DSC291: Advanced Statistical Natural Language Processing

Advanced Topics: Generative Adversarial Learning

Zhiting Hu

Lecture 18, May 26, 2022

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Outline

- GANs (for text)

- 3 Paper presentations (15 x 3 mins)

Generative Adversarial Networks

Generative modeling

- In generative modeling, we'd like to train a network that models a distribution, such as a distribution over images.
- One way to judge the quality of the model is to sample from it.
- This field has seen rapid progress:

2009

2015

2018

Courtesy: Grosse CSC321 Lecture 19

Generative modeling

- In generative modeling, we'd like to train a network that models a distribution, such as a distribution over images.
- One way to judge the quality of the model is to sample from it.
- This field has seen rapid progress:

Figure courtesy: Ian Goodfellow

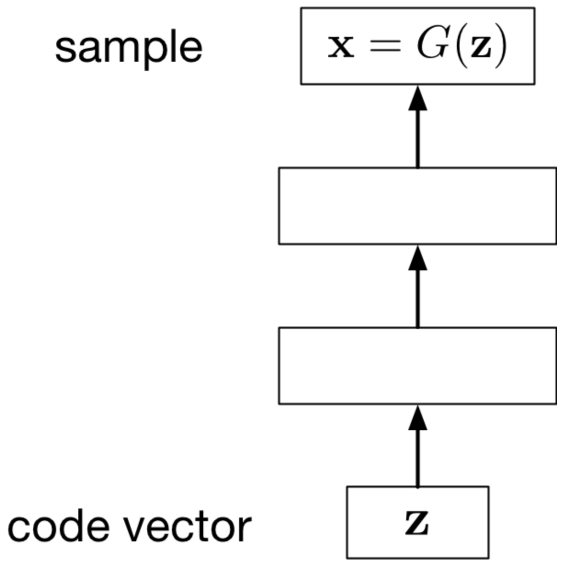
Generative modeling

- Modern approaches to generative modeling:
  - Variational Auto-encoder (Lecture #8)
  - Auto-regressive models (e.g., language model) (Lecture #3)
  - Generative adversarial networks (today)
  - Flow-based models, diffusion models (not covered)

Courtesy: Grosse CSC321 Lecture 19

Implicit Generative Models

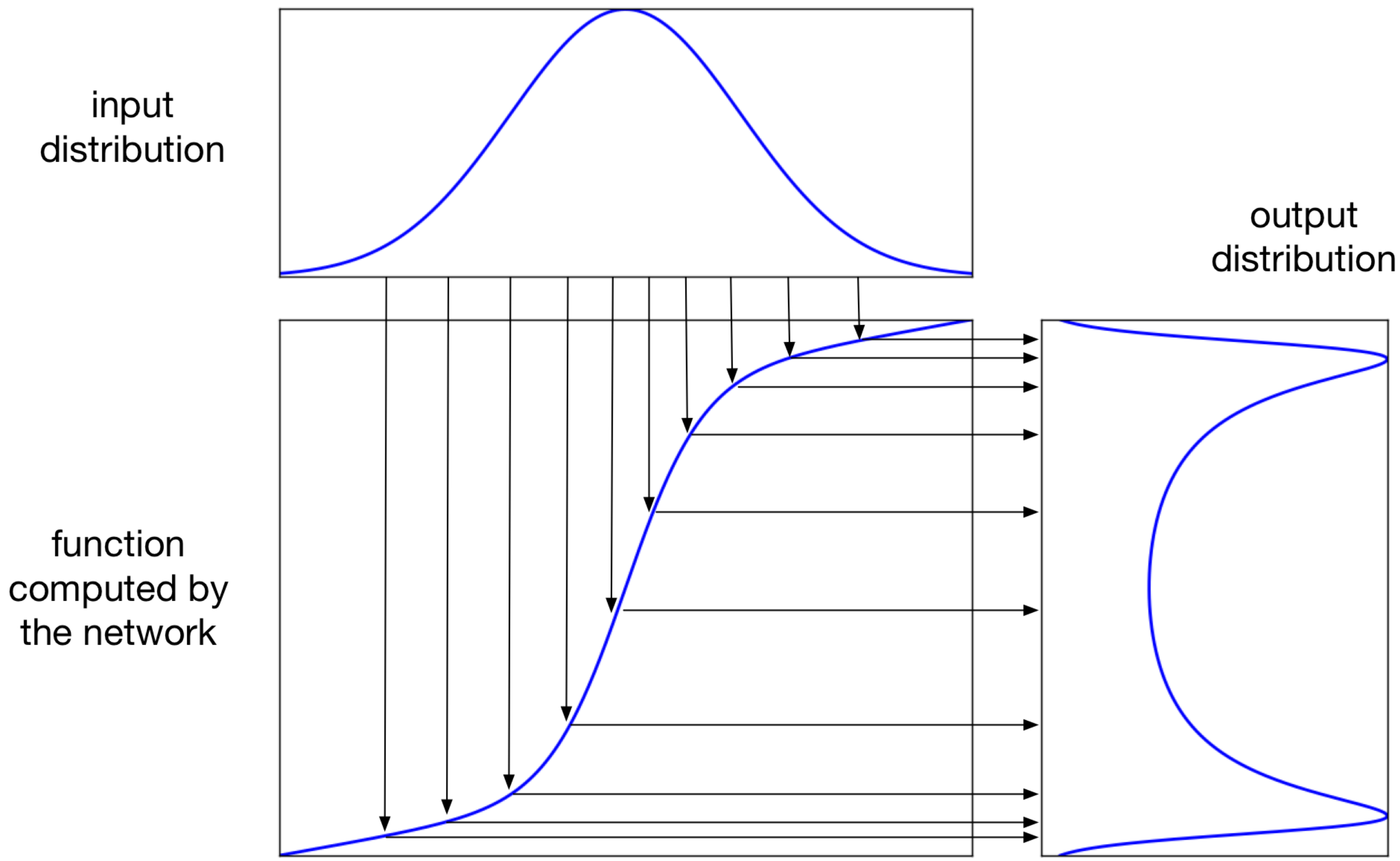
- Implicit generative models implicitly define a probability distribution
- Start by sampling the code vector $\mathbf{z}$ from a fixed, simple distribution (e.g. spherical Gaussian)
- The generator network computes a differentiable function G mapping $\mathbf{z}$ to an $\mathbf{x}$ in data space



