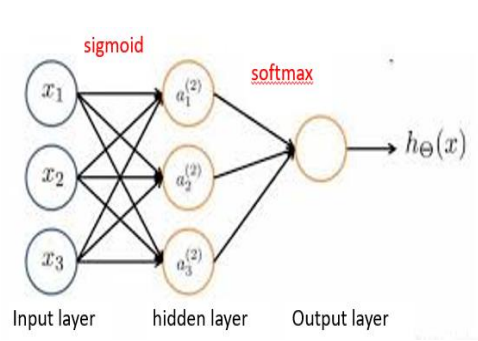
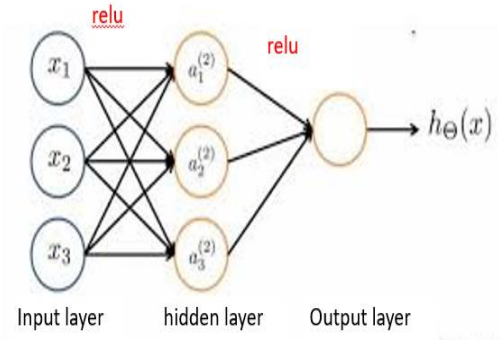
Deep Learning HW1_1 Report

0853411 劉書維

1. **You have to show your learning curve, training error rate and test error rate in the report.**

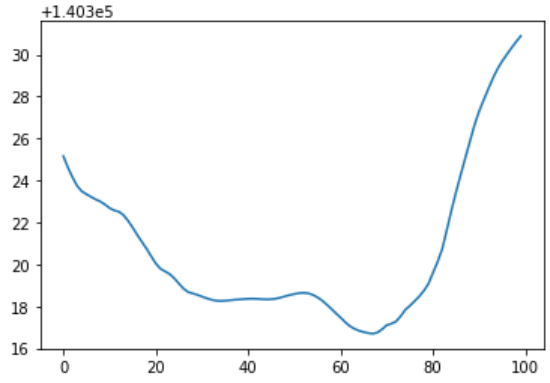
   我用的架構是 2-layer network，然後搭配不同激勵函數來觀察錯誤率，用了 2 種 model，兩種差異極大，2 個 model 圖形說明如下：





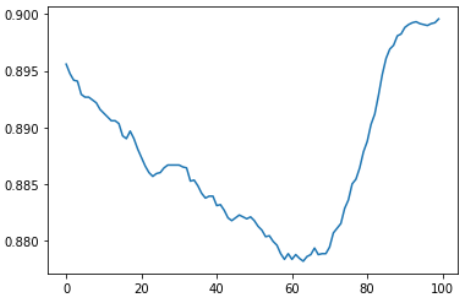
Model 1 : (sigmoid + softmax)

這個模型所產生的 output 容易集中在少數圖片，所以準確度極低，且很快就收斂，錯誤率約 87%。下圖是 train loss 的圖形，可以發現越訓練到後面，其 loss function (cross-entropy) 逐漸上升.....。參數：iters_num = 100；learning_rate = 0.001；init_weight = 0.01。

