作業概述：

跟 HW4 的資料集是一模一樣的，針對假新聞做檢測，分辨一篇文章是否是可以信任的。但不同的是 HW4 採用機器學習演算法 GBDT、LightGBM、xgboost 而 HW5 採用目前最常用於做 NLP 的深度學習演算法 RNN、LSTM 對 train.csv 進行建模，並將 test.csv 帶入訓練好的模型，跑出之預測結果與 sample_submission 裡已有的真正的 label 進行比較計算 Accuracy，並分別 plot 出 RNN、LSTM 訓練過程中的 Accuracy 與 Loss 值變化。

結果與討論：

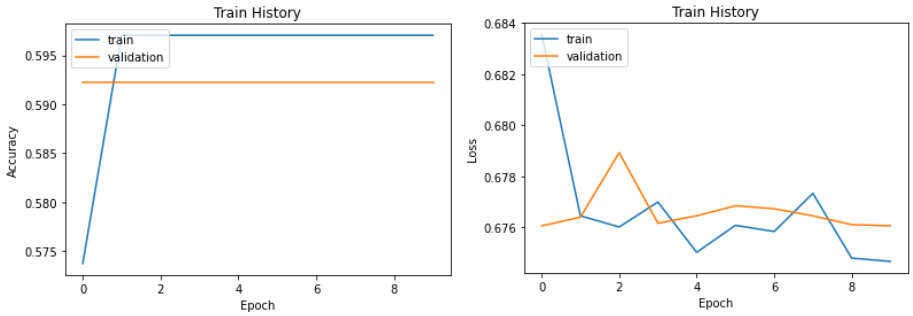
文字探勘前處理的辦法：

首先一開始發現其中一筆資料的 label 有問題，將那筆資料剔除，然後將 train.csv 裡所有文本都放入一個 list 裡面，接著透過 sklearn.feature_extraction.text 裡的 CountVectorizer 套件，先將內建預設的 stopwords 從文本中剔除掉，接著將 token 出現在至少 20 篇文本且出現在不超過 1/4 的文本當成 term 建立 term frequency 然後透過 TfidfTransformer 套件，利用 tf-idf 調整各個 term 在不同文本的權重，最後由於跑出來的 term 仍然有 7000 多個，利用 pca 進行降維，降維之後的結果搭配 label 值進行建模。

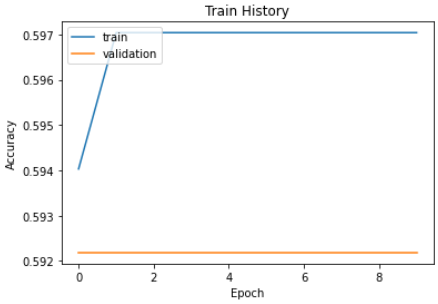
RNN、LSTM 模型之結果比較：(在測試集上)

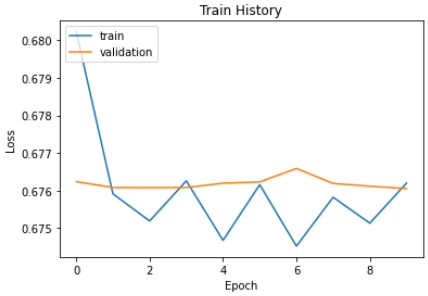
	RNN	LSTM
Accuracy	0.5052	0.5052

RNN 訓練過程中的 Accuracy 與 Loss 值變化



LSTM 訓練過程中的 Accuracy 與 Loss 值變化





XGboost、GBDT、lightgbm 模型之結果比較：(在測試集上)

	LightGBM	GBDT	xgboost
Accuracy	0.4916	0.4836	0.4916

有趣的是，HW4 出現了將訓練集代入訓練好的模型分析準確率達到了極高的水準，但在 HW5 卻沒有出現這樣的事情，準確率大約都在五六成打轉而已。然後最終的預測結果大概就是五成左右，表現有比 HW4 好一點，但仍不夠高，結合 HW4HW5 的經驗，可能原因有兩點：