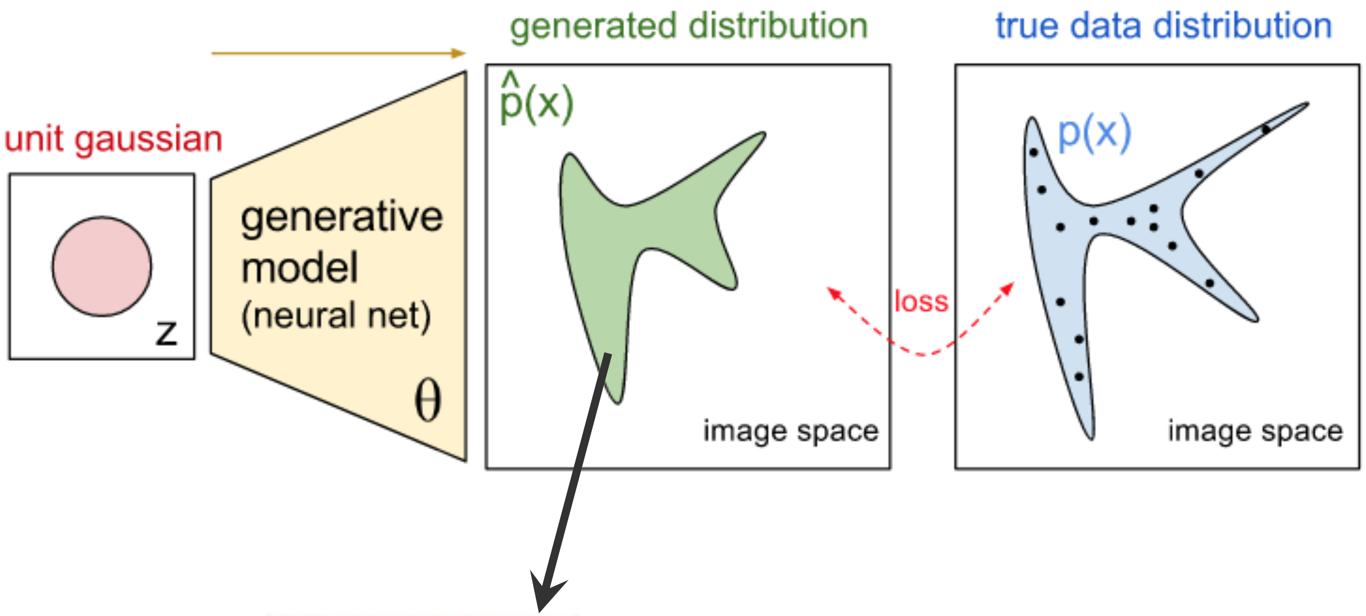
- a stochastic process to simulate data $\boldsymbol{x}$
- Intractable to evaluate likelihood

Implicit Generative Models

A 1-dimensional example:



Implicit Generative Models



https://blog.openai.com/generative-models/

Implicit Generative Models

- The advantage of implicit generative models: if you have some criterion for evaluating the quality of samples, then you can compute its gradient with respect to the network parameters, and update the network's parameters to make the sample a little better
- The idea behind **Generative Adversarial Networks (GANs)**: train two different networks
  - The generator network tries to produce realistic-looking samples
  - The discriminator network tries to figure out whether an image came from the training set or the generator network
- The generator network tries to fool the discriminator network

Generative Adversarial Nets (GANs)

- Generative model $\boldsymbol{x} = G_\theta(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$
  - Maps noise variable $\boldsymbol{z}$ to data space $\boldsymbol{x}$
  - Defines an implicit distribution over $\boldsymbol{x}$: $p_{g_\theta}(\boldsymbol{x})$
- Discriminator $D_\phi(\boldsymbol{x})$
  - Output the probability that $\boldsymbol{x}$ came from the data rather than the generator

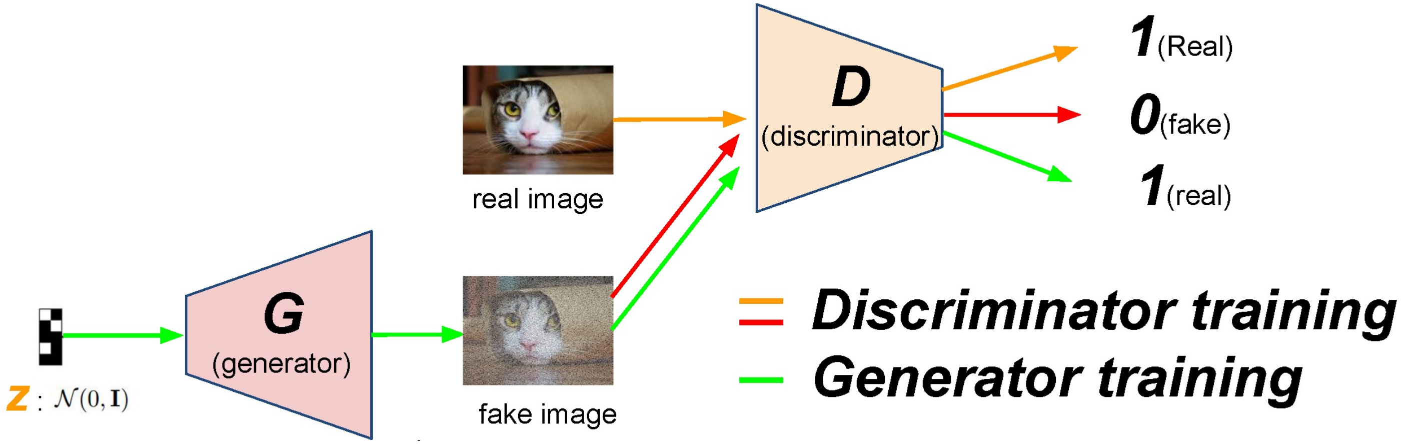


Figure courtesy: Kim

Generative Adversarial Nets (GANs)

