**NLP學習筆記\_用CNN建立NLP模型**

自然語言處理Natural Language Processing(NLP)為語言學，電腦科學及人工智慧等三大科技文明領域集結之應用。

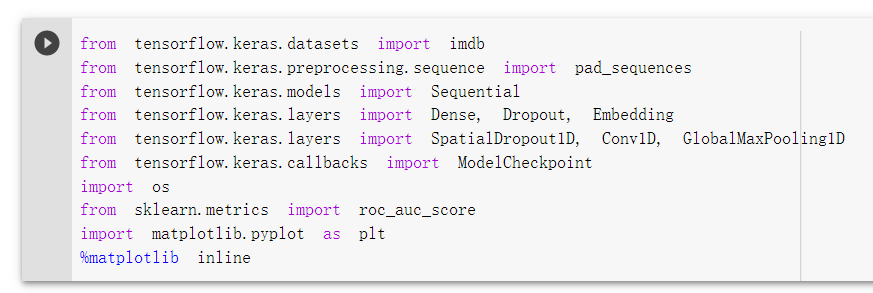
了解了自然語言處理one-hot編碼(one-hot encoding)與詞向量(Word vector)的概念後，我們要試著訓練深度學習模型來處理自然語言，本篇使用的是卷積神經網路(CNN)技術，來捕捉字跟字間的關聯特徵。

**自然語言NLP卷積神經網路模型訓練範例**

同樣的，我們使用Cola開放平台(可參考文末說明)練習此次的課題。

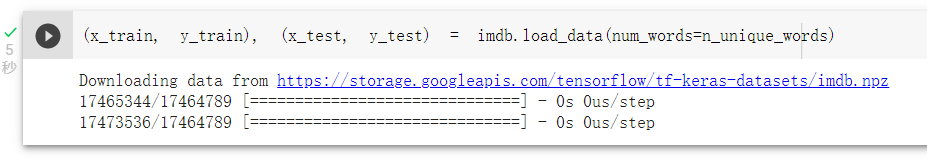
實作流程如下:

#### 步驟一， 載入模組，使用tf.Keras模組快速建立能夠分析影評內容為正評或負評的CNN模型。





#### 步驟二，載入影評資料集



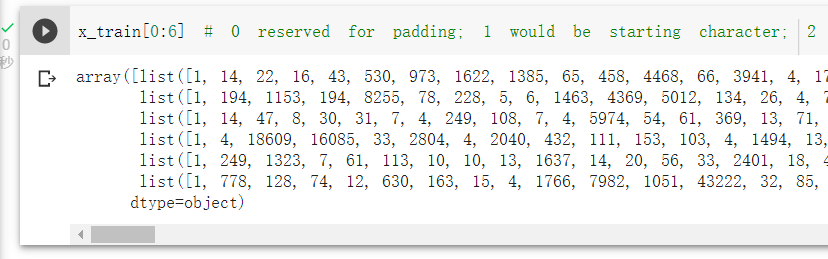
該資料集是Andrew Mass等人於2011年收集lMDb(imdb.com)的影評，作為訓練資料用，共有50000則影評，一半為訓練資料集，另一半是測試資料集。評分等級最高10顆星，二元標籤依據為:

1.4星以下為負評(=0)

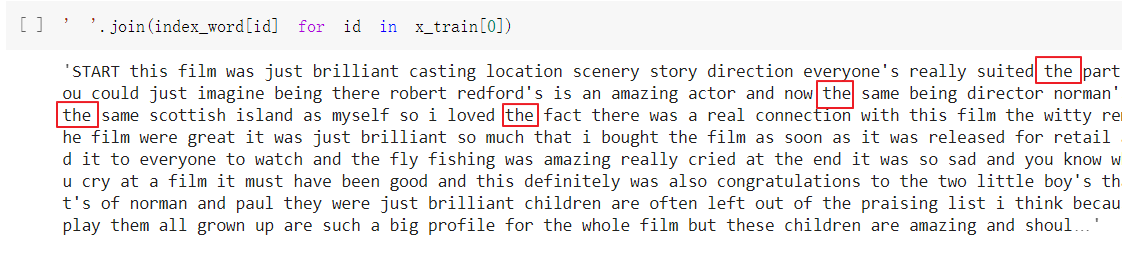
2. 7星以上為正評(=1)

3.5~6星不列入資料集

我們來看每筆影評資料長相如下。

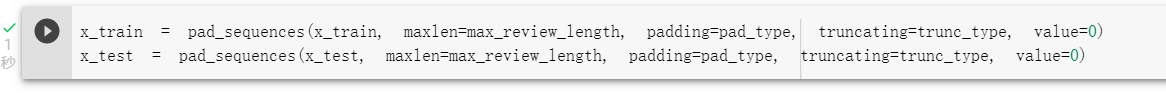
每個單字已被處理成整數格式，值越小代表出現次數越多。

那原文長怎樣呢?



