**電腦網路實驗實驗報告**

**< 智慧聯網霧端運算-應用聯盟學習於邊緣網路 >**

**姓名: 翁佳煌　　　學號: 409430030**

# 實驗名稱

Classify images of clothing

# 實驗目的

使用TensorFlow套件訓練一個深度學習模型，以辨識服飾圖像。通過探索和處理訓練和測試資料，建立模型，進行預測，並評估其預測結果的準確性。最終，利用訓練好的模型，對單張測試圖像進行預測。此實驗旨在加深對深度學習的理解，並提高對TensorFlow套件的使用能力。

# 實驗設備

Linux作業系統之電腦。

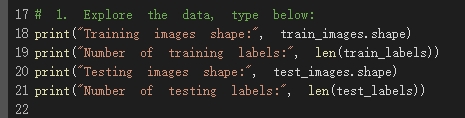
Google Colab

Tensorflow套件

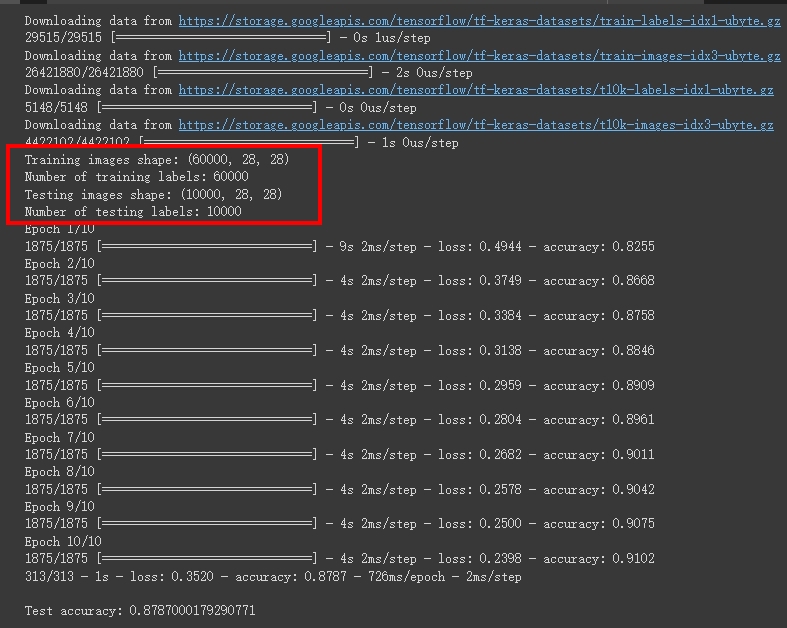
# 實驗步驟

1. **Explore the data**

完成第一個要求只須按照下圖1即可完成，下圖2為印出之結果，會在問題與討論中探討其印出的數字意義。

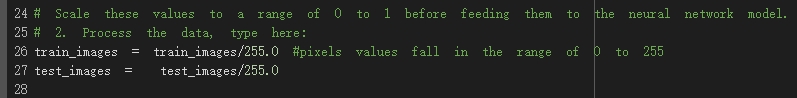


▲圖1



▲圖2

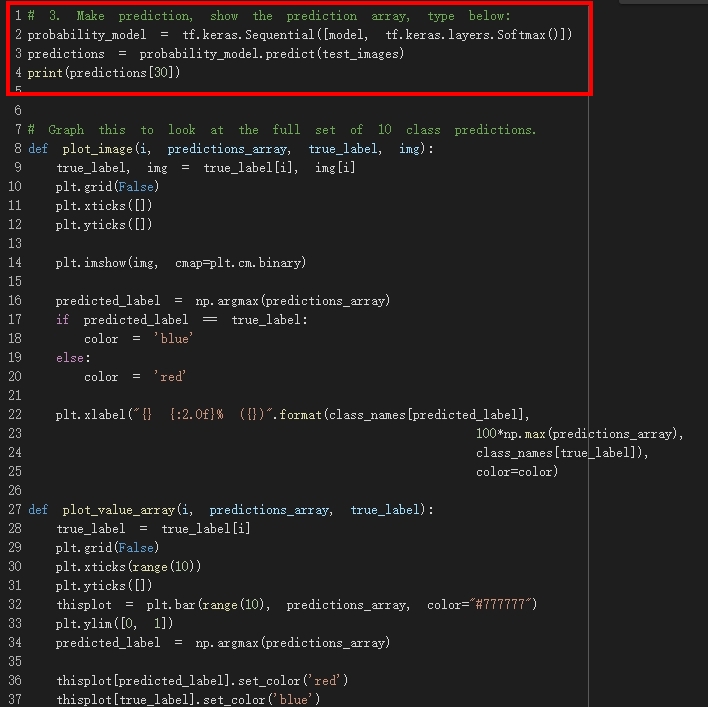
**2. Processthe data**

根據圖3所示，即可完成將這些值提供給神經網絡模型之前，將這些圖片數值縮放到 0 到 1 的範圍內。

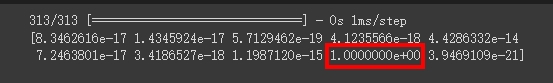
▲圖3

**3. Make prediction**

下圖4，使用tf.keras.Sequential()建立一個新的模型，其中包含原先訓練的模型和一個Softmax層。接著，使用predict()方法對測試圖像進行預測，將預測結果存儲在predictions陣列中，並印出第x個10個數字的預測數組，x是我學號的最後兩個號碼，為30。



