學號：B06901180　系級：電機三　　　姓名：鄭謹譯

1. (1%) 請說明你實作的RNN的模型架構、word embedding 方法、訓練過程(learning curve)和準確率為何？ (盡量是過public strong baseline的model)

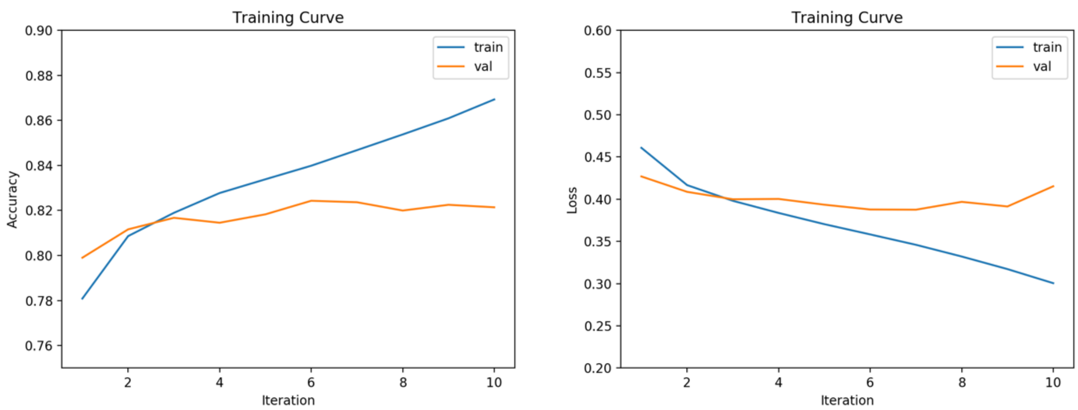
RNN架構如下：

```
LSTM_Net(count #200000
  (embedding): Embedding(55705, 250)
  (lstm): LSTM(250, 150, batch_first=True, bidirectional=True)
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=300, out_features=1, bias=True)
    (2): Sigmoid()
  )
)
```

Word Embedding：

w2v有嘗試skip gram和cbow兩種方式，其中skip gram較cbow的準確率好2-3%，以下均使用skip gram說明。另外在add padding時均把unk放在句子最前方。

Training Curve：



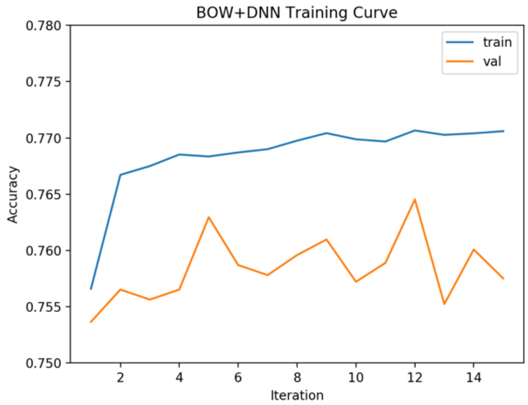
準確率：

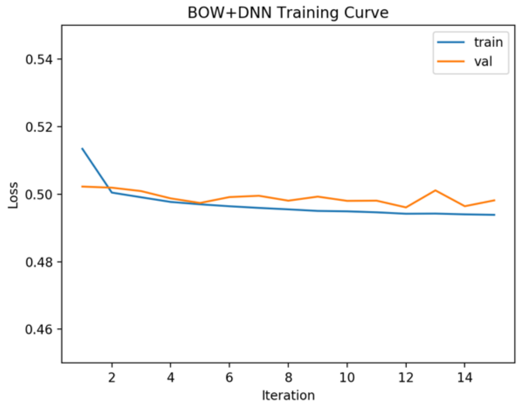
Training 0.86930304　　　Val 0.82429339

predict_ensemble.csv　　0.82496
13 days ago by b06901180_
Strong

註：這筆testing data是單一筆資料只是檔名寫死故誤植為ensemble

2. (2%) 請比較BOW+DNN與RNN兩種不同model對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的分數(過softmax後的數值)，並討論造成差異的原因。






```
------------------BOW+DNN---Score------------------------
today is a good day, but it is hot :        0.7047868932
today is hot, but it is a good day :        0.7047868932
---------------------RNN--Score--------------------------
today is a good day, but it is hot :        0.7627845112
today is hot, but it is a good day :        0.8912403256
```

