EPART Neural Networks Lab Report

Rustenis Tolpeznikas

January 12, 2025

1 Implementation

Note: Since in this solution contains no rejection rates, while also having the OK rate being $1 - error\_rate$, I will only include error rates in the tables.
The initial implementation part went quite smoothly, as I just followed the instructions of the video uploaded at *MS Teams*. With 100 hidden layer neurons, 50 epochs, leaning rate of 0.001 and $tanh()$ activation function, I obtained a reference solution, which will be used as the benchmark solution:

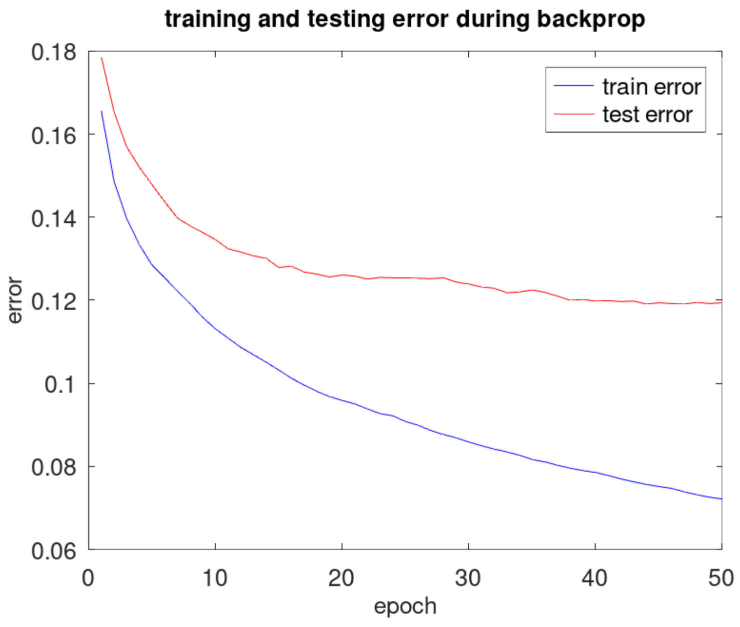

Figure 1: Reference Solution

Train Error	Test Error
0.0721	0.1194

Reference Solution Error Rates

2 Testing Different Hyperparameters

Next, I tested different Hyperparameters (Hidden Layer Neuron amount as well as the Learning Rate) to see if I can achieve better results.

2.1 Hidden Layer Neurons

I have tried 2 different amounts of hidden layer neurons, 90 and 110:

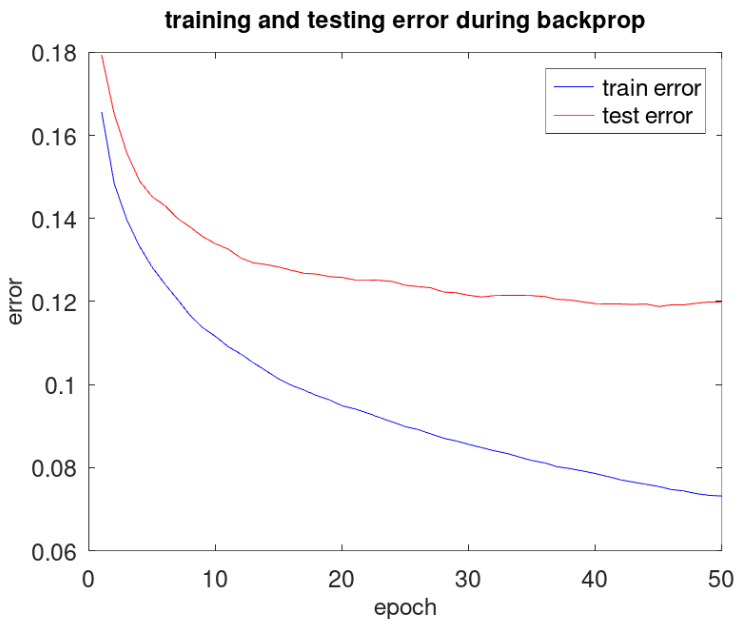

Figure 2: 90 Hidden Layer Neurons

Train Error	**Test Error**
0.0731	0.1199

Reference Solution Error Rates

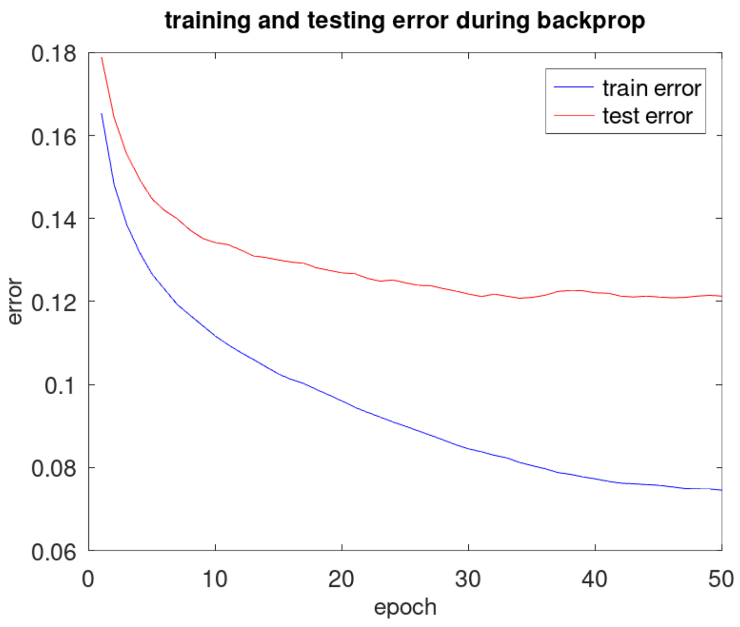

Figure 3: 110 Hidden Layer Neurons

Train Error	**Test Error**
0.0745	0.1213

Reference Solution Error Rates

The results were underwhelming, as the reference solution still had the lowest error in the test set, the one that matters.

2.2 Learning Rate

The next hyperparameter I tested was the learning rate, with 2 different values, 0.005 and 0.0005:

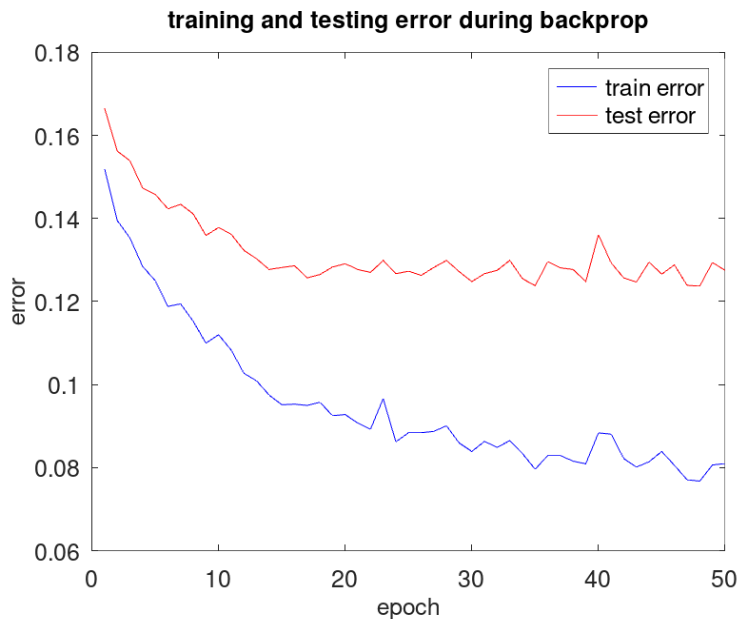

Figure 4: Learning Rate 0.005

Train Error	Test Error
0.0810	0.1275

Reference Solution Error Rates

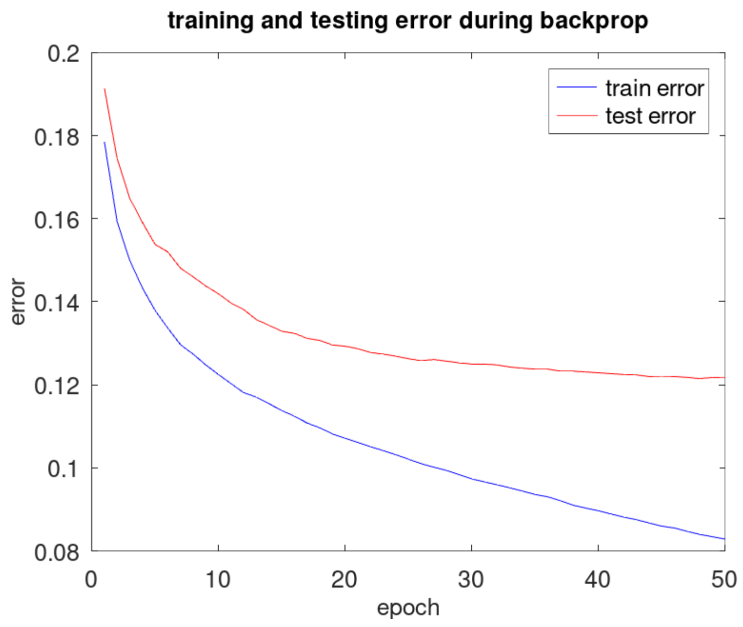

Figure 5: Learning Rate 0.0005

Train Error	Test Error
0.0828	0.1217

Reference Solution Error Rates

Once again, these results were outperformed by the reference solution. Therefore, for trying new possible improvements, I will use the reference solution as the base.

3 Possible Improvements

3.1 Data Normalization

The first possible improvement tried was normalization (standardization) of the data around the mean.

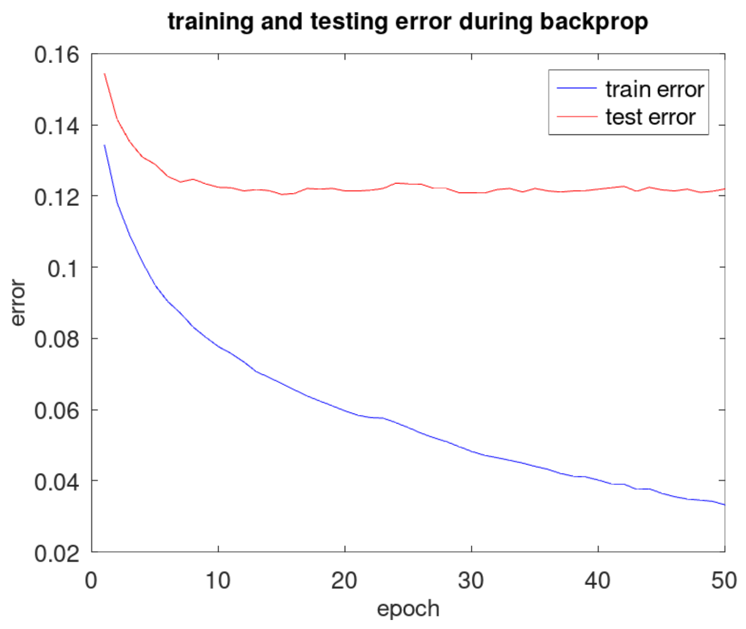

Figure 6: Normalized Data Solution

Train Error	Test Error
0.0332	0.122

Reference Solution Error Rates

There were big improvements in the error rates for the training set (From 0.07 to 0.03), but no improvements in the test set, meaning possible overfitting. This means the hidden layer neuron size can be decreased.

3.2 Momentum

As the next possible improvement, momentum was added to the learning process:

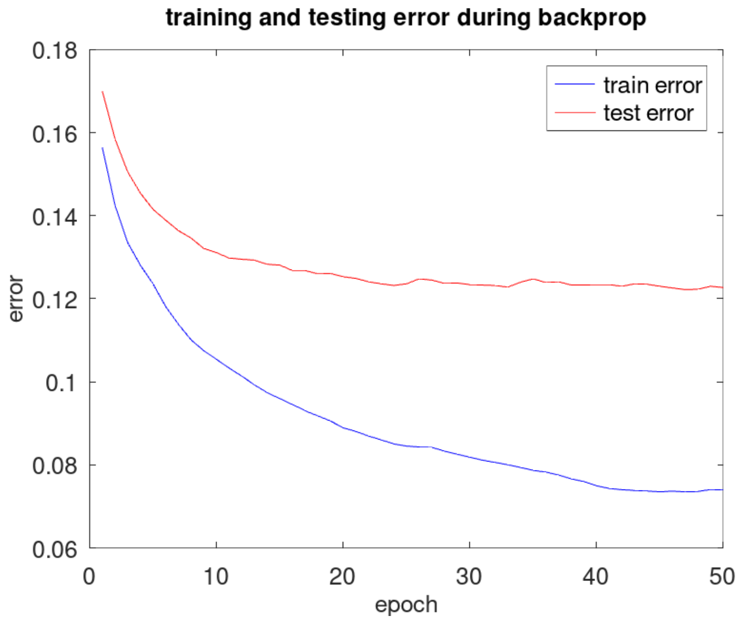

Figure 7: Momentum Solution

Train Error	**Test Error**
0.0739	0.1227

Reference Solution Error Rates

While it took a bit more time to train each epoch (roughly 3.5 seconds more), the improvement was faster in the beginning compared to the reference, then it slowed down and once again the reference solution outperformed it.

