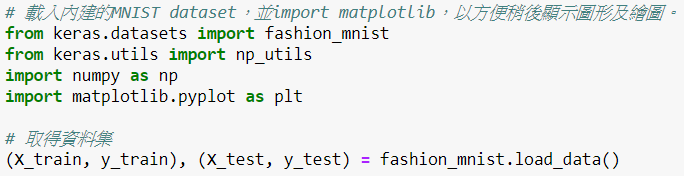
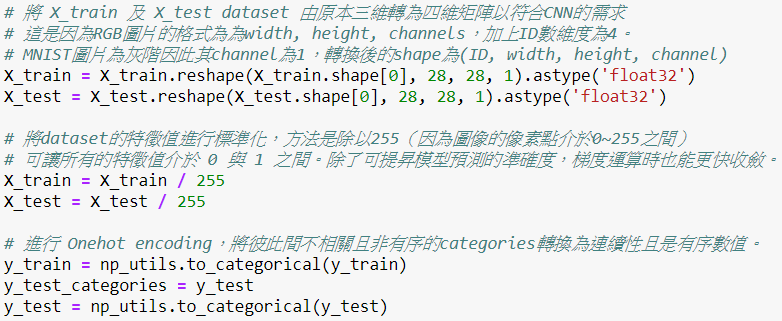
**ECT\_HW9**

**利⽤ Keras 套件於 tensorflow 上，使⽤ CNN 深度學習演算法對**

**fashion\_mnist 資料及進行分類，依序完成以下步驟及問題：**

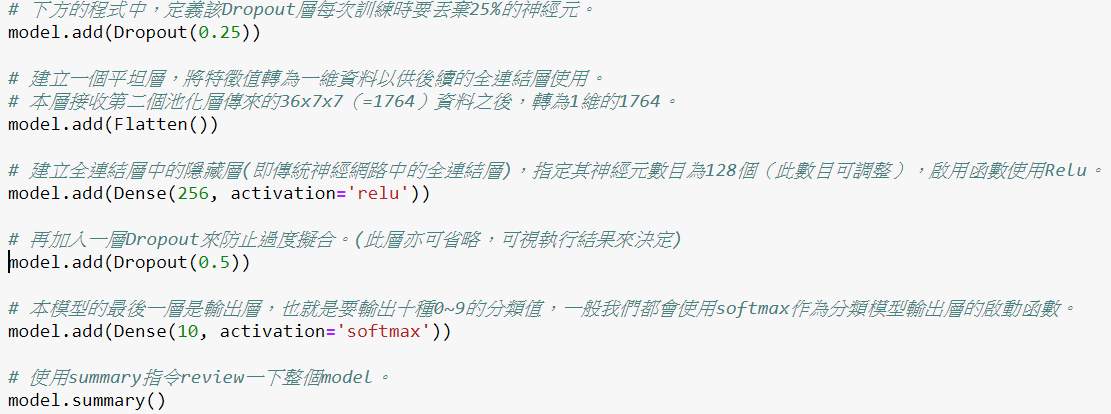
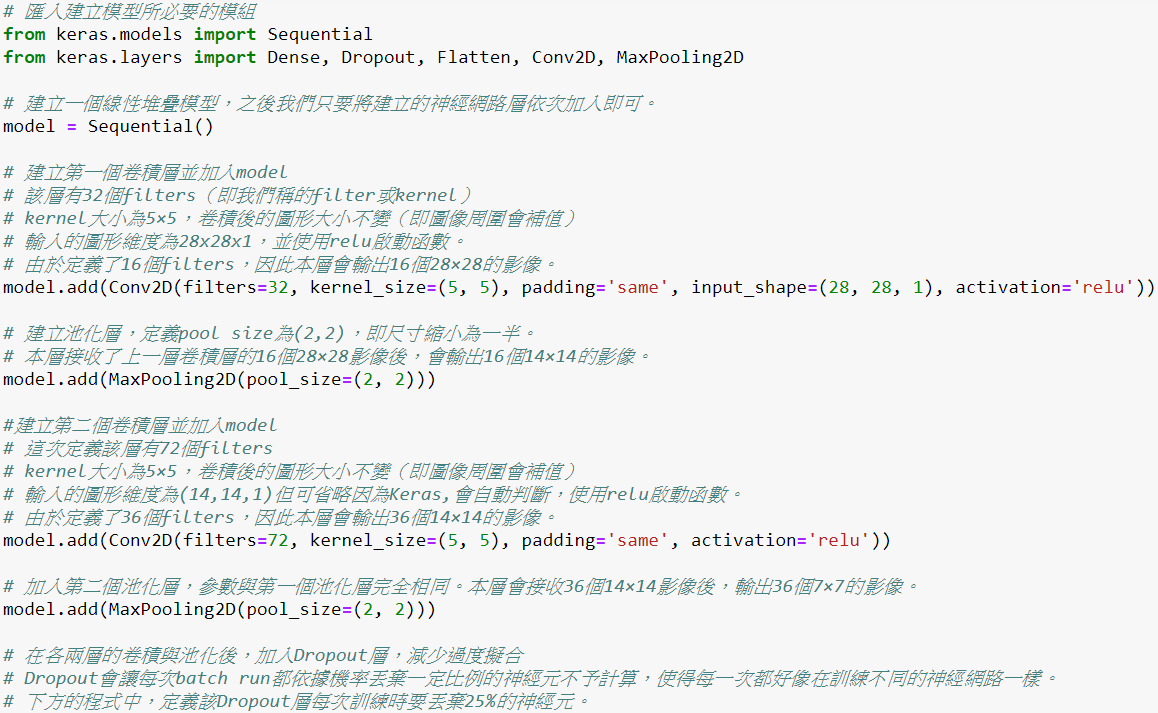


