Roi Herzig ID: 300360310 EMAIL:roeiherzig@mail.tau.ac.il

Moshe Raboh ID:300611878 EMAIL:shikorab@gmail.com

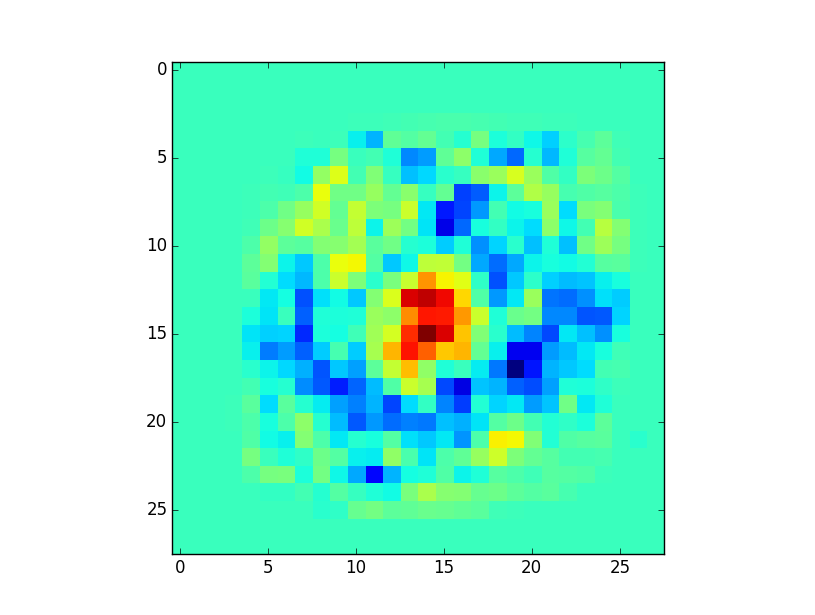
**Introduction To Machine Learning – EX 2**

1. **A:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_a" in file "q1.py"

**B:**

Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_b" in file "q1.py"

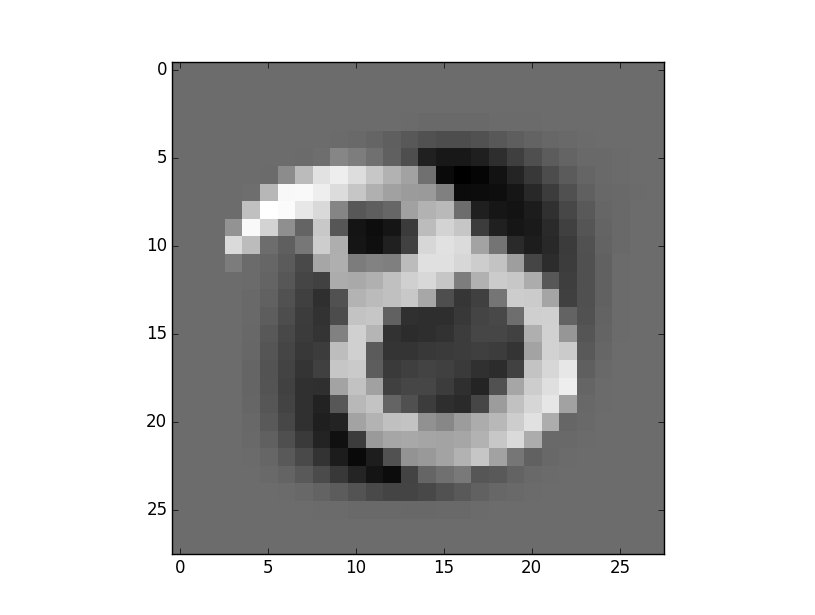
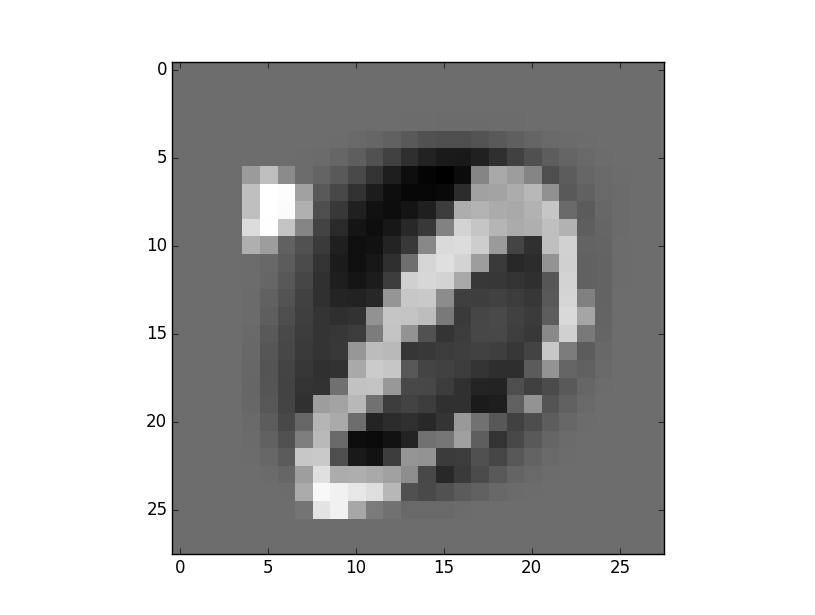
Name of the image: “part\_1b.png”



