

Code Files:

For this exercise we broke down the code into multiple files:

- ex2.py - here is the code for the GAN training the AE (including the novel case)
- ex2_2.py - here we implemented the inverse GAN, GAN interpolation, AE novel tester ,AE interpolation
- models.py - here we have the different models we tried for the MNIST and celebs dataset and for the auto encoder. In addition, we added here our implementation for the Non-Saturated Cross Entropy loss function.
Last but not least, in this file we implemented a class to represent an enum in order to choose dynamically the type of Models we would like to train. (See argos -d and -g in the "Running the code" section)
- common.py - here we have code that is shared for ex2.py and ex2_2.py
- utils.py - Other utility functions

Running the code:

- ex2.py: to run the GAN training, run 'python ex2.py'. There are many possible arguments that can be given to the script. We urge you to run 'python ex2.py --help' to see the possible arguments. In particular, the most important flags are:
 - --dataset: defines if to train on the MNIST or Celeb dataset
 - -d: this enables one to choose the model for the discriminator. As can be seen in models.py, there are around 18 different models for the Celebs dataset and 5 for the MNIST data set. The syntax is by -d M<n> where n is the number of the model, e.g. -d M3.
 - -g: same as -d but for the generator.
 - -l (lowercase L): defines the loss function for the training. Legal arguments are:
 - "mse"
 - "cross_entropy"
 - "non_saturated"
 - -k: the number of unrolling iterations to use
 - -ns: the number of iterations that the generator is trained over a single batch

In case there were changes it would indicate it here

Today

Last time we edited the pdf