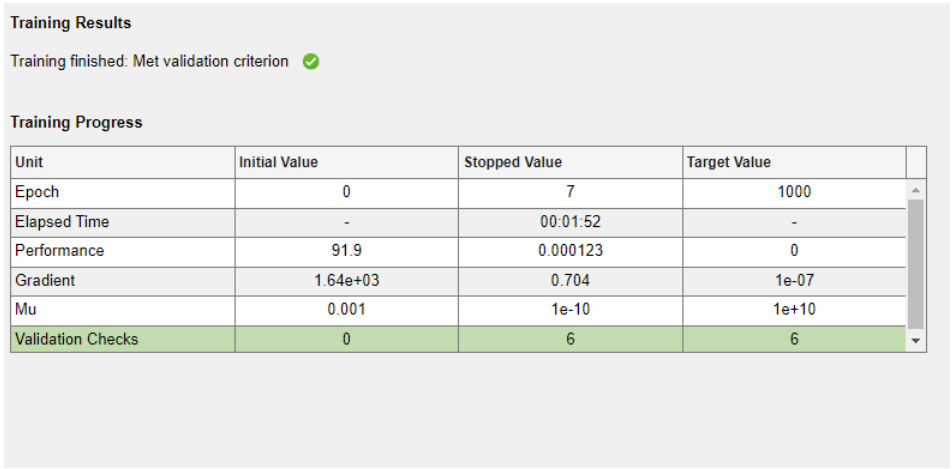
Matlab script(hw8.m)

```
%Just stuff from hw8 really, down to mnist_reduced being used again
load('mnist.mat');
load('mnist_reduced.mat');
mnistReducedX = trainX(1:1000, :);
mnistReducedY = trainY(1:1000).';
%Respectively, and reminder for myself, trainX and trainY are response
% and predictive variables respectively.
```

Output on Matlab

I used network time series to train the variables(trainX and trainY, which are loaded with only 1000 images). I was tempted to do the slowest training method possible but ultimately chose the fastest.