原來the的值為4。

#### 步驟三，資料預處理



#### 步驟四，CNN網路架構



1.Embedding()為輸入層，能依據語料庫訓練出一個詞向量空間，但多加了一個dropout(設為drop\_embed)。

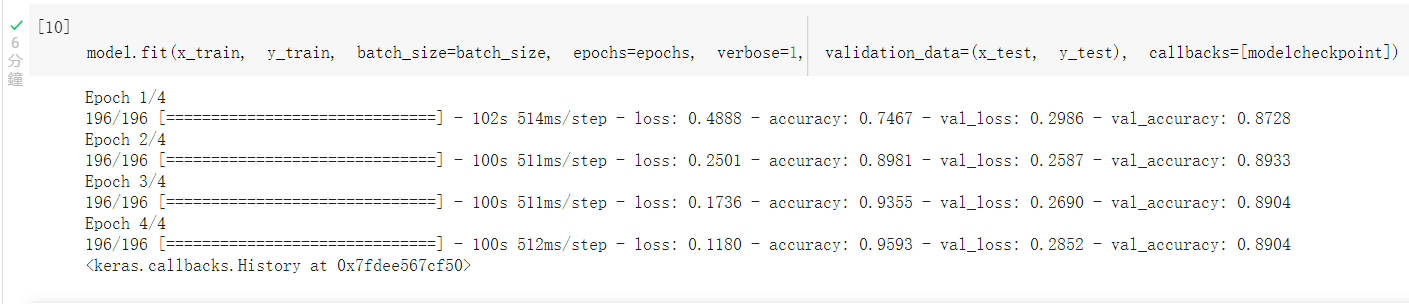
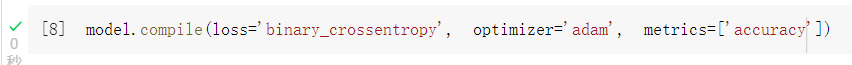
2.跟ANN不一樣省略Flatten()，由Conv1D()卷積層承接輸入層傳遞的二軸資料。

