Neural Network Classification of Digit Images

James Chang

Introduction

Neural networks and machine learning techniques have applications to image recognition. With the right architecture and many training iterations on known data, a neural network can classify new data. In this paper we classify images of digits using a feed-forward neural network and examine the performance of different network architectures on two digits. In addition, we compare the results of adding L1 and L2 regularization. Lastly, we tackle the problem of multiclass classification and apply L1 regularization to three digits.

Methods

Each image had 28 × 28 features and represents a digit. Data points were standardized and labels range from 0 to 1. Denote the training sets as Xtr, ytr and test sets as Xts, yts. Code for experiments was adapted from teaching assistants Mason McGill and Matteo Ronchi. The training set was first parsed using parseData.mat, which removed images from the training and test sets that did not correspond to the testing digits. It trimmed Xtr and ytr, and reshaped each image as a 1 by 784 array of pixels. These images were passed through a feed-forward neural network with backpropagation for the number of training iterations. Then, we modified the network's layer architecture, the amount of neurons per layer, and conducted performance testing.

Initial experiments began with differentiating 2's from 7's and studied which neural network architectures have the best accuracy and runtime. Our learning rate and regularization term were both 0.01, and we trained for 200 iterations (learning and regularization terms were not altered for the rest of the experiments). We then selected the best-performing architecture for L1 and L2 regularization to improve fitting on 2's vs. 7's, training for 2000 iterations.

The resulting error minimization over and weight changes over training iterations were compared for L1, L2, and no regularization (1500 iterations). Lastly, we attempted multiclass classification by differentiating between three digits (2's, 7's, and 9's) and choosing the digit that gives the greatest testing confidence as our final answer for each image.

Multiclass classification employed a one vs. one method, where separate networks trained on each combination of two digits, and the network that provided the greatest confidence when presented with a test image was chosen.

Architecture (Input, Hidden, Output)	Accuracy (%)		Runtime (s)
	Train	Test	
784, 2, 1	99.80	98.01	491
784, 2, 2, 1	99.70	98.06	816
784, 2, 3, 1	99.71	98.15	744
784, 3, 1	99.93	98.93	513
784, 3, 2, 1	99.86	98.40	770
784, 3, 3, 1	99.89	98.50	796

Table 1: Training Accuracies and Runtimes on Classifying 2 and 7 on 200 Training Iterations. Layer architecture [784, 3, 1] had the greatest accuracy with the second-lowest runtime of 513s.

Regularization	Accuracy (%)	
	Train	Test
None	99.91	98.35
L1	98.67	97.86
L2	98.07	97.04

Table 2: Comparing Regularizations when Classifying 2 and 7 with 2000 Training Iterations. As expected, no regularization produces the highest training accuracy. There may

Results

For the initial problem of differentiating 2's from 7's, the simpler and more naïve architectures achieved comparable performance in shorter runtimes (Table 1). Networks with more than ten neurons in any hidden layer produced marginally better accuracy but with much longer runtime. A network architecture of input layer of size 784, one hidden layer of size 3, and one output layer of size 1 produced the highest training and testing accuracy for differentiating 2's and 7's. In addition, the architecture was second fastest in training with a runtime of 770 seconds. This architecture was selected to test L1 and L2 regularization with.

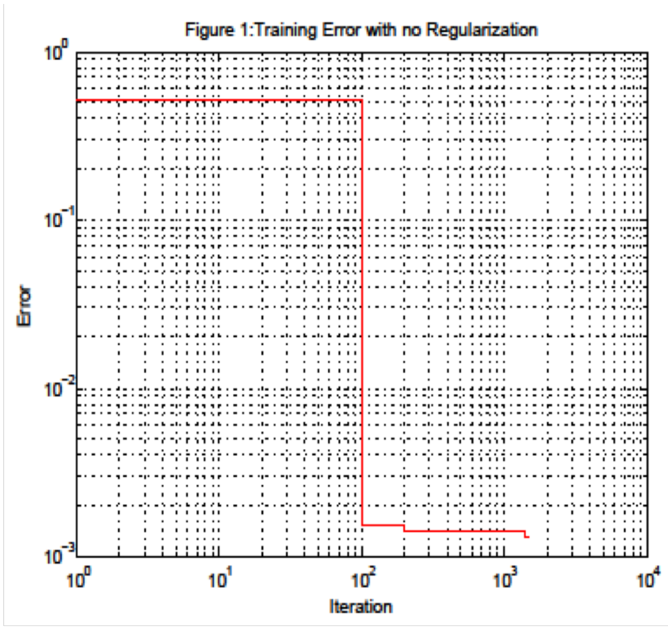

Figure 1: Training error minimizes with little perturbation.

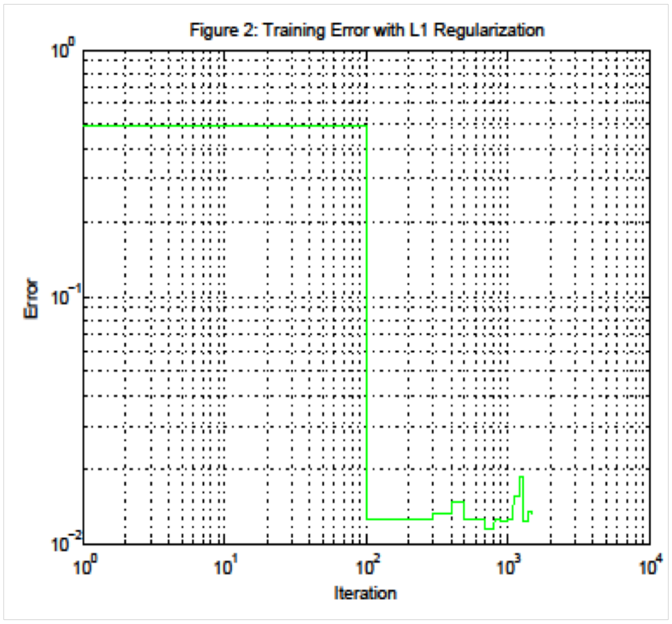

Figure 2: Training error minimizes with perturbations. L1 has potentially many sparse outcomes, and is robust to outliers.

