學號：R06943148 系級：電子碩一 姓名：謝孟洋

1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？ (Collaborators:)
   答：

Model 架構

```
60      # RNN
61      return_sequence = False
62      dropout_rate = args.dropout_rate
63      if args.cell == 'GRU':
64          RNN_cell = GRU(args.hidden_size,
65                         return_sequences=return_sequence,
66                         dropout=dropout_rate)
67      elif args.cell == 'LSTM':
68          RNN_cell = LSTM(args.hidden_size,
69                          return_sequences=return_sequence,
70                          dropout=dropout_rate)
71      RNN_output = RNN_cell(embedding_inputs)
72
73      # DNN layer
74      outputs = Dense(args.hidden_size//2,
75                      activation='relu',
76                      kernel_regularizer=regularizers.l2(0.1))(RNN_output)
77      outputs = Dropout(dropout_rate)(outputs)
78      outputs = Dense(1, activation='sigmoid')(outputs)
79      model =  Model(inputs=inputs,outputs=outputs)
80
81      # optimizer
82      adam = Adam()
83      print ('compile model...')
84
85      # compile model
86      model.compile( loss=args.loss_function, optimizer=adam, metrics=[ 'accuracy',])
```

訓練過程

(semi-supervised、epoch: 2、optimizer: adam、loss function: binary crossentropy)

```
192         # repeat 10 times
193         for i in range(10):
194             # label the semi-data
195             semi_pred = model.predict(semi_all_X, batch_size=1024, verbose=True)
196             semi_X, semi_Y = dm.get_semi_data('semi_data', semi_pred, args.threshold, args.loss_function)
197             semi_X = np.concatenate((semi_X, X))
198             semi_Y = np.concatenate((semi_Y, Y))
199             print ('-- iteration %d  semi_data size: %d' %(i+1,len(semi_X)))
200             # train
201             history = model.fit(semi_X, semi_Y,
202                                 validation_data=(X_val, Y_val),
203                                 epochs=2,
204                                 batch_size=args.batch_size,
205                                 callbacks=[checkpoint, earlystopping] )
```

在 Kaggle 上的分數為 0.80117

3. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？
   (Collaborators:)
   答：

Model 架構

```
74      # BOW
75      bow_input = inputs
76      outputs = Dense(args.hidden_size,
77                      activation="relu",
78                      kernel_regularizer=regularizers.l2(0.1))(bow_input)
79      outputs = Dropout(dropout_rate)(outputs)
80
81      # DNN layer
82      outputs = Dense(args.hidden_size//2,
83                      activation='relu',
84                      kernel_regularizer=regularizers.l2(0.1))(outputs)
85      outputs = Dropout(dropout_rate)(outputs)
86      outputs = Dense(1, activation='sigmoid')(outputs)
87      model =  Model(inputs=inputs,outputs=outputs)
88
89      # optimizer
90      adam = Adam()
91      print ('compile model...')
92
93      # compile model
94      model.compile( loss=args.loss_function, optimizer=adam, metrics=[ 'accuracy',])
```

訓練過程同上題
(semi-supervised、epoch: 2、optimizer: adam、loss function: binary crossentropy)

在 Kaggle 上的分數為 0.51375

4. (1%) 請比較 bag of word 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的情緒分數，並討論造成差異的原因。
   (Collaborators:)
   答：

用 BOW 所預測的成績為 0.4965364、0.49584532
用 RNN 所預測的分數則為 0.4845556、0.97791356
因為 RNN 會考慮上下文、文法等差異，所以語句順序顛倒造成預測成績的差異很顯而易見，相反的，BOW 不會，所以兩句話分數差異不明顯。

6. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。
   (Collaborators:)
   答：

無標點符號版在 Kaggle 上分數為 0.79708、0.79493
有標點符號版在 Kaggle 上分數為 0.80237、0.79988
可見標點符號對此測資是有影響的。

7. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi-surpervised training 對準確率的影響。
   (Collaborators:)
   答：

標記 label 的方法為 self-training，對準確率有一定程度的影響。
有無 semi 的成績分別為 0.80164、0.801125，所以有用 semi-supervised 的方法的話，可以得到比較準確的預測結果。