Multiclass Sentiment Analysis for movie reviews using Deep Learning

Keval Hiren Sharma
Student ID: 1114333
Faculty of Graduate Studies
Dept. Masters of Computer Science
Lakehead University
Thunder Bay, Ontario
ksharm13@lakeheadu.ca

Abstract—**The report provides implementation of Sentiment analysis for movie reviews using deep learning. For this purpose, the dataset of movie reviews of Rotten Tomatoes is used. The solution is based on 1D Convolutional Neural Network. Also, multi class sentiment analysis is performed rather than binary classification. TF-IDF is used for extraction of features from the text. The model is trained using Keras library which is an open source library for training neural network based deep learning models. CNN classifier is used for training the model.**

I. INTRODUCTION

In the recent years, a large number of websites have been developed that allow users to contribute, modify and grade the content. As a result, people express their opinions on these websites. Usually, the reviews are longer and may consist of longer paragraphs. The analysis of these reviews can help in understanding the mindset of people towards particular product or a movie.

Movie reviews are important way to gauge the performance of the movies. The numerical ratings or stars given to a movie tells about the success or failure of a movie.

Sentiment analysis is the interpretation and classification of emotions i.e. positive, negative or neutral within the text data using text classification techniques. It helps in identifying the customer sentiment toward products, brands or services in online conversations and feedback.

A sentiment analysis model basically determines polarity within a text (for e.g. a positive or negative opinion). The understanding of people's opinion are important for business as their feedbacks can be helpful in determining their needs and meeting their needs.

There are various types of sentiment analysis :

1) **Fine grained sentiment analysis :-** Here, a variety of polarities are considered such as very positive, positive, neutral, negative, very negative and the ratings are given in the form of stars (for e.g. 5 stars for very positive and 1 star for very negative).

2) **Emotion detection :-** It aims at detecting various emotions such as happiness, frustration, anger, sadness and so on.

3) **Aspect based sentiment analysis :-** Here, a particular aspect mentioned by people in positive, negative or neutral way for a sentiment is analyzed.

4) **Multilingual sentiment analysis :-** It involves the analysis of the sentiments written in different languages. It is the most difficult among other types.

Rotten Tomatoes is an American review-aggregation website for film and television. It first collects the online reviews from writers who are certified members of various writing guides or film critic associations. The website keeps the track of all the reviews counted for each film and calculates the percentage of positive reviews. If the positive reviews make up more than 60 percent, then the movie is considered as "fresh" otherwise it is considered as "rotten".

II. LITERATURE REVIEW

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. In 2002, it was performed as a document level classification task. Thereafter, in 2004, it was handled as a sentence level task. Furthermore, it was developed at phrase level in 2009.

The paper **"Sentiment analysis for Movie Reviews"** by Ankit Goyal, and Amey Parulekar worked on the "Large Movie Review" dataset used by the AI department of Stanford University for associated publication. The dataset contained 50,000 training examples collected from IMDB where each movie review is labelled with rating on a scale of 1 to 10. They have categorized the ratings as either 1 (like) or 0 (dislike) based on the IMDB ratings. If the rating was above 5, it was deduced that the person liked the movie otherwise he did not. The Cross Validation was used in which the complete dataset was divided into multiple folds with different samples for training and validation each time and the final performance of the statistic of classifier was averaged over all results. The 3 methods used for feature extraction were Bag of words, N-gram modelling and TF-IDF modelling. They have tried multiple classification models on various feature representations of the textual information in the reviews. Out of all the models, the logistic regression model showed best performance with accuracy around 89 percent. The general order of performance was LogisticRegression, then NaïveBayes, followed by SGDClassifier, RandomForestClassifier and kNNClassifier.

Another paper **"Deep Learning for sentiment analysis on movie reviews"** by Hadi Pouransari and Saman Ghili used various natural language processing methods for sentiment analysis. They have considered 2 different datasets, one with binary labels and another with multi class labels. The main dataset used was that of IMDB for movie reviews. The first step performed was preprocessing of data for removal of stop words and punctuations. For binary labels, they have used Bag of Words and skip-gram word2vec for converting a cleaned sequence of words to numerical feature vectors. Thereafter, they applied classifiers like Random forest, Logistic regression and SVM for training the model. For the multi class, they implemented the recursive neural tensor networks (RNTN). In case of binary classifier, the results showed that using combination of Bag of Words and Random Forest classifier yielded highest accuracy of 0.84 in comparison to combination of others. Apart from this, in multi class, it was observed that training RNTN was computationally expensive. So, they used low-rank RNTN which achieved comparable accuracies to that of RNTN much faster.

III. PROPOSED MODEL

Firstly, the necessary libraries are imported. Thereafter, extraction of full sentences from dataset was carried out and were appended and stored in a dataframe. The stopwords and punctuations are removed from the data after that step. Lancaster stemmer is used for stemming of words. In the next step, dataset is splitted into training and testing data with 70:30 ratio i.e. 70 percent data for training and 30 percent for testing. The first five rows of dataset are shown in table I.Thereafter, TF-IDF is used to transform text into vector of numbers.

Additionally, the libraries needed for training the model are imported. Here libraries of Keras are used for implementation of CNN technique. The version of CNN used is CONV1D i.e. one dimensional convolutional neural network as the data to be used is one dimensional in nature.

