The Frontier of GAN

Jesse wang

AiYunxiao Research

wangjunjie@iyunxiao.com

January 30, 2019

Overview

1. Conditional GAN
   - Spectral normalization
   - Self-attention GAN
   - Projection Discriminator
   - Big GAN
2. Embedding condition, style
3. Unconditional GAN: face generation
   - Progressive GAN
   - Style-based progressive GAN

Standard form of GAN

$$\min_G \max_D E_{x\sim q_{data}}[logD(x)] + E_{x'\sim p_G}[log(1 - D(x'))] \tag{1}$$

For a fixed generator:

$$D_G^*(x) := \frac{q_{data}(x)}{q_{data}(x) + p_G(x)} = sigmoid(f^*(x)) \tag{2}$$

where $f^*(x) = \log q_{data}(x) - \log p_G(x)$

$$\Delta_x D_G^* = \Delta sigmoid(f * (x)) \left(\frac{1}{q_{data}(x)} \Delta_x q_{data}(x) - \frac{1}{p_G(x)} \Delta_x p_G(x) \right) \tag{3}$$

The derivative of sigmoid is bounded. However, the latter part is unbounded

Miyato, T., Kataoka, T., etc. al (2018) Spectral normalization for generative adversarial networks. networks.

For Multi-layers network

$$f(x) = D_N W_N \cdots D_1 W_1 x \tag{4}$$

$$\left\| \Delta_x(f(x)) \right\|_2 = \left\| D_N W_N \cdots D_1 W_1 x \right\|_2 \leq \left\| D_N \right\|_2 \left\| W_N \right\|_2 \cdots \left\| D_1 \right\|_2 \left\| W_1 \right\|_2 \tag{5}$$

Because:

$$\left\| A \right\|_2 := \max_{\left\| x \right\|_2 = 1} \left\| Ax \right\|_2 \tag{6}$$

We have:

$$\left\| \Delta_x(f(x)) \right\|_2 \leq \prod_{n=1}^{N} \left\| D_n W_n \right\|_2 \tag{7}$$

for some activations, for example Relu:

$$\left\| D_n \right\|_2 = 1 \tag{8}$$

Hence, the Gradient can be bounded to 1 by:

$$\bar{W}_{SN} := W / \left\| W \right\|_2 \tag{9}$$

see wiki for the math: https://en.wikipedia.org/wiki/Matrix_norm

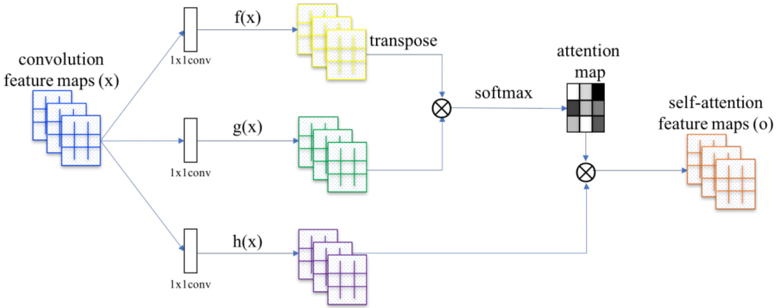


Figure 2: The proposed self-attention mechanism. The $\otimes$ denotes matrix multiplication. The softmax operation is performed on each row.

Residual connection is also added:

$$y_i = \gamma o_i + x_i \tag{10}$$

where γ is a learnable scale parameter. Imbalanced learning rate, spectral normalization are integrated in the work.

Zhang, Han , et al. "Self-Attention Generative Adversarial Networks." (2018).

Projection Discriminator

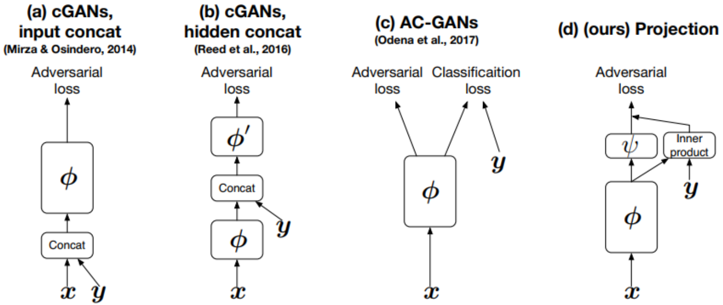


Figure 1: Discriminator models for conditional GANs

easy to implement:

$$f(x, y) := y^T V\phi(x) + \psi(\phi(x)) \tag{11}$$

Miyato, Takeru , and M. Koyama . "cGANs with Projection Discriminator." (2018).

