**Final Project Report**

**Department of Management Information System master 2**

**7107029022 邱靖詒**

1. **概述 Abstract**

類型 : 文本分類

目標 : 創建一個模型來預測每個評論的惡意類型(type of toxicity)

資料集 : Toxic comment dataset

<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data>

輸入 : 評論

輸出 : 評論的惡意類型

模型 : 長短期記憶網路 (Long Short Term Memory Network, LSTM)

1. **資料集描述Data Descriptions**

資料集為來自Wikipedia的大量評論([Wikipedia's talk page](https://en.wikipedia.org/wiki/Help:Talk_pages))，每個評論有6個輸出標籤，如下表一惡性類別所示。評論可以屬於這些類別，也可以屬於這些類別的子集，這使其成為多標籤分類問題。

|  |  |
| --- | --- |
| toxic | 惡意 |
| severe\_toxic | 極度惡意 |
| obscene | 猥褻 |
| threat | 恐嚇 |
| insult | 侮辱 |
| identity\_hate | 種族歧視 |

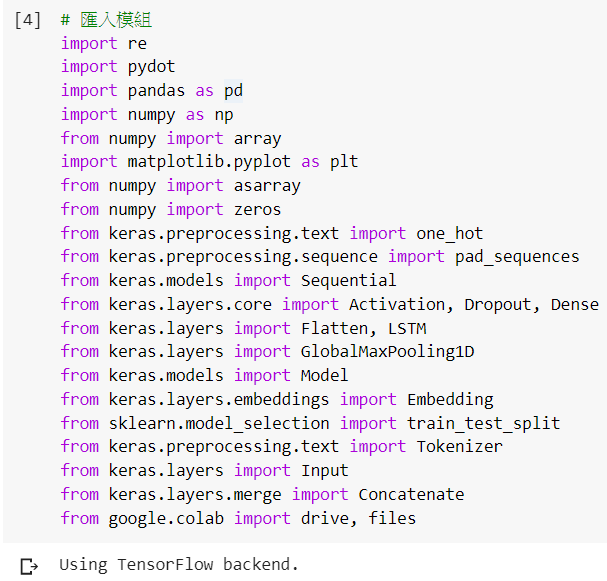
可以從下方Kaggle連結下載本文的數據集，只使用包含160,000筆評論的train.csv檔案。

<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data>

1. **檔案描述File Descriptions**

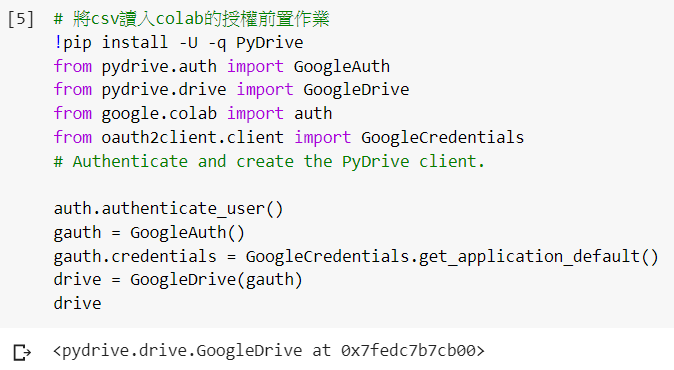
* **train.csv** - 訓練集，包含二進制標籤的評論。
* **test.csv** – 測試集，預測這些評論的惡意類別的機率，為了不使用手工標記，測試集包含一些未包含在評分中的評論。
* **sample\_submission.csv** - 格式正確的樣本提交文件。
* **test\_labels.csv** - 測試數據標籤，值為-1表示未用於評分。