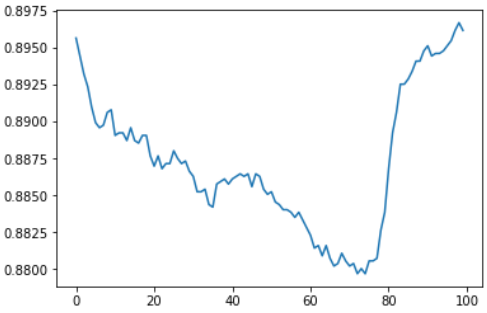


所以其 train error rate 和 test error rate 都很高，且在 iter=60 時就已經是最佳解了。

Train error rate：



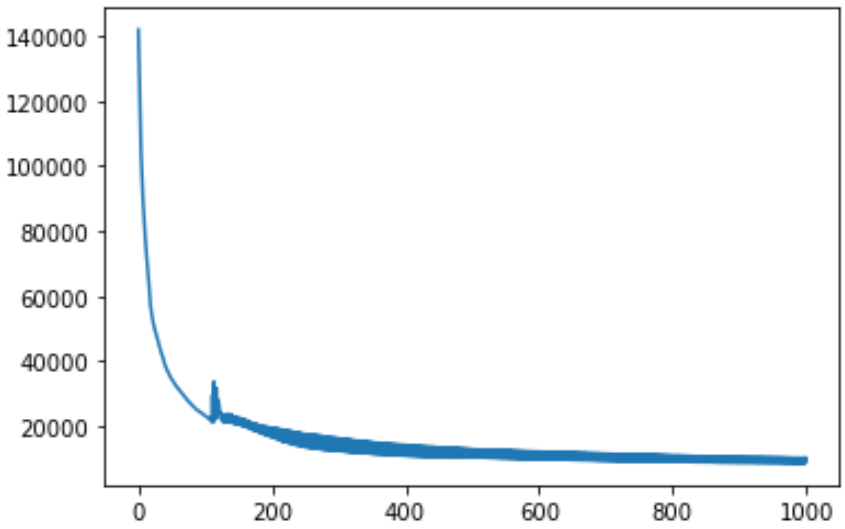
Test error rate：



所以這個結果很糟糕。

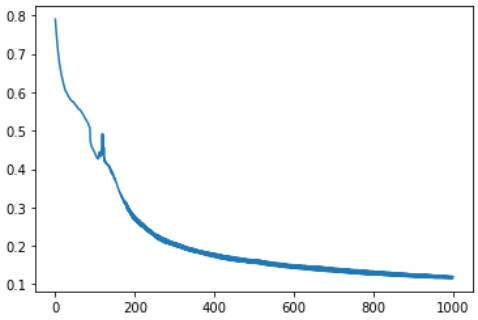
Model 2：(ReLU)

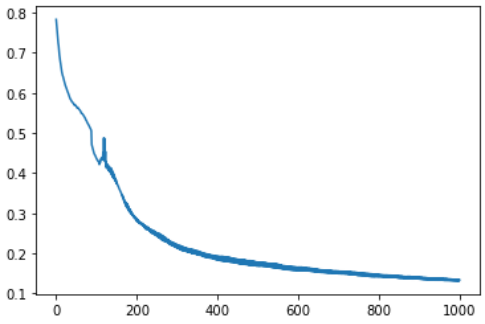
後來都改用 ReLU 效能則是很不錯，且錯誤率可以降到 10%左右（準確率＝90%），可能是因為 R eLU 有明顯線性性質，適合此類多類別的分類。參數：iters_num = 1000；learning_rate = 0.001；init_weight = 0.01。其 loss function（cross-entropy）如下圖：



隨著 iter 越多次，error 也隨之下降，最後降至差不多 10%。

Train error rate：　　Test error rate：

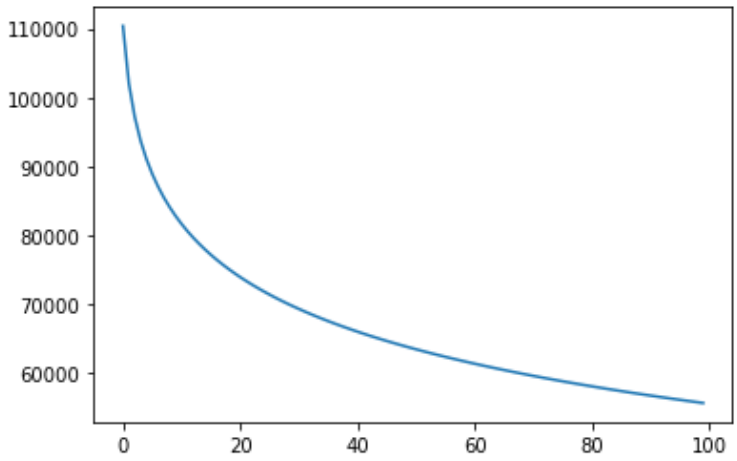




所以以下題目結果都以 Model 2 為主。

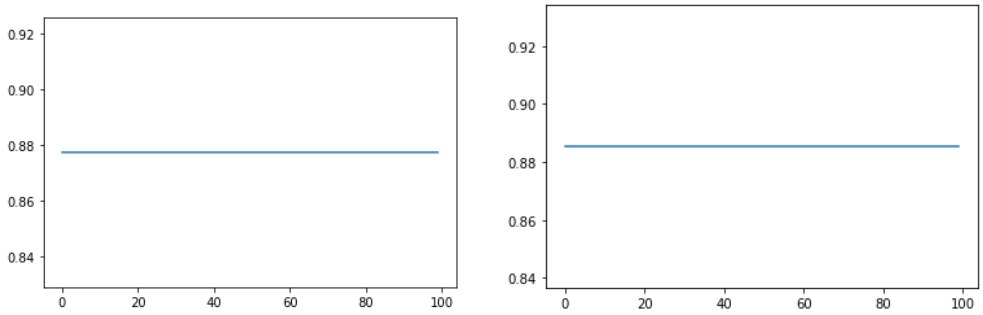
2. **Please perform zero and random initializations for the model weights and compare the corresponding error rates.**

