Julian Singkham
CS 2300/11
Dr. Berisha

Lab 8: Self-Driving Car DNN

Abstract

The purpose of this lab is to familiarize ourselves with training a Deep Neural Network for a self-driving car and evaluating the effects of data augmentation and parameters of model accuracy.

Questions

1. What operation(s) take the most time to train the model?

Increasing the epoch size by far takes more training time than batch size. This is due to how batch size deals with slicing the dataset and epoch is how many iterations over the dataset.

2. Is the model training time proportional to the amount of data collected? Support this answer with evidence (a table or plot would be ideal).

Table 1 details the timings of three datasets: 37, 1144, 3047.

	Small	Medium	Large
Epoch 1	2	30	27
Epoch 2	0	26	24
Epoch 3	0	26	24
Epoch 4	0	26	24
Epoch 5	0	26	24

Table 1: Dataset Timings

Training time is proportional with the amount of data collected, but because of parallelization there has to be a vast difference (in magnitudes of 10x) in order to see a difference in timings. The medium and Large data set differ by a factor of 3, yet the medium dataset took longer than the large dataset. This could be due to how the DNN algorithm works and the amount of data slicing is the same between the medium and large dataset.

3. With a larger dataset size, did you see a proportional reduction in the loss?

Comparing the small data set used in Figures 6&7 with the large dataset used in Figure 8, one can see how vastly more accurate, by a factor of 2, the larger dataset is.

4. Did you see overfitting in any of your models? What did you do to reduce it, and was your approach effective?

Overfitting occurred in Figures 4-6 and was primarily caused by having too large of an epoch and/or batch size for the dataset used. Lowering the batch size proved to reduce overfitting as can be seen in Figure 6 where the batch size was set to 1 since the dataset only contained 37 entries. Comparing Figures 5 and 6, one can see how overfitting was delayed by 4 epochs and accuracy increased overall by ~0.10.

Julian Singkham
CS 2300/11
Dr. Berisha

Figure 1 displays the model using a batch size of 32 and epoch of 5.

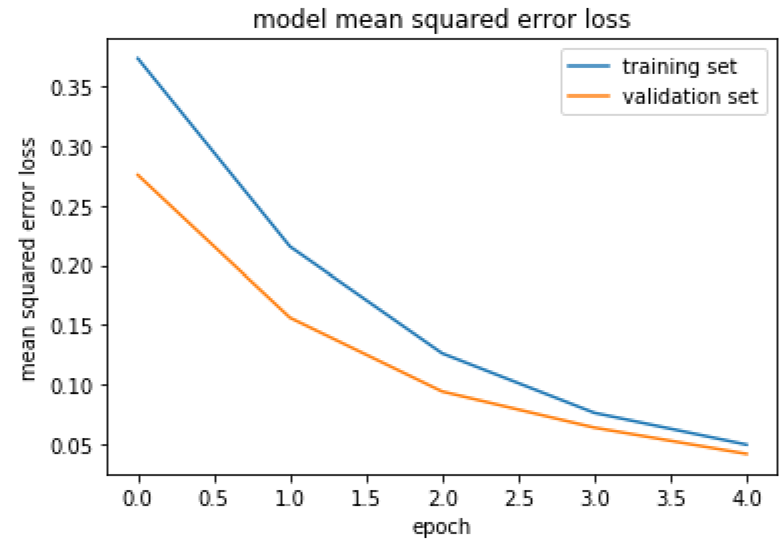

Figure 1: Model Mean Squared Error Loss over Epoch, Batch

Given the base parameters, overfitting does not occur. Based on how close the training set and validation set are at epoch = 5, I believe any more than 8 epochs will result in over fitting. The lowest recorded mean square error loss is ~0.05 which means the model is very accurate and can be used as a good baseline for the modifications that will be discussed later.

Figure 2 displays the self-driving car DNN using a batch size of 64.

