Sentiment Analysis of Movie Reviews using a Deep Learning Convolutional Neural Network

Ahrim Han, Ph.D.
July 29 2019

Contents

Introduction

- Sentiment analysis (opinion mining)
  - Task of identifying and classifying the sentiment expressed in a piece of text as being positive or negative
  - Applications
    - Forecasting market movements based on sentiment expressed in news and blogs
    - Identifying customer satisfaction and dissatisfaction from their reviews and social media post
    - Basis for other applications like recommender systems

Problem Statement

- Past years of studies
  - Bag-of-words
    - A review text was converted to fixed-length vector using bag-of-words and these vectors were later used to train the classifier such as Naive Bayes or Support Vector Machine
      - Lack of consideration of semantic relationship between words
      - Data sparsity and high dimensionality
  - Tons of reviews and posts
- → Strong need for the more accurate and automated sentiment analysis technique

Goal

- In this project,
  - I will build deep learning models using various parameters to classify the positive and negative movie reviews using the high-edge deep learning techniques
  - I will compare models and observe the parameters affecting the performance in accuracy

Expected Beneficiaries

- Businesses can find consumer opinions and emotions about their products and services
- E-commerce companies (e.g.,Amazon) can identify fake reviews
  - For example, if majority reviews are positive, but the sentiment analysis determines that reviews should not be positive
    - → Should inspect the reviews manually if those are fake or not
  - Fake reviews are not only damaging both competing companies and customers, but they also lead to reduced trust in companies and lower sales.
- Customers can check if the reviews are trustable before they make the decisions on buying products or services

Data Preprocessing

- Environment
- Data Preparation
- Data Encoding
- Cleaning Documents

Environment

- For this project, we used my own Linux machine having AMD Ryzen 7 2700X, 16GB Memory, Geforce RTX 2070.

- Keras with Tensorflow backend is used

Data Preparation and Data Encoding

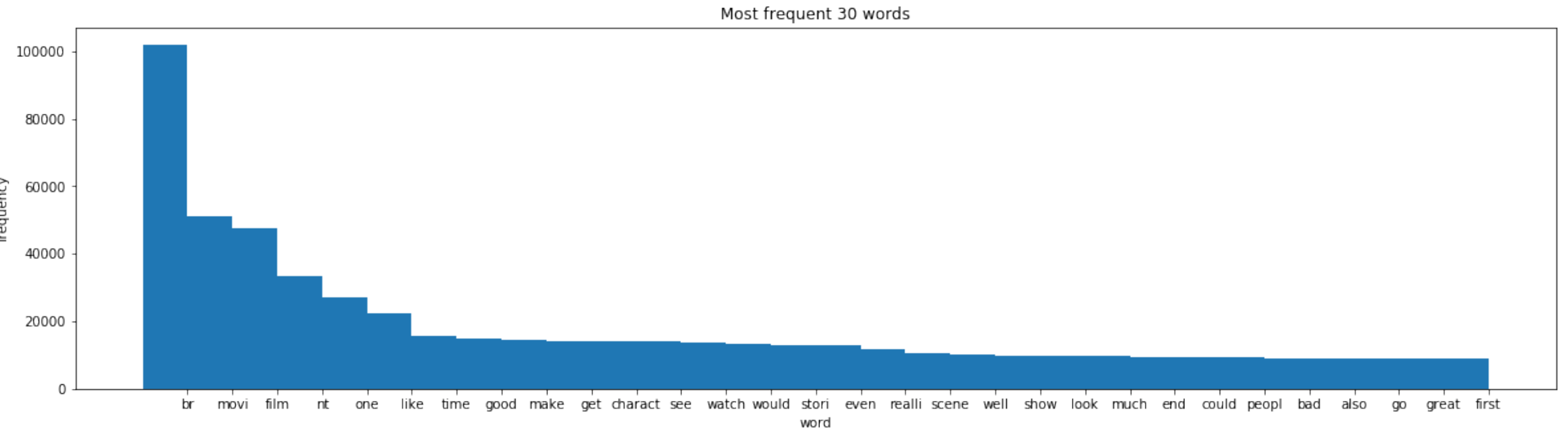
- A large dataset for binary sentiment classification of movie reviews
  - https://ai.stanford.edu/~amaas/data/sentiment/
  - 50,000 reviews (25,000 for training and 25,000 for testing)
    - The train and test sets contain a disjoint set of movies
    - Positive reviews: 7~10, Negative reviews: 1~4 (out of 10)
- Data Encoding
  - Encode the documents as sequence objects
  - Pad the sequences to the maximum length of the training dataset
  - ‘0’ for negative review, ‘1’ for positive review