1. **步驟說明Step Descriptions**
2. **匯入模組**

****

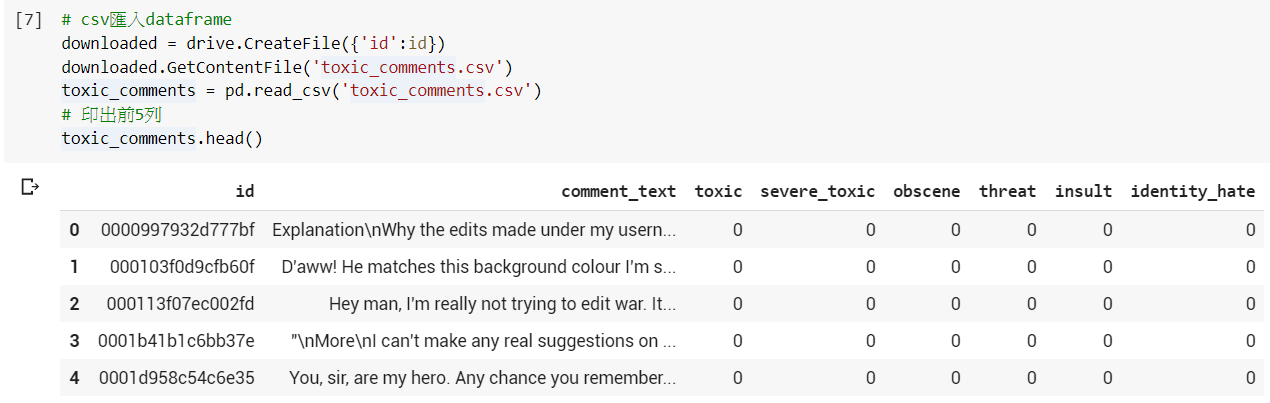
1. **讀取csv**

* colab的授權前置作業，出現提示時，點擊連結進行身分認證，允許訪問你的Google Drive，許可後將驗證碼貼製colab的驗證框中。完成驗證後，到Google Drive中的CSV文件，右鍵選擇「取得檔案共用連結」，該連結將被複製到剪貼板中，並貼到Colab中的link變數。

