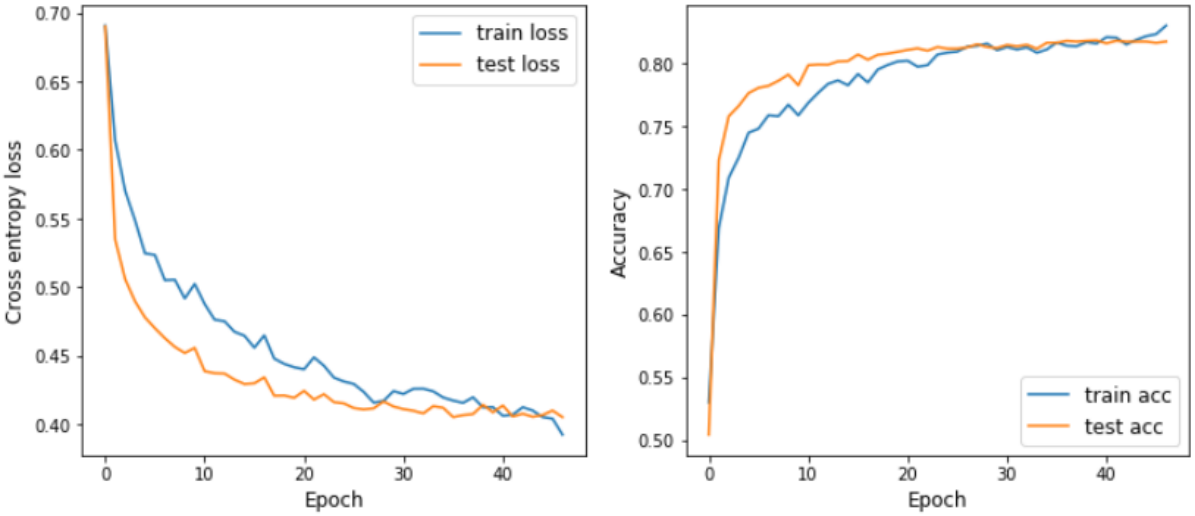
學號：B03902130 系級： 資工四 姓名：楊書文

註：報告中的 Epoch 其實應為 Iteration，因為我 train 時是 random sample。一個 Iteration 為 random sample 100 次。

1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？ (Collaborators: 無)

   答：我的模型架構主要有三個部分，分別為 pre-train 的 embedding layer、三層雙向且 hidden_size=128 的 GRU、和三層遞減的 DNN，最後降維到兩個 cell，分別代表 0 和 1，過 softmax 後做 cross entropy。Pre-train embedding layer 使用了 gensim 的 word2vec，將 training data 中出現次數少於十次的詞砍掉後(剩 8859 個詞)，交給 word2vec 算出每個詞代表的 100 維向量，embedding layer 的 shape 為 (8859, 100)。每句英文句子轉成整數 token 後，先過 embedding layer 得到 (sequence_length, batch, 100)，接著過 GRU 得到 (num_layers * num_directions, batch, hidden_size)，前兩個維度 transpose 並 reshape 成 (batch, -1) 後可得(batch, num_layers * num_directions * hidden_size)，將第二個維度當作 feature，過 DNN 先降到 128，再降到 64，最後降到 2，取 softmax 後做 categorical cross entropy。



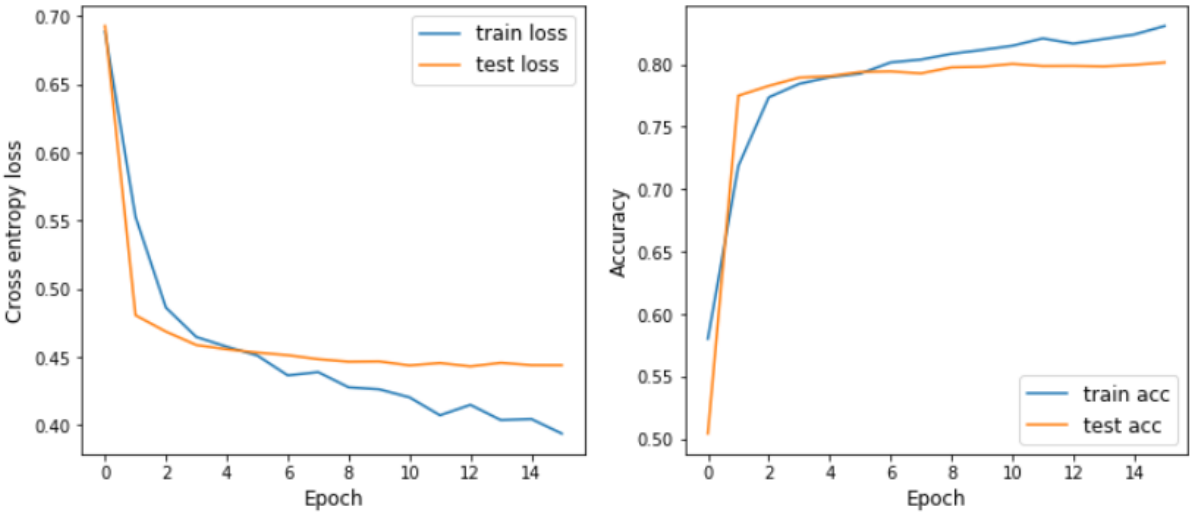
   訓練過程如上圖。在 tune model 時發現 pre-train embedding 滿重要的，如果 embedding layer 是 random initialized 的話 acc 只能剛好過 0.80，有 pre-train 的話則可以達到 0.818。接著取不同 validation data 訓練出來兩個 model 做 ensemble，acc 可以到 0.822。後來發現不砍標點符號的效果較好，可達 0.82188。最後將達到 0.82188 的 model 做 semi supervised 後可達 0.824。

