Arie Cattan 336319314
Moche Uzan F957022

Advanced Techniques in Machine Learning
Assignment 3 — GANs

At the beginning, we implemented the Discriminator and the Generator as very simple models with only one hidden layer, we it worked only for the line distribution.
Then, we tried several learning rate, optimizers, adding layers, Dropout, activate functions, loss functions, noise dimension… until we got best results.

Instead of updating the discriminator k times and the generator one time (like proposed in the GAN algorithm), we used two distinct learning rate, such as the bigger was assigned to the Discriminator.

We also tried multiple combinations of optimizers, and get best performance with **RMSProp** for both Discriminator and Generator.

In our case, we don't need Dropout because at each iteration, we sample 100 **new** samples, the model cannot overfit on the training set.

When we used tanh as activate function on the output layer in the generator, it learns to generate points only in a small interval, such as between -0.2 and 0.4 and not all the possible points between -1 and 1.

Discriminator:

3 hidden layers:
-Linear Layer 2*16, Leaky-relu(0,2)
-Linear Layer 16*16, Leaky-relu(0,2)
-Linear Layer 16*2, Leaky-relu(0,2)

Output Layer:
-2*2, softmax

The loss function is cross entropy loss.
Learning rate: 0.002

Generator:

Noise dimension: 2

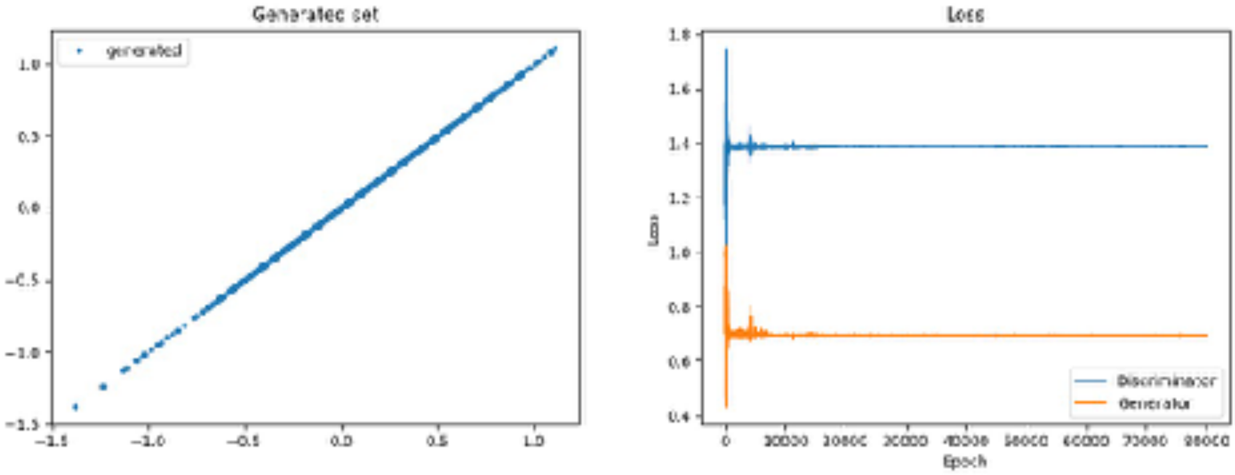
2 hidden layers:
-Linear Layer 2*16, Leaky-relu(0,2)
-Linear Layer 16*16, Leaky-relu(0,2)