上圖為BOW+DNN的結果，下圖為RNN的結果

BOW (Bag of Words)是將句子文字裝成袋子來呈現，因此不在意單字的前後文，就這兩筆資料而言，轉換為BOW都是today + 2*is + hot + but + it +a + good + day，因此結果相同，而BOW判斷正負面通常都以單字來歸類，這些字都是中性字，附帶一個Good，因此分數大約比中間高些。而就RNN而言，會考慮前後文，因此語序調換就會有不同意思，推測因為會考慮前後文的關係所以分數比較高一些，可能在model裡面這些組合有特別的意思因此第二句有0.89分。

3. (1%) 請敘述你如何 improve performance（preprocess、embedding、架構等等），並解釋為何這些做法可以使模型進步，並列出準確率與improve前的差異。（semi supervised的部分請在下題回答）

首先將Unlabeled training data加入Training Data

(1)Preprocess：

把句子中出現” x” 的地方刪掉，x是表情符號，移除表情符號干擾。犧牲部分x開頭的字，但英文中這種單字不多故忽略。為何不刪除 “ x ” 來減少x開頭單字被刪除的原因是因為data裡有很多連續的表情符號，例如xx,xx x， “ x ” 無法處理。程式碼如下：

```
lines = [line.strip('\n').replace(' x',' ').replace(' x',' ').replace(' x',' ').split(' ') for line in lines]
```

以下參考nltk的方法，由於不能使用，故自己刻簡略版

將第三人稱單數動詞還原，去除es,s；將名詞複數還原，去除es,s；將過去是動詞還原，去除ed。不考慮不規則變化，誤刪者忽略，如red變成r。程式碼如下：

```
x[i] = [chrs.replace('ed ',' ') for chrs in x[i]]
x[i] = [chrs.replace('es ',' ') for chrs in x[i]]
x[i] = [chrs.replace('s ',' ') for chrs in x[i]]
```

將Stop words刪除，通常是一些助詞或時間複詞，無關判斷，列表如下

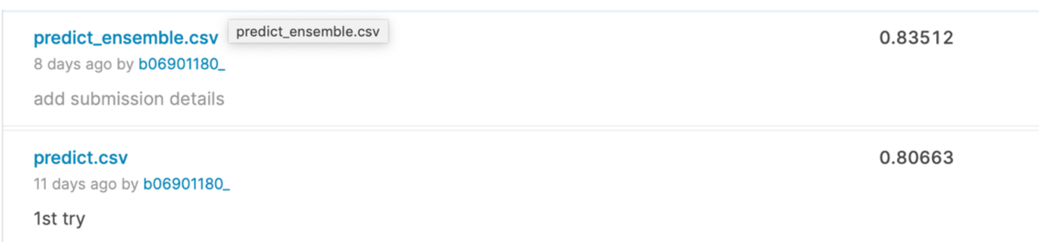
Stop_words =[" i ", " me ", " my ", " myself", " we", " our", " ours", " ourselves ", " you ", " your ", " yours ", " yourself ", " yourselves ", " he ", " him ", " his ", " himself ", " she ", " her ", " hers ", " herself ", " it ", " its ", " itself ", " they ", " them ", " their ", " theirs ", " themselves ", " what ", " which ", " who ", " whom ", " this ", " that ", " these ", " those ", " am ", " is ", " are ", " was ", " were ", " be ", " been ", " being ", " have ", " has ", " had ", " having ", " do ", " does ", " did ", " doing ", " a ", " an ", " the ", " and ", " but ", " if ", " or ", " because ", " as ", " until ", " while ", " of ", " at ", " by ", " for ", " with ", " about ", " against ", " between ", " into ", " through ", " during ", " before ", " after ", " above ", " below ", " to ", " from ", " up ", " down ", " in ", " out ", " on ", " off ", " over ", " under ", " again ", " further ", " then ", " once ", " here ", " there ", " when ", " where ", " why ", " how ", " all ", " any ", " both ", " each ", " few ", " more ", " most ", " other ", " some ", " such ", " no ", " nor ", " not ", " only ", " own ", " same ", " so ", " than ", " too ", " very ", " s ", " t ", " can ", " will ", " just ", " don ", " should ", " now "]