The image above is the weight matrix. As we can see, the weights that are related to the different features between 8 and the 0 will be the most significant while predicting a sample.

**C:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_c" in file "q1.py"  
Mean accuracy of the full train samples: 99.13%

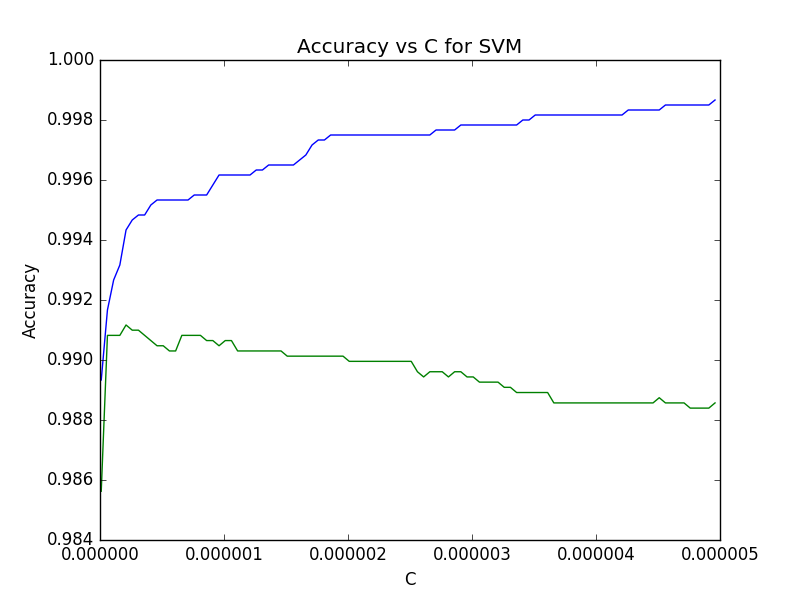
**D:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_d" in file "q1.py"Image file: "part\_1d\_rotat0.png" and “part\_1d\_rotat8.png”



In both cases, we can see images are miss-classified because the images are distorted. We can see that our classifier is not invariant to some affine transformation such as scale, translation and rotation.

