Sentiment Analysis of Movie Reviews

NLP Final Project

Deepa Borkar

Due: August 10, 2022

Contents

1 Abstract

There are 3 primary approaches to sentiment analysis: machine learning classifiers, semantic orientation of extractive relevant k-grams of text and labeling them either positive or negative, and using SentiWordNet. For this project, I explored using deep learning techniques for the NLP task of sentiment analysis on 50,000 IMDB movie reviews. In the final results, the accuracies for BOW, LSTM, and GRU classifiers are 0.866, 0.860, 0.853 respectively.

2 Related Work

This is a summary of the research papers reviewed before starting this project as part of the original project proposal.

2.1 Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews

This paper explores common deep learning techniques used for sentiment analysis. For this type of NLP task, the most common deep learning techniques used are CNNs and LSTMs. In this paper, sentiment analysis is done on IMDB movie reviews and the authors compare CNNs, LSTMs, and a combination LSTM-CNN model. From the results, the accuracies for CNN, LSTM, and LSTM-CNN are 0.90, 0.88, and 0.89 respectively.

2.2 Sentiment Analysis of Movie Reviews using Machine Learning Techniques

In this paper, the authors explore sentiment analysis of movie reviews using 3 machine learning techniques: Naïve Bayes, K-Nearest Neighbors, and Random Forests. The dataset for the study is composed of 2,000 movie reviews from IMDB, and 1,000 are classified with a negative sentiment while the other 1,000 reviews are classified with a positive sentiment. The researchers use a software tool called WEKA (Waikato Environment for Knowledge Analysis), which is written in Java and helps with data processing and implementation of machine learning algorithms. In order to validate their findings, the researchers have used 10-fold cross validation. From their results, they have determined that the algorithm with the highest accuracy is Naïve Bayes. The accuracies for Naïve Bayes, K-Nearest Neighbors, and Random Forest are 81.4%, 55.30%, 78.65% respectively.

2.3 Sentiment Analysis of Movie Reviews and Blog Posts

This paper explores the third approach of using SentiWordNet in detail. The researchers look at adjectives and adverbs in movie reviews and blog posts to determine the sentiment of the text. There are 4 different approaches discussed:

using just adjectives, using both adjectives and adverbs, variable scoring for adverbs and adjectives, and using a scaling factor for adverbs in comparison to adjectives. The main ways that the work is evaluated is through the accuracy, f-measure, and entropy. The researchers also compare their SentiWordNet approaches with ML techniques Naïve Bayes and SVM in the results. In the results, the SentiWordNet approaches all have accuracy scores in the 60% range while the Naïve Bayes and SVM techniques have accuracies in the high 70-80% range.

2.4 Applications of Deep Learning to Sentiment Analysis of Movie Reviews

This paper reviews the results and differences of using deep learning techniques, like Recurrent, Recursive, and Convolutional Neural Networks, with the baseline approach of Naïve Bayes to conduct sentiment analysis of movie reviews. The dataset is composed from the Stanford Sentiment Treebank. From the results and the conclusion of the paper, the researchers were able to show that the Recurrent Neural Network model was not able to represent the contextual complexities of the sentences and was not an effective model. In comparison, the Recursive and Convolutional Neural Network models had accuracy results that were similar to the accuracy results of the baseline Naïve Bayes model. When adding in the word2vec model as input for the CNN, the accuracy is even higher for CNNs than that of the baseline Bayes model. The study also demonstrated that while the Naïve Bayes model may be good at predicting polar classes like positive and negative, the Bayes model is not as consistent as the deep learning models when predicting classes on a more granular level (i.e. strongly positive, positive, strongly negative, etc).

3 Introduction

Sentiment analysis is a very common NLP task. There are many papers exploring different techniques to complete this task. I wanted to explore some of these models to get a better understanding of some of the models within deep learning that are used for NLP tasks. So, for this project, I set out to complete a sentiment analysis on movie reviews using deep learning techniques.

4 Data Processing

For this project, I used a dataset from the Hugging Face library. The Hugging Face library has an IMDB dataset. This dataset consists of 25,000 training examples and 25,000 test examples. Each example consists of a full text review with a maximum number of words of 2,752 and a minimum number of words of 11. Each example also consists of a label of negative or positive that classifies the text review.

