Image Synthesis with GAN

Nilakshi Pokharkar

Image Synthesis

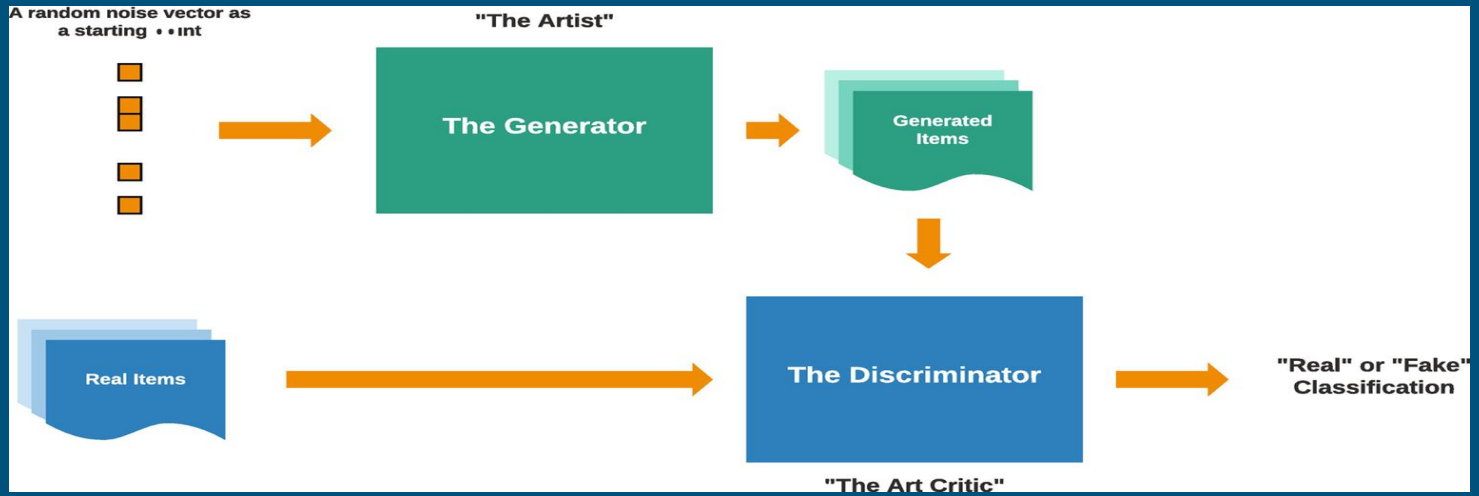
- Process of artificially generating images
  - Particular desired content
- Simplified - Creating new image from
  - Some form of image description
- Modify or enhance existing images
- Goal - Create realistic and visually appealing images
- Application
  - Advertisements
  - Medical Imaging
  - Video Game Development

Datasets

- MNIST
  - Training and testing data
  - 28*28 px, grayscale images
  - 60000 Training images
- CIFAR10
  - 32*32 px, color images
  - 60000 Training images
  - 10 Object classes, 6000 images per class

Generative Adversarial Networks

- Generative Adversarial Networks
  - Generative: Generates new data, as opposed to picking the output from the training
  - Adversarial: Two networks opposes each other



Architecture

- Generator
  - Generating images that look real
  - Uses random noise vector

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
   ConvTranspose2d-1        [-1, 1024, 35, 35]       1,638,400
       BatchNorm2d-2        [-1, 1024, 35, 35]           2,048
              ReLU-3        [-1, 1024, 35, 35]               0
   ConvTranspose2d-4         [-1, 512, 70, 70]       8,388,608
       BatchNorm2d-5         [-1, 512, 70, 70]           1,024
              ReLU-6         [-1, 512, 70, 70]               0
   ConvTranspose2d-7        [-1, 256, 140, 140]       2,097,152
       BatchNorm2d-8        [-1, 256, 140, 140]             512
              ReLU-9        [-1, 256, 140, 140]               0
  ConvTranspose2d-10        [-1, 128, 280, 280]         524,288
      BatchNorm2d-11        [-1, 128, 280, 280]             256
             ReLU-12        [-1, 128, 280, 280]               0
  ConvTranspose2d-13          [-1, 3, 560, 560]           6,144
             Tanh-14          [-1, 3, 560, 560]               0
================================================================
Total params: 12,658,432
Trainable params: 12,658,432
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.39
Forward/backward pass size (MB): 445.02
Params size (MB): 48.29
Estimated Total Size (MB): 493.70
----------------------------------------------------------------
```

Architecture

- Discriminator
  - Telling them apart from the real ones

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 128, 64, 64]           6,144
         LeakyReLU-2         [-1, 128, 64, 64]               0
            Conv2d-3         [-1, 256, 32, 32]         524,288
       BatchNorm2d-4         [-1, 256, 32, 32]             512
         LeakyReLU-5         [-1, 256, 32, 32]               0
            Conv2d-6         [-1, 512, 16, 16]       2,097,152
       BatchNorm2d-7         [-1, 512, 16, 16]           1,024
         LeakyReLU-8         [-1, 512, 16, 16]               0
            Conv2d-9          [-1, 1024, 8, 8]       8,388,608
      BatchNorm2d-10          [-1, 1024, 8, 8]           2,048
        LeakyReLU-11          [-1, 1024, 8, 8]               0
           Conv2d-12             [-1, 1, 5, 5]          16,384
          Sigmoid-13             [-1, 1, 5, 5]               0
================================================================
Total params: 11,036,160
Trainable params: 11,036,160
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 18.50
Params size (MB): 42.10
Estimated Total Size (MB): 60.79
----------------------------------------------------------------
```

