

# Sentiment\_Analysis-binary-classification-BRNN-CuDNNGRU-Batchnormalization-AttentionLayer

January 25, 2020

## 1 Sentiment Analysis with an RNN

Run in Google Colab

View source on GitHub

<http://www.polyvista.com/blog/wp-content/uploads/2015/06/sentiment-customer-exp-large.png>

### 1.1 What is Sentiment Analysis?

Sentiment Analysis also known as opinion mining refers to the identification, extraction and study of sentiment states by using natural language processing, text analysis, computational linguistics and biometrics.

### 1.2 Sentiment Analysis with an Recurrent Neural Network

We will use a RNN for sentiment analysis because we care for the sequence in the data.

#### 1.2.1 Imports

```
[1]: import re
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

from tensorflow.keras.models import Sequential, load_model
from tensorflow.compat.v1.keras.layers import CuDNNGRU, Embedding,
↳Dropout,Dense, Bidirectional, BatchNormalization
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.optimizers import RMSprop, Adamax , Adam

from attention.layers import AttentionLayer
```

```
# import keras
# from keras.models import Sequential, load_model
# from keras.layers import Dense, Embedding, Dropout
# from keras.preprocessing.text import Tokenizer
# from keras.preprocessing.sequence import pad_sequences
import tensorflow as tf
from tensorflow.python.client import device_lib
```

```
[2]: from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.6
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
```

```
[3]: from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all" #This is for multiple print_
→statements per cell
```

```
[4]: value = tf.test.is_gpu_available(
    cuda_only=False,
    min_cuda_compute_capability=None
)
print ('***If TF can access GPU: ***\n\n',value) # MUST RETURN True IF IT CAN!!
```

WARNING:tensorflow:From <ipython-input-4-cb50da41978a>:3: is\_gpu\_available (from tensorflow.python.framework.test\_util) is deprecated and will be removed in a future version.

Instructions for updating:

Use `tf.config.list\_physical\_devices('GPU')` instead.

\*\*\*If TF can access GPU: \*\*\*

True

```
[5]: value = tf.config.list_physical_devices('GPU')
print(value)
```

```
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

```
[6]: print(device_lib.list_local_devices())
```

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
```

```

incarnation: 6462018811370594117
, name: "/device:XLA_CPU:0"
device_type: "XLA_CPU"
memory_limit: 17179869184
locality {
}
incarnation: 14460907413952961314
physical_device_desc: "device: XLA_CPU device"
, name: "/device:XLA_GPU:0"
device_type: "XLA_GPU"
memory_limit: 17179869184
locality {
}
incarnation: 11134824595277584103
physical_device_desc: "device: XLA_GPU device"
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 1259942707
locality {
  bus_id: 1
  links {
  }
}
incarnation: 3514069066457173148
physical_device_desc: "device: 0, name: GeForce MX150, pci bus id: 0000:02:00.0,
compute capability: 6.1"
]

```

```
[7]: tf.debugging.set_log_device_placement(True)
```

```
[8]: tf
print("Num GPUs Available: ", len(tf.config.experimental.
↪list_physical_devices('GPU')))
```

```
[8]: <module 'tensorflow' from '/home/erolerten/anaconda3/envs/venv-
tensorflow/lib/python3.7/site-packages/tensorflow/__init__.py'>
```

```
Num GPUs Available:  1
```

## 2 Place tensors on the CPU

### 3 with `tf.device('/GPU:0')`:

```

a = tf.constant([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]]) b = tf.constant([[1.0, 2.0], [3.0, 4.0], [5.0, 6.0]])
c = tf.matmul(a, b) print(c)

```

### 3.0.1 Loading in Dataset

```
[9]: data1 = pd.read_csv('Tweets.csv')
data2 = pd.read_csv('stanford-tweets.csv', sep=',')
# data1 = data1.sample(frac=1).reset_index(drop=True)
# data2 = data2.sample(frac=1).reset_index(drop=True)
print(data1.shape)
print(data2.shape)

data1.head()
data2.head()
```

(14640, 15)

