Assignment 2 Report

Ace English

Problem Statement

For this problem, we were instructed to use neural networks to identify whether an incoming network packet was malicious or not. We were given a reasonably sized CSV full of network requests, which could be considered dirty. We needed to clean the data to make it usable to a neural network, and then teach it to determine whether incoming attacks are safe or malicious.

Methodology

To solve this problem I used Tensorflow, Jupyter, SKlearn, and Pandas libraries of Python.

The most important step was cleaning the data. First I deleted duplicates, of which there were many. Then I used value_counts() to examine the quantity of data and identify the presence of any significant outliers. I learned that the occurrence of errors was fairly low, and feared that the model may simply disregard the error data. To counteract this, I downsampled the “good” network connections, so they occurred roughly as often as the attacks.

I also dropped data columns that were mostly homogenous. Because of this, I rarely had any columns drop after normalization.

After that data was either normalized or one-hot encoded to be fed into a neural network. I took a 10% sample to work with to speed up operations. The output was one-hot encoded into two columns, for “good” and “attack”.

Experimental Results and Analysis

Before using a neural network I sent this data through a k-nearest neighbors algorithm and got very good results – consistent accuracy and recall of 1.0, and very good performance on random test cases. Fully-connected networks also worked very well, with a few exceptions of very bad combinations. **Sigmoid** and **tanh** models displayed a tendency to label everything as an attack. **Relu** was much more reliable.

The performance is so good I feared I was doing something wrong. I had difficulty getting a convolutional neural network to cooperate due to unfamiliarity with the library but eventually got one working. Its performance was similar to the performance of the fully-connected neural network, but operated more slowly. I would not recommend using a CNN for this problem.

In conclusion, I was able to get a model working that can reliably identify network attacks.

Task Division and Project Reflection

I, Ace English, handled this project entirely by myself. I tend to work better alone and am already involved in a group project with my senior project, so I am not eager to have others to depend on. I found the workload manageable but challenging by myself. I feel I understand the concepts at play but continue to have a very difficult time working with Python and Tensorflow. I had difficulty getting some lab demonstrations to compile. However, I got a model working despite these challenges, which is much better than I did last time.