Cleaning Documents

- Removing punctuations
- Removing Stopwords
  - NLTK packages
- Stemming
  - PorterStemmer in NLTK packages

Removing Non-Frequent words

- Building vocabulary dictionary
  - Count the words' occurrences on training dataset (52,826 words)
  - Filter out the words that have occurrences under 2 times (30,819 words)



- Removing non-frequent words that are not existing in the vocabulary dictionary

Building Deep Learning Models

- Basically the sequential model of Keras
- Main components
  - Embedding Layers with Word Embeddings
  - 1-Dimensional Convolutional Neural Network (CNN) with ReLu activation function and and MaxPooling
  - At the end, after flattening layer, fully connected Dense layers are added with the Sigmoid activation functions
  - Dropouts to avoid overfitting

Embedding Layers with Word Embeddings

- Word embedding
  - Way of representing text where each word in the vocabulary is represented by a real valued vector in a high-dimensional space
  - Options
    - Reusing the **pre-trained word embeddings**
      - 200 dimensional **Glove** embeddings of 400k words (dump of 2014 English Wikipedia)
    - Learning a **new word embedding from scratch**
      - Requires the large amount of text data to ensure that useful embeddings are learned
    - Hybrid: pre-trained embedding is used to seed the model, but the embedding is updated jointly during the training of the model
- Load embedding matrix into a Keras Embedding layer

1-Dimensional Convolutional Neural Network (CNN) and Pooling

- CNN proven to be successful at document classification problems
  - We used the **96 filters and a kernel size of 5 for a base model** with a rectified linear (‘**ReLU**’) **activation function**
  - This extracted set of features are then passed to **MaxPooling layer 10**, which filters the dominating terms from the extracted features

Training Deep Learning Models

- Model Configurations
- Results of Models
  - Effect of the Use of Word Embeddings
  - Effect of the Number of Convolutional Network Layers and Filters
  - Effect of the Number of Units in Dense Layer
  - Effect of Cleaning Documents
  - Effect of Dropouts
- Preventing Overfitting

Model Configurations

- Cleaning review documents or not
  - use_cleaned_docs: True or False
- Convolution Neural Network layers
  - number_of_additional_conv_layers: (0, 2)
  - number_of_filters: (96, 48, 24)
- Using pre-trained embedding GloVe word embedding
  - use_pre trained_embedding: True or False
- Training the embedding layer with training data set or freezing
  - trainable for embedding: True or False
- Using Dropout or not
  - use_dropout: 0.5 (50% neurons are randomly deactivated)
- Number of units in Dense layer
  - num_units: (128,64,32,8)

Callback function of “EarlyStopping” during Training

- One way to avoid overfitting it to terminate the process early
- EarlyStopping arguments
  - ‘monitor’= val acc (test accuracy) and ‘patience’=2.
    - The ‘patience’ indicates the number of epochs with no improvement after which training will be stopped

