GAN Assignment

QUESTION 1: What is the relationship between the input and X1 and X2?

The relationship is:

- **X1**: Random inputs in ([-0.5, 0.5]), representing (x).
- **X2**: Calculated as (X1^2), representing (y = x^2).

Together, (X1) and (X2) describe the quadratic function (y = x^2), forming points on a U-shaped curve.

QUESTION 2: What is He weight initialization?

He weight initialization sets weights to small random values optimized for layers using ReLU activation. It helps maintain consistent signal flow through the network, avoiding issues like vanishing or exploding gradients during training.

QUESTION 3: What is Adam?

Adam (Adaptive Moment Estimation) is an optimization algorithm used for training neural networks. It combines the benefits of two other methods:

- **Momentum**: Uses past gradients to smooth updates and accelerate convergence.
- **RMSProp**: Adapts learning rates for each parameter based on recent gradient magnitudes.

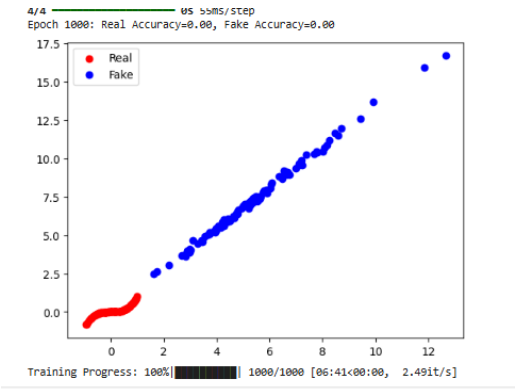
Adam adjusts the learning rate during training for faster and more efficient convergence, making it widely effective and popular for deep learning tasks.

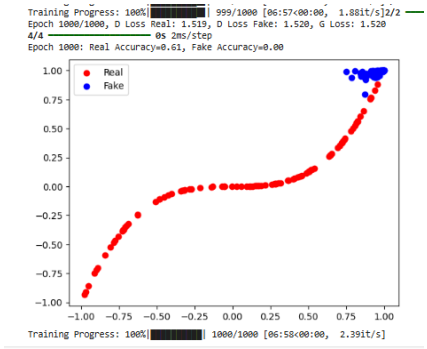
QUESTION 4: At this point in the exercise, why is the generator so poor at generating points in the distribution?

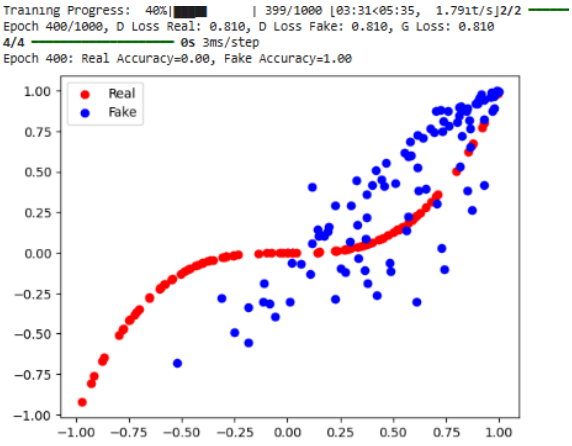
At this point, the generator is poor at generating points in the distribution because it has not been trained yet. The generator starts with random weights and does not "know" the target distribution or function. Training the generator through feedback from the discriminator is necessary for it to learn how to map points from the latent space to the desired output distribution.

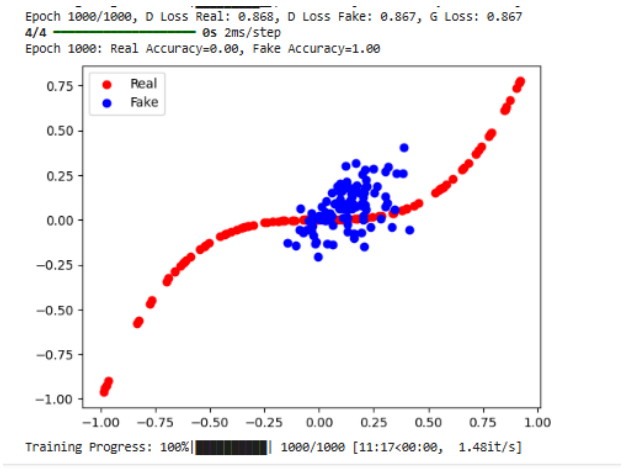
QUESTION 5: Repeat the above lab but this time use a different function. Submit the completed script along with a word/PDF document containing the answers to each of the lab questions.

We tried working with X^3 initially. We only trained up to 1000 epochs, tweaking different parts as we went. First, after 1000 epochs it was a linear function, not in the right range, then we got the generated in the right range but they bunched up in the corner. Then we got them more spread out, and finally they bunched up closer to the X^3 real points, at 1000 epochs. We even tried working with absolute value for a bit, but it is very time consuming with all of the training, and colab won't let us use the GPU anymore, so it has been going extra slow. The script we are submitting may not be the best version as we continued tweaking it.

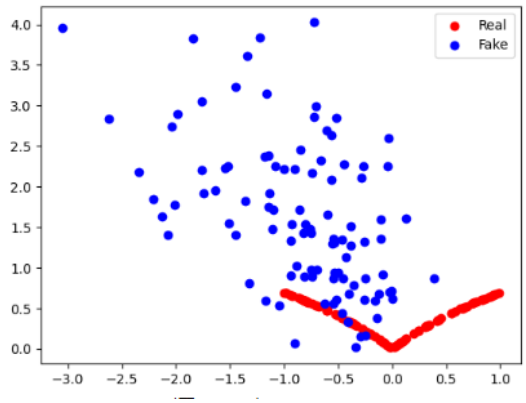








Training Progress: 20%| | 199/1000 [00:53<03:47, 3.52it/s]2/2
Epoch 200/1000, D Loss Real: 0.906, D Loss Fake: 0.909, G Loss: 0.909
4/4 0s 51ms/step
Epoch 200: Real Accuracy=0.53, Fake Accuracy=0.01



Training Progress: 20%| | 199/1000 [00:59<03:56, 3.38it/s]2/2
Epoch 200/1000, D Loss Real: 0.978, D Loss Fake: 0.979, G Loss: 0.979
4/4 0s 21ms/step
Epoch 200: Real Accuracy=0.53, Fake Accuracy=0.01