Homework-1 (Chapter 3. Classification)

110321006 蔡秉翰

110321007 劉竑毅

為了使用 SGDClassifier 進行 MNIST 數據集的多類別分類，首先我們使用 GOOGLE 的 COLAB 開啟了專案，下載了 MINST 的資料集，並進行資料的初步整理，將 7000 筆資料分成了 6000 和 1000，其中 6000 用來訓練模型，1000 用來測試，先用 SGDClassifier **訓練出模型後，**使用交叉驗證評估模型性能（cv=3）來測量準確性，最後在 1000 筆資料的測試集上進行測試得到了以下結果

```
Cross-Validation Scores: [0.87365 0.85835 0.8689 ]
Mean Accuracy: 0.8669666666666668
Test Set Accuracy: 0.874
```

訓練集平均交叉驗證和測驗集準確率差不多，可能代表資料平均，訓練得很好，測試集準確率也有到 87，有機會可以更好，以下是 confusion matrix

```
Confusion Matrix:
[[ 902    0    8   11    1   13    2    4   39    0]
 [   0 1095    2    3    0    2    4    1   28    0]
 [   1   10  803   69    6    4    4   10  122    3]
 [   0    1    6  931    1   21    3    7   35    5]
 [   2    2    9   15  778    4    2    9   62   99]
 [   6    2    1   71    3  709   12   12   67    9]
 [   5    3   12   13    5   21  854    0   45    0]
 [   0    3   18   20    3    4    1  919   18   42]
 [   3    5    2   30    4   43    5    5  872    5]
 [   3    5    2   33    7    5    0   20   57  877]]
```

感覺是 3 和 8 很容易誤判，可能需要修正資料集的錯誤

到了第二小題，我們使用人工成長訓練集來測試看看是否可以提高準確性，我們將 6000 筆用來訓練的資料，利用系統的 scipy.ndimage.interpolation 內的 shift 函數位移了上左右下各一份副本然後合併到新的訓練集，標籤也是放入新的，依舊用 1000 個原始資料做測試，然而我忘了將原始的訓練資料也放入，訓練一次模型要很久，尤其是交叉驗證，所以我放棄加入原始資料的想法了，訓練集維持為四份位移過的副本，所以應該會有大誤差，最後，得到了以下結果

```
Cross-Validation Scores: [0.81533 0.78999 0.80644]
Mean Accuracy: 0.8039200000000001
Test Set Accuracy: 0.8696
```

這結果反而不如原始數據，所以我決定重新用原始數據加上四個副本重新訓練模型，又要消耗兩個小時，以下是 confusion matrix

```
Confusion Matrix:
[[ 968    0    0    2    0    3    5    1    1    0]
 [   0 1118    1    6    0    2    7    0    1    0]
 [  15    4  875   71    2    9   25    5   22    4]
 [   7    3   21  891    1   41    8   16   16    6]
 [   6    7   19    2  811   11   20    5    4   97]
 [  12   10    9   79    4  731   17    6   19    5]
 [  46    4   16    1    3   22  866    0    0    0]
 [  10   15   42    8    1    5    3  908    0   36]
 [  24   38   19   49   14   93   24   16  658   39]
 [  19   10    5   19   16   27    1   37    5  870]]
```

