AI 新手村指南

初學技能--深度學習

對你來
說....

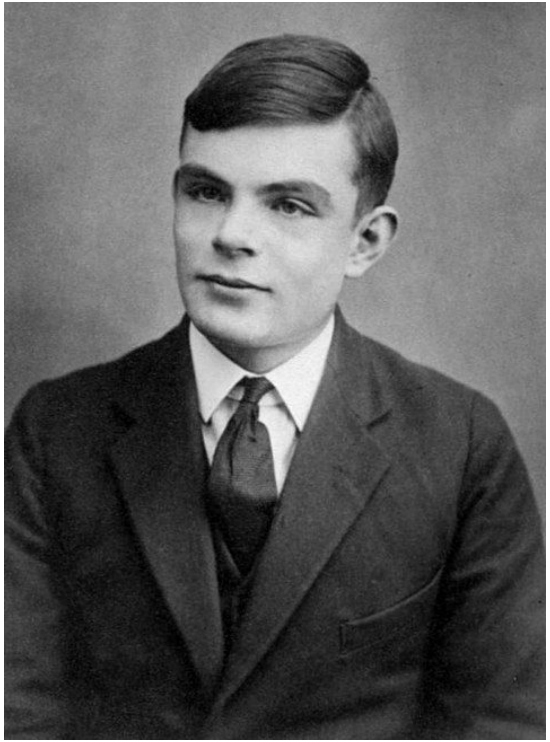
什麼是 AI ??

什麼東西叫 AI ??

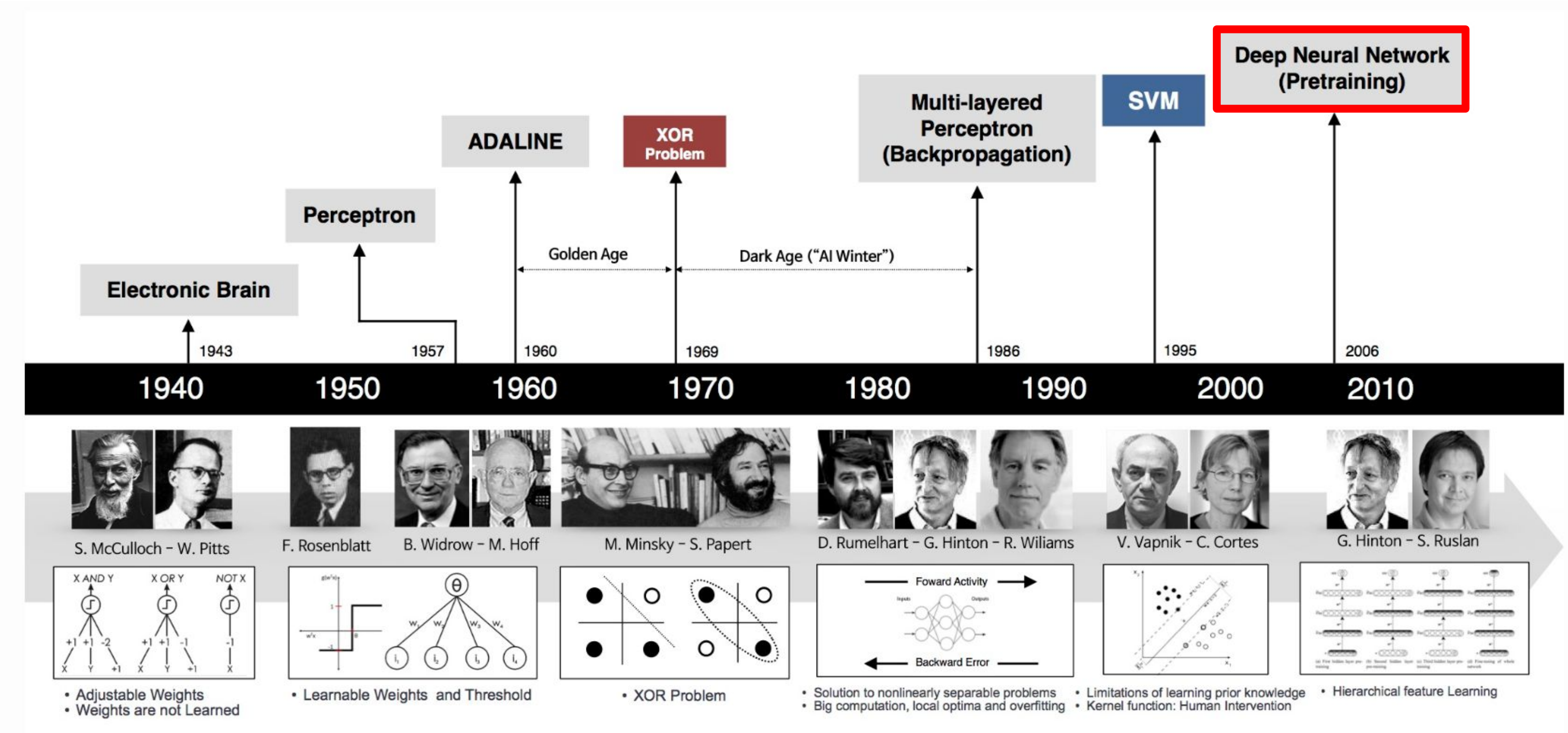
其實 AI 是種演算法

Alan Mathison Turing (1912-1954)

- 是計算機科學家、數學家、邏輯學家、密碼分析學家和理論生物學家
- 被譽為「計算機科學之父」、「人工智慧之父」
- 著名的文章--- *Computing machinery and intelligence*，提出「Can Machines Think?」
- Turing Test 圖靈測試(1950)



站在巨人的肩膀--AI 人工智慧的里程碑



https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html

哪時候你開始發現....

AI 離我好像很近....