(1600000, 2)

```
[9]:      tweet_id  airline_sentiment  airline_sentiment_confidence \
0  570306133677760513          neutral                1.0000
1  570301130888122368         positive                0.3486
2  570301083672813571          neutral                0.6837
3  570301031407624196         negative                1.0000
4  570300817074462722         negative                1.0000

      negativereason  negativereason_confidence      airline \
0              NaN                NaN  Virgin America
1              NaN                0.0000  Virgin America
2              NaN                NaN    Virgin America
3    Bad Flight                0.7033  Virgin America
4    Can't Tell                1.0000  Virgin America

      airline_sentiment_gold      name  negativereason_gold  retweet_count \
0              NaN    cairdin                NaN                0
1              NaN    jnardino                NaN                0
2              NaN  yvonnalynn                NaN                0
3              NaN    jnardino                NaN                0
4              NaN    jnardino                NaN                0

      text  tweet_coord \
0  @VirginAmerica What @dhepburn said.                NaN
1  @VirginAmerica plus you've added commercials t...                NaN
2  @VirginAmerica I didn't today... Must mean I n...                NaN
3  @VirginAmerica it's really aggressive to blast...                NaN
4  @VirginAmerica and it's a really big bad thing...                NaN

      tweet_created  tweet_location      user_timezone
0  2015-02-24 11:35:52 -0800                NaN  Eastern Time (US & Canada)
1  2015-02-24 11:15:59 -0800                NaN  Pacific Time (US & Canada)
2  2015-02-24 11:15:48 -0800    Lets Play  Central Time (US & Canada)
```

```

3  2015-02-24 11:15:36 -0800      NaN  Pacific Time (US & Canada)
4  2015-02-24 11:14:45 -0800      NaN  Pacific Time (US & Canada)

```

```

[9]:      sentiment      text
0  negative  @switchfoot http://twitpic.com/2y1zl - Awww, t...
1  negative  is upset that he can't update his Facebook by ...
2  negative  @Kenichan I dived many times for the ball. Man...
3  negative  my whole body feels itchy and like its on fire
4  negative  @nationwideclass no, it's not behaving at all...

```

Removing all columns except the `airline_sentiment` and `text` column.

```

[10]: data1 = data1[['airline_sentiment', 'text']]
      new_columns = ['sentiment', 'text']
      data1.columns = new_columns
      data1.head()

```

```

[10]:      sentiment      text
0    neutral      @VirginAmerica What @dhepburn said.
1  positive  @VirginAmerica plus you've added commercials t...
2    neutral  @VirginAmerica I didn't today... Must mean I n...
3  negative  @VirginAmerica it's really aggressive to blast...
4  negative  @VirginAmerica and it's a really big bad thing...

```

```

[11]: df = data1.append(data2, ignore_index = True)
      print(df.shape)
      df

```

(1614640, 2)

```

[11]:      sentiment      text
0    neutral      @VirginAmerica What @dhepburn said.
1  positive  @VirginAmerica plus you've added commercials t...
2    neutral  @VirginAmerica I didn't today... Must mean I n...
3  negative  @VirginAmerica it's really aggressive to blast...
4  negative  @VirginAmerica and it's a really big bad thing...
...
1614635  positive  Just woke up. Having no school is the best fee...
1614636  positive  TheWDB.com - Very cool to hear old Walt interv...
1614637  positive  Are you ready for your MoJo Makeover? Ask me f...
1614638  positive  Happy 38th Birthday to my boo of alll time!!! ...
1614639  positive  happy #charitytuesday @theNSPCC @SparksCharity...

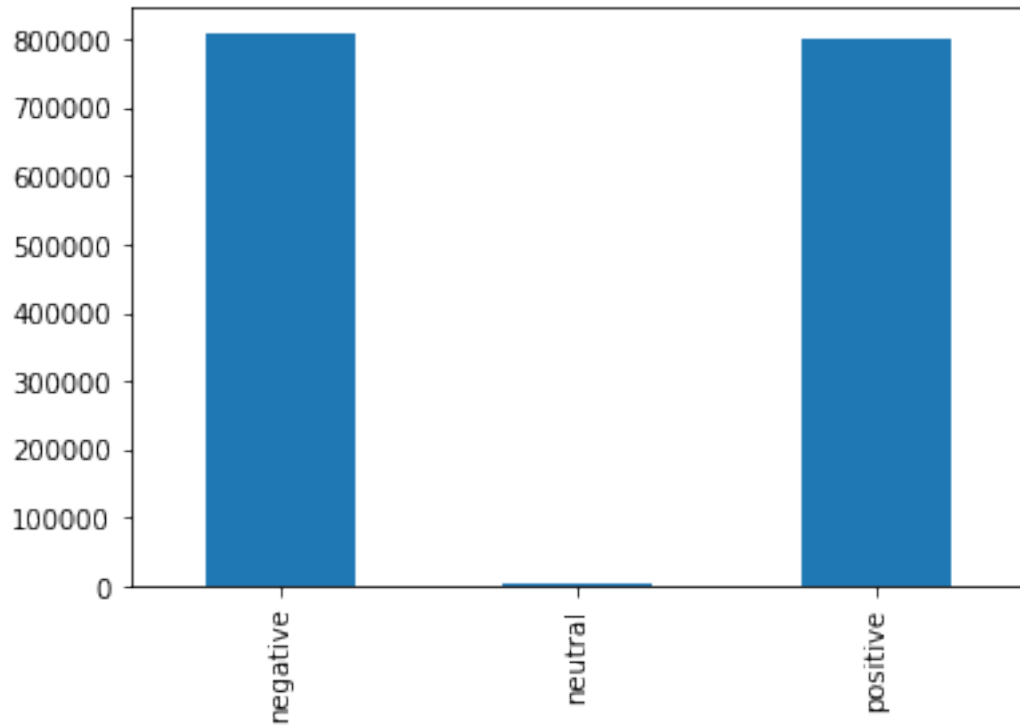
```

