Reading Report

2019.5.20

1 Towards Principled Methods for Training

1.1 prove the instability and saturation when training GAN

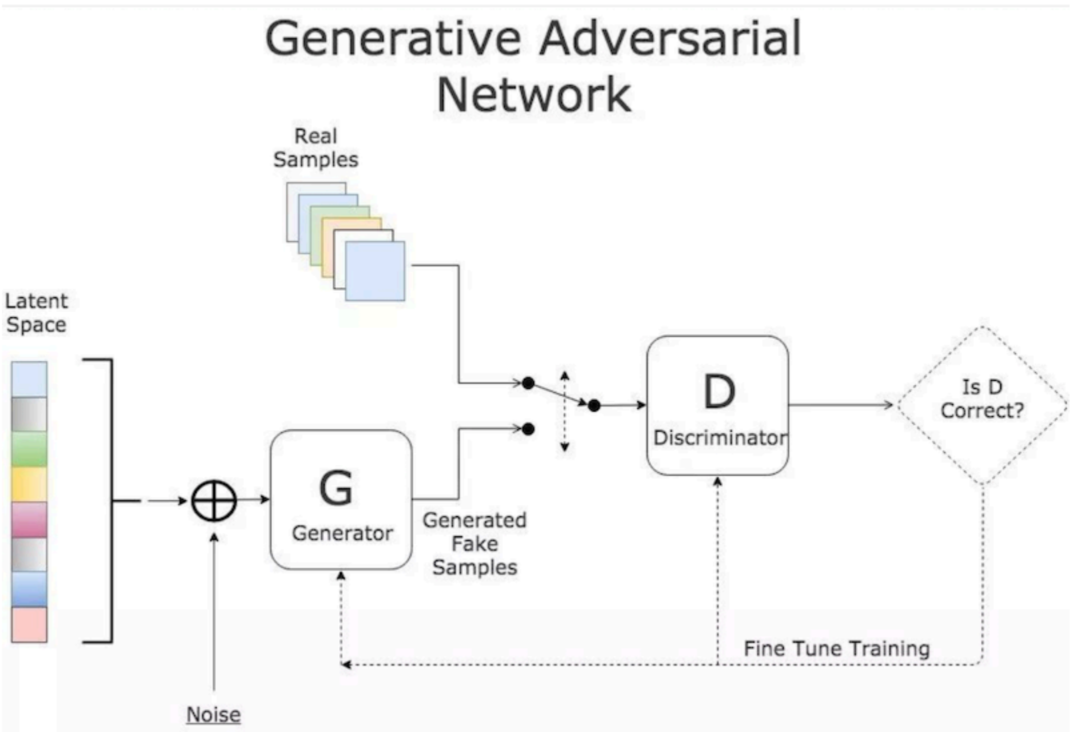

图 1: GAN

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.

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

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

Theoretical Results

Global Optimality of $P_g = P_{data}$
We first consider the optimal discriminator D for any given generator G
Proposition 1
For G fixed, the optimal discriminator D is :

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

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

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

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

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

Theorem 1
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 $-log4$.
proof
For $P_g = P_{data}$,$D_g^* = \frac{1}{2}$it is easy to compute that $C(G) = -log4$
we can also get:

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

Use Jensen−Shannon divergence we get:

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

where Jensen−Shannon divergence is always non-negative,and zero if they are equal.

1.2 GAN by Example using Keras on Tensorflow Backend

https://towardsdatascience.com/gan-by-example-using-keras-on-tensorflow-backend-1a6d515a60d0

1.3 GAN with Keras: Application to Image Deblurring

https://blog.sicara.com/keras-generative-adversarial-networks-image-deblurring-45e3ab6977b5

2 Image Blind Denoising With Generate Adversarial Network Based Noise Model

3 All you need is a good init

3.1 ideals

- Pre-initialize weights of each convolution or fc layer with orthonormal matrices.
- Normalizing the variance of the output of each layer to be equal to one.

3.2 algorithm

Algorithm 1 Layer-Sequential Unit-Variance Initialization

Input: mini-batch: $\{\widetilde{x_i}^{(i)}\}_{i=1}^{m}$; Parameters to be learned:$W, \widetilde{b}$; Hyperparameters:ε, T
Output: $W, \widetilde{b}$

```
Temporarily initialize W with orthonormal matrices
while t < T and |σ² − 1| < ε do
    Compute temporary output {ã_i^(i)}_{i=1}^m
    Compute variance σ²
    Rescale weight W ← W/σ
end while
```