Since this is a pretty clean dataset, not much data processing was needed. There were also an equal number of negative and positive examples, so I will be using accuracy to evaluate the performance of the models used in this project.

Because the training dataset consists of 12,500 negative examples followed by 12,500 positive examples, I first started by shuffling the dataset to add more randomization to the training. I also used the nlkt library to remove stopwords (i.e. an, the) from the original text review before tokenizing the review.

I split the training set into a training set and validation set for training purposes. The validation set is 25% of the original training dataset and the training data is the rest. With the tokens from the training data, I created a vocab that consisted of all the words in the data that had a minimum frequency of 50. This vocab had 5,705 tokens that I then used to modify the data for input for the models.

5 Models

The models that I was able to explore for this project were a simple BOW Classifier and two RNN models (LSTM and GRU). I have provided more details in the following sections on the specifics of training these models.

5.1 BOW

As a base model, I started by just classifying the reviews with a Bag of Words model. In general, because sentiment analysis is a more simple NLP task, the BOW model is expected to work well for this task.

For this model, the tokens for each review are transformed into BOW vectors. I then used this as the input to train the model in batches. I used a batch size of 500 examples. The input dimensions were the size of the vocab and the output dimensions were 2 because there are only 2 labels for this task. I used a learning rate of 5e-3. I passed the BOW vectors through a linear layer and used Cross Entropy Loss to calculate the loss of each epoch. Using the Adam optimizer, I was able to train a BOW model on this dataset with a test accuracy of 0.866.

