Machine Learning, Spring 2020

DL Project Two – GAN

Python tutorial: http://learnpython.org/

TensorFlow tutorial: https://www.tensorflow.org/tutorials/

PyTorch tutorial: https://pytorch.org/tutorials/

GAN-Human Face Synthesis

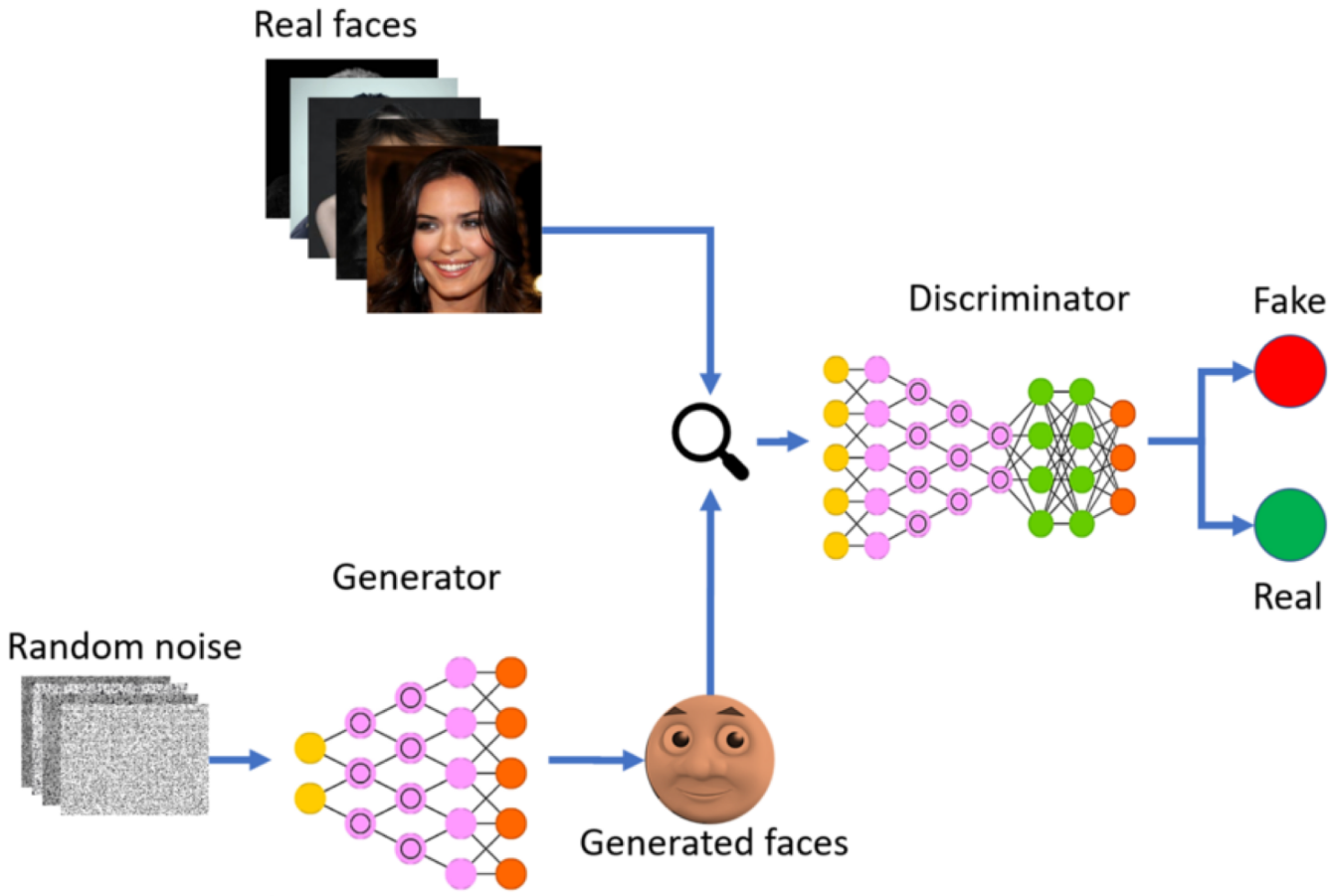
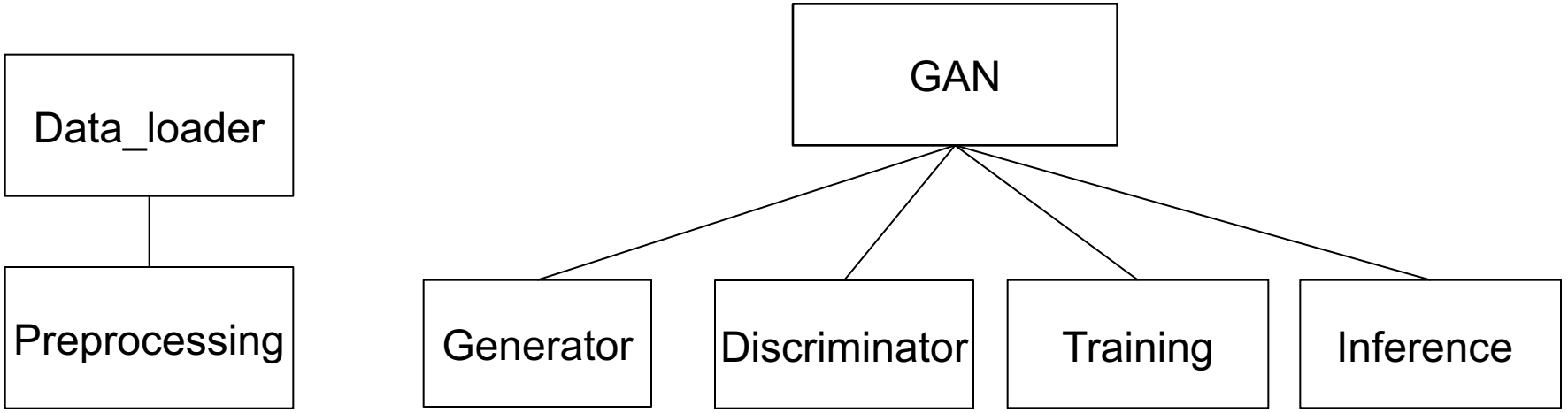