****

****

* 透過id來匯入csv，並建立DataFrame (toxic\_comments)，印出前5行

****

1. **檢查DataFrame**

* 查看大小

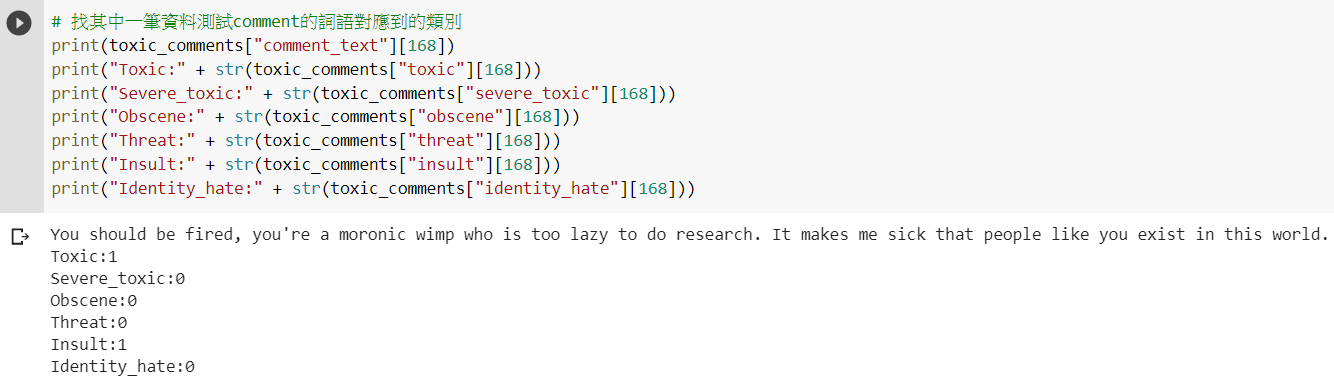
**** → 數據集包含159571條評論和8個欄位。

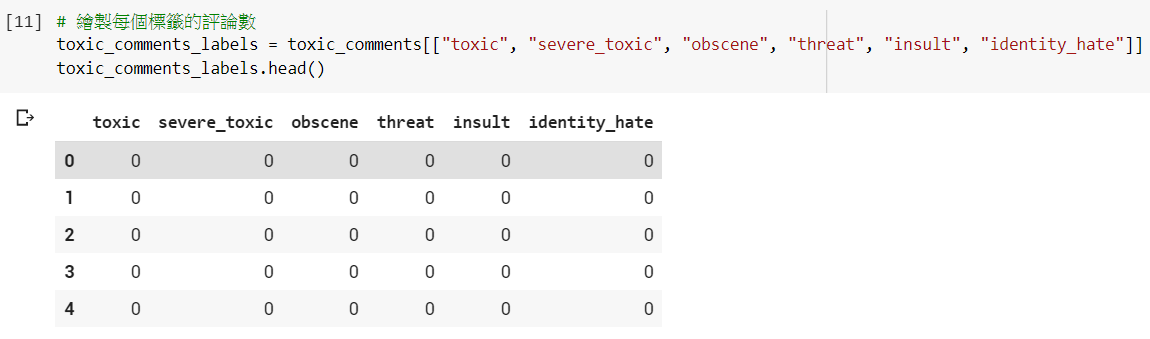
* DataFrame整理，刪除所有列包含空值或空字符串



* 找其中一筆comment測試並查看對應到的惡意類別 (以第168筆評論為例)

這顯然是惡意評論，查看與此評論相關的標籤，並為每個標籤繪製評論個數。





* 使用toxic\_comments\_labelsData Frame繪製柱狀圖，顯示不同標籤的總評論個數。

|  |  |
| --- | --- |
|  |  |

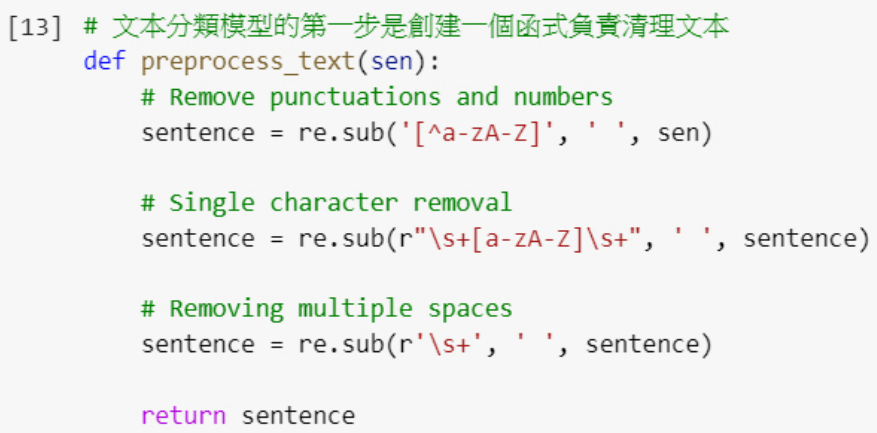
→ 可看出"toxic"評論的出現頻率最高，其次分別是"obscene"和"insult"。

已經成功分析了數據集，在下一部分，將使用該數據集創建多標籤分類模型。

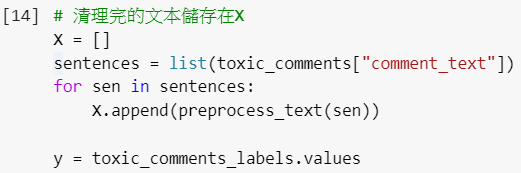
1. **文本預處理**

* 建立文本清理的函式

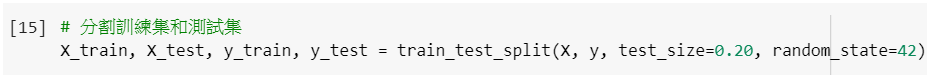
包含移除標點符號和數字、移除多個空格、移除單一字母。



創建輸入集和輸出集，輸入是DataFrame中comment\_text欄位的評論，把預處理過的評論儲存在X變量中。標籤或輸出儲存在DataFrame的toxic\_comments\_label中，使用該DataFrame的值將輸出儲存在y變量中，輸出的標籤已經是one-hot encoded向量的形式。

