Lab 3-1 Report

Introduction:

這次Lab的目的在於讓我們熟悉各種加速深度學習訓練過程的方法，首先第一部分是學會如何實作early stopping，接著了解使用pretrained model的好處以及效果。

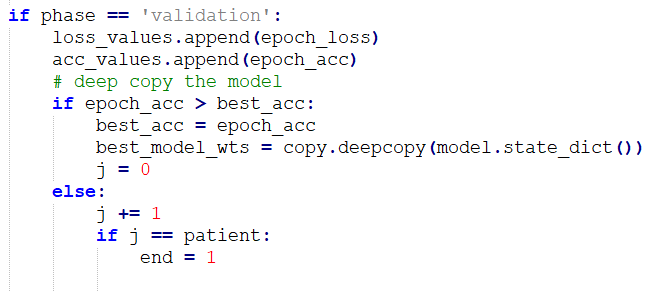
Experiment setup:

我是在win10下使用Anaconda的jupyter notebook進行實驗，顯卡是GTX1070。

Result:

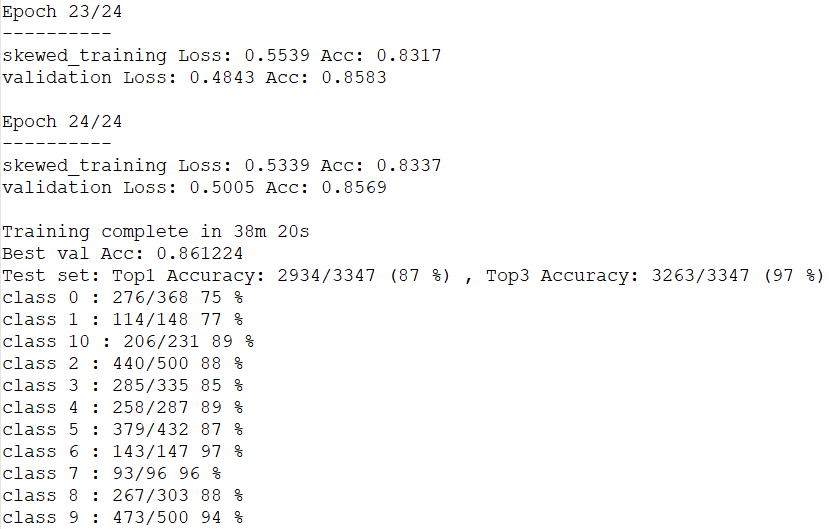
(a)

我們預設training的epoch是25，不算很多，因此我們將patient設成5，另外設一個flag來判斷當超過5個epoch沒有得到更好的參數時即跳脫迴圈。

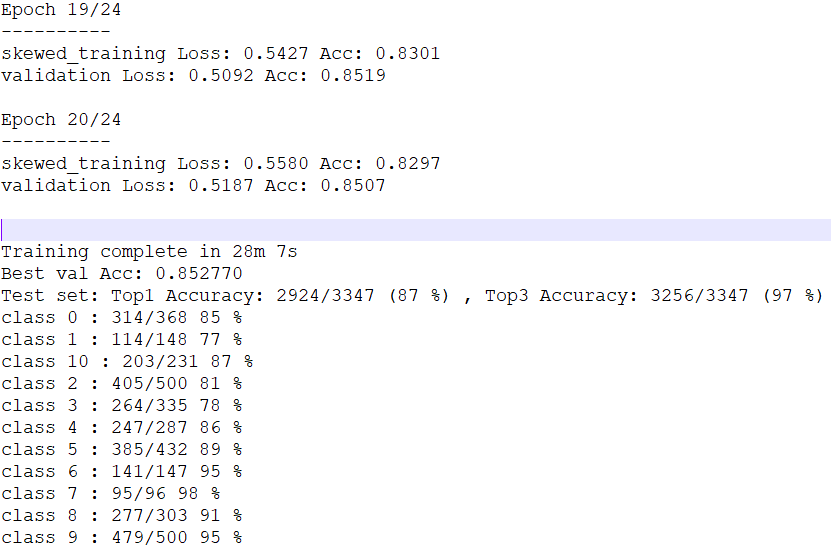


在validation的時候，只要得到更好的參數就將模型儲存起來，並將j設回0，沒有就將j加1，只要5次沒有更新參數就把end設成1，之後就會跳出迴圈。

原先的結果如下:

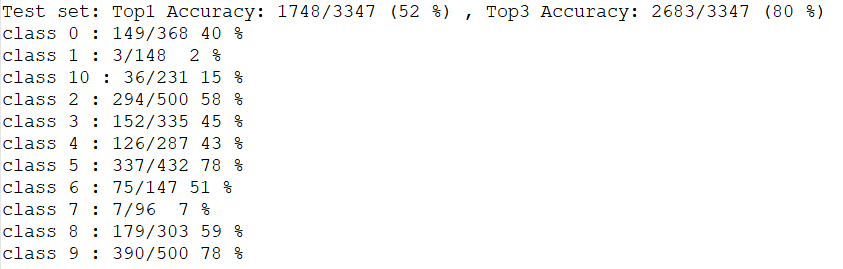


使用early stopping之後:

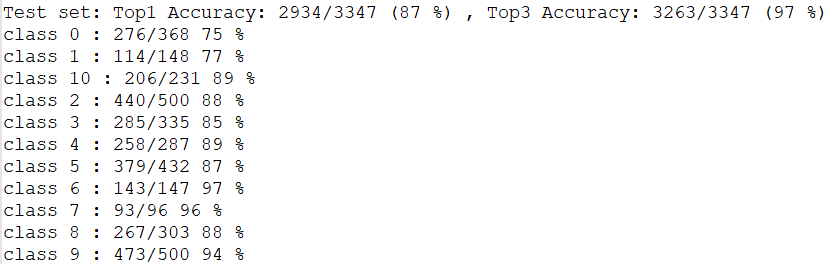
可以看到epoch減少到了20個，因為到這邊累計5次沒有更好的參數因此跳脫迴圈，而訓練時間減少了10分鐘左右。另外，準確率幾乎沒有任何影響，因為我們原本的model本來就會去紀錄最好的權重，並當作最後輸出的結果。

(b)

我們這部份使用的CNN model是resnet18，使用資料是skewed food11，首先這是尚未使用pretrained model的結果。

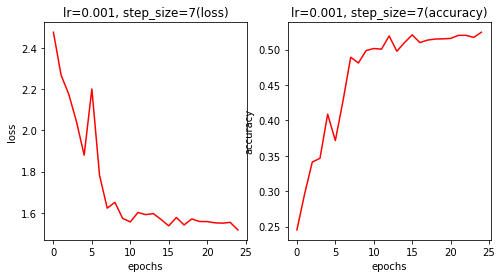


準確率很低，尤其是資料很少的類別，然後使用pretrained model的結果如下。

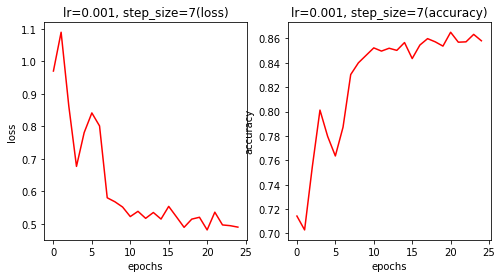


整體的結果有相當大的改善，平均準確率提升了35%左右。

有關收斂速度的部份，下圖是epoch對loss以及accuracy的曲線圖，尚未使用pretrained model的情況如下。

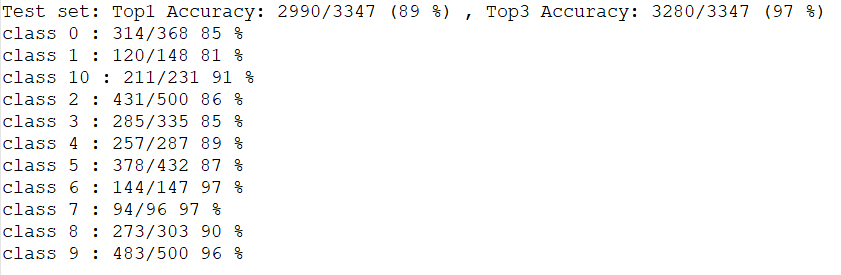


約莫在第10個epochs左右的時候收斂，而使用pretrained model的情況如下。

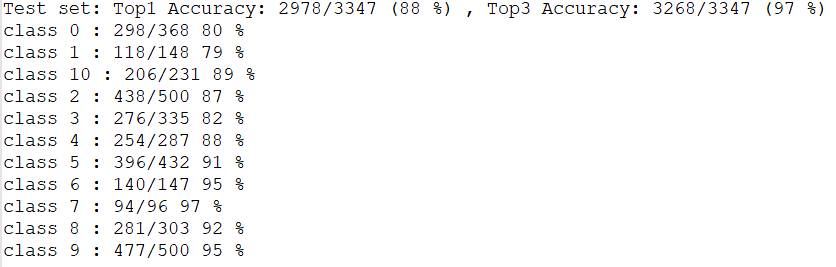


也是約莫在第10個epochs左右的時候開始收斂，因此我認為兩者收斂速度並無太大差異。

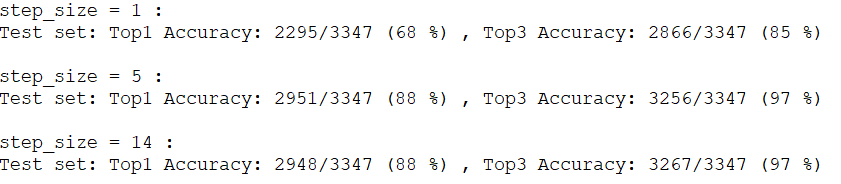
在調整learning rate的部份，通常在transfer learning當中，傾向於使用較小的learning rate，避免太快去破壞掉先前model已經trained好的權重，並防止失去某些資訊。在前面的實驗，我們使用的learning rate一開始是0.01，每過7個epoch做一次下降的動作(乗以0.1)。現在，我們降低一開始的learning到0.005，得到的結果如下。



