CSE 574

PROJECT 3 : CLASSIFICATION

ANIKET THIGALE (50168090)

Objectives

This project is to implement and evaluate classification algorithms. The classification task will be to recognize a 28 x 28 grayscale handwritten digit image and identify it as a digit among 0, 1, 2, ... , 9.

You are required to do the following three tasks.

1. Implement logistic regression, train it on the MNIST digit images and tune hyperparameters.

2. Implement single hidden layer neural network, train it on the MNIST digit images and tune hyperparameters such as the number of units in the hidden layer.

3. Use a publicly available convolutional neural network package, train it on the MNIST digit images and tune hyperparameters.

Data Set Used

1) Training Data

For the training of our classifiers, we will use the MNIST dataset. The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems. The database contains 60,000 training images and 10,000 testing images.The original black and white (bilevel) images from MNIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. the images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

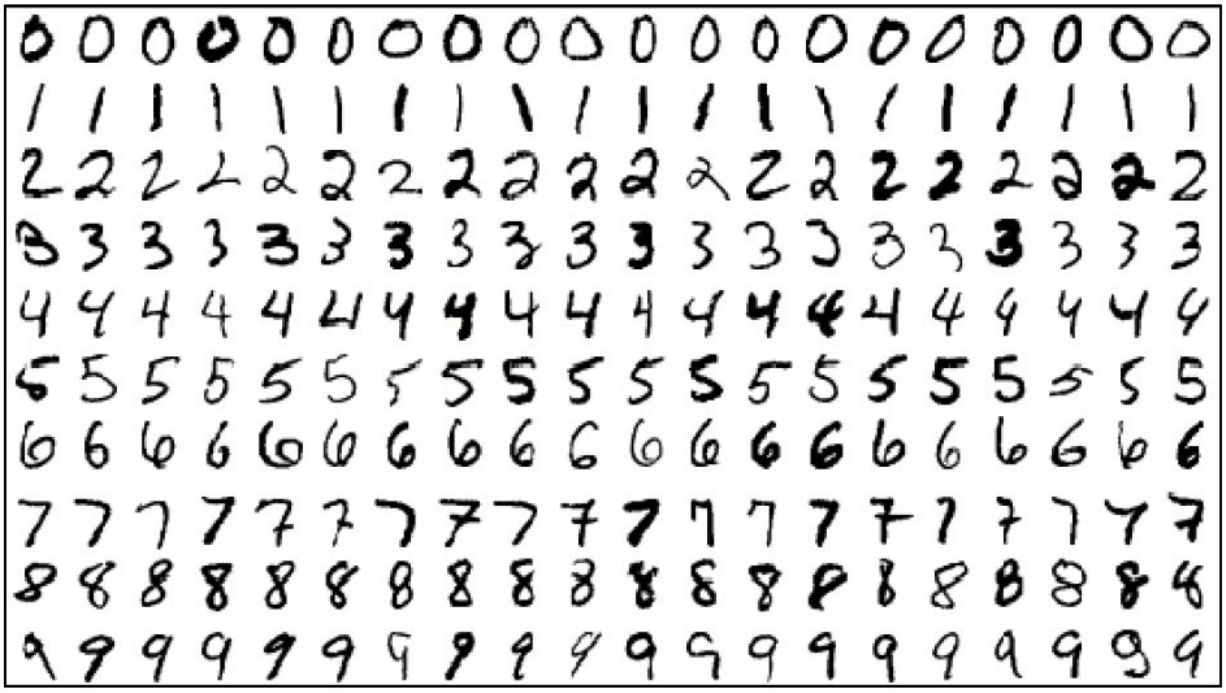


fig.) MNIST Dataset

2) Testing Data

We use USPS handwritten digit as the testing data for this project to test whether your models could be generalize to unseen data. Examples of each of the digits are given below. Each digit has 2000 samples available for testing. These are segmented images scanned at a resolution of 100ppi and cropped. This dataset is for grading use.

