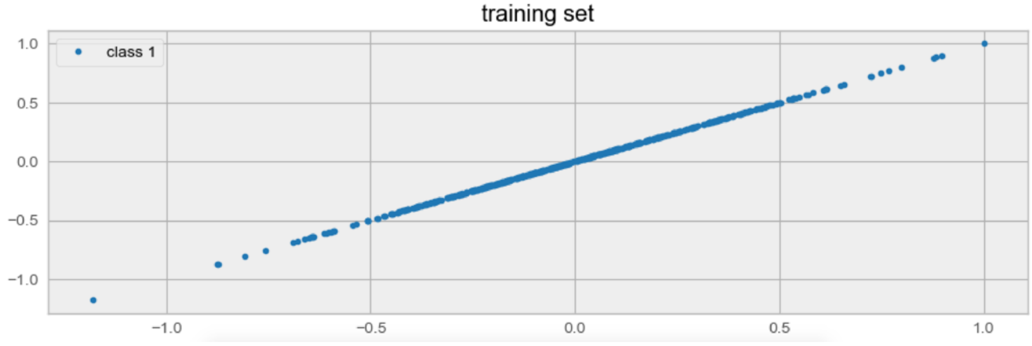
Advanced Techniques in Machine Learning

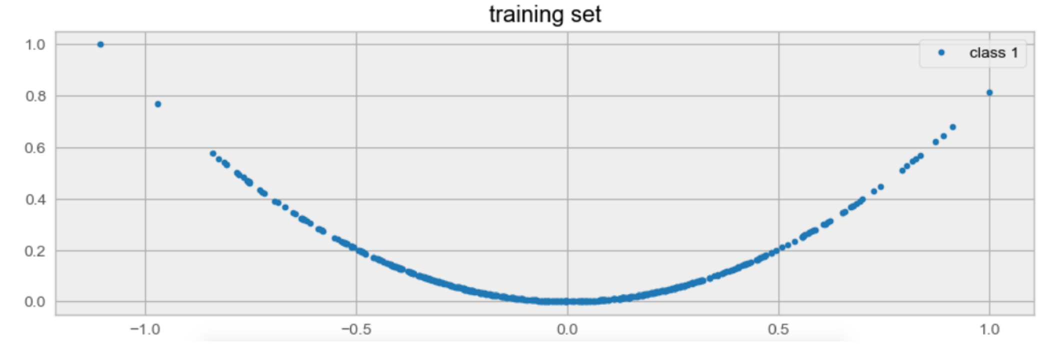
Exercise 1
Due Date: 11/06/2021
Yossi Keshet, Roni Chernyak

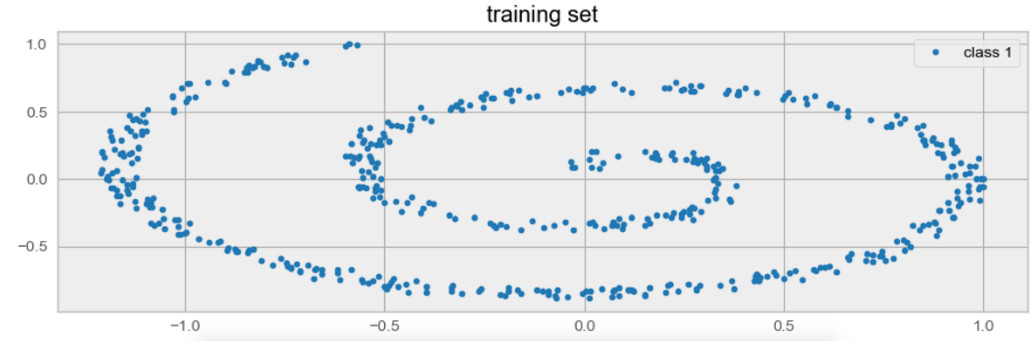
In this exercise, you will implement and analyze Generative Adversarial Networks (GANs) in PyTorch. Recall, GANs are very powerful models that can match distributions. In this exercise, you will train a GAN to map data from a uniform distribution to some unknown target distribution. You should report results for all three settings.

Overall you should generate data from a uniform distribution and train three separate GANs, each for one of the three target distributions.
You can train your model for how many iterations you want, but it should converge after roughly 60,000-80,000 iterations, depend on the target distribution (training time is a few minutes on standard CPU).

Data:
You are provided with a python function to generate synthetic data for three types of distributions: Line, Parabola and Spiral. The data is 2-Dimensional.


training set


training set


training set

Models:

1. You can implement whatever GAN version you feel comfortable with; however, the most basic version should work for you.
2. You can implement whatever architecture you want for the generator and discriminator. Recall, that the data is 2-Dimensional.

Implementation hints:

1. Try to use different learning rates for the generator and discriminator if you feel one of them is takingcontrol over the game.
2. Try to use different optimizers.
3. Try to schedule the generator, update gradients every k iterations.
4. Try different activation functions and different model architectures.
5. Try to switch the discriminator loss from –log(1 – D(G(x))) to log(D(G(x)))

What to submit?

1. Three plots, one for each model, where you sample 1000 points and run them through the generator, and 1000 samples from the target distribution.
2. A brief PDF report with your conclusion from the experiments you run; what is your model architecture? What works best for you? What problems did you encounter during optimizing you model?
3. Python3.6 code of the project.

Online code:

You can find plenty of GAN implementations over the web. You can use them if you want to; however, none of them will perfectly work for you, so you will probably need to make code adjustments, and this can take a while, especially since you do not have GPU. My advice is to implement it by yourself, it will be the fastest way to make it work, and you will actually learn the difficulties of GAN training.