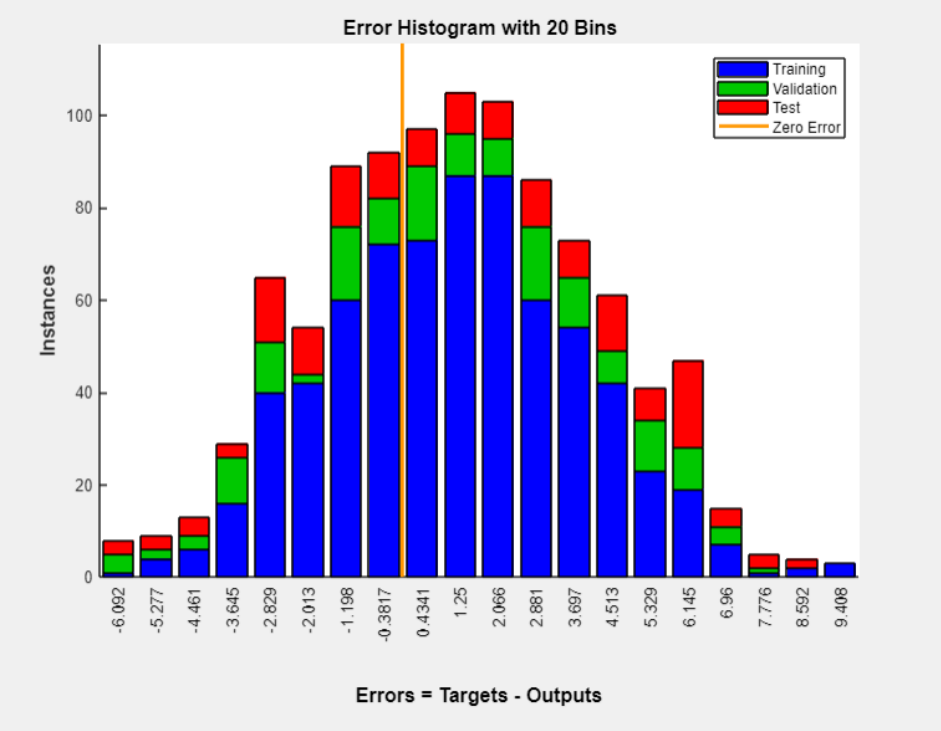


Training Results

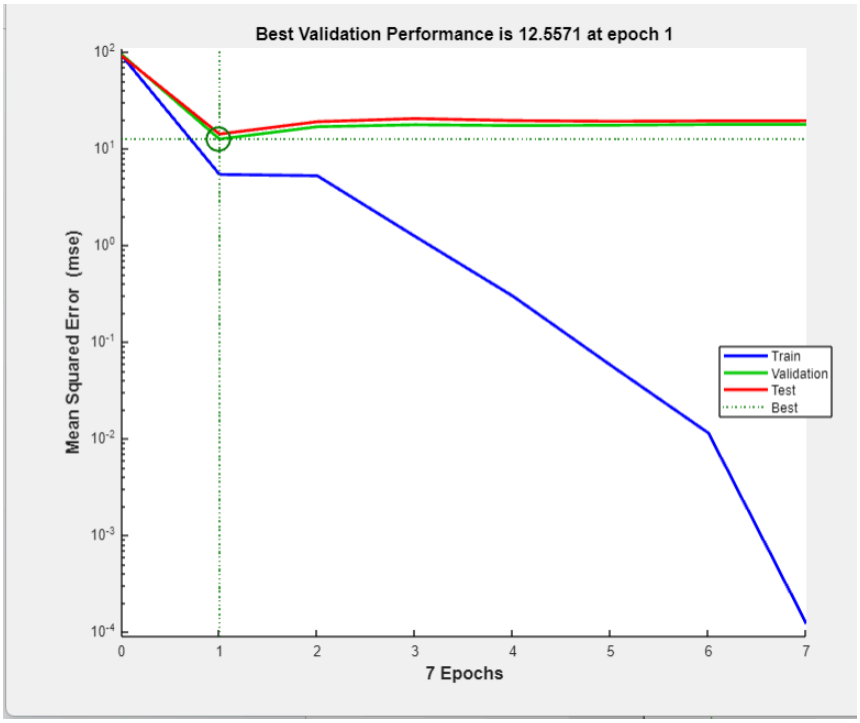
Training finished: Met validation criterion

Training Progress

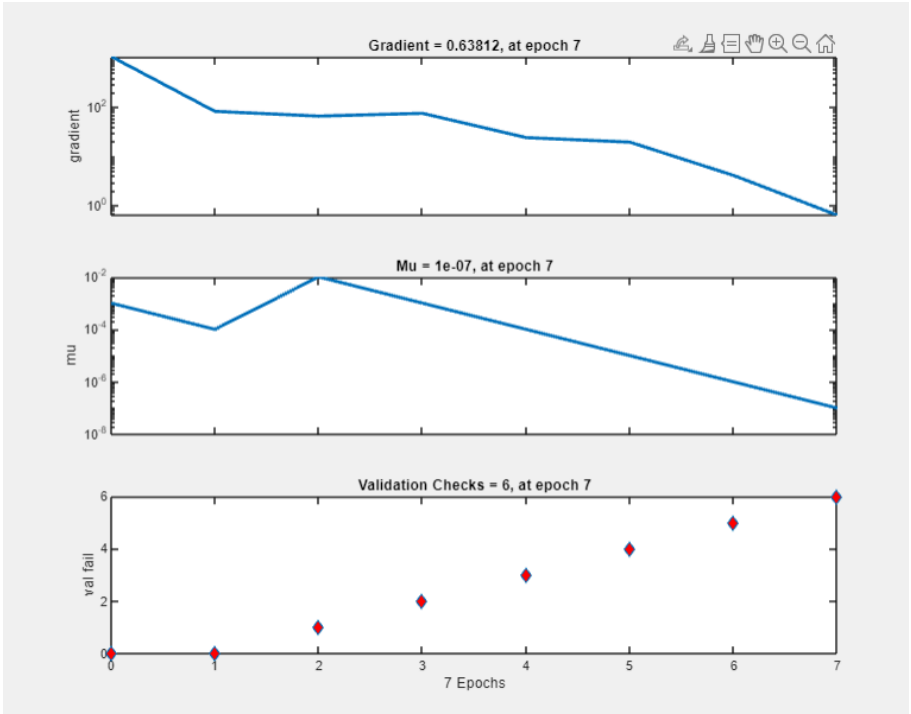
Unit	Initial Value	Stopped Value	Target Value
Epoch	0	7	1000
Elapsed Time	-	00:01:52	-
Performance	91.9	0.000123	0
Gradient	1.64e+03	0.704	1e-07
Mu	0.001	1e-10	1e+10
Validation Checks	0	6	6



It had a zero error around the or in between the 8th and 9th bin. I believe anything before the orange is underestimating the data while everything after is overestimated(at least based off of a few forum posts and matlab guides online).



As seen here, the best performance is at the 1st epoch or first test run with the images.


Gradient = 0.63812, at epoch 7
gradient
Mu = 1e-07, at epoch 7
mu
Validation Checks = 6, at epoch 7
val fail
7 Epochs