

**CSC 180-01 Artificial Intelligence Mini-Project 2: Network Intrusion Detection System**

**Due at 2 pm, Friday , October 16, 2020**

**BY**

**Perry Gill**

**Rajvee Modi**

**Mary Ballesteros**

**Problem Statement**

Use google colaboratory to run our code on a cloud-based Jupyter notebook environment. Use CNN without transfer learning on colab and compare it to using CNN with transfer learning.

**Methodology**

In our Data Preprocessing, we discovered the large numbers of redundancy in our data. We cleaned our dataset by dropping the redundant values. This reduced our data by 348435. Next, we used encode\_text\_index to normalize our outcome.

**Experimental Results and Analysis**

We implemented a Sequential Model, and Logistic Regression Model in order to display important metrics used for comparison when implementing (CNN) Convolutional Neural Networks. For our CNN, we used Sequential as our model. We found that our activations: relu, and softmax gave us the best accuracy of 76%. We set our range to 5, and used Early Stopping to stop training once our model performance stopped improving. For our Logistic Regression Model (figure 2) we split our data into training and test and displayed our accuracy score, confusion matrix, classification report, actual values, predicted values, and ROC Curve. Figure 4, we implemented a Sequential Model. For our Sequential Model, we were able to output the three different layers: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. Our hyper parameters that gave us the best result were sigmoid and relu for our activation. We dropped the null and redundant values in our dataset. We reported FALSE as the number of unique rows, while TRUE had the repeated values (figure 3).

**CNN Model Comparison**

| Activation functions | Results: |
| --- | --- |
| Relu  Relu  Relu  Sigmoid  Sigmoid  Softmax | Test accuracy: 0.9968060851097107 |
| Sigmoid  Sigmoid  Sigmoid  Sigmoid  Sigmoid  Softmax | Test accuracy: 0.9954323768615723 |
| Relu  Sigmoid  Relu  Tanh  Sigmoid  Softmax | Test accuracy: 0.9951919913291931 |

**Task Division and Project Reflection**

**Task Division:**

Perry Gill:

1. Data Visualization
2. Strategize and Implementation
3. Implemented splitting of train and test
4. Data cleaning and preprocessing
5. Neural Network Models

Rajvee Modi:

1. Data Visualization
2. Implemented splitting of train and test
3. Implemented Logistic Regression model
4. Strategize and Implementation
5. Implementation of TensorFlow regression neural network model
6. Finalize approach

Mary Ballesteros:

1. Write the report
2. Data visualization
3. Strategize and Implementation
4. Neural Network Models

**Project Reflection :**

From this project, we got a lot of practice with using google colaboratory. We used google colaboratory to run our code on a cloud-based Jupyter notebook environment. We used CNN without transfer learning on colab and compared it to using CNN with transfer learning.

The challenge was to find the best accuracy

The challenge was to understand the dataset and recognize how to clean the data. We were having trouble trying to implement the right models in order to achieve a desirable accuracy. We were getting accuracy from “.67,” until we were able to achieve “.998” accuracy. Implementing CNN on text data was a challenge, but we resolved this issue through studying through the labs, and trying different hyper parameters. Trying to understand the activation and optimizer was another challenge as well. While implementing them through our model, we were able to play around with our CNN to see the different results.

From this project, we got a lot of practice with using google colaboratory. We used google colaboratory to run our code on a cloud-based Jupyter notebook environment. We used CNN without transfer learning on colab and compared it to using CNN with transfer learning.

The challenge was to find the best accuracy