

# 深度學習 Lab3.

## 1 繳交注意事項

作業繳交期限為 12/11 23:59

繳交規定及評分方式如下，請符合以下要求，否則將斟酌扣分。

### 1.1 繳交檔案要求

輸出結果檔案格式為 json，圖片存成 png。

資料夾名稱及結構如下，並以 zip 壓縮後進行上傳：

- 班級 \_ 學號 \_ 姓名 \_ 深度學習 \_Lab3 [文字與底線之間無空格]
  - Lab3
    - \* Group\_B\_train.csv [請勿上傳]
    - \* Group\_B\_test.csv [請勿上傳]
    - \* main.py
    - \* output
      - output\_accuracy.png
      - output\_loss.png
      - output.json
      - test\_set\_prediction.json
  - readme.txt
- 班級 \_ 學號 \_ 姓名 \_ 深度學習 \_Lab3\_ 實驗心得報告.pdf [文字與底線之間無空格]

繳交檔案說明：

以下數值及圖片僅供參考用，並非為正確答案。

json 請保持相同型態，如純量與 list 等，圖片請保持要求之 title、label 名稱、X、Y 軸 label

- output\_accuracy.png (對應 Fig.1)
  - 包含每個 epoch 的訓練及驗證準確率
  - label 名稱為 Train accuracy 與 Validation accuracy
  - title 為 GroupB\_Accuracy
  - X 軸 label 為 Epoch，Y 軸 label 為 Accuracy
- output\_loss.png (對應 Fig.2)
  - 包含每個 epoch 的訓練及驗證 loss
  - label 名稱為 Train loss 與 Validation loss
  - title 為 GroupB\_Loss
  - X 軸 label 為 Epoch，Y 軸 label 為 Loss

- 輸出 output.json 檔結構圖如下，可以不用排版，注意 key 的名稱要相同。冒號後方為同學的參數與結果。

```
{
    "Learning rate": 1e-3,
    "Epoch": 500,
    "Batch size": 64,
    "Final train accuracy": 0.93...,
    "Validation accuracy": 0.91...,
    "Final train loss": 0.23...,
    "Final validation loss": 0.34...
}
```

- 輸出 test\_set\_prediction.json 檔結構圖如下，可以不用排版，注意 key 的名稱要相同。冒號後方為同學的預測結果(另有輸出內容說明，請至1.4節查看)。

```
{
    "Predictions": [
        0,
        0,
        2,
        2,
        4,
        ...
    ]
}
```

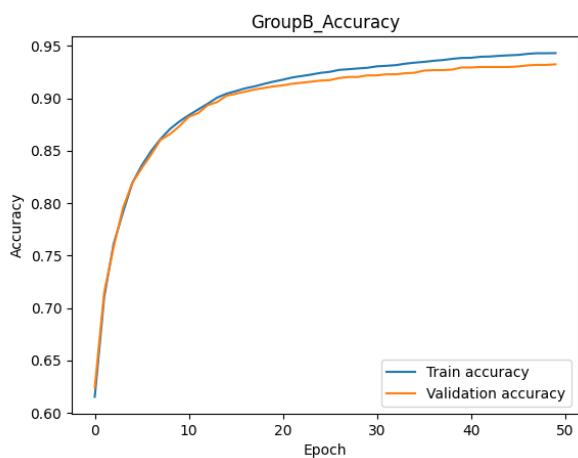


Figure 1: 每個 epoch 的訓練與驗證準確率

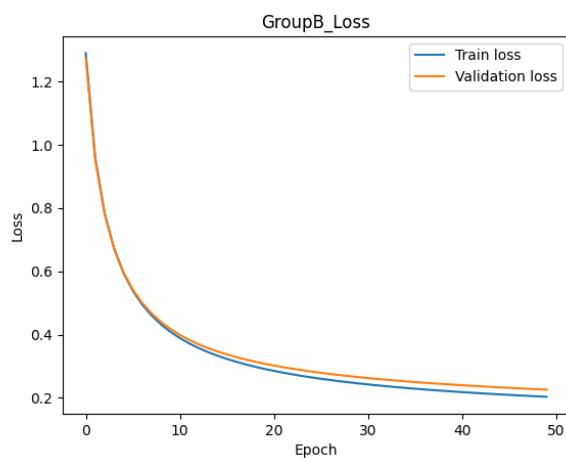


Figure 2: 每個 epoch 的訓練與驗證 loss

當中 output 資料夾中的檔案需要透過 main.py 當中創建。若有需要額外說明事宜，可另行付上 readme.txt。

請勿上傳輸入資料(如 train data)，但是程式中請保持使用相對路徑，讀資料時需假設所有輸入資料與 main.py 同級，若批改作業時程式讀取路徑錯誤將不予給分。