Results of Models

Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 0	FALSE	0	TRUE	FALSE	96	FALSE	128	86.62
Model 1	FALSE	0	FALSE	TRUE	96	FALSE	128	87.50
Model 2	FALSE	0	TRUE	TRUE	96	FALSE	128	88.82
Model 3	FALSE	2	TRUE	FALSE	96	FALSE	128	88.63
Model 4	FALSE	2	FALSE	TRUE	96	FALSE	128	89.11
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73
Model 6	FALSE	0	TRUE	TRUE	24	FALSE	32	88.39
Model 7	FALSE	0	TRUE	TRUE	24	TRUE	32	89.49
Model 8	FALSE	2	TRUE	TRUE	48	FALSE	128	89.62
Model 9	FALSE	2	TRUE	TRUE	24	FALSE	128	89.51
Model 10	FALSE	2	TRUE	TRUE	96	FALSE	32	89.42
Model 11	FALSE	2	TRUE	TRUE	96	FALSE	64	90.14
Model 12	FALSE	2	TRUE	TRUE	96	FALSE	8	89.20
Model 13	FALSE	2	TRUE	TRUE	24	FALSE	32	89.12
Model 14	FALSE	2	TRUE	TRUE	96	TRUE	128	89.18
Model 15	FALSE	2	TRUE	TRUE	48	TRUE	128	88.99
Model 16	TRUE	2	TRUE	TRUE	96	TRUE	128	86.98
Model 17	TRUE	2	TRUE	TRUE	96	FALSE	128	87.36

Table 1: Results of the accuracy of classification for the deep learning models and configurations.

→ The best accuracy is 90.14%.

Effect of the Use of Word Embeddings

Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 0	FALSE	0	TRUE	FALSE	96	FALSE	128	86.62
Model 1	FALSE	0	FALSE	TRUE	96	FALSE	128	87.50
Model 2	FALSE	0	TRUE	TRUE	96	FALSE	128	88.82
Model 3	FALSE	2	TRUE	FALSE	96	FALSE	128	88.63
Model 4	FALSE	2	FALSE	TRUE	96	FALSE	128	89.11
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73

Table 2: Accuracy results comparing of the use of word embeddings.

→ Models that use the pre-trained word embedding and updating weights during training provide the best performance.

Effect of the Number of Convolutional Network Layers

Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 6	FALSE	0	TRUE	TRUE	24	FALSE	32	88.39
Model 13	FALSE	2	TRUE	TRUE	24	FALSE	32	89.12
Model 2	FALSE	0	TRUE	TRUE	96	FALSE	128	88.82
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73
Model 0	FALSE	0	TRUE	FALSE	96	FALSE	128	86.62
Model 3	FALSE	2	TRUE	FALSE	96	FALSE	128	88.63
Model 1	FALSE	0	FALSE	TRUE	96	FALSE	128	87.50
Model 4	FALSE	2	FALSE	TRUE	96	FALSE	128	89.11

Table 3: Number of CNN layers (0,2): Results of the accuracy.

→ The model with the 3 layers (additional 2 layers) works better than the model with only one layer.

Effect of the Number of Filters in CNN

Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73
Model 8	FALSE	2	TRUE	TRUE	48	FALSE	128	89.62
Model 9	FALSE	2	TRUE	TRUE	24	FALSE	128	89.51

Table 4: Number of filters (98, 48, 24) in CNN: Results of the accuracy.

→ The largest number of filters of 96 in convolutional neural network gives best performance.
But it does not guarantee that the larger number of filters always produces the better performance.

Effect of the Number of Units in Dense Layer

Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73
Model 10	FALSE	2	TRUE	TRUE	96	FALSE	32	89.42
Model 11	FALSE	2	TRUE	TRUE	96	FALSE	64	90.14
Model 12	FALSE	2	TRUE	TRUE	96	FALSE	8	89.20

Table 5: Number of units (128, 64, 32, 8) in Dense Layer: Results of the accuracy.

→ The model with the 64 number of units in the Dense layer has the highest accuracy.
Exception of that model, the larger number of units in the Dense layer has, the better performance.

Effect of Cleaning Documents

Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73
Model 17	TRUE	2	TRUE	TRUE	96	FALSE	128	87.36
Model 14	FALSE	2	TRUE	TRUE	96	TRUE	128	89.18
Model 16	TRUE	2	TRUE	TRUE	96	TRUE	128	86.98

Table 6: Cleaning documents: Results of the accuracy.

→ The cleaning degrades or does not improve the accuracy.

