Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,[*] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[†] Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Sajid Hussain
2022.11.14

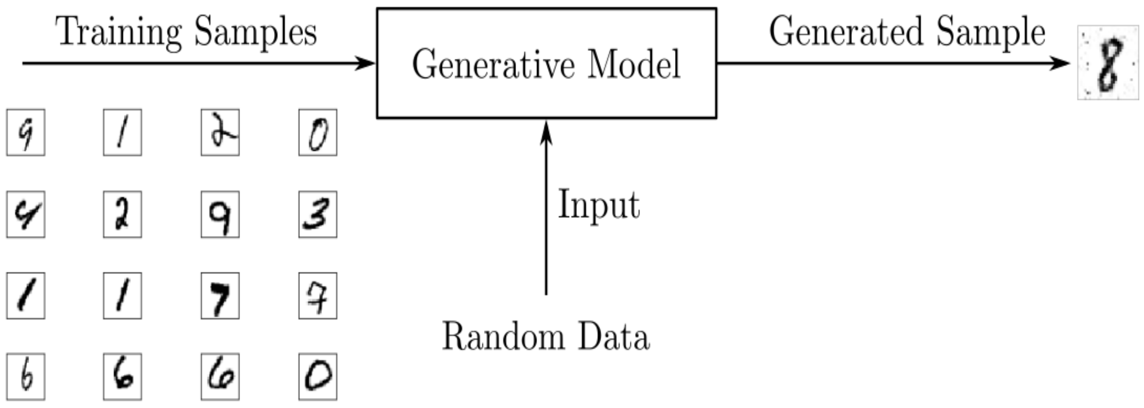
Contents

History of GAN

- Idea of two networks beating each other.
- Technique enables computer generate realistic data
- Results are good than other models