首先，載入所需套件並取得資料集。



並對資料進行前處理，將 X\_train 及 X\_test dataset 由原本三維轉為四維矩陣以符合CNN的需求，並將dataset的特徵值進行標準化，方法是除以255（因為圖像的像素點介於0~255之間），進行 Onehot encoding，將彼此間不相關且非有序的categories轉換為連續性且是有序數值。

**1. 建立 CNN 模型**



**2. 解釋設的參數 (filter 數量、大小、 activation function 設置、 dropout)為何 ?**

下方的程式中，建立第一個卷積層並加入model，該層有32個filters（即我們稱的filter或kernel），kernel大小為5×5，卷積後的圖形大小不變（即圖像周圍會補值），輸入的圖形維度為28x28x1，並使用relu啟動函數。由於定義了16個filters，因此本層會輸出16個28×28的影像。



Dropout會讓每次batch run都依據機率丟棄一定比例的神經元不予計算，使得每一次都好像在訓練不同的神經網路一樣。下方的程式中，定義該Dropout層每次訓練時要丟棄25%的神經元。



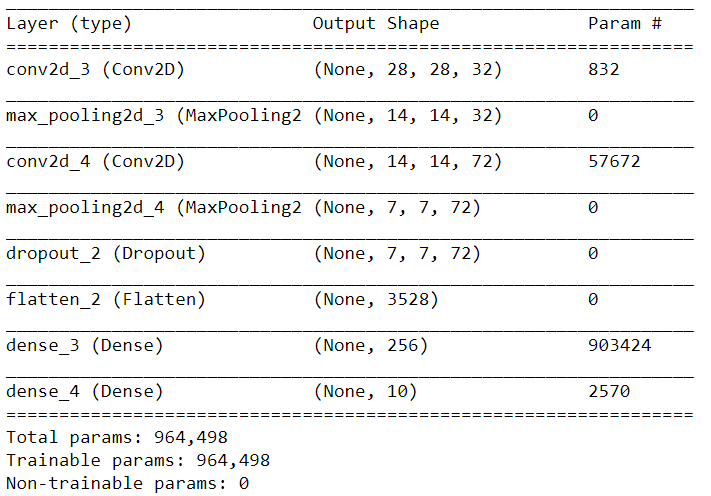
**3. 評估訓練結果** 

建立訓練模型，並設置validation\_split=0.2(20%的資料當測試集)，epochs=35 (執行35次訓練週期)，batch\_size=300 每個訓練週期會將所有資料以每批次300筆來執行。



第35次訓練所得之loss、acc、val\_loss、val\_acc。

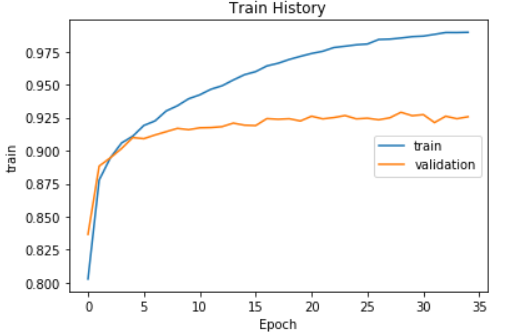
**4. 顯示 model.summary 結果**



**5. 準確率視覺化**

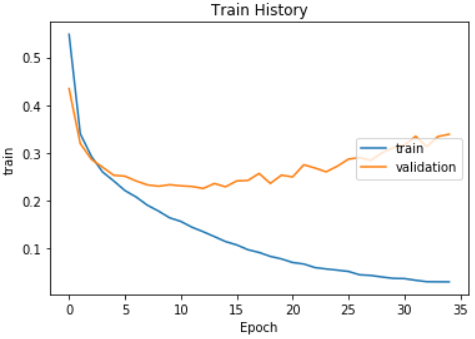
定義一個show\_train\_history函式，我們只要將結果丟入，便可繪成圖表。

輸出結果如下：

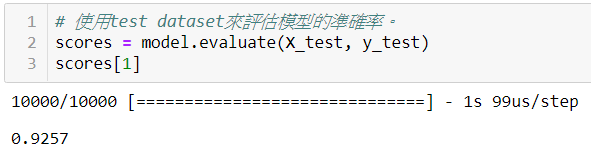


**6. 誤差率視覺化**

輸出結果如下：



**7. 整體模型準確率(於test set)**



使用test dataset來評估模型的準確率可得0.9257

**8. 混淆矩陣**

