深度學習之應用 HW1

學號：r10525062

姓名：呂雅芳

- Intent：
  1. Dataset：
     - 訓練資料集共有 18000 筆，分割成 15000 筆訓練資料、3000 筆驗證資料
     - 測試資料集共有 4500 筆
     - 預測類別 150 種
  2. Preprocess：
     - 將訓練資料裡的 text 欄位所有的字詞做 dictionary，代表著 dictionary 裡不會出現重複的字詞並且帶有唯一的 id
     - 訓練資料裡 intent 欄位如同上步驟相同
     - 建立完成 dictionary 之後，將訓練資料集每筆資料轉換為相對應 dictionary 的 id ，如此一來，訓練資料集變成 Vector 型態
     - 為了將訓練資料放進模型，每筆資料長度必須相同，也就是每筆 Vector 長度需相同(每句話都一樣長)，經過計算，從訓練資料中得到最大 Vector 長度為 28，故將每個 Vector padding 到長度為 30（保險起見）

```
<tf.Tensor: shape=(30,), dtype=int64, numpy=
array([ 12,  61,  70,   3, 301, 803,   8,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0])>
```

圖 1 一句話轉成長度為 30 的 Vector

     - Embedding：
       將創立好的字詞 dictionary mapping 到 glove 300d 中，可以求得每個字經過 embedding 之後的 Vector，轉換的 Vector 會是一維 Vector，元素都為 [0,1] 之間
  3. Model：
     第一層為 Embedding layer，將剛剛先前訓練好的 Embedding 放入；接著加入兩層雙向的 LSTM，雙層 LSTM 都有加 Dropout 值，避免 Model 會 Overfitting，最後用 Dense layer 作為輸出，units 為 150（因為預測類總共 150 種），激活函數用 softmax，softmax 特性為輸出一維陣列，預測某筆類別的機率，相加為 1

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 300)	1570200
bidirectional (Bidirectional	(None, None, 256)	439296
bidirectional_1 (Bidirection	(None, None, 256)	394240
dense (Dense)	(None, None, 150)	38550

Total params: 2,442,286
Trainable params: 2,442,286
Non-trainable params: 0

圖 2 模型架構

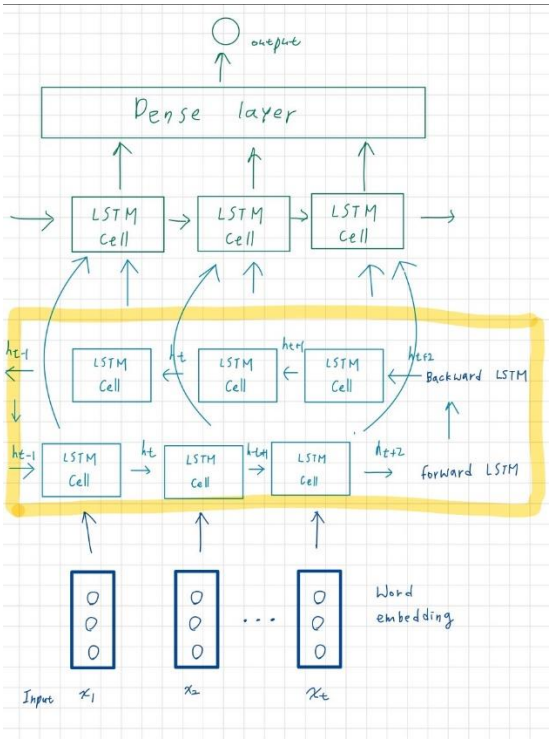

圖 3 模型架構圖

Private Score	Public Score
0.92088	0.91733

圖 4 Kaggle Score

Loss function ：sparse_categorical_crossentropy
Learning rate：0.001
Batch_size：128
Optimization algorithm：Adam

4. Result：

跑 20 次 epoch

```
loss: 0.0177 - accuracy: 0.9961 - val_loss: 0.3870 - val_accuracy: 0.9143
```

圖 5 模型準確度以及 loss value

- Slot：

1. Dataset：
   - 訓練資料集共有 8244 筆，分割成 7244 筆訓練資料、1000 筆驗證資料
   - 測試資料集共有 3731 筆
   - 預測類別 9 種
2. Preprocess：
   - 將訓練資料裡的 tokens 欄位所有的字詞做 dictionary，代表著 dictionary 裡不會出現重複的字詞並且帶有唯一的 id
   - 訓練資料裡 slot 欄位如同上步驟相同
   - 建立完成 dictionary 之後，將訓練資料集每筆資料轉換為相對應 dictionary 的 id ，如此一來，訓練資料集變成 Vector 型態
   - 為了將訓練資料放進模型，每筆資料長度必須相同，也就是每筆 Vector 長度需相同(每句話都一樣長)，經過計算，從訓練資料中得到最大 Vector 長度為 35，故將每個 Vector padding 到長度為 35
   - Embedding：