1.有可能是因為訓練集與測試集本身的文本相似度就不高，導致即使再怎麼樣的訓練，也不可能在測試集上有著很好的表現。

2.也有可能是前處理做的不夠，使得無法有效地取出適合的 term 來建模，或是建模時超參數調整的不好，但因為跑一次模型需要花的時間不少，因次只稍微條了兩三次但都沒有更好地表現。

結論：

1.NLP 與圖像辨識是目前最人工智慧的主流，如果未來想走這條路，一定要熟悉這些 SOTA 的演算法與模型。

2.這次的作業讓我理解到現實生活中的數據可能就是那麼的凌亂不堪，因此可能最重要也是最花時間的前處理顯得更重要了。

程式碼：

In [1]:
```
# 引入相關套件
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
```

In [2]:
```
# 用pandas導入訓練集，分割符號採用table的方式，最後儲存成df1
df1 = pd.read_csv('train.csv', sep='\t')
df1
```

Out[2]:

	text	label
0	Get the latest from TODAY Sign up for our news...	1
1	2d Conan On The Funeral Trump Will Be Invited...	1
2	It's safe to say that Instagram Stories has fa...	0
3	Much like a certain Amazon goddess with a lass...	0
4	At a time when the perfect outfit is just one ...	0
...	...	...
4982	The storybook romance of WWE stars John Cena a...	0
4983	The actor told friends he's responsible for en...	0
4984	Sarah Hyland is getting real. The Modern Fami...	0
4985	Production has been suspended on the sixth and...	0
4986	A jury ruled against Bill Cosby in his sexual ...	0

4987 rows × 2 columns

前處理的方式跟HW4類似，但由於HW4模型準確率表現不夠好，試著對前處理做一些調整

In [3]:
```
# 看一下df1的columns有哪些
df1.columns
```

Out[3]:
```
Index(['text', 'label'], dtype='object')
```

In [4]:
```
# 看一下df1的index有哪些
df1.index
```

Out[4]:
```
RangeIndex(start=0, stop=4987, step=1)
```

In [5]:
```
# 發現df1某一個label出錯
df1["label"].value_counts()
```

Out[5]:
```
0        2972
1        2014
label       1
Name: label, dtype: int64
```

In [6]:
```
# 看一下哪一列為True
df1["label"] == 'label'
```

Out[6]:
```
0       False
1       False
2       False
3       False
4       False
        ...
4982    False
4983    False
4984    False
4985    False
4986    False
Name: label, Length: 4987, dtype: bool
```

In [7]:
```
# 發現錯誤的列
df1[df1["label"] == 'label']
```

Out[7]:

	text	label
1615	content	label

In [8]:
```
# 刪掉該行
df1 = df1.drop([1615], axis=0)
```

In [9]:
```
# 確認df1的label還有無出錯，發現已恢復正常
df1["label"].value_counts()
```

Out[9]:
```
0    2972
1    2014
Name: label, dtype: int64
```

In [10]:
```
# 更新一下df1的參數並看一下其樣子是否正確
df1.index = range(1, 4987)
df1
```

Out[10]:

	text	label
1	Get the latest from TODAY Sign up for our news...	1
2	2d Conan On The Funeral Trump Will Be Invited...	1
3	It's safe to say that Instagram Stories has fa...	0
4	Much like a certain Amazon goddess with a lass...	0
5	At a time when the perfect outfit is just one ...	0
...	...	...
4982	The storybook romance of WWE stars John Cena a...	0
4983	The actor told friends he's responsible for en...	0
4984	Sarah Hyland is getting real. The Modern Fami...	0
4985	Production has been suspended on the sixth and...	0
4986	A jury ruled against Bill Cosby in his sexual ...	0