The Milestone of the Deep Learning

2006 Netflix | **2010 ImageNet** | **2016 Alpha Go**

Important Milestone

2006 RBM (Hinton)
- Deep Learning Base Recommandation

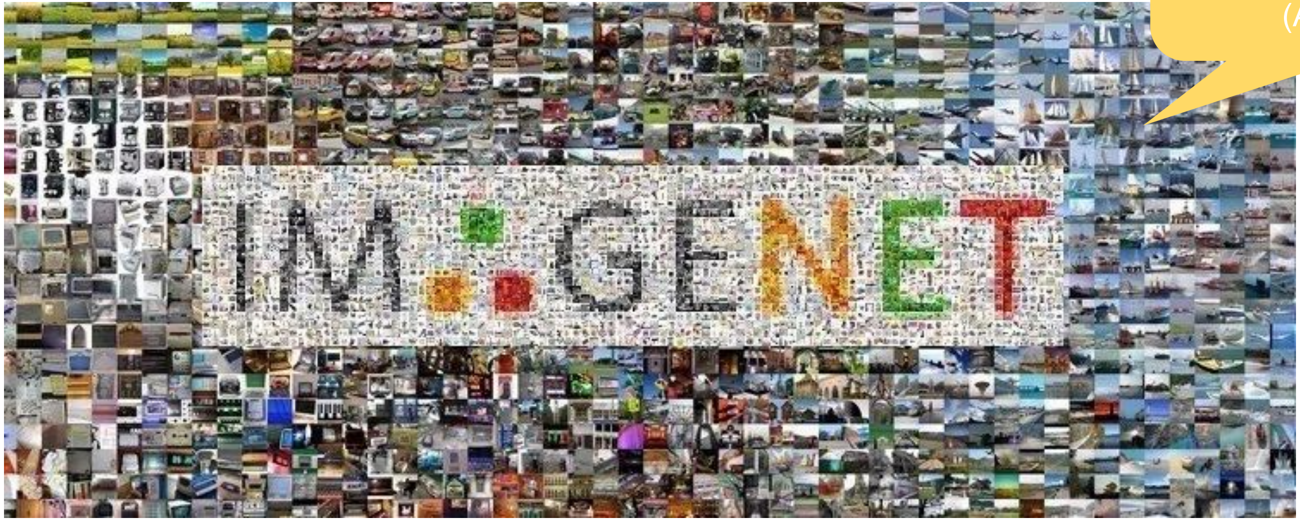
哪時候開始關注「深度學習」的呢？

2006
Netflix

2010
ImageNet

2016
Alpha Go

ILSVR（ImageNet Large Scale Visual Recognition Competition）



Important Milestone

2012 AlexNet
(Alex, Hinton)

哪時候開始關注「深度學習」的呢？

2006
Netflix

2010
ImageNet

2016
Alpha Go

<u>Important Milestone</u>

2016 AlphaGo
對戰世界冠軍李世石

讓機器下圍棋有多厲害？

棋類名稱	棋局所有變化可能	程式擊敗職業棋士紀錄	程式作者
西洋跳棋 Checkers	10^{32}	Chinook，6 平手 Marion Timsley (1994)	Jonathan Schaeffer (Alberta University)
黑白棋 Othello	10^{58}	Logistello，6:0 勝村上健 (1997)	Michael Buro (Othello strong player)
九路圍棋 9 X 9 Go	10^{85}	東華七號，七番賽 4:3 勝職業棋時黨希昀 (2013/6) 七番賽 4:1 勝職業棋士蕭愛霖 (2014/6)	周政緯、顏士淨 (東華大學)
西洋棋 Chess	10^{123}	DeepBlue，3.5-2.5 勝卡斯帕羅夫 (1997/5)	許峰雄深藍團隊 (IBM)
象棋 Chinese Chess	10^{150}	Shiga，紅先勝陳振國八段 (2006/ 11)	鄭明政、顏士淨 (東華大學)
日本將棋 Shogi	10^{226}	Tsutsukana 勝森下卓九段 (2014/4)	一丸貴則 (名古屋大學)
圍棋 19 X 19 Go	10^{400}	AlphaGo，4:1 勝李世九段 (2016/3)	谷歌 DeepMind 團隊

2019 兩大史事

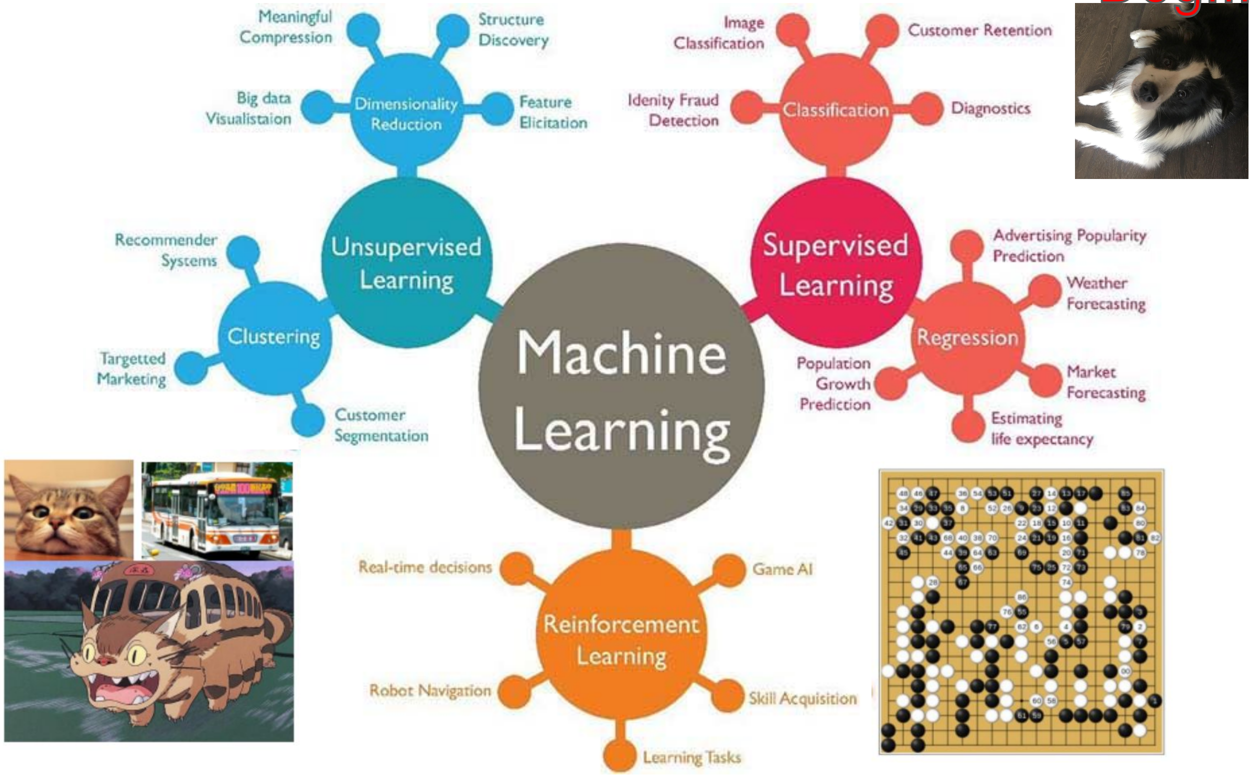
Deep Mind -- AlphaStar

Microsoft -- Suphx 天鳳

什麼問題都用 AI
就好了嗎??