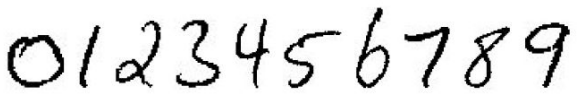


Figure 1: Examples of each of the digits

Building the Models

1) Logistic Regression

In statistics, logistic regression is a regression model where the dependent variable (DV) is categorical. There are are two types of cases : binary dependent variables—that is, where it can take only two values, such as pass/fail. Cases with more than two categories are referred to as multinomial logistic regression. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

We use 1-of-K coding scheme t = [t_1 , ..., t_k]. for our multiclass classfication task. Our multiclass logistic regression model could be represented in the form:

$$p\left(\mathcal{C}_k|\mathbf{x}\right) = y_k\left(\mathbf{x}\right) = \frac{\exp\left(a_k\right)}{\sum_j \exp\left(a_j\right)}$$

where the activation a_k are given by a_k = w_k^{τ} x + b_k. The cross-entropy error function for multiclass classification problem seeing a training sample x would be

$$E\left(\mathbf{x}\right) = -\sum_{k=1}^{K} t_k \ln y_k$$

where yk = yk (x). The gradient of the error function would be

$$\nabla_{\mathbf{w}_j} E\left(\mathbf{x}\right) = \left(y_j - t_j\right)\mathbf{x}$$

You can then use stochastic gradient descent which uses first order derivatives to update

$$\mathbf{w}_j^{t+1} = \mathbf{w}_j^t - \eta \nabla_{\mathbf{w}_j} E\left(\mathbf{x}\right)$$

to find the optimum of the error function and find the solution for w_j .
I tested my model for different values of η and used adaptive learning rate method.

Results

Grading logistic regression:

--- Misclassification rate on the 60000 train set is 9.76%

--- Misclassification rate on the 10000 test set is 10.74%

2) Neural Network

A neural network for handwriting recognition is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a sigmoid function the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read.

We are using a neural network with one hidden layer. Suppose the input layers is denoted by xi and the output is yk. The feed forward propagation is as follows:

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + b_j^{(1)}\right)$$

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + b_k^{(2)}$$

$$y_k = \frac{\exp\left(a_k\right)}{\sum_j \exp\left(a_j\right)}$$

where zj are the activation of the hidden layer and h (·) is the sigmoid activation function for the hidden layer.

$$E\left(\mathbf{x}\right) = -\sum_{k=1}^{K} t_k \ln y_k$$

where yk = yk (x). The backpropagation is done as follows,

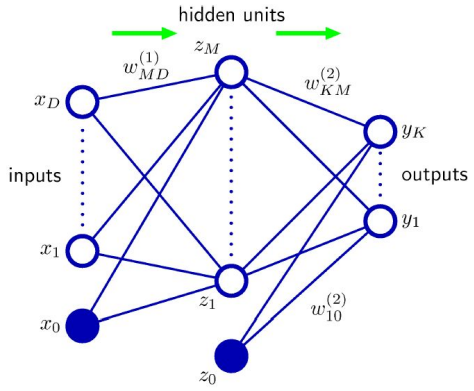


Figure 2: Network diagram for the two- layer neural network

$$\delta_k = y_k - t_k$$

$$\delta_j = h'(z_j) \sum_{k=1}^{K} w_{kj} \delta_k$$

The gradient of the error function would be

$$\frac{\partial E}{\partial w_{ji}^{(1)}} = \delta_j x_i, \qquad \frac{\partial E}{\partial w_{kj}^{(2)}} = \delta_k z_j$$

Having the gradients, we will be able to use stochastic gradient descent to train the neural network.

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla_{\mathbf{w}} E(\mathbf{x})$$

I tested my model for various values of η = [10 1 0.1 0.01 0.001 0.0001 0.00001 0.000001] and J (hidden layers) = [5 10 15 17 20 50 100 200 300 400 500 784]. Some of the results I got were,

eta : 1.000000e-02 J : 17 Correctly Classified : 8286 Wrongly Classified: 1714
eta : 1.000000e-03 J : 17 Correctly Classified : 7517 Wrongly Classified: 2483
eta : 1.000000e-06 J : 17 Correctly Classified : 974 Wrongly Classified: 9026
eta : 1.000000e-02 J : 50 Correctly Classified : 8780 Wrongly Classified: 1220
eta : 1.000000e-02 J : 300 Correctly Classified : 8877 Wrongly Classified: 1123
eta : 1.000000e-04 J : 300 Correctly Classified : 6130 Wrongly Classified: 3870
eta : 10 J : 400 Correctly Classified : 980 Wrongly Classified: 9020
eta : 1 J : 400 Correctly Classified : 7346 Wrongly Classified: 2654
eta : 1.000000e-02 J : 500 Correctly Classified : 8833 Wrongly Classified: 1167

