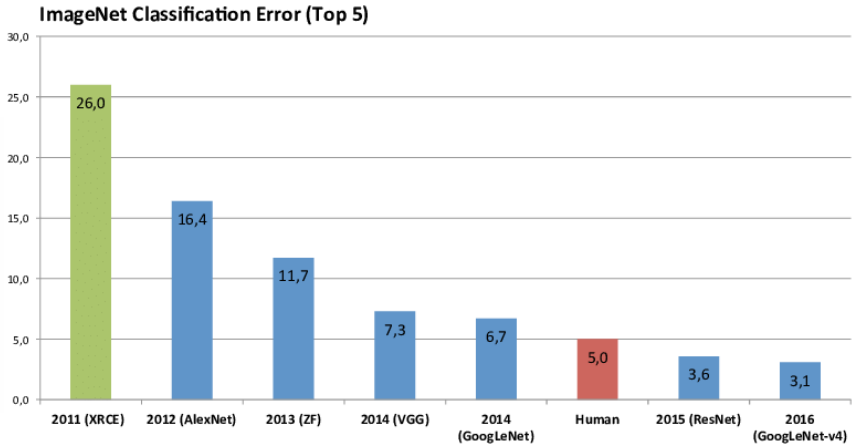


H(x)=F(x) + x

- H(x) : Resnet Block
- F(x) : Residual Mapping (殘差映射)
- x : Identity Mapping(恆等映射)//Skip Connections

ILSVRC Competition History





Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

Thank You

for Your Attentions !!!