Deep Learning for Natural Language Processing

Tweet Sentiment Analysis

Sneha Sriram Kannan CSE C 3rd Year 185001157

18th April 2021

Problem Statement

Process the training data consisting of various recent themes centered around current events. There are a series of sentiments given in the training data. Predict the sentiment of the test data.

Overview of the code

The original tweet and sentiment columns are extracted from the training data and stored. The data preprocessing consists of the initial cleaning phase which removes text such as website URLs, numbers, and converts all the text to lowercase. Then the output data is encoded using label encoding which assigns from 1 to 5 based on the class. For the ML models no further preprocessing is done. For the DL models further preprocessing is required. One hot encoding is done which converted the output to an array of size 5, which consists of 1 and 4 zeros, with the position of 1 indicating the class. For the RNN model, tokenizer is used to convert works to numbers and does the padding with leading zeros. For the RNN Glove model, a pretrained glove word embedding is used.

Methods Attempted

A total of 4 models were trained and tested. A description of each of them is given below:

1. Ensemble Learning

Ensemble learning makes use of multiple models combined. It makes use of decision trees and can be used for classification problems. Gradient boosting from sklearn is a powerful ensemble learning method and is the model used. The number of boosting stages chosen is 100 to prevent overfitting. The model did not perform well and gave an accuracy of only 0.43.

2. Support Vector Machine

SVM is another supervised learning method which can be used for classification. SVM constructs a hyperplane and creates the classification boundaries. RBF kernel is used in the model, and it gave an accuracy of 0.5

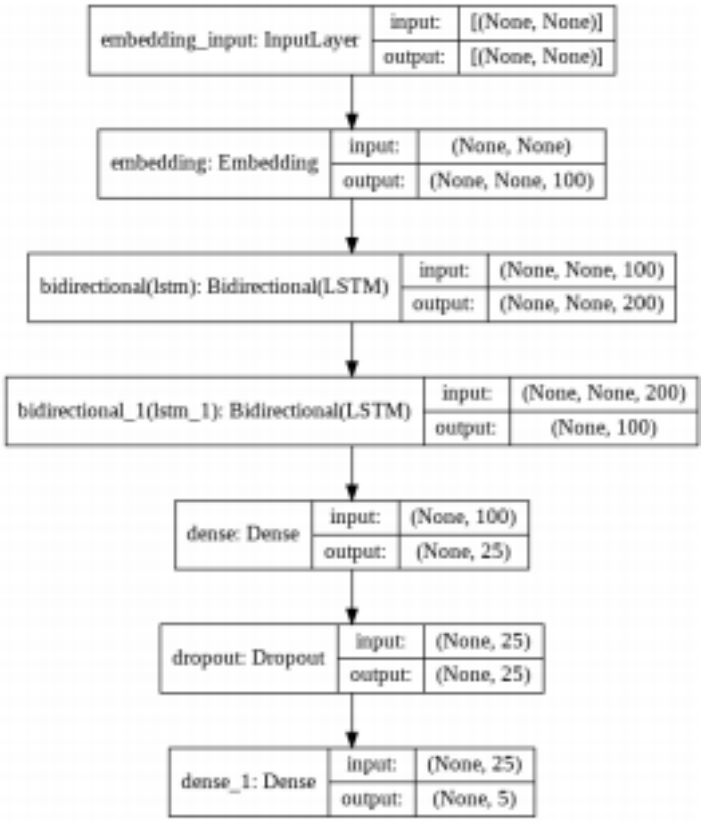
which was only slightly higher than ensemble learning. Therefore Deep Learning methods were tried.

3. RNN

Recurrent neural networks are especially useful for evaluating sequences in which the ordering of words is important. Bidirectional RNNs take both directions into account and not only the past so are even more powerful. LSTMs help overcome the problem of vanishing gradients. Therefore 2 bidirectional LSTM layers are included in the model. Dense layers are used to connect the LSTM outputs to the final output and dropout is used between the dense layers to prevent overfitting. Softmax is the activation function used in the last layer which produces a vector of probabilities. The loss function used is categorical cross entropy.