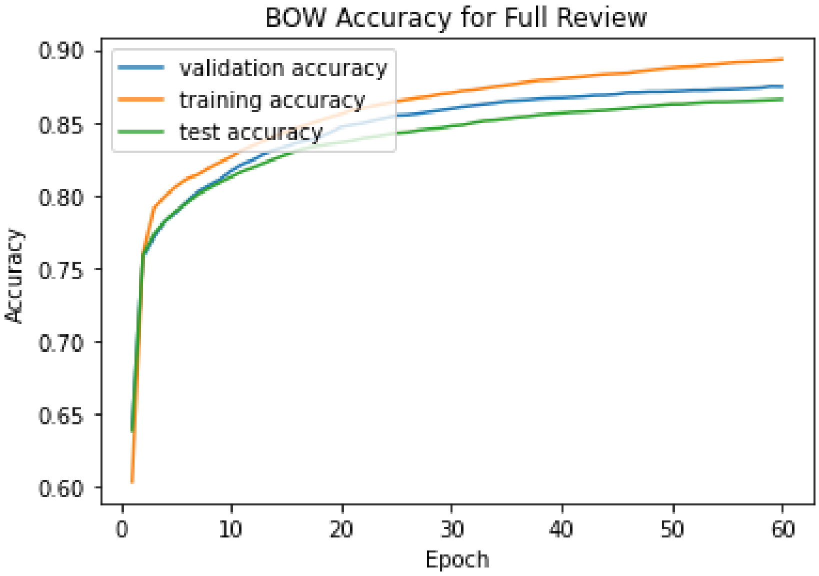

Figure 1: Accuracy for Training BOW Model

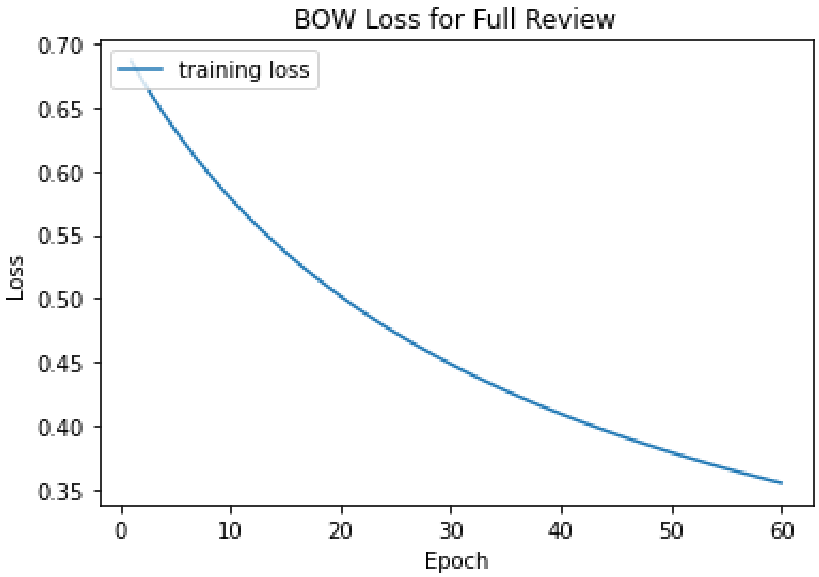

Figure 2: Loss for Training BOW Model

5.2 RNN

Next, I worked on training an RNN models. For these models, I transformed the tokens for each example to a numerical value. The numerical value is just the index of the token in the vocab dictionary, which unknown tokens mapped to the value 0 and padding mapped to the value 1. The padding comes into place when collating the datasets into batches.

I first tried training a simple RNN model but was unable to get a model that converges where the loss is minimized. Because I believe this might be an issue with how RNNs handle long sequences (due to the vanishing gradient problem) and many of the reviews were very long, I decided to try different RNN models that are better with long-range dependencies, like the LSTM and GRU models.

5.2.1 LSTM

Similar training techniques as the BOW Classifier were used for training the LSTM model. However, the LSTM model was composed of more layers. The input is put through the embedding layer, LSTM layer, dropout, and linear layer. After fine tuning, the parameters used for the model that was finally trained was embedding dimensions of 32, hidden dimensions of 128, output dimensions of 2, dropout parameter of 0.10, and learning rate of 5e-3. With the best model from training, which is the one with the lowest validation loss, the test accuracy for the LSTM model is 0.860.