Big GAN

Positive

- Large batch size, up to 2048; large channel size
- Projection Discriminator, SA-GAN architecture
- Conditional BatchNorm layers in G with shared embedding
- The noise vector z is fed into multiple layers of G
- Orthogonal Initialization
- Truncation Trick

Negative

- Progressive growing
- Double the depth
- Sharing class embedding between both G and D
- Adding BatchNorm to D
- Inserting multiple attention block
- Dilation conv ...

trained on 128 to 512 cores of a Google TPU v3 Pod !!!

Brock, Andrew ,etc. al. "Large Scale GAN Training for High Fidelity Natural Image Synthesis." (2018).

Embedding condition, style

- Spatial Batch Norm

$$BN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right)+\beta$$

Compute for each channel

- Instance Norm

$$IN(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right)+\beta$$

Instance Norm
既然是针对每一个样本
那干嘛还要running mean

Compute for each channel and each sample

问题2： 为什么两个affine
parameters 只针对C?
肯定要只针对C， 后来同一个
位置会不断的有独立的图像来

- Adaptive Instance Norm

$$AdaIn(x,y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right)+\mu(y)$$

No affine parameters, but a style input y.

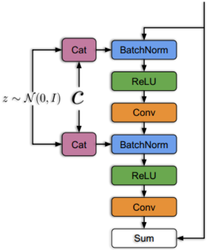
Huang, Xun , and S. Belongie . "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization." International Conference on Computer Vision. (2017).

比较奇怪的是，在pytorch
中instance Norm 的
affine 和 track 都是默
认false，但是Batch Norm
是True

Conditional Batch/Instance Norm

Modulate the affine parameters by condition c.

$$\gamma := \gamma + \Delta\gamma$$
$$\beta := \beta + \Delta\beta$$
$$\Delta\gamma = MLP(c)$$
$$\Delta\beta = MLP(c)$$



Class & z embedding used in bigGAN

Or using different affine parameters for each class

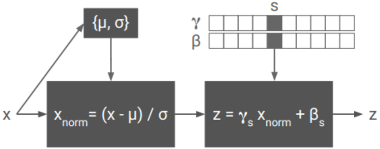


Figure 3: Conditional instance normalization. The input activation x is normalized across both spatial dimensions and subsequently scaled and shifted using style-dependent parameter vectors γ_s, β_s where s indexes the style label.

De Vries, Harm , et al. "Modulating early visual processing by language." (2017).
Dumoulin, Vincent , J. Shlens , and M. Kudlur . "A Learned Representation For Artistic Style." (2016).

Progresses on face generation

Progressive GAN

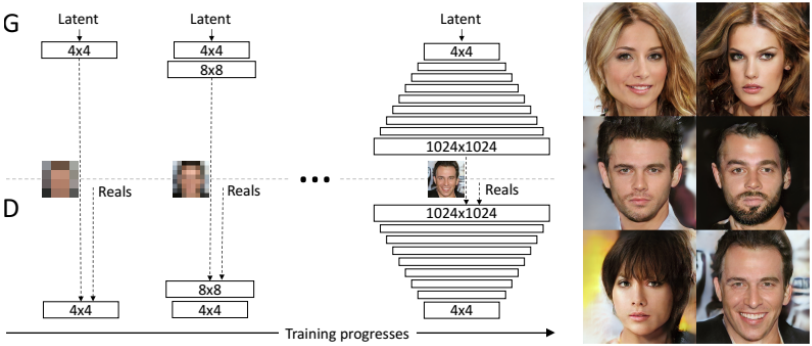
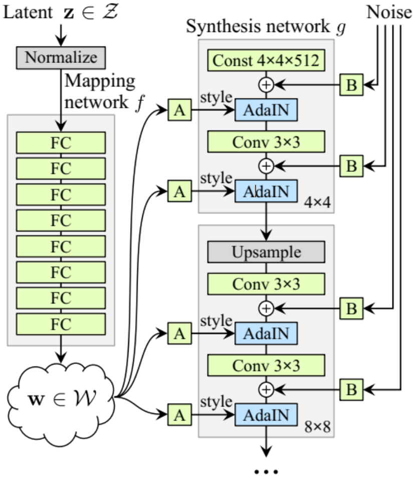


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $\boxed{N \times N}$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

Karras, Tero , et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation." (2017).

Style-based progressive GAN

1. Map latent z to latent space W through multiple layers
2. Transform w to styles y
3. Use AdaIn to integrate styles y
4. The Gaussian noise is broadcasted to all feature maps using a learned scaling factor



(b) Style-based generator

Karras, Tero , S. Laine , and T. Aila . "A Style-Based Generator Architecture for Generative Adversarial Networks." (2018).

To be continued...