- Learning
  - A minimax game between the generator and the discriminator
  - Train D to maximize the probability of assigning the correct label to both training examples and generated samples
  - Train G to fool the discriminator

$$\max_D \mathcal{L}_D = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z}),\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log(1 - D(\boldsymbol{x}))\right]$$
$$\min_G \mathcal{L}_G = \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z}),\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log(1 - D(\boldsymbol{x}))\right].$$

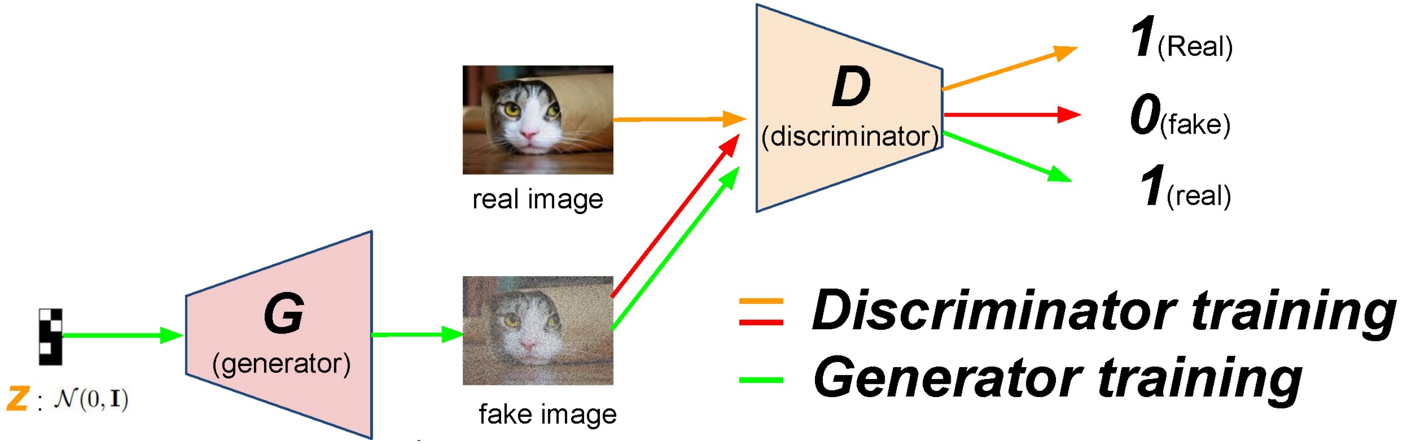
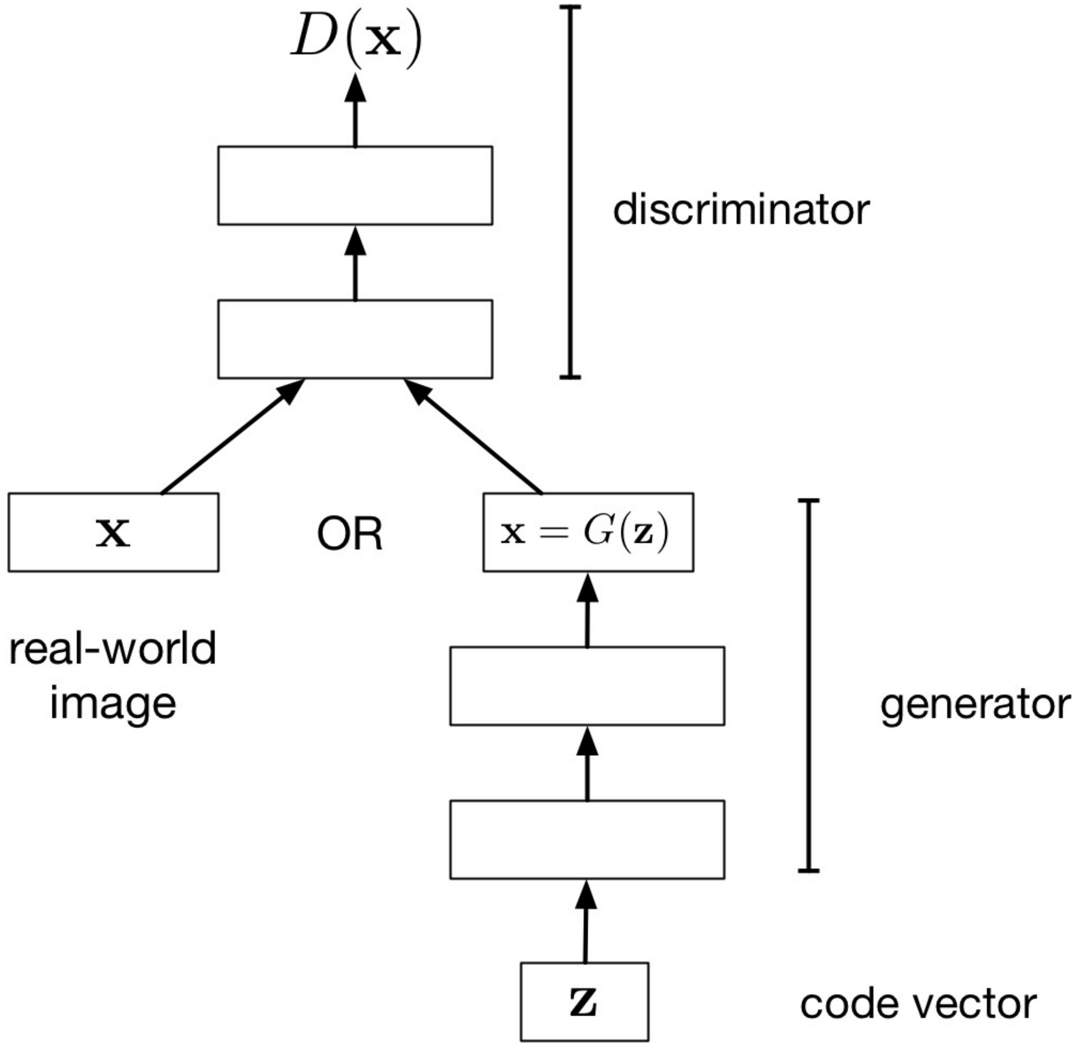