Optimizers and Loss Functions

- Adam
- BCELoss

Results - MNIST

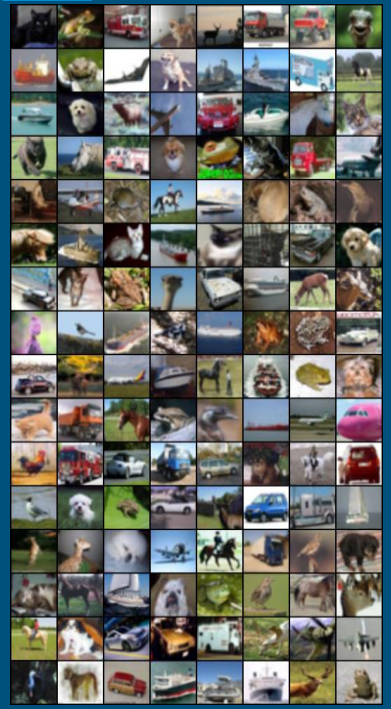
Real Images

Epoch 0

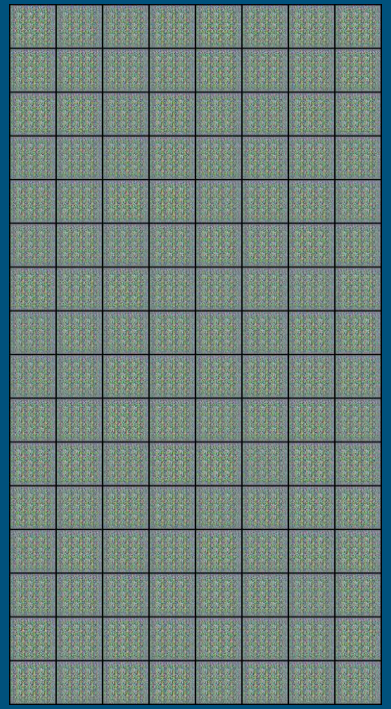
Epoch 12

Epoch 24

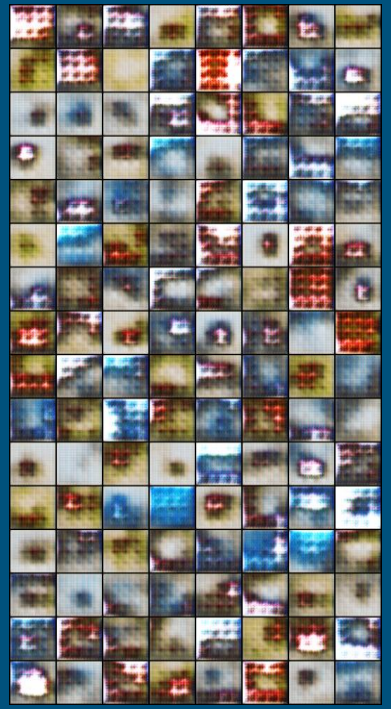
Results - CIFAR10



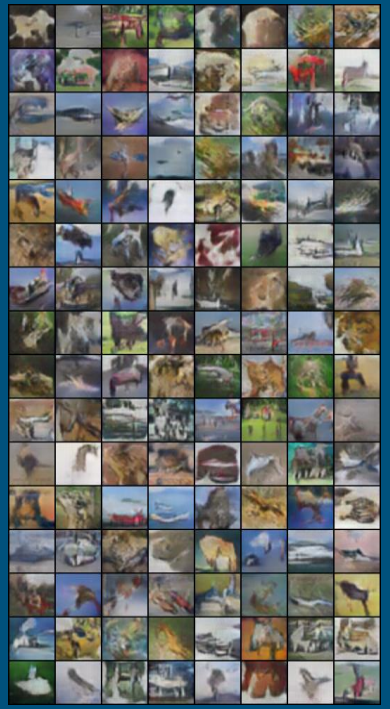
Real Images



Epoch 0



Epoch 43



Epoch 69

Results

```
Epoch 0 [0/391] loss_D_real: 0.4699 loss_D_fake: 1.6642 loss_G: 19.7419
Epoch 0 [100/391] loss_D_real: 0.0002 loss_D_fake: 0.0000 loss_G: 42.5108
Epoch 0 [200/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.4907
Epoch 0 [300/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.4903
Epoch 1 [0/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.5273
Epoch 1 [100/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.2327
Epoch 1 [200/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.1599
Epoch 1 [300/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.2663
Epoch 2 [0/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.3064
Epoch 2 [100/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.1524
Epoch 2 [200/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.1055
Epoch 2 [300/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.0496
Epoch 3 [0/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.1514
Epoch 3 [100/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 62.9673
Epoch 3 [200/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.2096
Epoch 3 [300/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.2598
Epoch 4 [0/391] loss_D_real: 0.0000 loss_D_fake: 0.0000 loss_G: 63.0447
```

Results

```
Epoch 43 [0/391] loss_D_real: 0.4824 loss_D_fake: 0.4212 loss_G: 1.8468
Epoch 43 [100/391] loss_D_real: 0.3366 loss_D_fake: 0.3151 loss_G: 2.8630
Epoch 43 [200/391] loss_D_real: 0.8520 loss_D_fake: 0.2113 loss_G: 1.2174
Epoch 43 [300/391] loss_D_real: 0.1238 loss_D_fake: 0.5629 loss_G: 4.3717
Epoch 44 [0/391] loss_D_real: 0.4363 loss_D_fake: 0.2508 loss_G: 2.6324
Epoch 44 [100/391] loss_D_real: 0.1837 loss_D_fake: 1.1413 loss_G: 3.0155
Epoch 44 [200/391] loss_D_real: 1.4962 loss_D_fake: 0.4299 loss_G: 1.5471
