Eda Gür

50488

HW2

I read the image and label data as in the previous hw. Then, I define the variables for K (number of classes) , d (number of pixels per sample), N\_training (number of samples for training), epsilon, onematrix, X (data matrix), X\_test (data matrix for the test), Y (label matrix for training) and Y(label matrix for test). I also set the seed to 521 as instructed. I didn’t define eta here, but I give it as variable to the gradient function.

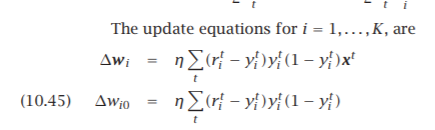
I have a for loop to feed the data to the previously mentioned variables X, X\_test, Y, Y\_test.

I define the sigmoid function based on the formula on the book:



My score variable in the sigmoid function is what is inside the exp in this formula.

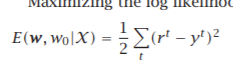
Then, I define the gradient\_W function and gradient\_W0 functions. They are based on the formulas:



r(it) represents the real label and y(it) represents the predicted label scoring matrix in the formula. These correspond to the Y and Y\_scoring variables in my functions. The X is the data matrix. I use colsums for the sigma notation. I multiply the results with eta.

I assign W and w0 from a uniform distribution from -0.01 to 0.01. W is a matrix w0 is a scalar. I define a vector for objective values.

This is the while loop: I get a scoring matrix for both test and training with the sigmoid function I created. The objective value is calculated by the formula:



The r(t) is the real value (Y) and the yt is the Y\_scoring\_training. I calculated W and w0 with the gradient functions I defined.

If the sum of squares of W and W0 converge to epsilon, we can break out of the loop.

After the loop, I draw the graph for the objective values with the existing plot function.

After this, I compare the real labels with the predicted labels. (These are derived from the label matrices with the which.max function.) The comparison with the table function gives the confusion matrices for the training and test.

Below are my results:

