Machine Learning HW5 Report

學號：r07521603　系級：土木所碩二　姓名：蔡松霖

Collaborator: 程式部分有與r08521602王鈞平，r08521610鄭羽霖同學討論

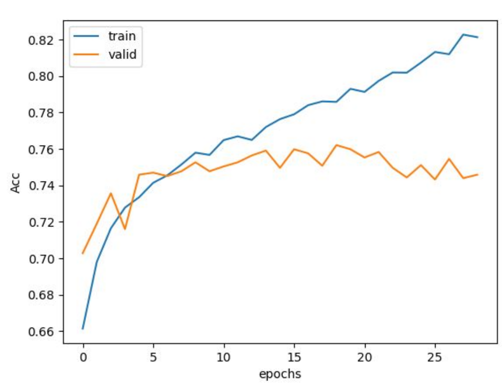
1. (1%) 請說明你實作之 RNN 模型架構及使用的 word embedding 方法，回報模型的正確率並繪出訓練曲線。

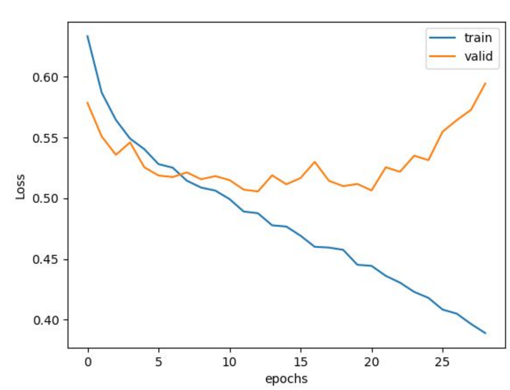
Word embedding的部分是使用訓練好的word2vec的model將每個token轉成200維的向量，並把input sequence長度padding成100，不足100以及OOV的部分都全部pad成0。

RNN的部分model先經過三層bi-LSTM(前兩層return sequence，第三層沒有; dropout p=0.3)，後面先接一層64 units的Dense layer，再加一層dropout (p=0.5)，最後輸出2個units使用cross entropy作為training的loss function。詳細RNN model架構如下圖所示：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 100, 200)          0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 100, 256)          336896
_________________________________________________________________
bidirectional_2 (Bidirection (None, 100, 256)          394240
_________________________________________________________________
bidirectional_3 (Bidirection (None, 128)               164352
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
=================================================================
Total params: 903,874
Trainable params: 903,874
Non-trainable params: 0
```

在kaggle上的成績public/private score是0.81395/0.83953，model的training history如下圖所示：(左為accuarcy，右為loss)





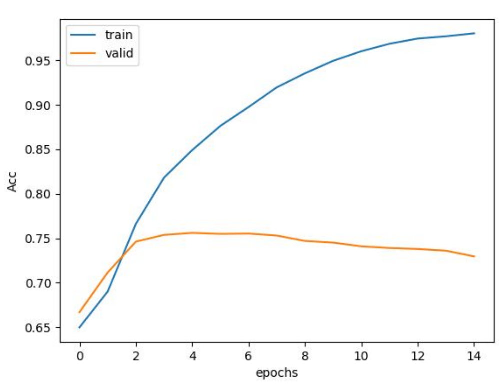
2. (1%) 請實作 BOW+DNN 模型，敘述你的模型架構，回報模型的正確率並繪出訓練曲線。

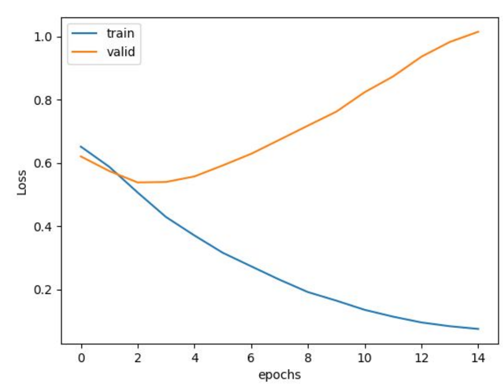
BOW的實作上我是使用keras.preprocessing.text.Tokenizer來實作，取所有token中出現頻率最高的1600個作為字庫，將tokens序列依照出現的次數轉成長度為1600的sequence。DNN model的部分前兩層的units分別是256跟64，再加一層dropout (p=0.5)，最後同RNN model輸出成2 nodes的output，使用cross entropy作為training的loss function，詳細結構如下圖所示：

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               409856
_________________________________________________________________
dense_2 (Dense)              (None, 64)                16448
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 130
=================================================================
Total params: 426,434
Trainable params: 426,434
Non-trainable params: 0
```

BOW+DNN模型所需訓練的時間比起RNN快上非常多，用CPU去跑就足夠了，準確度也還能保持在一定的水準，比RNN略差一些而已。

在kaggle上的成績public/private score是0.79534/0.80930，model的training history如下圖所示：(左為accuarcy，右為loss)





