Machine Learning HW3 report

姓名：顏修溫
學號：r05922034
系級：資工所 碩一

1. Supervised learning: Use only labeled data to train a model, record its performance, and describe your method

此次作業我所使用的 toolkit 是 Keras。我把 all_label.p 裡頭的資料讀進來，然後處理成 X_train(5000 張[3*32*32]的 image)和 y_train(label)這兩個資料結構。接著對 X_train 做 normalize 後，把 X_train 和 y_train 都丟進去 CNN model 裡開始 train。我的 CNN model 是由 2 層 Convolution2D+MaxPooling2D 接上 3 層 layer 的 fully connected neuron network，最後再加上 Dropout(0.5)所組成。Batch size=128，然後 train 60 個 epoch。

經過嘗試各種 model 後，使用 Supervised 方法最好大概可以得到 kaggle 上 0.47 的分數。這顯示 5000 筆資料量實在太少了，CNN 很難從這少量資料中萃取出能有效區別類別的 feature 與參數，所以需要 semi-supervised。

剛開始在 Train 的時候，我有遇到一些問題像是 loss 降不下來，原本是懷疑我 input 資料處理的不好。但經過各種交叉測試後發現 model structure 的影響甚大、以及 optimizer 的選擇也有差異。後來我選擇把 layer 層數加深後才得到較滿意的 model。雖然不是很嚴謹的驗證，但我多少也體會到老師上課所說的 Do deep nets really need to be deep?　YES!。

2. Semi-supervised learning (1): Use whole data to train a model, record its performance, and describe your method

Semi-supervised learning (1)的部分，我是**使用 CNN 做 self-training**。此方法的 model structure 就是拿上一題的延用，而且我把 test data 也拿來當作 unlabeled data 使用。先 train 在 5000 筆 label data 上後得到一個 model，接著拿此 model 去預測所有 unlabeled data 的類別，由於 prediction 的結果會列出 input 屬於某一類的機率有多少，所以我就以這個機率當作信心程度來評估該 input 該不該加到 training data 裡頭，若機率高於某個 threshold=0.8 時，input 與 predict label 才會被加到 training data 裡。接著再重新建一個 model 去 train。這些步驟來回重複個 5 遍後，大概可以把 90% 的 unlabeled data 加到 training data 裡頭。

目前丟到 Kaggle 上的最好結果就是使用 self-training，得到 0.536 的分數。

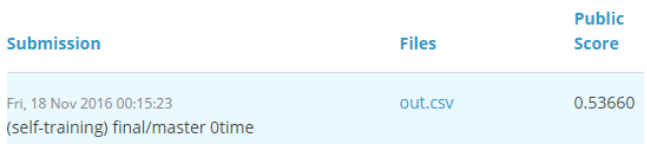

Fig 1. Self-training score

3. Semi-supervised learning (2): Use another method, record its performance, and describe your method

Semi-supervised learning (2)的部分，我是**使用 autoencoder + self-training**，我同樣把 test data 也拿來當作 unlabeled data 使用。而我的方法流程如下：

step 1. 使用 label data + unlabel data + test data 去 train 一個 autoencoder

step 2. 把 label data 丟進去 autoencoder，從 autoencoder 的中間 layer 取出每張 image 的 code (encoded_dimension 我設為 256)。

step 3. 把 labeled data 的 code 當作 X_train，label 當作 y_train。丟進 fully connected neuron network 去 train 出一個 model。

step 4. 接著把 unlabel data 和 test data 的 code 丟進去這個 model 進行 predict，從 predict 結果去判斷當 input 屬於某一類的機率大於 threshold 的話，才會對這些資料進行標記並加進去 label data。然後又回到 Step 3。

這些步驟進行 5 輪後，已經把大部分的 unlabeled data 加進去 training data 了。目前丟到 Kaggle 上的約得到 0.35 的分數，效果並沒有很好。

另外有個比較不一樣的地方是，當我進行完上述 Step 1.後需要先把 encoder model 存起來、也需要把 fully connected net model 也存起來。因為在執行 test.sh 時，需要先把 test data 經由 encoder 轉為 code，接著把 code 丟進去 fully connected net 後才能 predict 該 test data 的所屬類別。

Fig 2. Autoencoder score

4. Compare and analyze your results

比較我所做的這幾種方法，CNN 做 self-training(上述第二題)這個方法的效能最佳。而且我 supervised learning 與 self-training 這兩種方法所用的 model structure 是一致的，然而使用 self-training 的效能有提升，這也就意謂著使用 unlabel data 確實可以優化修正 supervised 方法所得到的分類器。然而 autoencoder 的效果卻是挺差的，是以上方法中效能最不好。

除此之外我做了一些實驗，在 semi-supervised 方法中每一輪我都會把部分的 unlabeled data 加進去 training data 後重新 train model，然而我有實驗過完全重新建一個新的 model 去 train、或者是拿前一輪的 model 繼續 train。經過多次比較後發現當我重新建一個新的 model 去 train 的話效果會比較好，所以我後來採用的方式都是重新建 model。

另一個關於 semi-supervised 的實驗就是，當 model 對 unlabel data 進行 predict 後，我們可以選擇把 predict 後的 label 全部加進去 training data 裡、或者是只把信心程度較高的加進 training data。我自己去切 validation set 後的比較結果顯示，設定一個信心程度 threshold 的方式可以得到較佳的結果。