學號：R05222003 系級： 物理碩二 姓名：吳愷訢

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:)

我實作了兩種不同的 NN, 一種是 pure RNN (Using GRU) , 另一種是 CNN+RNN(GRU). model 架構如下:

RNN (GRU) : Kaggle Public score : 0.80311	CNN + GRU : Kaggle public score : 0.81327
RnnGRU ((emb): Embedding(50001, 128, padding_idx=0) (emb2): Sequential ((0): Linear (128 -> 128) (1): ReLU () (2): Dropout (p = 0.1)) (rnn): GRU(128, 128, batch_first=True) (out): Sequential ((0): Linear (128 -> 128) (1): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True) (2): ReLU () (3): Dropout (p = 0.2) (4): Linear (128 -> 64) (5): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True) (6): ReLU () (7): Dropout (p = 0.6) (8): Linear (64 -> 1)) (sgactiv): Sigmoid ())	cnn_rnn_GRU ((emb): Embedding(50001, 128, padding_idx=0) (cnn): Sequential ((0): Conv2d(1, 128, kernel_size=[3, 128], stride=(1, 1), padding=(1, 0)) (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True) (2): ReLU () (3): Dropout2d (p=0.1) (4): Conv2d(128, 256, kernel_size=[3, 1], stride=(1, 1)) (5): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True) (6): ReLU () (7): Dropout2d (p=0.1) (8): MaxPool2d (size=(2, 1), stride=(2, 1), dilation=(1, 1))) (rnn): GRU(256, 128, batch_first=True) (out): Sequential ((0): Linear (128 -> 64) (1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True) (2): ReLU () (3): Dropout (p = 0.4) (4): Linear (64 -> 1)) (sgactiv): Sigmoid ())

在建字典方面,我取最常出現的前 50K 個字（包含標點符號）,因爲 label data 跟 testing data 最長皆沒有超過 40 個 word,所以 Pad 所有句子至固定長度 40 個字, learning rate = 0.002， batch-size = 256, L2- regularization = 0.0001 with Adam optimizer and 4~5 epoch. 另外, 也作了 semi-supervise , 取 output 90% confidence 以上的 unlabel data 加入當作新的 data set 後重新 train 一次. Validation 準確率與 Kaggle 想近. Kaggle 分數標示如上. 可以看到在 RNN 前加入 CNN, 除了 parameter 數降低外, 準確率也提升.

2. **(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:)

BOW 模型架構如下：

```
bow (
 (bow): Sequential (
          (0): Linear (50001 -> 256)
          (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True)
          (2): ReLU ()
          (3): Dropout (p = 0.1)
          (4): Linear (256 -> 128)
          (5): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True)
          (6): ReLU ()
          (7): Dropout (p = 0.6)
          (8): Linear (128 -> 1)
          (9): Sigmoid ()
 )
)
```

一樣只對出現頻率最多的前 50k 個字作 BOW, 其他沒出現的字歸類爲“其他”. learning rate = 0.0001， batch-size = 256, L2-regularization = 0.00001 with Adam optimizer and 10 epoch. Validation 分數爲 0.7241.

3. **(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：

以下爲兩個 model 對兩句的 output：

A：today is a good day, but it is hot
B：today is hot, but it is a good day

標點符號	**A**	**B**
BOW	**0.6021**	**0.6021**
CNN+RNN（GRU）	**0.2132**	**0.7341**

可以看出 CNN+RNN 的預測較符合預期, 而 BOW 則對 AB 兩句的 confidence 不高, 這是因爲在 BOW model 裏,我們不考慮字在句子前後的關係, 以這個例子來說, 前後關係影響情緒上的表達. 故考慮前後關係的 RNN model 能夠抓到不同.

4. **(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:)

標點符號	有	無
RNN（GRU）	**0.80311**	**0.77441**
CNN+RNN（GRU）	**0.81327**	**0.78201**

一樣只針對出現頻率最高的 50k 個字作 tokenizer. 由結果可以看出有標點符號的準確率較高. 以常理來說, 因爲標點符號在句子中對整個句子的判讀是至關重要的,而結果的確符合預期, 有標點符號的準確率較高.

5. **(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:)

如第一題所述, 對於兩個 model（RNN / RNN+CNN）, 皆取 output 90% confidence 以上的 unlabel data 加入當作新的 data set 後重新 train 一次. 另外因爲固定 padding 至總長 40 個字, 捨去 unlabel data 裏大於 40 個字的 sentences. 只拿小於 40 個字的句子裏 condfidence 大於 90%的 其 semi-supervise 前後準確率如下, 可以發現在做了 semi-supervice 後兩個 model 準確率皆有提升：

	Without Semi-supervise	Semi-supervise
RNN（GRU）	**0.79808**	**0.80311**
CNN+RNN（GRU）	**0.79937**	**0.81327**