Effect of Dropout

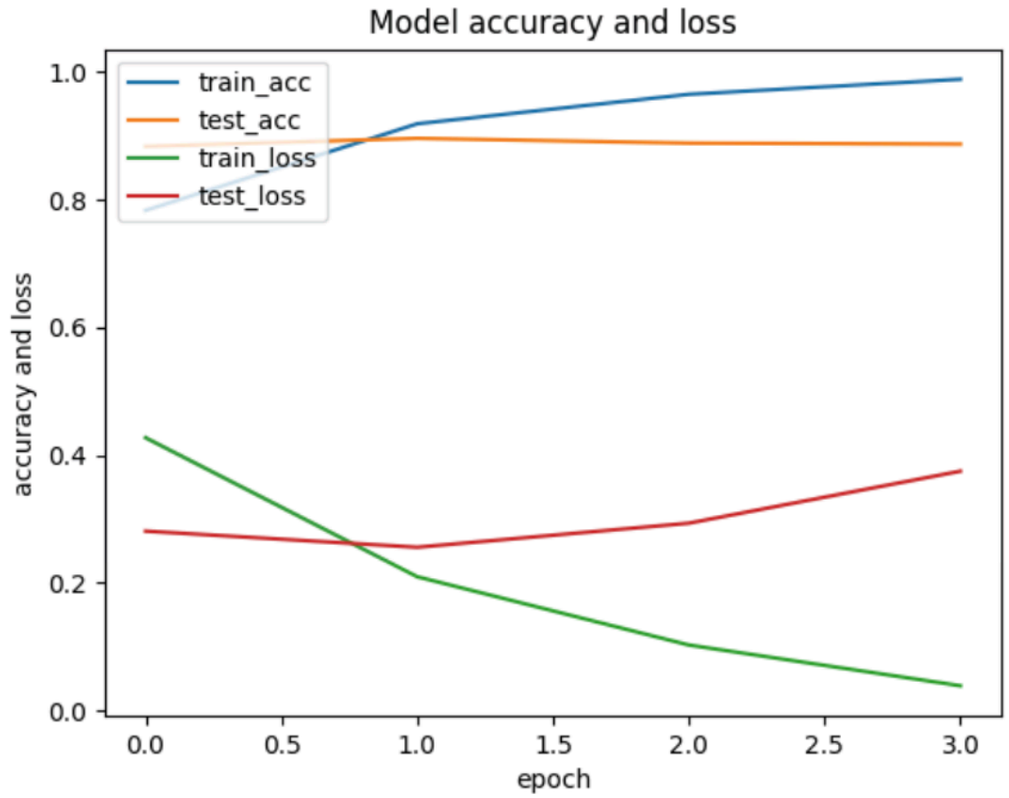
Model #	clean_docs	# conv.	pre-trained	trainable	# filters	dropout	# units	Accuracy
Model 8	FALSE	2	TRUE	TRUE	48	FALSE	128	89.62
Model 15	FALSE	2	TRUE	TRUE	48	TRUE	128	88.99
Model 5	FALSE	2	TRUE	TRUE	96	FALSE	128	89.73
Model 14	FALSE	2	TRUE	TRUE	96	TRUE	128	89.18
Model 17	TRUE	2	TRUE	TRUE	96	FALSE	128	87.36
Model 16	TRUE	2	TRUE	TRUE	96	TRUE	128	86.98
Model 6	FALSE	0	TRUE	TRUE	24	FALSE	32	88.39
Model 7	FALSE	0	TRUE	TRUE	24	TRUE	32	89.49

