

# Applied Deep Learning Homework 1

台灣科技大學 企管碩一

M11008019 吳英媛

## Q1. Data processing

### a. How do you tokenize the data.

#### 1. Intent Classification

- (1) 使用 LabelEncoder 將 train 資料集的「intent」欄位進行 Label encoding，並新增一欄名為「Label」的預測目標欄位
- (2) 使用 keras 的 tokenizer 套件，以文字間格的空白分割「text」欄位中的文字，並將其文本轉換為序列編碼，並加上 Padding。

#### 2. Slot Tagging

- (1) 先查看需要進行預測的 tag (B-date、B-first\_name、B-last\_name、B-people、B-time、I-date、I-people、I-time 和 0，共 9 個)
- (2) 創建 dictionary 資料型態的 label 來儲存這些 tag 並加上 Padding。
- (3) 接著使用 keras 的 tokenizer，篩選出詞頻較高的 1000 個詞彙當作字典 (num\_word=1000)，且標記未在字典中的字彙(oov\_token='OOV')
- (4) 以文字間格的空白分割「text」欄位中的文字，並將其文本轉換為序列編碼，並加上 Padding。

### b. The pre-trained embedding you used.

使用 GloVe (840B tokens, 300d vectors)進行預訓練。

## Q2: Describe your intent classification model.

### a. your model

模型包含 Embedding、BiLSTM、池化層、2 個全連接層和輸出層。

Embedding 維度需與預訓練的 GloVe 維度相同；GlobalAveragePooling 則對每個 Tensor 取平均值，加速模型泛化與收斂；透過 Bi-LSTM 找出句子的特徵向量，其後連接 2 個全連接層，激活函數為 swish，Kernel initializer 為 orthogonal；最後為輸出層，激活函數為 softmax，輸出 150 個介於 0-1 的機率值。

模型層	說明	架構
Embedding	輸入層	300 維
BiLSTM	RNN 模型	512 維，Dropout=0.2
GlobalAveragePooling1D	池化層	512 維
Dense	連接層	256 維，Dropout=0.2
Dense	連接層	128 維，Dropout=0.2
Dense softmax	輸出層	150 維，激活函數為 softmax

b. performance of your model.(public score on kaggle)

	Accuracy
Validation	0.9997
Public	0.94044
Private	0.93600

c. the loss function you used.

使用 Sparse Categorical Cross entropy (稀疏分類交叉熵)，因此最後輸出層的激活函數為 softmax，將每個節點輸出在 0-1 之間的機率值。

d. The optimization algorithm, learning rate and batch size.

使用 Adam，learning rate 為 0.001，batch size 為 32，epoch 為 20。先使用訓練資料集找出合適的 batch size 和 epoch 後，再利用 train 和 eval 資料集進行建模。

Q3: Describe your slot tagging model.

a. your model

模型包含 Embedding、BiLSTM、TimeDistributed 連接層和輸出層。

Embedding 維度需與預訓練的 GloVe 維度相同，透過 Bi-LSTM 找出詞之間的相互關係，而後面接 TimeDistributed 連接層，其激活函數為 ReLU，kernel\_initializer 為 orthogonal。最後為 TimeDistributed 輸出層，激活函數為 softmax，輸出 10 個介於 0-1 的機率值。

模型層	說明	架構
Embedding	輸入層	300 維
BiLSTM	RNN 模型	256 維，Dropout=0.2
TimeDistributed Dense	連接層	128 維，Dropout=0.2
TimeDistributed Dense	輸出層	10 維，激活函數為 softmax

b. performance of your model.(public score on kaggle)

	Accuracy
Validation	0.8197
Public	0.80160
Private	0.80010

c. the loss function you used.

使用 Sparse Categorical Cross entropy (稀疏分類交叉熵)，因此最後輸出層的激活函數為 softmax，將每個節點輸出在 0-1 之間的機率值。

d. The optimization algorithm, learning rate and batch size.