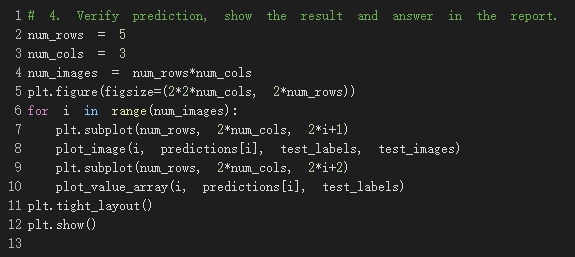
▲圖4

由下圖5可知，此圖像被預測為第8個類別，預測結果的概率為1.0。

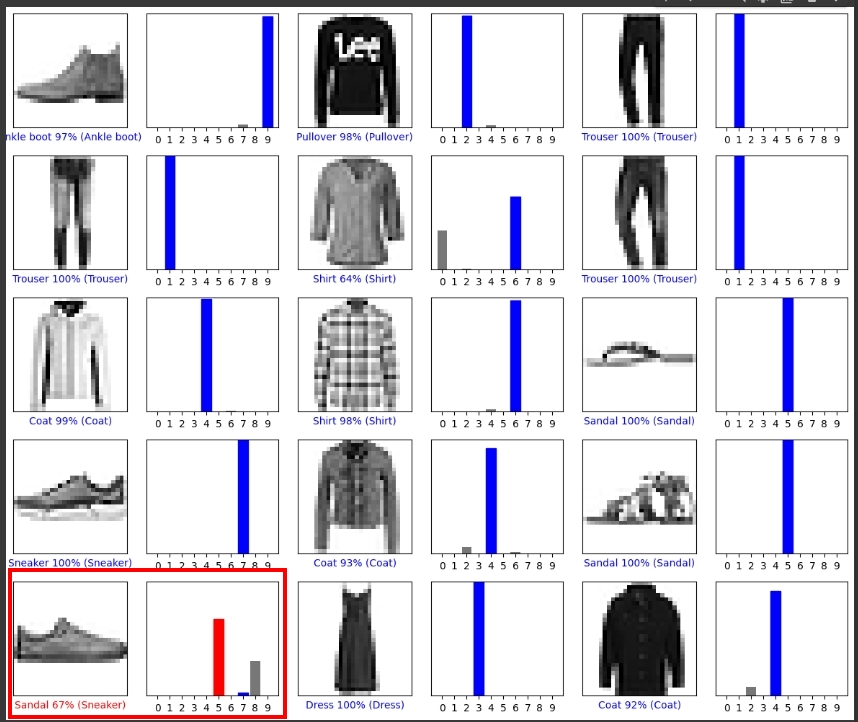
▲圖5

**4. Show your prediction results**

下圖6，程式碼從測試數據集中取出前15個圖像，並將它們以5x3的矩陣形式顯示在畫布上。每張圖像的左側顯示圖像本身，右側顯示模型對該圖像的預測結果，如下圖7。預測結果以一個條形圖的形式呈現，橫軸表示10個不同的類別，縱軸表示模型對該類別的預測概率值。若模型預測正確，該條形圖會以藍色顯示，否則會以紅色顯示。透過這個方式，可以快速驗證模型在測試數據集上的表現，並了解哪些圖像被正確預測，哪些圖像被錯誤預測。



▲圖6



**wrong prediction**

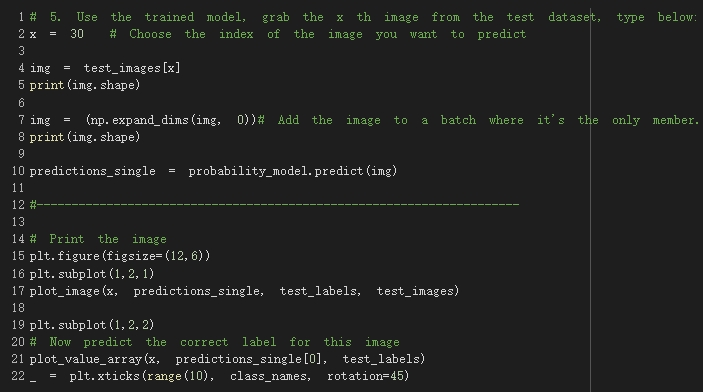
▲圖7

**5. Use the Trained model**

下圖8，對單個圖像進行預測使用測試數據集的第 x 個圖像(x為我的學號後兩碼30)。首先我們確定了x = 30表示我們要預測測試集中的第30張圖片。接下來我們取出測試數據集中的第30張圖片，印出來了它的形狀(下圖9)，發現是(28, 28)。但是我們的模型需要輸入的是一個batch的圖片，所以我們使np.expand\_dims()將這張圖片展開了一個維數，使它變成了(1, 28, 28)的形狀，表表明這是一個包含1張28x28圖片的batch。