Early stopping is done to prevent the model from overfitting. Without early stopping, the training accuracy increases, but the validation accuracy decreases. Therefore with early stopping, once the validation accuracy stops increasing, the model stops training. Maximum validation accuracy of 0.6538 was achieved in 3 epochs.

Based on the testing of hyperparameters, this model was found to give the best accuracy. Different number of epochs, different number of nodes, layers, vocab size were tried and the final model used is shown below.



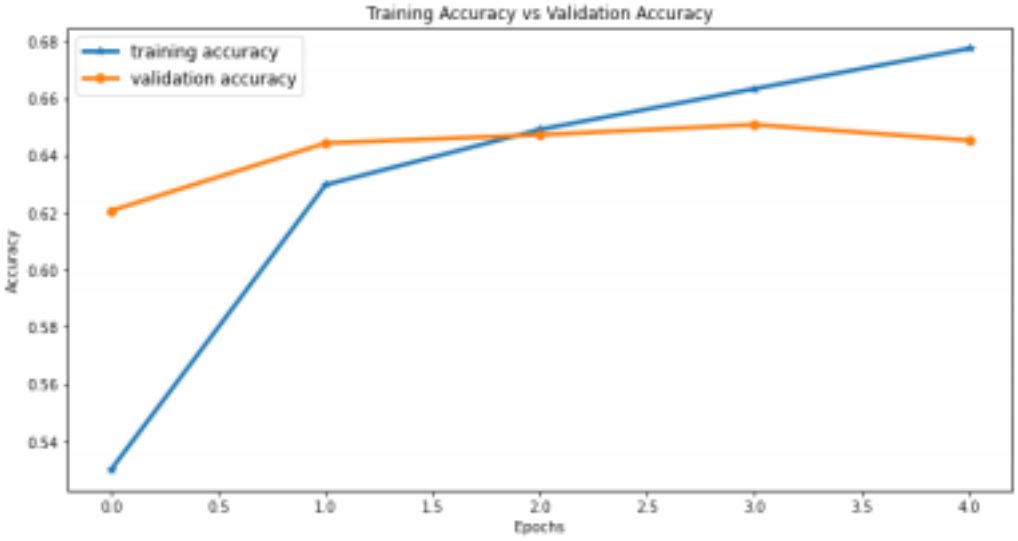
4. RNN with Glove

Glove is a pretrained embedding layer trained on a word2vec model and is really accurate in giving the relationship between words. Therefore, Glove was used to initialize the embedding layer weights to achieve better results. The resource used for Glove is: https://keras.io/examples/nlp/pretrained_word_embeddings/ . The glove used here is a 100 dimension variety. The remaining layers used are the same as the normal RNN above. Only the embedding layer is different. The model gave an accuracy of 0.683 which is better than the normal RNN model. This accuracy was achieved in just 3 epochs.