Epoch 44 [300/391] loss_D_real: 0.1951 loss_D_fake: 0.2271 loss_G: 2.5267
Epoch 45 [0/391] loss_D_real: 0.2502 loss_D_fake: 0.4718 loss_G: 3.1078
Epoch 45 [100/391] loss_D_real: 0.2519 loss_D_fake: 0.6943 loss_G: 2.2973
Epoch 45 [200/391] loss_D_real: 0.4043 loss_D_fake: 0.2569 loss_G: 1.9496
Epoch 45 [300/391] loss_D_real: 0.6508 loss_D_fake: 0.1489 loss_G: 1.8419
Epoch 46 [0/391] loss_D_real: 0.4050 loss_D_fake: 0.4054 loss_G: 2.5407
Epoch 46 [100/391] loss_D_real: 0.2168 loss_D_fake: 0.9137 loss_G: 3.3586
```

Results

```
Epoch 66 [0/391] loss_D_real: 2.8737 loss_D_fake: 0.0048 loss_G: 1.2074
Epoch 66 [100/391] loss_D_real: 0.0053 loss_D_fake: 1.7530 loss_G: 7.2025
Epoch 66 [200/391] loss_D_real: 0.0127 loss_D_fake: 0.0015 loss_G: 6.9632
Epoch 66 [300/391] loss_D_real: 0.1592 loss_D_fake: 0.1302 loss_G: 3.2989
Epoch 67 [0/391] loss_D_real: 0.0610 loss_D_fake: 0.4504 loss_G: 3.9392
Epoch 67 [100/391] loss_D_real: 0.7237 loss_D_fake: 0.1683 loss_G: 1.3079
Epoch 67 [200/391] loss_D_real: 0.0113 loss_D_fake: 0.0909 loss_G: 3.7508
Epoch 67 [300/391] loss_D_real: 0.2994 loss_D_fake: 0.1960 loss_G: 2.5880
Epoch 68 [0/391] loss_D_real: 0.0756 loss_D_fake: 0.4995 loss_G: 3.6180
Epoch 68 [100/391] loss_D_real: 0.6416 loss_D_fake: 0.2977 loss_G: 1.4312
Epoch 68 [200/391] loss_D_real: 0.1418 loss_D_fake: 0.3302 loss_G: 2.4292
Epoch 68 [300/391] loss_D_real: 1.8320 loss_D_fake: 0.0171 loss_G: 1.6105
Epoch 69 [0/391] loss_D_real: 0.2063 loss_D_fake: 0.0791 loss_G: 2.4943
Epoch 69 [100/391] loss_D_real: 0.0066 loss_D_fake: 0.0135 loss_G: 4.9216
```

References

- Generative Adversarial Nets
- A Gentle Introduction to Generative Adversarial Networks (GANs)
- Generative Adversarial Networks Cookbook
- A basic intro to GANs (Generative Adversarial Networks)
- An End-to-End Introduction to Generative Adversarial Networks(GANs)
- DCGAN TUTORIAL
- Hands-On Generative Adversarial Networks with PyTorch 1.x - John Hany, Greg Walters
- Machine learning in medical imaging - Ashnil Kumar, Lei Bi, Jinman Kim, David Dagan Feng