Assignment 8 - Language Modeling With An RNN

Executive Summary:

Our assignment is to evaluate deep learnings technologies and tools for language modeling (conversion of words and sentences into vectors and condensed vocabularies). We need to advise management on the best language modeling techniques and neural network model for classifying movie review sentiment correctly as positive or negative. We will explore various modeling techniques in a systematic 2x2 experimental design setting. In addition, we will evaluate various hyperparameter settings to determine our results and how to improve them.

Research and Design:

For this assignment we are using pre-trained word embeddings from the GloVe library. We created two versions of embeddings from the library. The first used the top 10,000 words, and the second used the top 30,000 words. The labeled data we used is comprised of 500 negative reviews and 500 positive reviews. The data is split into an 80% (Train) and 20% (Test) setup for modeling and testing purposes. Models were built for each language model using various RNNs, and classification accuracy scores for the training and test set were recorded.

Technical Overview:

We used the GloVe.6B.50d library to run our models and initially started with a dataset of 10,000 words and then increased it to 30,000 words to see the impact of training and testing accuracy scores with the most frequently used words likely to appear in movie review. The library is used to convert certain words into numerical vectors that symbolized its meaning, similarity to other words, and holds information about syntax.

We used Python's Keras API within the TensorFlow package for implementing RNN on the movie review dataset. For testing purposes, target scores and model accuracy were used to classify the movie review as either positive or negative. We established 35 epochs as our base in order for the model to learn as much as possible, given the limited number of training data and the quality of it. Our focus was to choose large enough epochs where training accuracy would hit its peak. We experimented with 3 different RNNs (SimpleRNN, GRU & LSTM) within the base startup code well also experimenting with activation type, learning rate, number of neurons, model learning rate, the number of layers, and dropout rate as our hyperparameter

settings modified to achieve the highest test accuracy possible. Model architecture and hyperparameter settings are extremely important in influencing classification accuracy as we have learned in past few assignments and therefore, we wanted to make sure this was fine-tuned to best settings prior to running on different language models. The final step was to run the best RNN on each of the different language models in an experimental design setting to see if we can further fine tune and improve model accuracy.

Findings:

In our model testing, we found that the best models were the simplest, 20 neurons, sigmoid activation, and one LSTM layer. We then used this model and tested whether using a larger vocabulary and more neurons would create a better model with higher test accuracy. Increasing neuron count from 20 to 200 always lowered accuracies. Vocabulary size was a little more unclear. While increasing the vocabulary size improved our model accuracy from 72% to 75% with the 20 neuron architecture, the testing accuracy decreased by from 69.5% to 66.5% using the 200 neuron architecture. This concludes that in our case, increasing neurons does not improve our model.

Conclusion and Recommendation:

To conclude, we suggest utilizing a larger data set since training on a small partial dataset is prone to overfitting. In addition, rating scores of the review typically fall on a spectrum or scale, so drawing a clear separation line between positive and negative reviews is an important consideration when talking about a large automated customer support system. Punctuation is also something that can be considered where yes words do tell a story, but they are emphasized based on punctuations which is not part of the language embeddings. Tone of voice is not something that can be brought in on an automated customer support, but as a data scientist there are alternatives such as punctuation and rating scales, that can be utilized more effectively in an automated customer service function to incorporate in customer feeling and sentiment.