4986 rows × 2 columns

In [11]:
```
# 引入自然語言處理套件
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

# 導入文字轉換成向量套件
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer # TfidfVectorizer為前兩者之混合，本程式碼因為中間需要拆開因此不會用到
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Steven\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Steven\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Steven\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [12]:
```
# 看一下隨機一個文本的樣子
ran = np.random.randint(4986)
df1.iloc[ran, 0]
```

Out[12]:
```
'Naomie Olindo has a voice, and she\'s not afraid to use it this season of Southern Charm. Of course, the southern belle has
never been afraid to share her true opinion, but this season, she\'s really not holding anything back, whether it\'s trying t
o get to the bottom of her issues with ex-boyfriend Craig Conover or calling out J.D. Madison.  But now as Naomie relives the
months following her breakup with Craig as this season of Southern Charm airs, she recently told The Daily Dish that she now
wishes she hadn\'t been "so mean to Craig." "Oh my gosh, I think that\'s probably my biggest regret. It\'s funny because now
I\'m in such a different headspace than I was eight or nine months ago, and so watching back how angry I was all the time, an
d I\'m like, \'Why did you let this drag out so long? Why didn\'t you just get over it and act like a nice human being?\'" Na
omie said during an interview over the phone. "It\'s very weird just to watch back, just the anger, really. I wish I had been
nicer to him because Craig is such a nice person. He\'s like a puppy. I wish I hadn\'t been such a bitch, but oh well. It is
what it is now."  After working on their relationship last season of Southern Charm, including going to see a counselor, Naom
ie and Craig ultimately decided to end their relationship prior to the start of Season 5. Naomie attributed the breakup to "a
million little things that added up" that eventually "just kind of blew up." "I think it was just a culmination of things tha
t at the end of it, we were just two very different people that couldn\'t get along," she explained. "It was sad because we d
id care about each other. You saw, we just could not get along. I was 22 when we started dating [she\'s 25 now]. When I was y
ounger, you just start dating someone because you have a couple things in common, and you\'re attracted to each other and you
have fun together. You don\'t think about the attributes that would make you get along with someone, their work ethic, and al
l the different adult things that matter that don\'t matter when you\'re a 22-year-old kid."  Those "adult things," as Naomie
put it, continued to be a topic of conversation between her and Craig, as we saw when she questioned his lifestyle choices at
Cameron Eubanks\' baby shower at Patricia Altschul\'s house earlier this season (clip above). "People think, \'Oh, why can\'t
```

In [13]:
```
# 將所有文本放入all_words_train這個串列內，等等要做詞頻處理
all_words_train = []
for i in range(len(df1)):
    all_words_train.append(df1.iloc[i, 0])

all_words_train
```

Out[13]:
```
['Get the latest from TODAY Sign up for our newsletter  No one ever truly gets over losing a loved one, and Blake Shelton is
no exception. He was just 14 when his older brother Richie died on Nov. 13, 1990. And, as Shelton noted in a tweet Monday, "I
t changed my life forever."  Richie was 24 when he died in a car accident in the Sheltons\' home state of Oklahoma. Two years
ago, Shelton sent out a message for the 25th anniversary of his loss:  Richie, who was Blake\'s half-brother (they shared a m
other), was a passenger in a car that collided with a school bus in Ada, south of Oklahoma City.  Richie, driver Redena McMan
us and a 3-year-old boy, Christopher McManus, all died during or shortly after the collision, while the bus driver and passen
gers were uninjured, according to police reports.  The accident has clearly remained with Blake, who told 60 Minutes in 2014,
"I remember picking up the phone to call him a week after he was dead, to tell him something. I was picking up the phone to c
all him, to tell him something I just saw on TV or, and it was like constantly a shock to me that he was dead."  Blake Shelto
n playing at TODAY\'s Halloween Extravaganza in New York City on Oct. 31. Getty Images  In 2011, Blake and his then-wife Mira
nda Lambert wrote a single called "Over You," which was inspired by Richie.  Still, the two brothers had bonded despite the a
ge difference; both shared a love of country music. "His bedroom was right across the hallway from mine when I was little," B
lake said in that interview. "And he was listening to Hank Williams, Jr. or Waylon, Lynyrd Skynyrd or Bob Seeger. I just, wha
tever was popular really, Richie loved all music.  "And I would be sitting there going, \'Man, that guy\'s my hero. That\'s t
he coolest guy. He’s my big brother.\'"  Follow Randee Dawn on Twitter.',
 '2d  Conan On The Funeral Trump Will Be Invited To - CONAN on TBS',
 'It’s safe to say that Instagram Stories has far surpassed its competitor Snapchat in popularity since it’s inception two ye
ars ago—and your favorite celebrities have hopped on the social media trend. Unlike a highly curated photo feed, Instagram St
ories is where celebrities seem to be comfortable enough to be raw and open.  Need something to do while you’re waiting in li
```

In [14]:
```
# 確認一下all_words_train的長度
len(all_words_train)
```

Out[14]:
```
4986
```

In [15]:
```
# 使用詞頻矩陣套件並儲存成CV，超參數稍微改一下，跟HW4比更嚴格：token最少出現在20個文本中、但不能出現在1/4的文本以上以及stop_words設為英文
CV = CountVectorizer(max_df=0.25, min_df=20, stop_words='english')

