

Homework 2

Question 1a: The plots are in the code. I plotted against time to show the entire dataset. We are applying standard normalization to ensure that all of the data has a std of 1 and a mean of 0. This helps prevent any data from being more seen than other data. That is any larger magnitude data from having a massive impact on the learning. This also helps the data to converge faster.

Question 1b: I ran 3 tests

- Test 1
 - 2 Layers
 - Layer 1 : 64 Neurons : Relu
 - Layer 2 : 1 Neuron : Linear
 - Hyper Parameters
 - Epochs : 50
 - Batch Size 32
 - MSE : 16000
 - R2 0.40
- Test 2
 - 3 Layers
 - Layer 1 : 64 Neurons : Relu
 - Layer 2 : 32 Neurons : Relu
 - Layer 3 : 1 Neuron : Linear
 - Hyper Parameters
 - Epochs : 50
 - Batch Size 32
 - MSE : 12000
 - R2 0.62
- Test 3
 - 3 Layers
 - Layer 1 : 64 Neurons : Relu
 - Layer 2 : 32 Neurons : Relu
 - Layer 3 : 1 Neuron : Linear
 - Hyper Parameters
 - Epochs : 50
 - Batch Size 16
 - I decided a smaller batch size would be more appropriate. It didn't introduce any instability and doubling the amount of time it updates had a pretty significant effect on my error.
 - MSE : 8000
 - R2 0.72

Question 1c: The only real change I would need to make is by square rooting my mean squared error which would be around 82. This makes my error value now comparable to theirs. I don't

think anything else matters all that much because you can now compare test sizes and hyperparameters directly now.

Question 2

- First Test
 - 1st Layer Convolution
 - 16 Filters
 - Kernel 3x3
 - Relu
 - 2nd Layer Max Pooling
 - 2x2
 - 3rd Layer Flatten
 - 4th Layer Dense
 - 128 Nodes
 - Relu
 - 5th Layer Dense
 - 4 Nodes
 - No activation Function
 - Must all be activated
 - Corresponds to x,y,width,height
- Best Test
 - Really long so it is in the code
 - Ended with a 0.0018 MSE
 - This is probably due to overfitting with a large epoch size and low dropout rate
 - This was the best at getting the entire box of a triangle but suffers on random sized squares and rectangles. It gets most of the shape but doesnt get all as it used to before the greater number of epochs. At 60 epochs i get around a 0.28 and it gets the squares pretty accurately but still suffers on the triangles