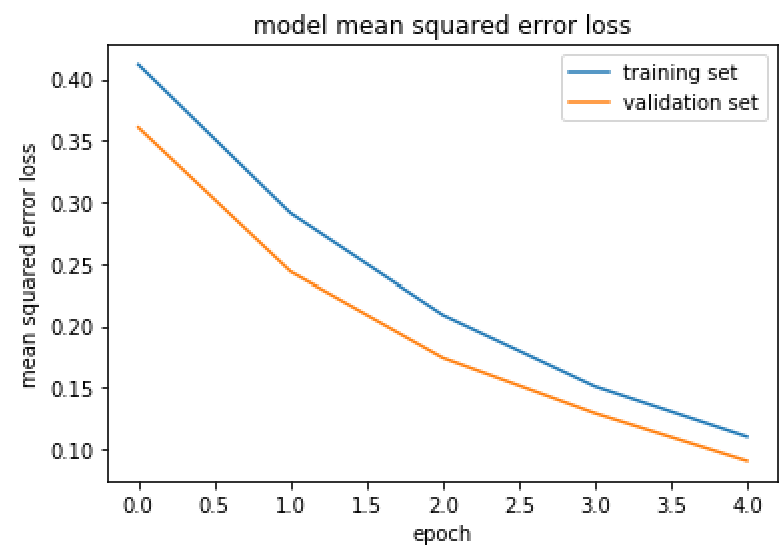

Figure 2: Model Mean Squared Error Loss over Epoch with Double Batch Size

Doubling the batch size causes a nearly linear increase in loss (0.05). This is to be expected due to how small the dataset is. Having a large batch size increases learning speed, but at the same time the model has less updates (does not learn as much).

Figure 3 displays the model using an epoch of 10.

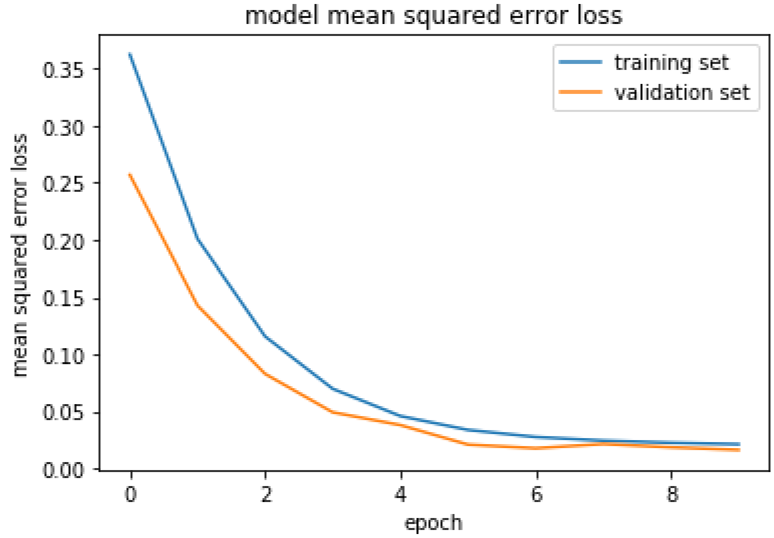

Figure 3: Model Mean Squared Error Loss over Epoch with Double Epoch

Increasing the epoch size lead to a miniscule increase in accuracy. Earlier it was hypothesized that at around the 8th epoch overfitting would occur. Based on the data gathered the model got very close to overfitting at the 8th epoch, but on close inspection the two curves never meet. With the given dataset the optimal epoch value is 6.

Figure 4 displays the model with a flipped generator.

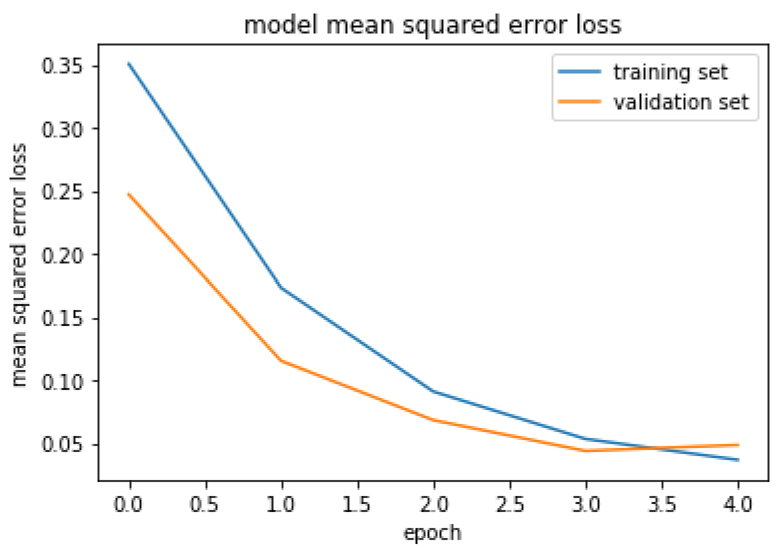

Figure 4: Model Mean Squared Error Loss over Epoch with Flipped Generator

Using a flipped generator, the model appears to be slightly more accurate, but overfitting occurs on the 5th epoch. Flipping the dataset increases accuracy by artificially diversifying the training data, assuming the flip is vertical (left-right). A horizontal flip (top-down) would be detrimental as the only time the car would be upside down is in a crash.

