ADL #HW1
陳姵如，資料科學學位學程
R10946029

Q1: Data processing (2%)

1. Describe how do you use the data for intent_cls.sh, slot_tag.sh:
   a. How do you tokenize the data.
   b. The pre-trained embedding you used.
2. If you use the sample code, you will need to explain what it does in your own ways to answer Q1.

我將 text 用空格 split 之後丟入助教的 utils.py 中 class Vocab 中的 encode_batch function，class Vacab 在 init 的時候會讀取 cache/intent/vocab.pkl，裡面會存有字轉換成 id 的對應關係，所以在 encode_batch 裡面的 self.encode 就是在做 token 跟 id 轉換。另外，斷詞的部分也可以使用 nltk.tokenize 來做。
我使用助教提供的 GloVe pre-trained embedding，embedding 權重在 train.py 讀取後會傳到 model 裡，然後在 model.py 使用 torch.nn.Embedding.from_pretrained 來做 token 到 embedding vector 的轉換，其中一個 token 會轉成 300 維。

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

1. Describe
   a. your model
   b. performance of your model.
      (public score on kaggle)
   c. the loss function you used.
   d. The optimization algorithm (e.g. Adam), learning rate and batch size.

a. 我使用 2 層各有 512 hidden neurons 雙向的 GRU 接上 dropout=0.4 最後接上一層 Linear(512*2, 1)。
b. Public score: 0.92222
c. torch.nn.CrossEntropyLoss()
d. Optimizer: Adam
   learning rate: 2e-3
   scheduler: torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer, 'min', factor=0.5, patience=3, min_lr=5e-5, verbose=True)
   batch size: 128

ADL #HW1
陳姵如，資料科學學位學程
R10946029

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

1. Describe
    a. your model
    b. performance of your model.
       (public score on kaggle)
    c. the loss function you used.
    d. The optimization algorithm (e.g. Adam), learning rate and batch size.

a. 我使用 2 層各有 128 hidden neurons 雙向的 GRU 接上 dropout=0.4 最後接上一層 Linear(128*2, 9)。
b. Public score: 0.72064
c. torch.nn.CrossEntropyLoss()
d. Optimizer: Adam
   learning rate: 2e-3
   scheduler: torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer, 'min', factor=0.5, patience=3, min_lr=5e-5, verbose=True)
   batch size: 32

Q4: Sequence Tagging Evaluation (2%)

- Please use seqeval to evaluate your model in Q3 on validation set and report classification_report(scheme=IOB2, mode='strict').
- Explain the differences between the evaluation method in seqeval, token accuracy, and joint accuracy.

	precision	recall	f1-score	support
date	0.72	0.72	0.72	207
first_name	0.92	0.91	0.92	103
last_name	0.63	0.83	0.72	59
people	0.74	0.73	0.74	242
time	0.86	0.84	0.85	222
micro avg	0.78	0.79	0.78	833
macro avg	0.77	0.81	0.79	833
weighted avg	0.78	0.79	0.78	833

Seqeval classification_report(scheme=IOB2, mode='strict') 將 9 類 tag 依不同屬性做分類將其子類做 precision / recall / f1 等指標分析。
Token accuracy 將每個 slot 視為獨立的，單純看對幾個除以所有 slot 數。
Joint accuracy 將一筆 data 所有 slot 視為一個 case，全對才得分，所以會是 slot 全對 data 數 / 所有 data 筆數。

ADL #HW1
陳姵如，資料科學學位學程
R10946029

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

- Please try to improve your baseline method (in Q2 or Q3) with different configuration (includes but not limited to different number of layers, hidden dimension, GRU/LSTM/RNN) and EXPLAIN how does this affects your performance / speed of convergence / ...
- Some possible BONUS tricks that you can try: multi-tasking, few-shot learning, zero-shot learning, CRF, CNN-BiLSTM
- This question will be grade by the completeness of your experiments and your findings.

以下模型設定除了變動參數以外的都與 Q2/Q3 一樣，取用 Adam with ReduceLROnPlateau、Dropout=0.4、batch 128/32、Epoch 10 的 eval Accuracy。

Intent Classification	GRU	LSTM
Hidden layer = 128, bidirection=False	88.63%	85.43%
Hidden layer = 128, bidirection=True	91.43%	91.67%
Hidden layer = 256, bidirection=False	90.47%	88.17%
Hidden layer = 256, bidirection=True	91.57%	91.03%
Hidden layer = 512, bidirection=False	91.50%	89.07%
Hidden layer = 512, bidirection=True	**92.67%**	90.20%

Slot Tagging	GRU	LSTM
Hidden layer = 128, bidirection=False	94.23%	94.60%
Hidden layer = 128, bidirection=True	96.36%	96.43%
Hidden layer = 256, bidirection=False	94.68%	94.54%
Hidden layer = 256, bidirection=True	**96.67%**	96.19%
Hidden layer = 512, bidirection=False	94.35%	94.79%
Hidden layer = 512, bidirection=True	96.49%	96.57%

從以上綜合來說 LSTM 收斂速度較 GRU 慢 （LSTM 多了兩個 gate）、hidden layer 愈多、bidirectional 訓練愈慢，因為多了很多參數要更新。
而在 performance 的部分， bidirection 普遍較好，可以說明句子前後會互相影響。GRU 第一題在 hidden layer 大的時候會比較好，而 LSTM 沒有太大差別，推測是 LSTM 參數量較大，因此對於 performance 影響不大。