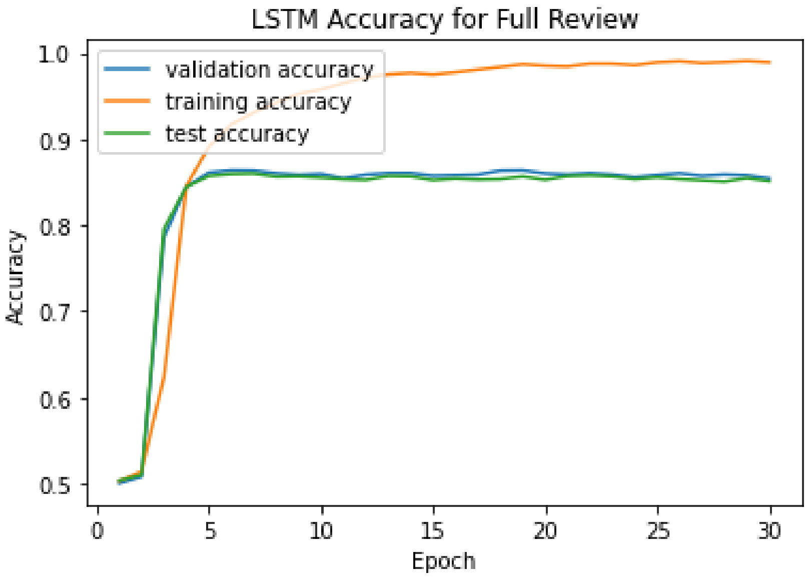

Figure 3: Accuracy for Training LSTM Model

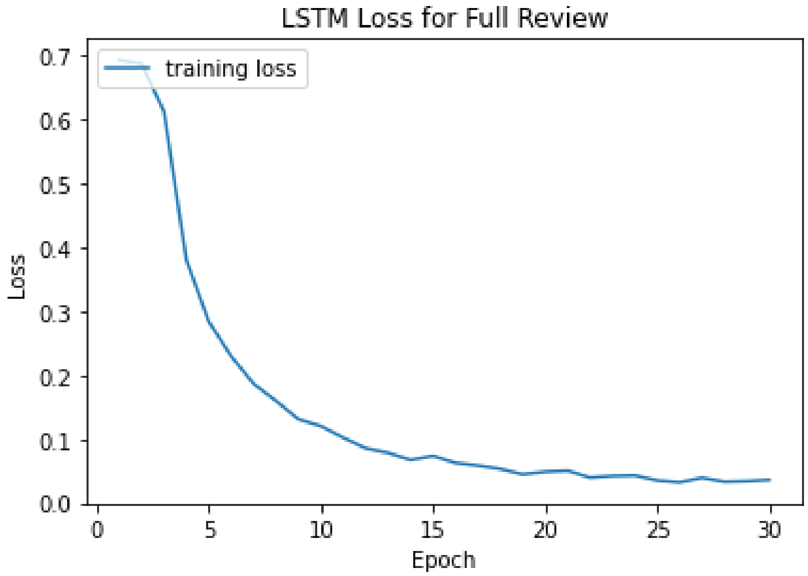

Figure 4: Loss for Training LSTM Model

5.2.2 GRU

Like the LSTM model, the layers for this model are very similar. The input is put through the embedding layer, GRU layer, dropout, and linear layer. After fine tuning, the parameters used for the model that was finally trained was embedding dimensions of 32, hidden dimensions of 128, output dimensions of 2, dropout parameter of 0.25, and learning rate of 5e-3. With the best model from training, the test accuracy for the GRU model is 0.853.