Based on the data gathered from Figures 1-3, adjusting the batch or epoch size has no effect on GPU utilization. This is to be expected because batch size changes how many data points are sent to the model before revisions are made and epoch means iterating through the entire dataset. Adjusting either would only affect computational time. In terms of loss doubling the batch size increased loss and doubling the epoch decreases loss. This is to be expected with how small the dataset is. Utilizing a flipped generator created interesting results as I hypothesized the effect on the model would decrease loss. The only noticeable affect is that from epochs 1-3 the rate at which loss decreased is higher than in Figure 1, but overall the end accuracy has not changed.

Figure 5 displays the model with a flipped generator using an epoch of 10.

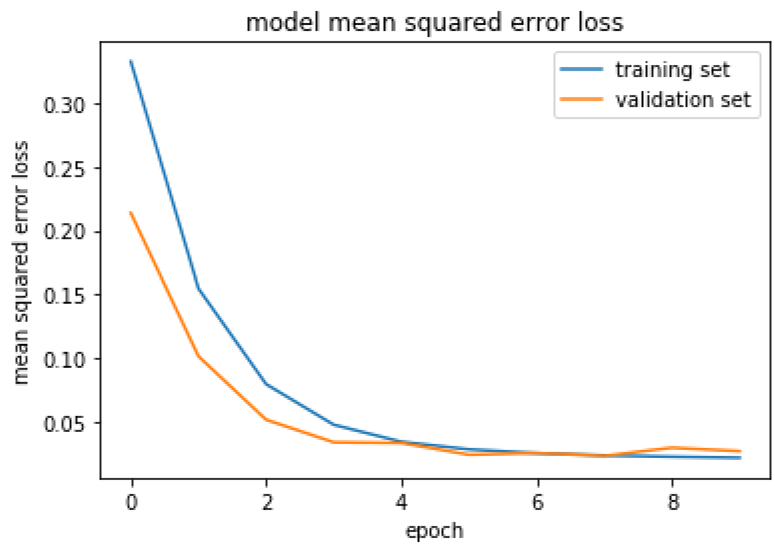

Figure 5: Model Mean Squared Error Loss over Epoch with Flipped Generator and Double Epoch

Since the flipped generator had a higher rate of accuracy, I chose that with an epoch of 10 as I believe that the overfitting on the 4th epoch may flip again given a higher epoch. When setting the epoch to 10 the validation set converges with the training set on epoch 5. Comparing Figure 5 to Figure 3, the flipped generator on the 6th epoch appears to be slightly more accurate.

Figure 6 displays the model with a small dataset gathered from the driving simulator.

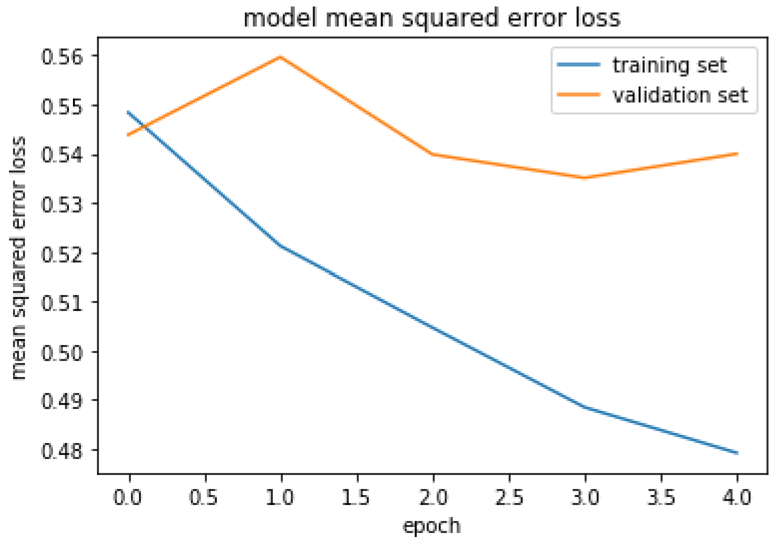

Figure 6: Model Mean Squared Error Loss over Epoch small dataset

Figure 7 displays the model with a small dataset gathered from the driving simulator. Since the data set is so small the batch size has been changed from 32 to 1.