若用 random 的方式產生 weight，其結果與上面結果差不多，但是若將 weight 設為 0，則會因沒有 slope 導致無法找出最佳權重與梯度，所以極快收斂，且錯誤率極高。其 loss function（cross-entropy）也較高，如下圖：



這個錯誤率更是高達 88%，且居高不下，也絲毫沒有下將的趨勢。

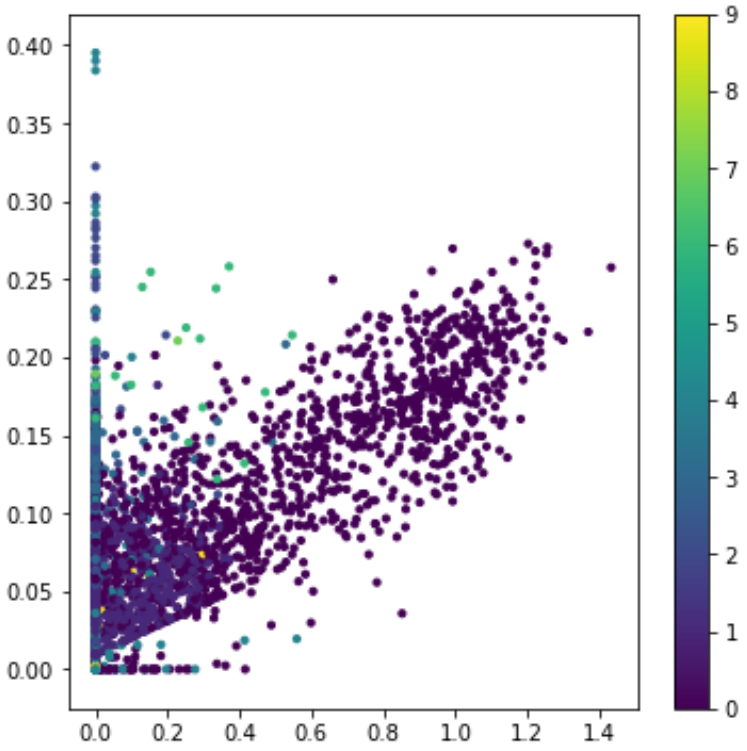
Train error rate：　　Test error rate：



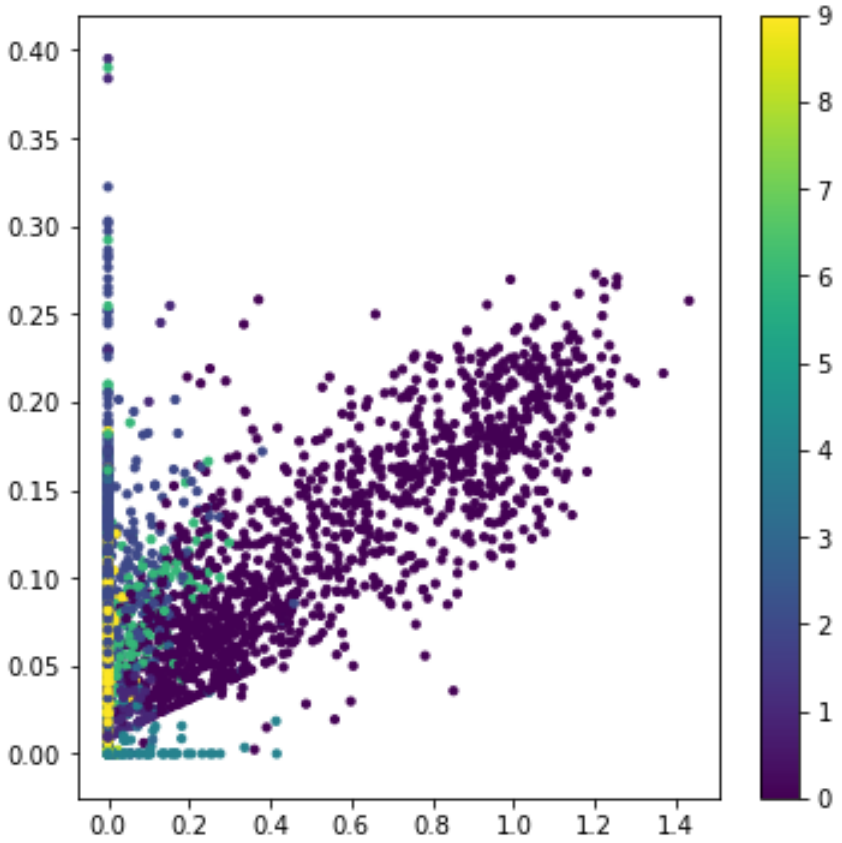
3. **Plot the distributions of latent features at differet training stages.**

