

Generative Adversarial Network (GAN) Workshop

Week 6 - Session 2

In this project, you will build a generative adversarial network (GAN) [1] on the MNIST dataset [2]. From this, you will be able to generate new handwritten digits similar to the original ones. The idea behind GANs is that you have two networks, a generator G and a discriminator D , competing against each other. What ends up happening is that the generator learns to make data that is indistinguishable from real data to the discriminator. For this project, you will need to setup the python 3 environment and install the tensorflow 2.0.0 packages.

Dataset You can download the MNIST dataset from <http://yann.lecun.com/exdb/mnist/>. Store train/test data and labels in a folder called `MNIST data`. You are provided with `input_data.py` script that extracts and loads each train/validation/test data and labels.

Model The structure of generator G and discriminator D are illustrated in Figure 1 below. Each G and D are simple feed forward neural networks with varying number of hidden units in each layer. In the provided `GAN MNIST.py` you will find the implementation of the whole GAN 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 build the inner structure of G and D networks based on Figure ?? and understand the training procedure of GAN on the MNIST training set by following the losses produced at each iteration.

Hyperparameters The following parameters are set for this GAN structure:

- Batch size: 100
- Learning rate: 2×10^{-4}
- training epochs: 100
- Dropout: 0.3

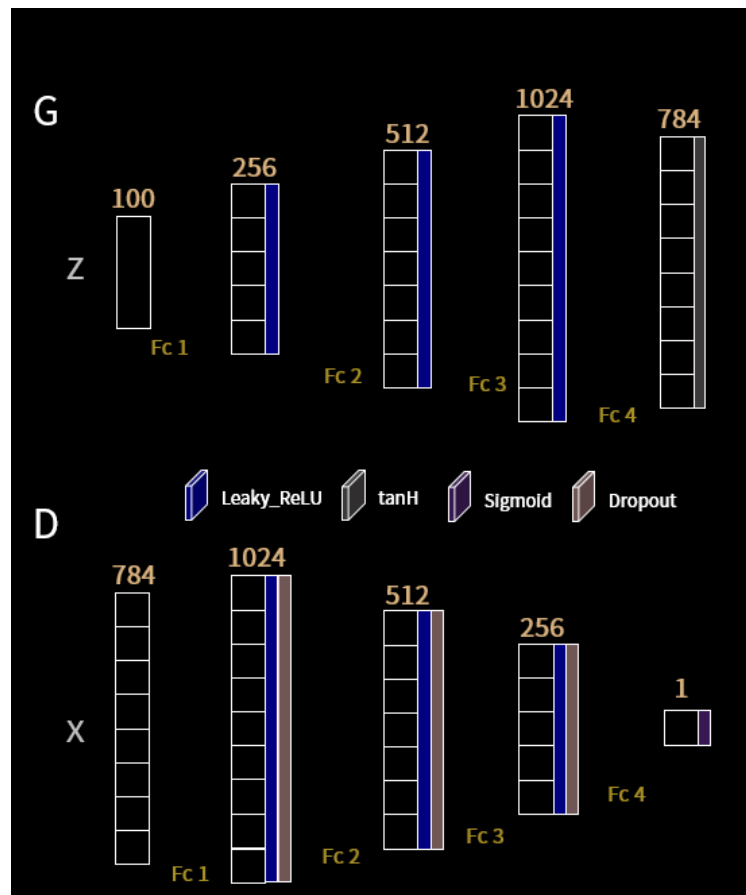


Figure 1: GAN internal structure

Results After training the GAN model, the program will store the outcomes in the `MNIST_GAN_results` folder. In your report, include the following:

- The training error plot and describe the behavior of G and D networks while training.
- The fake image generation improvement as a `.gif` file and your observation.
- (Optional) Try altering the hyperparameters and your observation of changes in the quality of generated fake images.

References

- [1] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.
- [2] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.