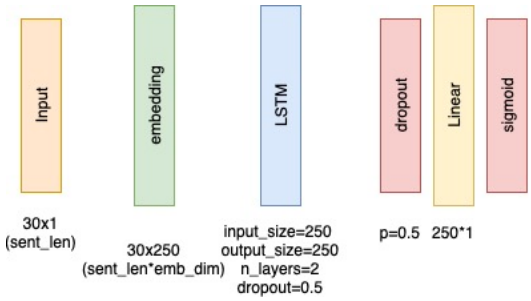
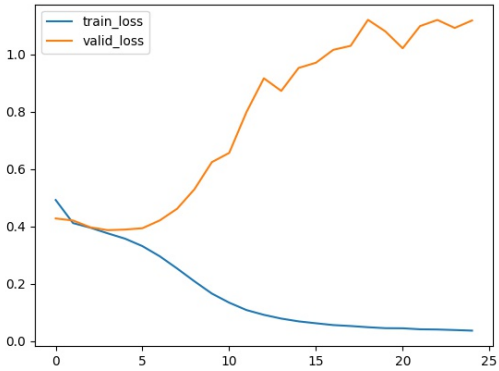
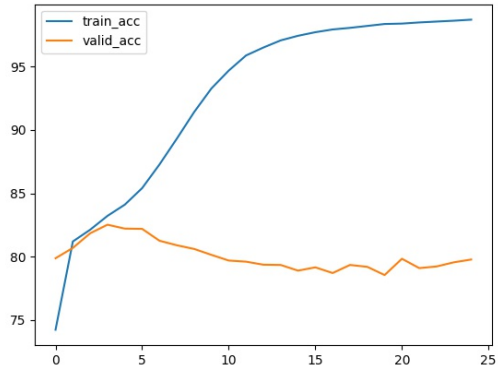
學號：B07902047　系級：資工二　　姓名：羅啟帆

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: word2vec.Word2Vec(x, size=250, window=5, min_count=5, iter=10, sg=1), sg=1 代表 skip-gram
- 訓練過程





- 依照 valid_acc 取最好的 model 得到 valid_acc = 82.524%，public_test = 82.576%

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	0.9861	0.6387
today is hot, but it is a good day	0.9861	0.7041

造成差異的原因是因為 BOW 沒有考慮字的順序，而 RNN 有。上面給的兩個句子都是用一樣的單字，所以對 BOW 來說他們長得一樣，預測出來的分數當然也就一樣。

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）

- 作法
    1. Preprocess: 將每個字根據 pretrained embedding 換成相對應的 word vector，然後把每個句子 pad/truncate 到相同的長度（30）
    2. Embedding: 在 train_x, train_no_label, test_x 上面訓練 word2vec embedding

        - 參數如下：word2vec.Word2Vec(x, size=250, window=5, min_count=5, workers=40, iter=10, sg=1)
    - 解釋
        1. Pretrained 過的 word vector 會比直接用 one-hot encoding 來得好或許是因爲 pretrained 過的 word vector 比較可以捕捉到字跟字之間的關係，而對 Model 來說也比較好去“學到”一句話的意思。
        2. 在多個文本上做 word2vec embedding 的訓練會比只在 train_x 上做訓練來的好是因為 data 越多，word2vec training 的過程越能捕捉到字跟字之間的關係。
    - Valid set 準確率差異（相同 hyperparameter，valid set = 5% train set）
        1. Embedding trained only on train_x: 81.340%
        2. Embedding trained on all data: 82.613%
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。

    - Methods：在 20000 筆資料（19000 training, 1000 validate）用不同的參數訓練多個（我用了 6 個） LSTM model 然後分類 unlabeled dataset。如果在一筆資料上有 5 個以上的 model 的預測都一樣的話，那就將那筆資料加入 labeled dataset。
    - 在 valid dataset 上那 6 個 model 的表現大概是 80% 的準確率
    - 經過以上過程，新的 train set 大小約為原本的 65 倍（1300000）。但重新在這個 dataset 上面 train 的時候卻讓 valid accuracy 比原本低很多（60%），只比亂猜好一些些。
    - 我想原因是因為自己 label 的那些 data 中有許多的錯誤，導致 model 無法好好的從資料中學到東西。