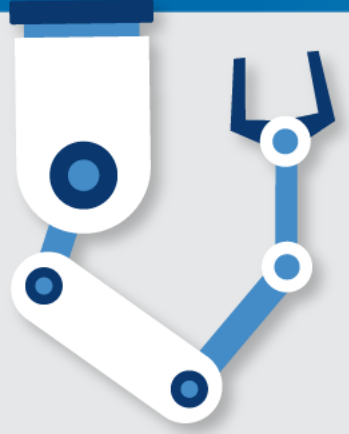
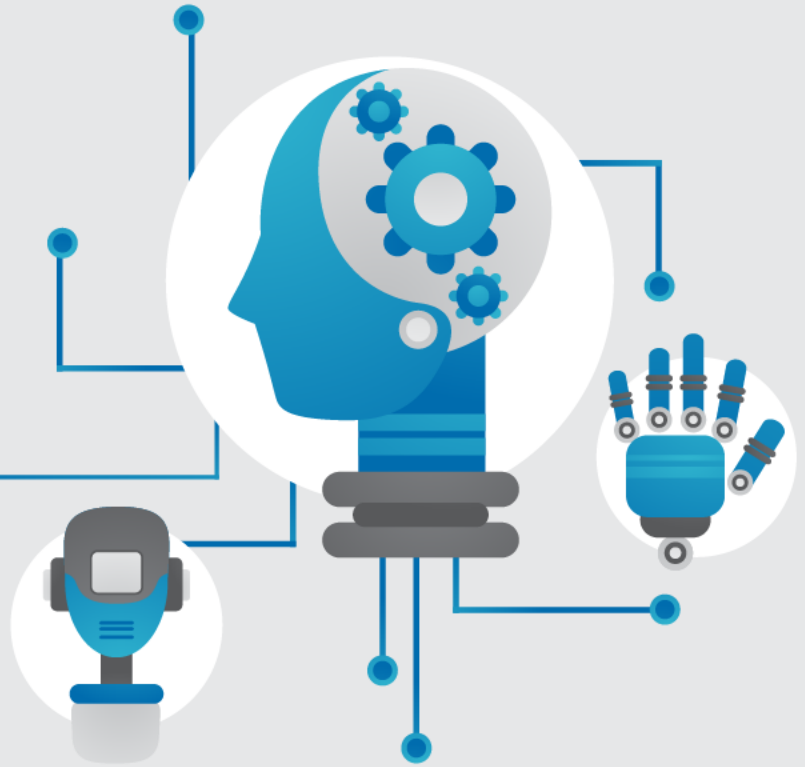
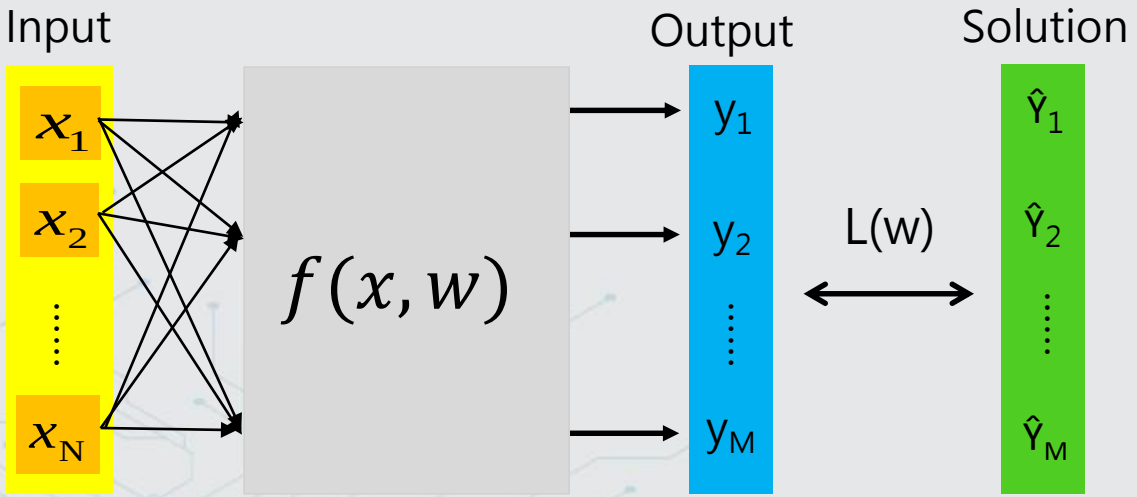


多層感知器網路概念(II)



