**Friday, September 29, 2023**

I finished organizing and handling the commai.ai dataset. Their provided code to handle batching and pre-processing was quite outdated, using Python 2, so I had to spend some time updating both the data processing and machine learning stages to reflect changes to the APIs from newer versions. I started training the model, but it is quite slow since it’s training on the CPU. I was able to fix some things with parallelization which made it slightly faster. However, I was only able to get through 1 epoch of training throughout the entire class period. While it was training, I started writing a sequence-based approach to training a model in which the input would be the last 5 images stacked together.

**Monday, October 2, 2023**

I continued working on the sequence-based model. The comma.ai dataset is too big to train on efficiently, and I can’t afford to wait for several days for each training session if I don’t already know it will work. I used TimeDistributed layers to distribute the convolution operation to each image in the sequence. However, I have to figure out a different method for loading the data – loading it into a giant array duplicated a lot of data – my dataset would be 134 gigabytes stored in this method. I wrote a basic generator subclassing Keras’ Sequence object which automatically gives the last 5 images from an array of the images themselves which I verified to work correctly.

**Wednesday, October 4, 2023**

The sequence-based model did not work. I am not sure why, but it predicted a steering value that was 0 for most images in my entire dataset. Given my work over the past 3 weeks, I conclude that this difficulty is why transfer learning has been abandoned in favor of autoencoder structures. It’s essentially impossible to debug or inspect the network I currently have further than the layers and what they are supposed to do. Here is the graph of predicted value vs. actual: A blue and orange lines

Description automatically generated

This is much *worse* than the graph from Journal #4 despite using more data and being twice as slow to process each frame (about 48 ms/frame = 21 FPS). Thus, I tried to download the simulator most commonly used for scale-model autonomous car research. However, my computer’s boot disk doesn’t have enough space. I think the TJ cluster can be used provided I find a way to get GUI output, but by default there’s no way to start an X server without sudo permissions.

**Thursday, October 5, 2023**

While the model was training yesterday, I was experimenting with a data logging phone app. It’s vital on the actual car that I have accurate values for the car’s position, acceleration, and angular velocity, and a phone already measures all of that. However, the app POSTs data to an HTTP server running on another computer. I noticed that the FCPS network blocked this connection, meaning I could not stream data from my phone to my laptop. However, at home it did end up working. This is a problem because it mean I can’t remotely access the Nvidia board through the school Wi-Fi either. It might be necessary to find a wireless router that the car and host computer connect to.