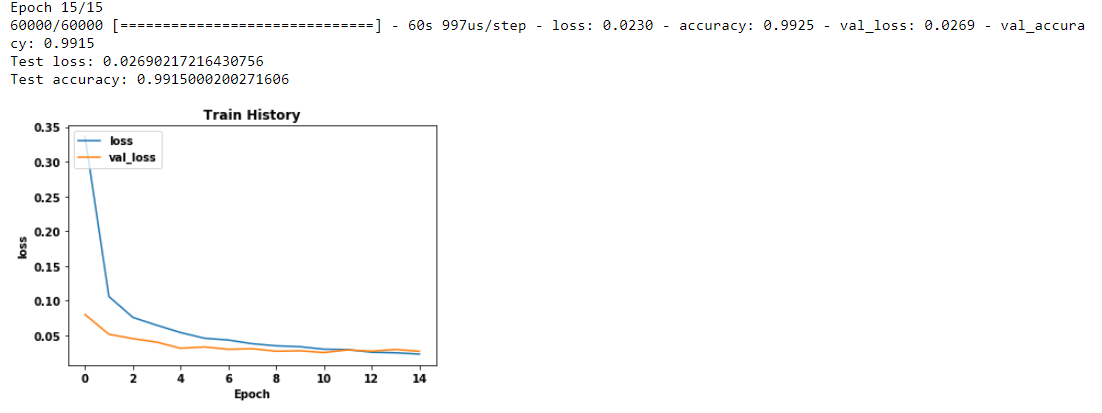
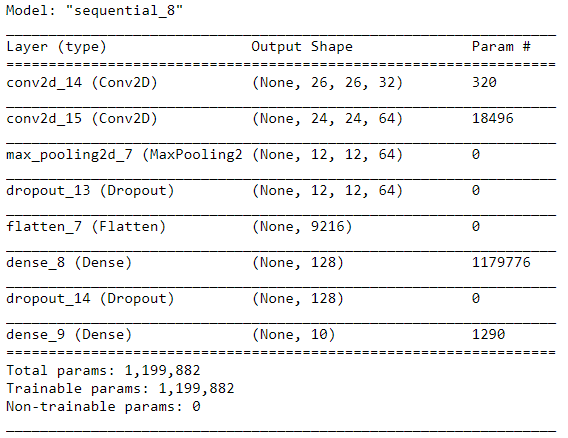
HW2 B063012054 林祐安



(未改變參數前)



(模型架構)

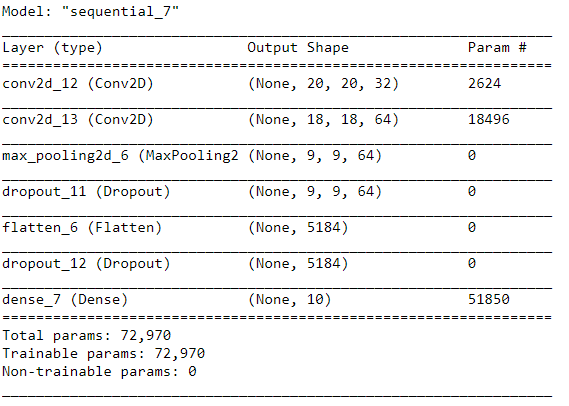
使用GPU GTX1066訓練每個epoch需耗時一分鐘左右，模型架構僅兩層卷積以及全連接層，理論上耗時不需太久，但在此次實作使用的樣本有60000份，數量不少，因此耗時長。為提高效率最簡單的方法是加大”batch\_size”，原先為256，由於資料有60000筆，可以試著把batch\_size提高至1024(每個epoch 約耗時47秒)、4096(每個epoch約耗時45秒)，觀察發現batch\_size=1024與batch\_size=4096的耗時差不多，這表示再增加下去意義不大，且訓練效果可能會非常差勁。較為進階點可以修改optimizer、甚至於模型架構。

--------------------------------------------------------以下為修改後模型--------------------------------------------------------