Choice of Parameters for each method

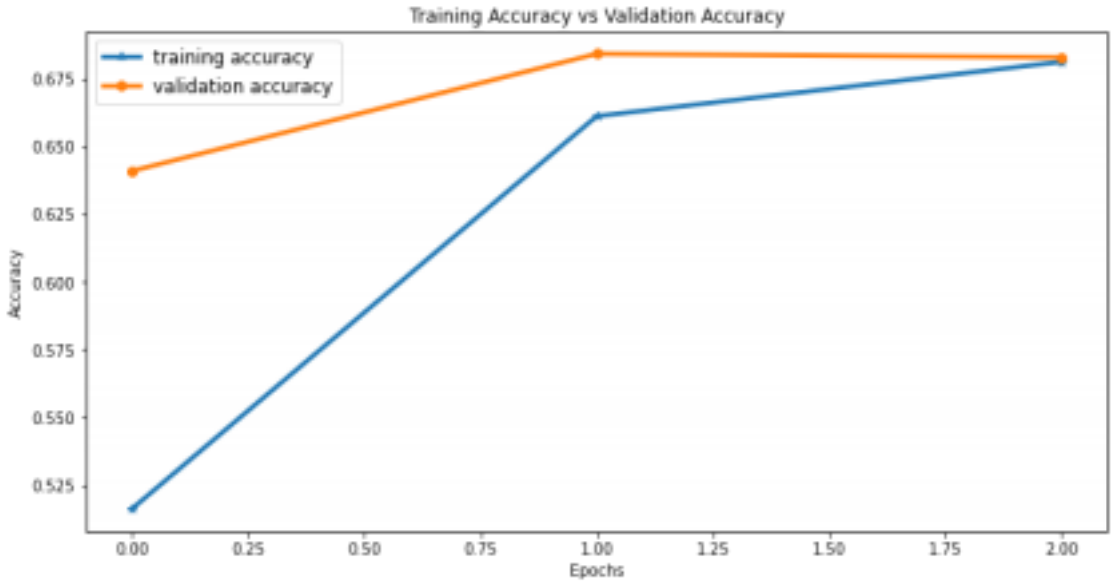
The plot of validation accuracy and loss against epochs is shown below:

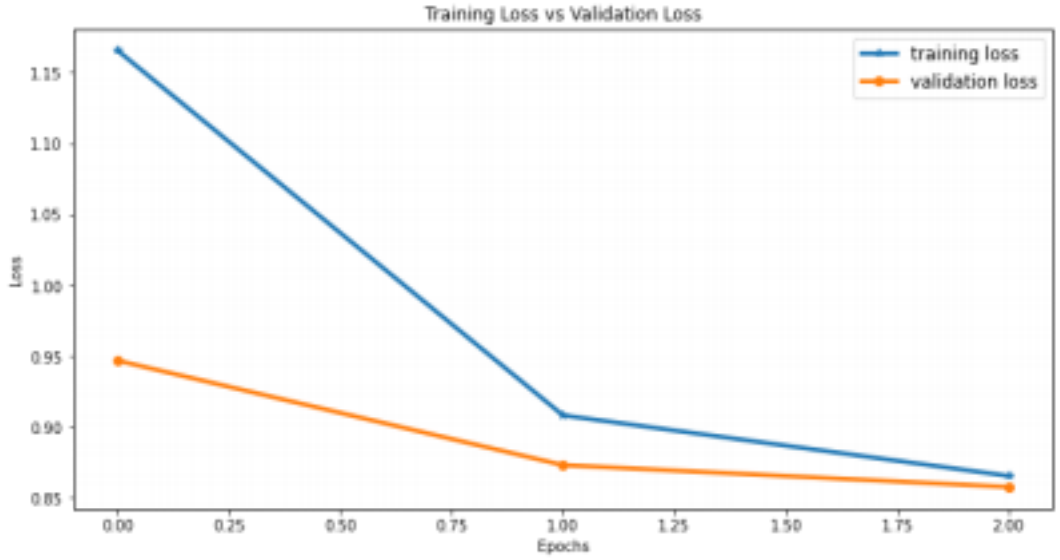
RNN (Due to early stopping, training stopped at 3 epochs)



The validation accuracy starts reducing after the 3rd epoch which is a sign of overfitting

RNN with Glove





Comparison of methods

On analyzing all the results, the RNN performed much better than the machine learning models used. The RNN with glove performed better than the normal RNN. The Machine learning models also took much longer to train than the RNN, but did not provide results.

Model	Ensemble SVM RNN RNN Glove
Time Taken to train in seconds	704 643 381 365

Highest Validation Accuracy	0.43 0.5 0.6508 0.6842

Therefore it can be concluded that the best model was the RNN with the pretrained glove word embedding.