Ofer Orgal
300459898
oferorgal@mail.tau.ac.il

The files are in my directory on nova.cs.tau.ac.il:
/specific/a/home/cc/students/cs/oferorgal/ml/hw3/

If for some reason the files are not accessible, I created a share in my google drive to my files:
https://drive.google.com/drive/folders/0B6K9SrEqgeqRTGdPTGlWUUpCeDg?usp=sharing

Folder content:
Python files:

hw3.py - programing exercise 3

Reports:

hw3.pdf - exercise report
README

Plots and images:

plot6aC.png
plot6aEta.png
plot6aC_with_kernel.png
plot6aEta_with_kernel.png
weights_digit_0.png
weights_digit_1.png
weights_digit_2.png
weights_digit_3.png
weights_digit_4.png
weights_digit_5.png
weights_digit_6.png
weights_digit_7.png
weights_digit_8.png
weights_digit_9.png

How to use the files:

Each section in the exercise is accessible by running the file name and the section number and letter (i.e. 6a , 6b , ... 7a, ...)

programing exercise 3

Question 6:

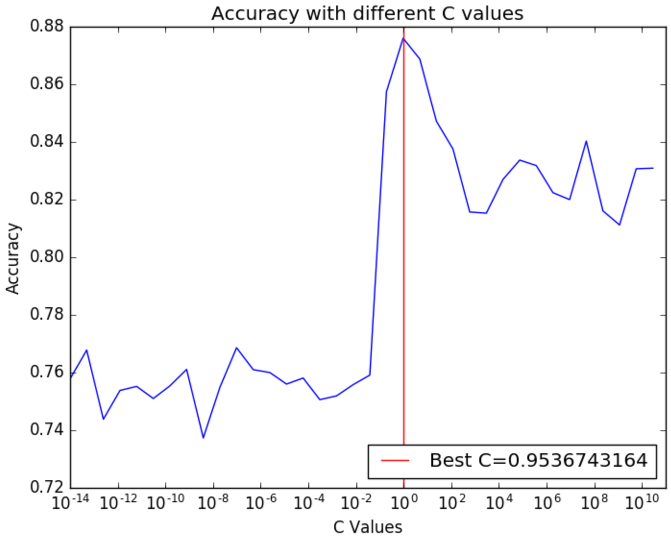
sec. A:

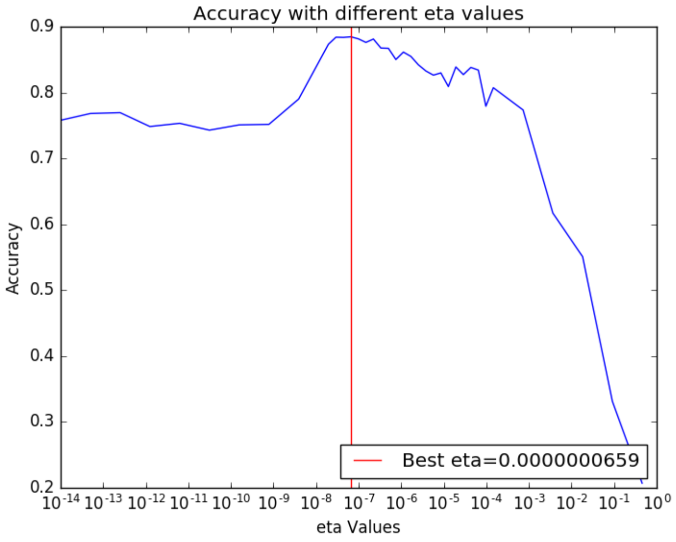
How to use: ***python2.7 hw3.py 6a***

The functions *find_best_eta(train_data, train_labels, C, T, K)* and *find_best_C(train_data, train_labels, best_eta, T, K)*, each run the multiClassSDG function with different eta and C values to determent witch are the best values basing on the validation set.

My search for eta and C begins by scanning from 10e-10 to 10e10 in multiplies of 10 and then I narrow it down and where I see the max value, I multiply be less (1.1 ~ 5) to find a more accurate value.

I got an accuracy according to the eta and C values as shown in the drawing:





sec. B:

How to use: ***python2.7 hw3.py 6b***

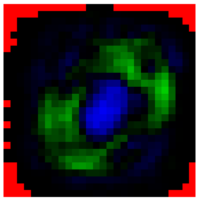
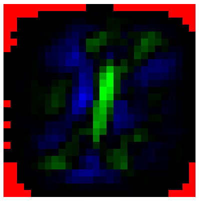
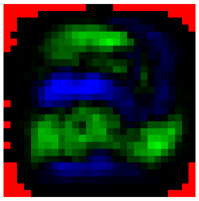
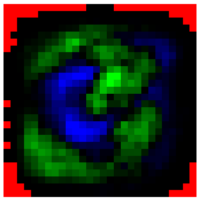
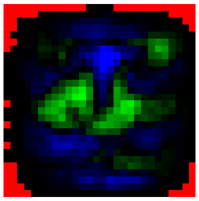
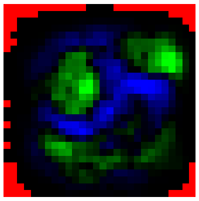
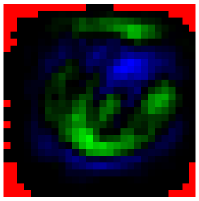
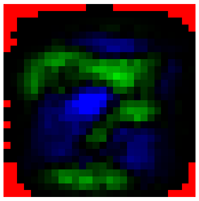
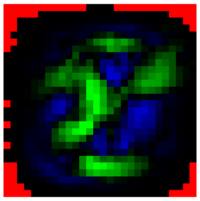
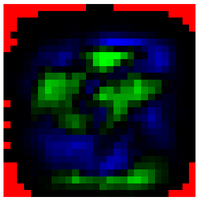
I use the best eta and C values I found before and re-train the algorithm and for each digit we get a weights vector.