機器學習的問題種類

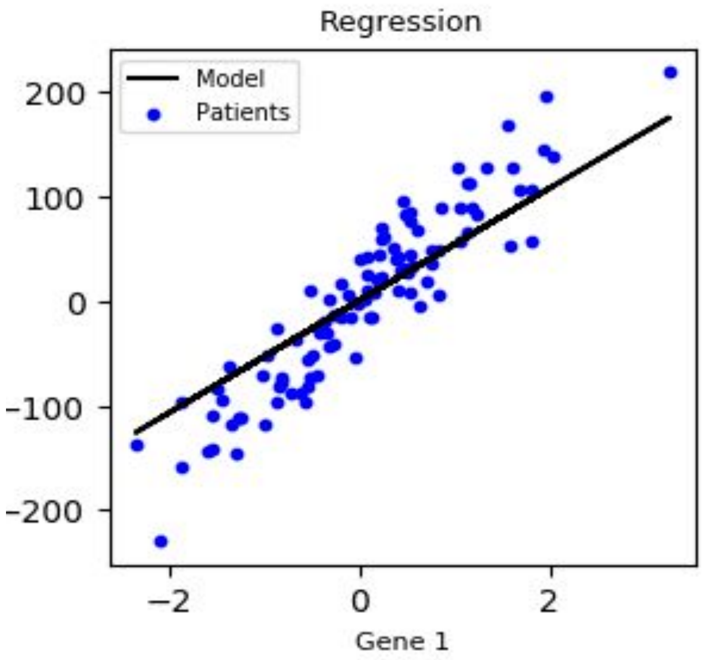
- 監督式學習
  Supervised Learning
  - Regression 迴歸
  - Classification 分類
- 非監督式學習
  Unsupervied Learning
  - Clustering 分群
  - Dimensionality Reduction
    維度縮減
- 強化式學習
  Reinforcement Learning

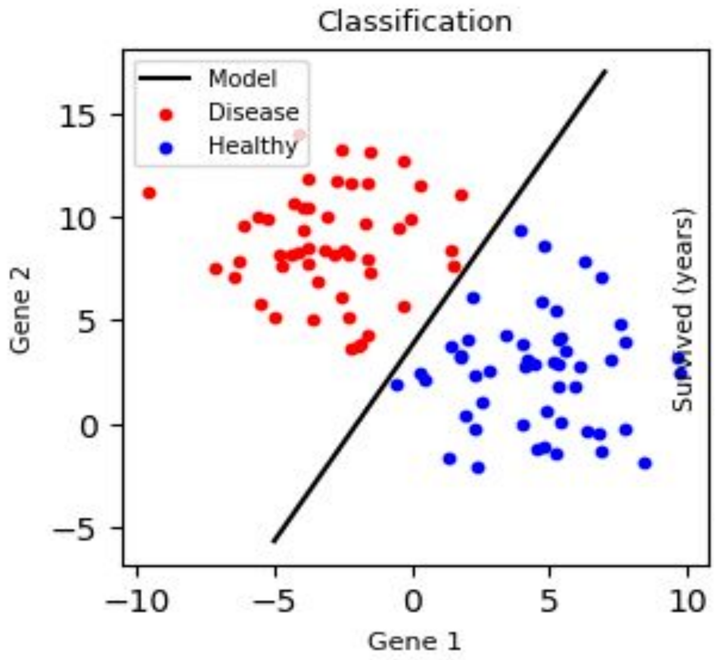


Dog!!!

https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/

了解問題--迴歸與分類問題

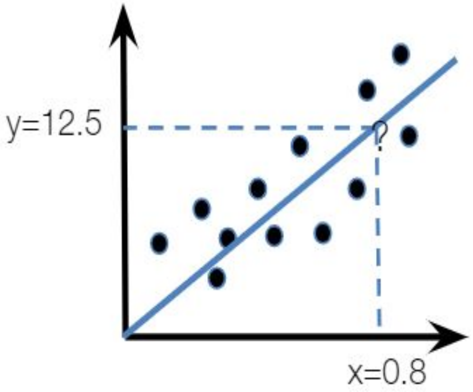




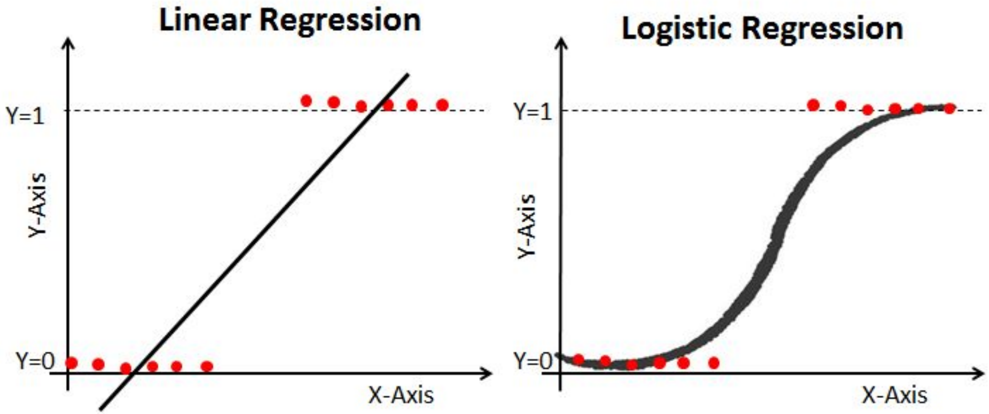
了解問題--迴歸與分類問題

線性迴歸
Linear Regression

$$y = wx + b$$

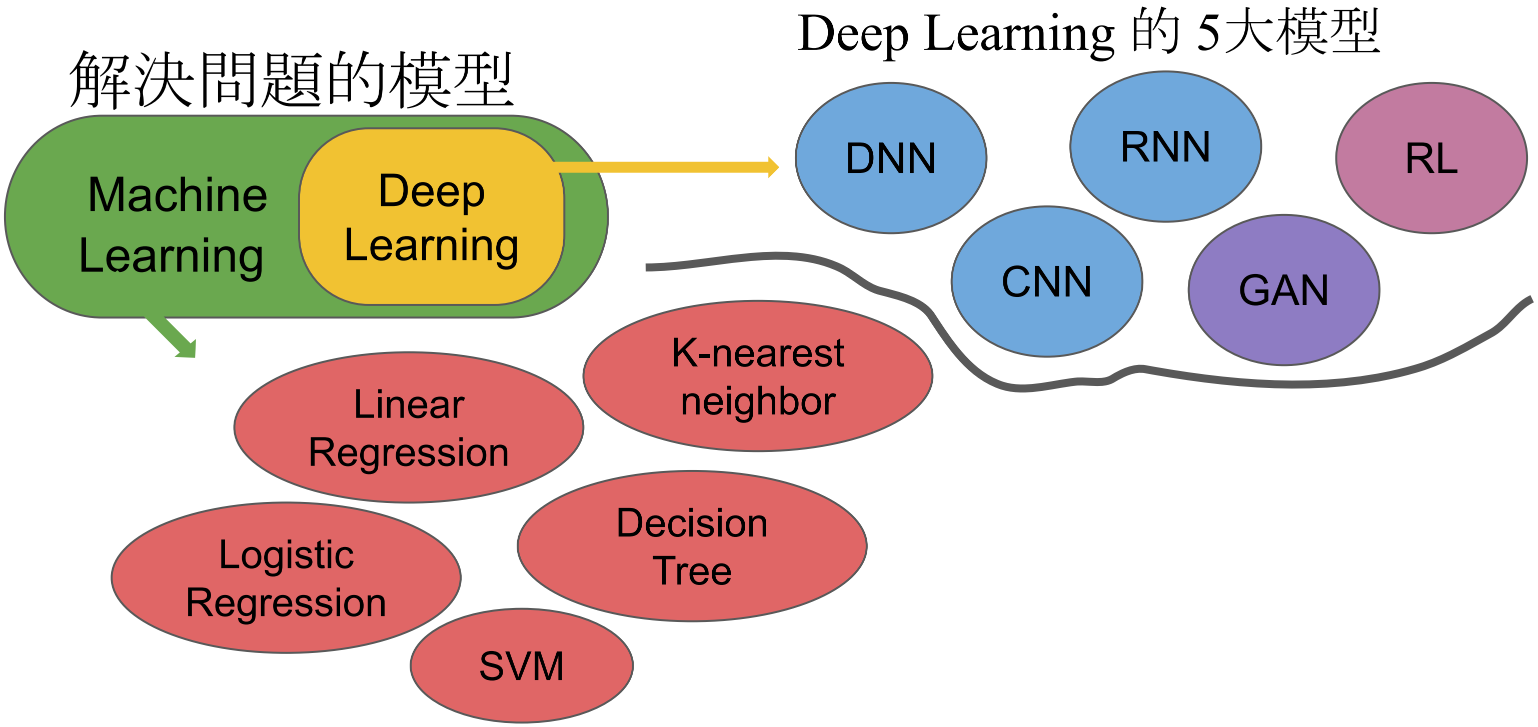


邏輯斯迴歸
Logistic Regression



$$f(x) = \frac{1}{1 + e^{-(wx+b)}}$$

機器學習方法百百種



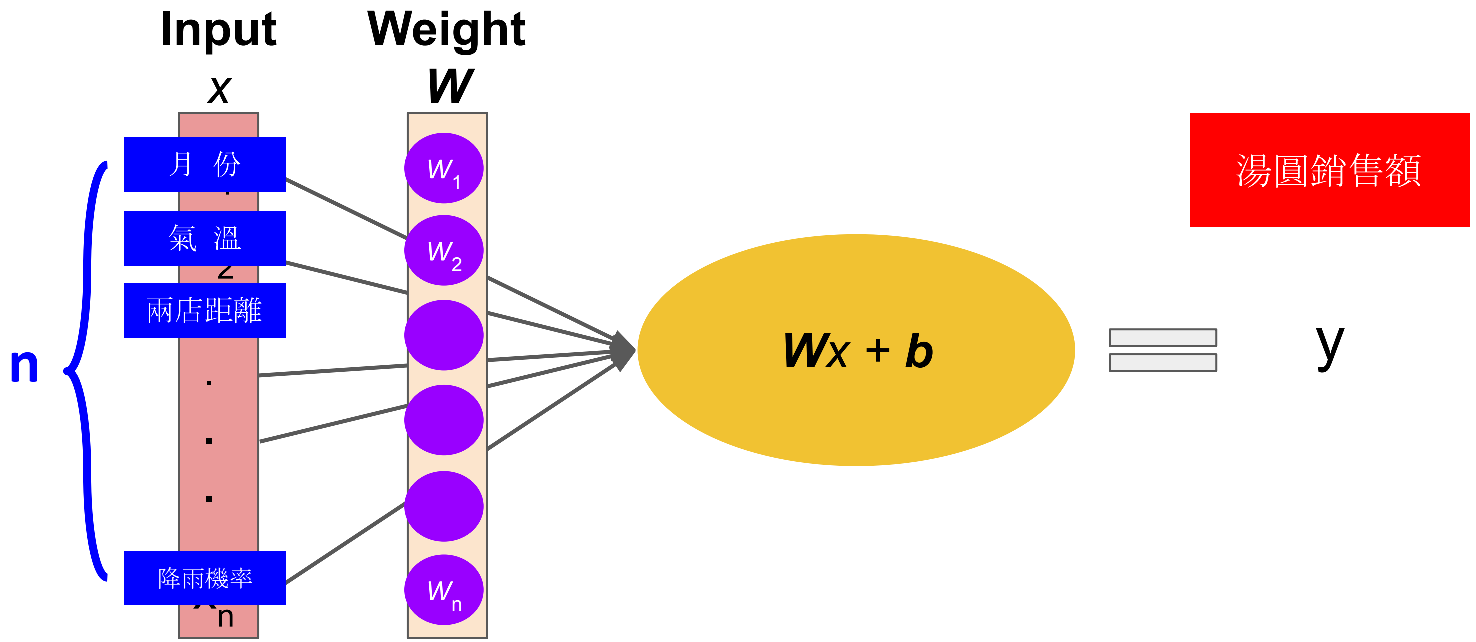
解決問題的模型
Deep Learning 的 5大模型
Machine Learning
Deep Learning
DNN
RNN
RL
CNN
GAN
K-nearest neighbor
Linear Regression
Decision Tree
Logistic Regression
SVM

