深度學習
Lab Assingment 3

1. 目的

瞭解反向傳播演算法(**Backpropagation algorithm**)如何學習多層網路的權重(Weights)和偏差值(Biases)；瞭解超級參數(Hyperparameters)，例如：隱藏神經元層數/個數和學習率，如何改變反向傳播演算法的性能。

2. 實驗進行步驟

2.1. 自行撰寫 **Backpropagation algorithm**，限定使用 Python 程式語言，不可使用套裝軟體現成程式。

2.2. 本程式作業隱藏層及輸出層神經元之激活函數(Activation Function)均採用 Sigmoid Function。

2.3. 本程式作業請使用**二元**交叉熵為輸出層神經元**的**損失函數交叉熵，即將每個輸出神經元的損失函數為:

$$-(y \log a + (1-y) \log(1-a))$$ ，其中 y 為標籤，a 為神經元輸出值。

※ 請注意講義"Backpropagation Algorithm"中，輸出層神經元**的**損失函數為方差。

2.4. 輸入資料

本實驗採用 **MNIST** 數字數據集，僅使用部分資料，辨識 0, 1, 2 三個數字。輸入資料為 784 維，助教將準備具有類別資料 8000 筆，提供同學訓練及驗證，另有 2000 筆無類別測試資料，請同學預測。

2.5. 輸出資料

當程序停止時，顯示隱藏神經元的層數/個數、學習率、世代(epoch)數、訓練準確率、驗證準確率、測試資料預測結果等。

※ 請依照助教指示之輸入/輸出格式要求

2.6. 實驗

- 建立一個多層神經元網路(輸入層–隱藏層–輸出層)，輸入層包含 784 個節點，而輸出層包含三個神經元。另外， 從實驗中找到適當數量的隱藏神經元層數/個數。
- 以上實驗，以不同的學習率重新進行實驗。

3. 說明

3.1 參數/超參數(Parameter/Hyperparameter)區別：

- 參數值是經由學習演算法訓練所得出，例如權重和偏差值(Weights and Biases)。
- 超參數是在學習演算法過程中，必需先設置的參數值。例如：學習率，隱藏層/輸出層的神經元個數等。
- 超參數協助學習演算法找到適當或最佳參數值。

3.2 One-Hot Encoding:

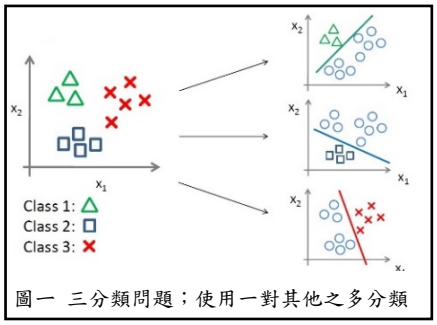
- 機器學習分類問題的標籤，常將類別以 One-Hot Vector 表示，即向量分量僅有一個維度的值是 1，其餘爲 0。例如三分類標籤第一、二、三類分別為：$\begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}$。將類別標籤轉換成 One-Hot Vector 的過程則稱 One-Hot Encoding。
- 本程式請採用 One-Hot Encoding。

3.3 訓練/驗證/測試準確率：

- 反向傳播演算法停止訓練後，固定類神經網路模型的權重和偏差值，計算每一筆資料的輸出值向量，決定其分類。類別之認定，取最大輸出值分量為其類別。例如：輸出層之值為 $\mathbf{a}=\begin{bmatrix}0.6\\0.8\\0.1\end{bmatrix}$，則認為此筆資料為第二類，若資料標籤 $\boldsymbol{y}=\begin{bmatrix}0\\1\\0\end{bmatrix}$，也為第二類，則這筆資料視為正確，否則為錯誤。如果，如果訓練集有 1000 筆，其中 900 筆正確，則訓練準確率 = 900/1000 = 90%。驗證/測試準確率也是相同計算方法，即正確筆數與驗證/測試集筆數的比率。

3.4 一對其他之多分類(One-Vs-Rest for Multi-Class Classification):

