# 第四周

第四周課程主要敘述訓練神經網路的一些方法、細節、演算法等。

1.損失函數(Loss function)

用來衡量機器學習成果的好壞，並以此為參考調整機器學習的過程、參數等。　　可以理解為評分標準。

損失涵數的類型，大致可以分成用來評估回歸問題和用來評估分類問題的損失函數：

①回歸問題

回歸問題的標籤和輸出皆為實數，因此損失函數主要計算標籤和輸出結果的差異。

1.MSE(離均差)

公式如下，假設有 n 筆資料：

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

因 MSE 的計算量太大，模型不好。

2.MAE(平均絕對值誤差)

公式如下，假設有 n 筆資料：

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

優點：計算量較少，且較 MSE 更接近線性。

缺點：更新參數時，無論 loss 的值大小，MAE 梯度始終相同。導致在接近最佳解時，常常進不到最佳解的位置。

②分類問題

在分類問題中，資料的標籤常用 one-hot vector(以 0 和 1 區分類別)；輸出則是將實數轉成機率分布。我們常用交叉熵(cross-entropy)來計算。

*熵(entropy)

**意即接受的所有訊息之中所包含的資訊平均量，可以解釋成用來觀看資料的亂度。**

公式如下：

$$H(X) = \sum_i -p_i log_2(p_i)$$

為何說可以用來觀察不確定度？當機率為 0.5，熵值最大，之後熵值往 0.5 的兩側下降。可以理解成：當機率特別大或特別小，我們可以往一個大方向進行預測；但機率若愈接近 0.5，我們就很難預測到底會發生甚麼，此時，不確定度就增加了。

回到交叉熵。交叉熵是將每個資料的熵算出，在全加起來。公式如下：

$$H = \sum_{c=1}^{C} \sum_{i=1}^{n} -y_{c,i} log_2(p_{c,i})$$

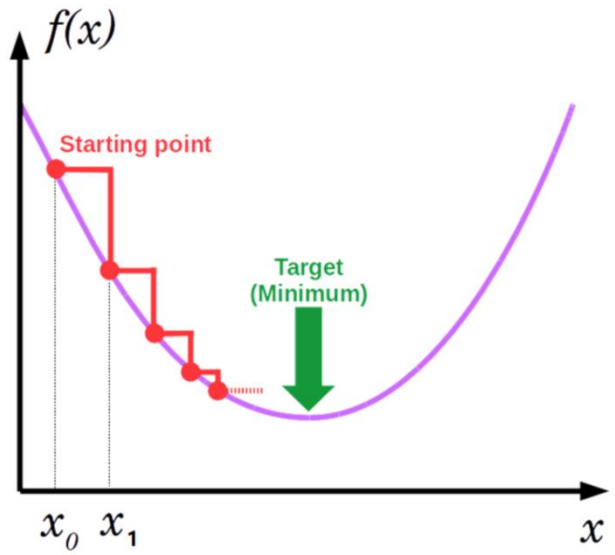
但是同樣的，我們希望不確定度愈小愈好(意即機率愈高愈好)。因此，當交叉熵算出的值愈小，愈接近我們想要的成果。

2.梯度下降演算法(Gradient descent)

我們常使用梯度下降演算法來解決最佳化問題。

Δ最佳化問題：由一個目標函數和多個限制條件組成，並追求最大或最小化的問題。機器學習的問題通常是最佳化問題。目標函數為最小化訓練誤差(讓損失函數值最小)；限制條件則為 Regularization。

①概念



如上圖，我們的目標是此函數的最低點(最小值)。因此，我們設定一起始位置，並讓機器計算 $f(x_0)$處的梯度，接著，從 $x_0$ 開始，沿著梯度的方向，根據指定的步長$\alpha$向前走一步(這裡的$\alpha$稱之為學習率，決定一次走多長)，走到的位置，為上圖的 $x_1$。以上步驟不斷重複，最後可以找到最接近最小值的點。

公式如下：

$X_{i+1} = X_i - \alpha \nabla_x f_x$ ，$\alpha$ 為學習率、$\nabla_x$為多維度向量 X 之梯度、$f_x$為純量 X 之梯度。