After further tuning the hyperparameters for η = 0.01, J = 300 and running 30 times I got the optimal result as :

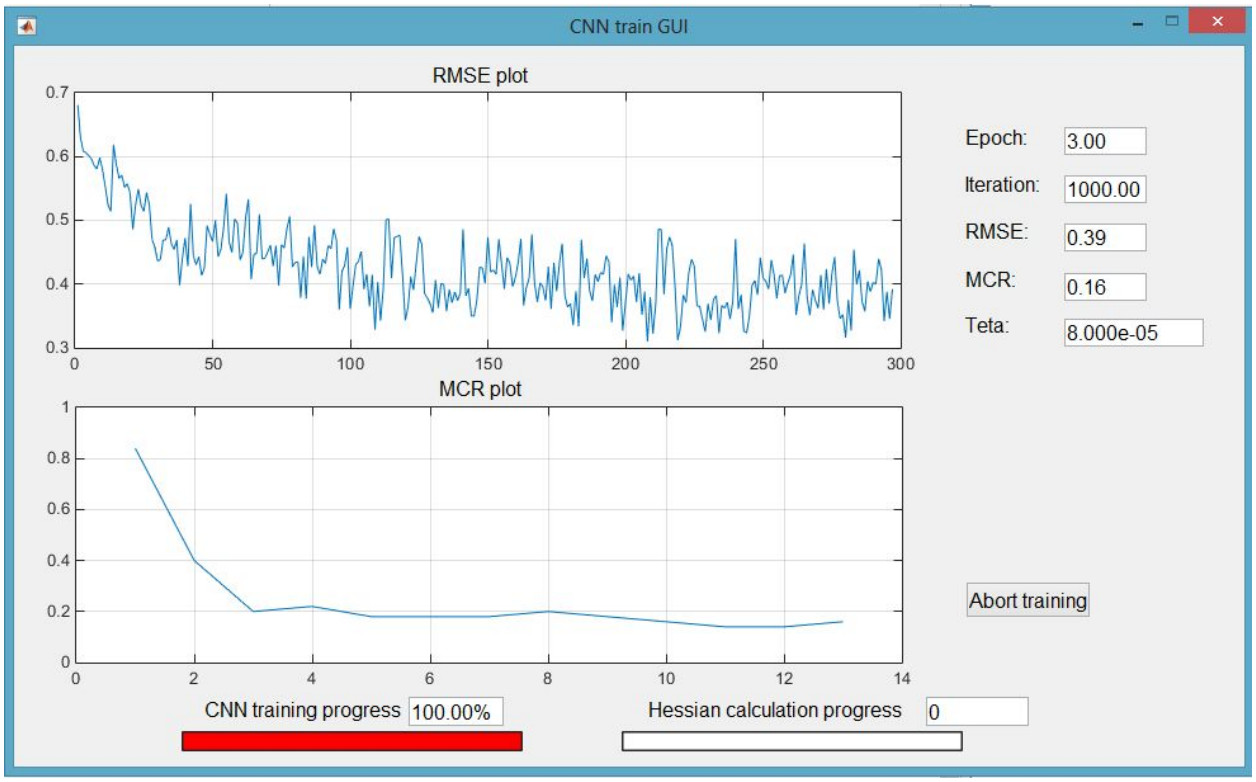
eta : 1.000000e-02 J : 300 Correctly Classified : 9325 Wrongly Classified: 675

Grading neural network:
--- Misclassification rate on the 60000 train set is 6.1267%

--- Misclassification rate on the 10000 test set is 6.47%

3) Convolutional Neural Network

I ran a CNN package available on MATLAB on the MNIST dataset with 8 layers which uses 1-of-K (10) encoding too.



Results :

I got a Misclassification Rate of 0.16 and Root Mean Square Error of 0.39.

Conclusion

We have thus trained and optimized a logistic regression model and a neural network model for classifying handwritten digits.