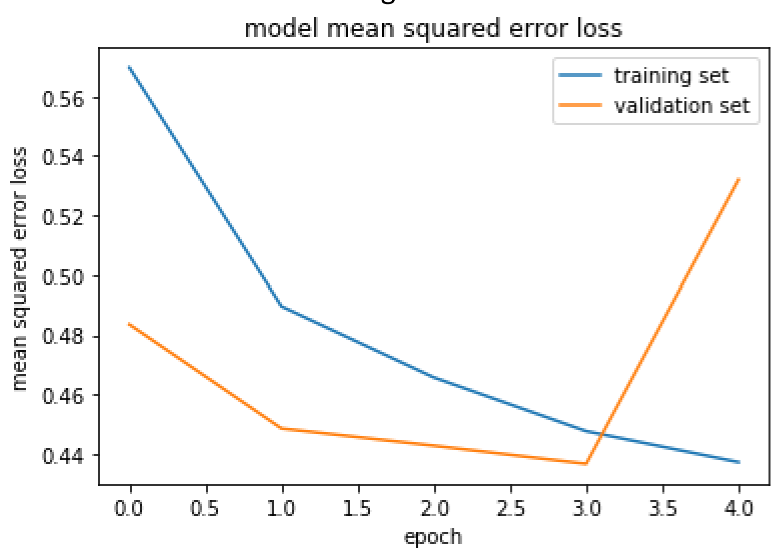

Figure 7: Model Mean Squared Error Loss over Epoch with small dataset Batch Size of One.

With the default settings of batch size = 32 and Epoch = 5, the model has a very high loss and overfitting occurs almost instantly. This is due to the data set containing only 37 entries. Such a small dataset cannot be expected to generate an accurate model, and on top of that the batch size is practically the same as the dataset. Using such a high batch size means the model only learns once before reiterating through the entire dataset. Reconfiguring the model to have a batch size of 1 causes a noticeable drop in loss. Overfitting now occurs on the 4th epoch instead of the 1st.

Figure 8 displays the model with a large dataset gathered from the driving simulator. The large dataset contains 3047 entries.

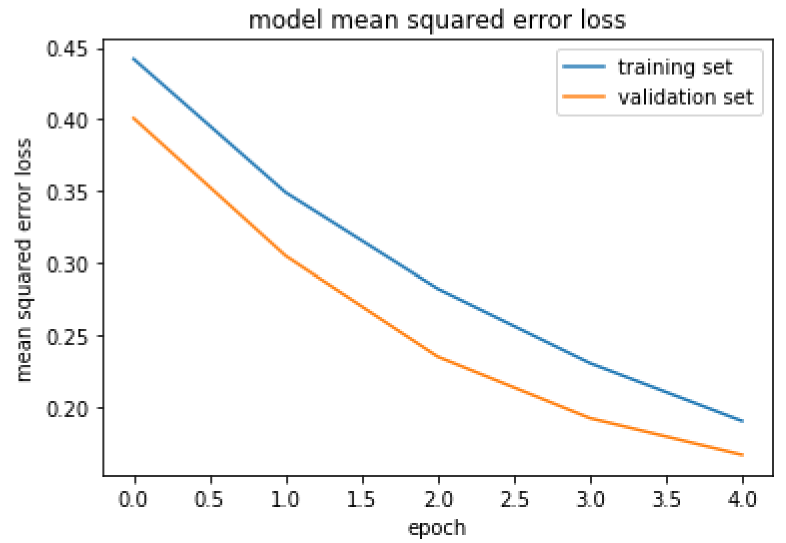

Figure 8: Model Mean Squared Error Loss over Epoch with large dataset.

Given a much larger dataset, the model improved on loss. The dataset was created by running the driving simulator and traversing the track 4 times. Due to how sensitive the steering was and how the program ran very slowly on the MSOE laptop, many iterations were needed in order to produce good data. Even then near the end of the loop there was a sharp curve that I had issues not driving over. Comparing this to Figure 1&2, one can see how closely the curves mimic each other.