CS461 Project #3 Report

Michael Day

Github Repo: https://github.com/mumkymikey/cs461_evaluate_test_scores

For our final CS461 project, we were tasked with developing a regression neural network that predicts a student's test scores in Math, Reading, and Writing exams using various different variables. This neural network was developed in Python using Keras and Sklearn.

In order to begin developing a neural network for this, I had to first manipulate the given CSV of student data in a way that a neural network could understand. Besides the individual exam scores for each student, the given variables were mainly categorical so this data needed to be translated into a numerical format rather than categorical. Using label encoding and the fit_transform function from Sklearn, I was able to encode the categorical variables, such as "race/ethnicity" and "parental level of education", into distinct integer values that the neural network could distinguish between. Along with label encoding, I also implemented one-hot encoding for the "gender" field, as the given dataset only had designated values for male or female. This changed the "gender" variable to a single column, "Male", that either contained a 1 or 0 to distinguish between genders. With the dataset now properly encoded, I could begin developing the neural network.

The first thing to do with the encoded data was to split it into training, testing, and validation datasets to use against the network. I split this data using the train_test_split function from Sklearn. I also learned that datasets being used within neural networks likely needed to be scaled to a given range. I used the MinMaxScaler function to scale the datasets. Now that the datasets were prepared for the neural network, I initialized the Sequential model that would be the heart of my network. My neural network was made of three layers: the first layer was the

input layer with 5 defined neurons, then a single hidden layer that contained 10 neurons, and an output layer containing 3 neurons, one for each output variable. Each of these layers within the Sequential model utilized a rectified linear unit as its activation function. After the neural network model was defined, the model was configured for training using the Adam optimizer. Once the model was configured properly, training was performed on the model over 1000 epochs using a batch size of 64. The model also implemented EarlyStopping from Keras, which monitored the performance of the network and would stop the network from evaluating if performance staggered.

After training and validation, the neural network was able to guess a student's exam scores with roughly 90% accuracy. I found this value by keeping track of the mean absolute error of each epoch using the val_loss variable. Training and validation were done using the fit() and evaluate() functions from Keras, respectively. I found that the network was rather performant, though, I felt more confident in the training of the network over the validation step. I also thought my model was rather simple, using only three individual layers and having them all use the same activation function. I think that the accuracy of the model, along with the simplicity of it, was benefited by the use of 1000 epochs.