The function: *print_image("weights_digit_%d.png" %i, w[i], 500000)* gets the image name to output, the data and *500000* is a brightness value.

The output:

The colors represent:

- RED: Zero values of the weights in that area.
- GREEN: The weights values are positive.
- BLUE: The weights values are negative.

0	1	2	3	4
				
5	6	7	8	9
				

It is almost clear to see the digit in the images.

sec. C:

How to use: ***python2.7 hw3.py 6c***

I use the *accuracy(test_data, test_labels, w, K)* with the *multiClassSDG(train_data, train_labels, Best_C, Best_eta, T, K)* function with the best eta and C values to find the algorithm accuracy on the test set.

The result I found is: 89.81 % in determining the digits.

Question 7:

sec. A:

How to use: ***python2.7 hw3.py 7a***

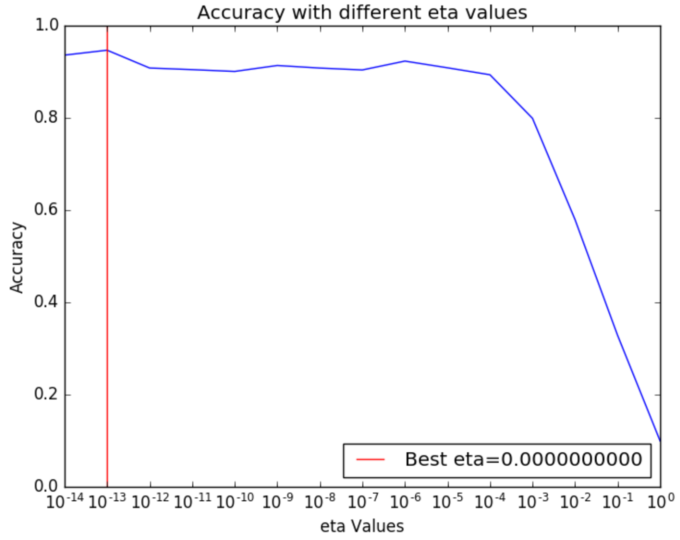
(THE KERNEL ALGORITHM IN THIS SECTION IS NOT MY ORIGINAL WORK, I FOUND IT ONLINE AND MADE MODIFICATIONS TO FIT MY CODE. I UNDERSTAND IF THERE WILL BE SIGNIFICANT POINT REDUCTION)

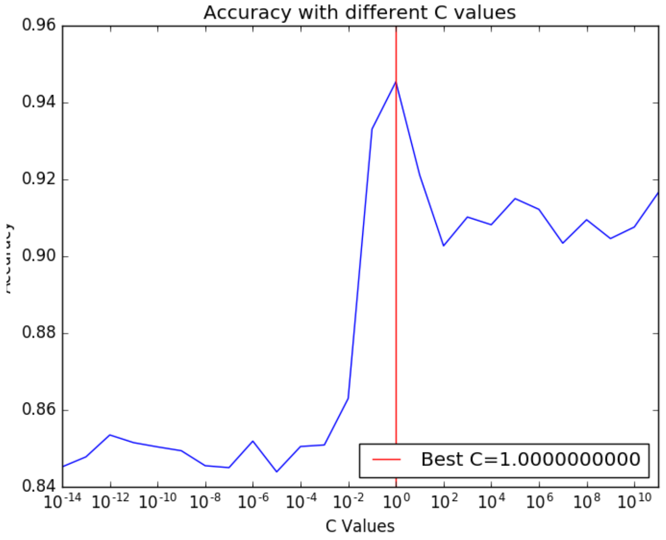
The functions *find_best_eta_with_kernel(train_data, train_labels, C, T, K)* and *find_best_C_with_kernel(train_data, train_labels, best_eta, T, K),* each run the multiClassSDG_with_kernel function with different eta and C values to determent witch are the best values basing on the validation set.

My search for eta and C begins by scanning from 10e-15 to 10e10 (for C, for eta I go up to 1) in multiplies of 10 I didn't scan more accurately because it takes a very long time.

Best eta value is at 10-e13 and best C value is 1.

I got an accuracy according to the eta and C values as shown in the drawing:





sec. B:

How to use: ***python2.7 hw3.py 7b***

I use the *accuracy_with_kernel(data, labels, alpha, kern_prod, K)* with the *multiClassSDG_with_Kernel(train, labels, C, eta, T, K)* function with the best eta and C values to find the algorithm accuracy on the test set.
The result I found is: 92.1 % in determining the digits.