Machine Learning Engineer Nanodegree

Capstone Project

Sujay Bhowmick
August 9th, 2018

I. Definition

Project Overview

In NLP Sentiment Analysis is a process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative or neutral.

Sentiment Analysis is useful in many ways. In my use case we are classifying Tweets of some relevance to domain like Finance related to both Micro and Macro events for which various credible financial analyst, activist, famous investors and financial news publishers are talking about in Twitter through their twitter handles (can not provide the Twitter handles of the users here due to privacy concerns). These Tweets are then used to guage the sentiment of the investors on certain topics and can be used to assess the investment decisions on that particular financial asset or stock after sentiments are combined with other analytical data dervided using different methodologies.

I am using a Machine Learning based approach to solve this problem and will develop a classification model, which is trained using prelabeled dataset of **positive**, **negative** content of the Twitter Tweets

Problem Statement

The goal of this Machine Learning Nanodegree Capstone project is to analyse the sentiment of various twitter tweets which is publicly available.

The tweets are related to financial news which have been labelled by a human for training and testing purpose of the Machine Learning model. There are approximately 8000 tweets which have been labelled with labels positive and negative for tweets indicating a **positive** sentiment and **negative** sentiment respectively.

Metrics

I have chosen accuracy as my metrics to assess whether model is working as expected. My training dataset is unbalanced between **positive** and **negative** labels (with *3843 positive* and *4508 negative* tweets), hence I would also like to look at confusion matrix and determine if whether the model is working as expected by looking at additional metrics like precision, recall and F1-Score

II. Analysis

Data Exploration

Data Collection

I have collected financially relevant messages from Twitter (had to do parallel project to make sure I only try to get financially relevant tweets, but its outside the scope of this project as it was used only to collected the data needed to train the model).

Data Preprocessing

Most of the tweets contains twitter handles (e.g. @sujay), hash tags, hyperlinks. Hence I need to preprocess the data and and replace each of them with normalized tags suchs as for hyperlinks, for hash tags, twitter handle with , special entities like quotes, ampersand and other various special characters which is outside the characted set of English

Exploratory Visualization

Following data is loaded using Pandas and explored. Below is Pandas Dataframe of the dataset

content	label	
0	omg its already 7:30 :O	1
1	Juuuuuuuuuuuuuuuussssst Chillin!!	1
2	handed in my uniform today . i miss you already	1
3	hmmmm.... i wonder how she my number	1
4	thanks to all the haters up in my face all day...	1
5	Feeling strangely fine. Now I'm gonna go liste...	1
6	You're the only one who can see this cause no ...	1
7	goodbye exams, HELLO ALCOHOL TONIGHT	1
8	uploading pictures on friendster	1
9	(: !!!!!! - so i wrote something last week. an...	1
10	... Do I need to even say it? Do I? Well, her...	1

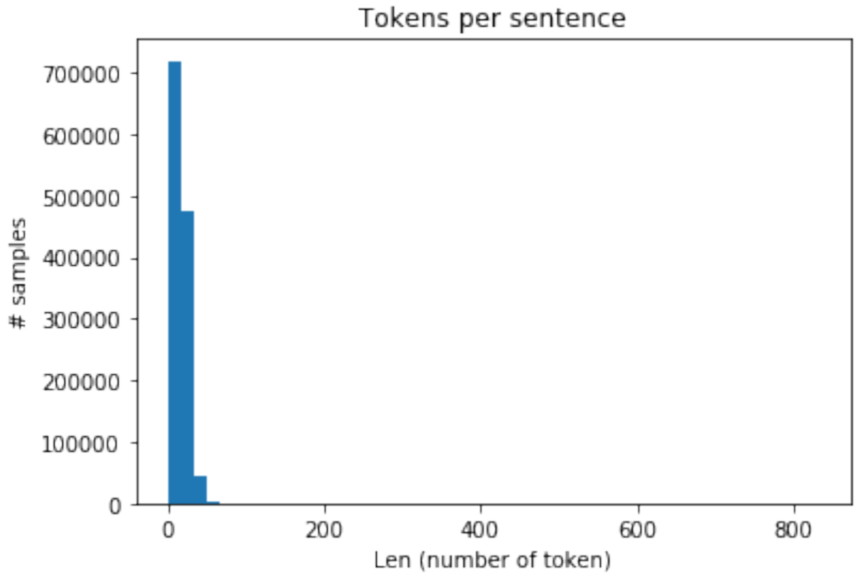
11	... health class (what a joke!)	1
12	< 3 GO TO THE SHOW TONIGHT	1
13	bathroom is clean..... now on to more enjoyabl...	1
14	boom boom pow	1
15	go give ur mom a hug right now.	1
16	Going To See Harry Sunday Happiness	1
17	- I always get what I want	1
18	I bend backwards	1
19	i get off work sooooon! i miss cody booo. have...	1
20	I hate allergies. Should I get my hair cut tom...	1
21	I'm really going to bed now...	1
22	Jin has a twitter.	1
23	just gonna smile...cuz it is what it is..and i...	1
24	Just got home, and I got to see my friend Zah...	1
25	oh thank you!	1
26	pleased	1
27	Rose and ood will be back in the Xmas Who spec...	1
28	Thanks, I need all the help i can get.	1
29	- that explains alot.	1
...	...	...
1552970	Yummy! Bummer...on my way to mom's for a famil...	0
1552971	Yummy, jess, your doingg everythingg I wanna b...	0
1552972	Yup - late for work! But not nearly as late as...	0
1552973	yup all alone on my bday happy bday every1 bor...	0
1552974	yup four more days to they get back and six mo...	0
1552975	yup gots a headache	0
1552976	yup the crying has started. Finished part I an...	0

1552977	Yup! So, they did take my phone. But, it will ...	0
1552978	Yup, I was right. I'm a sick girl, damn	0
1552979	Yup, Jr's collarbone is broken I can tell he i...	0
1552980	yup, my cat is still stranded she's just sitti...	0
1552981	Yup, needed the tissues for BSG but not quite ...	0
1552982	Yup, SPORE is updating now	0
1552983	Yup, that is all	0
1552984	Yup, the hints yesterday were correct. I'm now...	0
1552985	Yup. Bawling like a baby. So sad	0
1552986	Yup. Total sausage fest at the bar tonight. Ol...	0
1552987	Yupp. I'm going to miss have to frees a day	0
1552988	yuppies are real weird when you say " excuse ...	0
1552989	Yvonne left House is going to be in shambles n...	0
1552990	yzabellopez: ÃÂ yayayay cant wait, i just wis...	0
1552991	ze mother has spoken.. no playoffs fo' kathy t...	0
1552992	Zefron and BBV, please dont break-up.	0
1552993	Zese v posteli. Posloucham System of a down a ...	0
1552994	Zeta is getting old and I dont want her to go ...	0
1552995	Zicam Cold Remedy made my nose bleed i don't r...	0
1552996	zicam is being pulled from market!!! oh, the m...	0
1552997	Zigs the cat drat! you didn't win	0
1552998	Zip Lining today in Monteverde. Only 4 days left	0
1552999	Zipper flower fail Moving on to next project. ...	0

1553000 rows × 2 columns

Feature extraction

The below graph represents the distribution of token per sentence in the dataset samples. A custom tokenizer using Keras text preprocessing tokenizer is used to observe the distribution of words. We can then determine the maximum number of tokens in the training dataset. This is a good input feature which can be used for building the classifier.



Tokenization

```
First sample before preprocessing:
 <NAME/> actually if I were closer I'd stop by for some of your gluten free
pancakes! (with chocolate ice cream of course)

First sample after preprocessing:
 [    1   292    78     2   171  1893   401   339   121    12    66    13
    48 10960   375  2111    22   727   588   666    13   544]
```

Algorithms and Techniques

I have decided to use 1-D Convolution Neural to train the model on the training dataset. CNN model is very popular in image classification and very recently has show lot of success in text classification and natural language processing.

One of the desirable properties of CNN is that it preserves 2D spatial orientation in computer vision. Texts, like pictures, have an orientation. Instead of 2-dimensional, texts have a one-dimensional structure where words sequence matter. Words in the sentence are each replaced by a n-dimensional word vector, hence we fix one dimension of the filter to match the word vectors and vary the region size, h. Region size refers to the number of rows – representing word – of the sentence matrix that would be filtered. This is basic idea on how CNN can be useful for NLP and text classification.

For training and validation I am using train_test_split on the data set using sklearn to create the training and testing data sets.

Benchmark

The benchmark for this model is Afinn model which is currently in use but suffers from low accuracy of around 51%.

How Afinn model works

1. Methodology is keyword-matching,
2. Dictionaries of keywords and their Sentiment Value are pre-defined,
3. Input message is split by all non-alphanumeric characters into individual Tokens,
4. Each token is matched against the dictionary in the appropriate language,
5. Afinn's Sentiment Score is the sum of all Sentiment Values of the matched Tokens in the input message.

Afinn's weakness

1. Dictionary is more suited for analyzing product reviews
2. Methodology cannot reliably deal with even slightly complex language patterns (e.g. "not good")

The score from Afinn model is an integer value formula as Afinn returns the sum of values of all tokens in a message. Based on a small some test, this type of score made it more difficult to translate from Sentiment Score to Sentiment Label as the results may vary a lot.

III. Methodology

Data Preprocessing

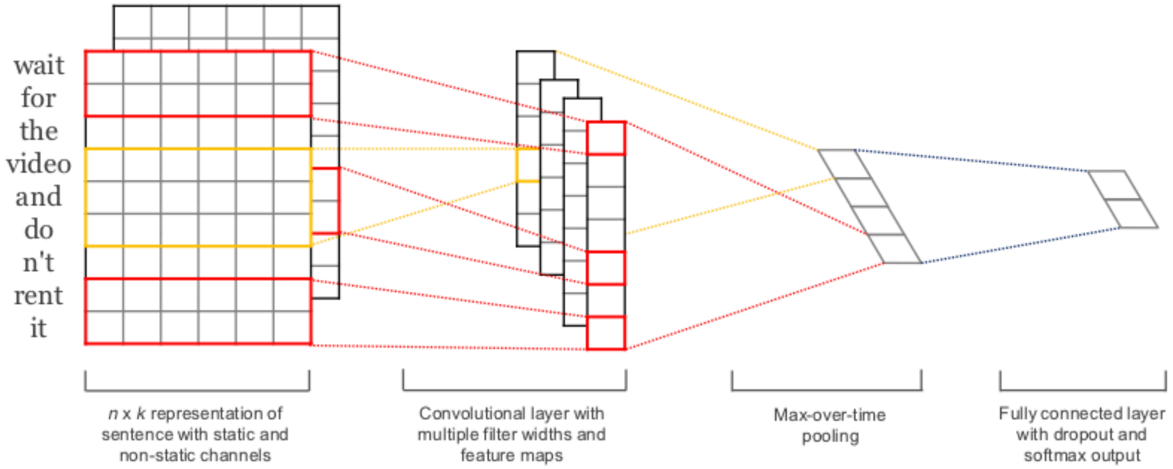
Data processing is part of the data analysis, the details of which is provided in the Data Analysis -> Data Exploration section, kindly refer to the same

Implementation

CNN Model

I have decided to use Convolutional Neural Network (CNN) classifier to predict the sentiment (positive or negative) of a tweet

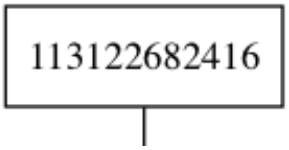
Following is the architecture diagram of the 1-D CNN which is implemented in this project and used

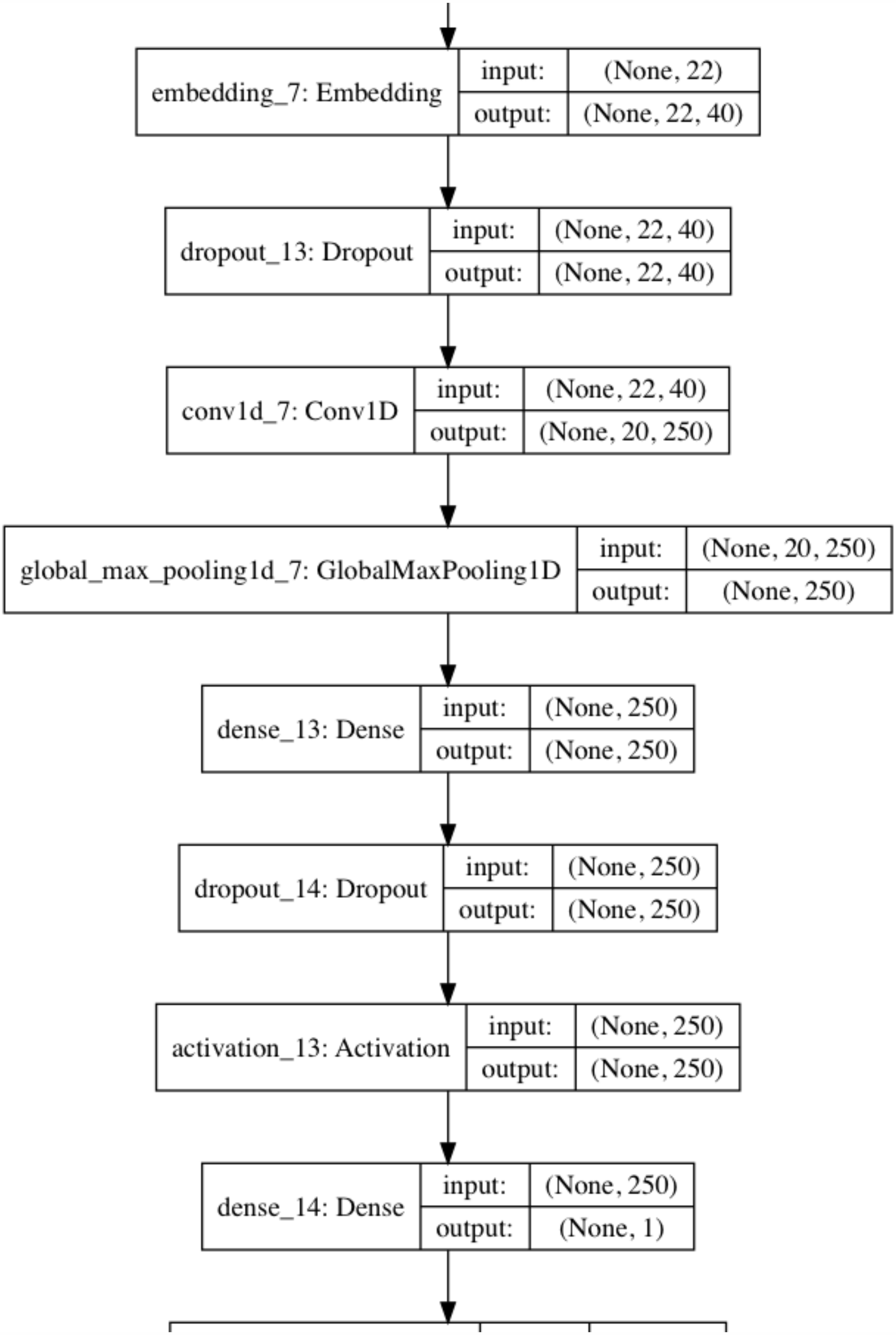

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

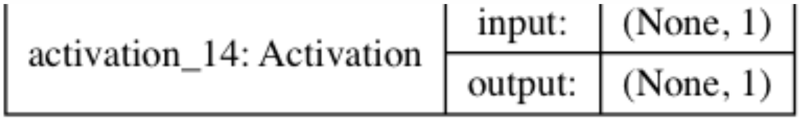
Kim Yoon’s [Convolutional Neural Networks for Sentence Classification]() as a reference architecture

Below is the model summary and plot

```
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_19 (Embedding)     (None, 22, 40)            1200000   
_________________________________________________________________
dropout_37 (Dropout)         (None, 22, 40)            0         
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 20, 250)           30250     
_________________________________________________________________
global_max_pooling1d_19 (Glo (None, 250)               0         
_________________________________________________________________
dense_37 (Dense)             (None, 250)               62750     
_________________________________________________________________
dropout_38 (Dropout)         (None, 250)               0         
_________________________________________________________________
activation_37 (Activation)   (None, 250)               0         
_________________________________________________________________
dense_38 (Dense)             (None, 1)                 251       
_________________________________________________________________
activation_38 (Activation)   (None, 1)                 0         
=================================================================
Total params: 1,293,251
Trainable params: 1,293,251
Non-trainable params: 0
_________________________________________________________________
```




embedding_7: Embedding
input: (None, 22)
output: (None, 22, 40)
dropout_13: Dropout
input: (None, 22, 40)
output: (None, 22, 40)
conv1d_7: Conv1D
input: (None, 22, 40)
output: (None, 20, 250)
global_max_pooling1d_7: GlobalMaxPooling1D
input: (None, 20, 250)
output: (None, 250)
dense_13: Dense
input: (None, 250)
output: (None, 250)
dropout_14: Dropout
input: (None, 250)
output: (None, 250)
activation_13: Activation
input: (None, 250)
output: (None, 250)
dense_14: Dense
input: (None, 250)
output: (None, 1)



```
def get_model():

    # CNN Model

    NUM_FILTERS = 250
    KERNEL_SIZE = 3
    HIDDEN_DIMS = 250

    model = Sequential()

    # We use embedding layer which maps our vocabulary indices into
EMBEDDING_DIM       dimensions
    model.add(Embedding(VOCAB_SIZE, EMBEDDING_DIM, input_length=MAX_LEN))
    model.add(Dropout(0.2))

    # Adding Convolution1D
    model.add(Conv1D(NUM_FILTERS,
                     KERNEL_SIZE,
                     padding='valid',
                     activation='relu',
                     strides=1))

    # Add a max pooling:
    model.add(GlobalMaxPooling1D())

    # Add a simple hidden layer:
    model.add(Dense(HIDDEN_DIMS))
    model.add(Dropout(0.2))
    model.add(Activation('relu'))

    # We project onto a single unit output layer, and use sigmoid function
    model.add(Dense(1))
    model.add(Activation('sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=
['accuracy'])
    return model
```

Word Tokenizer

Before using the data for the model, we need to process the tweet content to equivalent word vector. For this purpose we will use Keras Tokenizer to convert each word into a corresponding integer identifier. In order for us to use the content in the Model we must ensure the length of the content is same. We can do this by using the Keras **sequence.pad_sequences** function. All content greater than MAX_LEN will be truncated and text which are less than MAX_LEN will be padded to get the same length.

```
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding, GlobalMaxPooling1D, Flatten,
Conv1D, Dropout, Activation
from keras.preprocessing.text import Tokenizer
import tensorflow as tf
import numpy as np
from keras.utils.vis_utils import plot_model
from tensorflow import set_random_seed
from numpy.random import seed
seed(1)
set_random_seed(2)

tweet_tokenizer = Tokenizer(num_words=VOCAB_SIZE)
tweet_tokenizer.fit_on_texts(df['content'].values)
X = tweet_tokenizer.texts_to_sequences(df['content'].values)

X = sequence.pad_sequences(X, maxlen=MAX_LEN, padding="post", value=0)

y = df['label']

def print_sample_before_after_tokenizing():
    print('First sample before preprocessing: \n', df['content'].values[0],
'\n')
    print('First sample after preprocessing: \n', X[0])

print_sample_before_after_tokenizing()
```

Refinement

Since I have limited labelled data, the earlier model was running on dataset which is split 80:20 between training and test set and generally suffered from high bias. The accuracy of the model was about 79%. In order to overcome the limitation of the model, I decided to use corss validation technique to reduce bias and also reduces variance as most of the data is also being used in validation set.

Also adjusting the parameter MAX_LENGTH (max length of the tokens) in the data set helps in increasing accuracy of the model as observed during pre refinement activity.

IV. Results

Model Evaluation and Validation

We applied CNN to the final model and has shown improved results and reported a validation accuracy score of 79%. Also changing some of the hyper parameters did not affect the accuracy score and confusion matrix of the model.

The Hyperparameter values which holds good for the current model.

```
# Number of examples used in each iteration
BATCH_SIZE = 32
# Size of vocabulary dictionary
VOCAB_SIZE = 30000
# Max length of tweet as per the plot above
MAX_LEN = 22
# Dimension of word embedding vector
EMBEDDING_DIM = 40
```

```
SENTIMENT_LABELS = ['negative', 'positive']

filepath="models/weights-improvement-{epoch:02d}-{val_acc:.2f}.hdf5"

checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1,
save_best_only=True, mode='max')
callback_list = [checkpoint]


EPOCHS = 2 # Number of passes through entire dataset
history = model.fit(X_train, y_train,
              batch_size=BATCH_SIZE,
              epochs=EPOCHS,
              validation_split=0.1, callbacks=callback_list, verbose=0)

# Evaluate the model
score, acc = model.evaluate(X_test, y_test, batch_size=BATCH_SIZE)
print('\nAccuracy: ', acc*100)

pred_classes = model.predict_classes(X_test)

plot_confusion_matrix(y_test, pred_classes)
```

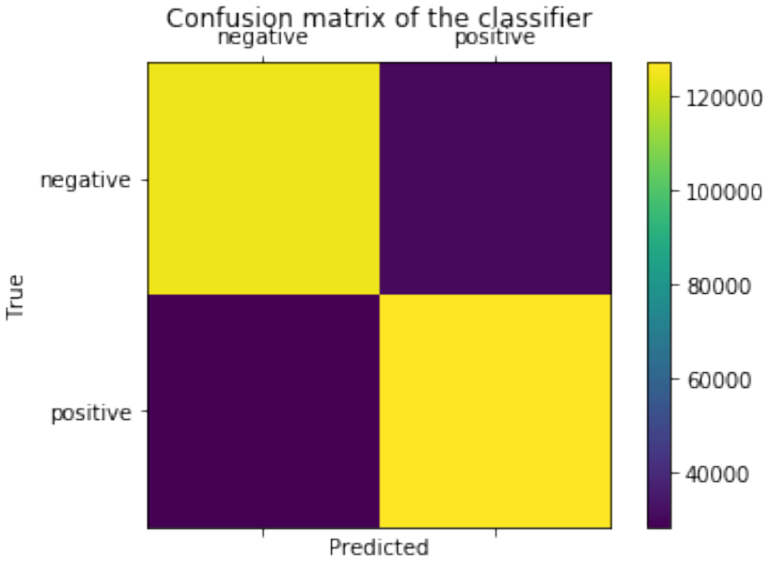
The model is tested with various inputs and has an accuracy of 79% . Further we have the confusion matrix which gives us better idea of our classification model

```
Epoch 00001: val_acc improved from -inf to 0.80551, saving model to
models/weights-improvement-01-0.81.hdf5

Epoch 00002: val_acc improved from 0.80551 to 0.81091, saving model to
models/weights-improvement-02-0.81.hdf5
310600/310600 [==============================] - 11s 34us/step

Accuracy:  81.0592401802962
```

Confusion Matrix:



```
  precision    recall  f1-score   support

   negative       0.80      0.82      0.81    152784
   positive       0.82      0.81      0.81    157816

avg / total       0.81      0.81      0.81    310600
```

