Generative Models

Shihua Zhang

December 22, 2021

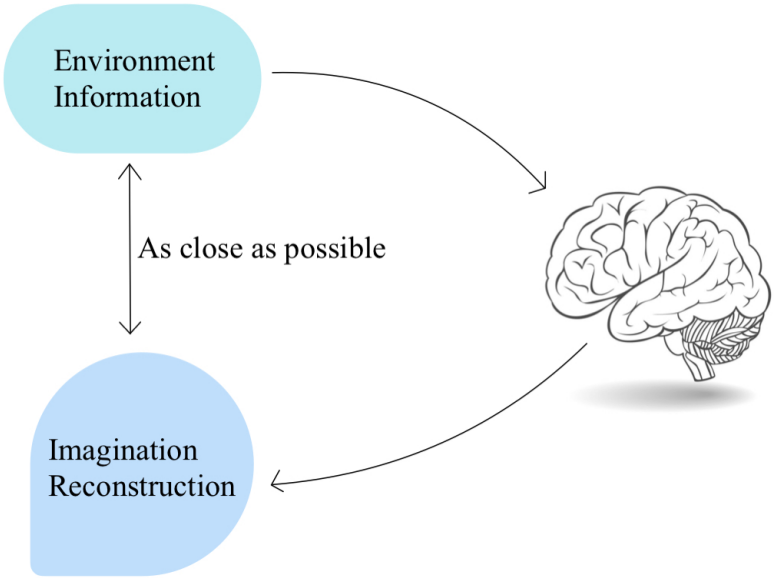
Outline

- Generative Models
- Variational Autoencoder (VAE)
- Generative Adversarial Network (GAN)
- Similarities between VAE and GAN
- Drawbacks of VAE and GAN
- Solutions by Wasserstein Metric
  - Wasserstein GAN
  - Wasserstein AE
- Unified Theory and Stronger Model
- Further Problems

Generative Models: Intuition

Human's learning process can be roughly divided into two parts:

- Recongnition Process:
  Observing objects and taking in new information.
- Imagination Process:
  Recalling information and reconstructing them in mind.

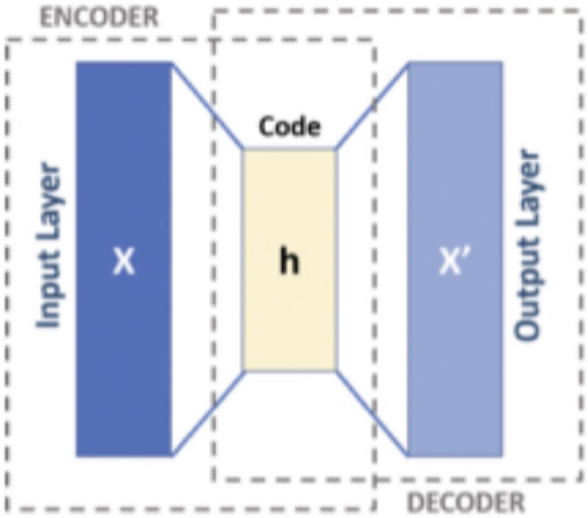


The imagination process can be simulated by a structure, called generative model, with compatible precision and generality.

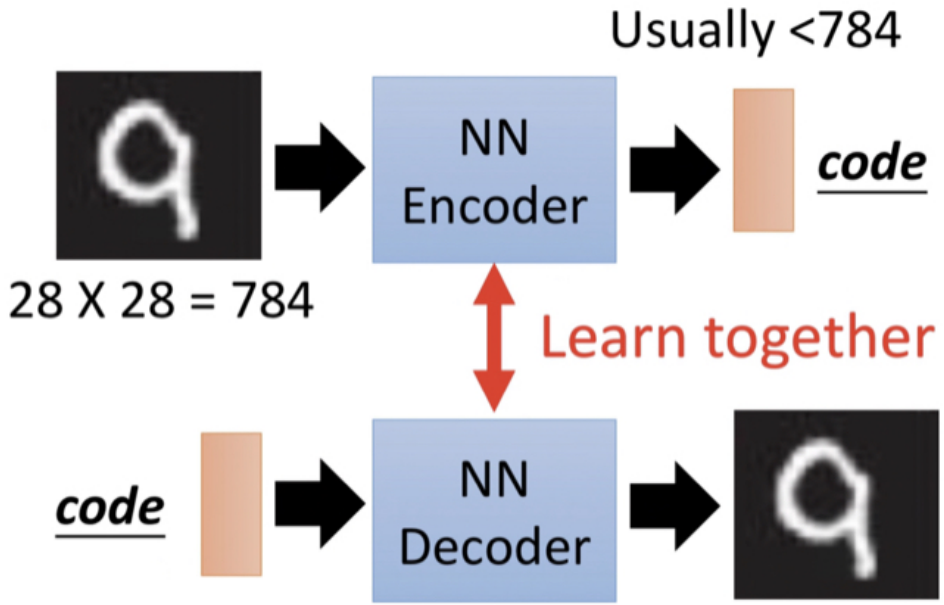
Generative Models: Early Attempts

Autoencoder (AE) can reconstruct input data.

- The simpliset AE is a neural network with one hidden layer.



Schema of a basic Autoencoder



Generative Model: Early Attempts

- AE reconstructs its inputs by minimizing the difference between inputs and outputs.
  - Encoder & Decoder:

$$\phi : X \to Z$$
$$\psi : Z \to X$$
$$\phi, \psi = \text{argmin}|x - (\psi \circ \phi)x|$$

Generative Model: Early Attempts

- AE reconstructs its inputs by minimizing the difference between inputs and outputs.
  - Encoder & Decoder:

$$\phi : X \to Z$$
$$\psi : Z \to X$$
$$\phi, \psi = \text{argmin}|x - (\psi \circ \phi)x|$$

Specify it as a NN:

$$h = \sigma(Wx + b)$$
$$x' = \sigma'(W'h + b')$$

Generative Model: Early Attempts

- AE reconstructs its inputs by minimizing the difference between inputs and outputs.
  - Encoder & Decoder:

$$\phi : X \to Z$$
$$\psi : Z \to X$$
$$\phi, \psi = \text{argmin}|x - (\psi \circ \phi)x|$$

Specify it as a NN:

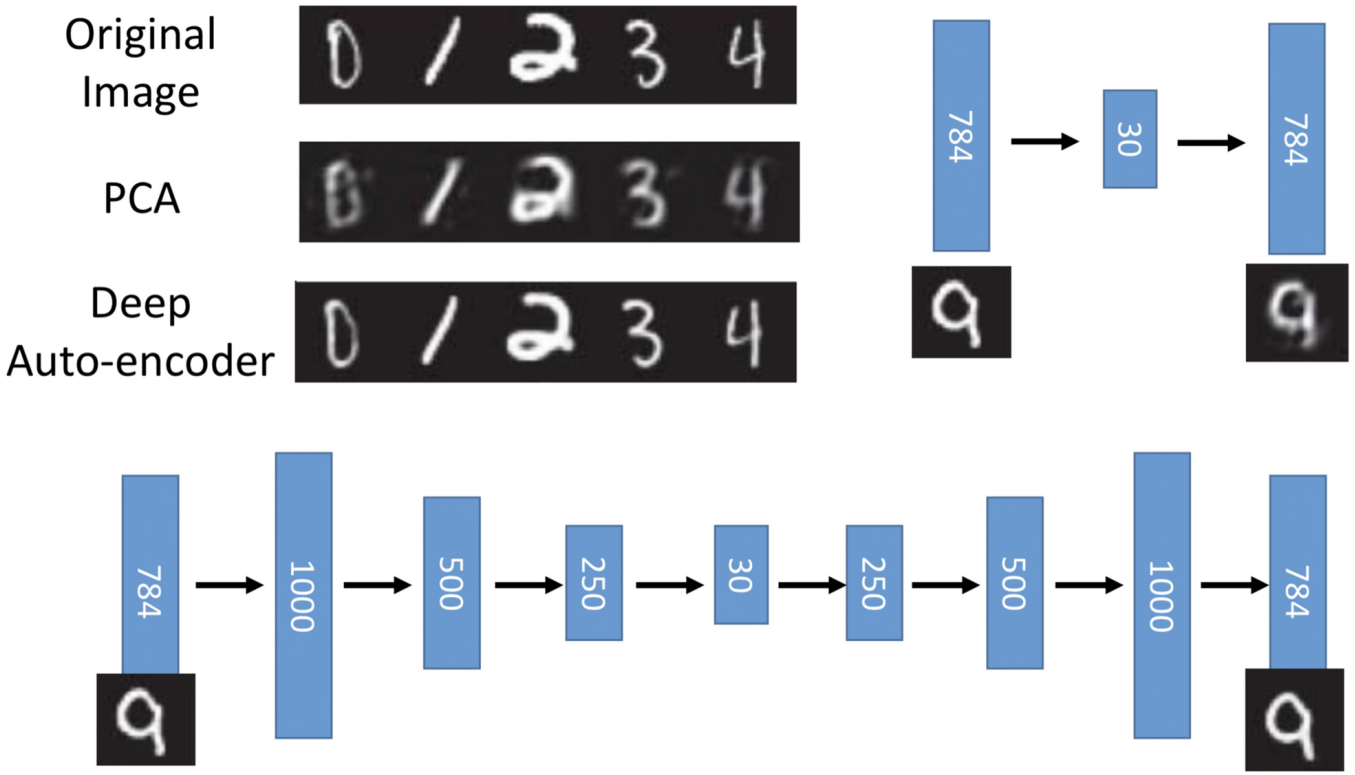
$$h = \sigma(Wx + b)$$
$$x' = \sigma'(W'h + b')$$

