Write up

Intro:
Our model is based on:
https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f

Because this code trains on pictures (get 2D tensor and flatten it to 1D array), we tried to take our line/parabula/spiral and transform it to picture and then train the model. After a few trials we understood that it's not the way and we lose information by transforming it to picture (it's hard to make from picture 1000 points that distributed uniformly).

After unsuccessful results with the first model we changed our mind set and moved to learning each point separately (now we are learning them in batches of 1000, each one is inserted separately to the neural nets).

How does it work?

Like every GAN our model contains the discriminator and generator, each one a neural net with some hidden layers (1-2 layers, depends on the model-par/line/spiral)
For both we used adam optimizer with learning rate 0.0002-0.0003, both use BCELoss.

Discriminator:
Input: point (x,y)
Output: whether the point is real or fake(generated by generator)

The discriminator goal is to correctly identify the fake and real points → maximize the probability that real data is classified as real P(D(x)) ,While minimize the probability That fake data is classified as real P(D(G(z)).

How we train it? We train it using both the real and fake data
(see train_discriminator function for more)

Cont next page

Itay Hassid 209127596 Aviv Shisman 206558157

Generator:

Input: noise z in form of point
Output: fake point

The generator goal is to correctly create fake data without the discriminator noticing→ maximize the probability that fake data is classified as real P(D(G(z)))

How we train it? By the output of the discriminator on the fake data it created (see train_generator function for more)

Model 1 - line (ex3_line.py)

Model: the same model as described above in the intro
Details :
The discriminator net have 1 hidden layers in → input in size 2 and output size is 80
We used leakyRelu as activation function (was the best for us) and used dropout for reduce overfitting, after the hidden layer the output of discriminator is binary → is it fake or real.

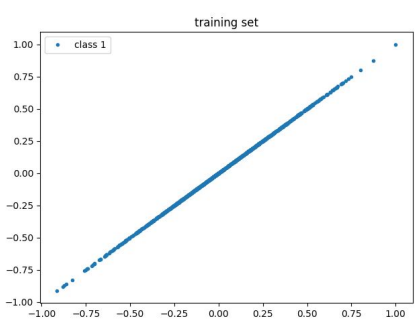
The generator net also have 1 hidden layers in → input in size 2 and output size is 10
We used leakyRelu as activation function, it output is a fake point.(uses tanh at output)

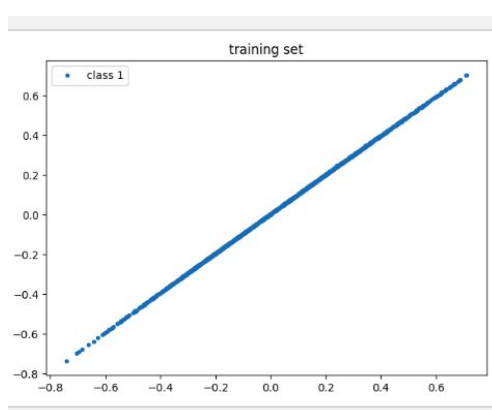
Difficulties/problems we had ?
We did not encounter any difficulties in this model as it is the simplest one,
Model 2 and 3 were the more challenging ones.

Results:(look at line/res for more results)

Our output

Your output(from generate func)





Itay Hassid 209127596 Aviv Shisman 206558157

Model 2 - parabula

Model: the same model as described above in the intro
Details: need to complete
Both of the nets have 3 hidden layers in →
Layer 1 input is size 2 output is size 4
Layer 2 input is size 4 output is size 8
Layer 3 input is size 8 output is size 16
Each one of them has only the activation function with low value(that was the best)

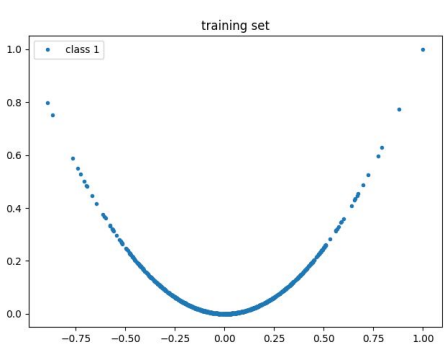
we found out that when we overfit the dots were concentrated in lowest point of the parabola or in the sides of the graph therefore we needed to balance our parameters to get the results.

Difficulties/problems we had ?
Mostly the first approach we discussed on the intro after some tries we understood that although the output looked great(we displayed the image) it was difficult to extract the points from it without information loss.

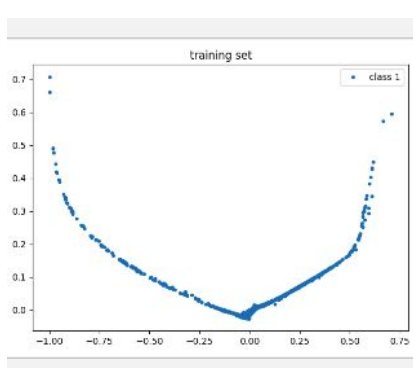
After we switched to by point models, the difficulty was to tune the parameters to get the right results, we needed to balance between overfit where the dots are spread in the sides or concentrated in the lowest point of the parabula to underfit where the dots don't have concrete structure.

We tried many combinations to get the right results but at the end it still was not perfect as you can see the lowest part of the parabula is a bit pointy, we tried many times to get the same curvy shape like the parabula but it was not perfect as you see. (please do not reduce us many points for it as we tried a lot of solutions and did not have much time -- we had tests…)
Results:(for more results look at par/res)

Your output



Our output



Itay Hassid 209127596 Aviv Shisman 206558157

Model 3 - spiral

Model: the same model as described above in the intro
Details: this model has 2 neural nets are:

The discriminator net have 3 hidden layers in →
Layer 1 input is size 2 output is size 64
Layer 2 input is size 64 output is size 32
Layer 3 input is size 32 output is size 16
Each one of them has only the activation function with low value(we found out that it was the best)
The output layer outputs a binary decision → fake or not. (activate sigmoid and return probability of being real)

The generator net also have the same layers in opposite order → we found that this 'hack' brings the best results (both have layers in size exp(2,i) , the discriminator ones are getting lower each layer and the generator are getting higher each layer)
His output layer uses tanh and returns a fake point.

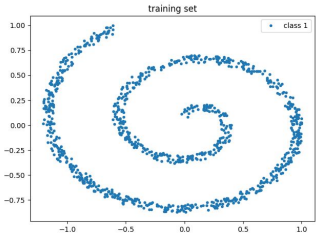
Because we have more layers we did less epochs here(250)

Difficulties/problems we had ?
That data was scattered(underfit) and after a lot of tuning we found out that we need to overfit to the examples, so we reduced the dropout and lowered the leakyRelu (it helped somehow) and added few layers for better learning of complex structures and it worked.

Another thing worth noticing is that our spiral tail is cut off we saw that when we had higher leakyRelu values the tail was not cut off but it was spread and there was a lot of noise in the graph, so we prefered this version.

Results: (look at spiral/res for more)

Your output



our output