CSE574 Introduction to Machine Learning

Programming Assignment 1

Handwritten Digits Classification

Group - 5

Rahul Derashri
Vidyadhar Reddy Annapureddy
Venkata Sai Krishna Tejaswi Kasavajhula

Handwritten Digits Classification Using Multilayer Perceptron Neural Network

The project recognizes handwritten digits by implementing Feed Forward and Back Propagation for a Multilayer Perceptron Neural Network. The input data consists of 60,000 data points, part of which is utilized to train the network and the rest is used to test the learning of the network in terms of % accuracy.

The project takes in as input 60,000 image samples of handwritten digits ranging from 0 to 9. We divide the dataset into training and validation datasets and their respective design matrices are constructed. Each data point consists of 784 features, each of which holds a normalized value between 0 and 1 depending upon the shade of grey it holds. The neural network is first trained on training data and its learning is tested against the validation dataset.

Feature selection could have been added to reject learning on pixels that don't affect the learning of the network. Considering the size of the data we're dealing with, and the tentative marginal reduction in execution times, we have not included that in our implementation.

We then compute the negative log likelihood error function with regularization with the parameters of the neural network hence constructed. For computing product of matrices, we have preferred the np.dot operation on np.matrixes over the nested iterative for loops which reduced the execution time by at least 10 times. After each round of feed forward, corresponding errors at each layer are used to update the weights w1 and w2. We update both w1 and w2 for each round of learning. At the end of maximum iterations, we obtain weight matrixes that give the least error.

Conclusion:

The input datasets for training and validation were fed to the 2-layer neural network with the hidden layer consisting of 45 nodes, and regularization parameter set to 0.5, we achieved maximum accuracy of 95.76% on the validation data and 96.74% on training data with an execution time of 150.194 seconds.

The maximum accuracy was observed for the set of configuration parameters

- Hidden nodes = 45
- Maximum Iterations = 90 (more or less constant after 60)
- Regularization Parameter = 0.5

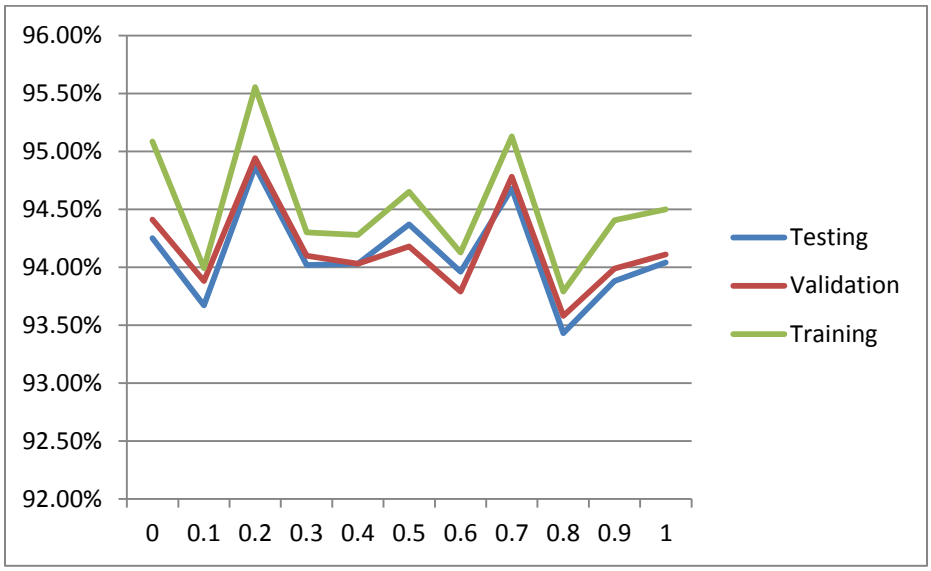

Fig1. Accuracy vs Regularization Parameter (λ) (Table1)