  - Reconstruction Error:

$$L(x, x') = |x - x'|^2 = |x - \sigma'(W'\sigma(Wx + b) + b')|^2$$

Generative Model: Early Attempts

- Experimental Result



- Remark: AE can only reconstruct (corresponds to precision) rather than generate (corresponds to generality).

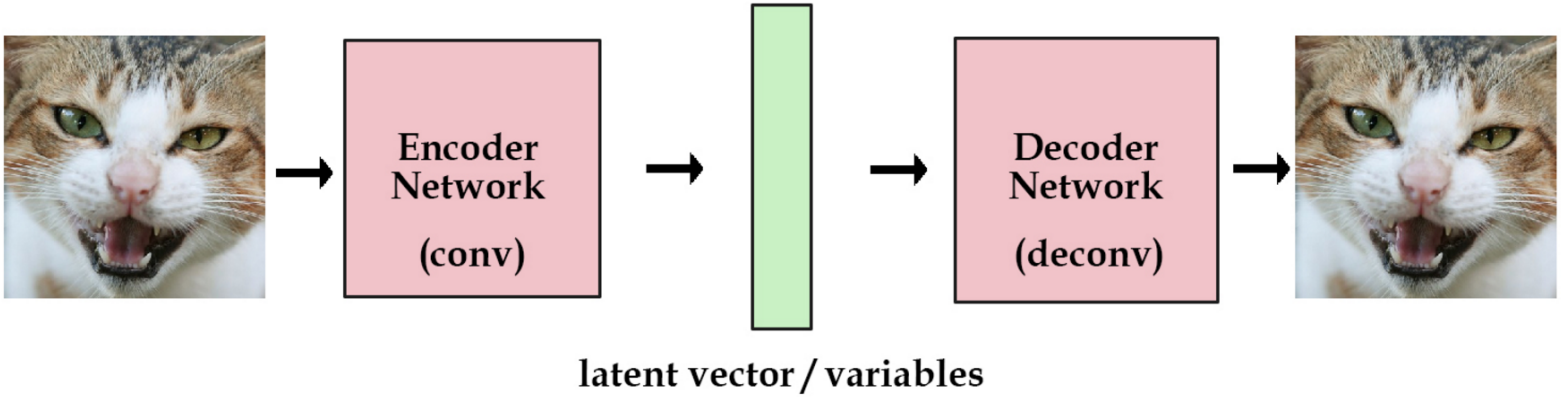
Outline

- Generative Models
- **VAE**[1]
- GAN
- Similarities between VAE and GAN
- Drawbacks of VAE and GAN
- Solutions by Wasserstein Metric
  - Wasserstein GAN
  - Wasserstein AE
- Unified Theory and Stronger Model
- Further Problems

[1] Kingma D.P., Welling M. Auto-Encoding Variational Bayes. ICLR, 2014.

Variational Autoencoder (VAE)

- Variational Autoencoder (VAE) originates from AE
- VAE can also generate new data based on given datasets



左：第1世代，中：第9世代，右：原始图像

VAE: Theoritical Analysis

VAE learns the distribution of a dataset by variational inference.

- Variational Inference
  - Goal: Given a dataset D and a parametrised probability distribution P_θ, we do maximum likelihood estimate on $P_\theta(D) = \prod P_\theta(x_i)$, $x_i \in D$.

VAE: Theoritical Analysis

VAE learns the distribution of a dataset by variational inference.

- Variational Inference
  - Goal: Given a dataset D and a parametrised probability distribution P_θ, we do maximum likelihood estimate on $P_\theta(D) = \prod P_\theta(x_i)$, $x_i \in D$.
  - Method:
    - $\log P_\theta(x^{(1)}, \cdots, x^{(N)}) = \sum \log P_\theta(x^{(i)})$
    - By introducing a new conditional distribution $q(z|x^{(i)})$, we have

      $$\log P_\theta(x^{(i)}) = \text{ELBO} + \text{KL}(q(z|x^{(i)}), p(z|x^{(i)}))$$

      where the ELBO (Evidence Lower Bound) satisfies

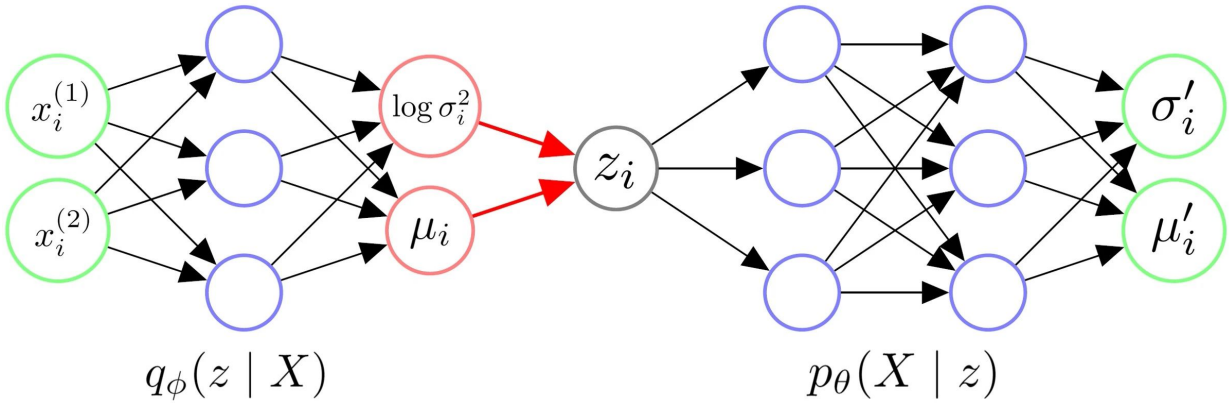
      $$\text{ELBO}^{(i)} = \mathcal{L}(\theta, \phi, x^{(i)}) = -\text{KL}(q(z|x^{(i)}, p_\theta(z)) + \mathbb{E}_{q_\phi(z|x^{(i)})}(p(x^{(i)})|z)$$

      The former one is regularization term; the latter one is nonnegative error.
    - Notice that $\log P_\theta(x^{(i)}) \geqslant$ ELBO, it is easier to improve lower bound ELBO than original likelihood.

Therefore, VAE tries to maximize ELBO to do the MLE task.

