

Evan Jenkins

Intro to AI CS461

Professor Hare

5 December 2021

Neural Network Report

My neural network was developed in Google Colab, published in GitHub, and employs TensorFlow's high level API for neural networks, Keras. The creation of the neural network appeared to be a daunting task from the outside, but with a good amount of time invested in research, it became much more manageable and understandable. I started my research and coding using TensorFlow's tutorial of creating a basic regression to predict fuel efficiency. From here, I was able to adapt much of the content to fit my own network such as creating clean data through one-hot coding, splitting the data into sets for training, testing, and validation, then splitting it further into features and labels, and beginning preparation for my model. The final configuration of my model consisted of 4 total layers. Of the 4 layers, the first is the normalizer layer with 17 neurons, the second and third, hidden layers with 64 neurons, and the final layer is the output layer with 3 neurons. Keras modeling does not offer support for the use of Gaussians in model generation, and thus, no Gaussians are used in my network. The addition of more layers impacted my mean absolute error dramatically so this is the best configuration I could find. By using the required 70/15/15 split, I avoided using K-fold cross-validation and resorted to using a very basic validation setup by comparing the error against how we are training it to generate a set of `val_loss` values that decrease over time as the network is trained. In the end, my neural network did not perform as well as I expected, clocking a mean absolute error value between 10 and 12 for each execution. One possible result of such a high error is the lack of use of Gaussians. While the regression can be calculated without Gaussians involved, they aid in smoothing when it comes to testing validation data and would have likely improved the error to a more desirable number.

References

https://www.tensorflow.org/tutorials/keras/regression#make_predictions

<https://machinelearningmastery.com/deep-learning-models-for-multi-output-regression/>