# 將文本all_words_train帶入套件CV中
CV.fit(all_words_train)
```

Out[15]:
```
CountVectorizer(max_df=0.25, min_df=20, stop_words='english')
```

In [16]:
```
# 看一下總feature數
fn = CV.get_feature_names()
len(fn)
```

Out[16]:
```
7806
```

In [17]:
```
# fit好的模型將all_words_train轉換
X1 = CV.transform(all_words_train)
X1
```

Out[17]:
```
<4986x7806 sparse matrix of type '<class 'numpy.int64'>'
        with 801936 stored elements in Compressed Sparse Row format>
```

In [18]:
```
# 確認一下矩陣的大小
X1.toarray().shape
```

Out[18]:
```
(4986, 7806)
```

In [19]:
```
# 但我們真正要取的是tf-idf，因為他比較全面，所以先使用這個套件部儲存成tfidf
tfidf = TfidfTransformer()
# 將前面得到的轉換出來的結果帶入tfidf去fit和transform
X2 = tfidf.fit_transform(X1)
```

In [20]:
```
# 看一下最終轉換出來的矩陣大小和模樣
matrix1 = X2.toarray()
print(matrix1.shape)
print(matrix1)
```

```
(4986, 7806)
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

In [21]:
```
# 將matrix1儲存成X_train
X_train = matrix1
```

In [22]:
```
# 將訓練集的label取出來並儲存成y_train
y_train = df1['label']
y_train
```

Out[22]:
```
1       1
2       1
3       0
4       0
5       0
      ..
4982    0
4983    0
4984    0
4985    0
4986    0
Name: label, Length: 4986, dtype: object
```

In [23]:
```
# 查看一下y_train的資料型態，發現是字串型態
y_train.dtype
```

Out[23]:
```
dtype('O')
```

In [24]:
```
# 將其轉換成整數的資料型態
y_train = y_train.astype('int')
y_train.dtype
```

Out[24]:
```
dtype('int32')
```

因為維度過高，因此先用PCA降維看看

In [25]:
```
# 導入PCA套件
from sklearn.decomposition import PCA
pca = PCA(n_components=0.9)
pca.fit(X_train)
```

Out[25]:
```
PCA(n_components=0.9)
```

In [26]:
```
# 做降維轉換
X_train_pca = pca.transform(X_train)
```

In [27]:
```
# 看一下降維之後的形狀
X_train_pca.shape
```

Out[27]:
```
(4986, 2142)
```

再來是對測試集進行處理

In [28]:
```
# 用pandas導入測試集，分割符號採用table的方式，最後儲存成df2
df2 = pd.read_csv('test.csv', sep='\t')
df2
```

Out[28]:

	id	text
0	2	The 2017 Teen Choice Awards ceremony was held ...
1	3	The concert, part of "The Joshua Tree Tour," w...
2	4	Selena Gomez refuses to talk to her mother abo...
3	5	This is worse than a lump of coal in your stoc...
4	6	Luann De Lesseps is going to rehab after her a...
...	...	...
1242	1244	Get the latest from TODAY Sign up for our news...
1243	1245	Jaden Smith claims that the Four Seasons Hotel...
1244	1246	Overview (3) Mini Bio (1) Faith Hill was bor...
1245	1247	CLOSE Aaron Paul dishes on 'The Path' Aaron P...
1246	1248	Meghan Edmonds was showered with love at her b...