Figure courtesy: Kim

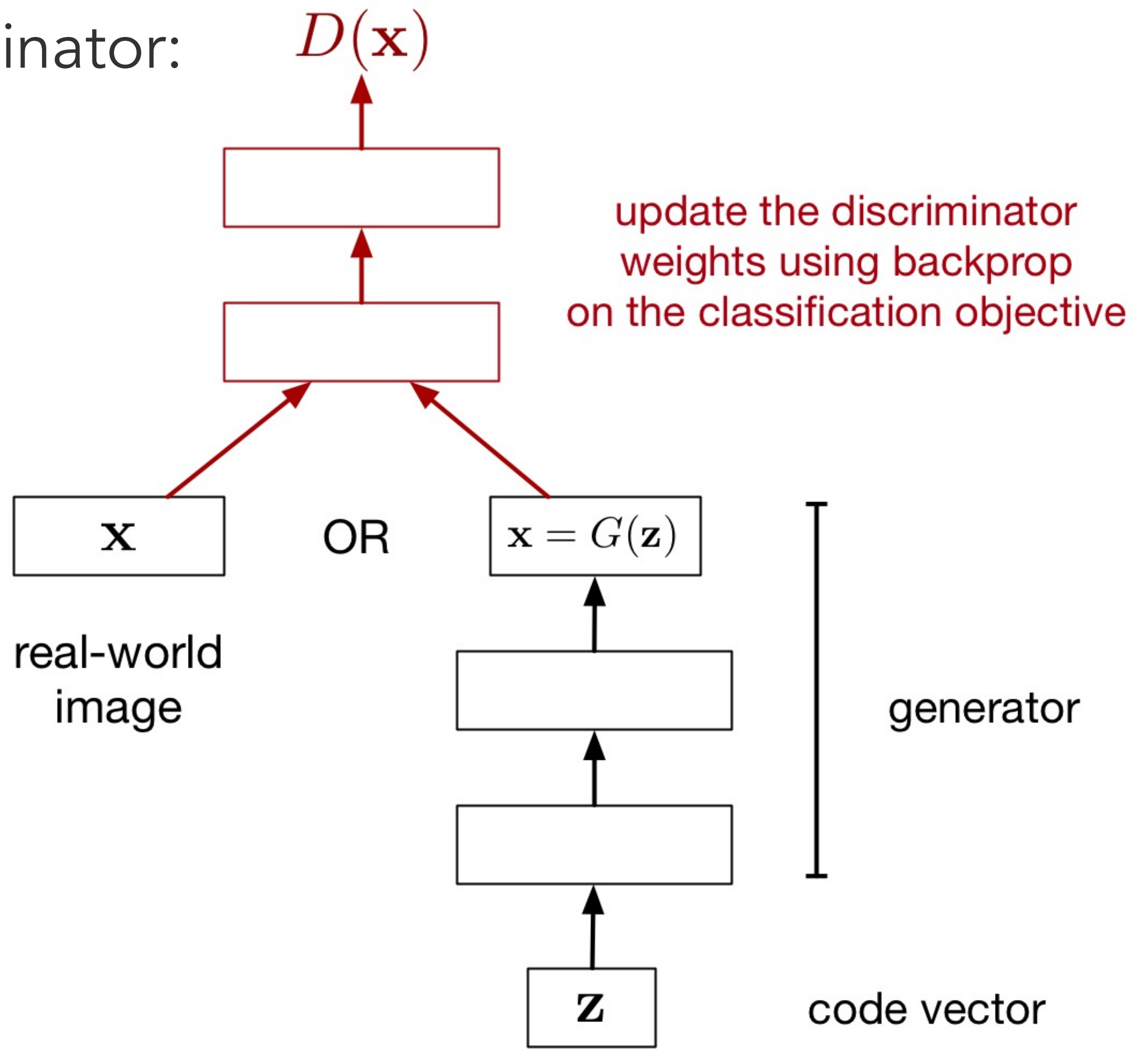
Generative Adversarial Nets (GANs)



Courtesy: Grosse CSC321 Lecture 19

Generative Adversarial Nets (GANs)

