In The name of God

University Tehran

Engineering Facility

Electrical and Computer Engineering

Pattern Recognition

HW # 6

Amin Fadaeinedjad

810195442

Spring 98

List

Abstract

In this Homework we are going to complete the code that has been given to use and the code that we have to complete is from mlp and rbf design.

Question 1

At this Homework we need to complete the parts that are mentioned and this problem you should train an artificial neural network (MLP and RBF) for classification of the given dataset. you should fill the missing sections to complete your neural network implementation.

At first we implement a toy data set and there is going to be 3 hidden lays and 3 output layer after that we defined a loss function for our MLP neural network to compute the score after that we can calculate the loss of an input, after that we need to use the gradient bias the improve our learning and we get the gradient of the loss function and then try to move the opposite direction.

After all of this we can know easily let the data to feed the neural network

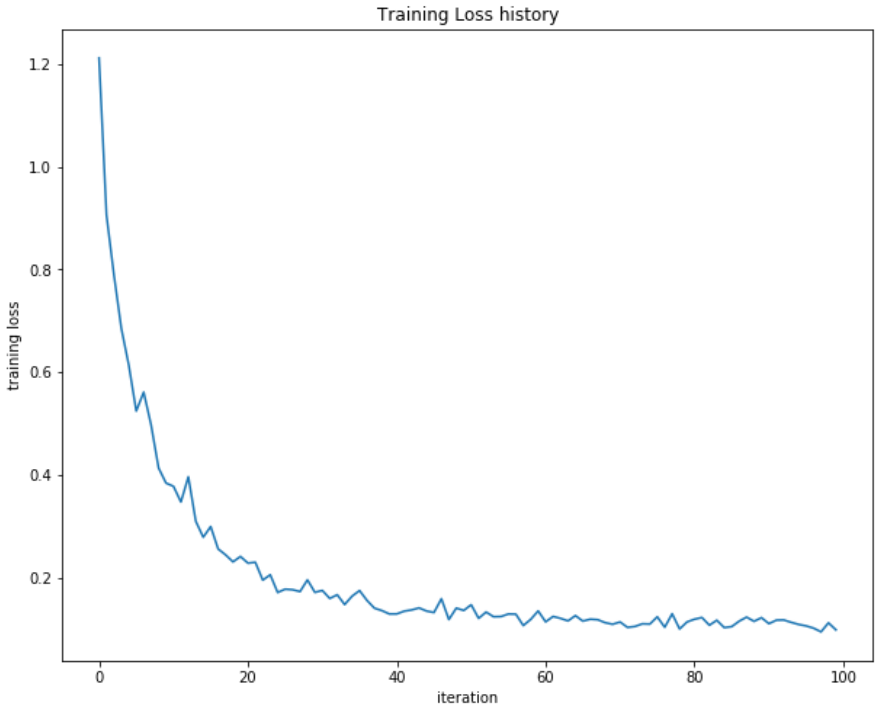


Figure 1

Figure 1 show the training low function amount. After this we need to develop our neural network with test data and we are going to us the MNIST data set.

For training our neural network we will use SGD (Stochastic Gradient descent). The loss of your network must decrease during epochs. And we start to train the NN, and after that we can plot the number of loss reduction we can see the loss function.

We define the loss as the differential of the score of the data points

The size of the data vase that we have Train data shape: (49000, 784)

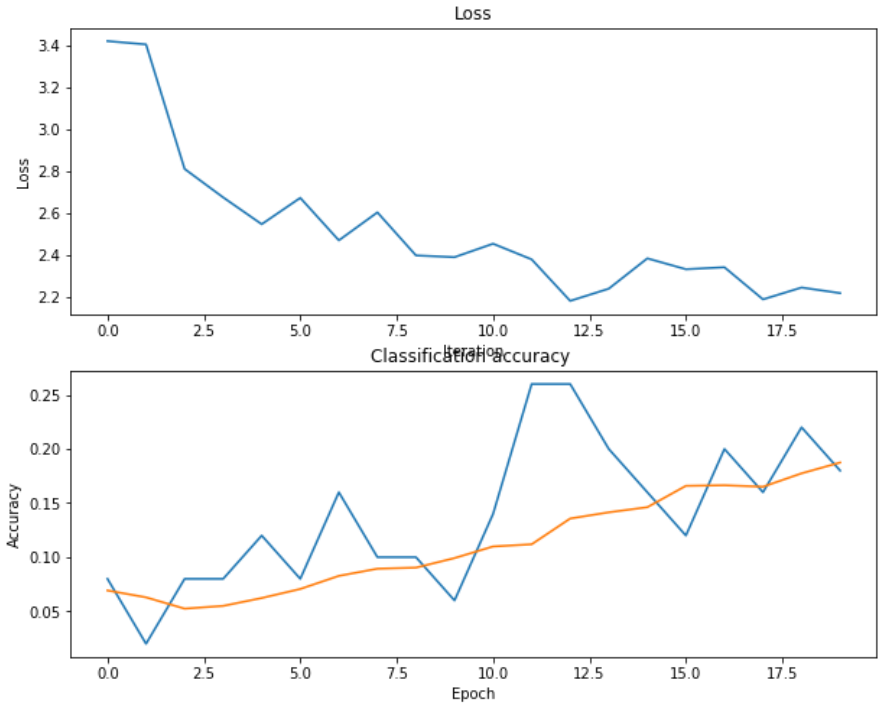
Train labels shape: (49000,)

Validation data shape: (11000, 784)

Validation labels shape: (11000,)

Test data shape: (10000, 784)

Test labels shape: (10000,)



Now we should try to tune the network's hyper parameters, as you know hidden layer size, learning rate and weight regularization are important in developing a neural network so you should tune them and get a better result and report the best hyper parameters know we want to find the hyper parameter of the network, and we are going to test it on some sample of data in the order to see that if or if not the hyper parameter the NN will get better and better and after this stuff we needed to normalize our data.

iteration 0 / 20: loss 3.420001

iteration 5 / 20: loss 2.671587

iteration 10 / 20: loss 2.452572

iteration 15 / 20: loss 2.330771

Validation accuracy: 0.18745454545454546

The following shows the loss in this time.

And in the end we need to test the NN to see that is it working ok? And we also printed the accuracy that was 82.25% with is a very high accuracy.

Know here is a quick review on what happens:

To measure the weight, they used back propagation in this way they us the error will be weighted and can be differentiated.

Question 2

In this section you should implement a RBF neural network that can be use as classifier. for do this you should fill missing part. just find optimal beta and write RBF training algorithm. We know that how the NN has been implemented and with different methods one of them was RBF. And the code works like this at first they need to choose some center of distribution and by using different type of methods we can improve the weight of the branches or the center of the distribution. In this cases we can us different methods of the center like k-mean is a good model that we have used in the past from this type of algorithm but after we need to define some things $S_i = e^{-\beta(x-c_i)^2}$ and there is another function $Y_i = centriodlabels()$ and after that we pass this data point to the RBF network and with those parameters and the thing that they have will determine the network

Process

In this homework we learned about designing a neural network and different methods around the things that have been mentioned

Reference

Automatic speech recognition a deep learning approach

stackoverflow.com

youtube.com

https://scholar.google.com/scholar?q=Forward+Algorithm+using+hidden+markov+model&hl=fa&as_sdt=0&as_vis=1&oi=scholart

https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/quadratic-approximations/a/the-hessianhttps://scikit-https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

https://math.stackexchange.com/questions/291318/derivative-of-the-2-norm-of-a-multivariate-function