小試一下身手～～

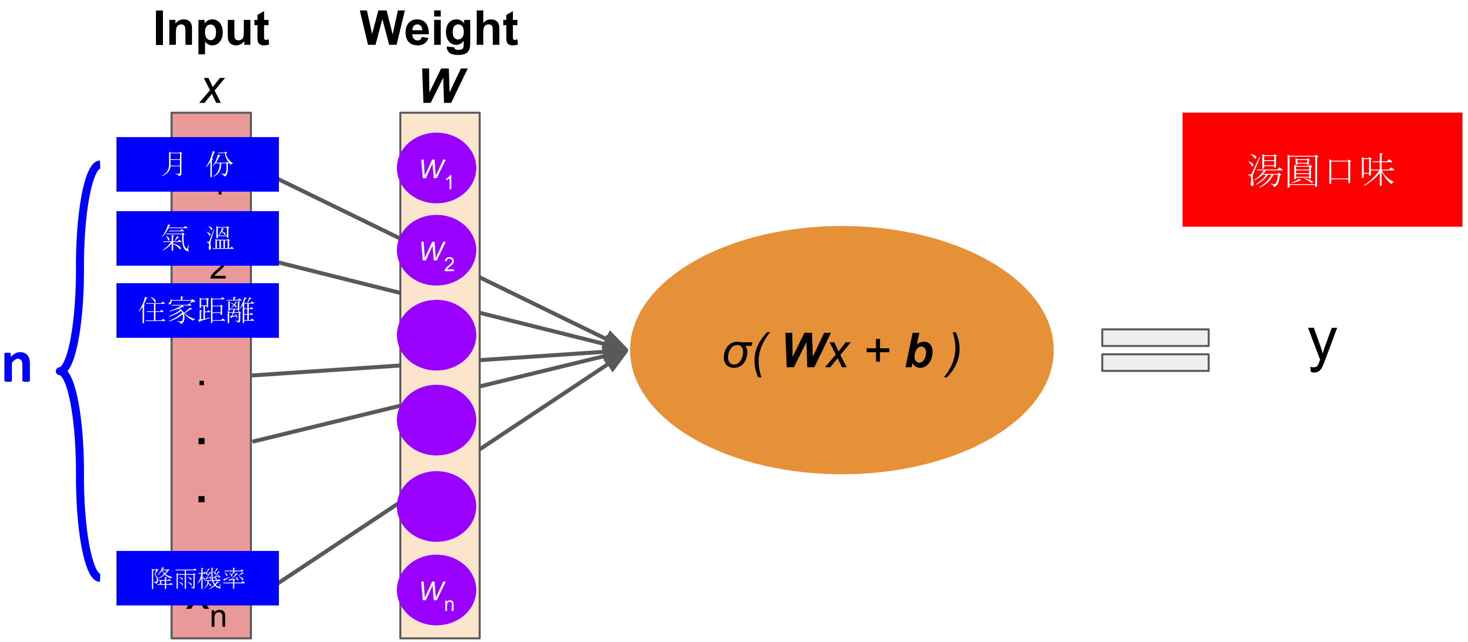
Deep Learning 深度學習

神奇的....
類神經網路模型

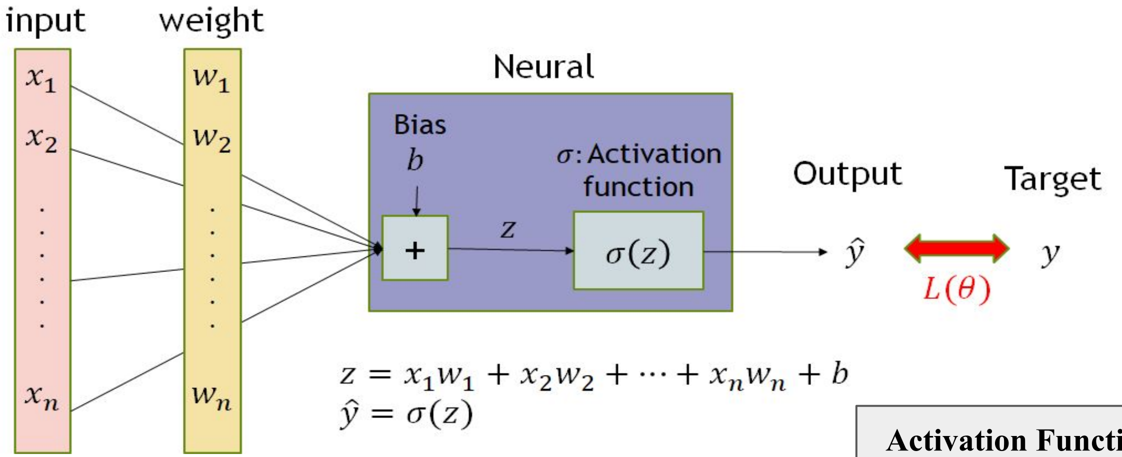
實際的迴歸問題....



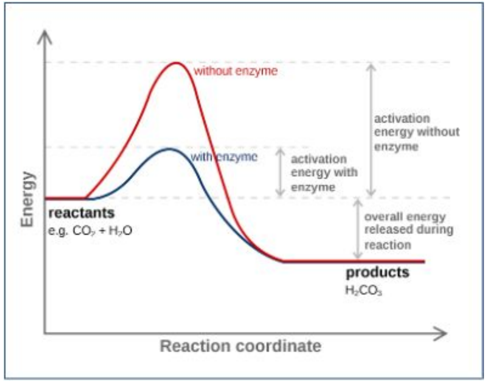
實際的分類問題....