- 任何二分類演算法(例如：Perceptron Learning Algorithm 或 Logistic Regression 等方法)，可以擴展它為多分類演算法，此分類法稱之為: 一對其他之多分類(One-Vs-Rest for Multi-Class Classification)，也有人稱為: 一對全部之多分類(One-Vs-All for Multi-Class Classification)。
- 以三分類問題為例，針對每個類別訓練一個單獨的二分類模型，以本實驗而言，輸出層第一個神經元，作為辨別第一類 “0”與非“0”的二分類問題。同理，第二個神經元，辨別“1”與非“1”。第三個神經元，辨別“2”非“2”。換言之，有三個獨立的二分類模型(見圖一示意圖)。



圖一 三分類問題；使用一對其他之多分類

3.5 停止條件通常包括：

- 超過最大世代數時停止。
- 當訓練集上某些錯誤度量的平均值足夠小時停止，例如平均交叉熵，均方根誤差，平均絕對誤差等。
- 當世代數增加，雖然訓練資料集準確率上升，而未參與訓練的驗證集準確率卻下降，此時可停止訓練。其功用為檢視是否有過度訓練(Over Training)而造成過度合適(Over Fitting)的問題。

3.6 **Stochastic Backpropagation** 演算法

// Use sigmoid neurons in hidden layers and the output layer
// For each output neuron, use the binary cross-entropy as the loss function
Initialize all network weights/biases to small random numbers
UNTIL one of the termination conditions is met, DO
FOR each (**x**, **y**) in the training dataset, DO

1. **Feedforward:**

// Compute the output for each neuron in the network
Input the instance **x (= a**^{0}**)**
For each $l = 1, 2, \ldots, L$
Compute $\mathbf{n}^l = \mathbf{W}^l \mathbf{a}^{l-1} + \mathbf{b}^l$ and $\boldsymbol{a}^l = \boldsymbol{\sigma}(\mathbf{n}^l)$

2. **Backward:**

Step 2.1 Calculate the error vector for the output layer:
$\Delta^L = (\mathbf{a}^L - \mathbf{y}^L)$
Step 2.2 Backpropagate the error for each hidden layer
For each $l = L-1, L-2, \ldots, 1$
Compute the error vector at layer l:
$\Delta^l = ((\mathbf{W}^{l+1})^T \Delta^{l+1}) \odot [\mathbf{a}^l(1 - \mathbf{a}^l)]$
Step 2.3 Update all of weight and bias values
For each $l = 1, 2, \ldots, L$
Compute $\mathbf{W}^l = \mathbf{W}^l - \eta \Delta^l (\mathbf{a}^{l-1})^T$ and $\mathbf{b}^l = \mathbf{b}^l - \eta \Delta^l$

4. 實驗討論(額外加分；最多 20 分)

- 那一種神經網路架構（即不同層數/數量的神經元）獲得了最佳訓練/驗證準確率。分析並解釋你的觀察。
- 比較不同學習率的表現。
- 使用表格或圖表總結實驗結果，加以討論和分析。
- 其他心得討論報告。

5. Softmax Regression（額外加分；最多 30 分）

將輸出層改為 Softmax Layer，類別採用 One-Hot Encoding，損失函數使用交叉熵: $-\sum_{j=1}^{K} y_j \log a_j$，其中$y_j$為標籤，$a_j$為神經元輸出值，$K$ 為類別個數。在報告中推導 Softmax 的隨機梯度下降法，並撰寫程式，重新進行實驗，比較其性能是否優於 3.4 的 One-Vs-Rest for Multi-Class Classification？ 討論並解釋你的觀察。

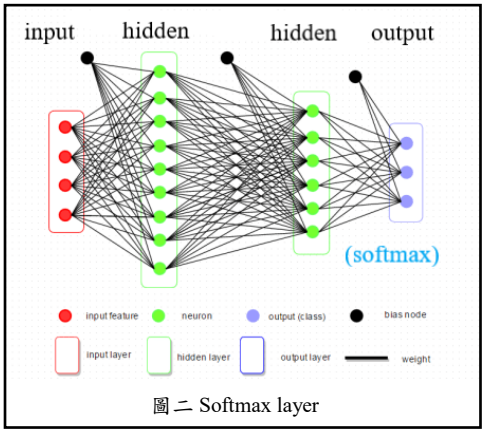
Softmax Regression 是 Logistic Regression 的推廣，可用於多分類。

