Intro To ML – HW2

students:

Dor Bank - 301380416 - dorbank@gmail.com, Avi Caciularu - 203056585 - avi.c33@gmail.com

**Programming Questions**

**The code source files stay under /specific/a/home/cc/students/cs/avicaciularu/ML/HW2/:**

**Q1- Q1.py**

**Q2- Q2.py**

**Q3- Q3.py**

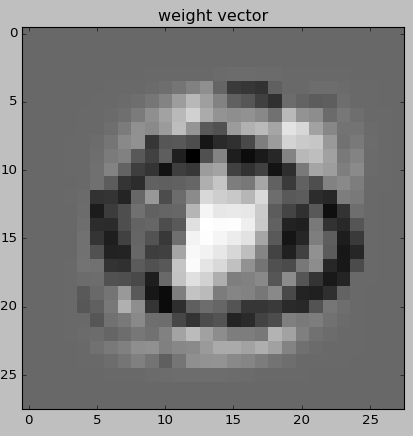
**Question 1:**

1. This table is available and printed during the code as well:

|  |  |  |  |
| --- | --- | --- | --- |
|  | mean | 5% | 95% |
| n=5 | 0.822451 | 0.580348 | 0.935005 |
| n=10 | 0.864985 | 0.692426 | 0.949335 |
| n=50 | 0.921760 | 0.803480 | 0.969806 |
| n=100 | 0.920793 | 0.817298 | 0.969806 |
| n=500 | 0.925855 | 0.842375 | 0.972876 |
| n=1000 | 0.927963 | 0.791709 | 0.972876 |
| n=5000 | 0.924693 | 0.823439 | 0.972364 |

As it can be seen from the mean column, the more samples we get, the better our model will be. In addition, we can see from the 5% column that it is also harder to get "unlucky" with the training samples and order when we have more samples. About 95% of these are cases, we got "lucky". We can see that the differences in this case are much minor, which also makes sense, because for 100 paces, it seems right we will get close to the optimal possiblly a few times.

1. In order to get a best-fitting , we ran the perceptron algorithm for 100 paces (just like in the previous section) and took the with the highest test accuracy. The plotted result:



Explanation: Recall that '8' is marked labeled in our case as 1 and '0' is marked as -1. Additionally, the higher a value of a coordinate in is, the brighter it is in the picture. This means that the brighter a pixel is, the more it is likely to appear in '8' and not in '0'.

Next, notice that the middle pixels (which really are far more related to '8') are white, while a black circle marks the pixels which are far more related to '0' than '8'.

That surrounding pixels are grey because they aren't relevant for decision making.

Interestingly, there are bright-grey pixels above and below the black circles. These pixels overlap the '8' and '0' digits and this probably means that most '0's are flatter than most '8's.

1. Since we ran the perceptron for 100 times, we made the code print the mean accuracy (so we can verify it is better than ), and the best accuracy (so we can point out our chosen ’s accuracy).

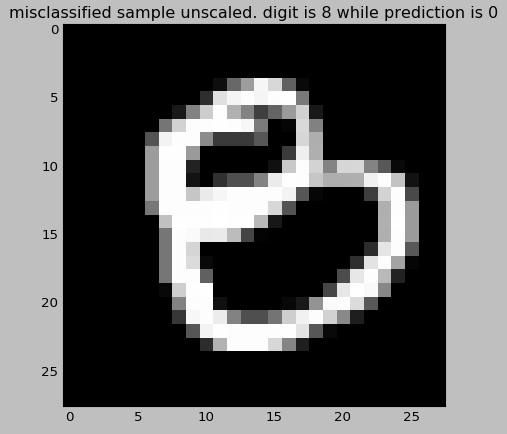
Prints from the code:

('fully trained perceptron accuracy averaged over 100 experiments: ', 0.93354657113613104)

('fully trained perceptron best test accuracy: ', 0.98106448311156602)

So we got accuracy of when averaged 100 experiments, when was the best.

1. Please notice that our code chooses 1 misclassified sample randomly, so at each run we get a different misclassified sample with an explanation at the title. One run plotted us the following image:

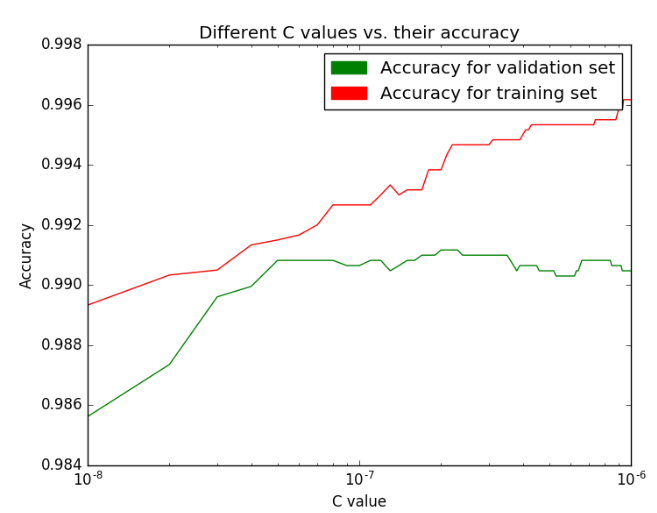


Explanation: following our explanation to , we can see than the bottom circle of this ‘8’ overlaps our black circle at w which corresponds to '0'. In addition, this '8' is flatter than the “normal” '8' and overlaps some of the left black pixels of . Finally, we will point that this '8' doesn’t include a part of the brightest pixels at , which are most distinctive towards '8'.