VAE: Realization by Neural Network

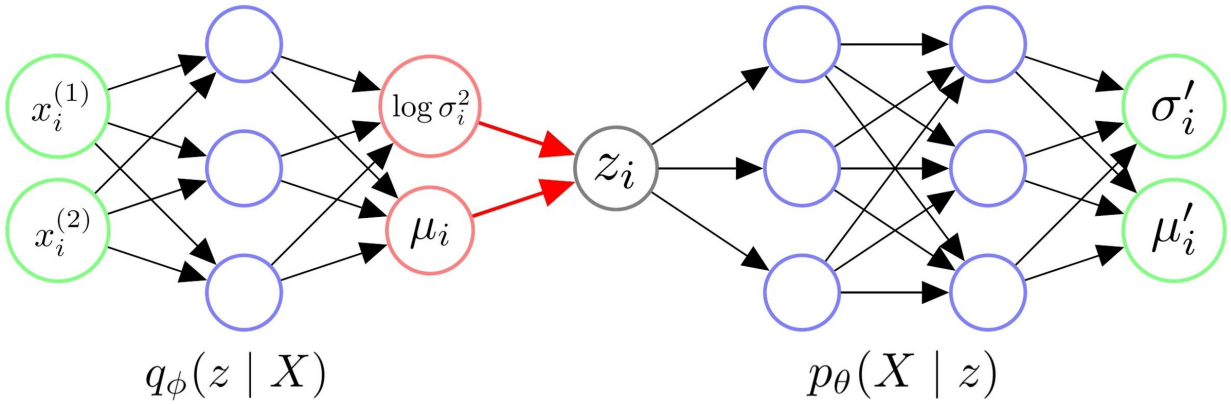
- Basic Encoder-Decoder structure:



When training: regularization term is integrable and error term can be replaced be Euclidean distance.
When generating: we draw samples from $P(Z)$.

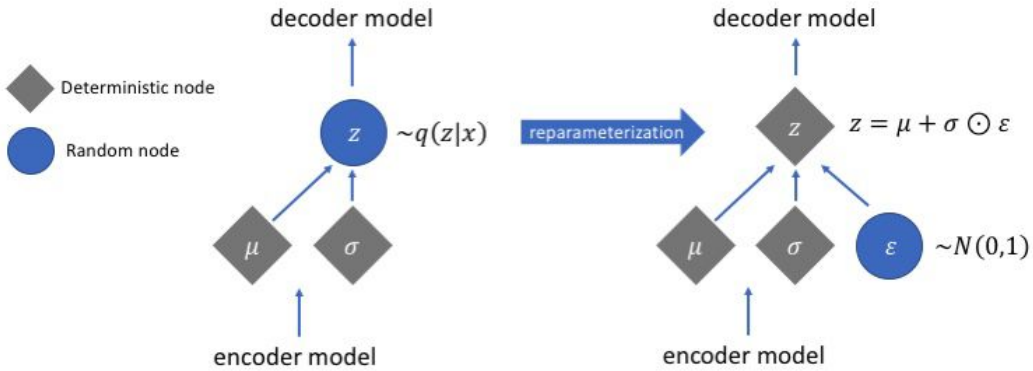
VAE: Realization by Neural Network

- Basic Encoder-Decoder structure:



When training: regularization term is integrable and error term can be replaced be Euclidean distance.
When generating: we draw samples from $P(Z)$.

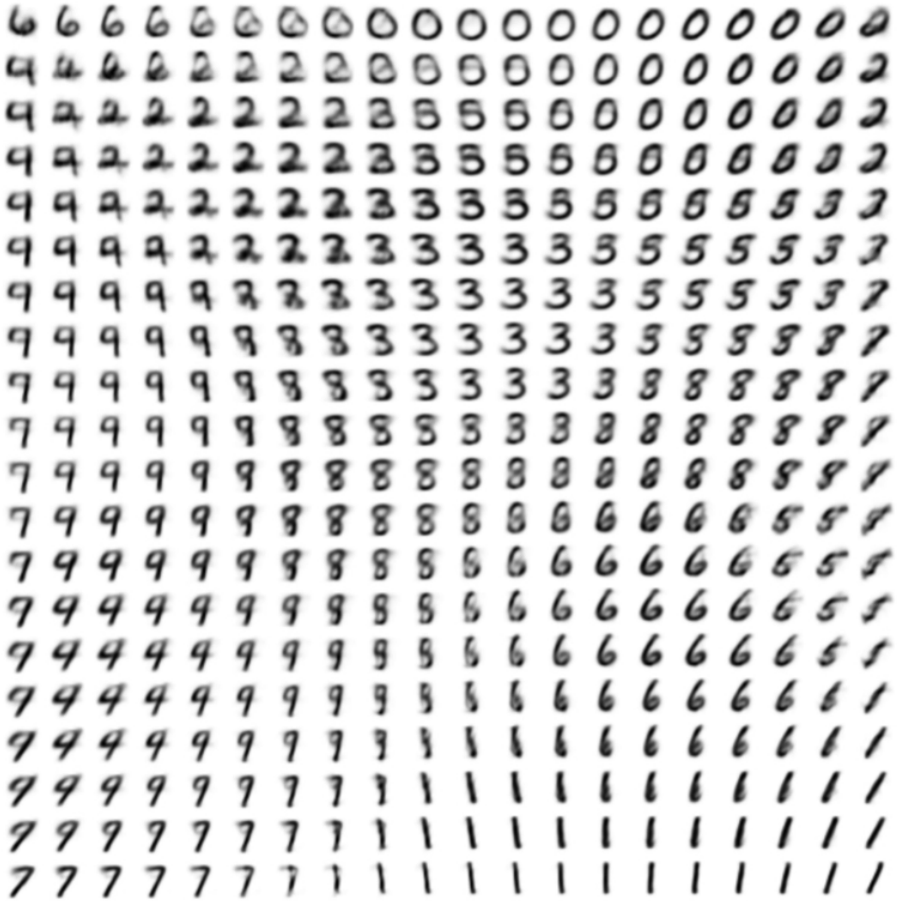
- Reparametrision Trick:



VAE: Experimental Result

- Genertes new data

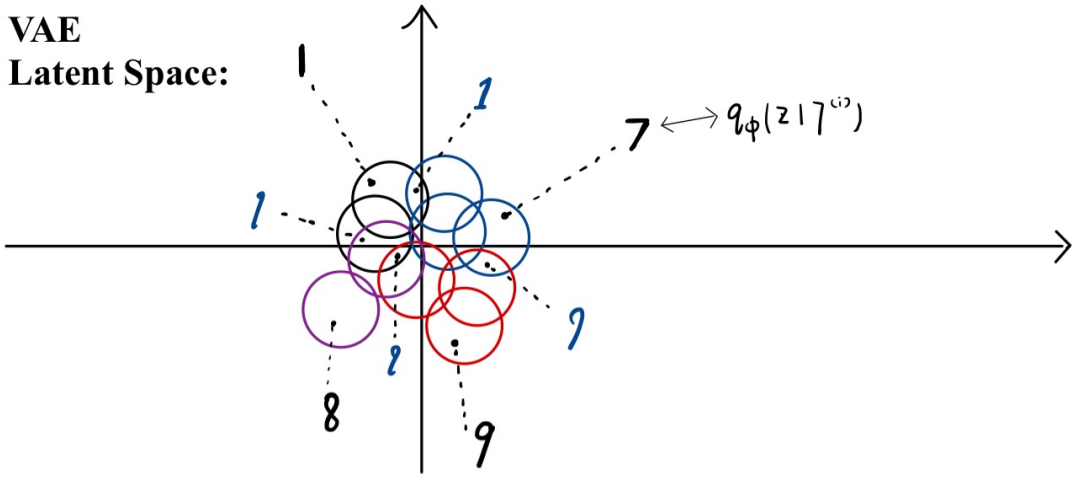
(a) Learned Frey Face manifold

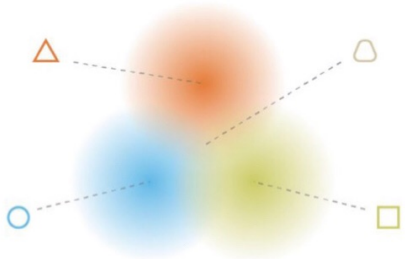


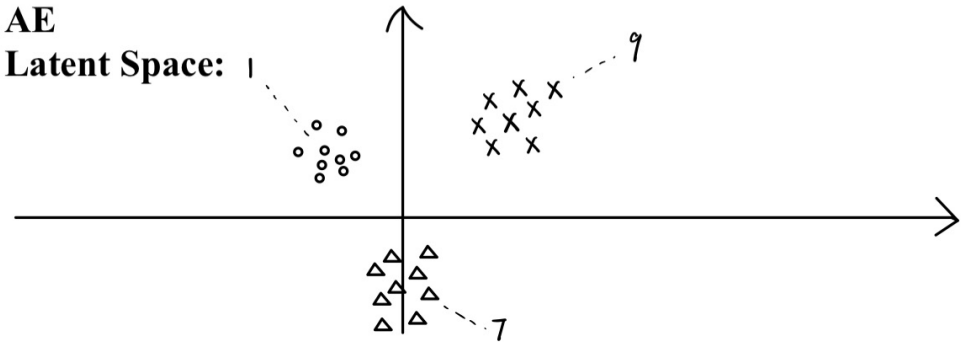
(b) Learned MNIST manifold

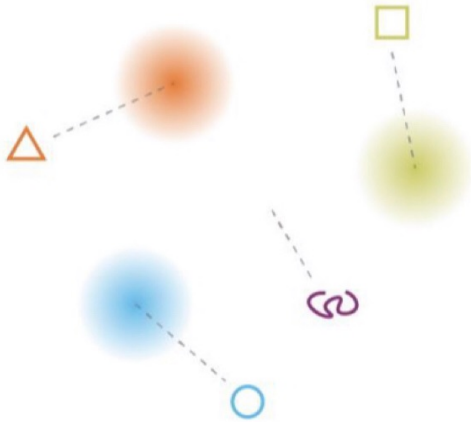
VAE: Experimental Result

- Explanation of Experimental Result









VAE: Problem

- Contrary to AE, VAE perform well on generality rather than precision. For instance, the images generated by VAE are often vague.

- GAN provides an new way to grant both precision and generality.

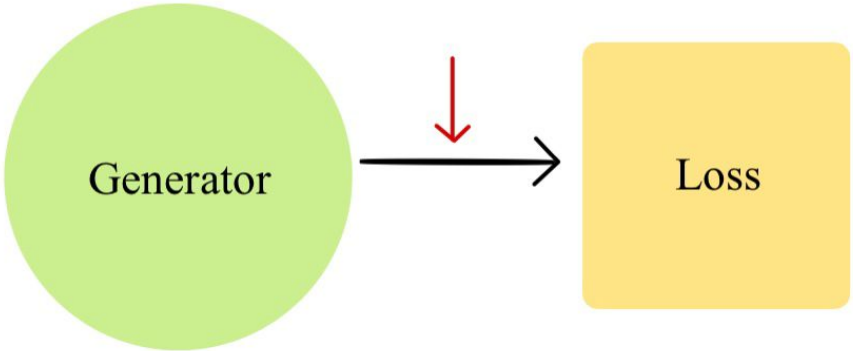
Outline

- Generative Models
- VAE
- **GAN**[2]
- Similarities between VAE and GAN
- Drawbacks of VAE and GAN
- Solutions by Wasserstein Metric
  - Wasserstein GAN
  - Wasserstein AE
- Unified Theory and Stronger Model
- Further Problems

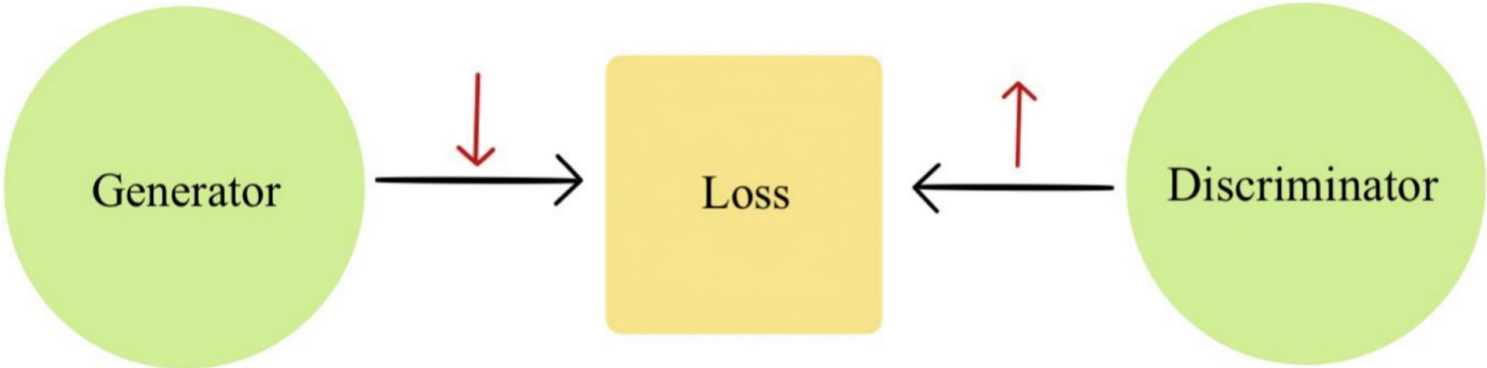
[2]Goodfellow I.J., et al. Generative Adversarial Networks. Advances in Neural Information Processing Systems 3(2014):2672-2680.

Loss Function: from Single to Adversarial

- Conventionally, loss function only decreases to optimize (VAE).

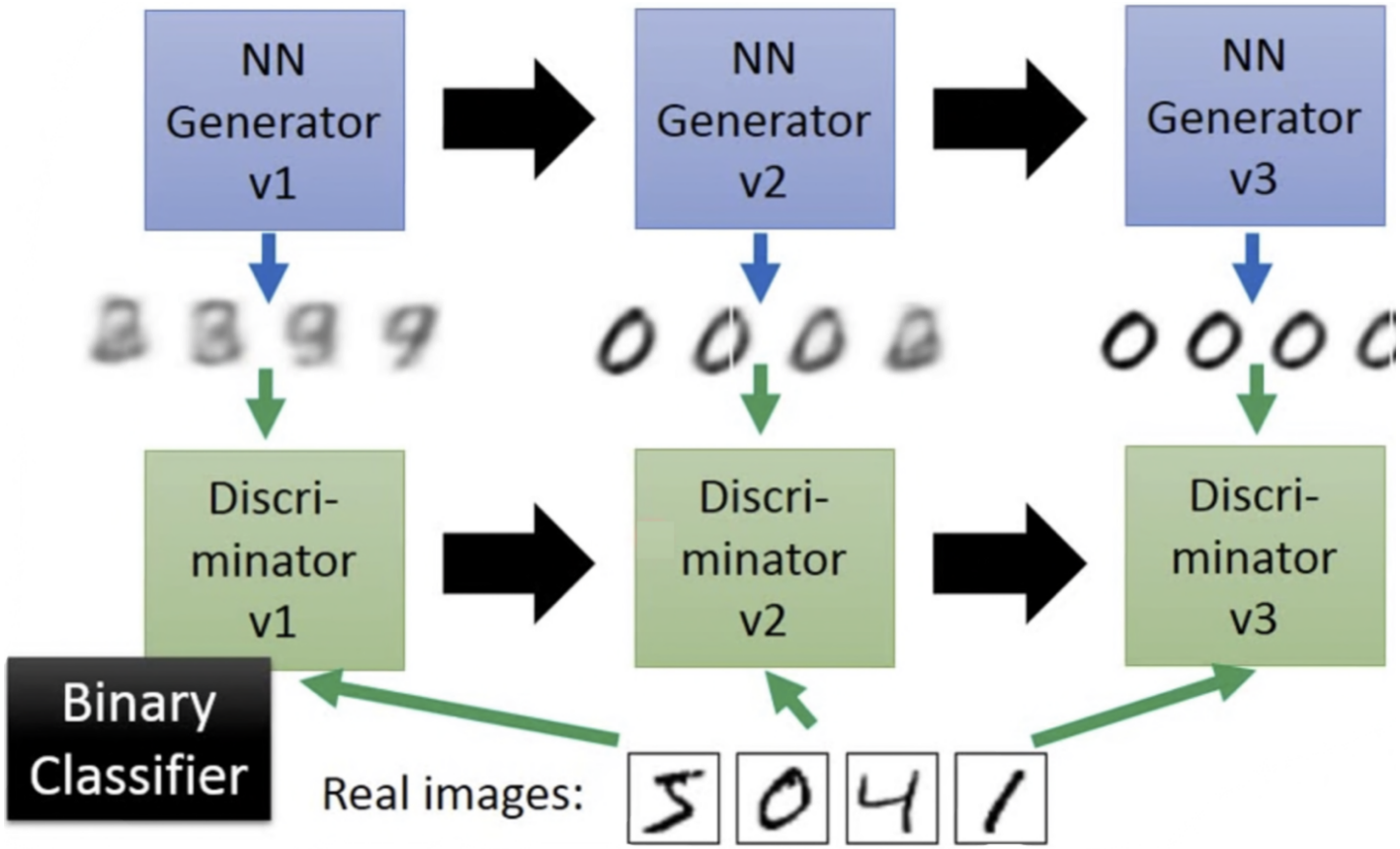


- GAN can reconstruct inputs and generate realistic data in an adversarial way.