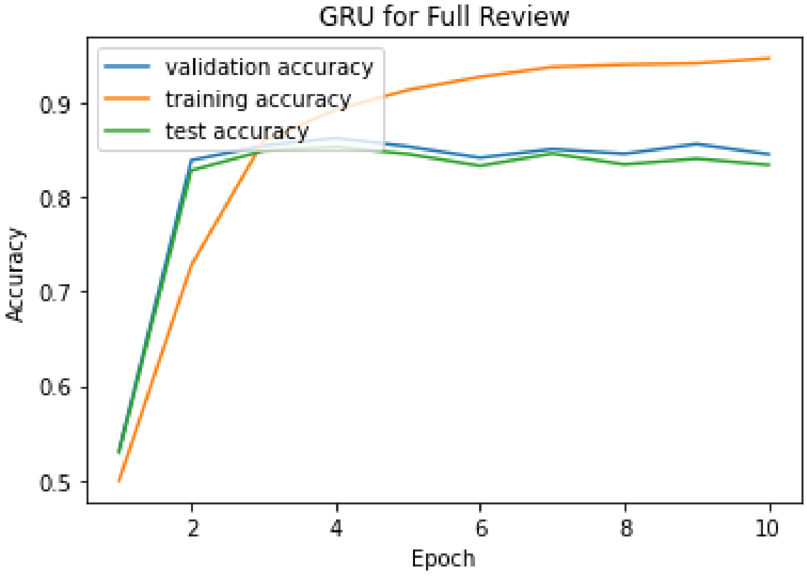

Figure 5: Accuracy for Training GRU Model

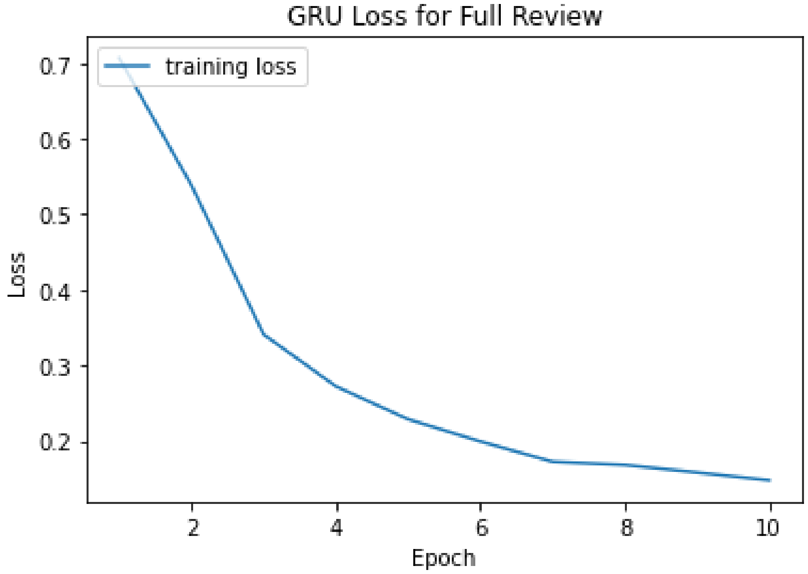

Figure 6: Loss for Training GRU Model

6 Results

In the Hugging Face dataset, there were 25,000 test examples. Using the best models from training with BOW, LSTM, and GRU, the accuracies for the test dataset with 25,000 examples are shown in the following table.

Test Accuracy		
BOW	LSTM	GRU
0.866	0.860	0.853

For comparison and to evaluate my results, I can compare these accuracies to the article mentioned in section 2.1 because this research paper also looked at IMDB reviews specifically. The LSTM accuracy found in this related work is 0.88. This is very close to the LSTM accuracy I achieved of 0.860, so this helps to further validate my findings.

7 Pitfalls and Lessons Learned

The main pitfall that I struggled with was training just a simple RNN model with the dataset. I spent a large amount of time trying to tweak the parameters of the RNN to try to get the model to converge during training. However, it was very difficult to figure out the correct parameters. After doing more research, I found that simple RNNs are very difficult to train for the sentiment analysis task because of the vanishing gradient problem and they are usually not good with long sequence data. Looking back at my project time, I wish I had spent less time trying to train an RNN with this particular dataset for this task. It was much easier to train RNNs with gated architectures, like the GRU and LSTM models.

Although I was able to train the GRU and LSTM models, these models still did not do as well as the simple BOW classifier. Because sentiment analysis is a relatively simple NLP task, it seems that BOW classifiers are sufficient enough for this task.

8 Future Work

I would have liked to test out some more models with this dataset. Specifically, I wanted to try out a CNN model with the data to get better familiarity with how this model would work with this dataset.

I also wanted to explore BERT and Transformers in more detail. I did get to train a Transformers model with the Hugging Face libraries. I used a pre-trained model and achieved a 0.92316 accuracy for the test dataset after 3 epochs of training. I did not include this in the final results because I used a pre-trained model with the help of a tutorial and I wanted to spend more time working on training these types of models and fine tuning these models.

Another interesting topic to explore would be just looking at the last sentence of the review instead of the full review. I did try to train a BOW model with just the last sentence but I was not able to get a model to converge with the data. It would be interesting to continue working on this to see if a model could be trained with just the last sentence of the review because this would be significantly less data needed for training. This could also potentially reduce training time.

9 Conclusion

From the results of this project, there is confirmation that because sentiment analysis is a simple NLP task, a simple BOW classifier is sufficient for this type of task. In this project, the accuracies for BOW, LSTM, and GRU classifiers are 0.866, 0.860, 0.853 respectively. These accuracies show that a simple BOW classifier can actually do slightly better than the more complicated LSTM and GRU models.

10 Github Repo

Link to Github Repo: `https://github.com/deepapborkar/NLPFinalProject`

11 Bibliography

Baid, P., Gupta, A., Chaplot, N. (2017). Sentiment analysis of movie reviews using Machine Learning Techniques. International Journal of Computer Applications, 179(7), 45–49. https://doi.org/10.5120/ijca2017916005

Haque, Md. Rakibul, et al. “Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews .” Performance Analysis of Different Neural Networks for Sentiment Analysis on IMDb Movie Reviews, July 2020, https://www.researchgate.net/publication/343046458.

“Hugging Face – the AI Community Building the Future.” Hugging Face –, https://huggingface.co/.

Shirani Mehrh - Stanford University CS224D: Deep Learning for natural ... (n.d.). Retrieved June 27, 2022, from https://cs224d.stanford.edu/reports/Shirani-MehrH.pdf

Singh, V. K., Piryani, R., Uddin, A., Waila, P. (2013). Sentiment Analysis of movie reviews and blog posts. 2013 3rd IEEE International Advance Computing Conference (IACC). https://doi.org/10.1109/iadcc.2013.6514345