1247 rows × 2 columns

In [29]:
```
# 將所有文本放入all_words_test這個串列內，等等要做詞頻處理
all_words_test = []
for i in range(len(df2)):
    all_words_test.append(df2.iloc[i, 1])

all_words_test
```

Out[29]:
```
['The 2017 Teen Choice Awards ceremony was held on August 13, 2017.[1] The awards celebrated the year\'s achievements in musi
c, film, television, sports, fashion, comedy, and the Internet, and were voted on by viewers living in the USA, aged 13 and o
ver through various social media sites.[2] A three hour musical festival called "Teen Fest" and hosted by Jake Paul was strea
med exclusively on YouTube with some of the event appearing during the Teen Choice broadcast.[3] Maroon 5 received the inaugu
ral Decade Award.[4] Throughout the show, several celebrities, including Vanessa Hudgens, Zendaya and Lauren Jauregui address
ed the aftermath of the 2017 Unite the Right rally and encouraged teens to speak out against violence and hate. This is the f
irst ceremony since 2002 to not include a host.  Performers [ edit ]  Presenters [ edit ]  Winners and nominees [ edit ]  The
first wave of nominations were announced on June 19, 2017.[8] The second wave was announced on July 12, 2017.[9] Winners are
listed first, in bold.[10]  Movies [ edit ]  Television [ edit ]  Movies & Television [ edit ]  Music [ edit ]  Digital [ edi
t ]  Fashion [ edit ]  Sports [ edit ]  Miscellaneous [ edit ]',
 'The concert, part of “The Joshua Tree Tour,” was slated to take place at The Dome at America’s Center. But this morning, th
e band and the concert promoter “regrettably” released a joint statement saying, “We have been informed by the St. Louis Poli
ce Department that they are not in a position to provide the standard protection for our audience as would be expected for an
event of this size. We have also been informed that local crowd security personnel would not be at full capacity. In light of
this information, we cannot in good conscience risk our fans’ safety by proceeding with tonight’s concert. As much as we regr
et having to cancel, we feel it is the only acceptable course of action in the current environment.”  Fans are reacting after
U2 canceled its St. Louis concert amid protests in the city. On Friday, activists began demonstrating after Jason Stockley, a
white police officer, was acquitted of murder in the shooting death of Anthony Lamar Smith, who is black. While people are pr
otesting both the verdict and police use of deadly force, cops are on the streets monitoring the demonstrations. Consequentl
```

In [30]:
```
# 確認一下all_words_test的長度
len(all_words_test)
```

Out[30]:
```
1247
```

In [31]:
```
# fit好的模型將all_words_test轉換
X3 = CV.transform(all_words_test)
X3
```

Out[31]:
```
<1247x7806 sparse matrix of type '<class 'numpy.int64'>'
        with 194250 stored elements in Compressed Sparse Row format>
```

In [32]:
```
# 確認一下矩陣的大小
X3.toarray().shape
```

Out[32]:
```
(1247, 7806)
```

In [33]:
```
# 將前面得到的轉換出來的結果帶入tfidf去fit和transform
X4 = tfidf.fit_transform(X3)
```

In [34]:
```
# 看一下最終轉換出來的矩陣大小和模樣
matrix2 = X4.toarray()
print(matrix2.shape)
print(matrix2)
```

```
(1247, 7806)
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

In [35]:
```
# 將matrix2儲存成X_test
X_test = matrix2
```

In [36]:
```
# 然後將前面做的pca套入X_test
X_test_pca = pca.transform(X_test)
```

In [37]:
```
# 看一下降維之後的形狀
X_test_pca.shape
```

Out[37]:
```
(1247, 2142)
```

In [38]:
```
# 將含有測試集label的csv檔導入
df3 = pd.read_csv('sample_submission (1).csv')
df3
```

Out[38]:

	id	label
0	2	1
1	3	1
2	4	0
3	5	0
4	6	0
...	...	...
1242	1244	0
1243	1245	0
1244	1246	1
1245	1247	1
1246	1248	1

1247 rows × 2 columns

In [39]:
```
# 將label儲存成y_test
y_test = df3.iloc[:, 1]
y_test
```

Out[39]:
```
0       1
1       1
2       0
3       0
4       0
       ..
1242    0
1243    0
1244    1
1245    1
1246    1
Name: label, Length: 1247, dtype: int64
```

In [40]:
```
# 將y_test轉變成array的形式
y_test = y_test.values
y_test
```

Out[40]:
```
array([1, 1, 0, ..., 1, 1, 1], dtype=int64)
```

In [41]:
```
# 查看一下y_test的資料型態，發現是整數，不需做處理
y_test.dtype
```

Out[41]:
```
dtype('int64')
```

使用RNN與LSTM進行建模

建模_RNN

In [42]:
```
# 引入相關套件
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.embeddings import Embedding
from keras.layers.recurrent import SimpleRNN
```

In [43]:
```
# 建立模型
modelRNN = Sequential()