Generative vs Discriminative models

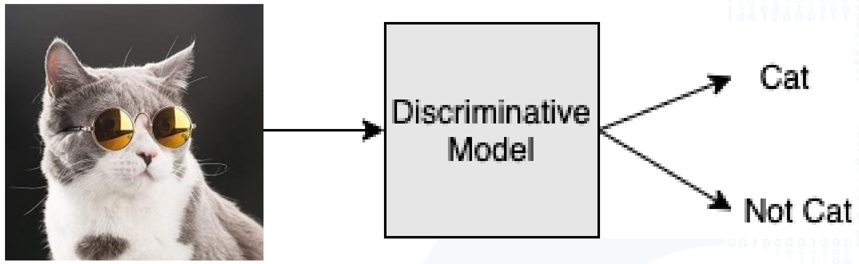
Genarative Models



Noise, Class

ξ Y → Features X

Discriminative Models



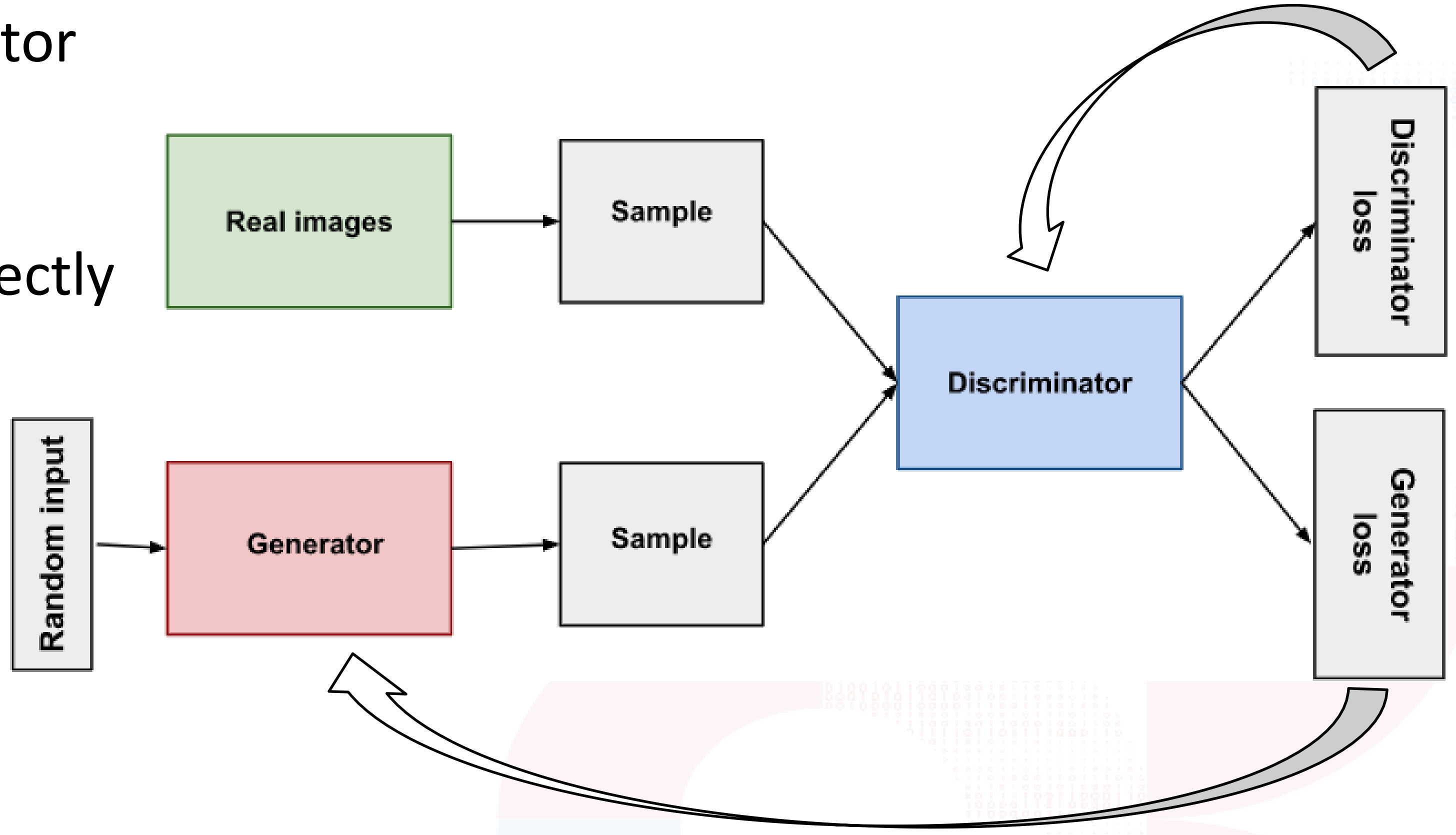
Features X → Class Y

What is GAN?

- GANs are composed of two models that compete with each other and reach a point where realistic examples are produced by the generator.
- The generator learns to make fool the discriminator.
- The discriminator learns to distinguish real from fake.

Overview of GAN structure

- Both the generator and the discriminator are neural networks.
- The generator output is connected directly to the discriminator input.
- Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights.