V. Conclusion

Free-Form Visualization

Below code predicts Sentiment Lables and compares it to Human provided lables and based on prediction results of few tweets and seems like my model is doing much better than the benchmark model

```
SENTIMENT_LABELS = ['negative', 'positive']
def get_prediction(tweet):
    # Preprocessing step
    tweet_words_array = tweet_tokenizer.texts_to_sequences([tweet])
    tweet_words_array = sequence.pad_sequences(tweet_words_array,
maxlen=MAX_LEN, padding="post", value=0)

    #Predict the sentiment label and score
    score = model.predict(tweet_words_array)[0][0]
    prediction = SENTIMENT_LABELS[model.predict_classes(tweet_words_array)[0]
[0]]
    print('Tweet:', tweet, '\nPrediction:', prediction, '\nScore: ', score)
    print('\n')
    return prediction, score

# Test Prediction
prediction = get_prediction(". RT @SpryGuy: The CEO of Papa John's stiffs and
cheats his own employees so he can live in this castle with a moat. NEVER buy
Papa John's pi…")
assert prediction[0] == "negative"
prediction = get_prediction(". GVC Holdings consummated the acquisition of
Ladbrokes Coral https://t.co/xaN4ACA0h6 https://t.co/ZNm0gmXLK7")
assert prediction[0] == "positive"
prediction = get_prediction(". #AtlasMara holding is a real ingenious feat in
The financial fraternity..am amazed at the forge ahead they posses..
#mindblown")
assert prediction[0] == "positive"
prediction = get_prediction(". Boeing hit hard by tariff and trade war
headlines today, down -3.5%. Also note the very ugly price/momentum diverge
https://t.co/h9bfT95yWZ")
assert prediction[0] == "negative"
prediction = get_prediction("Didn't see this one coming but makes so much
sense... Amazon to Buy Whole Foods in $13.4 Billion Deal
https://t.co/tKcF9dUwct")
assert prediction[0] == "positive"
prediction = get_prediction("Starbucks Corporation (SBUX) Stock Isn't as Bad
as it Looks. Starbucks Corporation (Nasdaq: SBUX) is making aggressive changes
to get its stock back on track. The latest change the company announced this
week is the departure of CFO Scott Maw, and analysts say")
assert prediction[0] == "positive"
```

```
Tweet: . RT @SpryGuy: The CEO of Papa John's stiffs and cheats his own
employees so he can live in this castle with a moat. NEVER buy Papa John's pi…
Prediction: negative
Score:  0.30618456



Tweet: . GVC Holdings consummated the acquisition of Ladbrokes Coral
https://t.co/xaN4ACA0h6 https://t.co/ZNm0gmXLK7
Prediction: positive
Score:  0.6592547



Tweet: . #AtlasMara holding is a real ingenious feat in The financial
fraternity..am amazed at the forge ahead they posses.. #mindblown
Prediction: positive
Score:  0.9901579



Tweet: . Boeing hit hard by tariff and trade war headlines today, down -3.5%.
Also note the very ugly price/momentum diverge https://t.co/h9bfT95yWZ
Prediction: negative
Score:  0.17169975



Tweet: Didn't see this one coming but makes so much sense... Amazon to Buy
Whole Foods in $13.4 Billion Deal https://t.co/tKcF9dUwct
Prediction: positive
Score:  0.56883657



Tweet: Starbucks Corporation (SBUX) Stock Isn't as Bad as it Looks. Starbucks
Corporation (Nasdaq: SBUX) is making aggressive changes to get its stock back
on track. The latest change the company announced this week is the departure
of CFO Scott Maw, and analysts say
Prediction: positive
Score:  0.7268185
```

As you can see our model predicted all the labels correctly for few sample tests of the data.

Reflection

The most important thing in this project I understood about the concept of overfitting or undercutting in some case.

Initially I decied to use KFolds for training and testing my model, but quickly the model started to show signs of overfitting, but the test validation was stable. I realised that my model will not generalize well.

I then decided to try another method for splitting the data and use the training set and test set to measure the metrics of the preformance and my model is not overfitting anymore and

Improvement

With more labeled data and with help of new architecture such RNN, we can definitely improve the quality of our model. I have not used RNN, but I was suggested to try using RNN for Sentiment Analysis and not CNN which is essential is a model know to generally used for image classification.