**2.**

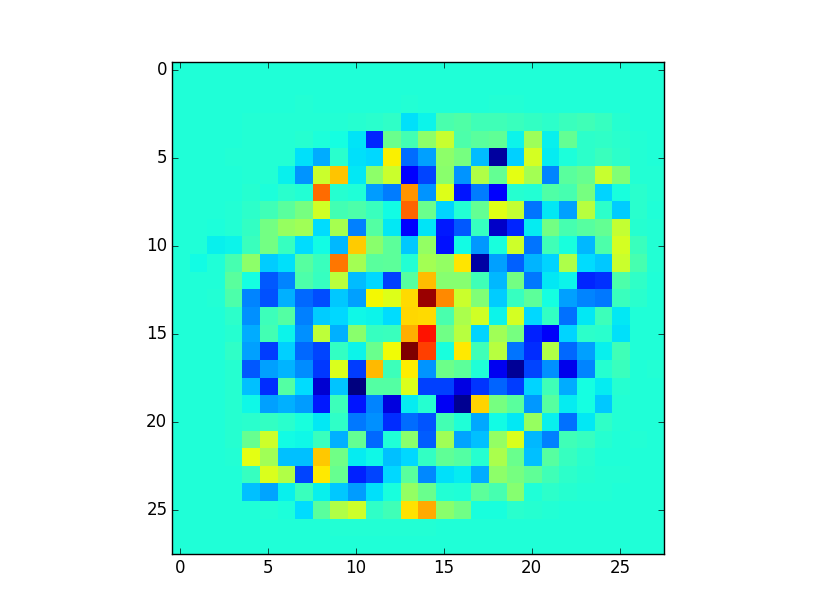
**A:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"Module main in file "q2.py"

The image file name: "q2\_part\_a.png"  


Blue line is the accuracy on the validation set and green line is the accuracy over the training set.

**B:**  
When c is large, we try to fit as close as possible to the training data with a risk overfitting,  
We can see the overfitting in the graph, for large c the training accuracy getting smaller while the validation accuracy getting larger.   
When c is small, there is no penalty for misclassification, so we might underfitting, and will get small accuracy in both validation and training.

**C:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"Function main in file "q2.py" (done in main module)

****

**D:**

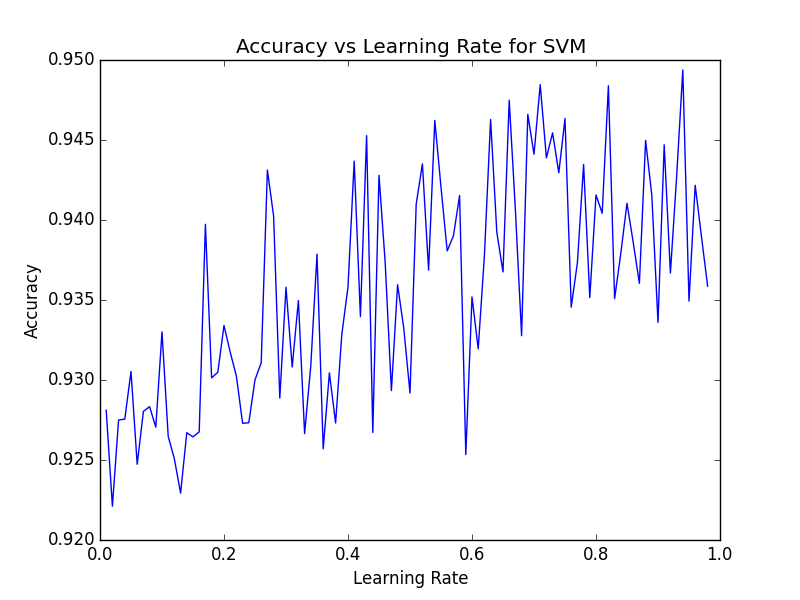
Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"Function main in file "q2.py"

(also done in module main)

