İrem Şahin

ENGR421

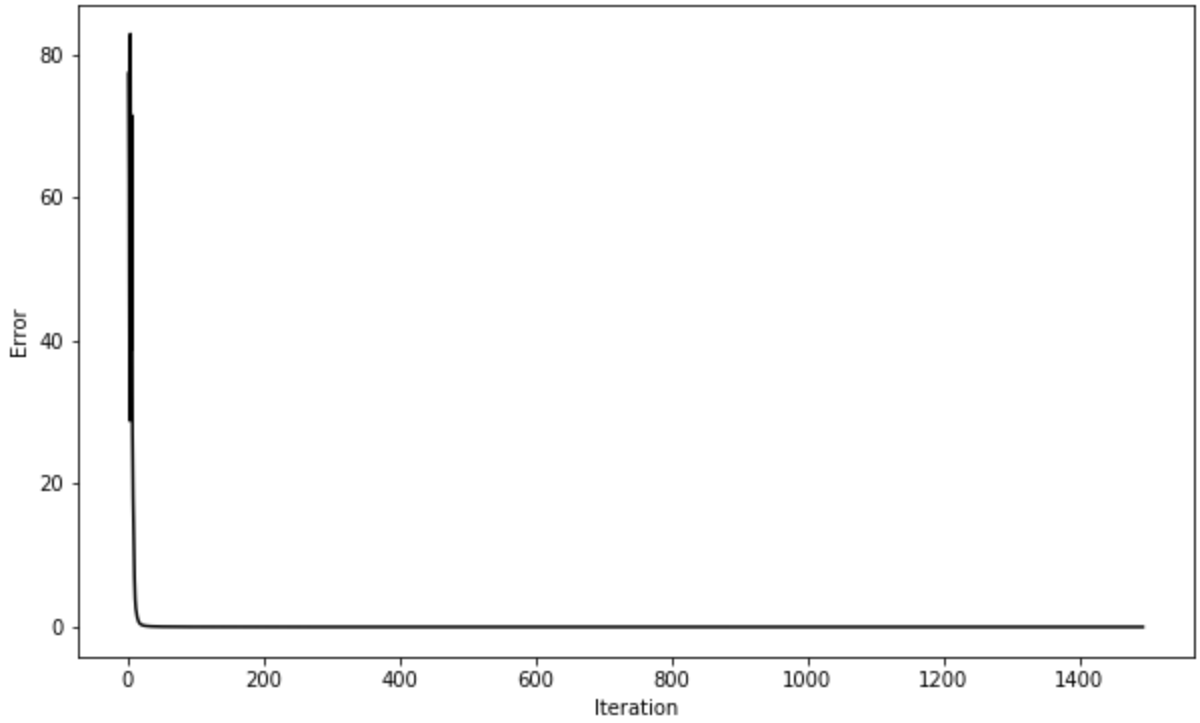
Mehmet Gönen

HW2-Report

For this project, I have written my code in a Jupyter Notebook .ipynb file. I have started my code by importing necessary libraries and writing our parameters down which were given in the pdf file. After this I have imported our data set and class labels from the given .csv files. I have continued by taking K(# of classes) and N(# of samples) values for future use. Then, as requested in the pdf file, I have divided the first 25 samples of each class for the training set, and the remaining 14 samples for the test set. Finally, I have created a one-hot encoded Train_truth variable for gradient descent loop. In the end, I had a training set which is composed of 125 points(125x320) and a test set composed of 70 points(70x320). I have used normal list appending in these as using numpy is still new for me and I couldn't comprehend how to implement appending in numpy arrays. After using normal list appending, I just turned them into numpy arrays to use their transpose,shape etc. built-ins.

Next, I have defined the sigmoid function and gradient w-w0 functions for gradient descent method as we learned them in the class. After randomly initializing w and w0 values with correct dimensions, it was time for gradient descent. By using the parameters we have defined before, I have calculated the w and w0 values. I have used half of the sum of squared errors. My condition for break was when the square root of the sum of squares of difference between old and new values. For feed 421, this resulted in 1493 iterations.

Next step was to visualize our convergence. By using the objective values I have gathered in the iterations, I had the following convergence plot:



Last step was to plot confusion matrices and check our test set results with the w and w0 we have found. Similar to the HW1, I have used argmax to get the index of the largest value which corresponds to the class of the data. The confusion matrix after crosstab was like below:

train_truth	1	2	3	4	5
train_pred					
1	25	0	0	0	0
2	0	25	0	0	0
3	0	0	25	0	0
4	0	0	0	25	0
5	0	0	0	0	25

By using the last values stored in the w and w0, I have calculated the sigmoid for the test set and calculated its confusion matrix either. The outcome was like below, with only 1 mislabeled point.

test_truth	1	2	3	4	5
test_pred					
1	14	0	0	0	0
2	0	13	0	0	0
3	0	0	14	0	0
4	0	1	0	14	0
5	0	0	0	0	14