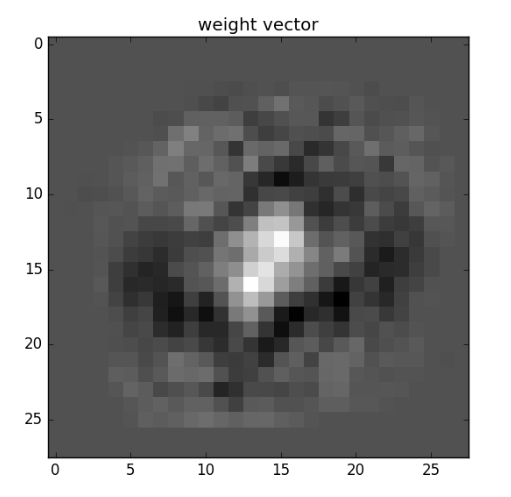
**Question 2:**

1. We performed grid search:
2. Firstly, for we found that gives the best accuracy.
3. Secondly, for we found that gives the best accuracy.

The last grid search is represented in the plotted graph, with the accuracy calculation over the validation test:



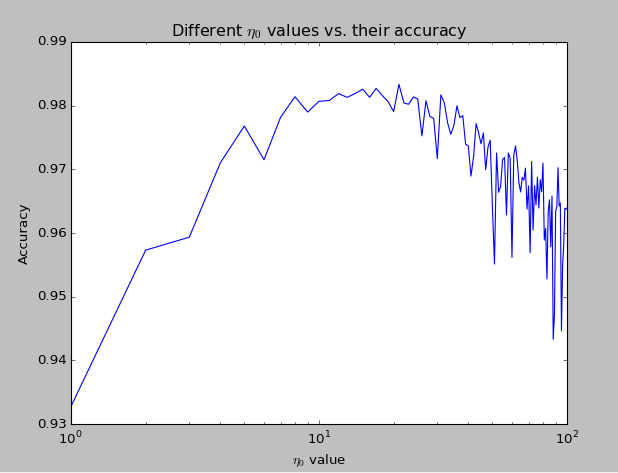
1. We can see that the accuracy of the validation test set increases as grows, until the optimum pick and then starts decreasing. On the other hand, the accuracy of the training test set increases almost monotonically as grows: in the smaller values of we get underfitting on the data until the pick of the validation, but for bigger values of the model overfits the training data.
2. The plotted ’s image is (see explanation at question 1, section b):



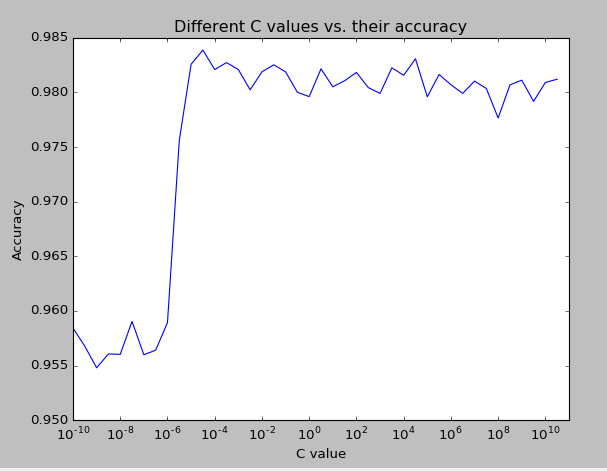
1. The accuracy of the linear SVM with the best C on the test set is .

**Question 3:**

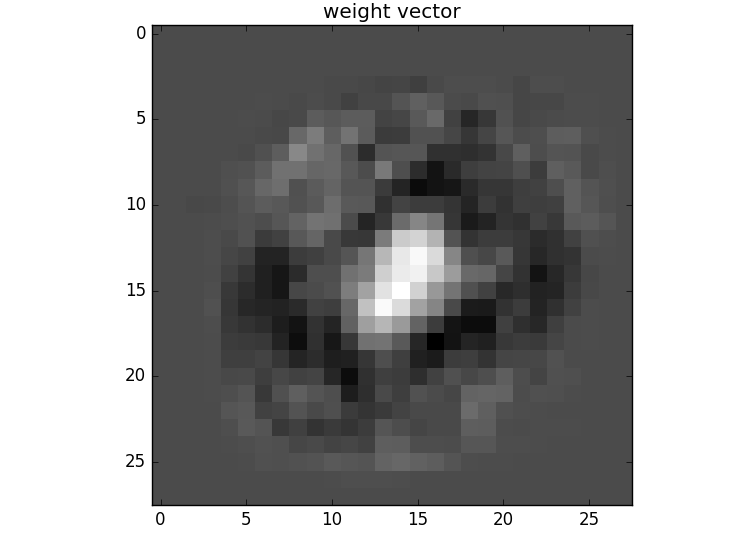
1. We found that gives the maximal accuracy on the validation set, which is .



1. We found that gives the maximal accuracy on the validation set, which is .



1. The plotted ’s image is (see explanation at question 1, section b):



1. The accuracy of the best classifier on the test set is .

**Question 4:**

We tried training the SVM classifier with the exponential kernel (looks almost identical to the code of question 2), and got even better accuracy. The exponential kernel is known of its best performance over handwriting digits’ images set.