GAN: Intuition

- Given a set of real images.
- Generator: draws fake pictures to imitate them.
- Discriminator: distinguishes these fake images from the true ones.



GAN: Theorical Analysis

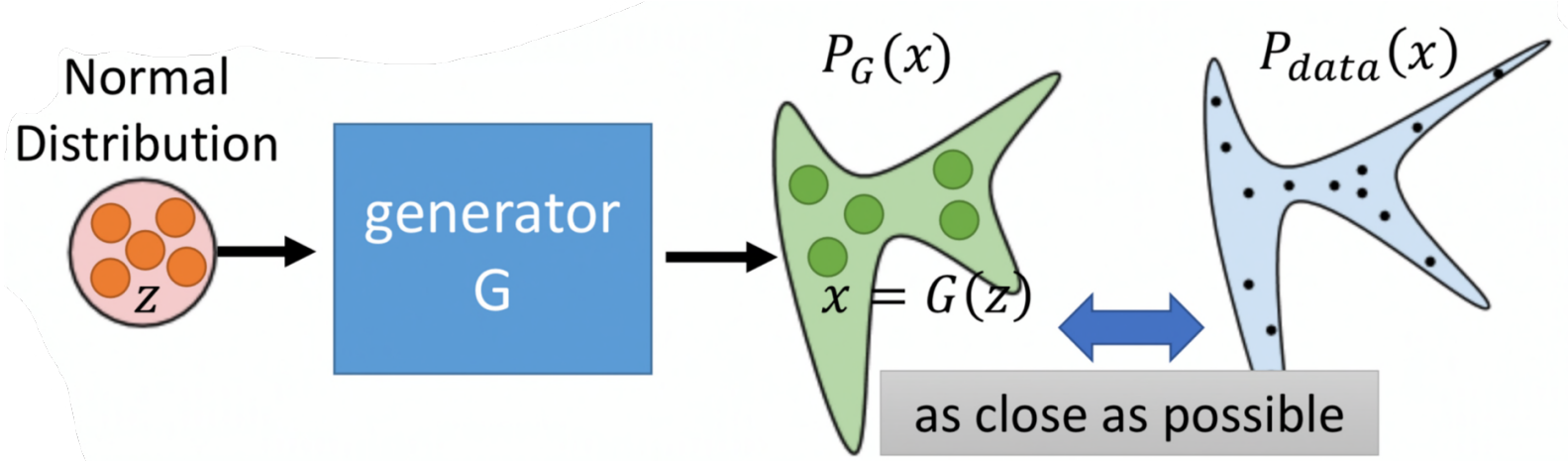
- Generator: A map G from random noise space Z to data space X.

$$G: \ Z \longrightarrow X, \ \text{random noise sample } z \mapsto \text{generated data } G(z)$$

- Discriminator: A valuation funtion D from X to a score ranged in [0, 1].

$$D: \ X \longrightarrow [0, \ 1], \quad x_{true} \xrightarrow{\text{close}} 1, \ x_{fake} \xrightarrow{\text{close}} 0$$

- Cheat-Distinguish Process



GAN: Theorical Analysis

- Optimization Objective

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(X)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Intuitively: D good $\uparrow$, G good $\downarrow$

GAN: Theorical Analysis

- Optimization Objective

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(X)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

Intuitively: D good $\uparrow$, G good $\downarrow$

- Realization of a Minimax Process
  - Max:

Theorem 1

For fixed G, the optimal discriminator D is $D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$

GAN: Theorical Analysis

- Realization of a Minimax Process (cont'd)
  - Max:

Proof.

$$V(D,G) = \int_x p_{data}(x)\log(D(x))dx + \int_z p_z(z)\log(1-D(G(z)))dz$$
$$= \int_x p_{data}(x)\log(D(x))dx + p_{data}(x)\log(1-D(x))dx$$

$a\log x + b\log(1-x)$ achieves maximum when $x = \frac{a}{a+b}$ □

  - Min:

Theorem 2

The global minimum of $C(G) = \max_D V(D,G)$ is achieved iff $p_g = p_{data}$.

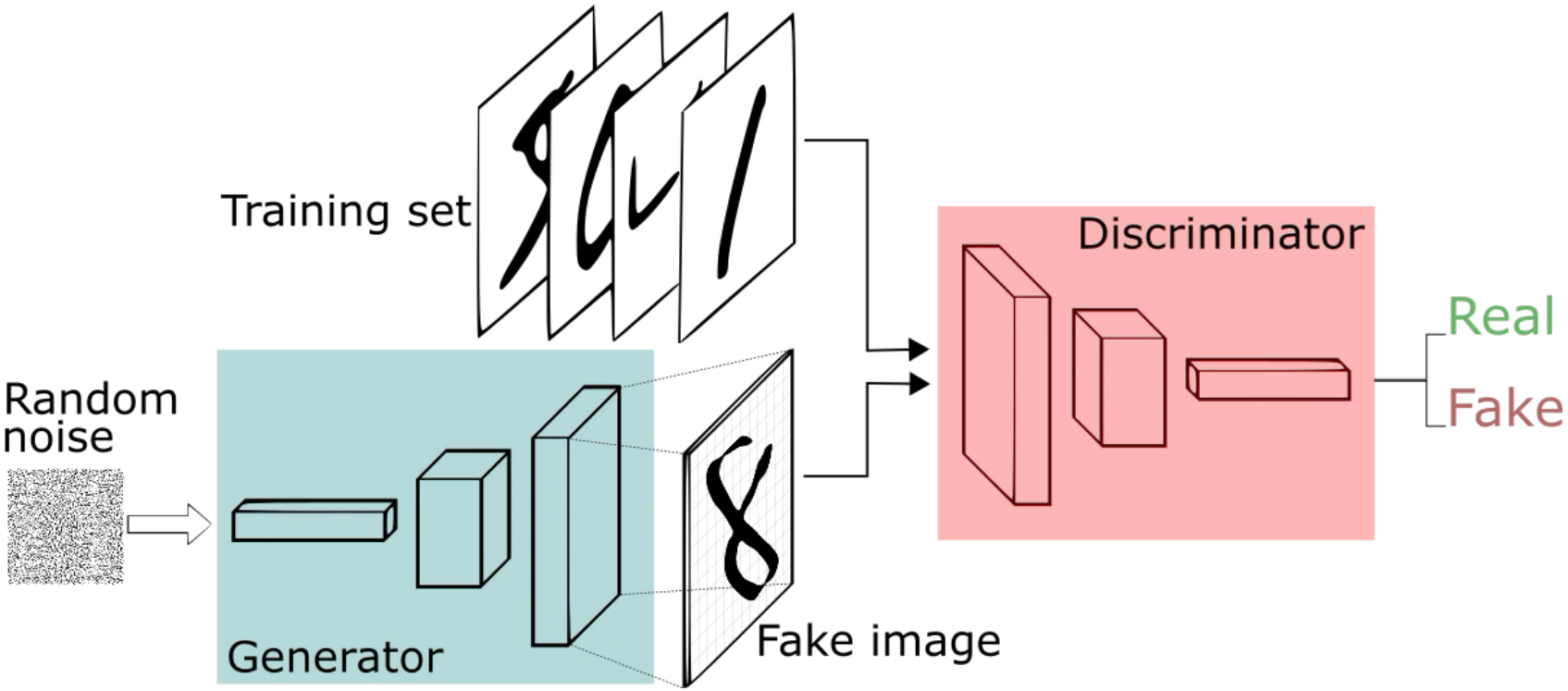
GAN: Theorical Analysis

- Realization of a Minimax Process (cont'd)
  - Min:

Proof.

$$
\begin{aligned}
C(G) &= \max_D V(D,G) \\
&= \mathbb{E}_{x\sim p_{data}}[\log D_G^*(x)] + \mathbb{E}_{z\sim p_z}[\log(1-D_G^*(G(z)))] \\
&= \mathbb{E}_{x\sim p_{data}}[\log D_G^*(x)] + \mathbb{E}_{x\sim p_g}[\log(1-D_G^*(G(z)))] \\
&= \mathbb{E}_{x\sim p_{data}}[\log \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}] + \mathbb{E}_{x\sim p_g}[\log \frac{p_g(x)}{p_{data}(x)+p_g(x)}] \\
&= -\log 4 + \mathrm{KL}(p_{data}|\frac{p_{data}+p_g}{2}) + \mathrm{KL}(p_g|\frac{p_{data}+p_g}{2}) \\
&= -\log 4 + 2\mathrm{JS}(p_{data}|p_g)
\end{aligned}
$$

GAN: Learning Process



GAN: Algorithm

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 $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

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

- Generative Models
- VAE
- GAN
- **Similarities between VAE and GAN**
- Drawbacks of VAE and GAN
- Solutions by Wasserstein Metric
  - Wasserstein GAN
  - Wasserstein AE
- Unified Theory and Stronger Model
- Further Problems

Similarity: VAE & GAN

- Measure the discrepancy between distributions
  - VAE
    There is a relation between MLE and KL divergence:

$$\begin{aligned} \mathrm{KL}(\mathbb{P}_x,\ \mathbb{P}_g) &= \int_x p_x \log \frac{p_x}{p_g} = \frac{1}{N} \sum_{x^{(i)} \in D} \log \frac{p_x(x^{(i)})}{p_g(x^{(i)})} \\ &= -\frac{1}{N} \sum_{x^{(i)}} \log p_g(x^{(i)}) + \frac{1}{N} \sum_{x^{(i)}} \log p_x(x^{(i)}) \end{aligned}$$

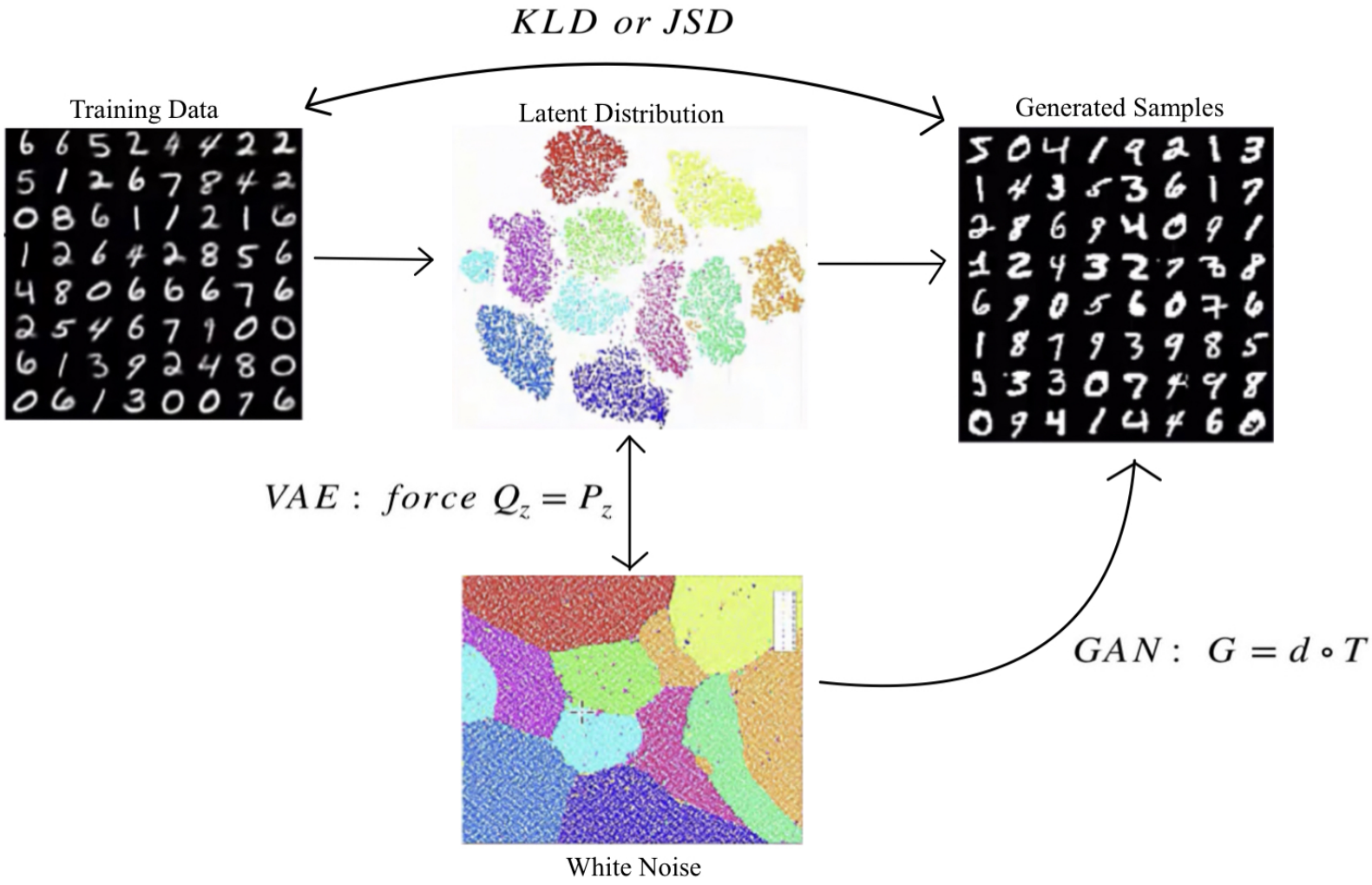
  - GAN

$$\log D^*(x) + \log(1 - D^*(G(z))) = \mathrm{JS}(\mathbb{P}_x,\ \mathbb{P}_g) + \log 2$$

Remark: Both learn distributions by minimizing certain distances.

Similarity: VAE & GAN

- Encoder-Decoder structure



Outline

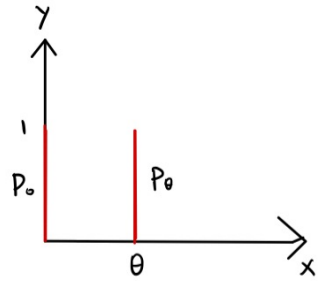
- Generative Models
- VAE
- GAN
- Similarities between VAE and GAN
- **Drawbacks of VAE and GAN**
- Solutions by Wasserstein Metric
  - Wasserstein GAN
  - Wasserstein AE
- Unified Theory and Stronger Model
- Further Problems

Drawback of GAN: Gradient Vanishing

- An easy example[3]

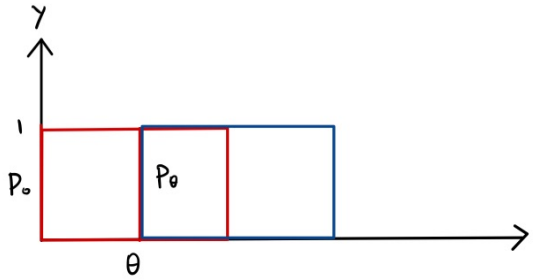
An easy example

Example 1 (Learning parallel lines). Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,



- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$,
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
- $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$

