Generative Adversarial Networks in TMVA

MENTORS

Manos Stergiadis
Gerardo Gutiérrez
Sergei Gleyzer

PRESENTED BY

Ashish Kshirsagar

INTRODUCTION

- **Generative Adversarial Networks** comprises of two models trained simultaneously by an adversarial process. A **generator** learns to create images that looks real i.e. similar to the one in the dataset, while a **discriminator** learns to classify whether the input image is real or fake (output by the generator)
- **Generator :** Inputs random noisy image and gives an output image resembling that in the input data distribution.
- **Discriminator :** Inputs an image and tell whether that is from the dataset or from the generator

CURRENT IMPLEMENTATION

- Input data formatting script to convert image into TTree format.
- MethodGAN class contains the implementation of Generator and Discriminator.
- Architecture passed using input string (TString) format.
- Tests passed but the function not aggregating.

DELIVERABLES

- MethodGAN.h, MethodGAN.cxx
  - Discriminator Architecture :
    - Input Layer
    - Convolution Layer, RELU, Avg. Pooling Layer
    - Dense Layer, Sigmoid
  - Generator Architecture :
    - Input layer
    - Dense Layer
    - Transpose Convolutional Layers, RELU
    - Output Layer, tanh
  - Generalised GAN with all the parameters as input including the layers details.
  - Specified training parameters

ISSUES

- Modifications done in MethodDL.h
- The input transformation script needed to be integrated
- Building the tests
- Testing the Convolution Layers
- Implementation of CGAN

TASKS

- Testing the convolution layer
- Implementation of Vanilla version on MNIST dataset
- Implementation of DCGANs
- Implementation on generalised dataset
- Implementation of ConditionalGANs