     將創立好的字詞 dictionary mapping 到 glove 300d 中，可以求得每個字經過 embedding 之後的 Vector，轉換的 Vector 會是一維 Vector，元素都為 [0,1] 之間
3. Model：

   第一層為 Embedding layer，將剛剛先前訓練好的 Embedding 放入；接著加入三層雙向的 LSTM，三層 LSTM 都有加 Dropout 值，之後加一層 TimeDistributed，是為應用於輸入每個字詞時有時間順序，unit 為 9(因為有 9 種詞性)，避免 Model 會 Overfitting，最後用 Dense layer 作為輸出

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 300)	1588500
bidirectional (Bidirectional	(None, None, 256)	439296
bidirectional_1 (Bidirection	(None, None, 256)	394240
bidirectional_2 (Bidirection	(None, None, 256)	394240
time_distributed (TimeDistri	(None, None, 9)	2313
dense_1 (Dense)	(None, None, 35)	350

Total params: 2,818,939
Trainable params: 2,818,939
Non-trainable params: 0

圖 6 模型架構

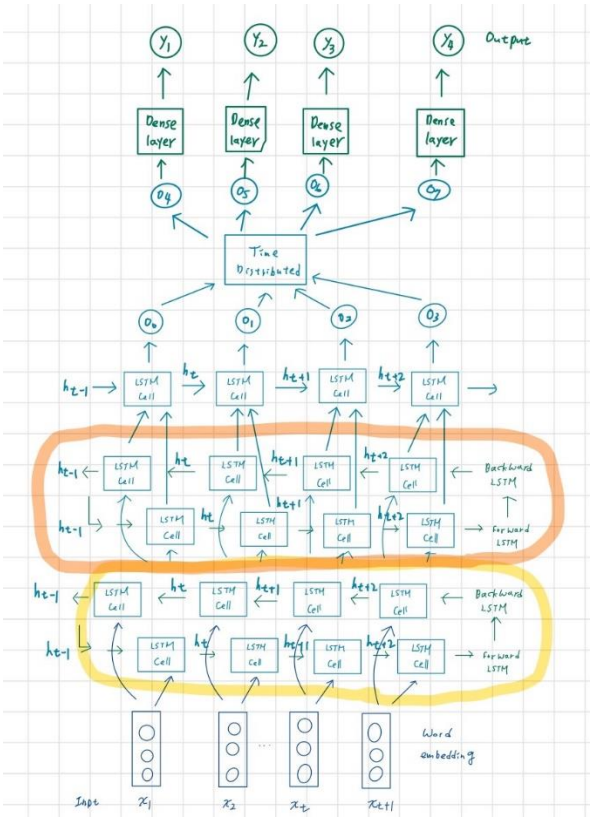

圖 7 模型架構圖

Private Score	Public Score
0.74062	0.71796

圖 8 Kaggle Score

Loss function ：sparse_categorical_crossentropy
Learning rate：0.001
Batch_size：128
Optimization algorithm：Adam

4. Result：
經過幾次實驗發現，epoch 不能超過 20 次，會有嚴重的 overfitting，所以增加一個 early stopping ，模型會當訓練到某個程度時會自動停止，以避免 overfitting；前期發現 dictionary 的字詞意外的很少，導致前期準確度非常低，所以加入驗證集資料的字詞放入 dictionary ，讓 dictionary 認識多一點單字。

```
loss: 0.0130 - accuracy: 0.9954 - val_loss: 0.0334 - val_accuracy: 0.9898
```

圖 9 模型準確度以及 loss value

5. Sequence Tagging Evaluation

	precision	recall	f1-score	support
date	0.75	0.69	0.72	206
first_name	0.88	0.82	0.85	102
last_name	0.67	0.78	0.72	78
people	0.69	0.69	0.69	238
time	0.70	0.70	0.7	218
micro avg	0.72	0.72	0.72	842
macro avg	0.74	0.74	0.74	842
weighted avg	0.73	0.72	0.72	842

- 總結：前處理十分重要，會影響到模型預測的參數。前期使用兩層 LSTM 去做預測 Accuracy 大概在 0.91 上下，之後將 LSTM 改成 CNN-BiLSTM 之後，Accuracy 可以上升至 0.98 ，代表字詞在句子當中前後關係非常重要，以至於加上 Backward、Forward 之後可以大幅提高準確度。