然後我們使用probability\_model.predict()對這批圖片進行預測，並將預測結果存儲在predictions\_single中。

接下來，我們使用plt.subplot()在畫布上創建了兩個子圖。第一個子圖中，我們使用plot\_image()將第30張圖片以及它的預測結果在畫布上展示出來；第二個子圖中，我們用plot\_value\_array()函數將第30張圖片的預測結果以柱狀圖的形式展示出來，並將圖例的橫坐標標籤設置為別名。最終使用plt.xticks()函數將橫坐標的類別名稱進行旋轉。



▲圖8



▲圖9

# 問題與討論

1. **Explore the data**

根據圖2可觀察到，訓練數據集包含了60000張28x28像素的圖像，測試數據集包含了10000張28x28像素的圖像。而每個圖像都有一個對應的標籤，訓練數據集和測試數據集的標籤數量都與圖像數量相同。這些資訊對於了解數據集的規模和構成非常有用，可以幫助我們更好地理解模型訓練的過程和結果。

**2. Processthe data**

在這裡將像素值從 0 到 255 的範圍縮放到 0 到 1 的範圍內，這樣有助於神經網絡模型的訓練。因為像素值為 0 到 255 的範圍，所以將每個像素值除以 255，這樣每個像素的值都會介於 0 到 1 之間。這樣做的好處是，神經網絡的權重和偏差參數會更加穩定，減少訓練時間，提高準確率。

**3. Make prediction**

在這裡使用softmax層的新模型probability\_model來做預測。從圖5可看出，第 9 個元素的值為 1.0000000e+00，代表著模型預測該筆資料屬於類別 8的機率值為 100%。由於 softmax 函數的作用，數組中每一個元素的值都在 0 到 1 之間，且所有元素的和為 1。

在深度學習中，softmax通常被用來將模型輸出轉換成機率分佈。這個機率分佈可以被用來作為多分類問題的預測結果。softmax 函數的輸入是一個向量，並將這個向量的每個元素轉換成一個介於 0 到 1 之間的值，並且所有值的總和等於 1。這個函數通常應用在最後一層的神經元上，並且用來轉換神經網路的輸出。

softmax 可以有效地將神經網路的輸出轉換為機率分佈，因此可以方便地對多分類問題進行預測。此外，softmax 的輸出是一個機率分佈，因此可以使用交叉熵損失函數來計算預測結果的誤差，這樣可以更好地訓練模型。

**4. Show your prediction results**

由圖6可知，如果預測結果正確，則將標籤的字體顏色設為藍色，否則設為紅色。從這個輸出可以看到，有些圖片的預測結果是錯誤的，這種錯誤通常是由於模型過度簡化或過擬合而導致的，可以透過增加模型的複雜度、使用更大的數據集、調整超參數等方法來改善預測準確率。

**5. Use the Trained model**

由圖9可知，其中 img.shape 輸出為 (28, 28)，這代表這張圖片的大小是 28x28 像素，經過 np.expand\_dims 處理後的 img.shape 為 (1, 28, 28)，表示此圖片加入 batch 後的形狀。另外，也可以嘗試使用更複雜的神經網路架構，例如卷積神經網路 (Convolutional Neural Network, CNN) 來提高模型的預測能力。

# 心得與感想

透過這次的實驗，我學習到了如何使用TensorFlow建構一個簡單的神經網路，進行影像分類。在這次實驗中，我們進行了資料探索、前處理、建模、訓練、預測等步驟，透過對TensorFlow函式庫的使用，成功建構一個能夠對影像進行分類的模型。

我也學習到了如何處理資料，將資料進行標準化，將像素值縮放到0到1之間，進行資料預處理，增加模型的訓練效果。同時，我也發現在訓練模型時，學習率和訓練迭代次數都會對訓練效果產生影響，因此需要不斷地優化模型參數。

透過這次實驗，我更深入了解了TensorFlow的使用，並了解到了影像分類的基本方法和技巧，對於未來進行機器學習和深度學習的相關研究，也能夠有所幫助。

# 參考文獻

<https://www.tensorflow.org/tutorials?hl=zh-tw>

<https://zh.wikipedia.org/zh-tw/Softmax%E5%87%BD%E6%95%B0>

<https://medium.com/%E6%89%8B%E5%AF%AB%E7%AD%86%E8%A8%98/%E4%BD%BF%E7%94%A8-tensorflow-%E5%AD%B8%E7%BF%92-softmax-%E5%9B%9E%E6%AD%B8-softmax-regression-41a12b619f04>

<https://www.google.com/search?q=tensorflwo&oq=tensorflwo&aqs=chrome..69i57j35i39i650j0i131i433i512j0i67i650l2j0i131i433i512l2j69i60.3269j0j7&sourceid=chrome&ie=UTF-8>