Thereafter, different layers of CNN are defined namely input layer, pooling layer and output layer. Each has different functionalities. Input layer feeds the input to the network. Pooling layer reduces the number of parameters and computation in the network, controlling overfitting by progressively reducing the spatial size of the network. The output layer outputs the returns the output of the network.

The feed method is used for feeding the input to the network and processing it through several layers of network. Various activation functions can be used with CNN like Sigmoid, tanh, ReLU etc. Out of all these, ReLU is used here as it does not activate all the neurons at same time. It increases the computation efficiency by triggering few neurons at a time. A flatten layer is also defined which collapses the spatial dimensions of the input into the channel dimension.

Keras provides various optimizers such as SGD, Adam, Adadelta, etc. for increasing the efficiency of the output. Here Adadelta optimizer is used for optimizing the output as it gives better performance in comparison to others. A method model-loss is also defined which is used for calculation of L1 Loss and accuracy.

Thereafter, batch size is chosen which determines the number of batches in which the data will be

TABLE I: First five rows of dataset

	PhraseID	SentenceID	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage...	1
1	2	1	A series of escapades demonstrating the adage...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2

TABLE II: Comparison of activation functions

Activation function	F1 Score
ReLU	0.515
Softmax	0.512
Sigmoid	0.212

sent to the model for training the model. After that, number of epochs are chosen. One epoch is when an entire dataset is passed both forward and backward through the neural network only once. At the end of all the epochs, the L1 loss and accuracy of the model are calculated. Other factors, namely F1 score, precision and recall value are also calculated.

IV. Experimental Analysis

The main focus is on improving the accuracy of the model. There are various factors that contribute to accuracy such as value of batch size, number of convolutional layers, activation functions used, optimizers used, etc.

The number of epochs are kept constant throughout the process and value is set to 3. The batch size is taken as 64.

The effect of various activation functions on accuracy and different optimizers are shown in table II and table III respectively.

ReLU activation function provided the best results out of all the three activation functions tested.

Considering optimizers, all the optimizers tested yielded almost similar results but among all Adadelta provided the best value of accuracy.

TABLE III: Comparison of optimizers

Optimizer	F1 Score
SGD	0.495
Adam	0.491
Adadelta	0.561
Adgrad	0.500

```

for l in range(len(documents)):
  label = documents[l][1]
  tmpReview = []
  for w in documents[l][0]:
    newWord = w
    if remove_stopwords and (w in stopwords_en):
      continue
    if removePuncs and (w in punctuations):
      continue
    if useStemming:
      #if useStemming is set to True
      #Keep one stemmer commented out
      #newWord = porter.stem(newWord) #User porter stemmer
      newWord = lancaster.stem(newWord) #Use Lancaster stemmer
    if useLemma:
      newWord = wordnet_lemmatizer.lemmatize(newWord)
    tmpReview.append(newWord)
  documents[l] = (tmpReview, label)
  documents[l] = (' '.join(tmpReview), label)

print(documents[0])
```

Listing 1: Preprocessing of data

```

model = Sequential()
model.add(Conv1D(filters=64, kernel_size=3,
                 activation='relu',
                 input_shape=(2000,1)))
model.add(Conv1D(128, kernel_size=5, activation='relu'))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(rate = 0.50))
model.add(Flatten())
model.add(Dense(num_classes, activation='softmax'))
```

Listing 2: Adding Covolutional Layers in Keras

```

model.fit(X_train, Y_train,
          batch_size=64,
          epochs=3)
# _, accuracy = model.evaluate(X_test, Y_test, batch_size=batch_size, verbose=0)
score = model.evaluate(X_train, Y_train, verbose=0)
print('Train loss:', score[0])
print('Train accuracy:', score[1])
print('Precision:', score[3])
print('F1 measure:', score[2])
print('Recall:', score[4])
```

Listing 3: Training the model

```

from keras.models import load_model
model.save('model.h5')
```

```
model=load_model(’/content/model.h5’,
    custom_objects = {’f1’: f1,  ’precision_m’
    : precision_m, ’recall_m’ : recall_m})
```

Listing 4: Save and load model

```

score = model.evaluate(X_test, Y_test, verbose
    =0)
print(’Test loss:’, score[0])
print(’Test accuracy:’, score[1])
print(’Precision:’, score[3])
print(’F1 measure:’, score[2])
print(’Recall:’, score[4])
```

Listing 5: Testing the model

V. Conclusion

The aim of the assignment was to perform multiclass sentiment analysis for movie reviews using deep learning. The dataset of Rotten Tomatoes was used for training the model and testing its accuracy. CNN classifier was used to train the model and perform the multi class classification. The challenge was to aggregate word vectors into a single feature vector for each review. Two vectorizers, countvectorizer and TF-IDF were used out of which TF-IDF proved most promising. The activation functions tested were ReLU and Softmax. However, Softmax proved to be more efficient in comparison to ReLU. Several optimizers were used including SGD, Adam, Adadelta and Adagrad but Adadelta yielded the best value of F1 score.

References

[1] https://raw.githubusercontent.com/cacoderquan/Sentiment-Analysis-on-the-Rotten-Tomatoes-movie-review-dataset/master/train.tsv

[2] Ankit Goyal, Amey Parulekar. “Sentiment Analysis for Movie Reviews”.

[3] Hadi Pouransari, Saman Ghili. “Deep learning for sentiment analysis of movie reviews”.

[4] https://en.wikipedia.org/wiki/Sentiment-analysis

[5] https://keras.io/models/model/