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。

(1) 首先是tokenize的部分，我覺得有一個improve performance的關鍵是使用spacy的lemmatizer做**詞形還原**，可以使不同詞性的字詞歸成同一個，更集中去訓練字詞之間的關係。
(2) **Stopwords的去除**，可以些微幫助，使用bag-of-words的模型進步的比較明顯，關鍵在於有很多字詞出現的頻率很高，但其實並不會影響到最後的判斷，在兩類的句子出現的比例差不多。在去除的時候，這次作業的題目最好要保留像是”not”等具有否定的一些stopwords，不然可能會造成語意整個相反，造成誤判。
(3) **Remove Punctuation跟數字**，幫助蠻大的，原理應該類似stopwords，在各個句子間出現的頻率蠻大的。移除掉標點符號可以讓tokenize的結果更乾淨，像是表情符號demojize後會有“：”夾在兩側。
(4) Word2vec **min count調大一點點**，把出現頻率較少的字詞排除，可以有些微幫助，應該是因為出現一兩次的token，學習的依據較侷限，全部加進去學可能會比較影響學習的成果。
(5) Word2vec 的**sample參數不能設太小**，我看他官方網站寫說useful range是(0, 1e-5)，實際上套用時，RNN完全train不起來，後來選用6e-5，accuracy才有上去，但是這部分因為時間關係沒有再嘗試更大的值。不過因為這些既有的嘗試讓我注意到word2vec model有沒有train好比起RNN model部分參數的設定更為重要。
(6) RNN **LSTM選用Bidirectional**的方式實作效果較佳，因為可以從雙向來學習字詞之間的關係。

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。

   Word2Vec model跟RNN model在一樣的參數設定下，不做斷詞用空白分開的效果會比起有做斷詞還差(public/private score：空白分開：0.793/0.812；斷詞分開：0.814/0.840)，主要的差別應該在於有些專有名詞像是U. K.如果以空白分開成 U. 跟 K. 就會失去原來的意思，有做斷詞可以保有U. K. 代表國家的意義。因此做好斷詞還是能保留一些字詞的原意。

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。

兩句經過tokenize處理的結果如下所示：
[today, be, hot, but, -pron-, be, happy]
[-pron-, be, happy, but, today, be, hot]

兩句的結果預測不論方法都會是預測出第0類(not offensive)，但是預測的分數有些微差異，如下表所示：

	"Today is hot, but I am happy."	"I am happy, but today is hot."
RNN	[0.9921089 0.00789109]	[0.9923666 0.00763344]
BOW + DNN	[0.9360585 0.06394146]	[0.9360585 0.06394146

可以看到RNN model出來的結果有些微差異，而BOW+DNN model則是兩句的結果一模模一樣樣。造成這個差異主要跟兩個model判斷依據的特徵有關，bag-of-words model 吃的特徵是針對字數做統計，因此假如組成句子的字詞一樣，不管排列的順序為何，所得到的結果都會一樣；而RNN model則會考慮到詞彙之間的排列順序，因此預測結果會有所差異。

Math Problem

1.

$$w = [0, 0, 0, 1], b = 0$$
$$w_i = [100, 100, 0, 0], b_i = -10$$
$$w_f = [-100, -100, 0, 0], b_f = 110$$
$$w_o = [0, 0, 100, 0], b_o = -10$$

$$z = w \cdot x + b$$
$$z_i = w_i \cdot x + b_i$$
$$z_f = w_f \cdot x + b_f$$
$$z_o = w_o \cdot x + b_o$$

$$c = f(z_i)g(z) + cf(z_f)$$

$$y = f(z_o)\ h(c)$$

以下計算均取到小數第四位(跟取到第一位結果一樣，為求簡潔以下均以取到第一位表示)

Start from $c = 0$

$$x_1 = [0, 1, 0, 3]$$
$$z = w \cdot x_1 + b = 3$$
$$z_i = w \cdot x_1 + b_i = 90$$
$$z_f = w \cdot x_1 + b_f = 10$$
$$z_o = w \cdot x_1 + b_o = -10$$
$$f(z_i) = 1.0, f(z_f) = 1.0, f(z_o) = 0.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 3.0$$
$$y_1 = f(z_o) \cdot c' = 0.0$$

$$x_2 = [1, 0, 1, -2]$$
$$z = w \cdot x_2 + b = -2$$
$$z_i = w \cdot x_2 + b_i = 90$$
$$z_f = w \cdot x_2 + b_f = 10$$
$$z_o = w \cdot x_2 + b_o = 90$$
$$f(z_i) = 1.0, f(z_f) = 1.0, f(z_o) = 1.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 1.0$$
$$y_2 = f(z_o) \cdot c' = 1.0$$

$$x_3 = [1, 1, 1, 4]$$
$$z = w \cdot x_3 + b = 4$$
$$z_i = w \cdot x_3 + b_i = 190$$
$$z_f = w \cdot x_3 + b_f = -90$$
$$z_o = w \cdot x_3 + b_o = 90$$
$$f(z_i) = 1.0, f(z_f) = 0.0, f(z_o) = 1.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 4.0$$
$$y_3 = f(z_o) \cdot c' = 4.0$$