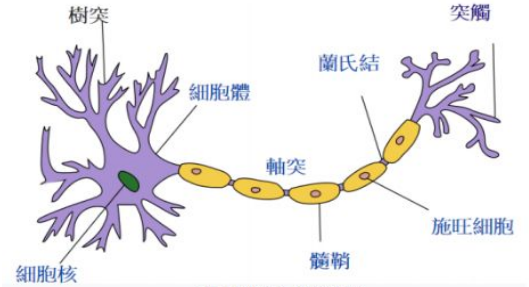


Neural Network 類神經網路 --- A Neuron



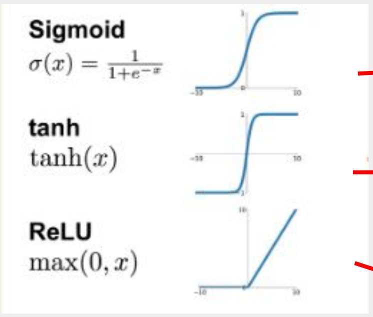
$$z = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b$$
$$\hat{y} = \sigma(z)$$





典型神經元的結構

Activation Function 基本款

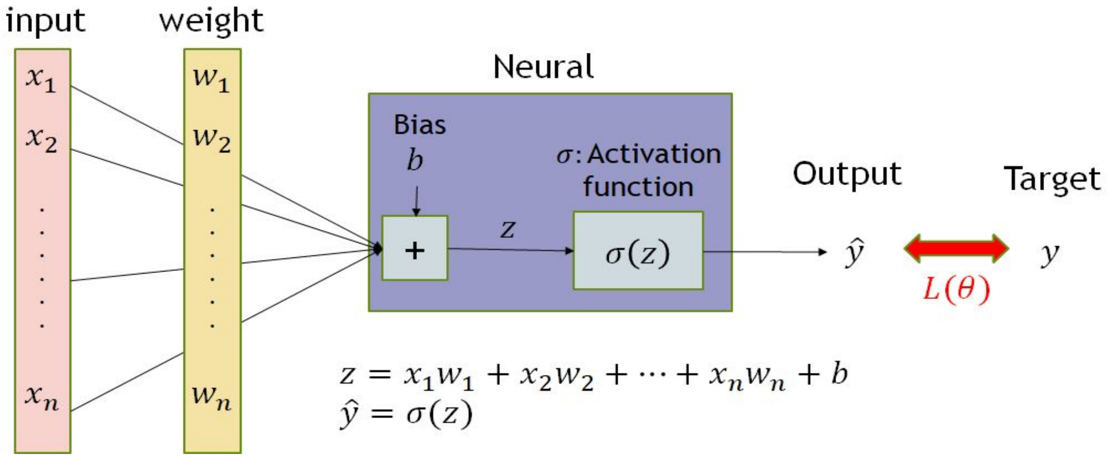


$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

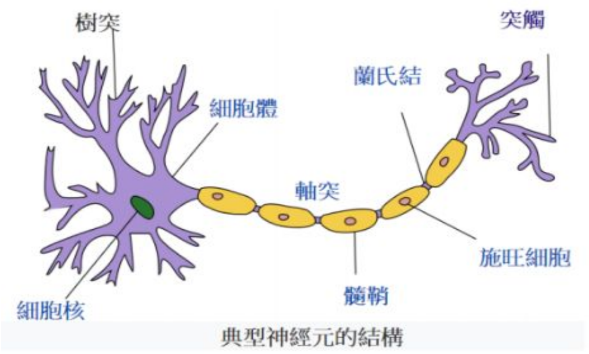
$$\sigma'(x) = 1 - \sigma(x)^2$$

$$\sigma'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

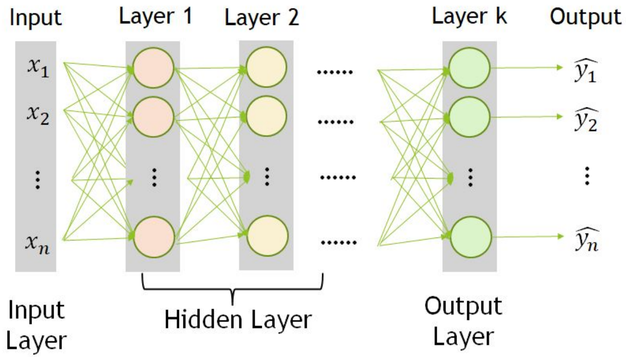
何謂 “Deep” Neural Network？



$$z = x_1w_1 + x_2w_2 + \cdots + x_nw_n + b$$
$$\hat{y} = \sigma(z)$$



x	input	從不同來源來的資訊
w	weight	對來源的訊號強弱敏感程度
b	bias	對某件事情的成見
z	$z = wx + b$	綜合所有來源資訊
σ	Activation Func.	資訊分析的條件
y	$y = \sigma(z)$	最後的結果



三大神經元、三個積木塊

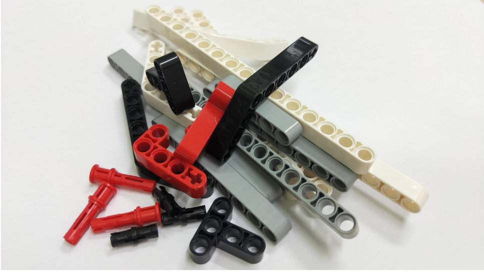
CNN

視覺神經

RNN

遞迴神經

DNN



大腦神經

定義模型好不好

評估模型好壞的指標---Loss Function

- mean absolute error

$$L(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{N}\|y_i - \hat{y}_i\|$$

- mean square error

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

- R^2 score

$$L(y,\hat{y}) = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

- binary crossentropy

$$\hat{y} = \sigma(z) = \frac{1}{1 + \mathrm{e}^{-wz+b}}$$

$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[\,y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\,]$$

- categorical crossentropy

$$\hat{s}_i = \frac{\mathrm{e}^{\hat{y}_i}}{\sum_j \mathrm{e}^{\hat{z}_j}}$$

$$L(y,\hat{s}) = -\frac{1}{N}\sum_{i=1}^{N} s_{i1}\log\hat{s}_{i1} + \cdots + s_{im}\log\hat{s}_{im}$$

- $F1$-score

$$Precision(y,\hat{y}) = \frac{I(y_i = \hat{y}_i = 1)}{I(\hat{y}_i)} = \frac{TP}{TP + FP}$$