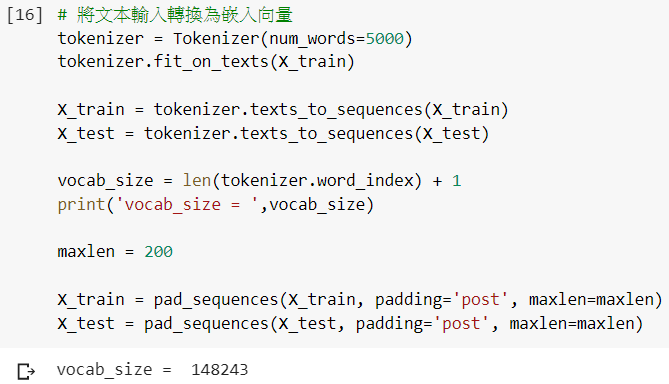


1. **分割訓練集和測試集**



1. **嵌入向量**

將文本輸入轉換為嵌入式向量。



使用GloVe預訓練模型做word embedding，把文本輸入轉換為向量輸入。

GloVe預訓練模型下載：<https://www.kaggle.com/terenceliu4444/glove6b100dtxt>

<http://nlp.stanford.edu/data/glove.6B.zip>

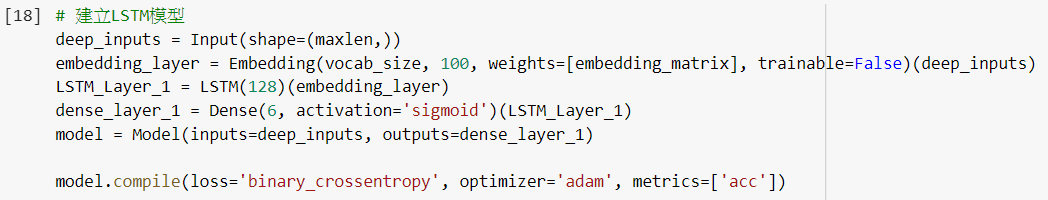


1. **創建多標籤文本分類模型**

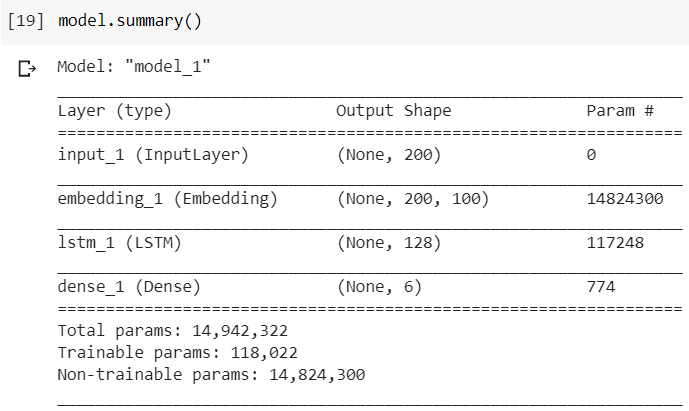
分成兩種方法，使用單個Dense輸出層和多個Dense輸出層。在第一種方法中，使用具有6個輸出的單個Dense層，並具有sigmoid激活函數和二進制交叉熵損失函數。Dense輸出層中的每個神經元代表6類輸出標籤的其中之一。Sigmoid激活函數將為每個神經元回傳0到1之間的值，如果任何神經元的輸出值大於0.5，則假定評論屬於特定一類。在第二種方法中，將每個標籤創建一個Dense輸出層，在輸出中總共包含6個Dense層，每一層都有自己的sigmoid激活函數。

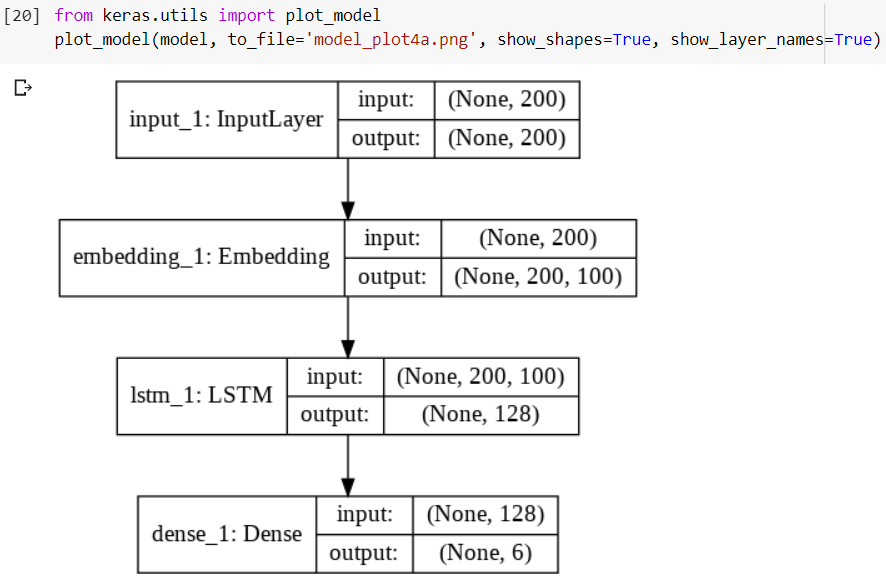
* **Model 1：Multi-lable Text Classification Model with Single Output Layers**