②梯度下降演算法的特點

1. 梯度指引方向、學習率決定要走多遠。
2. 時常會陷在局部最優，而非全局最優。

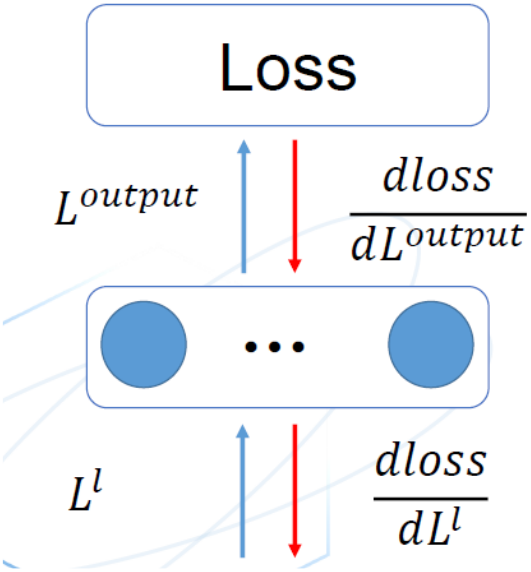

3. 計算簡單、快速(只需計算一階導數，相比於二階導數等輕鬆許多)。
4. 需自動調整學習率。

③梯度下降演算法和神經網路的訓練

1. 計算梯度

計算神經網路的梯度需要使用計算流程圖，而方法則是計算損失函數和各層、個參數的梯度後，再進行更新。

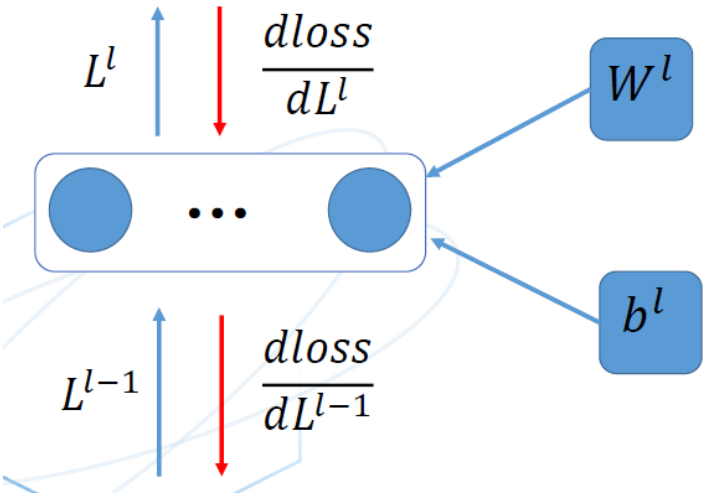
①計算損失函數對輸出層梯度




回歸問題：$\frac{\mathrm{dLoss}}{\mathrm{d}L^{output}} = L^{output} - y$ ，y 為標籤

分類問題：$\frac{\mathrm{dLoss}}{\mathrm{d}L^{output}} = P(c|x) - y$

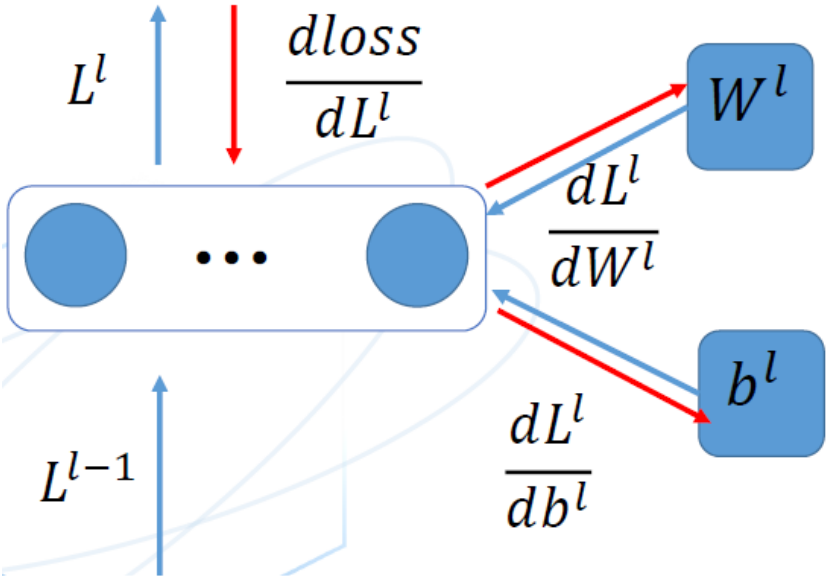
②計算損失函數對隱藏層梯度




$$\frac{\mathrm{dLoss}}{\mathrm{d}L^{l-1}} = \frac{dLoss}{dL^{l}}\frac{dL^{l}}{dZ^{l}}\frac{dZ^{l}}{dL^{l-1}}\text{，}Z^{l}=L^{l-1}W^{l}+b^{l}\text{，}L^{l}=g(Z^{l})$$

$$= \frac{dLoss}{dL^{l}} \times g'(Z){W^{l}}^{T} \text{，} {W^{l}}^{T} \text{為轉置矩陣}$$

③計算隱藏層參數的梯度



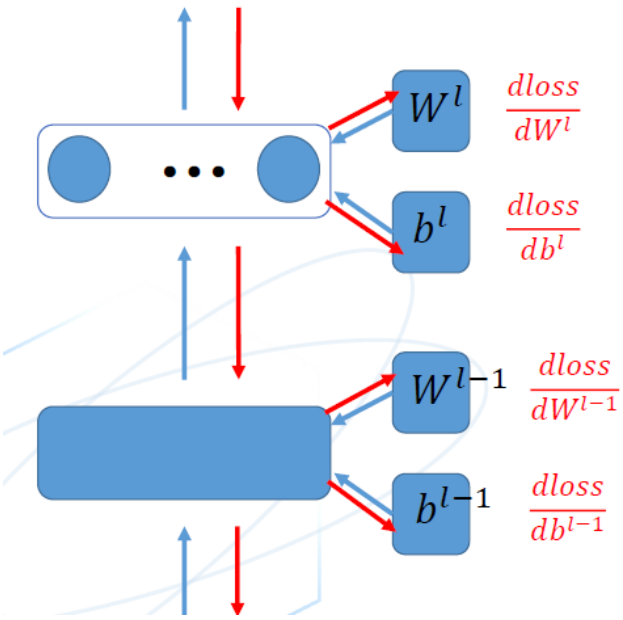

$$\frac{\mathrm{dLoss}}{\mathrm{d}W^l} = \frac{dLoss}{dL^l}\frac{dL^l}{dZ^l}\frac{dZ^l}{dW^l}$$

$$= \frac{dLoss}{dL^l} \times g'(Z){L^{l-1}}^T$$

$$\frac{\mathrm{dLoss}}{\mathrm{d}b^l} = \frac{dLoss}{dL^l}\frac{dL^l}{dZ^l}\frac{dZ^l}{db^l}$$

$$= \frac{dLoss}{dL^l} \times g'(Z)$$

④更新參數



$$X_{i+1} = X_i - \alpha \begin{bmatrix} \vdots \\ \frac{\mathrm{dLoss}}{\mathrm{d}W^l} \\ \frac{\mathrm{dLoss}}{\mathrm{d}b^l} \\ \frac{\mathrm{dLoss}}{\mathrm{d}W^{l-1}} \\ \frac{\mathrm{dLoss}}{\mathrm{d}b^{l-1}} \\ \vdots \end{bmatrix}$$

以上便是整個神經網路在計算並更新參數的計算過程。

④訓練會遇到的困難

①區域最小值

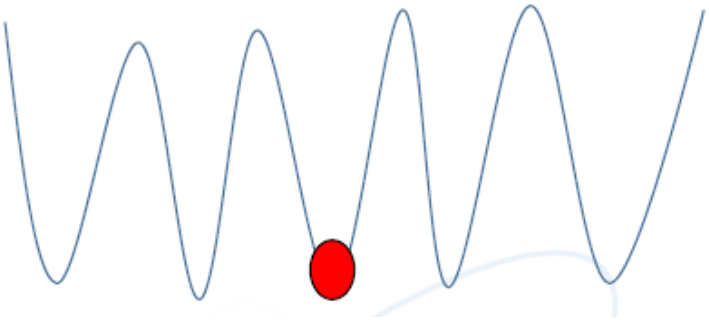

如圖所示，神經網路在訓練時常會卡在區域最小值(圖中紅點處)而非全域最小值(圖中綠點處)