# Embedding層將「數字list」轉換成「向量list」
# 輸出的維度是32，希望將數字list轉換為32維度的向量
# 輸入的維度是2142，也就是我們之前建立的字典是1969字
# 數字list截長補短後都是200個數字
modelRNN.add(Embedding(output_dim=32,
                       input_dim=2142,
                       input_length=200))

# 加入Dropout，避免overfitting
# 隨機在神經網路中放棄20%的神經元，避免overfitting
modelRNN.add(Dropout(0.2))
```

In [44]:
```
# 建立RNN層
# 建立16個神經元的RNN層
modelRNN.add(SimpleRNN(units=16))

# 建立隱藏層
# 建立256個神經元的隱藏層
# ReLU激活函數
modelRNN.add(Dense(units=256, activation='relu'))
modelRNN.add(Dropout(0.7))

# 建立輸出層
# 建立一個神經元的輸出層
# Sigmoid激活函數
modelRNN.add(Dense(units=1, activation='sigmoid'))
```

In [45]:
```
# 查看模型摘要
modelRNN.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 200, 32)           68544
_________________________________________________________________
dropout (Dropout)            (None, 200, 32)           0
_________________________________________________________________
simple_rnn (SimpleRNN)       (None, 16)                784
_________________________________________________________________
dense (Dense)                (None, 256)               4352
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257
=================================================================
Total params: 73,937
Trainable params: 73,937
Non-trainable params: 0
_________________________________________________________________
```

In [46]:
```
# 定義訓練模型
# Loss function使用Cross entropy
# adam最優化方法可以更快收斂
modelRNN.compile(loss='binary_crossentropy',
                 optimizer='adam',
                 metrics=['accuracy'])
```

In [47]:
```
# validation_split=0.2 設定80%訓練資料、20%驗證資料
# 執行10次訓練週期
# 每一批次訓練100筆資料
# verbose 顯示訓練過程
train_history1 = modelRNN.fit(X_train_pca, y_train,
                              epochs=10,
                              batch_size=100,
                              verbose=1,
                              validation_split=0.2)
```

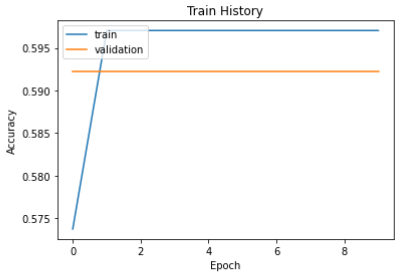
```
40/40 [==============================] - 18s 445ms/step - loss: 0.6770 - accuracy: 0.5970 - val_loss: 0.6762 - val_accuracy:
0.5922
Epoch 5/10
40/40 [==============================] - 18s 444ms/step - loss: 0.6750 - accuracy: 0.5970 - val_loss: 0.6764 - val_accuracy:
0.5922
Epoch 6/10
40/40 [==============================] - 18s 448ms/step - loss: 0.6761 - accuracy: 0.5970 - val_loss: 0.6768 - val_accuracy:
0.5922
Epoch 7/10
40/40 [==============================] - 18s 444ms/step - loss: 0.6758 - accuracy: 0.5970 - val_loss: 0.6767 - val_accuracy:
0.5922
Epoch 8/10
40/40 [==============================] - 19s 471ms/step - loss: 0.6773 - accuracy: 0.5970 - val_loss: 0.6764 - val_accuracy:
0.5922
Epoch 9/10
40/40 [==============================] - 17s 435ms/step - loss: 0.6748 - accuracy: 0.5970 - val_loss: 0.6761 - val_accuracy:
0.5922
Epoch 10/10
40/40 [==============================] - 17s 430ms/step - loss: 0.6747 - accuracy: 0.5970 - val_loss: 0.6761 - val_accuracy:
0.5922
```

In [48]:
```
# 看一下此RNN模型迭代過程中'loss','accuracy','val_loss','val_accuracy'的變化
train_history1.history
```

Out[48]:
```
{'loss': [0.6835592985153198,
  0.6764490008354187,
  0.6760095357894897,
  0.6769891381263733,
  0.6750165224075317,
  0.6760736107826233,
  0.6758334636688232,
  0.6773284673690796,
  0.6747915148735046,
  0.6746616363525391],
 'accuracy': [0.573721170425415,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
```

In [49]:
```
def show_train_history_Acc(train_history):
    plt.figure()
    plt.plot(train_history.history['accuracy'])
    plt.plot(train_history.history['val_accuracy'])
    plt.title("Train History")
    plt.xlabel("Epoch")
    plt.ylabel("Accuracy")
    plt.legend(["train", "validation"], loc="upper left")
    plt.show()
```

In [50]:
```
show_train_history_Acc(train_history1)
```



In [51]:
```
def show_train_history_Loss(train_history):
    plt.figure()
    plt.plot(train_history.history['loss'])
    plt.plot(train_history.history['val_loss'])
    plt.title("Train History")
    plt.xlabel("Epoch")
    plt.ylabel("Loss")
    plt.legend(["train", "validation"], loc="upper left")
    plt.show()
```

In [52]:
```
show_train_history_Loss(train_history1)
```

In [53]:
```
# 使用test測試資料及評估準確率
# 1247/32大約等於38.96
scores1 = modelRNN.evaluate(X_test_pca, y_test, verbose=1)
scores1
```

