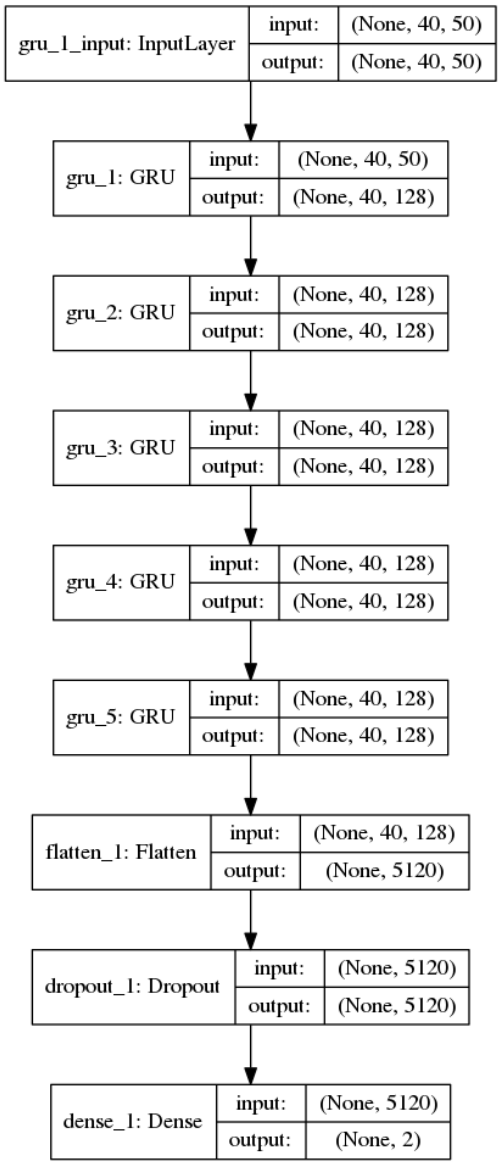
學號：B03901161 系級： 電機四 姓名：楊耀程

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: B03901165 謝世暐，B03901001 廖宜倫，B03901096 周晁德)
答：

模型架構:

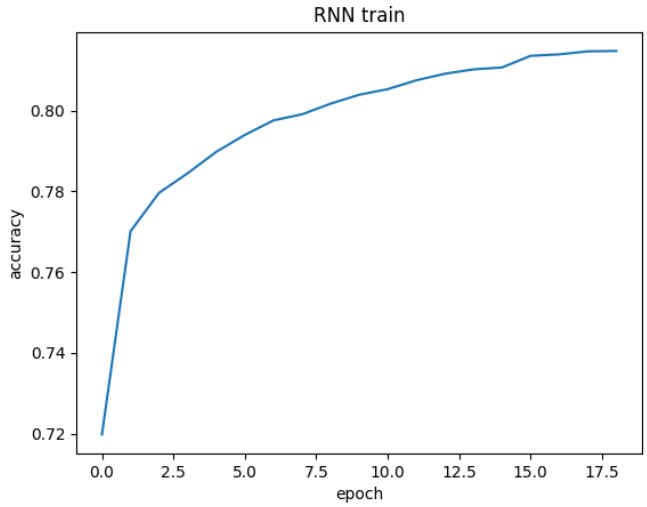


上圖是我實作 RNN model 的架構，首先利用 training data, testing data, no label data 裡面的所有字詞(沒有包含標點符號)，用 genism word2vec 訓練一個 word2vec model，**embedding dimension = 50**。

RNN model 的如圖所示，是 5 層的 GRU with dropout，加上 output layer。Max article length 為 40，加上 embedding dimension = 50，因此一開始的 input dimension 為 (number of training data, 40, 50)。

對於每個句子，我們都先通過 pretrained 好的 word2vec model，再通過 RNN model，來得到預測結果。

訓練過程與準確率:



先用 genism word2vec 訓練一個 word2vec model，embedding dimension = 50。接著訓練 RNN model。

Optimizer 為 Adam，learning rate = 0.001，decay = 0

Epochs = 19

Batchsize = 1024

使用 10000 筆 training data 作為 validation data，剩下 190000 筆 training data 拿來 train model。

可以發現在前幾個 epoch 訓練準確度就來到了 0.78，後來上升較慢，但是還是穩定上升。

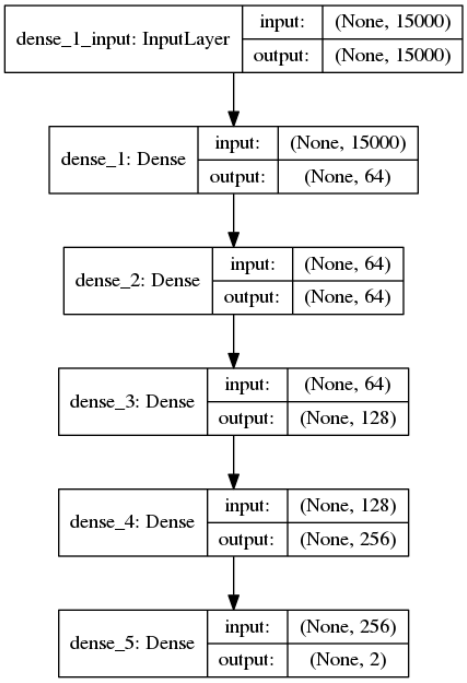
最終訓練準確率達到了 81 點多 percent，而 kaggle score 如下。

	kaggle score
RNN	0.81822

2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率何？(Collaborators: B03901165 謝世暐，B03901001 廖宜倫，B03901096 周晁德)

答：

模型架構:



上圖是我實作 BOW 的架構，首先做 Bag of word，再將轉換為向量的句子輸入 dense network 中。Dense 的安排如圖所示，是五層的 dense network，由除了最後一層之外，其餘 dense 中 node 的數量是遞增的(64->64->128->256)。此處用比較大的 max features(多看一些使用的字)，15000 個，但是訓練過程也發生了記憶體不足的問題，此處在下節會詳述。

訓練過程與準確率:

Optimizer 為 Adam，learning rate = 0.0001，decay = 0

Epochs (下方會詳細說明)

Batchsize = 2048

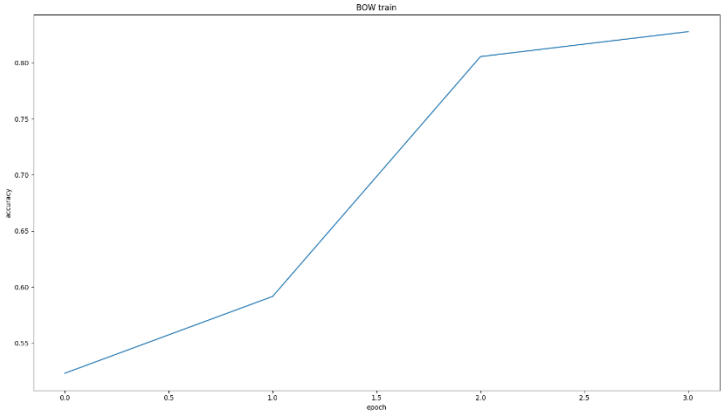
max_features = 15000

使用 10000 筆 training data 作為 validation data，剩下 190000 筆 training data 拿來 train model。

希望用比較大的 max features，15000 個，(多看一些使用的字)，一次跑的話記憶體不夠，因此使用分批輸入的方法: 將 training data 切成五份，一次輸入一份 data 並做 fitting，此 fitting 跑兩個 steps(epochs)，如此進行，跑滿五份 data 為一個完整的 epochs。

一共跑滿兩個完整的 epochs。

此處為了方便表示: 只顯示五份 data 中的其中一份的 training accuracy。一個完整的 epochs 中，會跑兩個 steps，因此也可以計算為: 跑了四個 epochs。因為中途 model 有經過另外四份 data 的 fitting，因此在此份 data 分割的訓練準確度是鋸齒狀，如圖所示。



訓練準確率(在第一份 data 分割中)達到了 0.8277，而 kaggle score 如下。

	kaggle score
BOW	0.79666

大約是 simple baseline 附近，總之，BOW 的結果較 RNN 差。

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: B03901165 謝世暐，B03901001 廖宜倫，B03901096 周晁德)
答：

	Score of "today is a good day, but it is hot"	Score of "today is hot, but it is a good day"
RNN	0.31209 (label = 0)	0.807776 (label = 1)
BOW	0.92311 (label = 1)	0.92311 (label = 1)

討論造成差異的原因:
使用 RNN 預測，得到的分數相差非常多，label 是不同的。因為 RNN 會考慮輸入的順序，所以這兩句話分數不同是可以預期的。"today is a good day, but it is hot"被判定為負面，推測是先 …good(正面)，然後接 but(轉折語意)，所以這樣的句子結構較容易被判斷為負面，而"today is hot, but it is a good day"被判定為正面，推測是先 ……(不知正面還負面)，然後接 but(轉折語意)+ good(正面)，這樣的句子結構感覺是正面的。以上的推論是考慮: today, hot 等語氣沒有 good 強烈，或是 hot 較負面(至少一定沒有 good 正面)。

而 BOW 得到的分數是一樣的(label 都是 1)，這也在預期之內，因為 BOW 考慮的是字詞出現的次數，這兩句話以 BOW 的方式換成 vector 時，應該是一樣的，因此 output 的 score 應該也要一樣，裡面有 good，為明顯的正面字詞，因此用 BOW 預測出來的 label 是正面的，也很合理。

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: B03901165 謝世暐，B03901001 廖宜倫，B03901096 周晁德)
答：

	kaggle score
沒有標點符號	0.81822
有標點符號	0.82077

有沒有標點符號，差別在於 keras 裡面的 text_to_word_sequence 中，filters 有沒有濾掉標點符號，此外在考慮標點符號時，word2vec 的 embedding model 也必須加上標點。在沒有標點符號的情形下(也就是預設的 filter)，準確率為 0.81822，但是有標點的情形下，kaggle 上準確率是 0.82077，上升了一些。此外，在 validation set 上面的準確度也微幅提升。
我打開了 testing data 來看，發現標點符號都有隔開，也還算整齊，除了逗號句號可以幫助分隔句子之外(當沒有連接詞時)，裡面的問號與驚嘆號應該多少也能幫忙理解語意，因此有涵蓋標點符號，應該是能提升結果的。

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: B03901165 謝世暐，B03901001 廖宜倫，B03901096 周晁德)
答：

	kaggle score
沒有 semi-supervised	0.81822
有 semi-supervised	0.82011

首先利用訓練好的 RNN (第一題的)，load model。讀取 no label data 並分批輸入(一次 200000 筆，避免記憶體不夠)，利用此 model 預測 no label data，**Threshold 設定為 0.95 與 0.05**，如果 output value 的值大於 0.95(很有信心 label 為 1)，或是 output value 的值小於 0.05(很有信心 label 為 0)，這樣的 data 再保留，output value 的值大於 0.95 的 label 設為 1，或是 output value 的值小於 0.05 的 label 設為 0。這些超過 threshold 的 unlabeled data 被加上 label 後與原本的 training data 放在一起，一起再 train RNN model。

在上述的操作之下，總共的 training data(包含原本的)有 418513 筆，使用他們來 train RNN，準確率有些微提高(如表所示)，理論上，可靠的 training data 數變多，準確率提高也很合理。