

## HW Submission Guidelines:

## General Instructions

1. Write the theory and programming assignments on different pages. Indicate at the top your names and e-mails. (Assignments can be done in pairs.)
2. You should email a **single pdf file** solution of the HW to ml.intro.2013@gmail.com. In addition, the printed submission should be placed in the mailbox of the TA (Number 399, Schriber second floor).

## Theoretical Assignments

1. Follow the exercise specifications exactly. Seek clarifications if needed.
2. Your proofs should be both rigorous and clear.

## Programming Assignments

1. Follow the exercise specifications exactly.
2. Your code should run properly on nova.cs.tau.ac.il: Your Matlab code should run using /usr/local/bin/matlab. (Test it before submitting it!)
3. For each programming assignment: In the printed submitted material indicate the path to the directory from which the program is executed. Make sure that the directory and files are read accessible by everyone (the output location of created/modified files should be a parameter to the program). The directory should contain a README file that has the student(s) name(s) and following detailed:
  - (a) A description of each relevant source/input file
  - (b) The ONE command to execute and its parameters if any. This command should also set any required environment.
  - (c) A description of each relevant output file
4. Your program should not output warnings or errors when run correctly with warnings on.
5. Make sure your program remains viable if your submitted files are moved (for example, don't refer to files you do not submit, and paths that are not necessarily accessible to others).

6. Your code must be clear. Use meaningful variable names and sufficient comments to achieve this.
7. Your plots should be clear. Set meaningful labels and ranges on the axes, as well as meaningful titles and legends. The axis, xlabel, ylabel, title, legend, plot and subplot Matlab commands allow you to do that comfortably.
8. It is recommended that you use Matlab vector and matrix operations instead of for loops with many iterations. This makes for faster, more compact programs. Use I/O, string and set operations economically to decrease run time.