多層感知網路效率

類神經網路就像一個函數$f(x,w)$



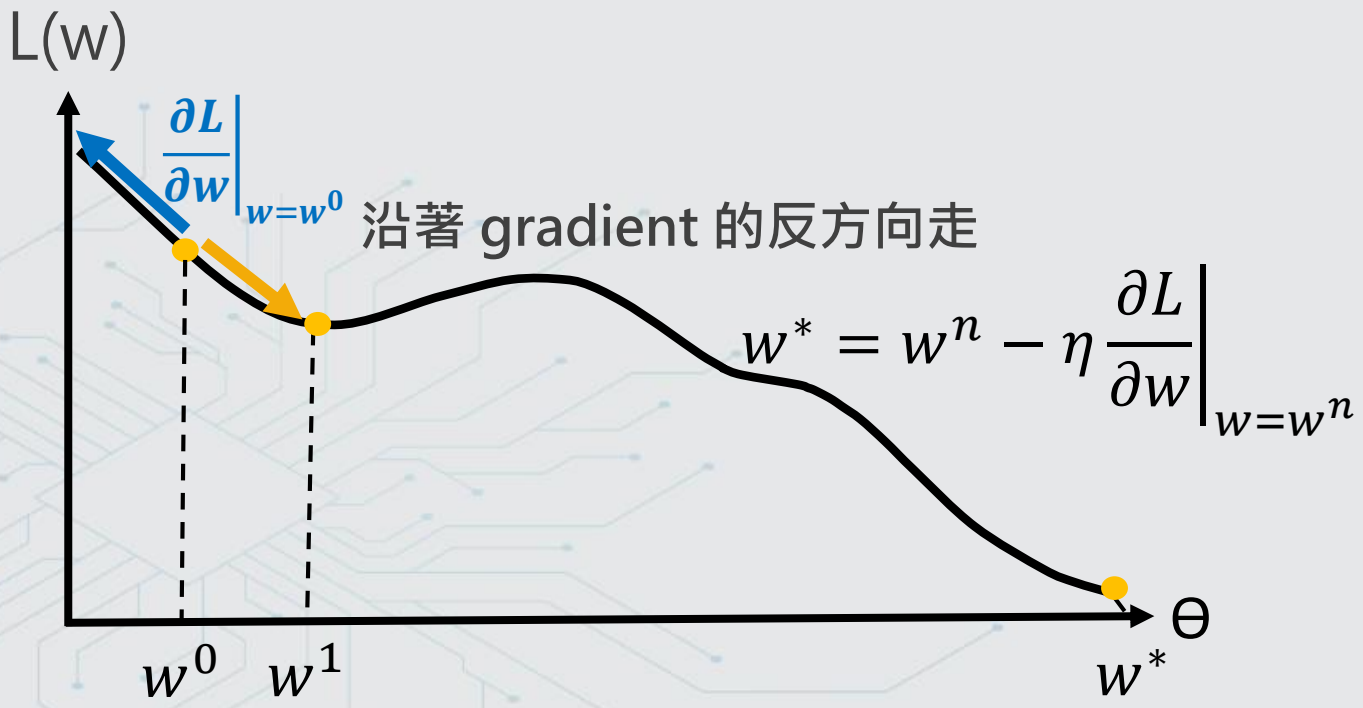
調整好的權重w*使得誤差L(w*)最小。

改進多層感知網路效率的方法

- 反向傳播（Backpropagation）
- 隨機梯度下降法（Stochastic Gradient Descent）
- 小批次梯度下降（Mini-Batch Gradient Descent）
- 動量法（Momentum）

反向傳播（Backpropagation）

反向傳播法是將類神經網路中所有**權重**對**損失函數**計算**梯度**（Gradient），用來更新權重，使得損失函數的值最小化。



影響梯度的因素

$L(w) = Y - \hat{Y}, \quad Y = \sigma(f(x,w))$

➔ $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial w} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial f}\frac{\partial f}{\partial w}$ （by chain rule）

$\partial Y/\partial f$ ：受**激活函數影響**

反向傳播法的問題

Problem 1

一次用全部訓練集的數據去計算損失函數的梯度，然後才更新一次權重，收斂速度很慢。

➔使用隨機梯度下降法（Stochastic Gradient Descent，SGD）加速收斂

Problem 2

梯度下降法不能保證找到全域最佳解。

➔利用動量法（momentum）降低困在區域最小值（local minimum）的機率

隨機梯度下降法（Stochastic Gradient Descent）

- 隨機梯度下降（SGD）是一種隨機逼近的梯度下降優化方法。
- 一次跑一個樣本，算出一次梯度即更新權重，至於樣本的選取則是採用隨機抽取的方式。
- 但是一筆一筆更新，時間上也很慢。