- 首先，回想 Logistic Regression，使用 Sigmoid Function 為激活函數(Activation Function)於單個輸出神經元，即 $a=\sigma(n)$，其中 n 為淨輸入(Net Input)。我們知道 $0<\sigma(n)<1$，若 $\sigma(n)=0.8$，可表示兩個類別的概率分別為 80%及 20%。顯然，這兩個類別的概率總和為 100%。

- Softmax Regression 將 Logistic Regression 擴展到多分類問題。換言之，Softmax Regression 在多分類問題中，為每個類別給定一個概率，並且令這些類別概率的總和必須為 100%。

- 有別於其他激活函數，作用於隱藏層或輸出層的任何一個神經元。激活函數 Softmax Function 僅可使用於輸出層，並且作用於輸出層所有神經元，因此稱為 Softmax Layer。

- Softmax Function 作用於輸出層的淨輸入向量，即 $\mathbf{n}^{(L)}=\mathbf{W}^{(L)}\,\mathbf{a}^{(L-1)}+\mathbf{b}^{(L)}$，其中 L 為輸出層。假設我們處理 K 個類別的多分類問題，Softmax Function 令第 j 個輸出神經元為：$a_j^{(L)}=\frac{e^{n_j^{(L)}}}{\sum_{i=1}^{K} e^{n_i^{(L)}}}$。



圖二 Softmax layer

- 例如：若圖二輸出層淨輸入向量

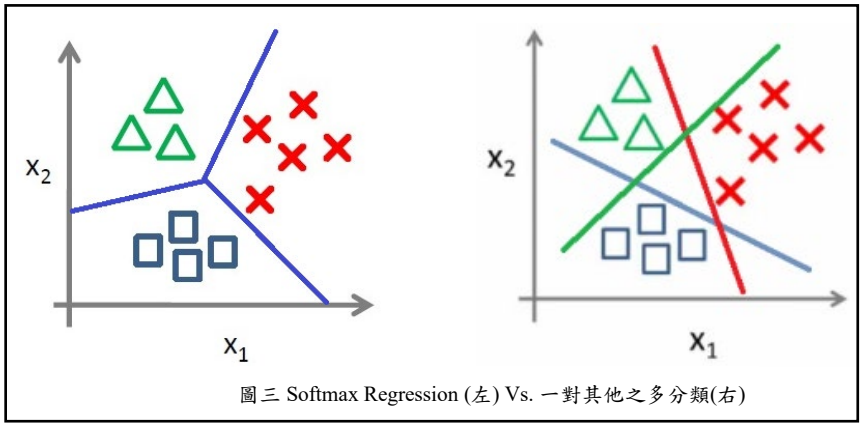
$\mathbf{n}^{(3)}=\begin{bmatrix}3\\-1\\5\end{bmatrix}$, 則 $\sum_{i=1}^{3} e^{n_i^{(3)}}=e^3+e^{-1}+e^5$

$=20.09+0.37+148.41=168.87$

而 $\mathbf{a}^{(3)}=\begin{bmatrix}\frac{20.09}{168.87}\\\frac{0.37}{168.87}\\\frac{148.41}{168.87}\end{bmatrix}=\begin{bmatrix}0.119\\0.002\\0.879\end{bmatrix}$

- Softmax Layer 損失函數使用交叉熵: $-\sum_{j=1}^{K} y_j \log a_j$，其中 y_j 為標籤，a_j 為神經元輸出值，K 為類別個數。

- 圖三左圖為三分類 Softmax Regression 示意圖；圖三右圖為三分類一對其他之多分類示意圖，即圖一中三個獨立二分類



圖三 Softmax Regression (左) Vs. 一對其他之多分類(右)