GAN implementation on MNIST

To understand how GANs are constructed and how the data flows in between the layers, I followed a tutorial to create new images using the MNIST dataset.

Image size. = 28 X 28 = 784.

As it is a greyscale image, just 1 color channel, so image size id 784.

Discriminator architecture –

Input – image (784)

Hidden layer1 – 1024 || LeakyRelu

Hidden layer2 – 512 || LeakyRelu

Hidden layer3 – 256 || LeakyRelu

Output layer – 1 || Sigmoid.

Output layer is 1 because discriminator just outputs the probability of the input image being a sample from the given input or from the generator.

Values close to 1 mean that the output is from the input sample. Values close to 0 mean output is from the generator.

Generator architecture –

Input – 100 (random sample space)

Hidden layer1 – 256 || LeakyRelu

Hidden layer2 – 512 || LeakyRelu

Hidden layer3 – 1024 || LeakyRelu

Output – 784 || tanh

Output is 784 because we want to generate a final image with the same dimensions as the image.

Chosen learning rate = 0.0002

Chosen loss function – Binary Cross Entropy loss

BCELoss = ∑li, li = -wi (yi log(vi) + (1-yi) log(1-vi))

y = targets

v = inputs

w = weights (considered 1)

For the discriminator network:

Should be “max” player – maximize loss

This can be compared to the BCE loss.

If yi = 1 and vi =D(xi)

BCELoss = - (log D(xi))

This gives us the loss for **Real Images**.

If yi = 0 and vi =D(G(zi))

BCELoss = - (log (1 - D(G(zi))))

This gives us the loss for **Fake Images**.

Hence, during the discriminator training, we are minimizing these 2 functions.

We calculate these 2 gradients and then add them.

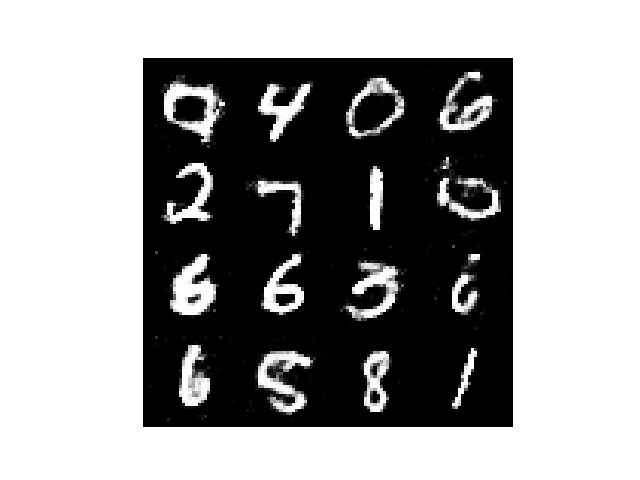
For generator:

If yi = 1 and vi =D(G(zi))

BCELoss = - (log D(G(zi)))

This gives us the loss for **new images.**

New generated images -



As we can see, we have generated images of numbers, which visually can be understandable.

Code by - https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f

Concerns / Points of discussions –

1. How do we check the quality / validity of the generated output?
2. Experiment with architectures based on what criteria?