**James Folk**

**DATA 527 – Predictive Modeling**

**Project 2**

**DEADLINE: May 6, 2024**

**Spring 2024**

**Overview**

The purpose of this exercise is to implement a feed forward neural network to predict the car market prices. The main goal is to work on data exploration and preparation beside the network model implementation. I am using a dataset that contains 3 columns with 5404 entries. I leveraged the Tensor SDK to implement the feed forward neural network.

Create the neural network using the Tensor SDK

Clean the dataset.

Determine which data is relevant

**Methodology**

* **Create the Neural Network:** The neural network architecture was established with a Keras sequential model, comprising an input layer, three hidden layers, and an output layer. Each hidden layer is composed of 20 neurons employing a Rectified Linear Unit (ReLU) activation function, chosen due to the positive nature of all inputs. The input layer accommodates the number of input neurons corresponding to the dimensions utilized in training the model. Conversely, the output layer consists of a single neuron, as the objective is solely to predict car prices based on their attributes.
* **Normalize the dataset:** I opted to apply min max scaling normalization to scale each entry, ensuring effective handling of any anomalies.

**Implementation**

A black background with a black square

Description automatically generated with medium confidence

**Results**

Learning Rate: 0.001

Number of Iterations: 500

Final Coorelation Coefficent: 0.7293365435980965

Structure:

3 input nodes

5 hidden layers with 40 nodes each

1 output node

A graph with blue lines

Description automatically generated

**Mean Square Errors**

**Discussion**

Challenges Faced and Solutions

* **Determining the Hidden Layer Structure:** I knew for sure that Car Name was unnecessary. I noticed that there was a unit of measurement on the prices, so I made sure to normalize that by scaling the number by the unit. I categorized the Fuel Type, Transmission, Ownership and Seats and labeled them with numbers. I thought that since there was so many engine sizes, it wouldn’t give a good measurement, so I was going to leave them out. I decided to train every combination of dimension between milage, fuel type, transmission, ownership, manufacture, engine, and seats.

**Conclusion**

It took several days to train the 127 different models, so I didn’t want to redo anything once I made my choices. I am curious if I can get a better correlation coefficient if I categorized the engine sizes. The combination that gave the best correlation coefficient between milage, fuel type, transmission, ownership, manufacture, engine, and seats were to omit the use of ownership and seats.

**References**

1. Pfeiffer, Simon. “Creating Your First Neural Network in Python w/ Tensorflow.” *DEV Community*, DEV Community, 14 Aug. 2021, dev.to/codesphere/creating-your-first-neural-network-in-python-w-tensorflow-4l5p.
2. Netron. (2024, April 28). https://netron.app/
3. Team, K. (n.d.). *Keras Documentation: Adam*. https://keras.io/api/optimizers/adam/