

R10723050 財金碩二陳韻帆

Q1 Data processing

使用 sample code

在兩題中皆先將每個 batch 的樣本取出

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 做轉換。

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 做轉換。

使用助教提供的 glove 檔案，每個 token 有預先訓練好，對應的 300 維向量。

Q2 Describe your intent classification model

a. my model

分為 Encoder 與 Decoder，使用 Encoder 輸入 embed 好的 token 向量資料到模型進行訓練，使用 Decoder 將訓練好的 vector 轉為每個 intent 的分數，在 Encoder 與 Decoder 中皆使用 Dropout。

Encoder：使用雙向 LSTM 模型，超參數設定如下：

input size = 300
hidden size = 512
num layers = 2
dropout = 0.1

訓練過程如下：將分割好的 intent 輸入 embedding 層中，將每個 token 轉為 300 維的向量，後將這些向量放入雙向 LSTM 模型中，得到兩個大小為 hidden size 的 output，因此問題輸入整句後只需要一個 output，故只要取正向與反向的最後一個 output 即可，LSTM 模型的 output 只需看 h_n 即可。使用 torch.cat 函式將正反向 output 拼接產出 1024 維的 output 準備於 Decode 使用，以式子表示如下：

$$\begin{aligned} \text{intent} &= X \\ X &= \{x_1, x_2, \dots, x_n\}, \quad n \text{ 為樣本數} \\ h_t, (h_n, c_n) &= \text{LSTM}(X, (h_0, c_0)) \\ h_{\text{output}}(1024) &= \text{torch.cat}(h_n[-1], h_n[-2]) \quad () \text{ 內數字代表維度} \end{aligned}$$

Decoder：建立一個線性層，將 1024 維的 output 轉成 150 維的分數 (有 150 種 intent)，最後取分數最大的 intent 當作預測結果以式子表示如下：

$$\begin{aligned} h_{\text{output}}(150) &= \text{Linear}(h_{\text{output}}(1024)) \quad () \text{ 內數字代表維度} \\ \text{Predicted intent} &= \text{Max}(h_{\text{output}}(150)) \end{aligned}$$

b. performance of my model

public score : 0.92222

c. loss function

將使用 Cross entropy loss，將預測的 150 個分數與實際 intent 對比做 loss，以式子表示如下：

$\text{loss} = \text{CrossEntropyLoss}(h_{\text{output}}(150), \text{real intent})$

- d. optimization algorithm, learning rate, batch size
使用 Adam optimization algorithm · learning rate 為 0.001 · batch size 為 128

Q3 Describe your slot tagging model

a. my model

分為 Encoder 與 Decoder，使用 Encoder 輸入 embed 好的 token 向量資料到模型進行訓練，使用 Decoder 將訓練好的 vector 轉為每個 tag 的分數，在 Encoder 與 Decoder 中皆使用 Dropout。

Encoder：使用雙向 GRU 模型，超參數設定如下：

input size = 300
hidden size = 512
num layers = 2
dropout = 0.2

訓練過程如下：將 token 輸入 embedding 層中，將每個 token 轉為 300 維的向量，後將這些向量放入雙向 GRU 模型中，得到兩個大小為 hidden size 的 output，因此問題需要一對一的 tag 故需要使用所有 output(h_t)，其中每個元素 (每個元素皆有 1024 維) 都代表一個詞的 tag 準備於 Decode 使用，以式子表示如下：

$$\begin{aligned}\text{intent} &= X \\ X &= \{x_1, x_2, \dots, x_n\}, \quad n \text{ 為樣本數} \\ h_t, h_n &= GRU(X, (h_0)) \\ h_t &= \{h_1, h_2, \dots, h_n\}, \quad n \text{ 為樣本數}\end{aligned}$$

Decoder：建立一個線性層，將每個 1024 維的 output 轉成 9 維的分數 (有 9 種 tags)，最後取分數最大的 tag 當作預測結果以式子表示如下：

$$\begin{aligned}h_{\text{output}}(9) &= \text{Linear}(h_{\text{output}}(1024)) \quad () \text{ 內數字代表維度} \\ \text{Predicted tag} &= \text{Max}(h_{\text{output}}(9))\end{aligned}$$

b. performance of my model

public score : 0.81394

c. loss function

將使用 Cross entropy loss，將預測的 9 個分數與實際 intent 對比做 loss，以式子表示如下：

$$\text{Loss} = \text{CrossEntropyLoss}(h_{\text{output}}(9), \text{real tag})$$

d. optimization algorithm, learning rate, batch size

使用 Adam optimization algorithm，learning rate 為 0.001，並使用 learning rate scheduler 中的 ReduceLROnPlateau 做 learning rate 衰減，標的為 Joint Accuracy，希望讓 Joint Accuracy 上升，batch size 為 128

Q4 Sequence Tagging Evaluation

Joint Acc: 0.816000 (816/1000) Token Acc: 0.969459 (7650.0/7891.0)				
sequeval classification report:				
	precision	recall	f1-score	support
date	0.79	0.77	0.78	206
first_name	0.96	0.88	0.92	102
last_name	0.85	0.82	0.84	78
people	0.76	0.77	0.77	238
time	0.86	0.85	0.86	218
micro avg	0.83	0.81	0.82	842
macro avg	0.85	0.82	0.83	842
weighted avg	0.83	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 (例如：計算 first name tag 的數據)，以此為依據計算特定項目，首先針對特定變數說明：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 $TP+FP=$ 所有的預測為該特定 tag 的數量， $TP+FN=$ 所有的真實特定 tag 數量，再來說明 Sequeval 數據內容：

precision= $\frac{TP}{TP+FP}$ 代表特定 tag 的預測正確率

recall= $\frac{TP}{TP+FN}$ 代表真實的特定 tag 的預測準確率

f1-score= $\frac{2}{\frac{1}{precision} + \frac{1}{recall}}$ precision 與 recall 的調和平均

support= 真實的特定 tag 的數量 ($=TP+FN$)

micro avg= 計算所有 tags 組成的 $\frac{TP}{TP+FP}$ or $\frac{TP}{TP+FN}$

macro avg= 計算每種數據的在各 tags 的平均值 (未加權)

weighted avg= 計算每種數據的在各 tags 的平均值 (有加權)

Q5 Compare with different configurations

1. improve Q3

起初只使用一般 LSTM 模型訓練，後用 GRU 模型對比發現 LSTM 模型有一點過擬合的狀況，當 Training 組的 Joint Accuracy 達到約 95%，Validation 組的 Joint Accuracy 仍卡在約 73-74% 無法改善，後加入控制 learning rate 的 scheduler 套件，讓模型能有效的降低 Validation 組的 loss，並且改用 GRU 模型去減少參數減緩過擬合問題，後最高訓練分數可達約 81%。

2. 加入 CRF 模型連接原 GRU 模型

嘗試使用 pytorch-crf 套件用 CRF 評分方式提高分數，實際使用後發現對模型訓練結果有改善，但幅度有限，Joint Accuracy 約從 81.0% 升高到 81.7%，而實際帶入 baseline 後也發現分數只有上升 0.001 分左右，並不算是太大的進步。