Generative Adversarial Networks

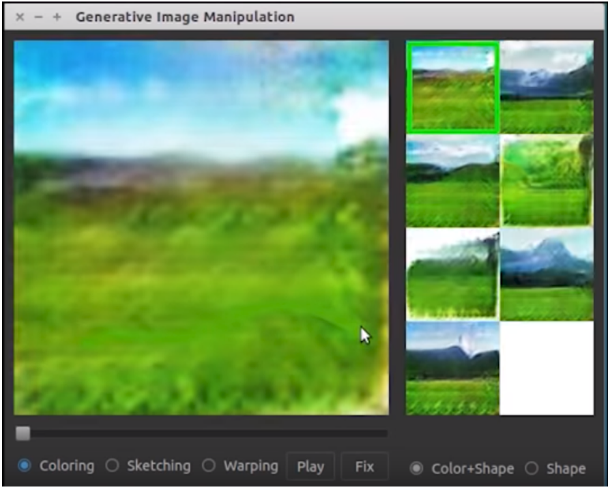
Nan Ye

School of Mathematics and Physics
The University of Queensland

Generative Adversarial Networks (GANs)

Images generated by the vanilla GAN

Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio, Generative adversarial nets, 2014



iGAN

https://www.youtube.com/watch?v=9c4z6YsBGQ0

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACGAN - Coverless Information Hiding Based on Generative adversarial networks
- acGAN - On-line Adaptative Curriculum Learning for GANs

GAN Zoo

https://github.com/hindupuravinash/the-gan-zoo

GANs vs. VAEs

- GANs often produce higher quality images than VAEs.
- GANs do not produce likelihood estimates on held-out data.
- Both GANs and VAEs can be hard to optimize.

Generative Modelling as a Game

- GANs formulate generative modelling as an adversarial game between a generator G and a discriminator D.
  - G tries to fake realistic data.
  - D tries to distinguish generated data and real data.
- G and D play the game, and in the process, both become better and better.

Mathematical formulation

- G generates fake data by mapping a simple variable Z to $G(Z)$.
- D computes the probability that an example x is real as $D(x)$.
- The discriminator's objective
  - D need to assign real example x a large $D(x)$ and a generated example $G(z))$ a small $D(G(z))$.
  - D's discriminating power can be measured by its likelihood on a mixture of real data and fake data

$$L(D, G) = \frac{1}{2} \mathbb{E}_{x \sim p_{data}} \ln D(x) + \frac{1}{2} \mathbb{E}_{z \sim p_Z} \ln(1 - D(G(z))).$$

- The generator's objective
  - G need to fool D by generating $G(z)$'s that make $D(G(z))$ large.
  - G can try to minimize $L(D, G)$.
- Overall, we want to solve the minimax problem

$$\min_G \max_D L(D, G).$$

Optimal discriminator

- If $p_G(x)$ is very different from $p_{data}(x)$, then we expect a good D to be able to confidently decide whether x is real or generated.
- For a fixed G, the optimal discriminator maximizing $L(D, G)$ is

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)},$$

where $p_G(x)$ is the probability that x is generated by G.

Optimal generator

- Assume that G is playing the game with the optimal discriminator, then the optimal G minimizing $L(D, G)$ is G^* such that

$$p_{G^*}(x) = p_{data}(x).$$

- That is, the objective $\min_G \max_D L(D, G)$ encourages the generator to be close to the true data distribution.

Training

- At each iteration
  - For k steps
    - Sample a mini-batch $z_1, \ldots, z_m$ from p_Z.
    - Sample a mini-batch $x_1, \ldots, x_m$ from p_{data}.
    - Update discriminator by ascending along its stochastic gradient

$$\nabla_{\theta_d} \frac{1}{m} \sum_i [\ln D(x_i) + ln(1 - D(G(z_i)))]$$

  - Sample a mini-batch $z_1, \ldots, z_m$ from p_Z.
  - Update generator by descending along its stochastic gradient

$$\nabla_{\theta_d} \frac{1}{m} \sum_i ln(1 - D(G(z_i)))$$

Better objective functions

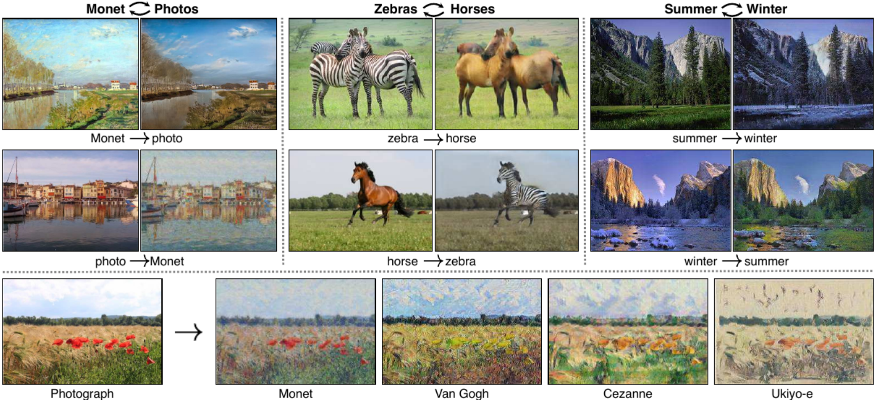
- The loss $\ln(1 - D(G(z)))$ saturates when $D(G(z))$ is close to 0.
  $\Rightarrow$ use $-\ln(D(G(z)))$.
- The loss $L(D, G)$ often leads to unstable behavior
  $\Rightarrow$ replace $\ln D(x)$ by $(D(x) - 1)^2$ and $\ln(1 - D(G(z))$ by $(D(G(z)))^2$.

Deep Convolutional GANs (DCGANs)

- GANs are hard to train without suitable architectural constraints.
- While deep convolutional nets are commonly used, DCGANs use a set of architectural constraints that make GAN training more stable.
  - Replace pooling by strided convolutions (discriminator) and fractional-strided convolution (generator).
  - Use batch normalization in both the discriminator and the generator.
  - ReLU for hidden layers and Tanh for output layer in generator.
  - LeakyReLU for all layers in discriminator.

Radford, Metz, and Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015

CycleGAN



Zhu, Park, Isola, and Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017

- Learn two GANs that translate from one domain to the other simultaneously.
- Enforce a cycle consistency: composition of the two GANs is near identity.

What You Need to Know

- GAN: generative modelling as a game
- GAN training
- Improved GAN objectives
- DCGAN, CycleGAN