雖然 Test Set Accuracy 有到 86，但使還是低於 87，這不應該，可能

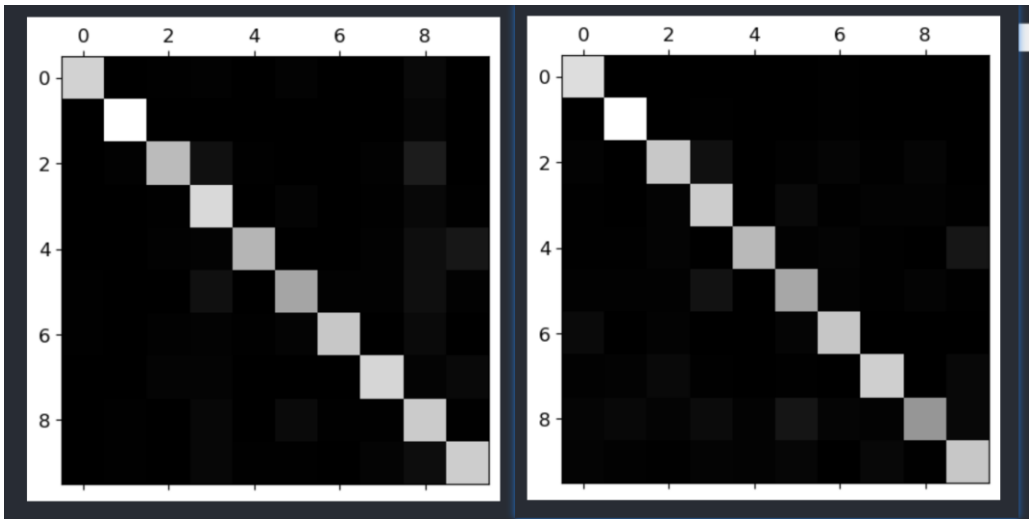
是因為我一開始忘了加原始數據導致的

又重新做了一次，將五份副本轉換為 NumPy 數組，以便進行後續的模型訓練，結果是輸出和四份副本一模一樣，我開始懷疑人生，我也不知道為什麼會一模一樣

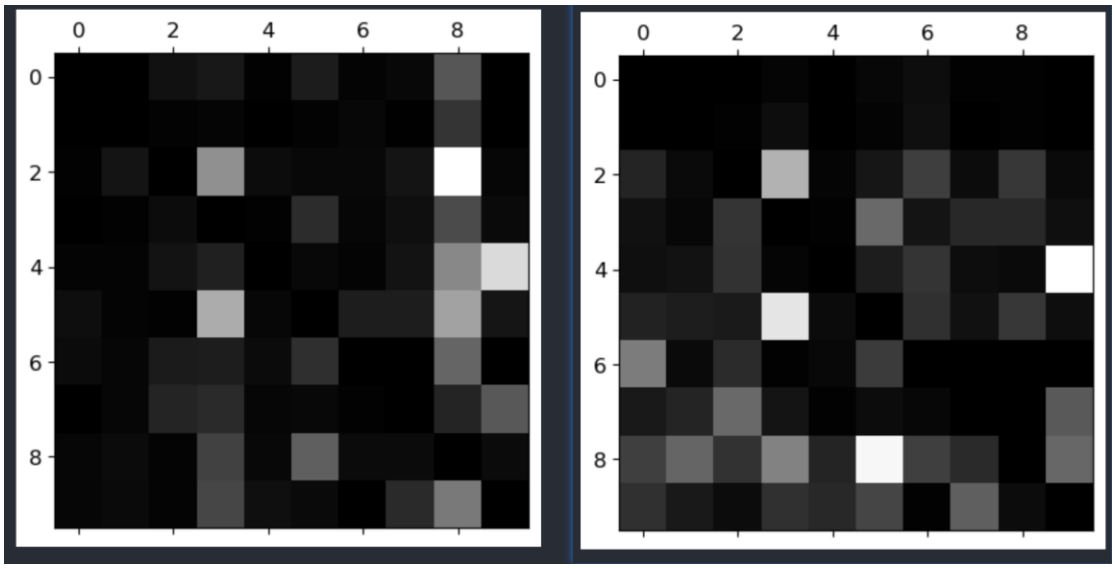
```
Cross-Validation Scores: [0.81533 0.78999 0.80644]
Mean Accuracy: 0.8039200000000001
Test Set Accuracy: 0.8696
```

得到的 MATRIX

```
Confusion Matrix:
[[ 968    0    0    2    0    3    5    1    1    0]
 [   0 1118    1    6    0    2    7    0    1    0]
 [  15    4  875   71    2    9   25    5   22    4]
 [   7    3   21  891    1   41    8   16   16    6]
 [   6    7   19    2  811   11   20    5    4   97]
 [  12   10    9   79    4  731   17    6   19    5]
 [  46    4   16    1    3   22  866    0    0    0]
 [  10   15   42    8    1    5    3  908    0   36]
 [  24   38   19   49   14   93   24   16  658   39]
 [  19   10    5   19   16   27    1   37    5  870]]
```



左邊是原始的，右邊是資料加強後的，加強資料後的看起來比較好，但是數值上準確率比較差，僅限於我的實驗結果



左邊是原始的，右邊是資料加強後的

SGDClassifier 的標準化或超參數調整 ，可以讓模型的準確率上升，例如 Standard Scaler ()、選擇適當的損失函數、調整學習率和正則化、使用交叉驗證來評估模型性能，並選擇最佳的超參數組合，但因為時間有限我沒有做，第一題就到此為止。