When D^*, $\nabla_\theta G = 0$ *(no intersect case)*

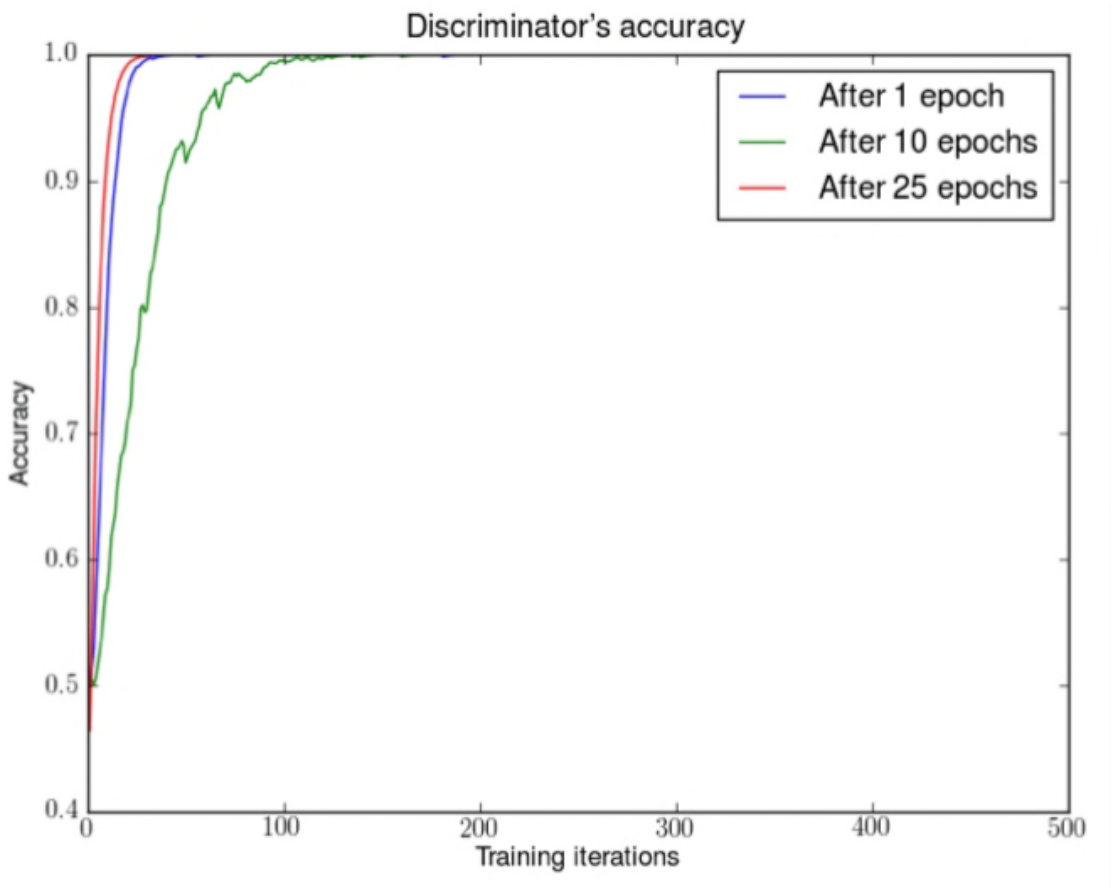


When D^*, $\nabla_\theta G \neq 0$ *(intersect case)*

[3] Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks. Stat, 2017, 1050.

Drawback of GAN: Gradient Vanishing

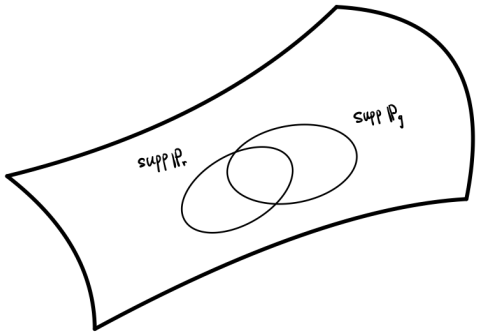
- Experiment: With different G fixed, train D and test D's accuracy:



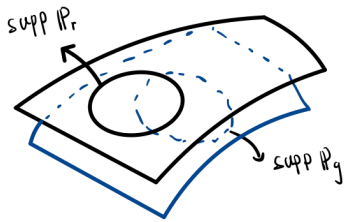
- Observation: Perfect discriminator D_G^* can always be trained.

Drawback of GAN: Gradient Vanishing

- Perfect Discriminator Theorem
  In fact, D_G^* can always be trained theoritically.
  - Key Lemmas:
    1. $\mathbb{P}_g$ and $\mathbb{P}_{data}$ are supported on low dimensional manifolds.
    2. In almost all cases, $\text{supp}\mathbb{P}_g \cap \text{supp}\mathbb{P}_{data} = \varnothing$ *or N*, *N null set*.
  - Examples:



$$\mathbb{P}(happen) = 0$$



$$\mathbb{P}(happen) = 1$$

Drawback of GAN: Gradient Vanishing

- Perfect Discriminator Theorem

Theorem 3

If two distributions $\mathbb{P}_{data}$ and $\mathbb{P}_g$ have support contained on two disjoint compact subsets M and P respectively, then there is a smooth optimal discrimator D^* that has accuracy 1 and $\nabla_x D^*(x) = 0$ for all $x \in M \cup P$.

Theorem 4

Let $\mathbb{P}_{data}$ and $\mathbb{P}_g$ be two distributions that have support contained in two closed manifolds M and P that don't perfectly align and don't have full dimension. We further assume that $\mathbb{P}_{data}$ and $\mathbb{P}_g$ are continuous in their respective manifolds, meaning that if there is a set A with measure 0 in M, then $\mathbb{P}_{data}(A) = 0$ (and analogously for $\mathbb{P}_g$). Then, there exists an optimal discriminator D^* that has accuracy 1 and for almost any x in M or P, D^* is smooth in a neighbourhood of x and $\nabla_x D^*(x) = 0$.

Drawback of GAN: Gradient Vanishing

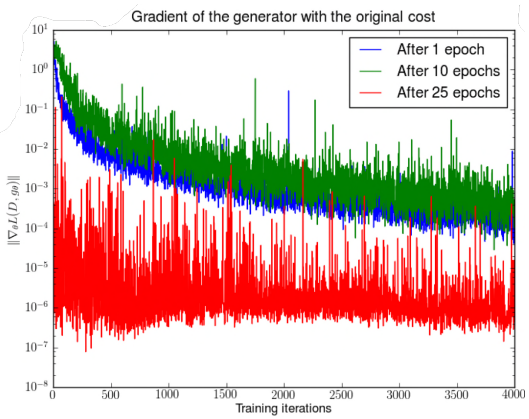
- Gradient Vanishing Theorem

Theorem 5

Let $g_\theta : Z \to X$ be a differentiable function that induces a distribution $\mathbb{P}_g$. Let $\mathbb{P}_{data}$ be the real data distribution. Let D be a differentiable discriminator. If the conditions of Theorems 3 or 4 are satisfied, $|D - D^*| < \epsilon$, and $\mathbb{E}_{z\sim p(z)}|J_\theta g_\theta(z)|_2^2 \leqslant M^2$, then

$$\nabla_\theta |\mathbb{E}_{z\sim p(z)}[log(1 - D(g_\theta(z)))]|_2 \leqslant M\frac{\epsilon}{1-\epsilon}$$