請保持執行 python 的路徑為 main.py 的同級資料夾(即使用 python main.py，就能夠執行)，python 中所謂的相對路徑是相對於執行 python 的路徑，因此不同路徑執行相同的 python 檔案可能會導致相對路徑錯誤。若有此類情況，將會扣分。

有使用 AI 查詢之處務必寫入 `readme.txt`。例如：查詢 ChatGPT `numpy random` 怎麼使用、矩陣乘法原本寫 `A*B`，但 ChatGPT 告訴我要使用 `A@B` 等。此 `readme.txt` 內容並不會列入計分，主要是紀錄同學詢問內容及學習過程，因此希望有需要的同學能認真撰寫。

## 1.2 程式碼要求

- 為確保實驗可重現性，每份作業請設定亂數種子，如 `np.random.seed(seed=42)`
- 請詳細撰寫程式碼，並在每一行加上註解，助教才能確定同學是否理解
- 根據各題提供的提示與要求，完成對應的程式內容
- 命名要具有可讀性，避免使用過於簡單或不具描述性的變數或函數名稱，例如 `a`、`los` 等
- 請勿將參數直接寫死，如沒有 Epoch 相關變數，直接在對應位置寫下數值
- 請使用程式處理 `label`、`label encoding`、`one-hot encoding`，如直接寫死 (`dict`) 或單純 `if else` 對應將給予扣分
- 僅可使用 `python` 內建函式庫、`numpy`、`matplotlib`
- Python 版本推薦使用 3.12.11，若因版本導致語法不可使用將不予計分

請同學 **不要使用 AI 工具生成作業**。學習過程中，比起單純追求高分，更重要的是理解與掌握程式設計概念。有疑問時，可利用 AI 工具輔助、線上資源或詢問助教來協助學習。

## 1.3 評分方式

本作業分為兩部分 (主要作業與實驗報告)：

- Fashion-MNIST 資料集分類：使用多層神經網路與不同激活函數。
- 實驗報告討論 (須包含必要實驗項目)(完整說明請看 Lab3 題目說明)：
  - 分析單層與多層的性能差異
  - 激活函數的比較
  - 網路結構的影響
  - 其他超參數的比較
  - 其他心得討論

完成情形	評分方式	說明
僅完成程式作業	最高 100 分	使用多層網路與激活函數解決分類任務，同時考驗最終分類準確度
完成實驗討論報告	最高額外加 20 分	使用圖表進行整理與比較，報告越完整、比較越詳細，加越多分數。 <b>請注意實驗時的控制變因與操作變因，若實驗方式有誤將不予以加分。</b>

Table 1: 作業完成標準與評分大綱

本次作業中，評分將依下列項目進行：

1. 正確性 (工具應用、題目要求等) (40%)

訓練模型時，可於以下情況停止：

- 設定最大的世代數 (epoch)。此條件為避免無窮迴圈，不可視為找到合理模型。
- 訓練損失降至足夠小或不再改善。
- 驗證準確率下降而訓練準確率上升 (出現 overfitting)。

請同學設計上述停止條件，不須全數使用，但不可只使用最大世代數作為停止條件。  
(實驗報告除外，同學可設計停止條件作為實驗的控制變因)

2. 邏輯性 (解法思路等) (40%)

3. 模型準確度 (訓練與驗證結果表現是否合理) (10%)

4. 變數名稱、排版、註解內容、檔案名稱 (10%)

本次準確率將列入評分依據，請同學自行判斷模型最佳程度並保留相關參數於 main.py 之中，助教將會以 main.py 中參數的表現作為本次準確率之成績。

若程式碼無法執行，或執行結果與繳交內容差異過大，則將不予以計分。

若發現抄襲行為，抄襲者與被抄襲者皆以 0 分計算。

直接使用 AI 工具產出作業者，該份作業以 0 分計算。

遲交一日，作業分數上限扣 10 分，直到 0 分。

對於作業有疑慮之處，助教將提出疑問，請同學解釋。

## 1.4 作業補充說明

- 作業提供的 Group\_B\_train.csv 包含同學訓練時應使用的資料 (有 label)，請同學自行分成 train set 以及 validation set。另外資料並沒有打亂順序，請同學自行打亂。
- 作業提供的 Group\_B\_test.csv 包含同學訓練時應使用的資料 (無 label)，同學應利用訓練完的模型預測這些資料，並且輸出到檔案中。助教將會對預測結果進行準確率評分，同學請勿改變測試資料的輸入與輸出順序。另外由於本次作業同學需要利用到 one-hot encoding，在 one-hot encoding 之前會需要做 label encoding (也就是 0 2 4 6 轉換成 0 1 2 3)，為了避免每個編碼方式不同，請同學輸出 0 2 4 6 等原本資料的 label。