Deep (5):
Generative models

Matthieu Cord LIP6 / SU
Generation

Writing Poems?

Drawing?
Generative models

Outline

1. Preview: Auto-Encoders, VAE
2. Generative models
3. GAN architectures for image generation
Review: Auto-encoder

As close as possible

Minimize reconstruction error

Randomly generate a vector as code

Image?
Review: Auto-encoder
Review: Auto-encoder
Auto-encoder

From a normal distribution $N(0,1)$

$X$ +

$\sigma_1$ $\sigma_2$ $\sigma_3$

$e_1$ $e_2$ $e_3$

$\sum$ $c_1$ $c_2$ $c_3$

$\times$

$\sigma_i e_i + m_i$

Problems of AE/VAE

• It does not really try to simulate real images

One pixel difference from the target

Realistic

Non Realistic
Problems of AE/VAE

GAN to tackle this pb:

Realistic  Non Realistic

GAN: generative adversarial networks

Game scenario:

Player1, Generator, produces samples
Player2, – Its adversary Discriminator, attemps to distinguish real samples from fake generated ones (produced by P1)!
Generative models

Outline

1. Preview: Auto-Encoders, VAE
2. Generative models
   Details in course:
   - AE notation and optimization
   - \( O: \) Objective for Gen optimization framework
   - \( 01: \) Maximum Likelihood
   - \( 02: \) MMD Max Mean Discrepancy
   - \( 03: \) GAN framework, optimization objective function, Algo
3. GAN architectures for image generation
GAN model

GAN – Generative Adversarial Nets

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 $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
    • Sample minibatch of $m$ examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
    • Update the discriminator by ascending its stochastic gradient:
      $$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(x^{(i)}\right) + \log \left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$$
  end for
  • Sample minibatch of $m$ noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
  • Update the generator by descending its stochastic gradient:
    $$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$
end for

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

\begin{itemize}
  \item \(D(x)\) tries to be near 1
  \item Differentiable function \(D\)
  \item \(x\) sampled from data
  \item \(D(G(z))\) tries to make near 0,
  \item \(G\) tries to make \(D(G(z))\) near 1
  \item \(x\) sampled from model
  \item Differentiable function \(G\)
  \item Input noise \(z\)
\end{itemize}

(Goodfellow 2016)
The evolution of generation

NN Generator v1

Disriminator v1

Real images: 5 0 4 1

NN Generator v2

Disriminator v2

NN Generator v3

Disriminator v3

Binary Classifier
GAN - Discriminator

Randomly sample a vector

Generator v1

Something like Decoder in VAE

Real images:

1/0 (real or fake)
GAN - Generator

Updating the parameters of generator

The output be classified as “real” (as close to 1 as possible)

Generator + Discriminator = a network

Using gradient descent to update the parameters in the generator, but fix the discriminator
One example GAN

Source of images: https://zhuanlan.zhihu.com/p/24767059
DCGAN: https://github.com/carpedm20/DCGAN-tensorflow
GAN

100 rounds
GAN

1000 rounds
GAN

2000 rounds
GAN

5000 rounds
GAN

10,000 rounds
GAN

20,000 rounds
GAN

50,000 rounds
Generative models

Outline

1. Auto-Encoders, VAE
2. GAN models
3. GAN architectures for image generation