經過Preprocessing後，可以減少words to vectors 後的數量，讓判斷更準確

(2)Add paddings時將用來補齊長度unknown加到前面，若RNN遺忘時通常遺忘掉不重要的unknown而非重要的文字。

(3)Train w2v時使用skip gram和cbow兩種方法增加變異性，使ensemble的結果更好

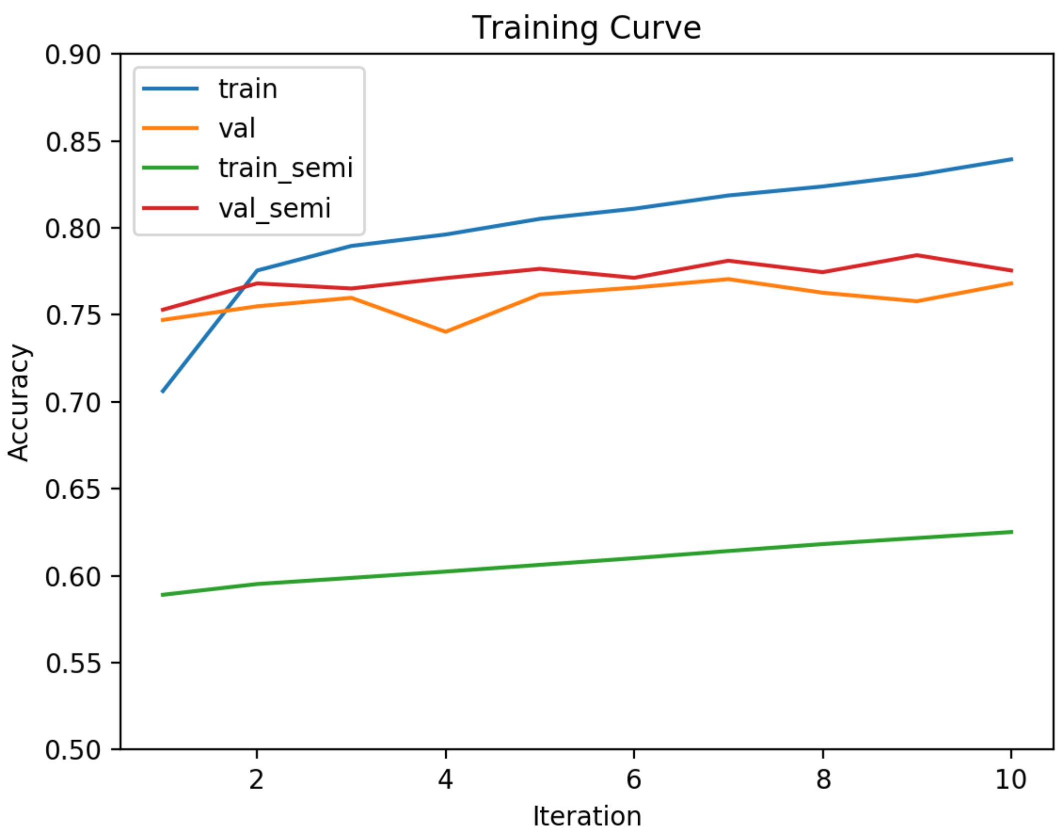
(4)Ensemble：使用不同的sentence lens，搭配skip gram、cbow兩種方式，產生 lens=25、30、35的model，skip gram各四個，cbow各二個，lens=20 skip gram二個，cbow一個，將這些model組合成結果，每個model配以不同權重，權重以各1為initial，使用簡單DNN train出。



以上結果分別為preprocessing前後的差別

4. (2%) 請描述你的semi-supervised方法是如何標記label，並比較有無semi-supervised training對準確率的影響並試著探討原因（因為 semi-supervise learning 在 labeled training data 數量較少時，比較能夠發揮作用，所以在實作本題時，建議把有 label 的training data從 20 萬筆減少到 2 萬筆以下，在這樣的實驗設定下，比較容易觀察到semi-supervise learning所帶來的幫助）。

先train一筆training data =20000，validation data = 2000 的 model，將20萬筆unlabeled data丟入，判斷其得分，若分數大於0.9即標上label 1；分數小於0.1即標上label 0，再將新標的data加入training，以同樣2000筆資料做validation。



圖中semi-supervised validation 0.78111328 ，無supervised validation 0.77050781 ，差距1%，結果算是有所提升。可能原因是因為labeled的資料量少，加入大量透過semi-supervised標註的data可以增加training data提升效果。