解決方式有二：

1. 調整學習率

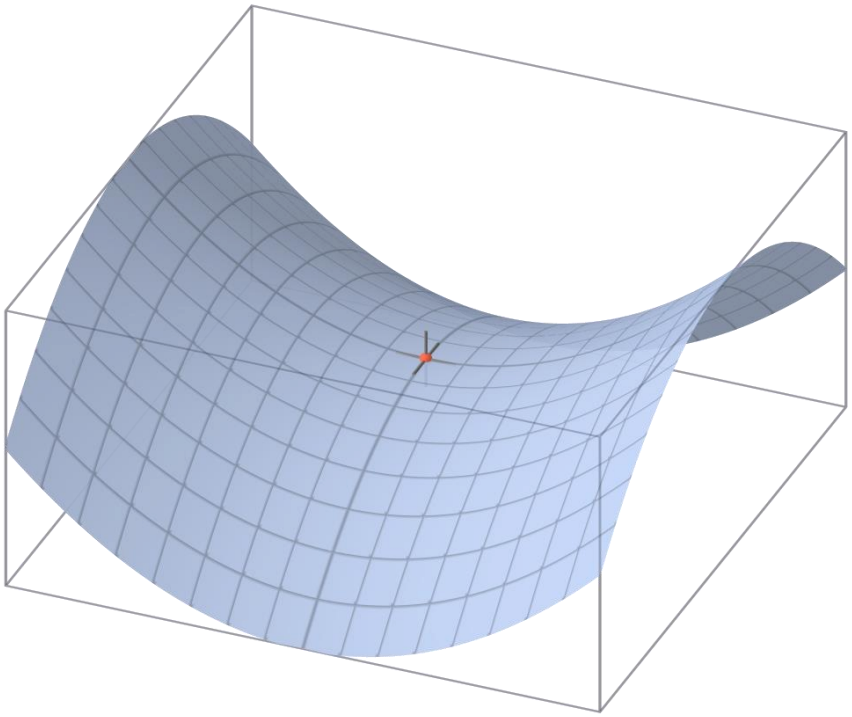
若我們嘗試調整學習率，避開局部最小值的位置，或許就有機會進到全域最小值

2. 增加模型規模



若我們增加模型規模，則圖形如圖所示。我們就不需在意全域最小值和區域最小值的差異(因為都差不多)。

②鞍點(saddle point)



在進行梯度下進演算法時，另一個常見的問題便是學習是可能卡在鞍點(途中紅點。鞍點的一階導數為零，但卻不是屬於最大或最小值。因此，在訓練過程中，若學習結果卡在鞍點，呈現出的結果可能會差。

解決方法：使用批次梯度下降演算法(抓取一部份資料來計算)，或是故意加上一點噪音(noise)來讓其逃離鞍點。

③梯度消失(gradient vanish)

因在隱藏層進行反向傳播時，梯度的計算需要乘上一個 $g'(Z)$。因此，當 $g'(Z) \to 0$ 時，梯度也會趨近於零，造成梯度消失。梯度消失帶來的影響包括無法有效更新部分神經元參數，造成學習效果不好。