[1614640 rows x 2 columns]

### 3.0.2 Data exploration

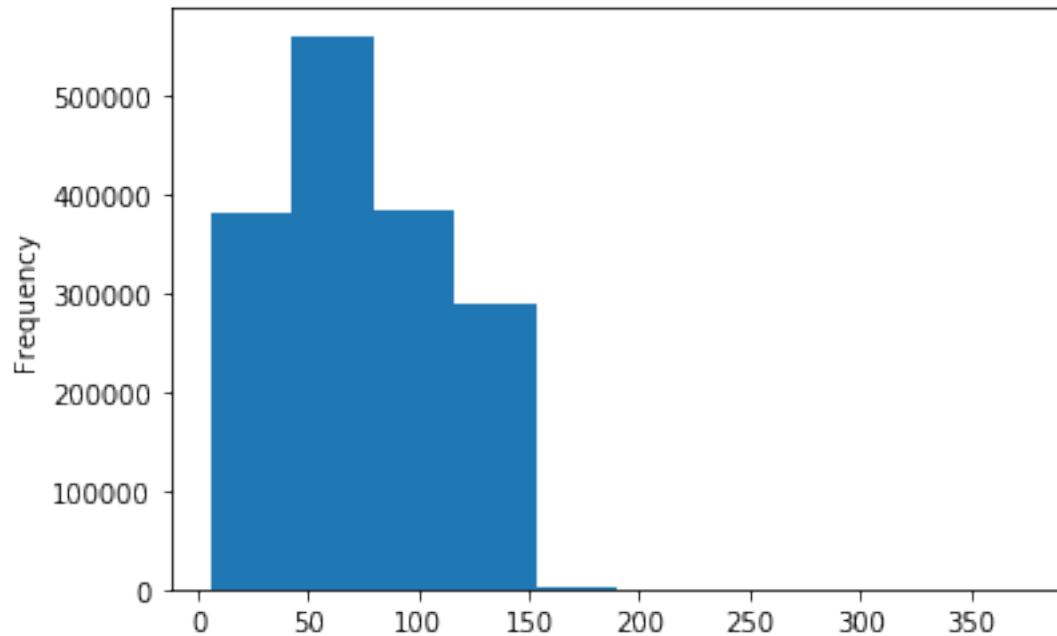
```
[12]: df['sentiment'].value_counts().sort_index().plot.bar()
```

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d3287f390>
```



```
[13]: df['text'].str.len().plot.hist()
```

```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d146b3a50>
```



### 3.0.3 Preprocessing

```
[14]: # How much of Dataset to be used
      frac = 0.2
```

```
[15]: # data['text'] = data['text'].str.replace('@VirginAmerica', '')
      # data.head()
      df = df.sample(frac=frac).reset_index(drop=True)
      df
```

