ADL HW1 Report

Q1:

基於sample code並做一些改變

Intent :

不使用collate\_fn而是透過max\_length = 32限制輸入的長度，多的截斷、少的補0，每個字再map為GloVe向量。

Slot :

在tag2idx中新增PAD類(PAD: 9)去處理padding的類別問題，與Intent相同，以max\_length限制輸入長度，在kaggle截止前由於max\_length設置過大(32)，而trainning data的平均約為8 token，因此表現不佳未過baseline，將限制改為24字元後即可有0.75~.76的表現。

Q2:

nn.LSTM(input\_dim = 300,hidden\_dim = 1024,layer\_num = 2,dropout = 0.2)

nn.Linear(hidden\_dim,class\_num)

b)

score:0.88933

c)

nn.CrossEntropyLoss

d)

Adam() default參數，batch\_size = 256 Epoch = 50

Q3:

nn.LSTM(input\_dim = 300,hidden\_dim = 1024,layer\_num = 2,dropout = 0.2,bidirection = True)

nn.Linear(2 \* hidden\_dim,class\_num)

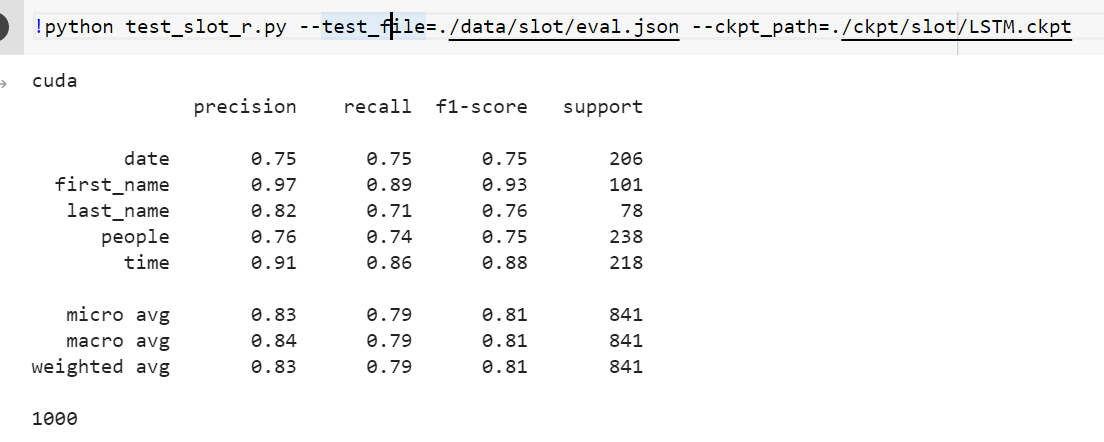
output,(hn,cn),將output reshape給Linear得到sequence的機率分布

score(max\_length=32) : 0.671

nn.CrossEtropyLoss

Adam() default參數，batch\_size = 32 Epoch = 30

Q4:

 a)

b)

Seqeval :

Precision= tp / (tp + fp) 即預測為true的label中有多少正確

Recall = tp / (tp + fn) 即實際為true的label有多少被預測中

F1-score = 2 / ( (1/ Precision) + (1/ Recall) )，是上述兩者的調和平均數

Token accuracy :

代表所有token有多少預測正確

Joint accuracy :

代表有多少sequence是完全正確

我認為Seqeval的公式較為全面，Joint accuracy在序列任務中最為嚴格，Token accuracy在部分任務較無參考價值，假設以病毒檢測而言：

*今天有10000名人員檢測，其中30名為感染者，*

*機器將10名感染者與未感染者20名判定為感染。*

*每一批有100人檢測*

此時若以Token accuracy判斷，正確率為99.6%，因為有太多O的label，因此正確率看似很高但在關鍵label實則不然。

以Joint accuracy判斷則會有非常不穩定的分數，可能30名集中於一批錯誤，正確率為99%，也可能分散於30批出錯，正確率為70%，因此也較無參考價值。

以F1-score判斷則會有Precision : 10 / 30,Recall : 10/30組成，為33%準確率，較為符合我們對於此任務的標準。

Q5:

由於我的slot任務中preprocess階段會將sequence長度統一為max\_len，因此max\_len設定過大會導致無關label過多，經實驗將max\_len從32調至24可以提升score(max\_length=24) : 0.758 (截止後late submission)。