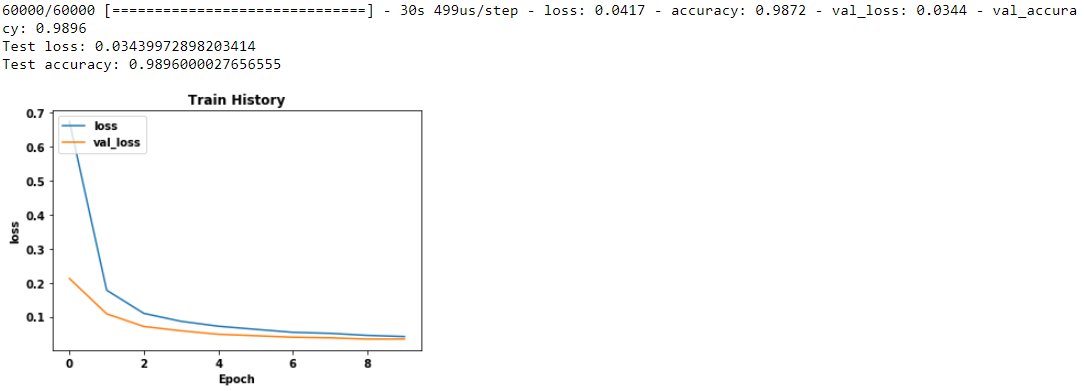
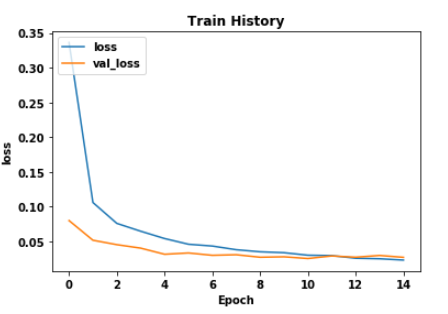


(修改後code)

首先，將第一層的卷積加大成9X9的Mask，因為圖片僅有黑白兩種顏色且只有1個channel，因此不需要小的Mask去抓取特徵，提高效率；第二層則維持不變，提高分辨率。修改第二次防止過擬合的Dropout至0.25，因為前面已Dropout一次，理論上第二層Dropout並不需要，因為模型非常簡單且只有黑白兩色，但資料不少，因此斟酌至0.25；特別注意到→處，原先有一層Dense，但我把它拿掉了，因為Dense兩層沒有甚麼意義。optimizer由Adadelta修改至Adam，Adam對於local min. 有極佳的效果，收斂速度雖然不是最快，但是最穩定。由前面的經驗可以知道，在epoch 12以後已開始收斂，且batch\_size=1024是瓶頸，預期更改後的模型效率較高，因為參數量大幅減少，因此調整epoch至10。

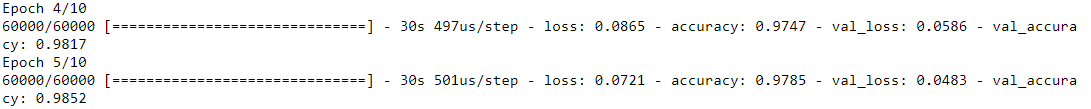


(修改後模型架構，刪除一層全連接層)

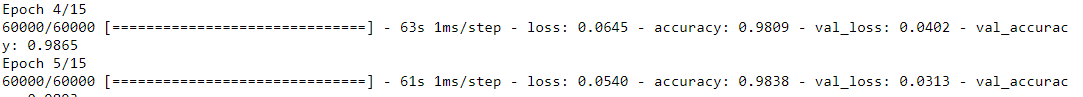


(修改後模型訓練10th epoch與修改前的模型Train History)

先觀察修改後的模型，可以看到每個epoch的時間由一分鐘減少到半分鐘，快了一倍的時間，效率大幅提高，事實上在4th epoch時loss已下降到0.0865，與原本的模型相比，雖然一開始的Loss較高，但由於動量關係，下降速度頗快，在epoch 4時兩者loss其實已差不多。



(修改後模型 4th、5th epoch)



(修改前模型 4th、5th epoch)

兩者的正確率、loss差不多，因此這是一個非常成功的改良版模型，大幅提高了訓練效率。