Generative Adversarial Networks in TMVA

MENTORS

Manos Stergiadis Gerardo Gutiérrez Sergei Gleyzer

> PRESENTED BY Ashish Kshirsagar

High Level Overview of GANs

- MiniMax game : Alternate between training discriminator and generator
- **Generator :** Train to generate samples that fool the discriminator
- **Discriminator**: Train a classifier to distinguish between the two distributions using samples



General Structure of GANs



Upsample Layer

- No weights/ filter.
- Used to increase the dimensions of input.
- Need: Inverse operation of a pooling layer in a Convolutional layer. Layer to translate from coarse salient features to a more dense and detailed output.

Details

- Nearest Neighbor Interpolation
- Allows batch input (Vector of TMatrix)
- Outputs TMatrix with higher resolution

10	4	22	
2	18	7	
9	14	25	

10	10	4	4	22	22
10	10	4	4	22	22
2	2	18	18	7	7
2	2	18	18	7	7
9	9	14	14	25	25
9	9	14	14	25	25

Tests & Results

- Implemented forward and backward passes Reference, CPU and Cuda Architectures.
- Successfully passed sample tests for batch input for both forward and backward propagation.

Transpose Convolution Layer

- Transposed convolution operation is similar as the normal convolution but in the backward direction.
- Allow batch input and generates corresponding upsampled matrix.
- Implemented forward and backward passes for CPU Architecture.
- Tests successfully passes for both passes.

Implementation Details





Generating Output Matrix

Architecture for GANs in TMVA



MethodGAN

- Base Class: MethodBase
- Friend Class: MethodDL



Tasks

- Implement GANs with fixed architecture
 - Resolve error for saving weights
 - Ensure convergence of GANs