ADL Hw1 Report **R11922150 游鈞皓**

Q1: Data processing (2%)

使用 sample code

a. How do you tokenize the data.

Intent Classification : 以空白分隔 input text

Slot Tagging : input tokens 已是 tokens list

接著皆是以 dataset 中最常見的{vocab_size} 個 tokens 建出 vocab，並將 tokens 由 str 轉為 int

b. The pre-trained embedding you used.

皆使用 glove.840B.300d

Q2: Describe your intent classification model. (2%)

a. your model

LSTM : hidden_size=512, num_layers=2 , bidirectional=True, dropout=0.1

lstm_output, (h, c) = lstm(x, (h0, c0))

其中 h0,c0 使用 default 值 , x = embedding(tokenize(input text))

Linear

output = linear(concatenate(h[-1], h[-2]))，其中 h 為 LSTM 之回傳值 h，因使用 bidirectional, 2-layers configuration，故 h[-1], h[-2] 代表第 2 層 layer 的 forward and reverse hidden states

linear 的 output 即為最終 model output

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

Eval Acc	Kaggle Public Score
0.924333333	0.91600

c. the loss function you used.

使用 CrossEntropyLoss, loss = CrossEntropyLoss(model output, ground truth intent)

d. The optimization algorithm (e.g. Adam), learning rate and batch size.

使用 Adam, learning rate = 1e-3, batch size =128

Q3: Describe your slot tagging model. (2%)

a. your model

LSTM : hidden_size=256, num_layers=2 , bidirectional=True, dropout=0.1

lstm_output, (h, c) = lstm(x, (h0, c0))

其中 h0,c0 使用 default 值 , x = embedding(tokenize(input tokens))

Linear

output = linear(lstm_output)

linear 的 output 即為的最終 model output

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

Eval Acc	Kaggle Public Score
0.82	0.77694

c. the loss function you used.

使用 CrossEntropyLoss

loss = CrossEntropyLoss(model output[i], ground truth tag[i])

output sequence 與 ground truth sequence 中的 element 一一對應算 loss

d. The optimization algorithm (e.g. Adam), learning rate and batch size.

使用 Adam, learning rate = 5e-4, batch size =32

Q4: Sequence Tagging Evaluation (2%)

seqeval :

	precision	recall	f1-score	support
date	0.78	0.77	0.78	206
first_name	0.94	0.95	0.95	102
last_name	0.86	0.82	0.84	78
people	0.76	0.74	0.75	238
time	0.85	0.84	0.85	218
micro avg	0.82	0.81	0.81	842
macro avg	0.84	0.82	0.83	842
weighted avg	0.82	0.81	0.81	842

Recall (召回率) = TP/(TP+FN)

Precision (準確率) = TP/(TP+FP)

F1-score = 2 * Precision * Recall / (Precision + Recall)

TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative)

token accuracy : $\frac{\boldsymbol{number\ of\ correctly\ \ predicted\ tokens}}{\boldsymbol{number\ of\ all\ predicted\ tokens}}$

joint accuracy : $\frac{\boldsymbol{number\ of\ correctly\ \ predicted\ texts}}{\boldsymbol{number\ of\ all\ predicted\ texts}}$

Q5: Compare with different configurations (1% + Bonus 1%)

Configurations Format : (hidden_size, batch_size, learing rate)

All (num_layers, dropout, bidirectional) is (2, 0.1, True)

在嘗試 GRU 及 RNN 時，參考 LSTM 之 performance 對兩個 task 分別設下 0.93, 0.8 的 threshold，accuracy 超過此 threshold 才會儲存 model，而在我的實驗中 RNN 的所有 configurations 在兩項 task 中均未能超過 threshold，可能因為 Vanishing/Exploding Gradient 使 RNN 的 performance 較差，而實驗也觀察到參數量增加及 learing rate 下降都會造成收斂速度變慢。

Intent Classification

Model		LSTM	GRU	RNN
Best Performance	Eval Acc	0.9243	0.9313	<0.93
	Public score	0.91600	0.92444	None
Best Configuration		(512,128,1e-3)	(512, 128, 1e-3)	None

Slot Tagging

Model		LSTM	GRU	RNN
Best Performance	Eval Acc	0.82	0.804	<0.8
	Public score	0.77694	0.75281	None
Best Configuration		(256, 32, 5e-4)	(256, 32, 5e-4)	None