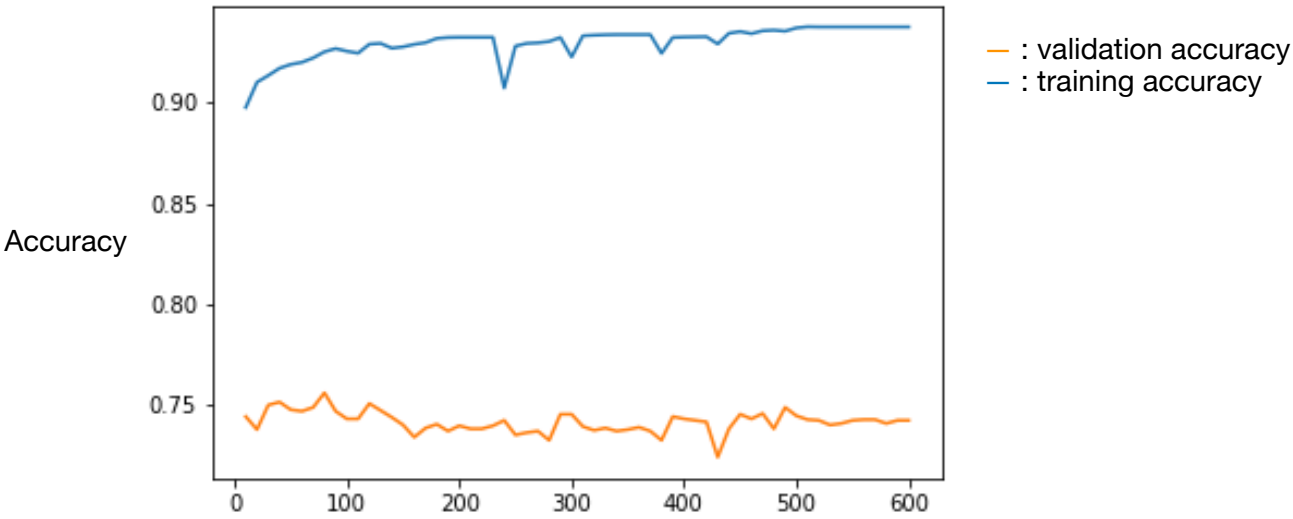
Machine Learning HW5 Report

學號：B05705001　系級：資管四　姓名：黃意芹

1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線。

   Embedding部分我用Word2Vec train 1000 iterations讓tokens轉為256維度的vector。之後把這些pretrained好的weight放入lstm model並鎖住gradient。
   LSTM model的部分有加入bidirectional，出來之後接著三層Linear中間穿插ReLU，最後一層輸出前通過Sigmoid讓每個class的預測值介於0至1。

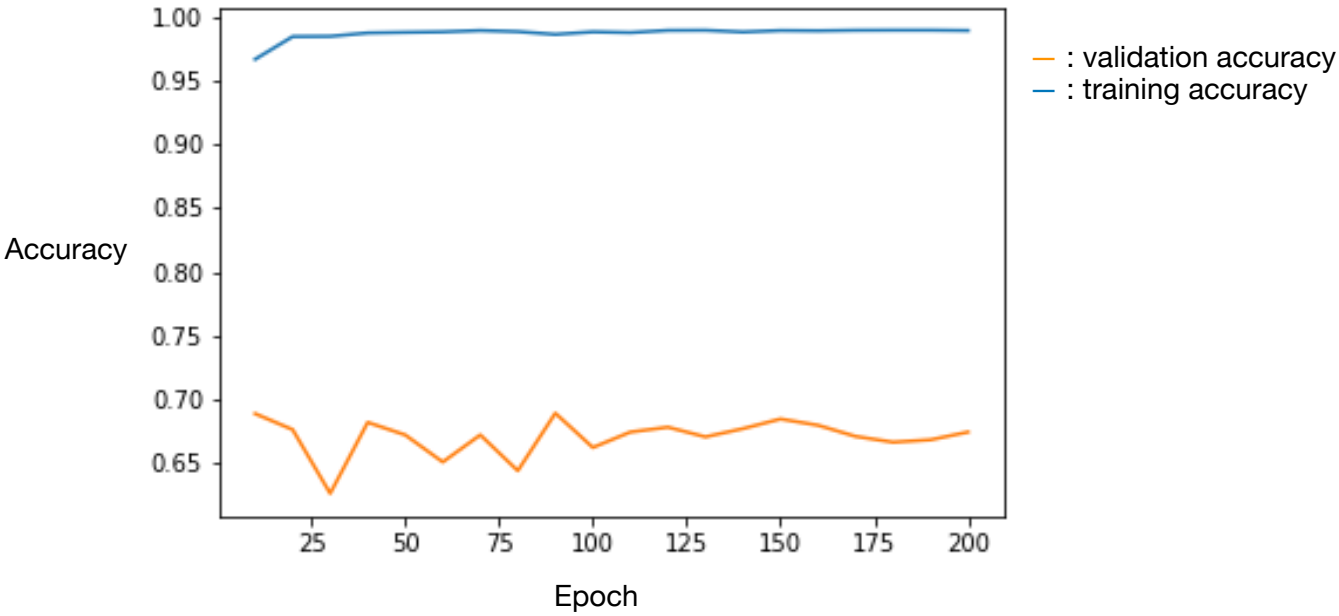


LSTM 訓練曲線

2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線。

   因為DNN只能輸入固定長度的vector，因此我先把每個句子都padding到與最長的句子等長。之後讓nn.Embedding轉為256 dim的向量，連接兩

層Linear穿插BatchNorm與LeakyReLU，最後輸出前通過Sigmoid。其實這個模型反而更難train，架了好幾個model validation accuracy都不增反降，後來試到一個比較好的雖然training很快收斂，但validation accuracy很崎嶇，上升趨勢不顯著。最後把train 100 epoch的predict結果放上kaggle分數也只有0.68。