模型1具有一個輸入層，一個嵌入層，一個具有128個神經元的LSTM層和一個具有6個神經元的輸出層，輸出有6個標籤。

****

模型概要(神經網絡架構)如下：



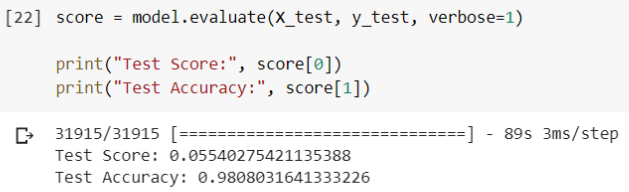


→ 可看出輸出層僅包含1個具有6個神經元的Dense層。

訓練模型(5個epochs)：



評估模型準確率：



→ 模型實現了約98％的準確率。

繪製訓練和測試集的損失和準確率曲線，以查看模型是否過擬合。

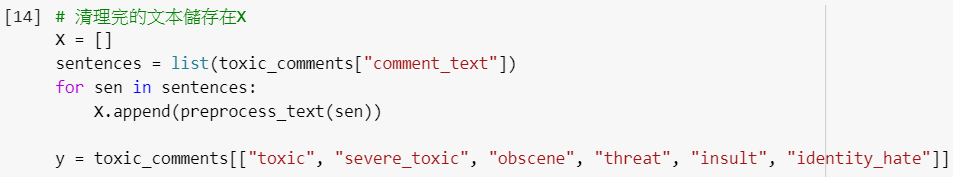


|  |  |
| --- | --- |
|  |  |

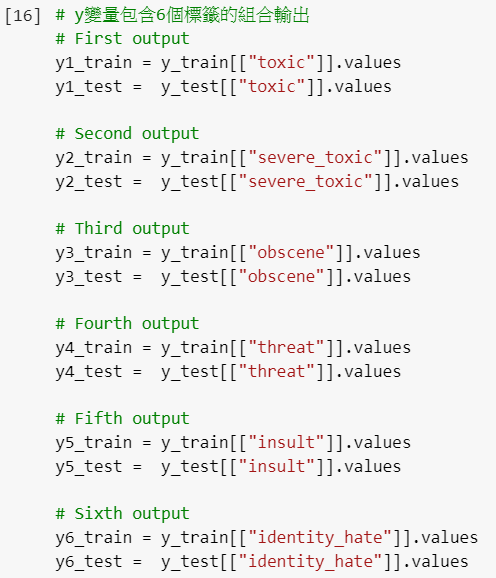
→ 模型在測試集上沒有過度擬合

* **Model 2：Multi-lable Text Classification Model with Multiple Output Layers**

創建一個多標籤文本分類模型，其中每個輸出標籤將具有一個專用的輸出Dense層。延續先前定義的預處理函式(步驟4)，接著模型創建輸入和輸出，輸入是文本評論，輸出是6個標籤。

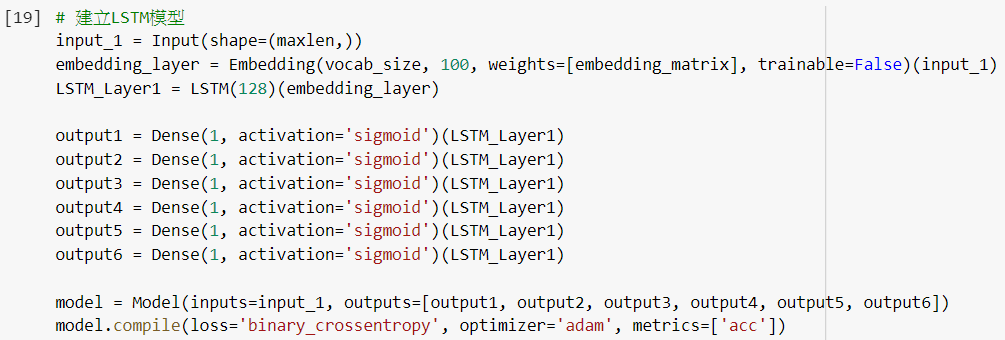


y變量包含6個標籤的組合輸出，但是，要為每個標籤創建單獨的輸出層，即創建6個變量，這些變量儲存來自訓練數據的各個標籤， 6個變量分別儲存測試數據的各個標籤值。以下程式碼把train和test各分成6個：

