Cornell Tech CS 5304 Assignments #4
May 5th, 2017
Due: 5 pm Eastern Time, May 19th, 2017

Assignments #4 is designed to be more like an independent project. You will have the option to select between 3 potential projects.

The solution should be built using Spark and the Spark libraries. Data set submissions should be compressed using zip. Ask the TAs in Slack for exact naming conventions.

Project Option 1: Deep Learning and Natural Language Processing

For this assignment, you may launch TensorFlow inside or outside of Spark. But the data processing to generate the TensorFlow input data should be done inside of Spark.

Large Movie Review Dataset
This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided. See the README file contained in the release for more details.

You will use the simple example using LSTM recurrent neural network to classify the IMDB sentiment data set. The starting point is:
https://github.com/tflearn/tflearn/blob/master/examples/nlp/lstm.py

The assignment:

- Part 1: Run the example program in TensorFlow and analyze the output
- Part 2: Pick a traditional non-deep learning machine learning algorithm available in Spark (Naïve Bayes, Random Forests, SVMs, logistic regression, etc.), setup the experiment, and analyze the output.
- Part 3: Using data uncovered from your analysis, explain which segments of the data benefit most from the use of either algorithm from Part 1 and Part 2.

Project Option 2: Creative Feature Engineering

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

The data set is at: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

The assignment:

Part 1: Take the raw data, propose an initial feature shaping approach, select a machine learning algorithm available in Spark (or TensorFlow), construct an experiment and analyze the output.
Part 2: Using analysis from Part 1, construct and justify 5 different "classes" of features engineering changes to the data. A "class" of a change is, for example, normalization. Normalizing all of the original numerical feature vectors would be one class of feature engineering change. (any operation that is repeated on multiple columns is a class)
Part 3: Using data uncovered from your analysis, explain which segments of the data benefit most from the use of feature engineering from Part 1 and Part 2.

Project Option 3: Performance of classifiers

Following Rich Caruana's ICML 2006 paper "An Empirical Comparison of Supervised Learning Algorithms" (https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf), we wish to compare performance of several different learning algorithms on a few data sets. The original paper selected only 5000 samples of training data to level the playing field. We would like to see how the most powerful algorithms perform over larger number of samples. To simplify, we focus on the best performing algorithms of the original paper: random forests, boosted trees, support vector machines and neural networks, plus LR and NB baselines, and restrict our attention to three small-to-medium size data sets from UCI Repository: Letter Recognition, Census and Poker Hand.

The data sets are at:
http://archive.ics.uci.edu/ml/datasets/Letter+Recognition
http://archive.ics.uci.edu/ml/datasets/Census+Income
http://archive.ics.uci.edu/ml/datasets/Poker+Hand

The assignment:

Part 1: Take the raw data, sample randomly to choose 5000 training, 1000 validation, and 1000 testing examples, choose a metric (Accuracy, F1, AUC, etc.), select three different machine learning algorithms available in Spark (or TensorFlow), and construct a baseline experiment. Report the results measuring performance.
Part 2: Build learning curves varying the training data set size from 5000 samples to 80% of the data set size.
Part 3: Discuss the differences in