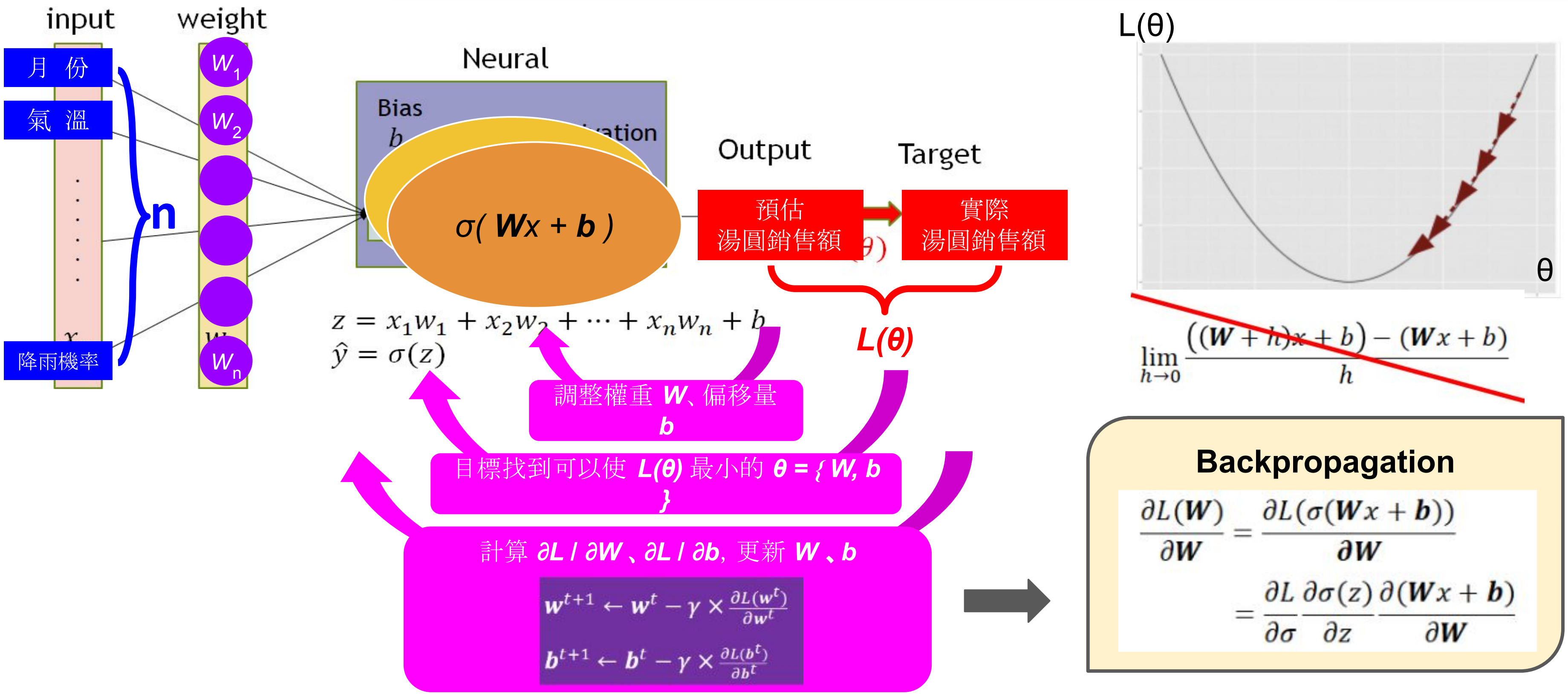
$$Recall(y,\hat{y}) = \frac{I(y_i = \hat{y}_i = 1)}{I(y_i)} = \frac{TP}{TP + FN}$$

$$F1(y,\hat{y}) = \frac{2pr}{p + r}$$

		predicted condition	
	total population	prediction positive	prediction negative
true condition	condition positive	True Positive (TP) 真陽性	False Negative (FN) (type II error) 偽陰性
	condition negative	False Positive (FP) (Type I error) 偽陽性	True Negative (TN) 真陰性

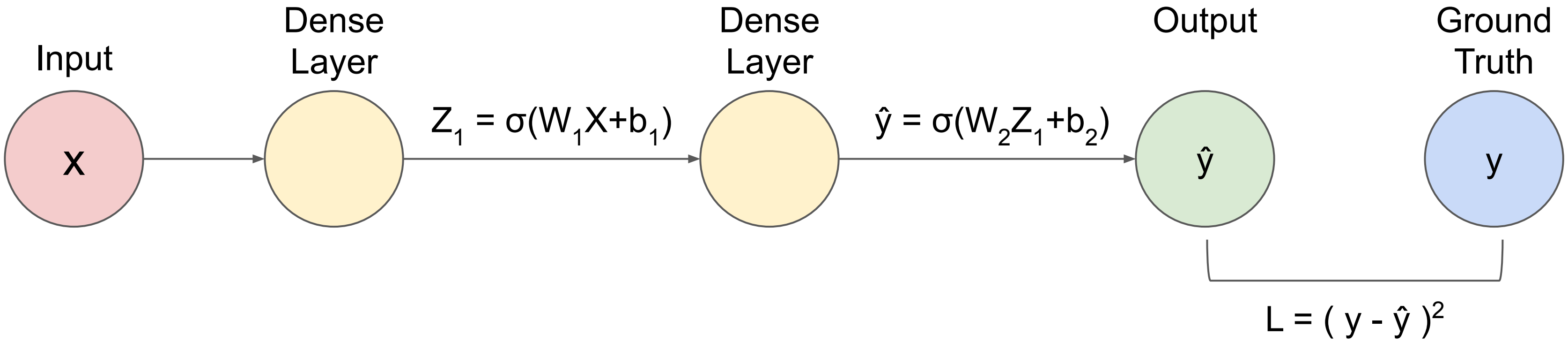
訓練神經網路....