準確率有些微的提升，繼續下降到0.0025，得到的結果如下。



並無明顯的提升情況。另外，我們測試改變step\_size(下降一次所經過epochs的數量)的結果，結果如下。

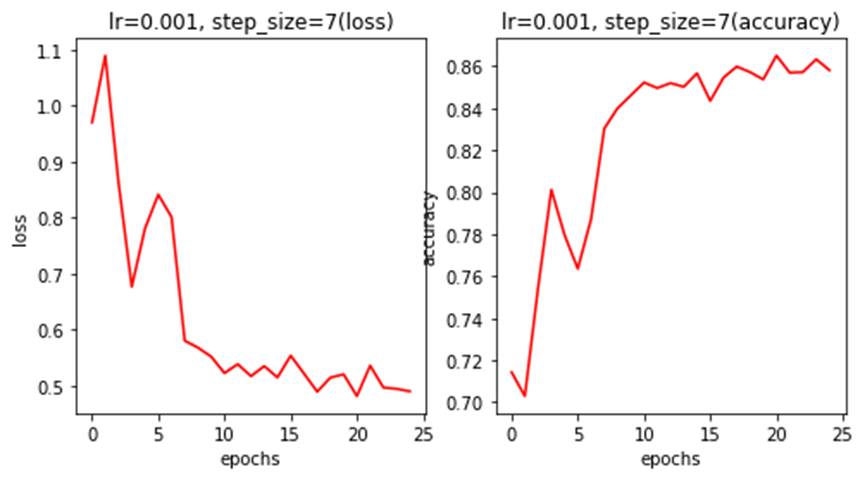


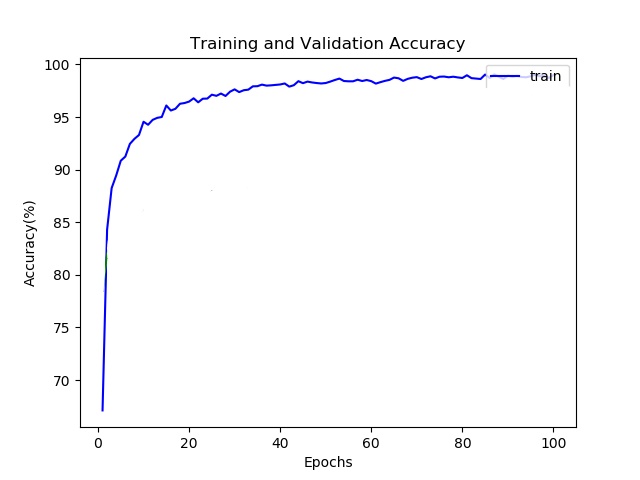
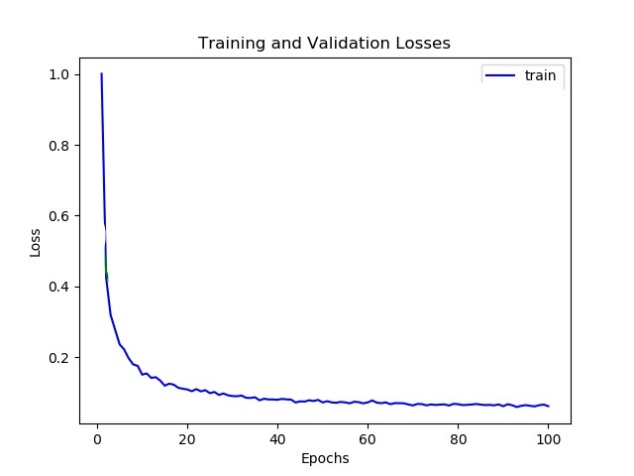
當設成1的時候，準確率明顯的下降，另外以7為基準上升或下降都沒有明顯變化。總體來說，learning一開始設成0.005左右，step\_size設成7左右的結果應該是最好的。

(C)

我選用resnet50作為teacher model，並與resnet18的student model做比較。batch\_size = 32, lr = 0.001

下圖為resnet18未經Knowledge distillation的訓練結果



下圖為經過resnet50 Knowledge distillation的訓練結果

可以發現在未經Knowledge distillation時，在25個epochs僅能收斂到86%之accuracy及 0.5的loss，而經過Knowledge distillation之後，在20個epochs就達到了95%以上的accuracy及0.2以下loss的優異表現

(d)

選擇的tuner : TPE、Evolutionary