```
[15]:
```

	sentiment	text
0	negative	I had a dream that all of da HATERS in the wor...
1	positive	@PoppyFlowerFibr Awesome!
2	negative	just about to go to a meeting
3	positive	@carnagefairy No, I'm sure there are still a m...
4	positive	@vero No no, done that already. I'm talking ab...
...	...	...
322923	negative	Working...With Broken Glasses
322924	negative	@Crystylepoppin what was funny lil sis ? Btw m...
322925	positive	@motherfuckinwar you will...im extremly jelous
322926	positive	-Going to underground today
322927	negative	My stone fell out of my moon ring and now it n...

```
[322928 rows x 2 columns]
```

```
[16]: df['text'].apply(lambda x: x.lower()) #transform text to lowercase
df['text'] = df['text'].apply(lambda x: re.sub('[^a-zA-z0-9\s]', '', x))
df['text'].head()
```

```
[16]: 0      i had a dream that all of da haters in the wor...
1              @poppyflowerfibr awesome!
2              just about to go to a meeting
3      @carnagefairy no, i'm sure there are still a m...
4      @vero no no, done that already. i'm talking ab...
...
322923              working...with broken glasses
322924      @crystylepoppin what was funny lil sis ? btw m...
322925      @motherfuckinwar you will...im extremly jelous
322926              -going to underground today
322927      my stone fell out of my moon ring and now it n...
Name: text, Length: 322928, dtype: object
```

```
[16]: 0      I had a dream that all of da HATERS in the wor...
1              PoppyFlowerFibr Awesome
2              just about to go to a meeting
3      carnagefairy No Im sure there are still a mill...
4      vero No no done that already Im talking about ...
Name: text, dtype: object
```

```
[17]: df['sentiment']
```

```
[17]: 0      negative
1      positive
2      negative
3      positive
4      positive
...
322923      negative
322924      negative
322925      positive
322926      positive
322927      negative
Name: sentiment, Length: 322928, dtype: object
```

```
[18]: df = df[df['sentiment'] != 'neutral']
```

```
[19]: df
```

```
[19]:      sentiment      text
0      negative  I had a dream that all of da HATERS in the wor...
1      positive              PoppyFlowerFibr Awesome
2      negative              just about to go to a meeting
```



```

3      positive  carnagefairly No Im sure there are still a mill...
4      positive  vero No no done that already Im talking about ...
...
322923 negative  WorkingWith Broken Glasses
322924 negative  Crystylepoppin what was funny lil sis Btw mis...
322925 positive  motherfuckinwar you willim extremly jelous
322926 positive  Going to underground today
322927 negative  My stone fell out of my moon ring and now it n...

```

```
[322313 rows x 2 columns]
```

```
[20]: vocabulary_size = 25000
```

```

[21]: tokenizer = Tokenizer(num_words=vocabulary_size, split=" ")
tokenizer.fit_on_texts(df['text'].values)

X = tokenizer.texts_to_sequences(df['text'].values)
X = pad_sequences(X) # padding our text vector so they all have the same length
X[:5]

```

```

[21]: array([[ 0,  0,  0,  1, 62,  4, 691, 18, 33,
              12, 738, 4162, 11,  3, 304, 618,  9, 26,
             106, 1021,  79, 26, 1159, 169, 197, 81, 10987,
              14, 794, 297, 10, 4162, 58, 87, 56, 437],
             [ 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, 156],
             [ 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, 20, 59,  2, 38,  2,  4, 665],
             [ 0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0, 37, 13, 193,
             79, 35, 68,  4, 1838, 231, 14,  3, 5562],
             [ 0,  0,  0,  0,  0,  0,  0,  0,  0,
              0,  0,  0,  0,  0,  0,  0,  0,  0,
              0, 17144, 37, 37, 189, 18, 192, 13, 504,
             59,  5, 1402, 749, 18959, 14604,  1, 1071, 1229]],
            dtype=int32)

```

### 3.0.4 Creating model

