

Machine Learning Ex3 Report

The Algorithm Itself:

At first we initialized our hyper parameters. Two weight matrixes, and two bias vectors. We randomized the values in each of them ranging from negative one to one. After that and reading the files we were given, we called the function to train the algorithm which returns the updated weights matrixes and bias vectors. First, in the training function, we normalized the values given to us by dividing each of the values in the nd.array by 255. Then, we zipped the training set and its correct predictions shuffled them and started iterating over them (20 times). Each iteration, called the Forward Propagation function that multiplied the weight matrix with a vector from the training set, then performed the active function, relu, normalized its values and calculated \hat{y} using the softmax function we learned in class. After that, we called the Backwards Propagation function which inside we calculated the derivative of what the relu function previously calculated, and calculated the 4 needed gradients. After we received the 4 gradients from the Backwards Propagation function, we updated each of the weights matrix and biases. The way we did that, is we just subtracted from each of the weights and biases their gradients multiplied by the learning rate. After the training algorithm finished, we returned the updated weights and ran the testing set, wrote it into a file as requested and finished the algorithm.

Below are explanations on how we chose each of our parameters and sizes of matrixes.

Choosing our parameters:

Amount of epochs: At first we tried a variety of different iterations. We tried using 5 then 10 and found the results to be good but not enough. After that we chose to keep it at 10 and change the learning rate and the size of the matrix in order to get better results. After changing them a few times, we came to realize that we might need more iterations than 10. As such we tried around 15, 20, 30 and found that our best results were given when it was 20 iterations.

Choosing our Hyper Parameters:

Our first weight matrix, we chose to be 150 by 784. 784 as the size of each item in the training set, the 150 was chosen after attempting many different numbers ranging from 10 to around 200. Prior to trying various numbers, 150 turned the best results, while above 150 gave us over fitting results and below that gave us lesser success rates by a few percent. The second layer weight matrix, we chose to be 10 by 150. We chose 10 there are 10 possible predictions whereas 150 was chosen so that we will be able to perform proper matrix multiplications later on. The sizes of the biases, b_1 and b_2 , were chosen so that we will be able to add them to the result of the matrix multiplication. As for the starting numbers of each of these hyper parameters, we chose to have it randomized between -1 and 1. After attempting a few ranges of numbers, one time between -0.05 and 0.05 and also between -0.2 and 0.2 as well as a few others, our best results were given with -1 to 1 randomized numbers.

The learning rate we chose to be 0.02. Similar to our last exercise, we tried different numbers ranging in that area, the first being 0.01 as we chose in the previous exercise. However, it did not provide the best results and we had to start choosing different numbers. Each time we tried adding a little bit more to the learning rate, then running the algorithm with a validation set. Our highest attempt was a learning rate of 0.1 which gave us pretty bad results and from all our tests we decided that 0.02 was the best learning rate for our algorithm.