小批次梯度下降（Mini-Batch Gradient Descent）

- 小批次梯度下降把訓練資料集隨機拆成很多小份（每次多筆資料），
  分批進行訓練。

- 小批次梯度下降的好處

  相較於 SGD：一個epoch的執行時間比較快

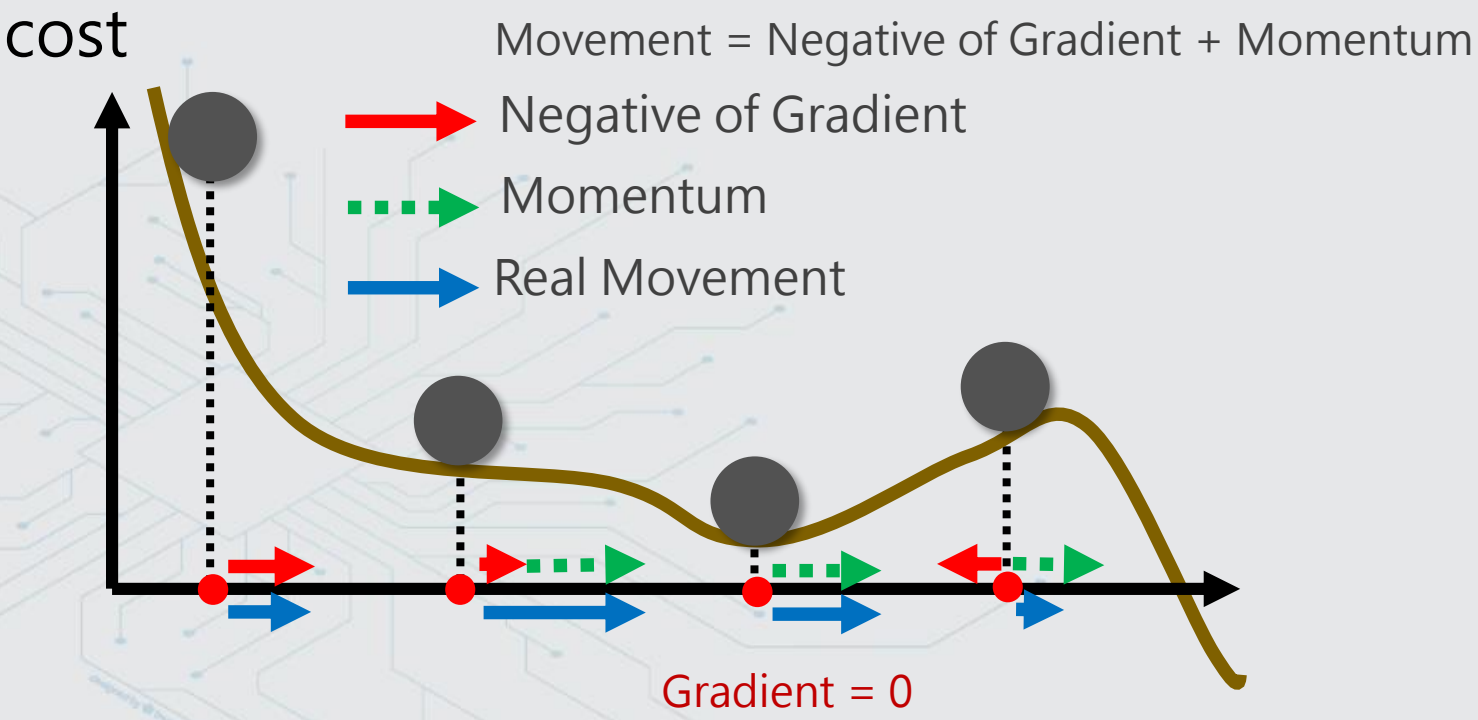
  相較於 GD：收斂較快

- 如何設定 batch size？

  不要設太大，常用 32, 128, 256, 1024, 2048

動量法（Momentum）

- 動量法是梯度下降法的變形。
- 如果當下梯度方向和歷史權重更新的方向一致，則增強此方向的梯度，否則梯度會衰退。



優化器（Optimizer）

- 優化器是使用數值方法，在不斷的批次訓練中更新**權重**（weight）和**偏差**（bias），使**損失函數**（loss function）的誤差值最小化。

常用的優化器（Optimizer）

- SGD - Stochastic Gradient Descent

- AdaGrad - Adaptive Gradient Methods

  ① AdaGrad的學習率不設置固定的值，每次迭代過程中，每個參數優化時使用不同的學習率。

常用的優化器（Optimizer）

- **RMSprop – Root Mean Square Propagation**
  - ① 是由 Geoffrey Hinton 提出的一種自適應學習率方法。
  - ② 為了解決AdaGrad學習率急劇下降的問題，此概念是將梯度除以過往梯度的均方根。
  - ③ 目前公認最好的優化器之一

- **Adam – Adaptive Moment Estimation**
  - ① 類似於RMSprop + Momentum
  - ② 目前最常用的優化器之一