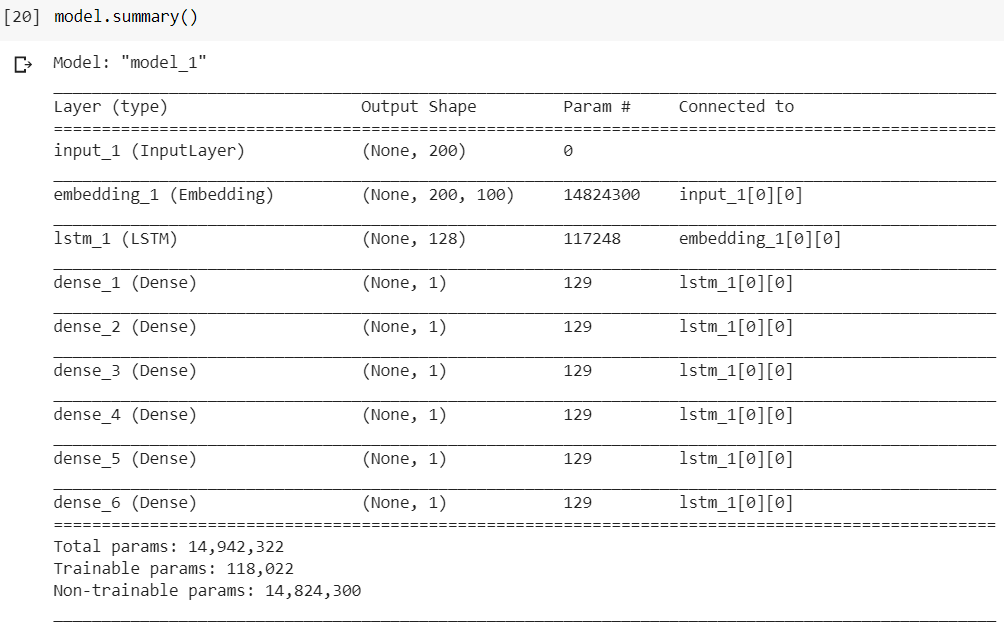


延續先前步驟6，將文本轉換成嵌入向量，並搭配GloVe word embeddings模型。

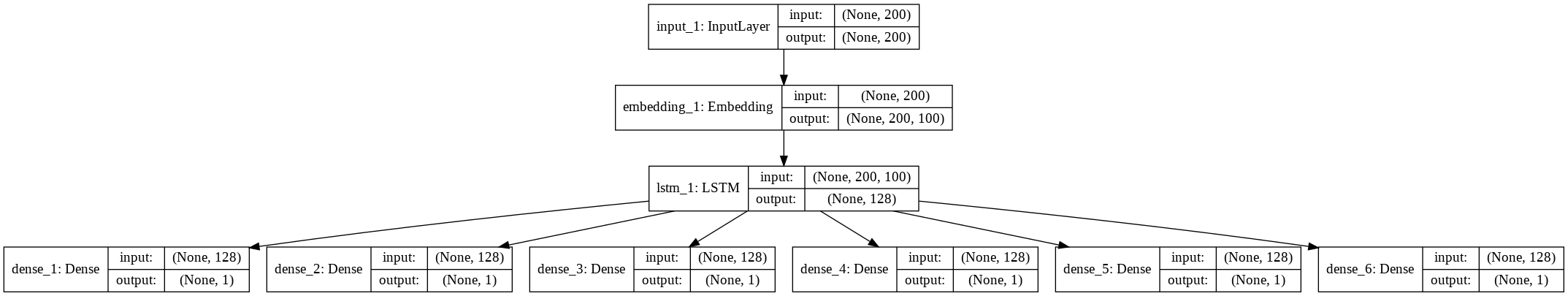
接著創建模型，模型具有一層輸入層，一層嵌入層，然後一層具有128個神經元的LSTM層。LSTM層的輸出將作為6個Dense輸出層的輸入，每個輸出層具有1個具有Sigmoid激活函數的神經元，每個輸出會預測0到1之間的整數值用來對應標籤。