3.3 Combined Solution

Observing that both the normalization and momentum solutions still had similar error rates in the test set, but took less epochs to reach them, both were combined and the training was done for 20 epochs:

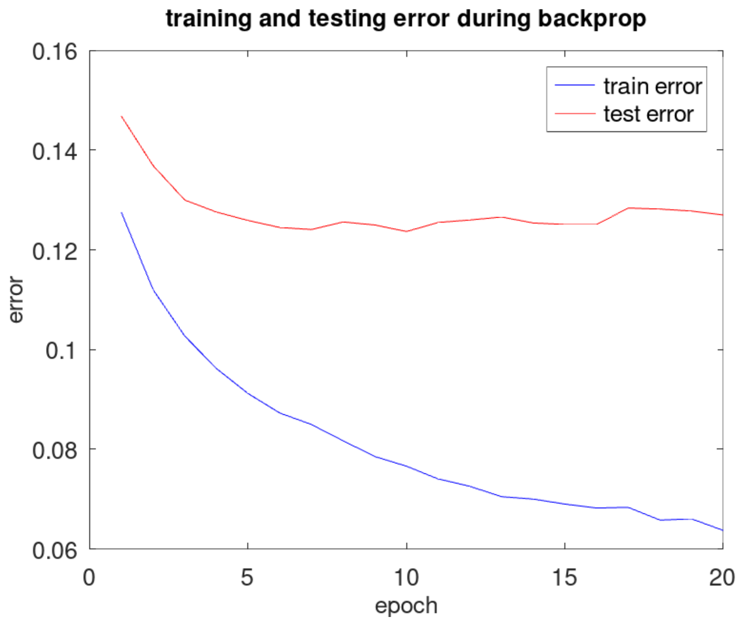

Figure 8: Momentum and Normalized Data Solution

Train Error	Test Error
0.0637	0.127

Reference Solution Error Rates

4 Final Comparison

Overall, I could not achieve any significant improvements in lowering the error rate for the test set. I did however managed to decrease the hidden layer neuron needed (100 to 90) as well as the epoch count (50 to 20), while having relatively similar results. Below is the error rate comparison for the reference and the final solution:

Model	Train Error	Test Error
Reference	0.0721	0.1194
Final	0.0637	0.127

Reference Solution Error Rates

Below are the confusion matrices for the reference and the final solution:

	Predicted										
Actual	0	1	2	3	4	5	6	7	8	9	Total
0	861	2	20	25	7	0	71	0	14	0	1000
1	3	967	3	19	5	0	1	0	2	0	1000
2	18	0	776	13	126	0	61	0	6	0	1000
3	26	13	11	891	35	0	18	0	6	0	1000
4	0	0	56	35	857	0	44	0	8	0	1000
5	0	0	1	1	0	944	0	33	2	19	1000
6	143	0	76	38	94	0	627	0	22	0	1000
7	0	0	0	0	0	18	0	961	0	21	1000
8	2	0	3	8	5	1	5	6	970	0	1000
9	0	0	0	0	0	6	0	41	1	952	1000

Reference Confusion Matrix

	Predicted										
Actual	0	1	2	3	4	5	6	7	8	9	Total
0	876	3	17	20	4	2	65	3	10	0	1000
1	8	965	0	17	4	0	6	0	0	0	1000
2	24	0	770	9	120	1	71	1	4	0	1000
3	40	11	14	875	36	2	19	0	3	0	1000
4	2	3	82	29	828	0	51	1	4	0	1000
5	0	0	1	1	0	931	0	44	2	21	1000
6	181	1	91	29	78	2	605	1	12	0	1000
7	0	0	0	0	0	9	0	970	0	21	1000
8	2	0	7	7	3	1	11	6	963	0	1000
9	0	0	0	2	0	8	1	42	0	947	1000

Final Confusion Matrix