MODEL	PUBLIC	PRIVATE
single, w/o punctuation	0.81891	0.81742
ensemble, w/o punctuation	0.82215	0.82184
single, w punctuation	0.82188	0.81995
single, w punctuation, semi supervised	0.82415	0.82231

2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？

(Collaborators: 無)

答：我的 BOW 模型架構中，會將每句話表示為一個字典大小維度的向量，其中每個 element 都為 0 或 1，與一般 BOW 會計算句中每個單字的數量不同，我直接僅記錄該句是否有某個單字。 我將單字出現次數少於 15 的砍掉，剩下 7174 個單字。因此每個 batch 的 shape 為 (batch, 7174)。接著過 DNN 降維到 1024，再降維到 512，最後降到 1 維，取 binary cross entropy。



使用 BOW 雖然可以過 simple baseline，但勉強過 0.8 之後就再怎樣都上不去了，而且會呈現很明顯的 overfitting。

MODEL	PUBLIC	PRIVATE
Bow	0.80006	0.79632

3. (1%) 請比較 bag of word 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"這兩句的情緒分數，並討論造成差異的原因。

(Collaborators: 無)

答：情緒分數越高代表越正向。從表格可以看出來，不同單詞順序造成的語意變化，RNN 可以分別，但 BOW 無法。甚至因此這兩句使用到的詞完全一模一樣，所以對於 BOW 來說這兩句話完全相同，因此出來的情緒分數也一樣。這是因為 BOW 中一句話的 embedding 和單詞順序完全無關，因此 DNN 無法認出其中差異。而 RNN 採取一個詞一個詞考慮順序的 embedding，因此不同的順序可以 embed 出不同的向量，最後的 DNN 就能以此辨認出不同的情緒。

MODEL	SENTENCE 1	SENTENCE 2
RNN	0.31528	0.94869
BOW	0.86891	0.86891

4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。

(Collaborators: 無)

答：做報告時才嚇一跳標點符號都留著的效果竟然比砍掉好很多(見下表)。原本

砍掉標點符號的 model 我怎樣都上不了 0.82 (而且還都 train 很久)，最後靠著 ensemble 才到 0.822。結果標點符號都留著，才 train 50 個 Epoch 就能有 0.82188。原本覺得標點符號會增加不必要的資訊量，而且正面句子和負面句子也都會使用那些標點符號，沒有哪個標點符號是僅限開心時使用的，因此似乎不是判斷情緒時有用的資訊。但實驗出來就是和我想的很不一樣。猜想可能是因為 data 中的推文都滿長的，常是多句話組成。加入標點符號有利於 model 判斷句子與句子間的分隔，不會把整個推文都當作同一句話，進而更能理解推文的語意與情緒。

MODEL	PUBLIC	PRIVATE
w/o punctuation	0.81891	0.81742
w punctuation	0.82188	0.81995

5. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-surpervised training 對準確率的影響。
(Collaborators: 無)
答：我的 semi supervised 是先用達 0.82188 (保留標點符號) 的 model 去 predict no-label data。取情緒分數高於 0.75 的和低於 0.25 的，且取的 no-label data 量與 label data 相近。從表格可以看到使用 semi supervised 有不錯的進步。在實驗 semi supervised 時一開始為了確保有取到 model 很有信心的 prediction，因此取了情緒分數高於 0.95 和低於 0.05 的 data，且一次取了原始 data 的好幾倍量。結果 train 的過程 train accuracy 直接飆到 1，而 validation accuracy 一直在原地跳動，完全沒有進步。我覺得是因為取那些 model 太有信心的 data 其實無助於他學習，等於要 model 一直複習他已經會的東西，且可能是他自以為是正確的東西。而一次取太大量的 no-label data 會變成 sample 到的幾乎都是 model 自己 label、且 train loss 都已經超低的 data，學不到新東西。因此後來我改取 model 沒有那麼確定的 data (score > 0.75 && score < 0.25)，這樣訓練過程就是讓 model 進一步確認自己是對的，是會學到東西的。但這樣還是會有 model 一開始就錯了的問題，可能要 ensemble 多個 model 再取 score > 0.75 && score < 0.25 的 data 比較好。但因為還沒 ensemble 就有不錯的進步，就沒有再進一步實驗了。

MODEL	PUBLIC	PRIVATE
w/o semi supervised	0.82188	0.81995
w semi supervised	0.82415	0.82231