

ADL Hw1 report

R10723059 財金碩二 胡祖望

October 13, 2022

1 Q1: Data processing

兩題都使用 sample code 將每個 batch 的樣本取出

在 slot 問題中將 data 中的 token 一組一組取出分開，再將每組 token 放入 sample code 的 encode batch。

在 intent 問題中將文字用空格分割，變成一組一組分好的 token，再將每組 token 放入 sample code 的 encode batch。

然後使用 glove 檔案，將每個 token 對應到它的向量。

2 Q2: Describe your intent classification model

2.1 my model

分成 Encoder 與 Decoder。

Encoder 輸入 embedded 好的 token 向量到模型進行訓練。

Decoder 將訓練好的向量轉成每個 intent 的分數。

Encoder 與 Decoder 皆使用 Dropout。

Encoder: 使用雙向 LSTM 模型，超參數設定如下：

input size = 300, hidden size = 512, number of layers = 2, dropout = 0.1。

training: 將分割好的 intent 輸入 embedding，每個 token 轉成 300 維的向量，再將這些向量放入雙向 LSTM 模型中，得到兩個 hidden size 大小的 output，問題輸入整句後只需要一個 output，所以只要取正向與反向的最後一個 output。LSTM 模型的 output 只需看 h_n 即可。使用 torch.cat 函式將正反向 output 合成 512*2 維的 output 在 Decode 上使用，以式子表示如下：

intent = X

$X = x_1, x_2, \dots, x_n$, n 是樣本數

$h_t, (h_n, c_n) = \text{LSTM}(X, (h_0, c_0))$

$h_{\text{output}}(1024) = \text{torch.cat}(h_n[-1], h[-2])$

Decoder 最後會有一個 linear layer，將 1024 維的 input 轉成 150 維的 output，而這 150 代表 intent 的種類，最後取分數最大的 intent 當作預測結果。

$h_{\text{output}}(150) = \text{Linear}(h_{\text{input}}(1024))$

Predicted intent = Max($h_{\text{output}}(150)$)

2.2 performance of my model

public score: 0.90933

2.3 loss function

使用 Cross entropy，比對預測的分數與實際 intent 做比對做 loss

loss = CrossEntropyLoss(h_{output} , real intent)

2.4 optimization algorithm, learning rate, batch size, epoch

optimization 用的是 Adam, learning rate = 0.001, batch size = 128, epoch = 100。

3 Q3: Describe your slot tagging model

3.1 my model

分成 Encoder 與 Decoder。

Encoder 輸入 embed 好的 token 向量到模型進行訓練。

Decoder 將訓練好的向量轉成每個 intent 的分數。

Encoder 與 Decoder 皆使用 Dropout。

Encoder: 使用雙向 LSTM 模型, 超參數設定如下:

input size = 300, hidden size = 512, number of layers = 2, dropout = 0.1。

training: 將分割好的 intent 輸入 embedding, 每個 token 轉成 300 維的向量, 再將這些向量放入雙向 LSTM 模型中, 得到兩個 hidden size 大小的 output, 問題輸入整句後只需要一個 output, 所以只要取正向與反向的最後一個 output。LSTM 模型的 output 只需看 h_n 即可。使用 torch.cat 將正反向 output 合在一起 512*2 維的 output 在 Decode 上使用, 以式子表示如下:

intent = X

$X = x_1, x_2, \dots, x_n$, n 是樣本數

$h_t, (h_n, c_n) = \text{LSTM}(X, (h_0, c_0))$

$h_{\text{output}}(1024) = \text{torch.cat}(h_n[-1], h[-2])$

Decoder 最後會有一個 linear layer, 將 1024 維的 input 轉成 150 維的 output, 而這 150 代表 intent 的種類, 最後取分數最大的 intent 當作預測結果。

$h_{\text{output}}(150) = \text{Linear}(h_{\text{input}}(1024))$

Predicted intent = Max($h_{\text{output}}(150)$)

3.2 performance of my model

public score: 0.73619

3.3 loss function

使用 Cross entropy, 比對預測的分數與實際 intent 做比對做 loss

loss = CrossEntropyLoss(h_{output} , real intent)

3.4 optimization algorithm, learning rate, batch size

optimization 用的是 Adam, learning rate = 0.001, 使用 torch.optim 的 learning rate.ReduceLROnPlateau 使 learning rate 下降, batch size = 128, epoch = 100。

4 Q4: Sequence Tagging Evaluation

```
Joint Acc: 0.821000 (821/1000) Token Acc: 0.970473 (7658.0/7891.0)
11 Model Save:
Best Model Save
sequeval classification report:
```

	precision	recall	f1-score	support
date	0.79	0.77	0.78	206
first_name	0.98	0.92	0.95	102
last_name	0.91	0.79	0.85	78
people	0.75	0.74	0.74	238
time	0.84	0.87	0.85	218
micro avg	0.82	0.81	0.81	842
macro avg	0.85	0.82	0.83	842
weighted avg	0.82	0.81	0.82	842

joint accuracy 代表完全預測正確的序列比例，token accuracy 代表 token 預測正確的比例。Sequeval 會依照每種 tag 的預測與真實表現來做計算，以此當依據計算特定項目，首先解釋變數：TP 表完全預測正確的特定 tag 數，FP 表預測錯誤的特定 tag 數，FN 代表真實與預測不符的特定 tag 數，故 $TP+FP$ = 所有的預測回該特定 tag 的數量， $TP+FN$ = 所有的真實特定 tag 數量。Sequeval 數據：

$\text{precision} = \frac{TP}{TP+FP}$ 代表特定 tag 的預測正確率
 $\text{recall} = \frac{TP}{TP+FN}$ 代表真實的特定 tag 的預測準確率
 $\text{f1-score} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$ precision 與 recall 的調和平均
 $\text{support} = \text{真實的特定 tag 的數量} = (TP+FN)$
 $\text{micro avg} = \text{計算所有 tags 組成的 } \frac{TP}{TP+FP} \text{ or } \frac{TP}{TP+FN}$
 $\text{macro avg} = \text{計算每種數據的在各 tags 的平均值 (未加權)}$
 $\text{weighted avg} = \text{計算每種數據的在各 tags 的加權平均}$

5 Q5: Compare with different configurations

5.1 improve Q3

起初只使用一般 LSTM 模型訓練，後用 learning rate 的 scheduler 套件，讓模型能有效的降低 Validation 組的 loss。