Natural Language Processing Dataset

Ranaa Ansari, Daniel Arenson, Yining Feng, Han Nguyen

Northwestern University

MSDS-422 Assignment 8: Language Modeling with an RNN

Summary and problem definition for management

This assignment focuses on applying language modeling to a classification problem where we classify customer movie reviews as either positive or negative. The language models are developed with a variety of pre-trained word vectors and hyperparameters (e.g., word vector size, vocabulary size) in Python to study the effect on classification performance and results. Ultimately, the objective is to determine the most effective method for managing customer service functions, and advise management on the relevant system and requirements to automate a customer service system that is able to detect negative customer feelings.

Research Design

For this problem we used a set of two pre-trained word embeddings from GloVe, short for Global Vectors for Word Representation, developed by researchers at Stanford University, to transform the written language content into its numeric representation. These encodings allow us to transform a sequence of words, or a sentence, into a sequence of numeric vectors of which we can derive mathematical models.

For the actual training and test data to conduct our research, we will use a pre-selected sample of 1,000 textual movie reviews. Each of the reviews has either a positive or negative sentiment, a thumbs up or a thumbs down, associated with the content. We will concatenate these reviews together as a single series with the pre-defined sentiment in an associative data structure. This design will allow us to shuffle the data and execute a split, train and test cross-validation approach that is industry standard for the validation phase of our research.

Using this framework, we set up a Recurrent Neural Network (RNN) algorithm for training our classification system with pre-trained word embeddings from the GloVe database. The RNN model is given a randomized mixture of both positive and negative reviews from the training dataset, which is 80% of the overall reviews, in batch sizes of one-hundred reviews each iteration, for a total of fifty iterations.

Programming Work

The programming dataset of movie reviews was extracted from this week's module on Canvas. The file contains 1,000 text reviews, split into 500 deemed as positive reviews and 500 deemed negative reviews. GloVe.6b.50d, GloVe.6b.100d, GloVe.6b.300d and GloVe.Twitter.100d were the various vectors selected and loaded into an array for further use to implement in our RNN models.

The entire dataset was preprocessed to define the vocabulary size for the language model with the most frequently used words, programmed by setting up a dictionary() containing all items. The review data was then loaded into the notebook. The data was then stored in a list of

lists where each list represents a document and each document is a list of words. Each document was then converted into a numpy array to further review the positive and negative sentiment in sequenced format and appended into the 4 vector embeddings we selected earlier.

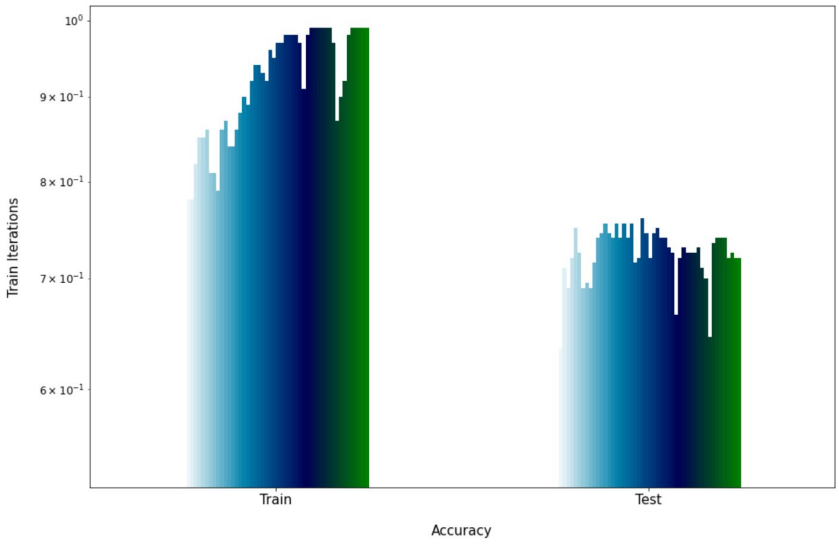
Each embedding was then split into train and test sets with scikitlearn's *train_test_split* and developed into 8 models using Tensorflow, each variation of neurons, epochs, layers, batch sizes, and further hyperparameters fine tuned to measure train and test accuracy of each model. To test two interventions within some of the same the models, we chose to run two different RNN's: (1) BasicRNN, using *tf.contrib.rnn.BasicRNNCell* and (2) Long Short-Term Memory RNN (LSTM), using *tf.contrib.rnn.BasicLSTMCell*.

Results and Recommendation

A total of 8 models were created to detect customer feelings. Models 1 to 6 are basic RNN models, and 7 & 8 are LSTM models. Models 1& 2 used *GloVe.6B.50d* word vector, models 3, 4, and 5 used *GloVe.6B.100d*, Model 6 used *Glove.Twitter.100d*, and models 7 & 8 used *GloVe.6B.300d* work vector.Based on the testing data, the most optimal model is 7. This model was a LSTM model with 20 neurons, 300 dimensions, and 50 epochs. Our recommendation for management is to use model 7 to identify negative customer feelings.

	Word Vector	RNN Model Type	Neurons	Dimensions	Epochs	TrainingAccuracy	TestingAccuracy
Model 1	GloVe.6B.50d	Basic	20	50	50	0.86	0.68
Model 2	GloVe.6B.50d	Basic	30	50	25	0.82	0.63
Model 3	GloVe.6B.100d	Basic	20	100	50	0.94	0.67
Model 4	GloVe.6B.100d	Basic	30	100	25	0.86	0.66
Model 5	GloVe.6B.100d	Basic	30	100	40	0.89	0.70
Model 6	GloVe.Twitter.100d	Basic	20	100	25	0.93	0.67
Model 7	GloVe.6B.300d	LSTM	20	300	50	0.88	0.76
Model 8	GloVe.6B.300d	LSTM	30	300	50	1.00	0.71

Model 7 Training & Testing Accuracy Visualization



References

Brownlee, J. (2019, August 07). Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras. Retrieved May 31, 2020, from https://machinelearningmastery.com /sequence-classification-lstm-recurrent-neural-networks-python-keras/

Brownlee, J. (2019, August 07). What Are Word Embeddings for Text? Retrieved May 31, 2020, from https://machinelearningmastery.com/what-are-word-embeddings/

Brownlee, J. (2019, August 14). How to Reshape Input Data for Long Short-Term Memory Networks in Keras. Retrieved May 31, 2020, from https://machinelearningmastery.com/ reshape-input-data-long-short-term-memory-networ ks-keras/

Helmini, S. (2019, March 06). All you need to know about RNNs. Retrieved May 31, 2020, from https://towardsdatascience.com/all-you-need-to-know-about-rnns-e514f0b00c7c

Li, S. (2018, June 04). A Beginner's Guide on Sentiment Analysis with RNN. Retrieved May 31, 2020, from https://towardsdatascience.com/a-beginners-guide-on-sentiment-analysis-with-rnn-9e100627c02e