分別是擷取早期訓練 iter=20 與 iter=80，來呈現資料，但好像因為沒有正規化（但我不會處理 QQ），所以資料分布較為密集，且難以看出界線。

Iter= 20：



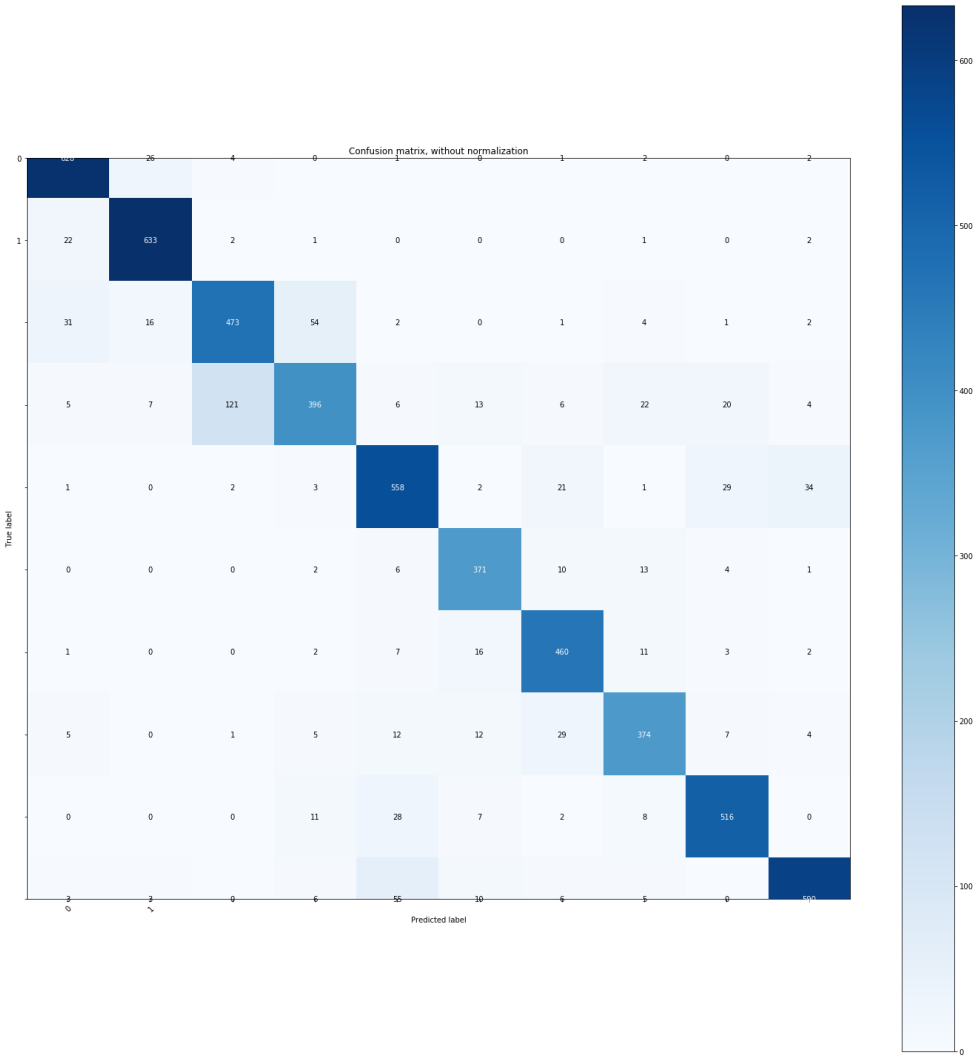
Iter= 80：



可以看出他們的分布有越來越散開，且訓練出新的 cluster（黃色），也因此，他們之間的差別與界線會越來越明顯，所以精準度會不斷上升。

4. **List your confusion matrix and discuss about your results.**

Confusion Matrix 如下圖。基本上都是 label 正確的，未來可能可以思考用更多技巧（如：batch-training 等）或者試試不同激勵函數（如：Leaky ReLU 等）來增加其精確度。


Confusion matrix, without normalization
True label
Predicted label
0
1
26
4
0
1
2
0
2
22
633
2
1
0
0
0
1
0
2
31
16
473
54
2
0
1
4
1
2
5
7
121
396
6
13
6
22
20
4
1
0
2
3
558
2
21
1
29
34
0
0
0
2
6
371
10
13
4
1
1
0
0
2
7
16
460
11
3
2
5
0
1
5
12
12
29
374
7
4
0
0
0
11
28
7
2
8
516
0
3
3
0
6
55
10
6
5
0
590
600
500
400
300
200
100
0