使用 Adam，learning rate 為 0.001，batch size 為 64，epoch 為 20。與 Intent 過程相同，原本也有將訓練集和測試集合併一起進行模型訓練，但由於上傳 Kaggle 的 Public 準確率不及僅使用訓練集建模之成效，因此 Slot 之最佳模型並無包含 full dataset 的訓練。

Q4. Sequence Tagging Evaluation

	precision	recall	f1-score	support
date	0.79	0.77	0.78	206
first_name	0.94	0.89	0.91	102
last_name	0.87	0.95	0.91	78
people	0.74	0.71	0.72	238
time	0.76	0.75	0.75	218
micro avg	0.79	0.78	0.79	842
macro avg	0.82	0.81	0.82	842
weighted avg	0.79	0.78	0.79	842

**True Positive : TP , True Negative : TN**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$f1 - score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

**Support = The correct tag number**

$$\text{Micro avg} = \frac{TP \text{ of tag number}}{TP \text{ of tag number} + FP \text{ of tag number}}$$

$$\text{Macro avg} = \frac{\text{Precision} + \text{Recall}}{2}$$

$$\text{Weighted avg} = \frac{\sum(\text{tag number} \times \text{Weight Vector})}{\sum \text{Weight}}$$

由上述統計學公式可得知，Precision 表示模型預測結果為正確樣本之準確度，Recall 則表示模型預測正確樣本之召回率，而 F1-score 為 2 者之調和平均數，因此要評估模型表現可以以 f1-score 進行參考。

$$\text{Token Accuracy} = \frac{\text{correct sequence predicted}}{\text{number of all sequence predicted}}$$

$$\text{Joint Accucarcy} = \frac{\text{correct token predicted}}{\text{number of all token predicted}}$$

	Accuracy
Token Accuracy	96.5%
Joint Accuracy	78.7%

由上表可得知雖然預測正確之 tag 數量高達 96.5%，但模型整句完全預測正確的準確率卻僅有 78.7%。

#### Q5. Compare with different configurations

與 Q2 和 Q3 的最佳模型架構相同，查看不同 Vocabulary size 和 Batch size 對模型之影響。

##### 1. Intent Classification

Vocab size	4000	5000	6000	7000
Val acc	0.9891	0.9963	0.9997	0.9984
Public acc	0.92622	0.93422	0.94044	0.94222
Private acc	0.93422	0.93022	0.93600	0.93200

單位: 正確率

Intent 最佳模型之 Vocab size 為 6000，與 tokenizer 進行切詞後的字詞數量相近(6,438 個字)。

Batch size	16	32	64	128
Val acc	0.9997	0.9993	0.9988	0.9990
Public acc	0.94044	0.93022	0.93822	0.93466
Private acc	0.93600	0.93244	0.92933	0.92177

單位: 正確率

Intent 最佳模型之 Epoch 為 20，Batch size 為 16，可知此資料集較適合小的 Batch size，小的 Batch size 讀取資料較具隨機性，訓練後之模型的泛化能力也會較好。

## 2. Slot Tagging

Vocab size	500	1000	2000	3000
Val acc	0.8133	0.8197	0.8194	0.8191
Public acc	0.77426	0.80160	0.79088	0.78123
Private acc	0.77759	0.80010	0.79957	0.78670

單位:正確率

Slot 最佳模型之 Vocab size 為 1000，若增加 Vocab size 有 Overfitting 之傾向，若減少則會 Underfitting。

Batch size	16	32	64	128
Val acc	0.8187	0.8184	0.8197	0.8195
Public acc	0.79323	0.78145	0.80160	0.79588
Private acc	0.78567	0.78657	0.80010	0.79757

單位:正確率

Slot 最佳模型之 Epoch 為 20，Batch size 為 64，由上表之準確率數值無法推估 Batch Size 對此模型之影響。