- Experimental support



Drawback of GAN: Instability

- log(1-s) case:

$$V = \cancel{E_{x \sim P_{data}}[\log D(x)]}$$

$$+E_{x \sim P_G}[\log(1 - D(x))]$$

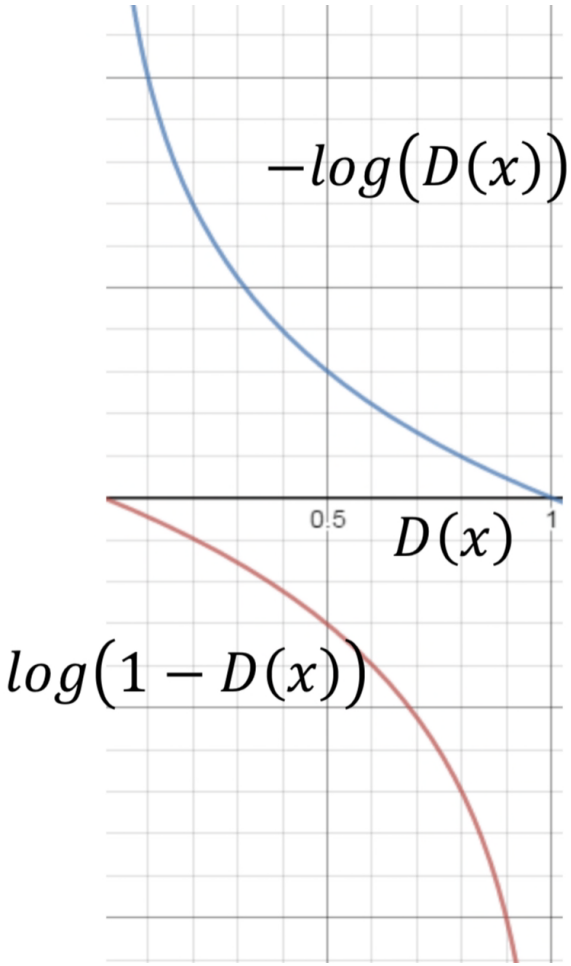
Slow at the beginning

Minimax GAN (MMGAN)

$$V = E_{x \sim P_G}[-\log(D(x))]$$

Real implementation:
label x from P_G as positive

Non-saturating GAN (NSGAN)



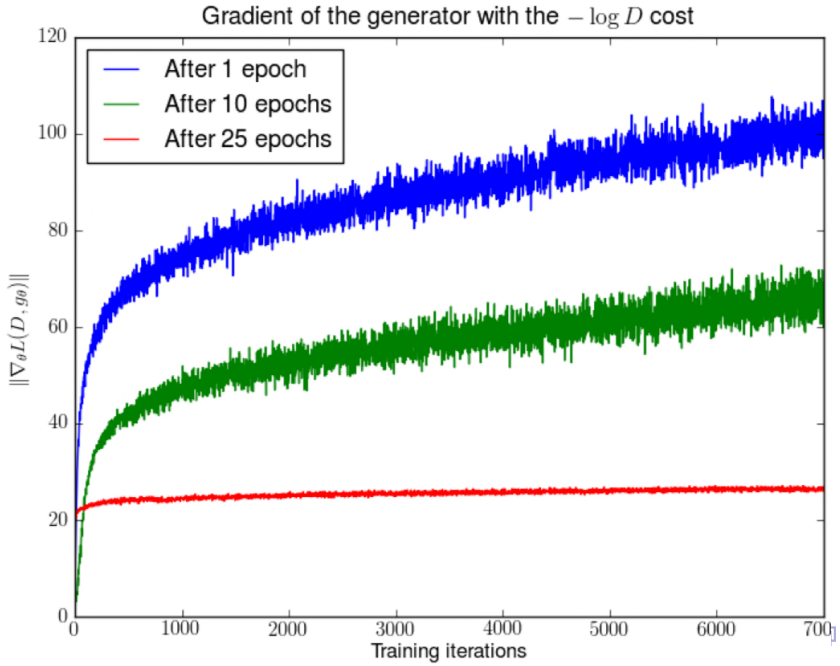
Drawback of GAN: Instability

- -log(s) case:

Theorem 6

Let $\mathbb{P}_{data}$ and $\mathbb{P}_{g_\theta}$ be two continuous distributions, with densities $\mathbb{P}_{data}$ and $\mathbb{P}_{g_\theta}$ respectively. Let D^* be the optimal discriminator, fixed for a value θ_0. Therefore,

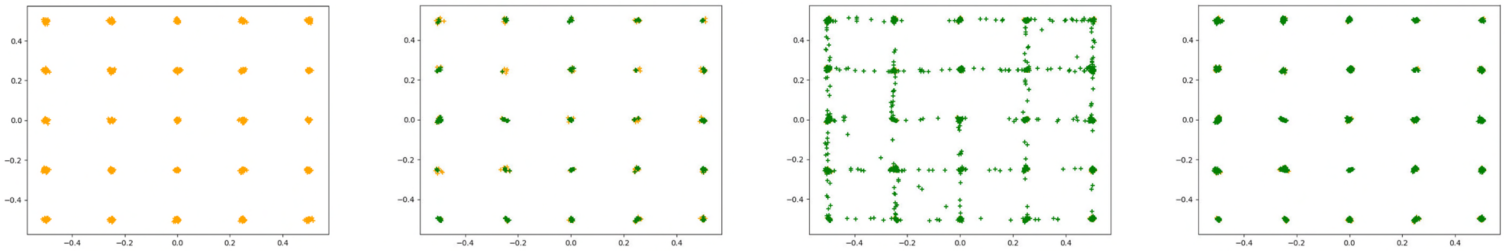
$$E_{z\sim p(z)}[-\nabla_\theta log D^*(g_\theta(z))|_{\theta=\theta_0}] = \nabla_\theta[KL(P_{g_\theta}|P_{data}) - 2JSD(P_{g_\theta}|P_{data})]|_{\theta=\theta_0}$$



Drawback of GAN: Mode Collapse & Mode Mixture

- Mode Collapse

- Mode Mixture



Drawback of VAE: Weak Regularizer

We sample from $Q(Z)$ when training and $P(Z)$ in testing in latent space, hence $Q(Z)$ must match $P(Z)$.

- Experimental Test:[4]

	ELBO	Avg. KL	Mutual info. ②	Marg. KL ③
2D latents	-129.63	7.41	7.20	0.21
10D latents	-88.95	19.17	10.82	8.35
20D latents	-87.45	20.2	10.67	9.53

$Q(Z)$ generally does not match $P(Z)$ in VAE.

- Reason: Weak Regularizer

$$\text{ELBO}^{(i)} = \mathcal{L}(\theta,\ \phi,\ x^{(i)}) = -\text{KL}(q(z|x^{(i)},\ p_\theta(z)) + \mathbb{E}_{q_\phi(z|x^{(i)})}(p(x^{(i)})|z)$$

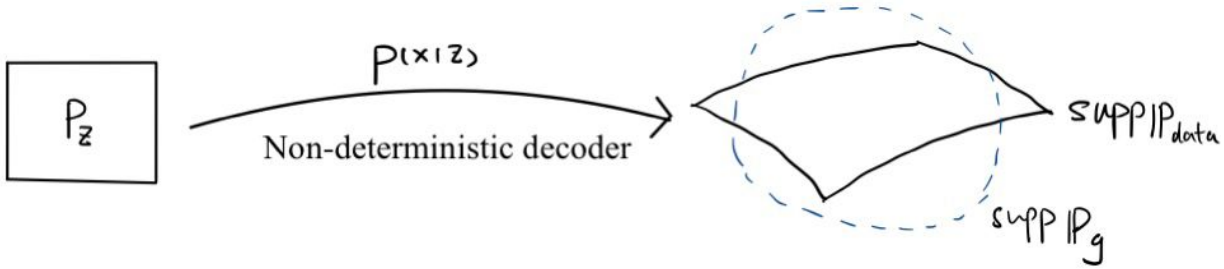
where $\text{KL}(q|p)$ as a regularizer is too weak for $Q(Z) = P(Z)$.

[4] Locatello F., et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. 2018.

Drawback of VAE: Blurry

Generated data are not concertrated on a low dimensional manifold!
This leads to blurry.

- Explanation



- Experiment[5]

Table 2: Quantitative comparison with FID

	Adversarial				Non-Adversarial		Reference	
Dataset	NS GAN	LSGAN	WGAN	BEGAN	VAE	GLANN	AE	Ours
MNIST	6.8±0.5	7.8±0.6	6.7±0.4	13.1±1.0	23.8±0.6	8.6±0.1	5.5	**6.2±0.2**
Fansion	26.5±1.6	30.7±2.2	21.5±1.6	22.9±0.9	58.7±1.2	13.0±0.1	4.7	**10.1±0.3**
CIFAR-10	58.5±1.9	87.1±47.5	55.2±2.3	71.4±1.6	65.4±0.2	46.5±0.2	28.2	**38.3±0.5**
CelebA	55.0±3.3	53.9±2.8	41.3±2.0	**38.9±0.9**	85.7±3.8	46.3±0.1	67.5	68.4± 0.5

[5] An D., et al. AE-OT-GAN: Training GANs from data specific latent distribution. 2020.

Summary of Drawbacks