Image source: https://medium.com/sigmoid/a-brief-introduction-to-gans-and-how-to-code-them-2620ee465c30

Main Modules for GAN



Main Modules for GAN

Data_loader

Preprocessing

- Read all images from given 'root/split'
- For each image (preprocessing):
  - Converts images into numbers
  - Normalize the data with mean (0.5, 0.5, 0.5)
  - Crop and rescale the data

  * Feel free to try other settings

```
In [20]: data_loader = get_loader(root='./data/CelebA',split='train',batch_

Found 162770 images in subfolders of: ./data/CelebA/splits/train
```

Example result:

Real image

Main Modules for GAN

Generator

- Input: latent_vector (i.e. random noise)
- Output: Normalized pixel values in [-1, 1]
- Base model consists of
  - Linear layer, ReLU layer, Tanh layer

  * Feel free to add convolution layers or different number of layers (e.g., Linear, ReLU, Tanh)

Discriminator

- Input: image_vector (i.e. normalized pixel values)
- Output: probability if given input is real
- Base model consists of
  - Linear layer, LeakyReLU layer, Sigmoid layer

  * Feel free to add convolution layers or different number of layers (e.g., Linear, LeakyReLU, Sigmoid)

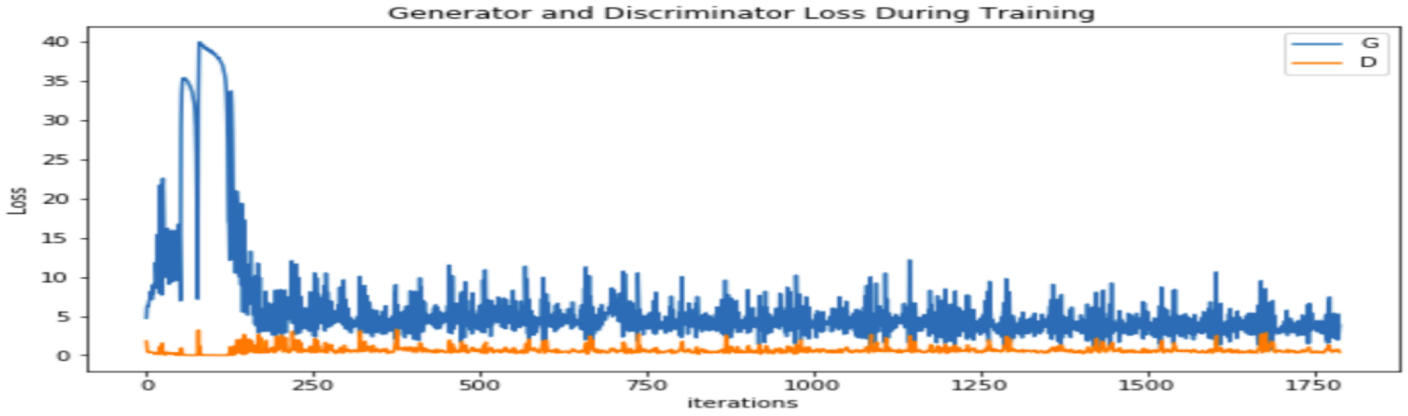
Main Modules for GAN

Optimizer

Training

- Since the generator and discriminator is trained with different losses, define one optimizer for each of them
- For each epoch
  - generate *labels* for real and fake images
  - get the prediction using *D(real_images)*
  - generate random noise *z* , then get the generated image using *G(z), and* get the prediction using *D(G(z))*
  - Compute the loss and update the parameters
    - Loss_d: -log(D(real_image) - log(1-log(D(G(z)))
    - Loss_g: log(1-D(G(z)))

Example result:



Main Modules for GAN

Inference

- Input: latent_vector (i.e. random noise)
- Output: Normalized pixel values in [-1, 1]
- Call trained G(Z)

Example result:

Main Modules for GAN