什麼是訓練模型？



input
weight
月 份
氣 溫
降雨機率
n
W_1
W_2
W_n
Neural
Bias
b
σ(**W**x + **b**)
Output
Target
預估
湯圓銷售額
實際
湯圓銷售額
$z = x_1w_1 + x_2w_2 + \cdots + x_nw_n + b$
$\hat{y} = \sigma(z)$
L(θ)
調整權重 **W**、偏移量 **b**
目標找到可以使 **L(θ)** 最小的 **θ = { W, b }**
計算 ∂L / ∂W、∂L / ∂b，更新 **W**、**b**
$\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \gamma \times \frac{\partial L(\boldsymbol{w}^t)}{\partial \boldsymbol{w}^t}$
$\boldsymbol{b}^{t+1} \leftarrow \boldsymbol{b}^t - \gamma \times \frac{\partial L(\boldsymbol{b}^t)}{\partial \boldsymbol{b}^t}$
L(θ)
θ
$\cancel{\lim_{h \to 0} \frac{((\boldsymbol{W}+h)x+b)-(\boldsymbol{W}x+b)}{h}}$
Backpropagation
$\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{\partial L(\sigma(\boldsymbol{W}x+\boldsymbol{b}))}{\partial \boldsymbol{W}}$
$= \frac{\partial L}{\partial \sigma}\frac{\partial \sigma(z)}{\partial z}\frac{\partial(\boldsymbol{W}x+\boldsymbol{b})}{\partial \boldsymbol{W}}$

算個來看看....微積分記得多少



Question:

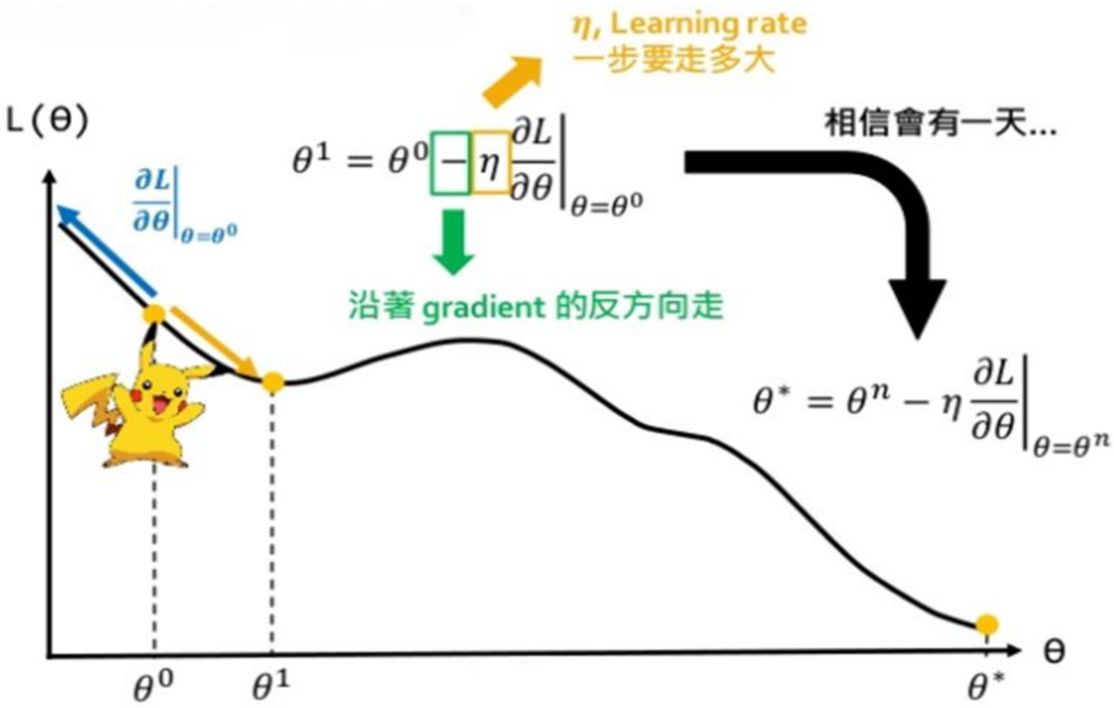
1. $\frac{\partial L}{\partial W_2}$ 、 $\frac{\partial L}{\partial b_2}$ = ?
2. $\frac{\partial L}{\partial W_1}$ 、 $\frac{\partial L}{\partial b_1}$ = ?

Activation Function:

1. $\sigma(x) = \frac{1}{1+e^{-x}}$

2. $relu(x) = \max(x, 0)$

如何學到最強攻略？



$z = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b$

$\hat{y} = \sigma(z)$

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \theta} = \boxed{\frac{\partial L}{\partial \hat{y}}} \boxed{\frac{\partial \hat{y}}{\partial z}} \frac{\partial z}{\partial \theta}$$

1.受loss function 影響

2.受activation影響

Optimizer 優化器

- Gradient Descent (GD)
  - 模型看完一輪所有資料，更新一次權重

  - 優點
    - 根據所有資料的訊息，更新模型
  - 缺點
    - 結果好壞很依賴起始點
    - 很容易掉進去 Local Minimum 就限制住找 Global Minimum 的可能

- Stochastic Gradient Descent (SGD)
  - 模型看完一小堆(batch)資料，更新權重一次

  - 優點
    - 收斂速度上，比 GD 快
  - 缺點
    - 可能會抽到極端的擾動值資料，偏離了 Global Minimum 的收斂方向

Optimizer 優化器

- RMSProp

  - 模型看完一小堆(batch)資料，更新權重一次

  - 優點
    - 不僅有依照當次計算的梯度做更新，還有參考過去的梯度做更新依據
  - 缺點
    - 還是有掉進去 Local Minimum 的可能

$$w^1 \leftarrow w^0 - \frac{\eta}{\sigma^0} g^0 \qquad \sigma^0 = g^0$$

$$w^2 \leftarrow w^1 - \frac{\eta}{\sigma^1} g^1 \qquad \sigma^1 = \sqrt{\alpha(\sigma^0)^2 + (1-\alpha)(g^1)^2}$$

$$w^3 \leftarrow w^2 - \frac{\eta}{\sigma^2} g^2 \qquad \sigma^2 = \sqrt{\alpha(\sigma^1)^2 + (1-\alpha)(g^2)^2}$$

$$\vdots$$

$$w^{t+1} \leftarrow w^t - \frac{\eta}{\sigma^t} g^t \qquad \sigma^t = \sqrt{\alpha(\sigma^{t-1})^2 + (1-\alpha)(g^t)^2}$$

- ADAM

  - 模型看完一小堆(batch)資料，更新權重一次
  - 加了 Momentum 的 RMSProp

  - 優點
    - 給予一點點小小的擾動 (Momentum)，協助翻出 Local Minimum 的小峽谷，找到 Global Minimum 的可能
    - 適合大部分的優化狀況使用

Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\widehat{m}_t \leftarrow m_t/(1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\widehat{v}_t \leftarrow v_t/(1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t/(\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)