```
WARNING:tensorflow:Model was constructed with shape (None, 200) for input Tensor("embedding_input:0", shape=(None, 200), dtype=
float32), but it was called on an input with incompatible shape (None, 2142).
39/39 [==============================] - 2s 42ms/step - loss: 0.7080 - accuracy: 0.5052
```

Out[53]:
```
[0.7080350518226624, 0.5052124857902527]
```

Keras中model.evaluate()：返回的是損失值和你選定的指標值（例如，精度accuracy)

In [54]:
```
# 因此我們取第二個值
scores1[1]
```

Out[54]:
```
0.5052124857902527
```

建模_LSTM

In [55]:
```
# 匯入相關套件
from keras.models import Sequential
from keras.layers.core import Dense,Dropout,Activation,Flatten
from keras.layers.embeddings import Embedding
from keras.layers.recurrent import LSTM
```

In [56]:
```
# 建立模型
modelLSTM = Sequential()
# Embedding層將「數字list」轉換成「向量list」
# 輸出的維度是32，希望將數字list轉換為32維度的向量
# 輸入的維度是2142，也就是我們之前建立的字典是1969字
# 數字list截長補短後都是200個數字
modelLSTM.add(Embedding(output_dim=32,
                        input_dim=2142,
                        input_length=200))

# 加入Dropout，避免overfitting
# 隨機在神經網路中放棄20%的神經元，避免overfitting
modelLSTM.add(Dropout(0.2))
```

In [57]:
```
# 建立LSTM層
# 建立32個神經元的LSTM層
modelLSTM.add(LSTM(32))


