

ML Report-HW3 Semi-supervised learning

姓名：許義宏 學號：B02901053

作業綜合介紹： 本次作業使用python的套件-Keras（以tensorflow作為backend）完成作業，Kaggle 上最好的分數是使用Semi-supervised learning- Self learning 為方法所達成。

1. Supervised-learning:

- 模型簡介：
 - 初步架構：以CNN為先（使用32個3x3的CNN疊2個之後,使用MaxPooling2D shape(2,2)來做subsampling, 重覆以上的架構再一次, 但CNN改為64個），接着攤平後使用DNN來進行training（使用Dense 1024+Dense512）。過程中的activation function皆以relu作為activation function。
 - 優化：在CNN與DNN的後面加上BatchNormalization進行優化(參考：<https://arxiv.org/pdf/1502.03167v3.pdf>)
- 資料處理：
 - 在5000筆的labeled data中, 切500筆作為Validation Set, 其中500筆是每個class均勻分佈（每個class50筆Data）
 - 將資料/255（因為256階色), 使資料的分佈能在0~1之間
 - 為了讓資料量增多, 使用Data Generator作為擴增資料的方法, 其中包含旋轉, 水平反轉...等方法
- Training：
 - 利用Callback Function, 以Validation Set的Accuracy作為判斷標準, 存下val_acc最高者作為model
 - 在使用Data Generator部分, 一次Train採用45000筆sampling(10倍資料量)
- Performance：
 - 在使用batch normalization 優化前, 最高可使validation accuracy 達到67%
 - 再使用batch normalization 優化後, 最高可使validation accuracy 達到72%
 - 藉由調整參數與微調架構（增加CNN的數目）, 可達到73.5%

2. Self-learning:

- 模型簡介：
 - 基本的model架構如同Supervised Learning
 - 用迴圈進行self learning, 迴圈內完成的事為三：
 - Training using “Fixed labeled data” only：一樣用data generator(sample 5x)進行label data的training
 - Predict：利用剛train好的model預測unlabel data的答案, 並且如果預測出來在class x 中有最高機率且機率大於threshold（初設0.995）的話, 即把此筆data從unlabel data 移至'label data'中, 並標記答案為class x
 - Training using“label data”:使用調整過的label data進行training
- 參數調整與Performance比較：(備註：前三點的比較batch皆為64)
 - 使用最基本的self learning(迴圈內只有Training using 'labeled data'+Predict), 最高可做到0.786的validation accuracy（Inductive）
 - 使用改進過後的方法可以做到0.818的validation accuracy（Inductive）

- 使用Transductive的方法(即把10000筆的testing data 也當做unlabeled data)可以將performance提高到0.832的validation accuracy
- 調整batch的大小，發現batch小一些時有更好的表現（以下用transductive）

Batch	128	64	32	24	16
Validation Acc	0.823	0.832	0.841	0.868	0.838

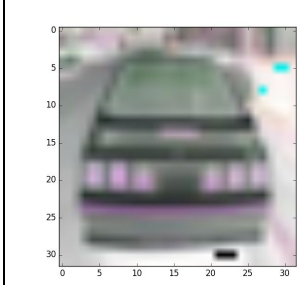
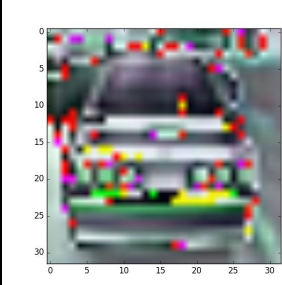
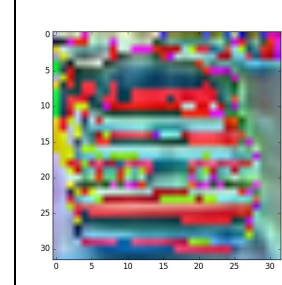
- 針對batch = 32 的training model，我調整CNN的層數與數目進行測試

CNN架構： (數字代表數目，逗號為分層)	64,64,128,128	32,32,64,64,128,128	64,64,128,128,256,256
Validation Accuracy	0.841	0.848	0.852

3. 利用AutoEncoder進行Pretrain:

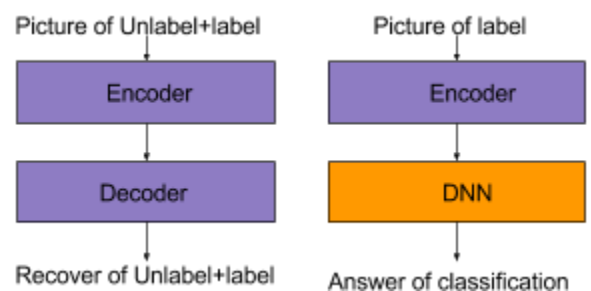
➤ 模型簡介：

- 先利用autoencoder將圖案進行處理，例用對稱的模型架構將unlabeled data+label data進行training model，我的方法是除了使用CNN以外，還加入DNN去training，然後發現使用regularizer會讓成果變好（成果如下）：

原圖	Model 1	Model 2
		

Model 1 用參數比較小的regularizer，Model 2 則是參數較大的，若不使用Regularizer的話成果不佳。其實從結果看得出來只能判別出一些輪廓，但色調與差別依然很大

- 接着將training好的Encoder當做pretrain的Model，將它接上DNN的架構
- Performance: 最好可以training到0.702的val_acc(註：由於上傳的Trained Model控制在大小小於100MB，Github 上的val_acc 只有0.67)



4. 比較與分析：

從以上的說明的Performance可以看出來，Self Learning>Supervised Learning >Auto encoder Pretrain(此方法則差強人意)，主要原因我認為是因為在cifar-10裡，很多圖片的背景都不是很單純（像MNIST背景就滿單純的，Autoencoder的表現就比較好），所以讓autoencoder的表現不是太好，才導致整體的classification效果不佳，甚至比supervised learning還要略差一點點。理想上，Encoder能因為看過比較多筆的Data而讓Encoder能夠作為一個好的濾feature的model給DNN（這裡我不做CNN+DNN的原因是因為我在Encoder就已經把model flatten然後DNN了），讓整體效果能比supervised learning 還要好。而self learning就如於預期透過比較多的假設與資料達到比較好的效果