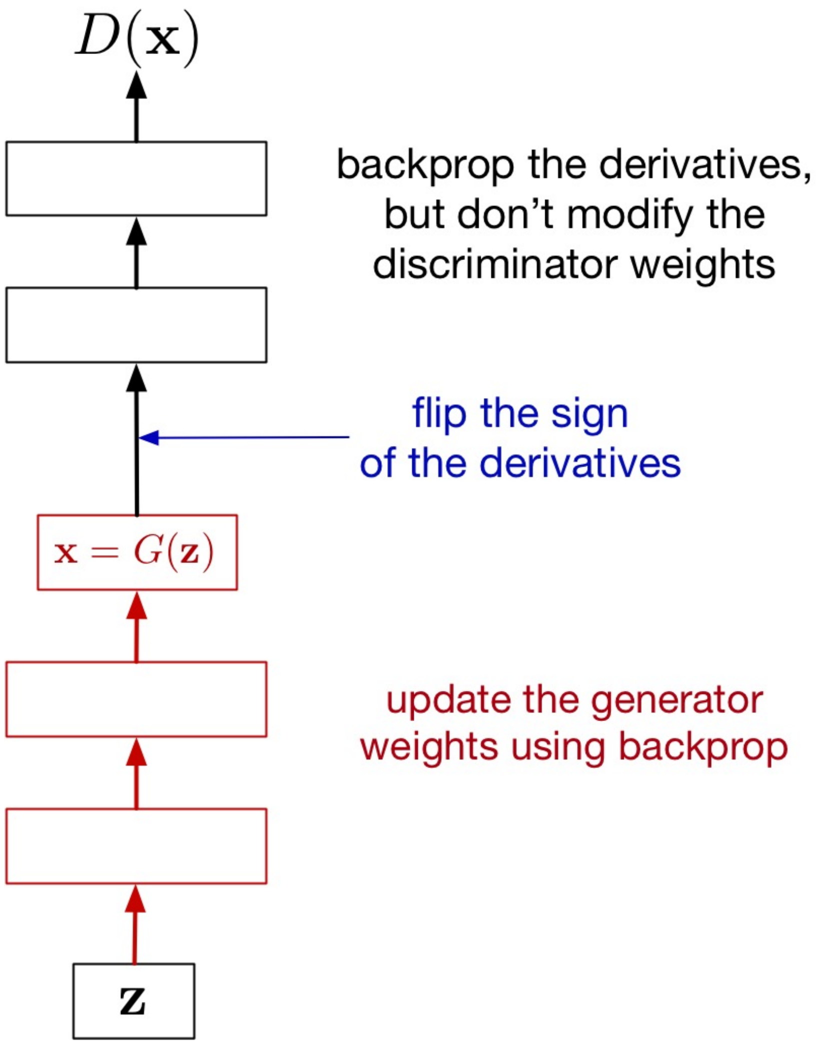
Updating the discriminator:



Courtesy: Grosse CSC321 Lecture 19

Generative Adversarial Nets (GANs)

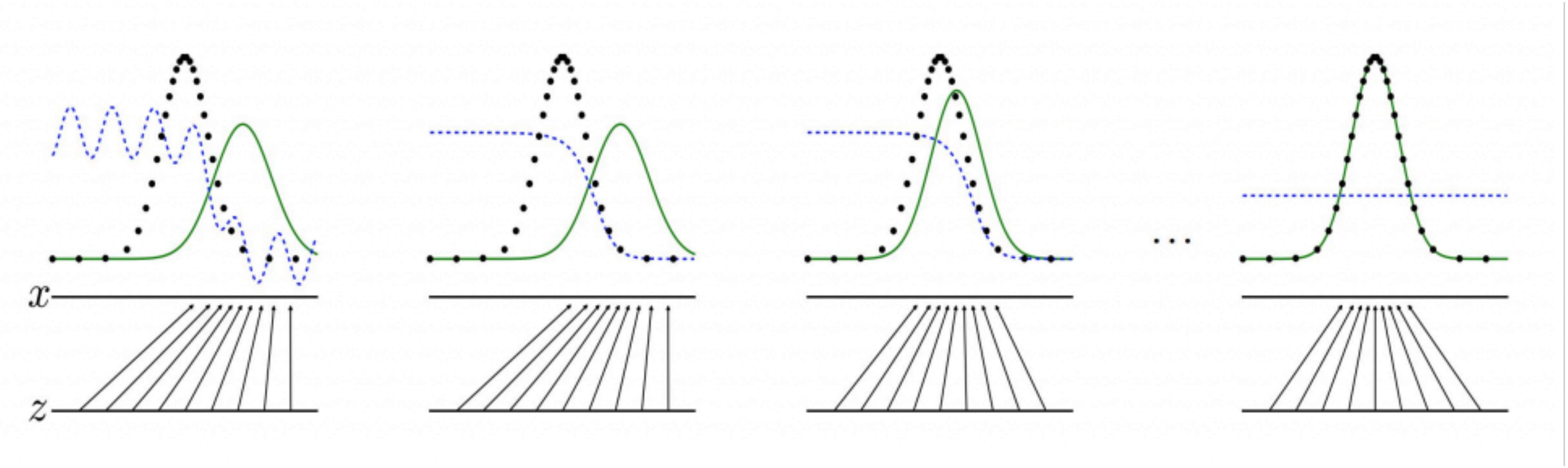
Updating the generator:



Courtesy: Grosse CSC321 Lecture 19

Generative Adversarial Nets (GANs)

Alternating training of the generator and discriminator:



Courtesy: Grosse CSC321 Lecture 19

Optimality of GANs

- Objectives:

$$\max_D \mathcal{L}_D = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z}),\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log(1 - D(\boldsymbol{x}))\right]$$
$$\min_G \mathcal{L}_G = \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z}),\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\log(1 - D(\boldsymbol{x}))\right].$$

- Global optimality: $p_g = p_{data}$
- Proof:

Optimality of GANs

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2}$$

[Goodfellow et al., 2014]

Optimality of GANs

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2}$$

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

$$\begin{aligned} V(G, D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned} \tag{3}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \rightarrow a\log(y) + b\log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$.

