Deep Learning & GAN

Week 6 - Session 2

Deep Convolutional Generative Adversarial Network (DC-GAN)

Following the original GAN architecture that you developed in the workshop, in this project you will develop a deep convolutional GAN (DCGAN) [3] to the MNIST data [2]. DCGAN is a variation of GAN that uses deep convolutional networks (layers) in the generator and discriminator networks to better capture the patterns in the image data. The steps for loading the dataset is similar to the GAN procedure. The structure of the generator G and the discriminator D are illustrated in the figure below. set.

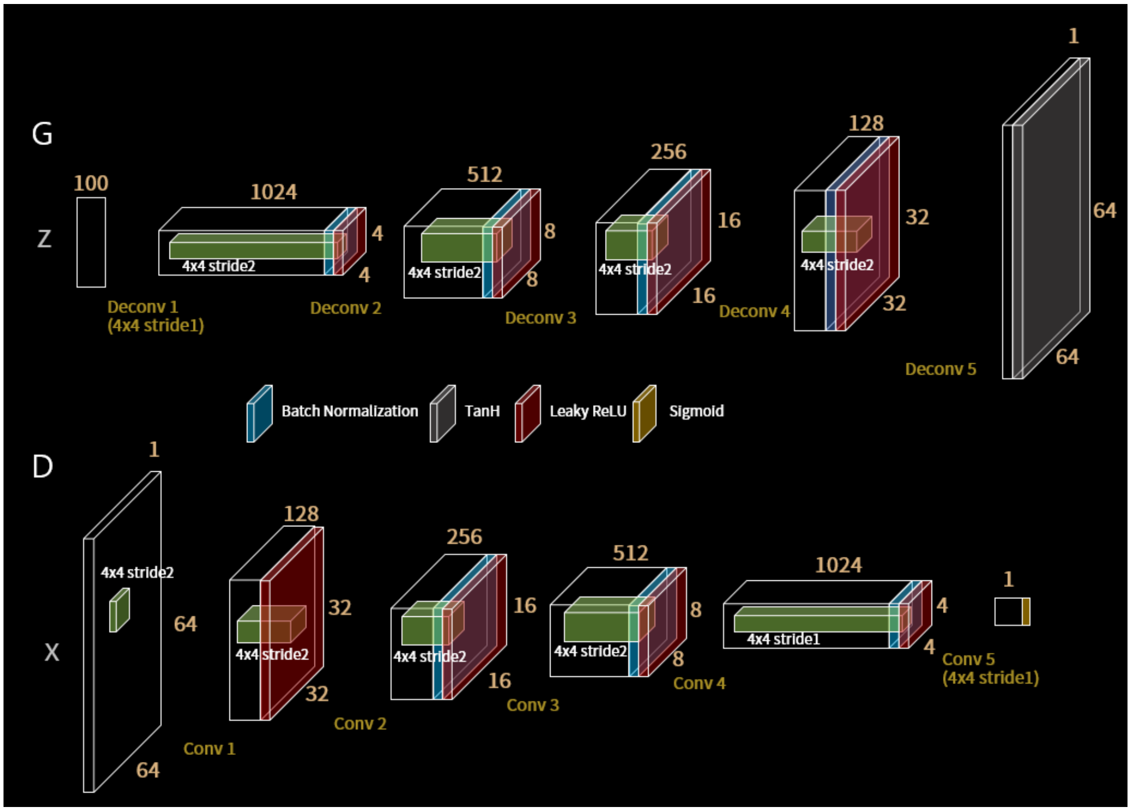

Figure 1: DCGAN internal structure

In the provided DCGAN_MNIST.py you will find the implementation of the DCGAN network and the training procedure. In each epoch, the discriminator and generator's weights are updated based on their defined loss functions and using Adam optimizer. Your task is to understand each part of the code and train this model on the MNIST training

Hyperparameters The following hyperparameters are set for this DCGAN structure:

- Batch size: 100
- Learning rate: 2×10^{-4}
- training epochs: 20
- Leaky ReLU: 0.2
- Adam Optimizer beta_1: 0.5
- weight initialization normal: mean - 0 and std - 0.02

Report After training the DCGAN model, report the following results from the experiment:

- The training error plot that represents generator and discriminator loss trace.
- The fake image generation improvement as a .gif file and your observation by comparing it against the GAN output.
- Compare the performance of DCGAN with the vanilla GAN. Which model generates more realistic images?
- (Optional) Experiment DCGAN with different hyperparameters and different latent space dimen-sion.