模型概要(神經網路架構)如下：



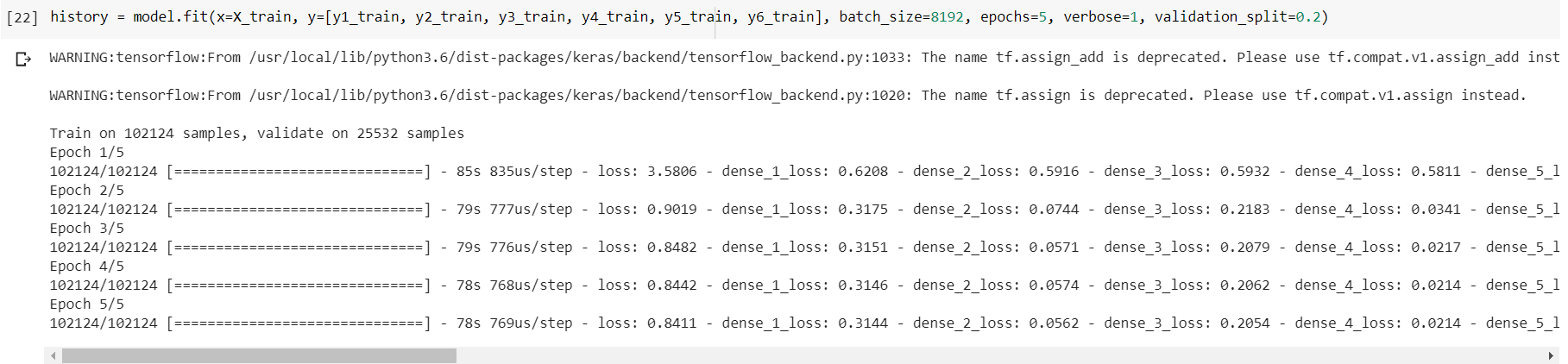




→ 可看出有6個不同的輸出層，和Model 1不同。

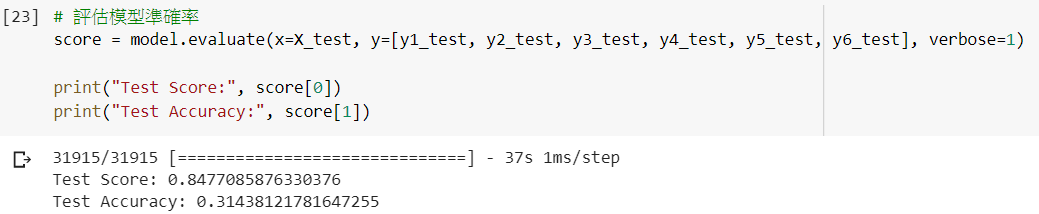
→ Model 1：具有單個輸入層的模型 VS Model 2：具有多個輸出層的模型。

訓練模型(5個epochs)：



可以看到在每個epoch，輸出所有6個Dense層都有loss、value loss、accuracy和 value accuracy。

評估模型準確率：



嘗試為5個epochs來訓練模型，但在測試集上過擬合，所以增加了batch\_size，但測試集準確率仍然不是很好。過擬合的原因可能是在這種情況下，每個標籤設置了單獨的輸出層，而增加了模型的複雜性，模型複雜度的增加通常會導致過度擬合。

→ 透過多個輸出層在測試數據集上只能達到31％的準確率。

繪製訓練和測試集的損失和準確率曲線，以查看模型是否過擬合。

|  |  |
| --- | --- |
|  |  |

從輸出中可看出，在第一個epoch之後，測試集的準確性並未收斂。而且，訓練和測試準確性之間的差異非常小，因此，模型在第一個epoch後就開始過擬合，因此得到一個性能很差模型。

1. **結論**

多標籤文本分類是最常見的文本分類問題之一。實驗了兩種用於多標籤文本分類的深度學習方法。在第一種方法中，使用具有多個神經元的單個Dense輸出層，其中每個神經元代表一個標籤label。在第二種方法中，每個神經元的標籤都會建立單獨的Dense層。實驗結果顯示，具有多個神經元的單個輸出層比多個輸出層的效果更好，可能可以更改激活函數和訓練測試集分割比例，也許可以得到更好的結果。