# 建立隱藏層
# 建立256個神經元的隱藏層
modelLSTM.add(Dense(units=256,activation='relu'))
modelLSTM.add(Dropout(0.7))

# 建立輸出層
# 建立一個神經元的輸出層
modelLSTM.add(Dense(units=1,activation='sigmoid'))
```

In [58]:
```
# 查看模型摘要
modelLSTM.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 200, 32)           68544
_________________________________________________________________
dropout_2 (Dropout)          (None, 200, 32)           0
_________________________________________________________________
lstm (LSTM)                  (None, 32)                8320
_________________________________________________________________
dense_2 (Dense)              (None, 256)               8448
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 257
=================================================================
Total params: 85,569
Trainable params: 85,569
Non-trainable params: 0
_________________________________________________________________
```

In [59]:
```
# 定義訓練模型
# Loss function使用Cross entropy
# adam最優化方法可以更快收斂
modelLSTM.compile(loss='binary_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])
```

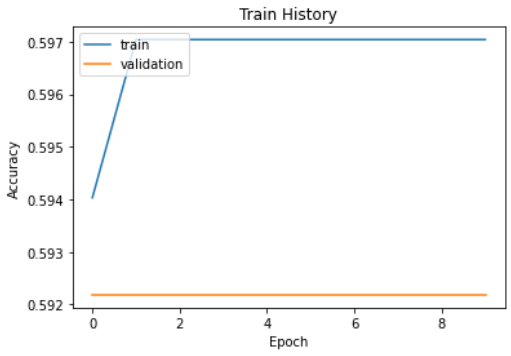
In [60]:
```
# validation_split=0.2 設定80%訓練資料、20%驗證資料
# 執行10次訓練週期
# 每一批次訓練200筆資料
# verbose 顯示訓練過程
train_history2 = modelLSTM.fit(X_train_pca, y_train,
                               epochs=10,
                               batch_size=100,
                               verbose=1,
                               validation_split=0.2)
```

```
Epoch 4/10
40/40 [==============================] - 41s 1s/step - loss: 0.6763 - accuracy: 0.5970 - val_loss: 0.6761 - val_accuracy: 0.5
922
Epoch 5/10
40/40 [==============================] - 42s 1s/step - loss: 0.6747 - accuracy: 0.5970 - val_loss: 0.6762 - val_accuracy: 0.5
922
Epoch 6/10
40/40 [==============================] - 42s 1s/step - loss: 0.6762 - accuracy: 0.5970 - val_loss: 0.6762 - val_accuracy: 0.5
922
Epoch 7/10
40/40 [==============================] - 42s 1s/step - loss: 0.6745 - accuracy: 0.5970 - val_loss: 0.6766 - val_accuracy: 0.5
922
Epoch 8/10
40/40 [==============================] - 42s 1s/step - loss: 0.6758 - accuracy: 0.5970 - val_loss: 0.6762 - val_accuracy: 0.5
922
Epoch 9/10
40/40 [==============================] - 43s 1s/step - loss: 0.6751 - accuracy: 0.5970 - val_loss: 0.6761 - val_accuracy: 0.5
922
Epoch 10/10
```

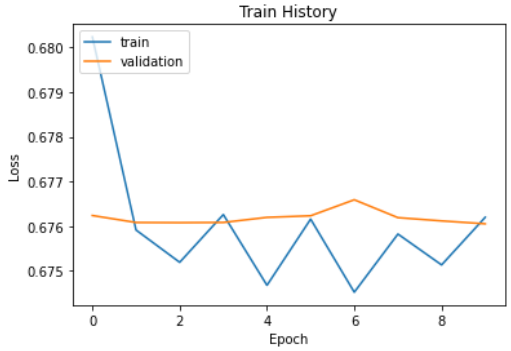
In [61]:
```
# 看一下此LSTM模型迭代過程中'loss','accuracy','val_loss','val_accuracy'的變化
train_history2.history
```

Out[61]:
```
{'loss': [0.6802324652671814,
  0.6759146451950073,
  0.6751920580863953,
  0.6762598752975464,
  0.6746779680252075,
  0.6761583685874939,
  0.6745235323905945,
  0.6758243441581726,
  0.6751336455345154,
  0.6762000918388367],
 'accuracy': [0.5940321087837219,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
  0.597041130065918,
```

In [62]:
```
show_train_history_Acc(train_history2)
```



In [63]:
```
show_train_history_Loss(train_history2)
```



In [64]:
```
# 使用test測試資料及評估準確率
scores2 = modelLSTM.evaluate(X_test_pca, y_test, verbose=1)
scores2
```

```
WARNING:tensorflow:Model was constructed with shape (None, 200) for input Tensor("embedding_1_input:0", shape=(None, 200), dtyp
e=float32), but it was called on an input with incompatible shape (None, 2142).
39/39 [==============================] - 4s 93ms/step - loss: 0.7085 - accuracy: 0.5052
```

Out[64]:
```
[0.7085297107696533, 0.5052124857902527]
```

In [65]:
```
# 取出accuracy
scores2[1]
```

Out[65]:
```
0.5052124857902527
```