Introduction

Text and document classification is an important part of every news aggregator. By grouping together similar articles, aggregators can provide customers with a more thorough way to research about relevant topics. I worked on creating a text and document classifier in my month long internship at Heckyl Technologies. I created two versions of the classifier - one that used support vector machines (SVMs) and another that used neural networks. The main objective of this project was to create two text classifiers - one that used SVMs and another that used neural networks - and to identify which of the two classifiers had higher accuracy.

Engineering

The classifier was written in Java. External libraries were used for the addition and usage of neural networks (neuroph) and support vector machines (libsvm). The basic premise for both the classifiers was based on the word frequency method. This method involves counting the number of times a specific word appears in a document and creating an n-dimensional map for the entire document (where n is the total number of distinct words in the entire database).

The one-vs-all method was used for the text classifier employing support vector machines. The method involves the creation of k-SVMs (where k is the number of classes). Each testing data point in run through each SVM to get a set of k probability values. To assign a label to the testing data point, the class with the highest probability is selected.

A sigmoidal neural network was used for the text classifier employing neural networks. A sigmoidal neuron takes in input(s) and returns an output with a value between 0 and 1. Input for the neural network was a word frequency chart similar to the one used in the text classifier using SVMs; however, the output for the neural network was a k-size (where k is the number of classes) array, where the index of the array with the value 1 was class assigned to the input data value.

Challenges

There were a number of challenges that I faced while working on this project. The biggest challenge was my lack of experience working with Java and Java APIs. I circumvented that challenge by spending a few days experimenting with Java and reading tutorials online. My experience with C++ made it easy to grasp the new language.

Another challenge was the lack of computing power to train a neural network. Since I was developing the project on my laptop, it didn't have enough power to train the neural network with tens of thousands of training examples.

Results

The objective of the project was partially achieved. I was able to create two text classifiers, one that used SVMs and one that used neural networks; however I was unable to compare them. The support vector machine text classifier that used the one-vs-all method was able to achieve 100% accuracy on the BBC Dataset and 90% accuracy on a subset of an internal Heckyl database. As mentioned earlier, the neural network text classifier was taking too long to train and hence, no numerical output could be determined.

Future Work and Application

Future work would include training the neural network and comparing its performance against the support vector machines.

I was also excited by the potential applications of the technology at the core of the text classifier. The algorithm and data structures being used for the text classifier can be used for a multitude of different applications. Applications include, but are not limited to, optical character recognizers, and spam classifiers.