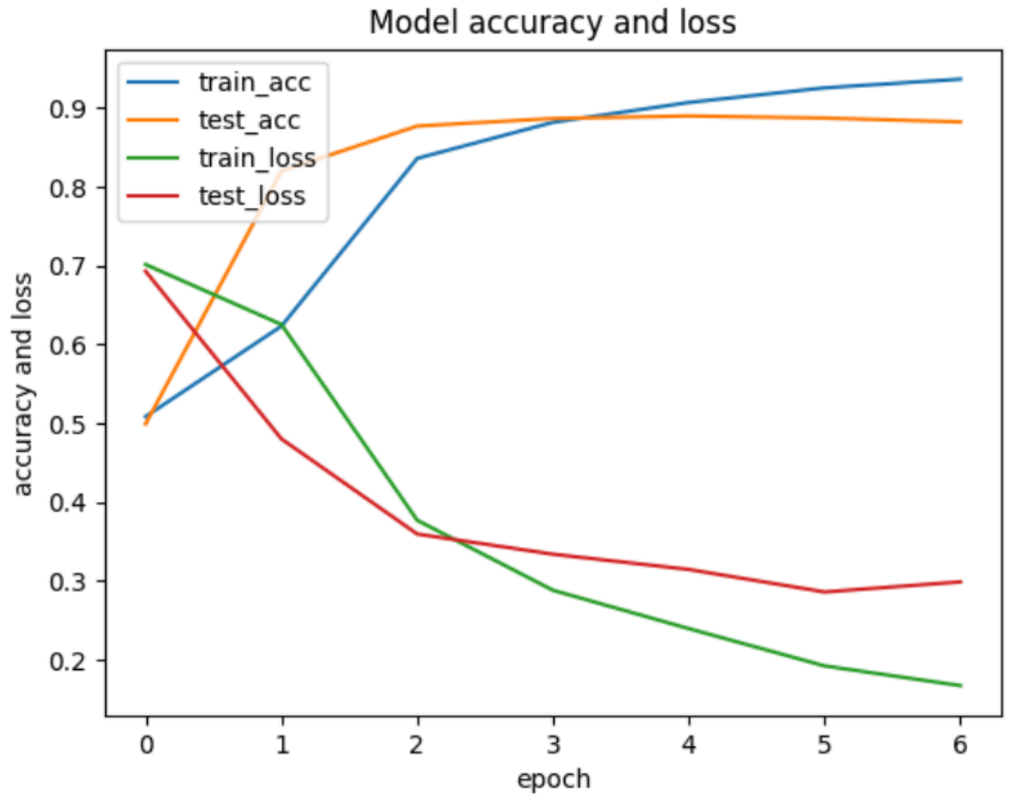
Table 7: Dropouts: Results of the accuracy.

→ Dropouts lead accuracy improvement only when the network size is small.
If the network size is large, too much data is missing so the dropouts may not be effective.

Preventing Overfitting

- No Dropout (Model 8)
  - Test (validation) loss starts to increase as from epoch 1, while training loss continues to lower as fitting to training data
- Adding Dropout (Model 15)
  - Test loss increases much later (i.e., epoch 5)
  - Test loss increases slower





(a) Model 8 (Dropout=False) to Model 15 (Dropout = True).

Limitation

- To improve the performance of accuracy, we can use the other techniques
  - 1. Get more training data.
  - 2. Add weight regularization.
  - 3. Data-augmentation.
  - 4. Batch normalization

Discussion

- In the previous study*, the performance of machine learning models are in range of 67.42% to 88.89%.

- In this project, the best model reaches 90.14% classification accuracy on the test (validation) set at the first epoch.

- We could probably **get to an even higher accuracy by 1) training longer with more overfitting preventing methods (such as dropout or regularization) or by 2) fine-tuning the Embedding layer.**

- However, this is the preliminary project for the classification of reviews using deep learning techniques. We think this accuracy is enough to observe the advantages of using deep learning models on NLP problems.

*** "Learning Word Vectors for Sentiment Analysis.", Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).**

Conclusion

- In this project, we generate the deep learning models based on CNN
  - The best accuracy of our model is 90.14%.
  - For the Word Embeddings, models that use the pre-trained word embedding and updating weights during training provide the best performance.
  - For the CNN and filters, the larger number of filters always produces the better performance.
  - The model with the 64 number of units in the Dense layer has the highest accuracy. Exception of that model, the larger number of units in the Dense layer has, the better performance.
  - For the use of the cleaned documents, it seems that the cleaning degrades or does not improve the accuracy in our project.

Thank You.