[Goodfellow et al., 2014]

Optimality of GANs

- The minimax game can now be reformulated as

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

[Goodfellow et al., 2014]

Optimality of GANs

- The minimax game can now be reformulated as

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} [\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g} [\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \right] + \mathbb{E}_{\boldsymbol{x} \sim p_g} \left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \right]
\end{aligned}
$$

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 $-\log 4$.*

[Goodfellow et al., 2014]

Optimality of GANs

- The minimax game can now be reformulated as

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right]
\end{aligned}
$$

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 $-\log 4$.*

$$
\begin{aligned}
C(G) &= -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) \\
&= -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right) \quad \text{Jensen-Shannon Divergence}
\end{aligned}
$$

[Goodfellow et al., 2014]

A better loss function

- We introduced the minimax cost function for the generator:

$$\mathcal{J}_G = \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))]$$

- One problem with this is saturation.
- Here, if the generated sample is really bad, the discriminator's prediction is close to 0, and the generator's cost is flat.

A better loss function: non-saturating GAN

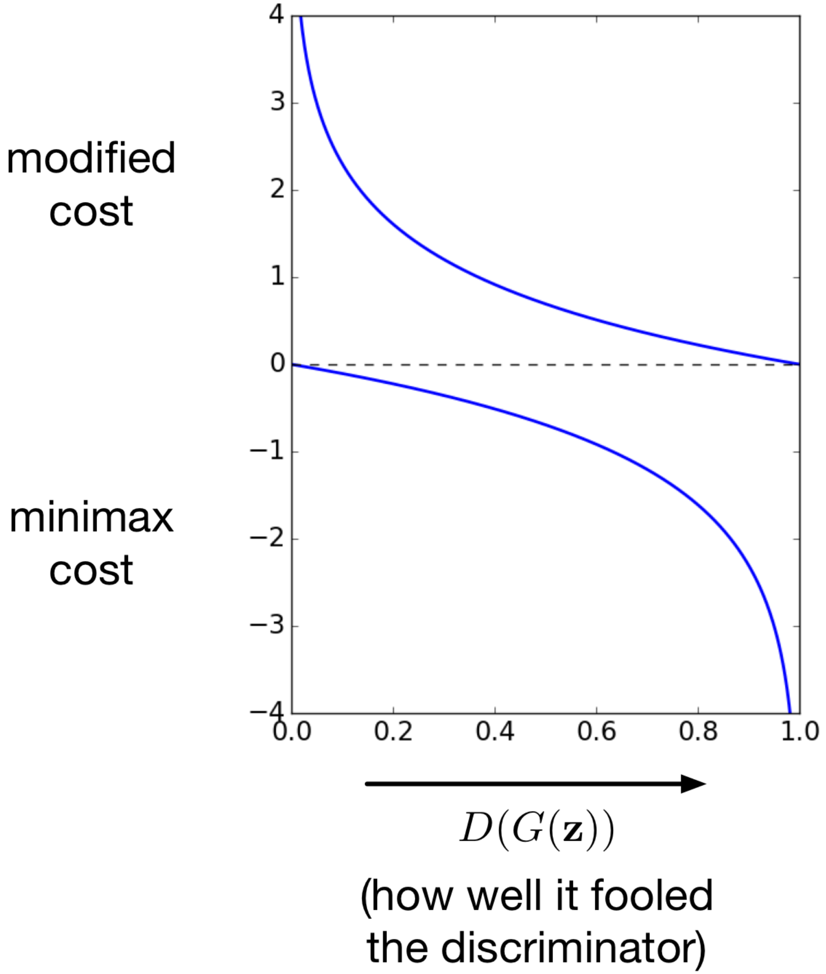
- Original minimax cost:

$$\mathcal{J}_G = \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))]$$

- Modified generator cost:

$$\mathcal{J}_G = \mathbb{E}_{\mathbf{z}}[-\log D(G(\mathbf{z}))]$$

- This fixes the saturation problem.



Wasserstein GAN (WGAN)

- If our data are on a low-dimensional manifold of a high dimensional space, the model's manifold and the true data manifold can have a negligible intersection in practice

[Arjovsky et al., 2017] Slide adapted from bhiksha

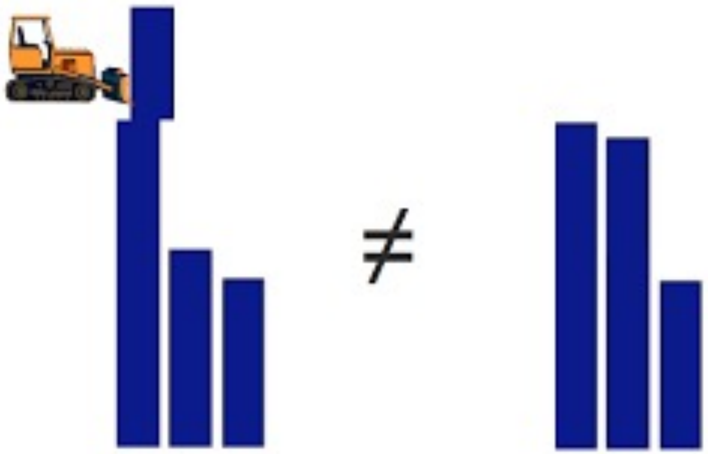
Wasserstein GAN (WGAN)

- If our data are on a low-dimensional manifold of a high dimensional space, the model's manifold and the true data manifold can have a negligible intersection in practice
- The loss function and gradients may not be continuous and well behaved

[Arjovsky et al., 2017] Slide adapted from bhiksha

Wasserstein GAN (WGAN)

