A Style-Based Generator Architecture for Generative Adversarial Networks

best paper honorable Mention in CVPR 2019

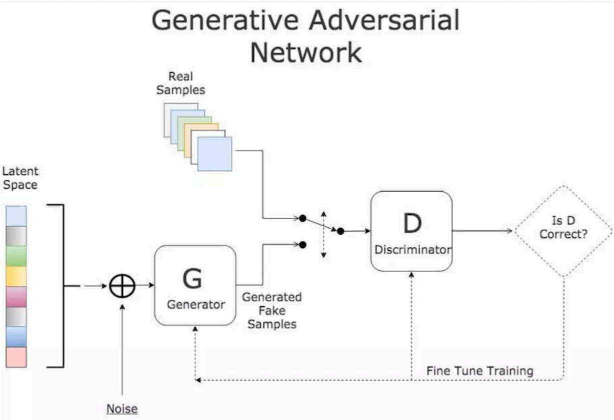
CAO SHILEI, M1

Informatics
Kyushu University

June 25, 2019

Table of contents

GAN



Generative Adversarial Network
Real Samples
Latent Space
D
Discriminator
Is D Correct?
G
Generator
Generated Fake Samples
Fine Tune Training
Noise

generator

The Generator G is used to approximate the actually generate distribution.
1.Given : $z \sim P(z)$, $P(z)$ is a given prior;
2.Parameters : Σ_g
3.Mathematic : using $G(z, \Sigma_g)$got $x \sim p_g$

Discriminator

The discriminator D is used to discriminate x (from empirical distribution or generative distribution)
1.Given :(1) x from real image; (2) x by generator G
2.Output : a scalar the probability of x from true data distribution.

loss fuction

$$\min_{G} \max_{D} V(D, G) = E_{x \sim P_{data}(x)}[logD(x)] + E_{z \sim P(z)}[log(1 - D(G(z)))]$$

Theoretical Results

Global Optimality of $P_g = P_{data}$
We first consider the optimal discriminator D for any given generator G
For G fixed, the optimal discriminator D is :

$$D_G(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

Theoretical Results

the training criterion for the discriminator D, given any generator G, is to maximize the quantity V(G,D):

$$V(G,D) = \int_x P_{data}(x) log(D(x))dx + \int_z P_z(z) log(1 - D(g(z)))dz$$
$$= \int_x P_{data}(x) log(D(x)) + P_g(x) log(1 - D(x))dx \quad (1)$$

For $\forall (a,b) \in R^2 \setminus 0$,the function $alogy + blog(1-y)$ achieves its maximum in [0,1] at $\frac{a}{a+b}$

Theoretical Results

The training objective for D can be interpreted as maximizing the log-likelihood for estimating the conditional probability $P(Y = y \mid x)$, where Y indicates whether x comes from$P_{data}(with\ y = 1)$or from $P_g(with\ y = 0)$.
define

Theoretical Results

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= E_{x\sim P_{data}}[log D_G^*(x)] + E_{z\sim P_z}[log(1 - D_G^*(z)))] \\
&= E_{x\sim P_{data}}[log D_G^*(x)] + E_{x\sim P_g}[log(1 - D_G^*(z)))] \\
&= E_{x\sim P_{data}}[log \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}] + E_{x\sim P_g}[log \frac{P_g(x)}{P_{data}(x) + P_g(x)}]
\end{aligned} \tag{2}
$$

Theoretical Results

The global minimum of the virtual training criterion C(G) is achieved if and only if $P_g = P_{data}$At that point, C(G) achieves the value $log4$.
For $P_g = P_{data}$,$D_g = \frac{1}{2}$it is easy to compute that $C(G) = log4$
Use JensenShannon divergence we get:

$$C(G) = log4 + 2JSD(P_{data} \parallel P_g)$$

where JensenShannon divergence is always non-negative,and zero if they are equal.