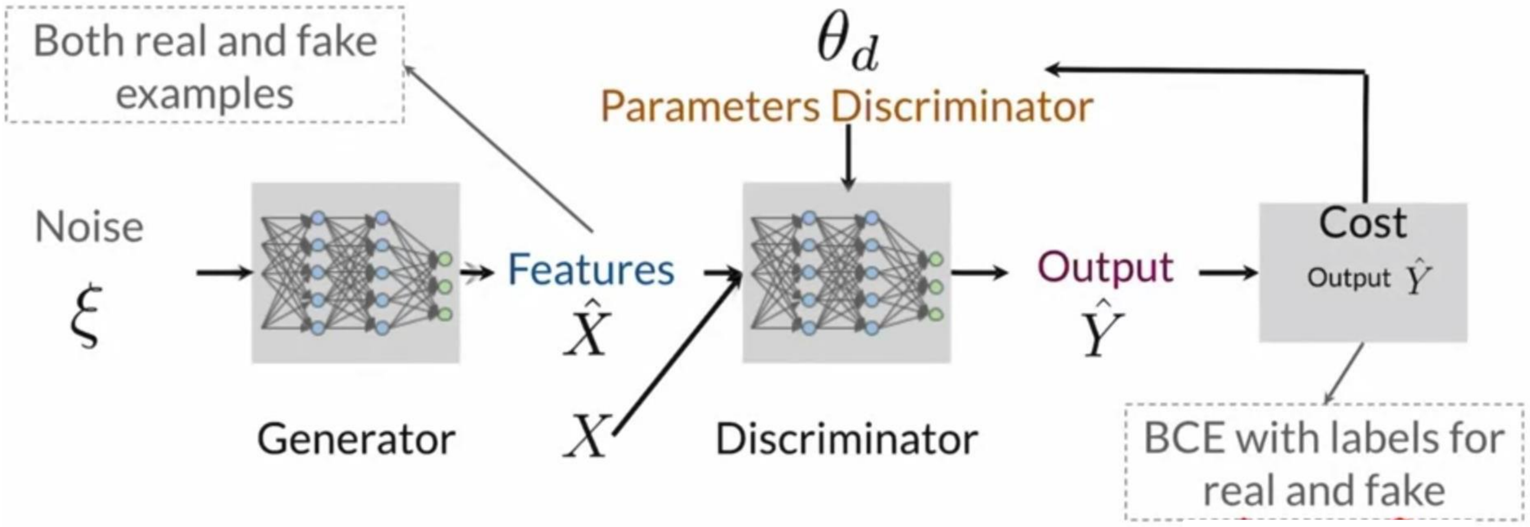


GAN Training – Discriminator Training

- The discriminator classifies both real data and fake data from the generator.

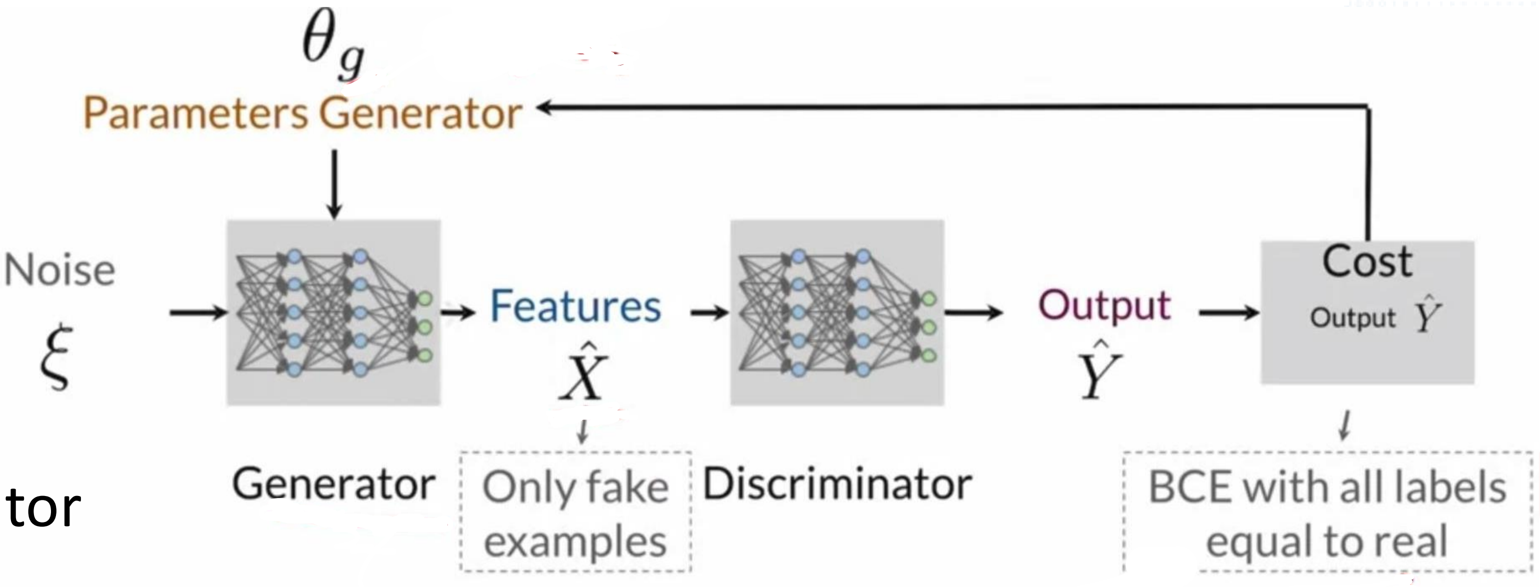
- The generator output is connected directly to the discriminator input.