解決方法：不使用 Sigmoid 等激活函數當作隱藏層激活函數，改用 Relu 當作隱藏層激活函數。

3.其他訓練神經網路的方式

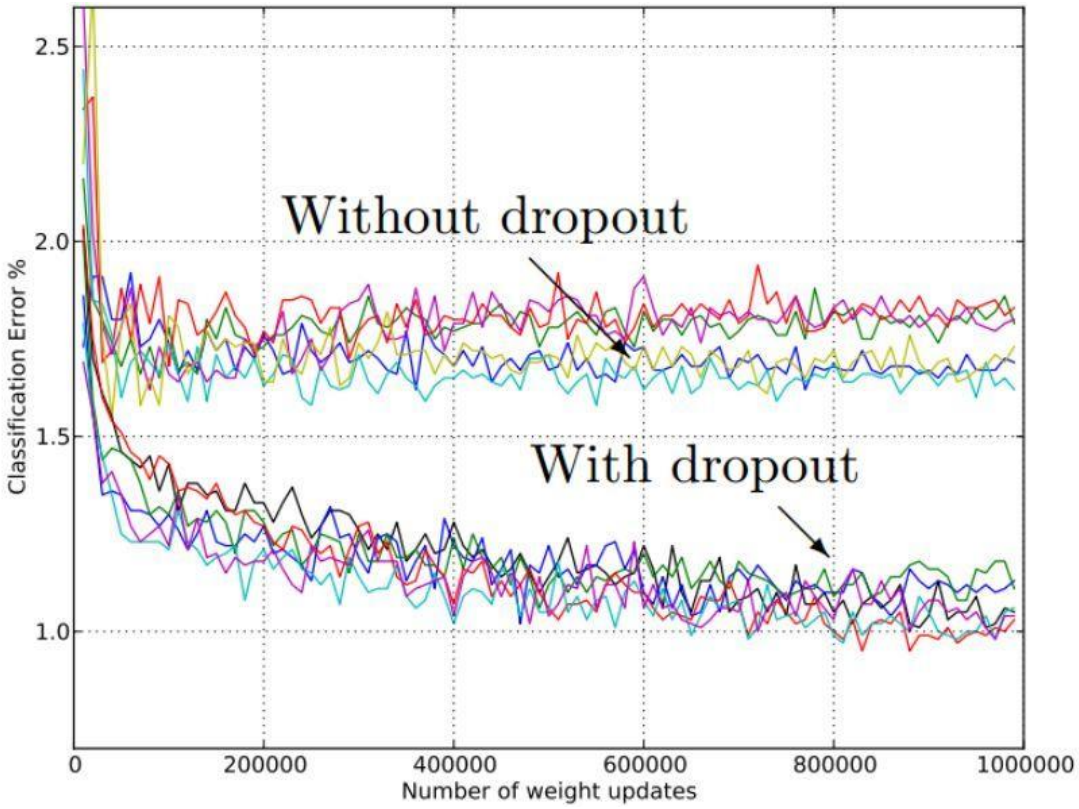
除了梯度下降演算法之外，也有一些方式，可以用來協助訓練神經網路，讓學習效果更好。

①Dropout

欲解決的問題：當模型規模太大，且資料不夠多時，時常會出現 overfitting 的情形。為了避免這種情形，我們使用 Dropout。

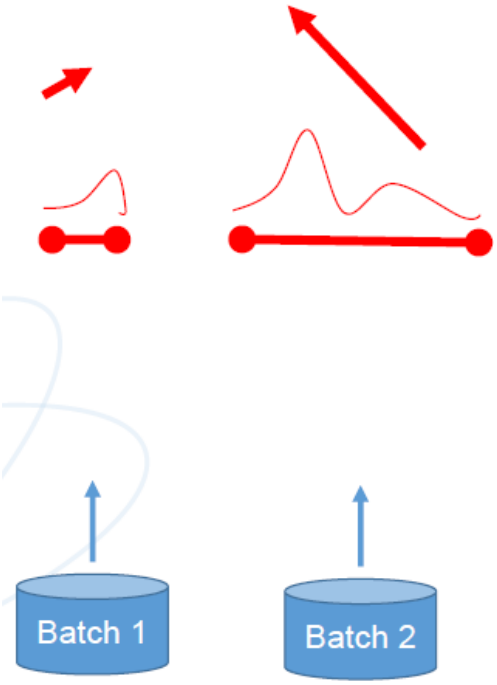
特點：每次訓練時，都隨機關閉幾個神經元，目的是減少模型的複雜度(可看成是對神經網路的 Regulation)。用這種方式訓練多次後，結果會較有 generation。

實際結果：以下圖片出自 Dropout 的原始 paper，比較使用 Dropout 前後的差異。可以看到裂使用 Dropout 後，錯誤率減低不少。




②Batch Normalization(批量標準化)

欲解決問題：有時，只抓取部分資料時，結果可能會有太大偏差，如下圖：




可以看到，Batch 1 和 Batch 2 出現的結果差異很大(可能是因數據的分布不同)，想要減少這樣的情形，我們酒可以使用，Batch Normalization。

特點：對數據做標準化，使其差距不要太大。這樣做可以加速學習(因為分布相對穩定)、減緩梯度消失(讓數據不要一直落在 Sigmoid、tahh 等激活函數的極端值，讓梯度不要太小)和可以實現 Regularation 的效果(減少資料不確定性和複雜度)。以上特點，讓 Batch Normalization 可以實現和 Dropout 等價甚至更好的效果。

# 參考資料


## 1. 機器學習每日一解(梯度下降演算法)

https://www.jianshu.com/p/62ed0793ccf3

## 2. 遞迴神經網路 RNN 、梯度下降與梯度消失

https://ithelp.ithome.com.tw/m/articles/10280019

## 3. RNN 裡面使用 dropout

https://blog.csdn.net/zhou_438/article/details/108577209