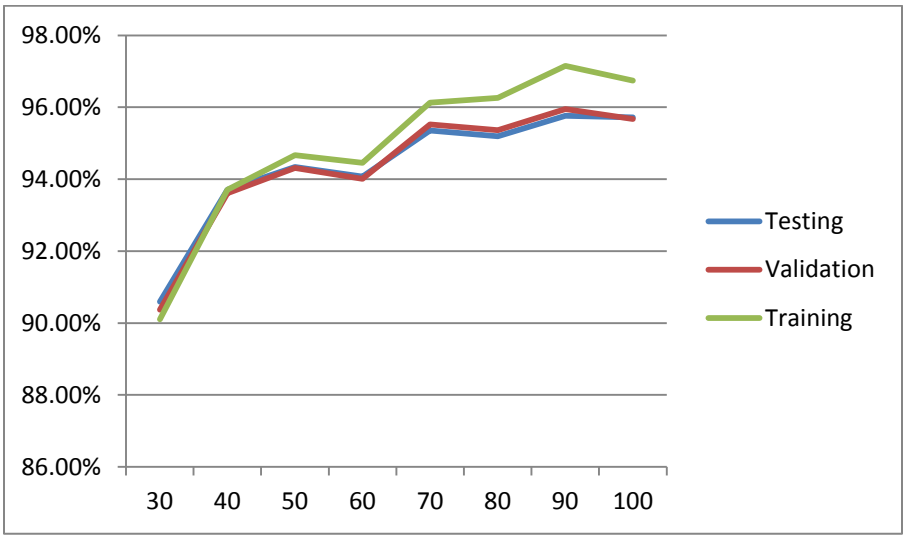

Fig2. Accuracy vs Maximum Iterations (Table2)

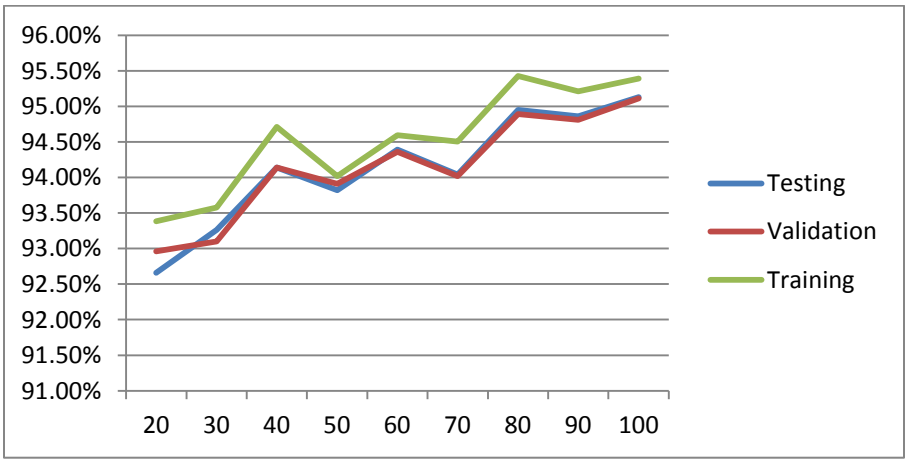

Fig3. Accuracy vs Nodes in Hidden Layer (Table3)

Appendix

Hidden nodes	Max Iterations	Lambda	Time	Training	Validation	Testing
45	50	0	76.373	95.08%	94.41%	94.25%
45	50	0.1	77.132	93.99%	93.88%	93.67%
45	50	0.2	78.151	95.55%	94.94%	94.87%
45	50	0.3	74.576	94.30%	94.10%	94.02%
45	50	0.4	73.437	94.28%	94.03%	94.03%
45	50	0.5	77.169	94.65%	94.18%	94.37%
45	50	0.6	69.049	94.13%	93.79%	93.96%
45	50	0.7	78.562	95.13%	94.78%	94.68%
45	50	0.8	73.404	93.79%	93.58%	93.43%
45	50	0.9	72.257	94.41%	93.99%	93.88%
45	50	1	77.855	94.50%	94.11%	94.04%

Table1

Hidden nodes	Max Iterations	Lambda	Time	Training	Validation	Testing
45	30	0.5	44.669	90.10%	90.37%	90.59%
45	40	0.5	58.15	93.70%	93.61%	93.71%
45	50	0.5	76.921	94.67%	94.31%	94.34%
45	60	0.5	95.082	94.46%	94.01%	94.07%
45	70	0.5	121.88	96.13%	95.52%	95.35%
45	80	0.5	131.916	96.26%	95.36%	95.19%
45	90	0.5	149.315	97.15%	95.95%	95.76%
45	100	0.5	150.194	96.74%	95.67%	95.72%

Table2

Hidden nodes	Max Iterations	Lambda	Time	Training	Validation	Testing
20	50	0.5	50.411	93.38%	92.96%	92.66%
30	50	0.5	70.662	93.58%	93.10%	93.26%
40	50	0.5	77.472	94.71%	94.14%	94.14%
50	50	0.5	82.451	94.02%	93.91%	93.82%
60	50	0.5	98.94	94.60%	94.36%	94.39%
70	50	0.5	112.641	94.50%	94.02%	94.04%
80	50	0.5	107.11	95.43%	94.89%	94.95%
90	50	0.5	108.487	95.21%	94.81%	94.86%
100	50	0.5	112.991	95.39%	95.11%	95.13%

Table3