使用 SGDClassifier 進行 Fashion MNIST 數據集的多類別分類，首先我們使用 GOOGLE 的 COLAB 開啟了專案，下載了 Fashion MNIST 的資料集，並進行資料的初步整理，將 7000 筆資料分成了 6000 和 1000，其中 6000 用來訓練模型，1000 用來測試，先用 SGDClassifier 訓練出模型後，使用交叉驗證評估模型性能（cv=3）來測量準確性，最後在 1000 筆資料的測試集上進行測試

```
Cross-Validation Scores: [0.78315 0.81355 0.82255]
Mean Accuracy: 0.8064166666666667
Test Set Accuracy: 0.8014
```

訓練集平均交叉驗證和測驗集準確率幾乎一致，訓練得很好，相較於一開始的 MNIST 數據集，Fashion MNIST 的 Cross-Validation Scores 三項都比較低，感覺是圖案比較複雜，性能還可以，但可以再加強，以下是 confusion matrix

```
Confusion Matrix:
[[666  12  11  70  11   0 203   0  24   3]
 [  2 959   0  24   4   0   7   1   3   0]
 [  5   8 494  11 152   0 301   0  29   0]
 [  7  33  13 833  38   0  58   1  16   1]
 [  0   3  46  30 699   0 201   0  21   0]
 [  3   3   4   0   0 816   1  66  41  66]
 [ 92   5  49  45  77   0 689   0  42   1]
 [  0   0   0   0   0  12   0 932   4  52]
 [  1   1   0   6   1   4  19   5 963   0]
 [  1   0   0   0   0   3   1  30   2 963]]
```

到了第二小題，我們使用人工成長訓練集來測試看看是否可以提高準確性，然而我的代碼和一開始一樣沒注意到，忘了加入原始數據最後，得到了以下結果

```
Cross-Validation Scores: [0.75881 0.68715 0.73772]
Mean Accuracy: 0.7278933333333333
Test Set Accuracy: 0.7365
```

這下比一開始更低了，準確率跌到剩下 73，我不太懂哪裡錯了也許是忘了加入原始數據最後，所以我也決定要重做，以下是 confusion matrix，

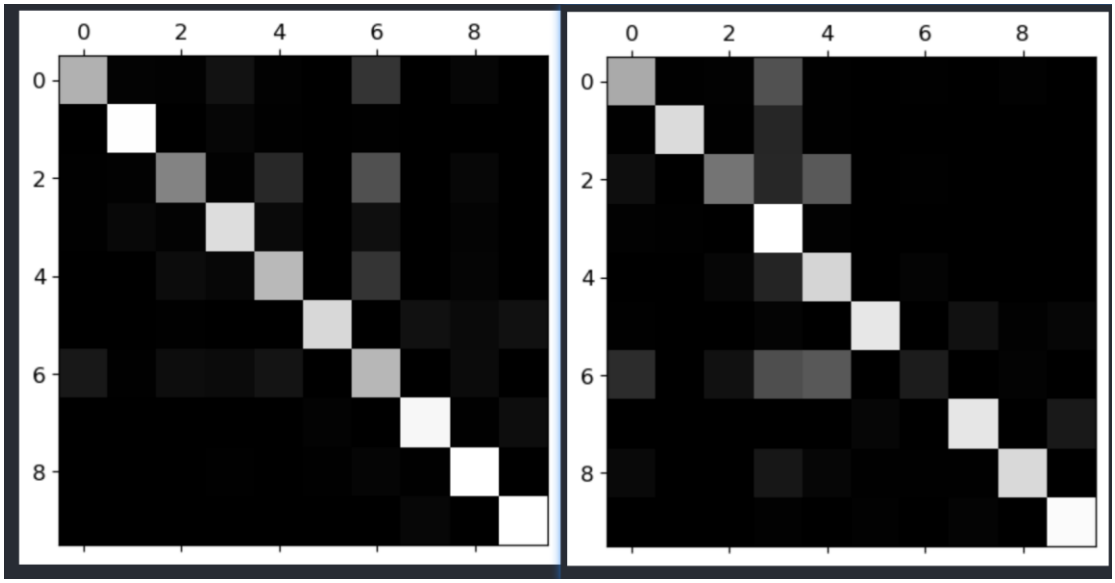
```
Confusion Matrix:
[[648   4  11 313   5   2   7   0  10   0]
 [  6 835   6 146   5   0   1   0   1   0]
 [ 55   3 443 149 344   0   6   0   0   0]
 [  8   4   1 974   8   0   3   0   2   0]
 [  3   1  24 144 812   0  16   0   0   0]
 [  4   0   0  19   0 881   0  65   8  23]
 [174   2  67 297 338   0 108   0  14   0]
 [  1   0   0   0   0  24   0 878   0  97]
 [ 38   1   1  91  25   5   4   7 828   0]
 [  2   0   1  10   0  11   0  17   1 958]]
```

重做之後發現準確率和 confusion matrix 沒變，不知道是我代碼的問題還是算法的問題，我明明確確實實的把原始參數和新家的參數放了進去，我覺得訓練集的不同應該會有差異阿

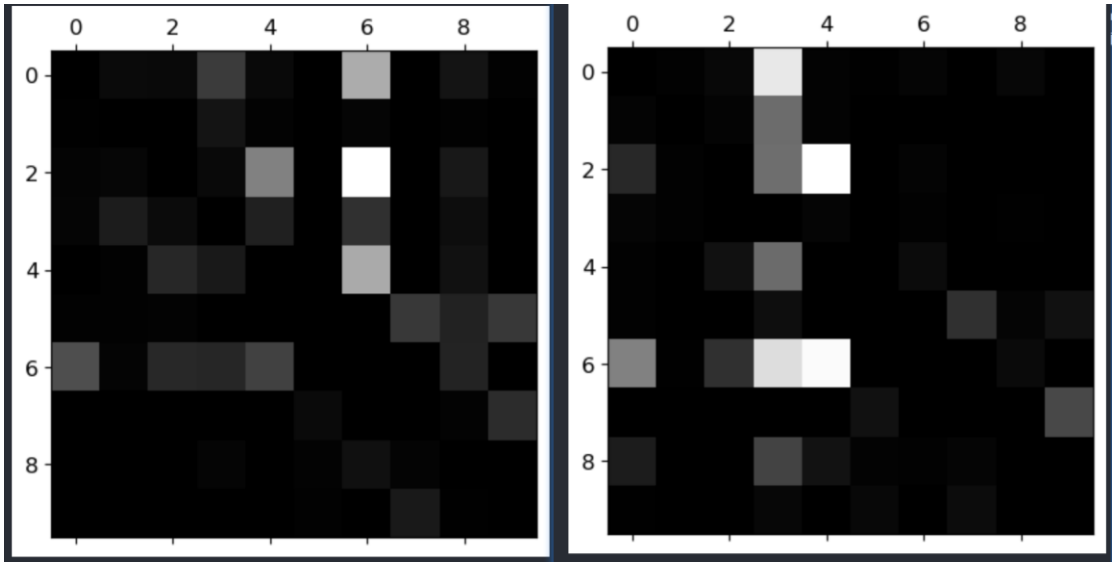
以下是重做之後發現準確率和 confusion matrix

```
Cross-Validation Scores: [0.75881 0.68715 0.73772]
Mean Accuracy: 0.7278933333333333
Test Set Accuracy: 0.7365
```

```
Confusion Matrix:
[[648   4  11 313   5   2   7   0  10   0]
 [  6 835   6 146   5   0   1   0   1   0]
 [ 55   3 443 149 344   0   6   0   0   0]
 [  8   4   1 974   8   0   3   0   2   0]
 [  3   1  24 144 812   0  16   0   0   0]
 [  4   0   0  19   0 881   0  65   8  23]
 [174   2  67 297 338   0 108   0  14   0]
 [  1   0   0   0   0  24   0 878   0  97]
 [ 38   1   1  91  25   5   4   7 828   0]
 [  2   0   1  10   0  11   0  17   1 958]]
```



左邊是原始的，右邊是資料加強後的



資料加強後的準確率甚至都是下降的，使我懷疑我的作法是錯誤的。SGDClassifier 的標準化或超參數調整 ，可以讓模型的準確率上升，但我感覺不如換一個 Classifier

以我的實驗去做結論的話，我認為 SGDClassifier 在第一個資料集上的效能比較好，在 Fashion MNIST 資料及上效果較差，而資料加強後再使用 SGDClassifier 在兩個資料集上都沒有取得更好的效果，在第一個資料集上資料加強效果較原始資料差一點，在第二個資料集上資料加強效果較原始資料差很多，我的結論是資料加強不一定帶來較好的性能，僅限於如果我的實驗是正確的，以上是功課一的報告內容，感謝觀看