

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每個輸出層神經元的損失函數：

$$-(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

，其中 y 為標籤， \hat{y} 為神經元輸出值。

2.4. 輸入資料

本實驗採用含有 10 個類別的 **Fashion-MNIST** 數據集，每筆資料為 28x28 灰階圖像 (即 784 維輸入屬性)，共有 60,000 筆訓練資料和 10,000 筆測試資料。

本作業僅採用部分資料：

- 請同學辨識以下四種類別：
 - label 0 T-shirt/Top
 - label 3 Dress
 - label 8 Bag
 - label 9 Ankle boot
- 助教準備具有類別資料 16,000 筆 (train.csv)，提供同學訓練及驗證
- 另有 4,000 筆無類別測試資料(test.csv)，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，而輸出層包含四個神經元。另外，從實驗中找到適當數量的隱藏神經元層數及其個數。
- 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，其餘為 0。例如四分類標籤第一、二、三、四類分別為：

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}。$$

- 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

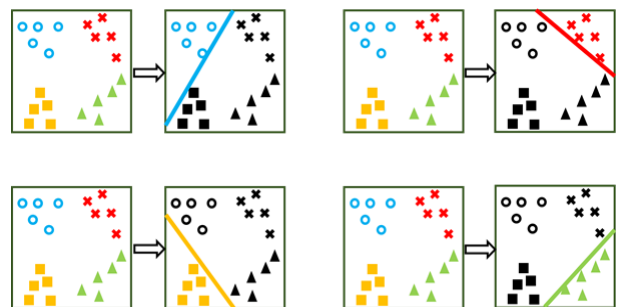
類別。例如：輸出層之值為 $\hat{y} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0.3 \\ 0.1 \end{bmatrix}$ ，則認為此筆資料為第二類，若資料標

籤 $y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ ，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 1,000 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)。
- 以四分類問題為例，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“T-shirt/Top”與“非 T-shirt/Top”的二分類問題。同理，第二個神經元，辨別“Dress”與“非 Dress”，第三個神經元，辨別“Bag”與“非 Bag”，第四個神經元，辨別“Ankle boot”與“非 Ankle boot”。換言之，有四個獨立的二分類模型(見圖一示意圖)。



圖一 四分類問題；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  $\mathbf{x}$  ( $= \mathbf{a}^0$ )
        For each  $l = 1, 2, \dots, L$ 
          Compute  $\mathbf{n}^l = \mathbf{W}^l \mathbf{a}^{l-1} + \mathbf{b}^l$  and  $\mathbf{a}^l = \sigma(\mathbf{n}^l)$ 

    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  $l = L-1, L-2, \dots, 1$ 
            Compute the error vector at layer  $l$ :
               $\Delta^l = ((\mathbf{W}^{l+1})^T \Delta^{l+1}) \odot [\mathbf{a}^l \odot (1 - \mathbf{a}^l)]$ 
              //  $\odot$ : Element-wise product
              // (also known as the Hadamard product)
          Step 2.3 Update all of weight and bias values
            For each  $l = 1, 2, \dots, L$ 
              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。