Output Layer:
-16*2, without activate function

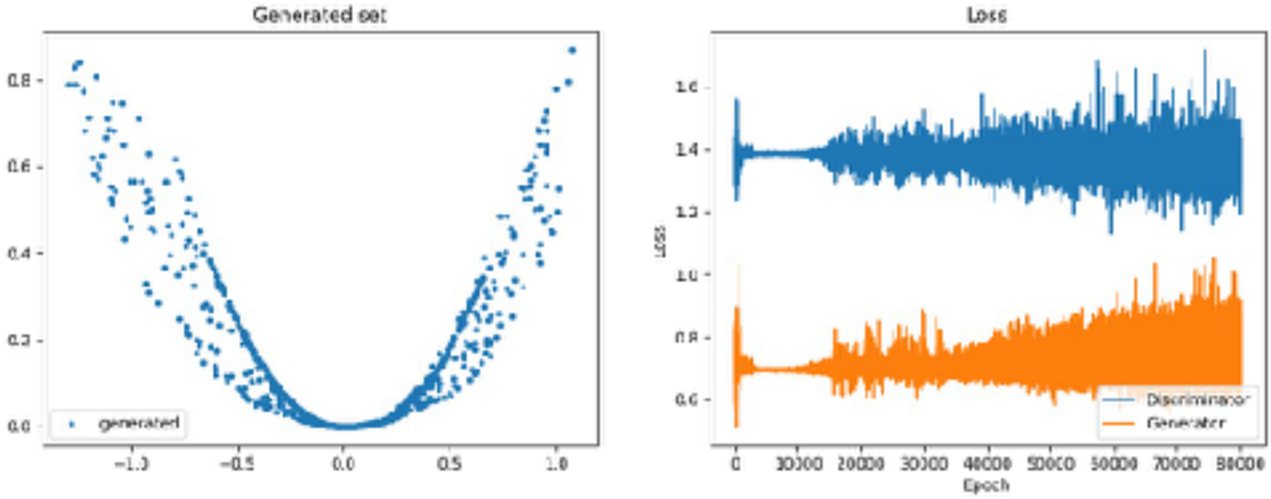
The loss function is cross entropy loss.
Learning rate: 0.001

Arie Cattan 336319314
Moche Uzan F957022

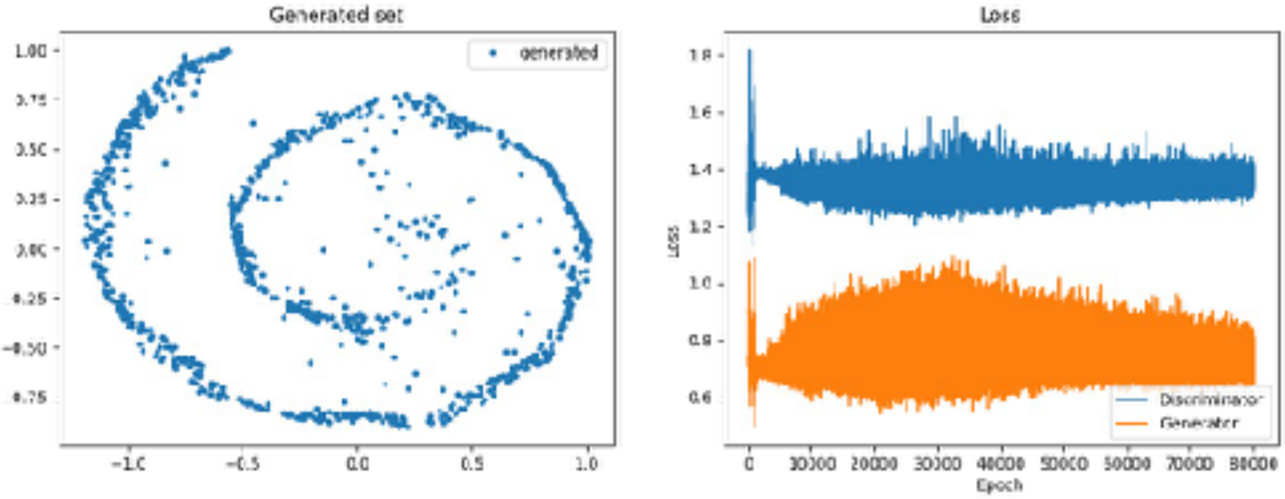
Line:



Parabola:



Spiral:


Generated set
generated
Loss
Loss
Discriminator
Generator
Epoch