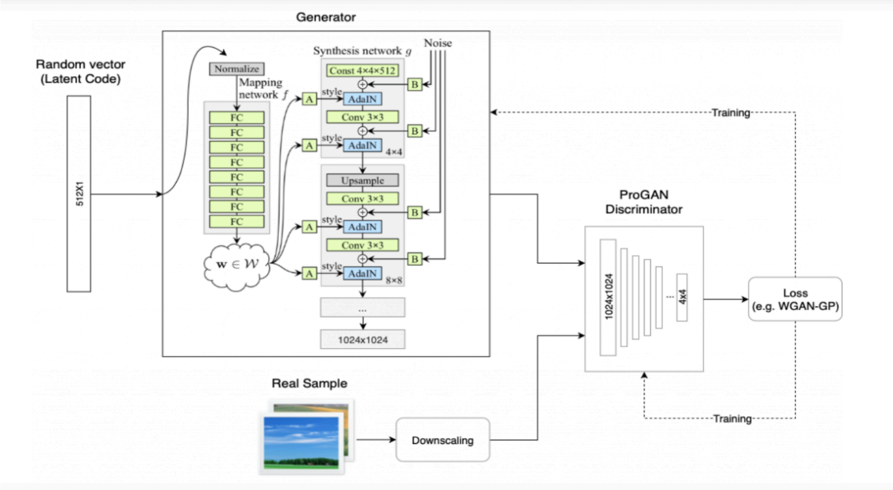
Steps

There are 3 major steps in the training:
1.use the generator to create fake inputs based on noise
2.train the discriminator with both real and fake inputs
3.train the whole model: the model is built with the discriminator chained to the generator.

SImple Example

Figure 3. The Adversarial model is simply generator with its output connected to the input of the discriminator. Also shown is the training process wherein the Generator labels its fake image output with 1.0 trying to fool the Discriminator.

structure

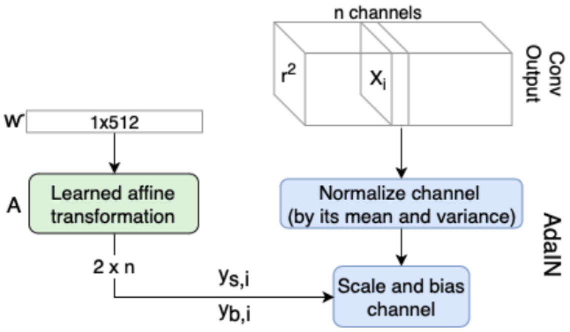

Generator
Random vector
(Latent Code)
512X1
Normalize
Mapping network f
FC
FC
FC
FC
FC
FC
FC
FC
w ∈ W
Synthesis network g
Noise
Const 4×4×512
style
AdaIN
Conv 3×3
4×4
Upsample
8×8
...
1024x1024
A
B
Training
ProGAN Discriminator
1024x1024
4x4
Loss
(e.g. WGAN-GP)
Real Sample
Downscaling

innovation points

Modules (AdaIN)
(1)Each channel of the convolution layer output is first normalized to make sure the scaling and shifting of step 3 have the expected effect.
(2)The intermediate vector w is transformed using another fully-connected layer (marked as A) into a scale and bias for each channel.
(3)The scale and bias vectors shift each channel of the convolution output, thereby defining the importance of each filter in the convolution. This tuning translates the information from x to a visual representation.

innovation points

Modules (AdaIN)



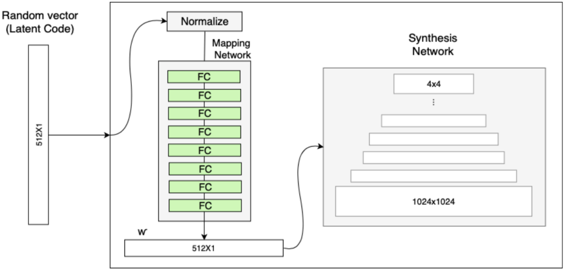
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

innovation points

Mapping Network
The Mapping Networks goal is to encode the input vector into an intermediate vector whose different elements control different visual features. This is a non-trivial process since the ability to control visual features with the input vector is limited, as it must follow the probability density of the training data.

innovation points

Mapping Network



The generator with the Mapping Network (in addition to the ProGAN synthesis network)

Conclusion

StyleGAN is a groundbreaking paper that not only produces high-quality and realistic images but also allows for superior control and understanding of generated images, making it even easier than before to generate believable fake images. The techniques presented in StyleGAN, especially the Mapping Network and the Adaptive Normalization (AdaIN), will likely be the basis for many future innovations in GANs.