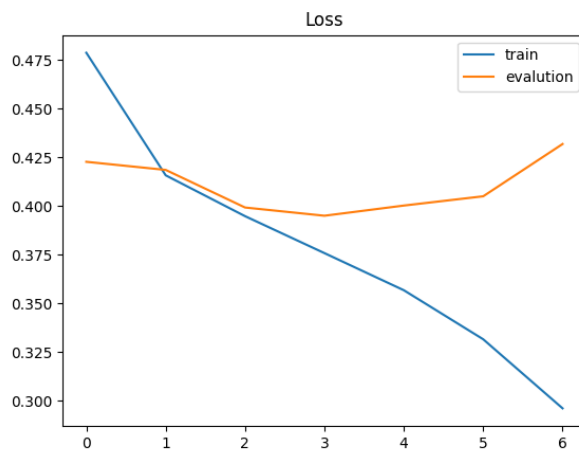
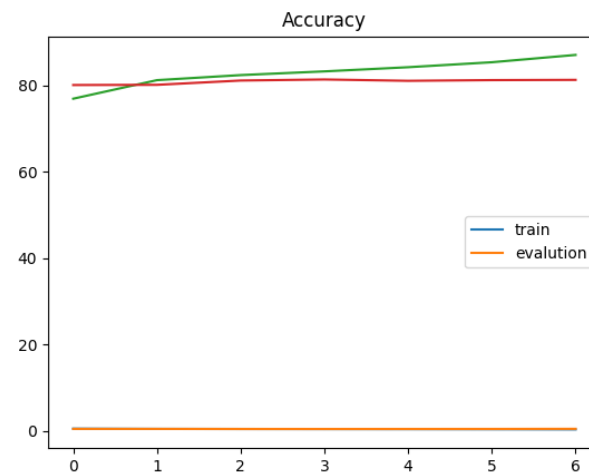


1. (1%) 請說明你實作的 RNN 的模型架構、word embedding 方法、訓練過程(learning curve)和準確率為何？(盡量是過 public strong baseline 的 model)

Word embedding: 使用 gensim 套件提供的 word2vec 將句子轉成 dimension 為 400 的向量

Model architecture: 一層 LSTM 以及 Full connected layer, 最後輸出一維的向量,代表辨識的結果, 其中 LSTM 輸入為 400 dimension 的向量, 輸出為  $1000 \times 4$  的向量, \*4 的原因是使用了 bidirection 的 LSTM 以及使用了 attention 的方法, 所以最後輸出的參數為指定的 hidden dimension 的 4 倍

訓練過程如下:



可以看出第四個 iteration 後就會遇到 overfitting 的問題，最終採用 evaluation accuracy 最佳的模型，在 kaggle 上 public 分數為 0.824

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	RNN
today is a good day, but it is hot	0.54	0.32
today is hot, but it is a good day	0.60	0.76

結果可以看出 BOW + DNN 沒有辦法成功預測，而 RNN 可以，可能的原因是因為 DNN 沒有辦法處理句子這種在不同位置會有不同解釋的特性，例如在此例 good day 在 but 前後就會有不同的解釋，而 RNN 在預測的時候，可以考慮到相鄰文字的關係，因此效果比較好，另外的 BOW 只能認出單個 word，沒辦法根據前後文來判斷 word 的意思，因此遇到同個 word 但不同意思的情況就沒辦法處理了。

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)

Embedding 差別: 原先使用 250 dimension word2vec, 後來實驗了不同的 dimension, 最大使用到 600, 但是結果其實不是越大越好, 太大的 dimension 可能會造成 vector 太分散, 反而造成反效果, 最終使用 400, 並且將 no label data 一起訓練

RNN 架構差別: 在訓練時, 最大的問題是 overfitting 很嚴重, 常常 training data 可以達到超過 90%的正確率,但 evaluation data 只有接近 80%, 這個問題在加大 drop out 參數以及使用 GRU 可以小幅度的改善, 但還是得不到太理想的結果, 最後 best model 是加入 attention, 將 LSTM hidden layer 的資訊拿來作運算後一起訓練, 而使用 bidirectional LSTM 也可以小幅增加正確率.

	BOW+DNN	RNN
Train accuracy	0.849	0.892
Evaluation accuracy	0.815	0.796

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-supervised 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-supervised learning 所帶來的幫助）。

註：此實驗僅使用 2 萬筆 training data

將 model 訓練 5 epoch 後，在 training accuracy 為 84.5% evaluation accuracy 則只有 72%，此時將該 model 去預測 no label data，如果預測出來的分數大於 0.7，則將此 data label 為 1 加入 training data，如果小於 0.3，則將 data label 為 0 加入 training data，最後將新的 training data 去訓練 5epoch 後，在 training accuracy 為 86.5% evaluation accuracy 則上升到 77%，因此 semi-supervised 有發揮其效果，另外如果嘗試設 0.5 當作 threshold，則正確率則為 73%，幾乎沒有增加，因為可能加入了不少 label 錯誤的資料。