Neural Network Report

By Thomas Yang

The construction of the neural network model and cleaning the data file to determine if an employee is interested in changing jobs was an interesting and informational experience, and a confusing experience too. Cleaning the data file was a new and fun thing to do since this is the first time I had to do something like that. The first thing I did to clean the data file was to convert the columns to one hot encode data so it would be easier to feed into the model. This is only for columns that do not have numerical values and the data is strings with various answers. After that I dropped all the columns that were converted to one hot encode and also dropped any unnecessary columns. This was how I cleaned the data file to be fed into the model, and it was fun learning how to do that with python. The model I designed has two hidden layers, each layer with a hundred and twenty-eight neurons. The two hidden layers had their activation set to linear and the output layer had its activation set to SoftMax as stated in the requirements. The data was broken into three sets: training, validation, and testing. The training set was trained in batches of sixty-four, validation was done in batches of thirty-two, testing was done in batches of one hundred and twenty-eight. The model has an optimizer, loss function, and metrics that are used in the neural network. The optimizer implants different algorithm that a person can use in their neural network, the one I am using is Adam. The loss function is used to calculate loss in the neural network, the loss function I am using is Sparse Categorical Cross Entropy. I have tried a few other loss functions but some were incompatible with the shape or expected this but I have that, so it was a bit of trial and error getting the one I needed with the way my data was. I am sure that if I had more time, I could restructure my data to a way that it would take it in but unfortunately time is not on your side always. The last component is the metrics, which is used to calculate certain events, the one I am using is accuracy which calculates how often my predictions equal the labels. Now to talk about the outputs, the accuracy I got from my training and validation was between seventy-five to seventy-seven percent and loss was around forty-six to fifty percent. When it came to testing the data set it would have an accuracy of around seventy-six to seventy-seven percent and loss being at forty-five to forty-six percent, which is decent but I wish I could have gotten the loss to a lower percentage. There was definitely more that I wanted to do like dropping other columns that could not be needed to see how that affects the data or trying out more optimizer or loss functions to see how that affects the accuracy and loss of the neural network. I feel like I accomplished a lot while working on this, but I know I am just scratching the surface of this field.