- Through backpropagation, the discr iminator's classification provides a s ignal that the generator uses to up date its weights.



GAN Training – Generator Training

- Produce generator output from sampled random noise.
- Get discriminator "Real" or "Fake" classification
- Calculate loss from discriminator classification.
- Backpropagate to obtain gradients.
- Use gradients to change the generator weights.



Value function

E_x is the expec ted value over all real data in stances

E_z is the expected value over all generated fake instances G(z)

G(z) is the g enerator's o utput when given noise z

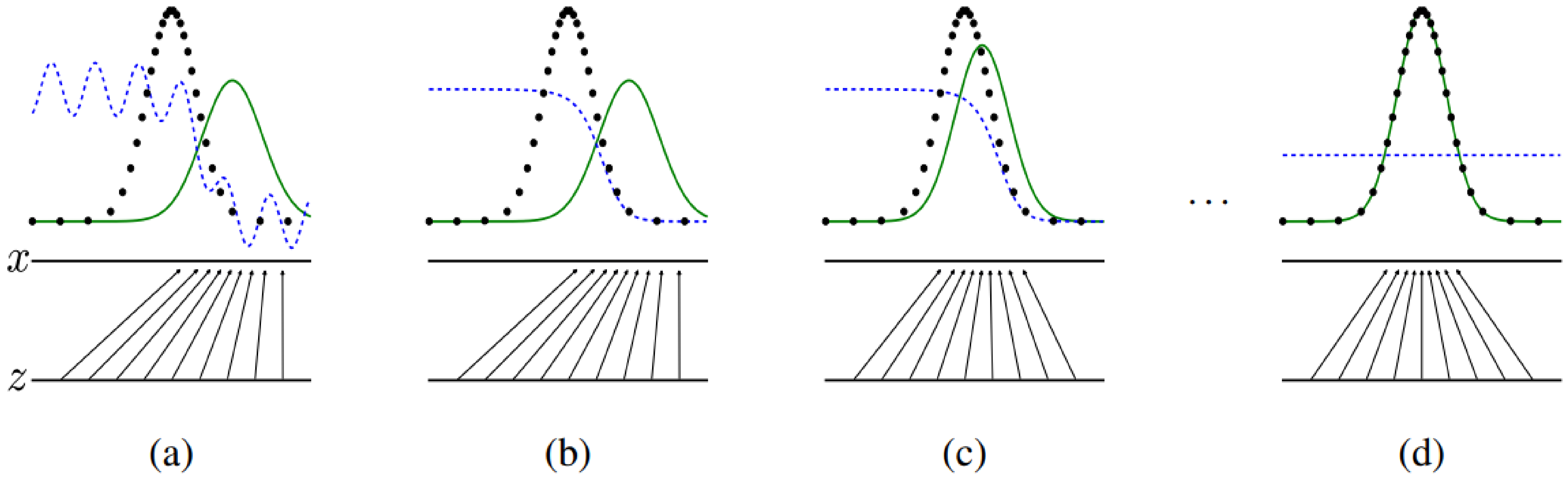
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

D(x) is the discriminator's estimate of the probabilit y that real data instance x is real

D(G(z)) is the discrimina tor's estimate of the pro bability that a fake insta nce is real

The Generator tries to minimize this function while the Discriminator tries to maximize it.
The Generator can't directly affect the **log(D(x))** term in the function, so it minimizesthe equiv alent **log(1 - D(G(z)))**.