$$x_4 = [0, 1, 1, 0]$$
$$z = w \cdot x_4 + b = 0$$
$$z_i = w \cdot x_4 + b_i = 90$$
$$z_f = w \cdot x_4 + b_f = 10$$
$$z_o = w \cdot x_4 + b_o = 90$$
$$f(z_i) = 1.0, f(z_f) = 1.0, f(z_o) = 1.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 4.0$$
$$y_4 = f(z_o) \cdot c' = 4.0$$

$$x_5 = [0, 1, 0, 2]$$
$$z = w \cdot x_5 + b = 2$$
$$z_i = w \cdot x_5 + b_i = 90$$

$$z_f = w \cdot x_5 + b_f = 10$$
$$z_o = w \cdot x_5 + b_o = -10$$
$$f(z_i) = 1.0, f(z_f) = 1.0, f(z_o) = 0.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 6.0$$
$$y_5 = f(z_o) \cdot c' = 0.0$$

$$x_6 = [0, 0, 1, -4]$$
$$z = w \cdot x_6 + b = -4$$
$$z_i = w \cdot x_6 + b_i = -10$$
$$z_f = w \cdot x_6 + b_f = 110$$
$$z_o = w \cdot x_6 + b_o = 90$$
$$f(z_i) = 0.0, f(z_f) = 1.0, f(z_o) = 1.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 6.0$$
$$y_6 = f(z_o) \cdot c' = 6.0$$

$$x_7 = [1, 1, 1, 1]$$
$$z = w \cdot x_7 + b = 1$$
$$z_i = w \cdot x_7 + b_i = 190$$
$$z_f = w \cdot x_7 + b_f = -90$$
$$z_o = w \cdot x_7 + b_o = 90$$
$$f(z_i) = 1.0, f(z_f) = 0.0, f(z_o) = 1.0$$
$$c' = f(z_i)g(z) + cf(z_f) = 1.0$$
$$y_7 = f(z_o) \cdot c' = 1.0$$

$$x_8 = [1, 0, 1, 2]$$
$$z = w \cdot x_8 + b = 2$$
$$z_i = w \cdot x_8 + b_i = 90$$
$$z_f = w \cdot x_8 + b_f = 10$$
$$z_o = w \cdot x_8 + b_o = 90$$

$f(z_i) = 1.0, f(z_f) = 1.0, f(z_o) = 1.0$
$c' = f(z_i)g(z) + cf(z_f) = 3.0$
$y_8 = f(z_o) \cdot c' = 3.0$

$y = [0, 1, 4, 4, 0, 6, 1, 3]$

2.

Ref: http://www.claudiobellei.com/2018/01/06/backprop-word2vec/

From chain rule we got

$$\frac{\partial L}{\partial W'_{ij}} = \sum_{k=1}^{V} \sum_{c=1}^{C} \frac{\partial L}{\partial u_{c,k}} \frac{\partial u_{c,k}}{\partial W'_{ij}}$$

and

$$\frac{\partial L}{\partial W_{ij}} = \sum_{k=1}^{V} \sum_{c=1}^{C} \frac{\partial L}{\partial u_{c,k}} \frac{\partial u_{c,k}}{\partial W_{ij}}$$

Calculate $\partial L / \partial u_{c,j}$

$$\frac{\partial L}{\partial u_{c,j}} = -\delta_{jj_c^*} + y_{c,j}$$

where δ_{jj^*} is a Kronecker delta, it is equal to 1 if $j = j^*$, otherwise it is equal to zero.

We get

$$\frac{\partial L}{\partial W'_{ij}} = \sum_{k=1}^{V} \sum_{c=1}^{C} \frac{\partial L}{\partial u_{c,k}} \frac{\partial u_{c,k}}{\partial W'_{ij}} = \sum_{c=1}^{C} \frac{\partial L}{\partial u_{c,j}} \frac{\partial u_{c,j}}{\partial W'_{ij}} = \sum_{c=1}^{C} (-\delta_{jj_c^*} + y_{c,j}) \left(\sum_{k=1}^{V} W_{ki} x_k \right)$$

$$\frac{\partial L}{\partial W_{ij}} = \sum_{k=1}^{V} \sum_{c=1}^{C} \frac{\partial L}{\partial u_{c,k}} \frac{\partial}{\partial W_{ij}} \left(\sum_{m=1}^{N} \sum_{l=1}^{V} W'_{mk} W_{lm} x_l \right) = \sum_{k=1}^{V} \sum_{c=1}^{C} (-\delta_{kk_c^*} + y_{c,k}) W'_{jk} x_i$$

Finally, we have

$$\frac{\partial L}{\partial W'_{ij}} = \sum_{c=1}^{C}(-\delta_{jj_c^*} + y_{c,j}) \left(\sum_{k=1}^{V} W_{ki}x_k\right)$$

and

$$\frac{\partial L}{\partial W_{ij}} = \sum_{k=1}^{V}\sum_{c=1}^{C}(-\delta_{kk_c^*} + y_{c,k})W'_{jk}x_i$$