The best c is 2.09999996059e-07 for error: 0.991167301697

**3.**

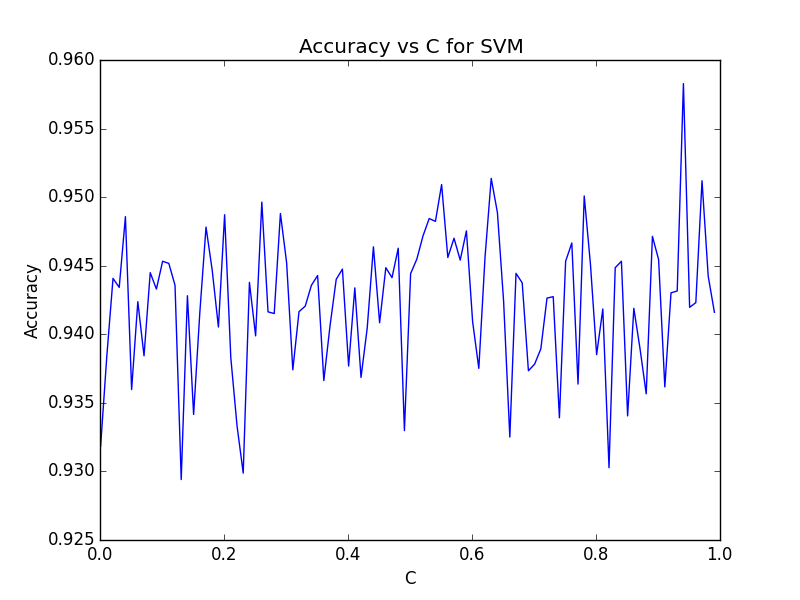
**A:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_a" in file "q3.py"

The image file name: "q3\_part\_a.png"  


We performed a grid search to find the best learning rate while T is fixed to 1000 and C is fixed to 1. The best learning rate is 0.94

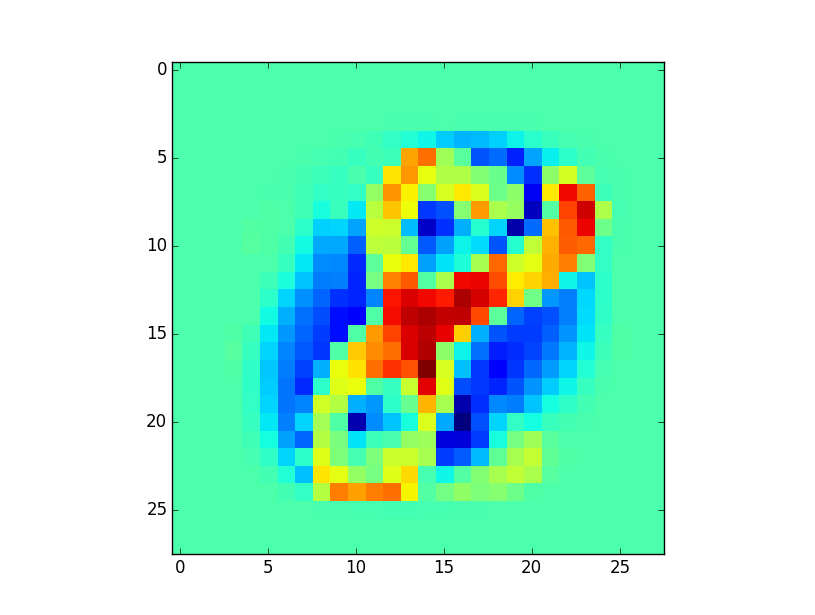
**B:**  
Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_b" in file "q3.py"

The image file name: "q3\_part\_b.png"



We performed a grid search to find the best C while T is fixed to 1000 and learning\_rate is fixed to 0.94. The best C is 0.94.

**C:**Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_c" in file "q3.py"

The image file name: "q3\_part\_c.png"

Using the best C and learning rate from previous sections, we trained our SGD SVM classifier with T=1000. The image above is the weight matrix. As we can see, the weights that are related to the different features between 8 and the 0 will be the most significant while predicting a sample.

**D:**

Directory: " /specific/a/home/cc/students/csguests/roeiherzig/ML/EX2"See function "part\_d" in file "q3.py"

The accuracy of the best classifier on the test set is 0.97