```
[22]: model = Sequential()
model.add(Embedding(vocabulary_size, 256, input_length=X.shape[1]))
model.add(Dropout(0.3))
model.add(Bidirectional(CuDNNGRU(256, return_sequences=True)))
model.add(Dropout(0.3))
model.add(Bidirectional(CuDNNGRU(256, return_sequences=True)))
model.add(AttentionLayer(name='attention'))
model.add(BatchNormalization())
model.add(Dense(2, activation='sigmoid'))
```

```
Executing op RandomUniform in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op Sub in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op Mul in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op Add in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op VarIsInitializedOp in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op LogicalNot in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op Assert in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op AssignVariableOp in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op RandomUniform in device
/job:localhost/replica:0/task:0/device:GPU:0
Executing op Sub in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Mul in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Add in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarIsInitializedOp in device
/job:localhost/replica:0/task:0/device:GPU:0
Executing op LogicalNot in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Assert in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op AssignVariableOp in device
/job:localhost/replica:0/task:0/device:GPU:0
Executing op RandomStandardNormal in device
/job:localhost/replica:0/task:0/device:GPU:0
Executing op Qr in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op DiagPart in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Sign in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Transpose in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Reshape in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op Fill in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
```

Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0  
 Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0  
 Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0

```
[23]: model.compile(loss='binary_crossentropy', optimizer='adam',  

  ↪metrics=['accuracy'])  

  model.summary()
```

Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0  
 Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 36, 256)	6400000
dropout (Dropout)	(None, 36, 256)	0
bidirectional (Bidirectional)	(None, 36, 512)	789504
dropout_1 (Dropout)	(None, 36, 512)	0
bidirectional_1 (Bidirectional)	(None, 36, 512)	1182720
attention (AttentionLayer)	(None, 512)	263168
batch_normalization (Batch Normalization)	(None, 512)	2048
dense (Dense)	(None, 2)	1026

Total params: 8,638,466  
 Trainable params: 8,637,442  
 Non-trainable params: 1,024

```
[24]: y = pd.get_dummies(df['sentiment']).values  

  [print(df['sentiment'][i], y[i]) for i in range(0,5)]
```

```
negative [1 0]  

positive [0 1]  

negative [1 0]  

positive [0 1]  

positive [0 1]
```

```
[24]: [None, None, None, None, None]
```

```
[25]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  

  ↪random_state=0)
```

### 3.0.5 Training model

```
[26]: batch_size = 32
epochs = 9
import time
from datetime import datetime
datetime = str(datetime.now())
csv_logger = tf.keras.callbacks.CSVLogger('training'+datetime+'.log')
start = time.time()
history = model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size,
    ↳ verbose=2, callbacks=[csv_logger])
end = time.time()
elapsed = end - start
print(elapsed/60, " minutes")
```

```
Executing op RangeDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op RepeatDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op MapDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op PrefetchDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op FlatMapDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op TensorDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op RepeatDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op ZipDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op ParallelMapDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op DatasetCardinality in device
/job:localhost/replica:0/task:0/device:CPU:0
Train on 257850 samples
Epoch 1/9
Executing op ModelDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op AnonymousIteratorV2 in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op MakeIterator in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op AssignVariableOp in device
/job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op VarHandleOp in device /job:localhost/replica:0/task:0/device:GPU:0
```

[illegible]

```
257850/257850 - 604s - loss: 0.3108 - accuracy: 0.8694
Epoch 5/9
257850/257850 - 598s - loss: 0.2748 - accuracy: 0.8865
Epoch 6/9
257850/257850 - 599s - loss: 0.2457 - accuracy: 0.9004
Epoch 7/9
257850/257850 - 607s - loss: 0.2251 - accuracy: 0.9095
Epoch 8/9
257850/257850 - 628s - loss: 0.2106 - accuracy: 0.9157
Epoch 9/9
257850/257850 - 600s - loss: 0.2033 - accuracy: 0.9188
Executing op DeleteIterator in device
/job:localhost/replica:0/task:0/device:CPU:0
92.0225713690122 minutes
```

### 3.0.6 Plotting Training History

```
[27]: # print(history)
```

```
[28]: import matplotlib.pyplot as plt

# Plot training & validation accuracy values
plt.plot(history.history['accuracy'])
# plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
# plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

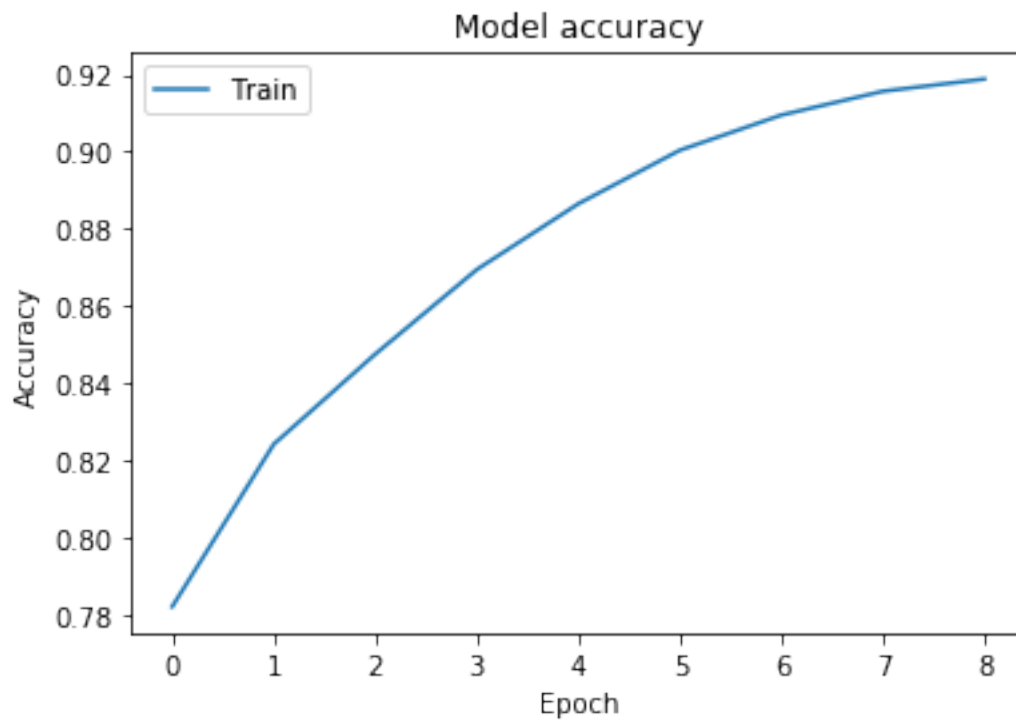
```
[28]: [<matplotlib.lines.Line2D at 0x7f7c18262a50>]
```

```
[28]: Text(0.5, 1.0, 'Model accuracy')
```

```
[28]: Text(0, 0.5, 'Accuracy')
```

```
[28]: Text(0.5, 0, 'Epoch')
```

[28]: <matplotlib.legend.Legend at 0x7f7c1b6e9c10>



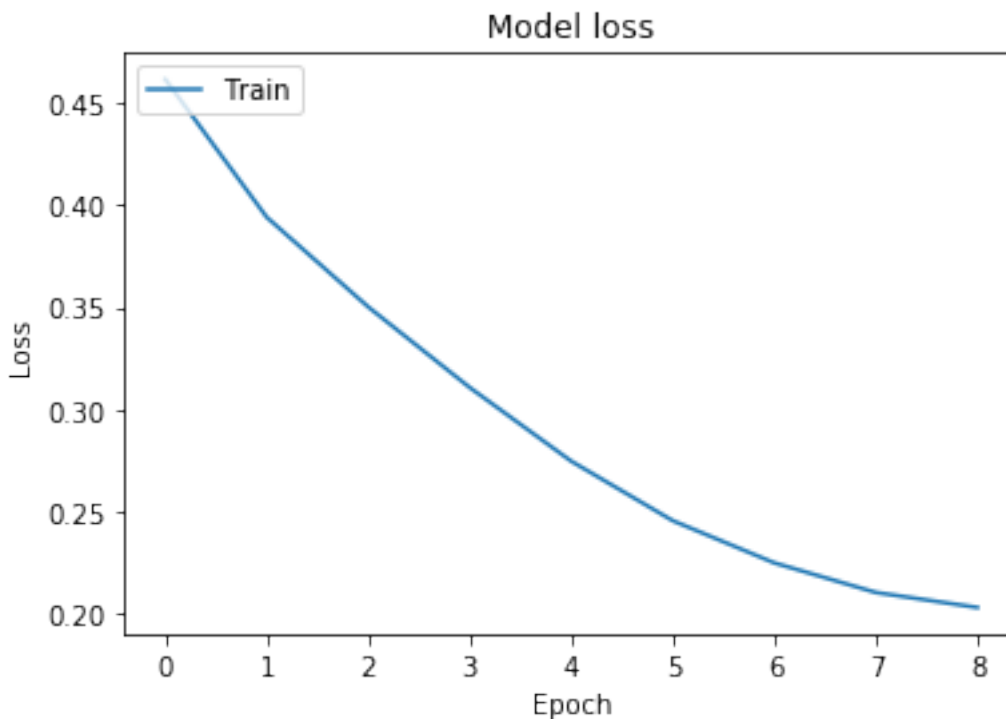
[28]: [<matplotlib.lines.Line2D at 0x7f7c1b676810>]

[28]: Text(0.5, 1.0, 'Model loss')

[28]: Text(0, 0.5, 'Loss')

[28]: Text(0.5, 0, 'Epoch')

[28]: <matplotlib.legend.Legend at 0x7f7c1b670950>



### 3.0.7 Testing model

```
[ ]: predictions = model.predict(X_test)
```

```
[print(df['text'][i], predictions[i], y_test[i]) for i in range(0, 5)]
```

```
Executing op RangeDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op RepeatDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op MapDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op PrefetchDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op FlatMapDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op TensorDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op RepeatDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op ZipDataset in device /job:localhost/replica:0/task:0/device:CPU:0
Executing op ParallelMapDataset in device
/job:localhost/replica:0/task:0/device:CPU:0
Executing op ModelDataset in device /job:localhost/replica:0/task:0/device:CPU:0
```



Executing op AnonymousIteratorV2 in device  
/job:localhost/replica:0/task:0/device:CPU:0  
Executing op \_\_inference\_distributed\_function\_222427 in device  
/job:localhost/replica:0/task:0/device:GPU:0

```
[ ]: accurate_prediction_count, inaccurate_prediction_count = 0, 0
for i, prediction in enumerate(predictions):
    if np.argmax(prediction)==np.argmax(y_test[i]):
        accurate_prediction_count += 1
    else:
        inaccurate_prediction_count += 1

total_predictions = accurate_prediction_count + inaccurate_prediction_count
print('Number of predictions: ', total_predictions)
print('Number of accurate predictions: ', accurate_prediction_count)
print('Number of false predictions: ', inaccurate_prediction_count)
print('Accuracy: ', accurate_prediction_count/total_predictions)
```

```
[ ]: name =_
      ↳'Sentiment_Analysis-binary-classification-BRNN-CuDNNGRU-Batchnormalization-AttentionLayer-9
```

```
[ ]: model.save(name+'.h5')
```

```
[ ]: # pos_count, neu_count, neg_count = 0, 0, 0
# real_pos, real_neu, real_neg = 0, 0, 0
# for i, prediction in enumerate(predictions):
#     if np.argmax(prediction)==2:
#         pos_count += 1
#     elif np.argmax(prediction)==1:
#         neu_count += 1
#     else:
#         neg_count += 1

#     if np.argmax(y_test[i])==2:
#         real_pos += 1
#     elif np.argmax(y_test[i])==1:
#         real_neu += 1
#     else:
#         real_neg +=1

# print('Positive predictions:', pos_count)
# print('Neutral predictions:', neu_count)
# print('Negative predictions:', neg_count)
# print('Real positive:', real_pos)
# print('Real neutral:', real_neu)
# print('Real negative:', real_neg)
```

```
[ ]: !jupyter nbconvert ↪ Sentiment_Analysis-binary-classification-BRNN-CuDNNGRU-Batchnormalization-AttentionLayer.  
      ↪ ipynb --to pdf
```

### 3.1 Improvements we could implement

Weight classes (because data is skew)

Train more epochs

Use bigger network

Try other word number

### 3.2 Resources

Recurrent Neural Networks Explained (my own post and video)

Sentiment Analysis (Wikipedia)

What is the best way to do sentiment analysis with Python? (Quora)

How to Do Sentiment Analysis (Siraj Raval)