Homework

Goals:

1. Introduction to Pytorch.
2. Introduction to neural network training.
3. How to work with the school's GPUs.

Autoencoder:

As explained in class, the Autoencoder contains two networks, E and D. E encode the input into latent space and D decode the latent representation to an image. Loss function is $|D(E(x)) - x|$, which means the output of the networks should be similar to the input. For more details, check out the slides of the first class or the web.

Pytorch tutorials:

1. Read https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
2. Read https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

Task:

Implement an Autoencoder.

1. The Autoencoder should be implemented in Pytorch and run over the GPUs using CUDA. You must read the Pytorch tutorials.
2. Use the CelebA dataset.
3. Image size should be 128X128 (Input and output).
4. Number of layers should be at least 8.
5. Latent representation size should be no more than 256 (flatten).
6. Loss function should be L1 or L2 (MSE).
7. Use batch size larger than 2.
8. Split the data to train/test (test should be at least 5%).

Submission:

1. Python file train.py contains all the code and PDF file contains all the visual results.
2. Visual results include pairs of original input and its output, you should show at least 16 pairs for each hyperparameters.
3. You have to show visual results of at least 4 different hyperparametrs.
4. The team with the best visual results will get 10 points bonus, second best will get 5 points.
5. Code should be readable and clean.

Tips:

1. Output should be similar to input, but far from being identical, think about why we won't get identical output.
2. Plot loss as function of the iteration, try to find better hyperparameters.
3. Recommended: Using BatchNormalization and LeakyReLU, Adam for optimization.

4. There are several ways to use CUDA, for one option check https://github.com/Natsu6767/DCGAN-PyTorch/blob/master/train.py.
5. There are several ways to choose your device, one of them is CUDA_VISIBLE_DEVICES=1 At the beginning of your command.

Work over school's GPUs – relevan also for final projects! :

1. Login to nova.cs.tau.ac.il, see http://cs.tau.ac.il/system/servers14 and http://cs.tau.ac.il/system/ssh_servers for more details (note that you need to use VPN when login from home).
2. Each team has storage in **`/home/dcor/ronmokady/workshop21`**, I already created folder for each team with the names team1/2/3/4. Teams are:

dir	Name
team1	שלו דרוקמן
	ברק מרדכי
	תומר יואלי
team2	נופר מרדכי
	'דניאל רויץ
	מורן דאורי
team3	אור טולדנו
	יניר מרמור
	דב גרץ
team4	בן לאמי
	רבקה יעל רקח
	קרן שאנן

3. **It's your responsibility to make sure you don't use more than 50GB**, to check folder size use "du -sh ./folder"
4. Connect to one of the servers:

rack-gamir-g07.cs.tau.ac.il
rack-gamir-g06.cs.tau.ac.il
rack-gamir-g05.cs.tau.ac.il
rack-gamir-g04.cs.tau.ac.il
rack-gamir-g03.cs.tau.ac.il

5. If there is technical issues with the servers (e.g. one of them is down), please send a message to the system: system@cs.tau.ac.il.
6. You can check if there is available GPUs using "nvidia-smi"
7. Please use at most 2 GPUs at the same time.
8. Note that all the servers use the same file system as nova, i.e. you can access your storage form each one of the servers.
9. You should create virtual env using **conda**:
   a. Get the download link from https://www.anaconda.com/distribution/ (linux)
   b. Use wget command to download into the server

c. Run the package using bash
d. Place the env in your team's storage
e. To initiate conda use bash or ~/.bashrc to reopen the terminal
f. Create environement using: conda create -n myenv python=3.7
g. Activate using: conda activate
h. Install required packages (search over the web, for example: conda install -c anaconda pillow)

10. Create session using tmux (so training will continue even when you log out), see https://gist.github.com/MohamedAlaa/2961058
11. CelebA dataset can be found at: **`/home/dcor/ronmokady/workshop21/celebA`**