BOW+DNN 訓練曲線

3. (1%) 請敘述你如何 improve performance（preprocess, embedding, 架構等），並解釋為何這些做法可以使模型進步。

   Preprocess：

   (1) Tokenize: 把一個句子斷成一個個詞，方便當作模型input的特徵

   (2) Remove non-alphabetic chars: 因為英文字以外的char比較沒有意義（這部分把emoji也當作有意義的英文字也許效果不錯，不過我沒有嘗試）

   (3) 去除stopwords: stopwords是幾乎在每篇文章都會出現且沒什麼意義的詞，因此去除以避免干擾model。

   (4) lemmatize tokens: 有些英文字是相同字根只是有多種變換，因此他們轉為同樣的詞一起train

   (5) 去除”url”, “@user”：我猜測這些詞是這份資料集做的隱私處理，在

非多項目裡都有出現，所以我當作stpowords把他們去除。
Embedding：
我試過三種方法，一個是把詞轉成vocab dictionary中的index後直接用nn.Embedding幫我train。第二種是先用Word2Vec pretrain好的weight，第三種是用pretrained weights且把gradient鎖住不動，結果發現第三種效果最好。我猜測可能是word2Vec本身train出來的效果就不錯，所以就用pretrained好的weight就好。
LSTM Model架構：
我曾試過多加幾層linear或維度調大，不過效果都不是很好，可能是參數過多也會使model不好。

4. (1%) 請比較不做斷詞 (e.g.,用空白分開) 與有做斷詞，兩種方法實作出來的效果差異，並解釋為何有此差別。

Kaggle Score	Private	Public
Spacy Tokenize	0.80930	0.79767
Split by space	0.73720	0.76744

上表為傳兩種作法到Kaggle的分數，可以發現不做斷詞的結果會比較做斷詞差。原因是斷詞與以空白分隔還是有些微的差異，例如Let's做tokenize會分成Let, ', s，以空白分隔就還是Let's，又例如SanFrancisco–LosAngles做tokenize會分成SanFrancisco, LosAngles，以空白分隔就還會是SanFrancisco–LosAngles。所以tokenize可以做得更精細一些。

5. (1%) 請比較 RNN 與 BOW 兩種不同 model 對於 "Today is hot, but I am happy."與"I am happy, but today is hot." 這兩句話的分數（model output），並討論造成差異的原因。

下表是兩個model對於這兩個句子output的結果。可以看到RNN對這兩個句子predict結果之間的差異比BOW大。經過前處理後，兩個句子剩下的詞分別是”today, hot, happy”與”happy, today, hot”。可能是因為RNN能夠學習到字詞的順序關係，所以對於這兩個句子比較能分出差

異。而BOW沒有分辨時序的能力，這個簡短的句子又需要加上很多padding，所以差異更不顯著。

Class 0 / Class 1	Sentence 1	Sentence 2
RNN model	0.99 / 0.37	1 / 0.11
BOW model	1 / 0.15	1 / 0.27

Sentence 1: Today is hot, but I am happy.

Sentence 2: I am happy, but today is hot.

ML HW5

t	x	z=g(z)	z_i	$f(z_i)$	z_f	$f(z_f)$	$cf(z_f)$	c'	z_0	$f(z_0)$	$y=f(z_0)h(c')$
1	$\begin{bmatrix}0\\1\\0\\3\end{bmatrix}$	3	90	1	10	1	0	3	-10	0	0
2	$\begin{bmatrix}1\\0\\1\\-2\end{bmatrix}$	-2	90	1	10	1	3	1	90	1	1
3	$\begin{bmatrix}1\\1\\1\\4\end{bmatrix}$	4	90	1	-90	0	0	4	90	1	4
4	$\begin{bmatrix}0\\1\\1\\0\end{bmatrix}$	0	90	1	10	1	4	4	90	1	4
5	$\begin{bmatrix}0\\1\\0\\2\end{bmatrix}$	2	90	1	10	1	4	6	-10	0	0
6	$\begin{bmatrix}0\\0\\1\\-4\end{bmatrix}$	-4	-10	0	110	1	6	6	90	1	6
7	$\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$	1	190	1	-90	0	0	1	90	1	1
8	$\begin{bmatrix}1\\0\\1\\2\end{bmatrix}$	2	90	1	10	1	1	3	90	1	3

final output = 3 *

2.

$$L = -\log \prod_{c \in C} P(W_{out,c} / W_{in}) = -\log \prod_{c \in C} \frac{e^{u_c}}{\sum_{i \in V} e^{u_i}} = -\sum_{c \in C} u_c + \sum_{c \in C} \log \sum_{i \in V} e^{u_i}$$

$$\frac{\partial L}{\partial W_{ij}^{T'}} = \sum_{k=1}^{V} \sum_{c=1}^{C} \frac{\partial L}{\partial u_{ck}} \frac{\partial u_{ck}}{\partial W_{ij}^{T}} = \sum_{c=1}^{C} (-\delta_{jj_c^*} + y_{cj}) \left(\sum_{k=1}^{V} W_{ki} x_k \right) \quad \text{※}$$

$$\frac{\partial L}{\partial W_{ij}^{T}} = \sum_{k=1}^{V} \sum_{c=1}^{C} \frac{\partial L}{\partial u_{ck}} \frac{\partial}{\partial W_{ij}} \left(\sum_{m=1}^{N} \sum_{l=1}^{V} W'_{mk} W_{lm} x_l \right) = \sum_{k=1}^{V} \sum_{c=1}^{C} (-\delta_{kk_c^*} + y_{ck}) W'_{jk} x_i$$

※