GAN Training - Intuition



GAN Training - Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

---

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

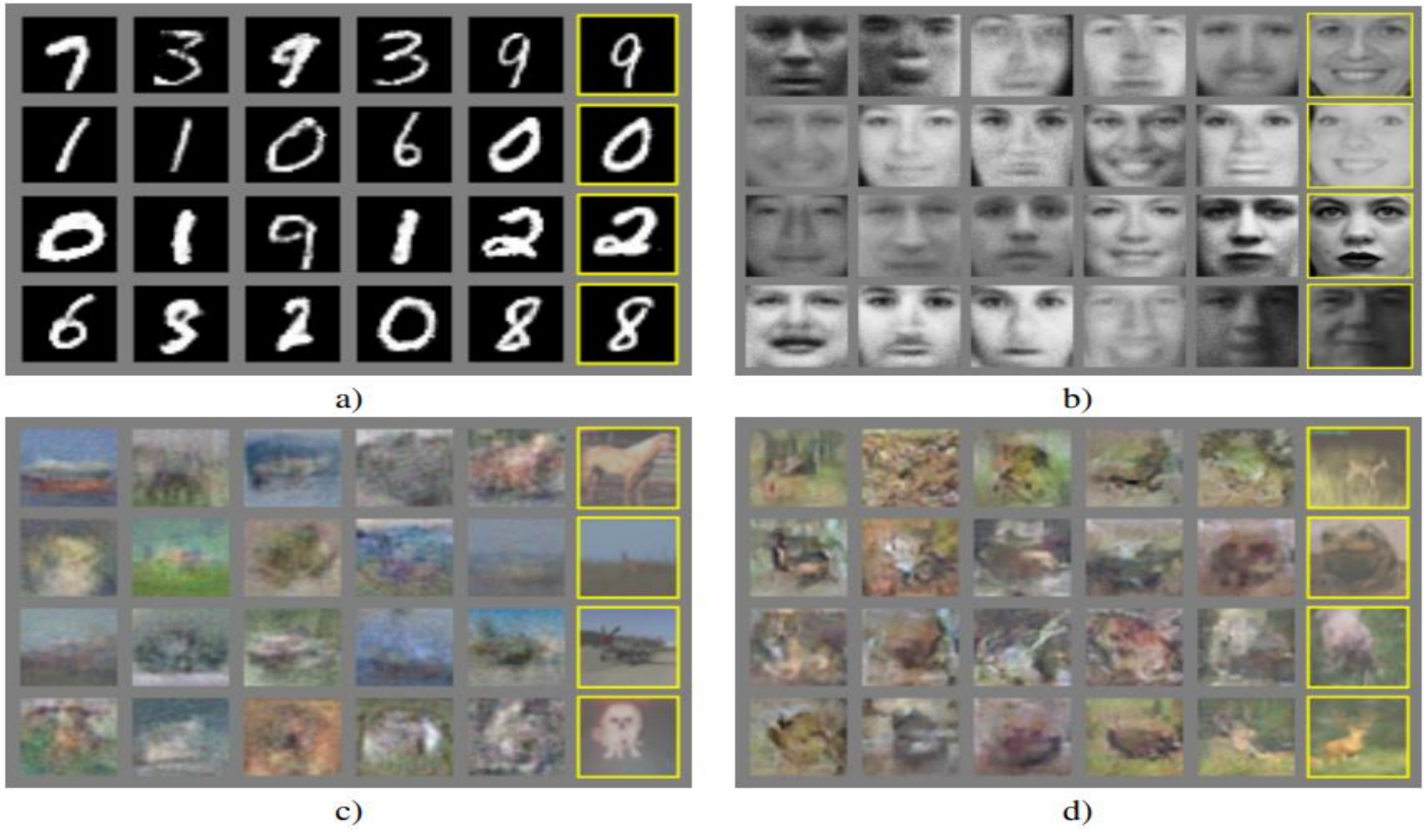
Experiments

Dataset:

- MNIST Digits
- CIFAR-10
- Toronto Face Dataset

Generator	Discriminator
Relu, Sigmoid	Maxout, Dropout

GAN Experiments



Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set.

GAN Training - Algorithm

Advantages

- Only backprop is used to obtain gradients
- Generator network not being updated directl-y with data examples, but only with gradients flowing through the discriminator => computational advantage
- GAN can represent very sharp, even degenerate distributions

Disadvantages

- D must be synchronized well with G during training

Thank you

Q & A