- If our data are on a low-dimensional manifold of a high dimensional space, the model's manifold and the true data manifold can have a negligible intersection in practice
- The loss function and gradients may not be continuous and well behaved
- The Wasserstein Distance is well defined
  - Earth Mover's Distance
  - Minimum transportation cost for making one pile of dirt in the shape of one probability distribution to the shape of the other distribution



[Arjovsky et al., 2017] Slide adapted from bhiksha

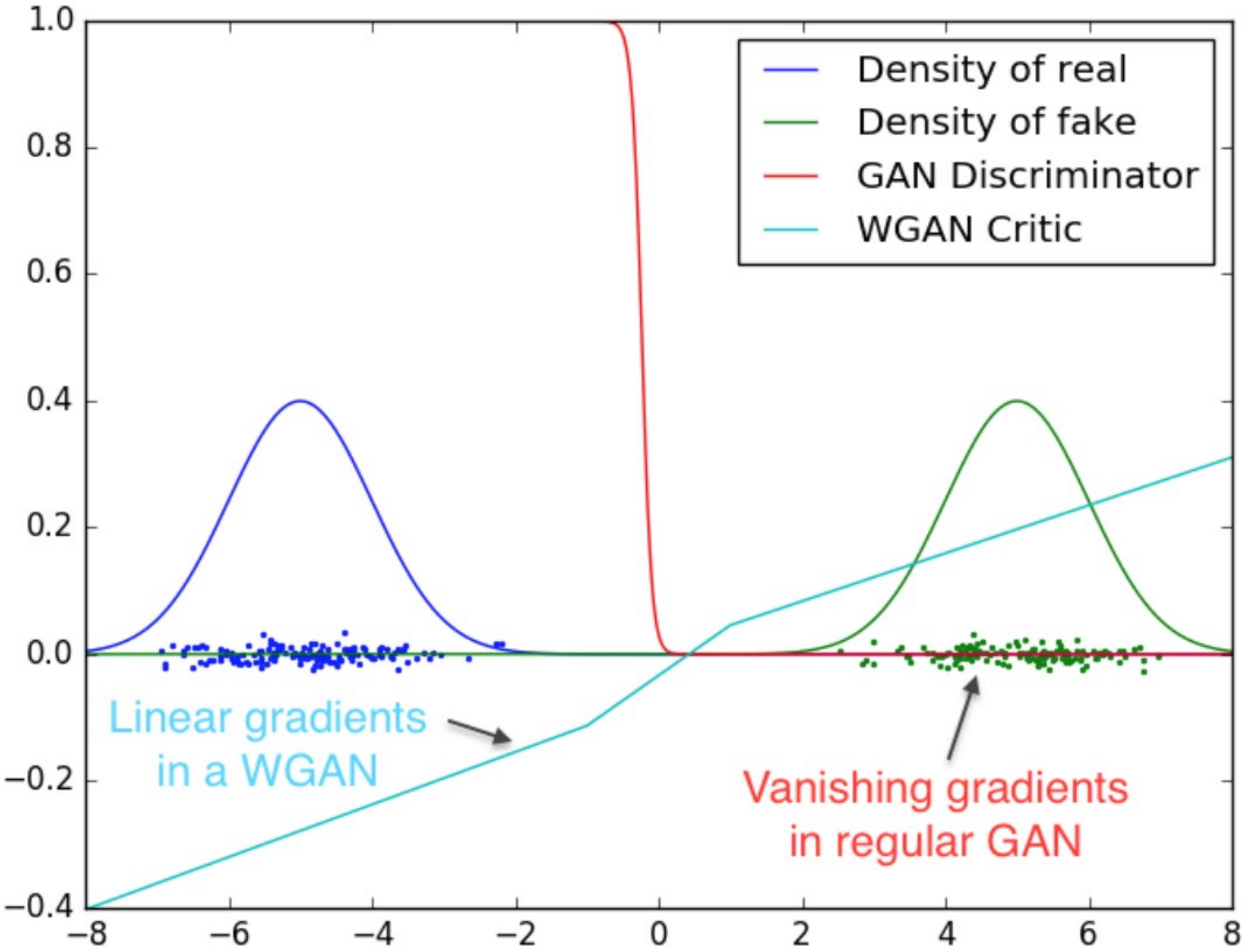
Wasserstein GAN (WGAN)

- Objective

$$W\left(p_{data}, p_g\right) = \frac{1}{K} \sup_{||D||_L \leq K} \mathrm{E}_{x \sim p_{data}}[D(x)] - \mathrm{E}_{x \sim p_g}[D(x)]$$

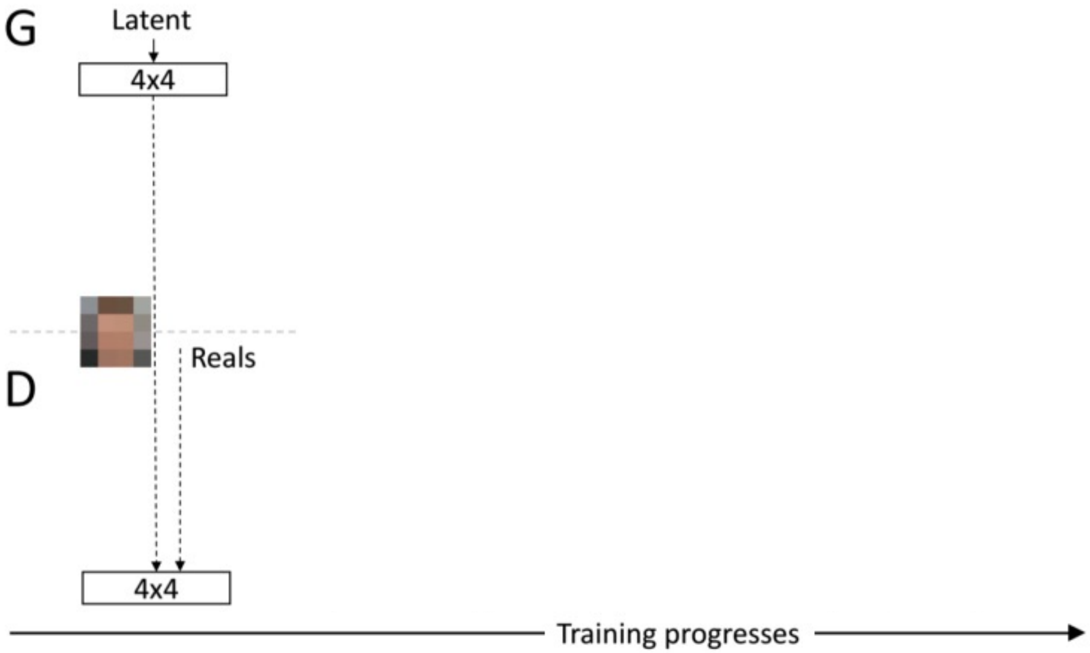
- $||D||_L \leq K$: K- Lipschitz continuous
- Use gradient-clipping to ensure D has the Lipschitz continuity