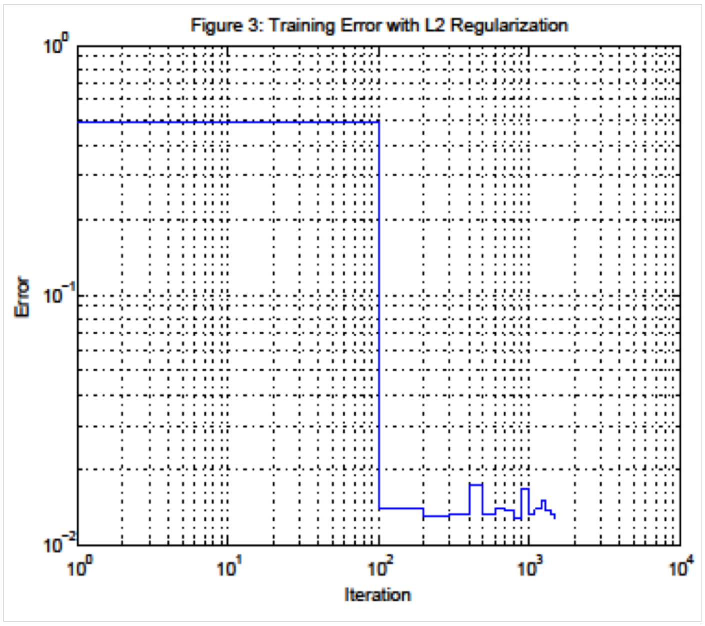

Figure 3: Training error minimizes with perturbations. L2 has potentially many sparse outcomes, but is less robust than L1.

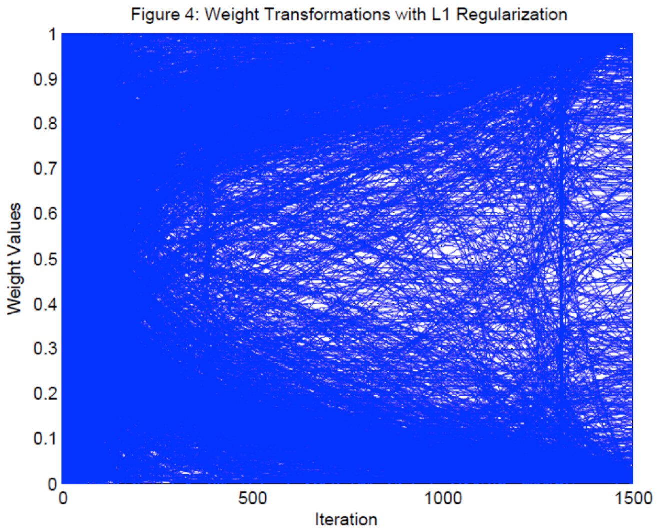

Figure 4: L1 regularization pushes irrelevant weights towards 0, producing a sparse result.

Regularization	Accuracy (%)	Iterations
None	72.01	2000
L1	80.86	2000

Table 3: Results of Three-Digit Classification. L1 regularization reduced the overall accuracy of the one vs. one multiclass classification. Perhaps there was less overfitting than originally

Regularization decreased accuracy on the training set by punishing overfitting (Table 2). Error perturbations were observed in for both L1 and L2 during training (Fig. 1 and 2). L1 regularization pushed irrelevant weights towards 0, leaving gaps between low weights and high weights, essentially creating sparseness (Fig 4).

Lastly, a one vs. one multiclass classification technique was used to distinguish between 2's, 7's, and 9's. Testing accuracy was 72.05% without regularization at 2000 iterations. Applying L1 regularization was not as successful as the two-digit case, producing 80.86% error with 2000 iterations.

Discussion

Architectures with one hidden layer were sufficient in classifying two digits. If the input could not be linearly separated, more layers would be required. As long as the classification was a continuous mapping to the output, one hidden layer was enough. The problem seemed simple enough that if more neurons were added to the hidden layer, unnecessary overfitting would occur.

Classifying two digits required a simple feed-forward network if there was little pixel overlap, and regularization performed relatively well. L1 and L2 regularization lowered accuracy in both training and testing results. This was most likely due to less pixel overlap between 2's and 7's, which did not make the reduced accuracy to alleviate overfitting tradeoff as effective.

In cases involving more significant overlap, such as the three digits' classification, regularization improved test accuracy with a significant 8.85%.

Future Work

Many investigations became feasible after this initial work. Pruning could improve the performance of any of these existing neural networks. Expanding one vs.

one multiclass classification to all ten digits would be one extension of this implementation. In addition, one vs. one techniques could be compared to one vs. many techniques. The learning rate could also be lowered for future experiments to better visualize weight and error changes over training, and regularization constants could also be changed to see how they affect weights throughout training.

Acknowledgements:
Instructor: Pietro Perona
Teaching Assistants: Mason McGill, Matteo Ronchi
Collaborators: Megan Keehan, Debbie Tsai, and Eddie Huang