

Homework 1 Instructions

Yaniv Benny and Tal Shaharabany

The homework assignment is to be done in groups of up to 3 students. Download the zip file provided (assignment1.zip) and unzip it.

1 Setup

Follow the following setup instructions:

1.1 Conda

We recommend using Conda as the python distributor and package manager. To install Conda: Download from <https://conda.io/miniconda.html> (for Python 3) and follow instruction on: <https://conda.io/docs/user-guide/install/index.html>.

The homework assignment comes with an `environment.yml` file which defines what third-party libraries we depend on. Conda will use this file to create a virtual environment for you. This virtual environment includes python and all other packages and tools we specified, separated from any preexisting python installation you may have.

After installing conda, create your environment called `hw1_env`, by typing from the assignment folder (In Windows, make sure you use "Anaconda Prompt" and not the "Command Prompt"):

```
conda env create -f environment.yml
```

When working on Windows, you may encounter an error with the above installation. In that case, try the alternative installation:

```
conda remove -n hw1_env --all
conda env create -f environment_win.yml
```

Both methods will install all necessary third party packages in the environment `hw1_env`.

Next, activate the environment by typing:

```
conda activate hw1_env
```

This homework assignment is an adaptation of the one from the cs236605 course: <https://vistalab-technion.github.io/cs236605/>

Activating an environment simply means that the path to its python binaries (and packages) is placed at the beginning of your \$PATH shell variable. Therefore, running programs installed into the conda env (e.g. python) will run the version from the env since it appears in the \$PATH before any other installed version.

After you finish the homework, deactivate the environment by typing:

```
conda deactivate
```

You can read more about conda here: <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

1.2 Jupyter Lab

To do the assignment, we recommend using Jupyter Lab. After activating the conda environment, inside the assignment folder, run:

```
jupyter lab
```

This will initialize a page in your browser where you can edit and run the code. Jupyter will be automatically installed when you create the conda environment with the provided environment.yml file.

1.3 Pycharm

Alternatively, it is also recommended to use PyCharm to write the code, though you can use any other IDE or software you wish. Using your university email, you can download a professional version for free here: <https://www.jetbrains.com/student/>.

2 Assignment

2.1 Folder Structure

After unzipping assignment.zip, you'll have an assignment folder which consists of the following:

1. hw1 folder: Python package containing the assignment code. **All** your solutions will be implemented here.
2. tests folder: A package containing tests that run all the assignment notebooks and fail if there are errors.
3. PartN_XYZ.ipynb where N is a number and XYZ is some name: A set of jupyter notebooks that contain the instructions that will guide you through the assignment. You won't need to edit these except if you wish to play around and to write your name at the beginning.
4. main.py: A script providing some utilities via a CLI. Mainly, you'll run it to create your submission after completing the assignment.

5. `environment.yml`: A file for conda, as described above.

2.2 Working with the Assignment

After activating conda (see above), then from the assignment folder, type:

```
jupyter lab
```

Follow the notebooks instructions starting from notebook 0. You are only required to change the code in the folder `hw1`.

1. The notebooks contain material you need to know, instructions about what to do and also code blocks that will test and visualize your implementations.
2. Within the notebooks, anything you need to do is marked with a `TODO` beside it. It will explain what to implement and in which file.
3. Within the assignment code package, all locations where you need to write code are marked with a special marker (`YOUR CODE`). Additionally, implementation guidelines, technical details and hints are in some cases provided in a comment above.
4. Sometimes there are open questions to answer. Your answers should also be written within the assignment package, not within the notebook itself. The notebook will specify where to write each answer.

Notes:

1. You should think of the code blocks in the notebooks as tests. They test your solutions and they will fail if something is wrong. As such, if you implement everything and the notebook runs without error, you can be confident about your solution.
2. Please don't put other files in the assignment directory. If you do, they will be added to your submission which is automatically generated from the contents of the assignment folder.
3. Always make sure the active conda env is `hw1_env`. If you get strange errors or broken import statements, this is probably the reason. Note that if you close your terminal session you will need to re-activate since conda will use its default base environment.

2.3 Submitting your work

To generate your submission, run:

```
python main.py prepare-submission --id ID1 --id ID2 --id ID3
```

where ID1, ID2 and ID3 are your id numbers.

If there are errors when running your notebooks, it means there's a problem with your solution or that you forgot to implement something. Additionally, you can use the `--skip-run` flag to skip running your notebooks (and just merge them) in case you already ran everything and you're sure that all outputs are present:

```
python main.py prepare-submission --skip-run --id ...
```

If you are unable to solve the entire assignment and wish to submit a partial solution you can create a submission with errors by adding an `allow-errors` flag, like so:

```
python main.py prepare-submission --allow-errors --id ...
```

Finally zip the output and submit it on Moodle.