Report

109511207 蔡宗儒

我設計了四層 Fully Connected layer 來做這次的作業，用的 Activation function 為 ReLu，用的 Loss function 為 Softmax，如下圖。

```
def __init__(self):
    self.fc1 = FullyConnected(28*28,256)
    self.act1 = ReLu()
    self.fc2 = FullyConnected(256,128)
    self.act2 = ReLu()
    self.fc3 = FullyConnected(128,32)
    self.act3 = ReLu()
    self.fc4 = FullyConnected(32,10)
    self.loss = SoftmaxWithloss()
```

我 train 了很多次後發現用三層的效果和五層的效果都沒有用四層的效果好，而這其中又以將節點設置成 2 的冪次為佳。至於 Activation function 我嘗試使用了 Sigmoid 跟 Tanh，但做出來的 Val Acc 都只有 80 初頭%而已，用 ReLu 和 SeLu 的效果明顯較佳。而用 ReLu 和 SeLu 的效果差不多，有些超參數用 ReLu 能有更高的 Acc，有些反而用 SeLu 能有更高的 Acc。我最後跑出最高的 Acc 是用 ReLu 做為 Activation function。

而最後我也採用了老師上課所講的 Learning Rate Decay 的方式去調整 Learning Rate，我採用的方式是每跑一個 epoch，Learning Rate 就乘上 0.9，如下圖。

```
for epoch in range(1, EPOCH+1):
    train_hit = 0
    val_hit = 0
    total_train_loss = 0
    total_val_loss = 0
    Learning_rate *= 0.9
```

而採用 Learning Rate Decay 的方式後我的 Test Acc 就從 89.0X%提升至 90.04%了，算是滿有感的提升。而我還有嘗試 step decay 的方式，然而 Test Acc 並沒有比每個 epoch 乘上 0.9 高。

至於超參數的調整方式，我是經過數次測試後發現將 EPOCH 設為 50，Batch_size 設為 10，Learning_Rate 設為 0.0006，val_image_num 設為 3000 會比其他數值有更好一點點的結果。

最後，我還有嘗試將 train data 切成 30 等份，每個 EPOCH 依序取出其中一份做為 validation data，剩下的做為 train data。我這麼做的原因是如果將 validation data 固定為原本的最後幾份 train data 的話，這樣整個網路就會有些資料學不到，所以我這麼做應該能讓我設計的網路學習到更多資料？
我實作並拿去 train 後發現 Val Acc 的確比較高，如下圖。

```
Epoch: 30 |Train Loss:  0.1115 |Train Acc:95.8439 |Val Loss:  0.1376 |Val Acc:94.9000
Epoch: 31 |Train Loss:  0.1097 |Train Acc:95.9053 |Val Loss:  0.1268 |Val Acc:95.0667
Epoch: 32 |Train Loss:  0.1069 |Train Acc:96.0772 |Val Loss:  0.1436 |Val Acc:94.2333
Epoch: 33 |Train Loss:  0.1056 |Train Acc:96.1193 |Val Loss:  0.1434 |Val Acc:94.3000
Epoch: 34 |Train Loss:  0.1037 |Train Acc:96.2140 |Val Loss:  0.1541 |Val Acc:94.2667
Epoch: 35 |Train Loss:  0.1024 |Train Acc:96.1947 |Val Loss:  0.1389 |Val Acc:94.8333
Epoch: 36 |Train Loss:  0.1012 |Train Acc:96.2842 |Val Loss:  0.1374 |Val Acc:94.1667
Epoch: 37 |Train Loss:  0.1005 |Train Acc:96.3281 |Val Loss:  0.1193 |Val Acc:95.5333
Epoch: 38 |Train Loss:  0.0996 |Train Acc:96.3579 |Val Loss:  0.1259 |Val Acc:95.1667
Epoch: 39 |Train Loss:  0.0989 |Train Acc:96.4123 |Val Loss:  0.1120 |Val Acc:95.8000
Epoch: 40 |Train Loss:  0.0980 |Train Acc:96.4702 |Val Loss:  0.1001 |Val Acc:95.9667
Epoch: 41 |Train Loss:  0.0971 |Train Acc:96.4860 |Val Loss:  0.0973 |Val Acc:96.3000
Epoch: 42 |Train Loss:  0.0965 |Train Acc:96.5035 |Val Loss:  0.0978 |Val Acc:96.8667
Epoch: 43 |Train Loss:  0.0956 |Train Acc:96.5246 |Val Loss:  0.0982 |Val Acc:96.4667
Epoch: 44 |Train Loss:  0.0949 |Train Acc:96.5456 |Val Loss:  0.0955 |Val Acc:96.6000
Epoch: 45 |Train Loss:  0.0941 |Train Acc:96.5895 |Val Loss:  0.0983 |Val Acc:96.2333
Epoch: 46 |Train Loss:  0.0935 |Train Acc:96.6158 |Val Loss:  0.1028 |Val Acc:96.0667
Epoch: 47 |Train Loss:  0.0930 |Train Acc:96.6211 |Val Loss:  0.0981 |Val Acc:96.2667
Epoch: 48 |Train Loss:  0.0930 |Train Acc:96.6456 |Val Loss:  0.0932 |Val Acc:96.2000
Epoch: 49 |Train Loss:  0.0924 |Train Acc:96.6474 |Val Loss:  0.0968 |Val Acc:96.5333
Epoch: 50 |Train Loss:  0.0919 |Train Acc:96.6789 |Val Loss:  0.0950 |Val Acc:96.5667
Epoch: 51 |Train Loss:  0.0919 |Train Acc:96.6474 |Val Loss:  0.0918 |Val Acc:96.6000
Epoch: 52 |Train Loss:  0.0910 |Train Acc:96.7070 |Val Loss:  0.1010 |Val Acc:96.0333
Epoch: 53 |Train Loss:  0.0908 |Train Acc:96.7228 |Val Loss:  0.0997 |Val Acc:96.3333
Epoch: 54 |Train Loss:  0.0900 |Train Acc:96.7456 |Val Loss:  0.1061 |Val Acc:96.2000
Epoch: 55 |Train Loss:  0.0901 |Train Acc:96.7298 |Val Loss:  0.0983 |Val Acc:96.6000
Epoch: 56 |Train Loss:  0.0900 |Train Acc:96.7702 |Val Loss:  0.0965 |Val Acc:95.8667
Epoch: 57 |Train Loss:  0.0900 |Train Acc:96.7333 |Val Loss:  0.0898 |Val Acc:96.8667
Epoch: 58 |Train Loss:  0.0894 |Train Acc:96.7702 |Val Loss:  0.0972 |Val Acc:96.4333
Epoch: 59 |Train Loss:  0.0894 |Train Acc:96.7754 |Val Loss:  0.0917 |Val Acc:96.8667
Epoch: 60 |Train Loss:  0.0894 |Train Acc:96.7877 |Val Loss:  0.0852 |Val Acc:96.8667
```

然而實際拿去 test 後發現 Test Acc 依舊只有 89%左右，並沒有顯著提升，反而還下降了。我認為這麼做只能讓他更會預測 train 的 6 萬筆資料，對於未知的 test data 沒有明顯提升。