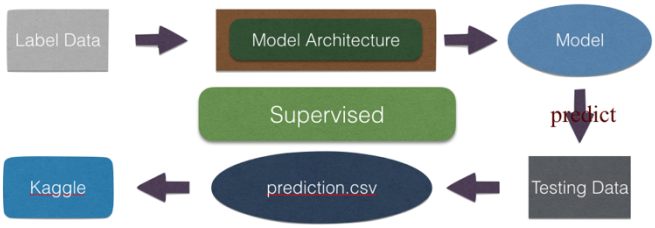
[Machine Learning HW3]　Cifar-10 Classifier　R04921006 謝仲凱

[Keras + Theano backend]

1. **(1%) Supervised learning(only Label data)**:

performance: private-> 0.65520



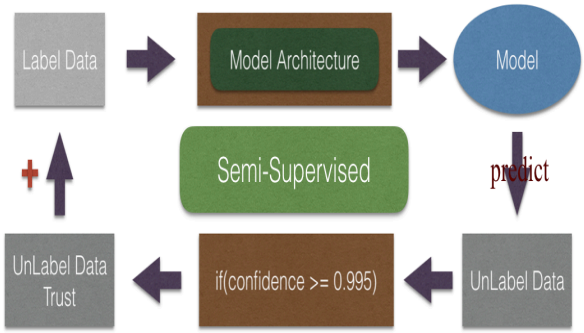
Model: 這次學著使用 CNN 對影像抽取 feature,最後在 flatten 拉成一維，最後使用 softmax 做 classifier。Model 要好的要件我認為有兩個

1:activation(ReLU)

2:optimizer(adam)

2. **(1%) Semi-supervised learning (Label data + unlabel data)**:

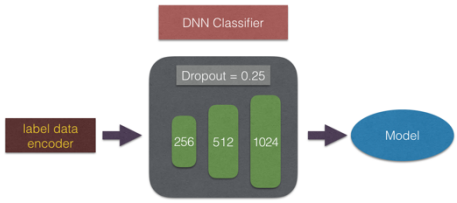
performance: private-> 0.70740



使用 supervised 設計出一個還行的 model。接著使用著個 model 去 predict unlabel data 把信心度比較高的 data (>= 0.995)加入 label 之中(self-trainning).重複做 4 次，performance 明顯提高。

3. **(1%) Semi-supervised learning (2)**:

performance: private-> 0.4172



這邊使用兩個 model. 1:CNN auto-encoder 2:DNN classifier。我認為這個 CNN auto-encoder 做的滿成功的，decoder 跟 input 滿像的，模糊是因為有做 pooling & upsampling. Performance 不如預期出在於 DNN classifier 沒設計好，從我的 training accuracy 也不夠好來做判斷得出的結論。

4. **(1%) Compare and analyze your results**

[Supervised]:model 設計->觀察 training data 上 accuracy >0.95 基本上 model 不錯，但是 val_acc 就廢的跟狗屎一樣，因此這次花很多心思在 dropout 上面，發現適當的 dropout 不要讓 train_acc 掉太多, val_acc 可以上升得不錯時，improve performance from 0.58 to 0.65(up 0.7) 。

[Supervised vs Semi-Supervised]: 原本只用 4500 來 train(500 拿去當 validation),後來使用 self-training 變成 13000 多筆 data performance(up 0.5) 可見多看一些沒有 label 過的 data 是很有用的。

[Auto-encoder]:很好玩哈哈，performance 不高是因為 DNNclassifier 沒設計好

[code 執行方式]

Master:

./train.sh path model

./test.sh path model prediction.csv

method2:

./train.sh path model

./test.sh path model prediction.csv

(為了不希望助教還要多打一個 argument 存 autoencoder 的 model, method2 在 train 的時候就會存一個 **X_test_autoencoder.npy** 在 test 中做 predict 使用)，因此必須 train 完 才能 test。

辛

苦

拉