3. Conv1D()搭配relu激活函數，有256個濾鏡，每當檢視窗口步移3個字(共398步)，就輸出256個激活值，故輸出的特徵圖為256\*398。

4最大池化將256\*398縮為256\*1，即398個激活值中僅取最大值。

5.輸出層使用sigmoid激活函數，輸出維介於0到1的機率。

#### 步驟五，編譯模型及訓練模型

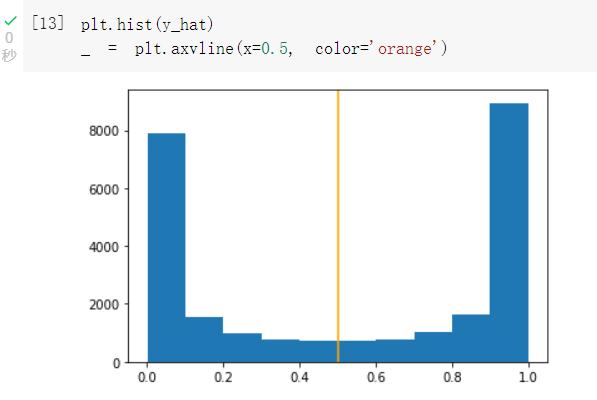


訓練結果第二周期最佳，故我們要將模型參數回朔至第二周期結束時的值。



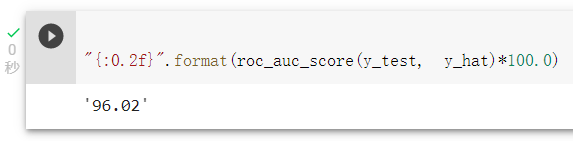
#### 步驟六，模型評估

方法一直方圖



模型對25000條影評的判斷結果有8000則(32%)的機率小於0.1，9000則(36%)大於0.9。機率超過0.5代表正評。

方法二ROC AUC分數



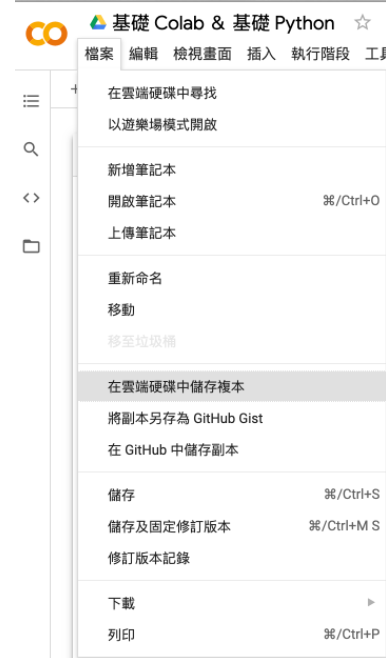
ROC AUC分數常用於二元分類的評估指標，只要算出來的分數越接近1(我們乘100比較好比較)，代表模型的預測能力越好，本案為96.02分。

#### **附錄:前置作業Colab環境準備**

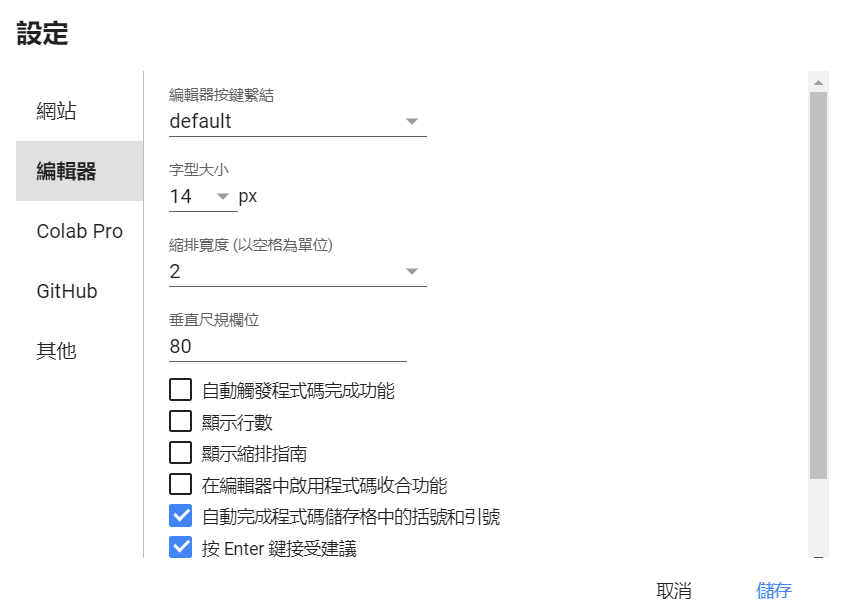
1.必須有自己的google帳號:使用自己的個人的 Google 帳號才可以執行自己 Colab上的程式。

2.先下載 Chrome瀏覽器，並登入 Google 帳號。

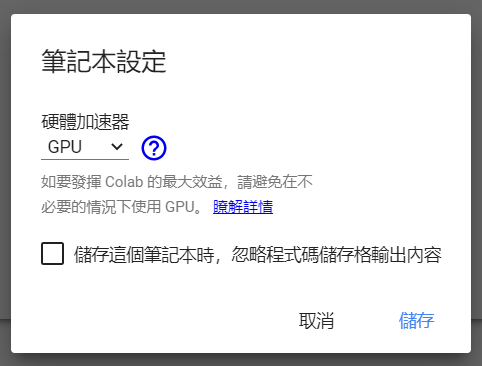
3.建立副本至個人目錄，否則不能存檔。



4.在⼯具 > 設定>編輯器中,反勾選「⾃動觸發程式碼完成功能」。



5.(必須做)設定執行階段類型。



6.療癒模式(非必須)，在⼯具 > 設定>其他, 勾選 「柯基⽝模式」和「貓咪 模式」，就有可愛的貓 貓狗狗來陪你寫程式了。