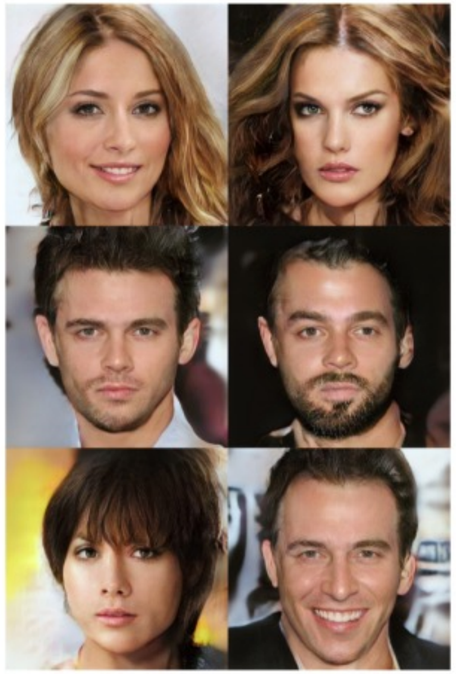
WGAN vs Vanilla GAN



Progressive GAN

Low resolution images



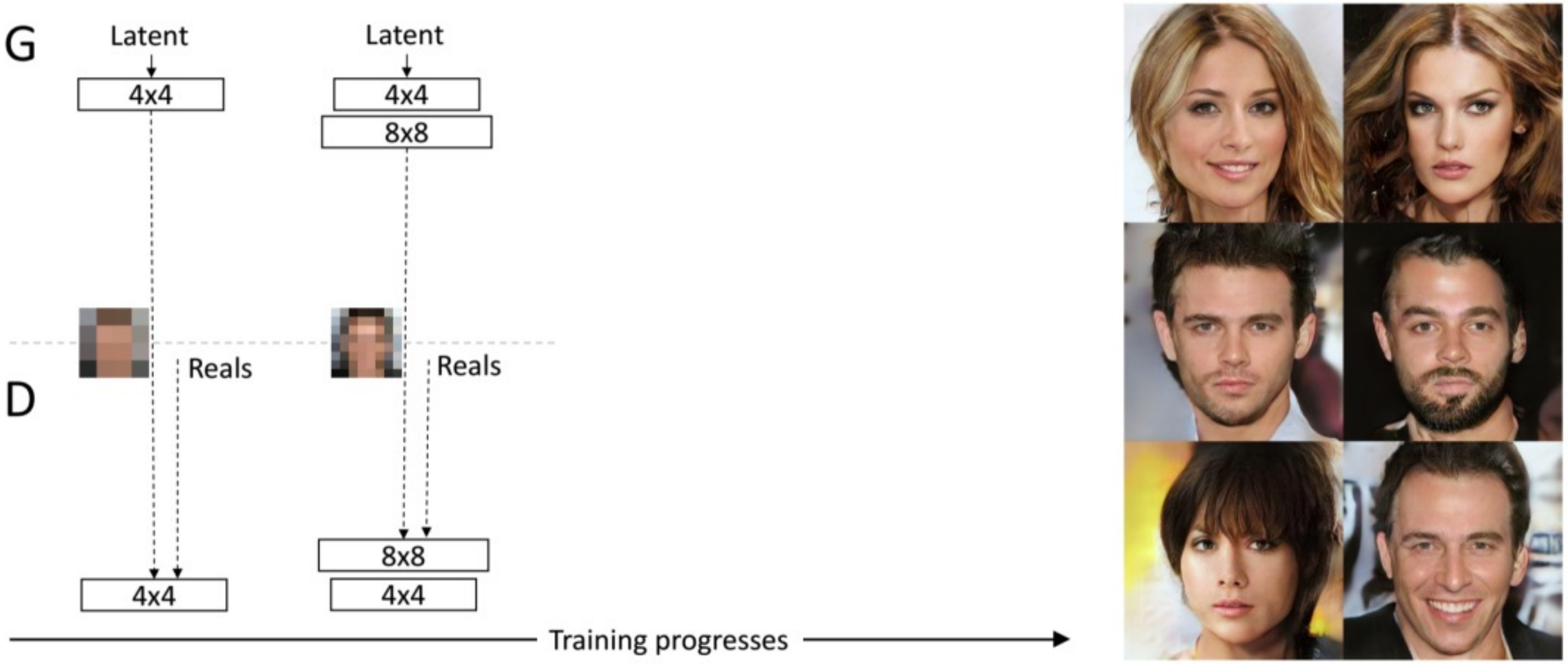


[Karras et al., 2018]

Progressive GAN

Low resolution images

add in
additional
layers



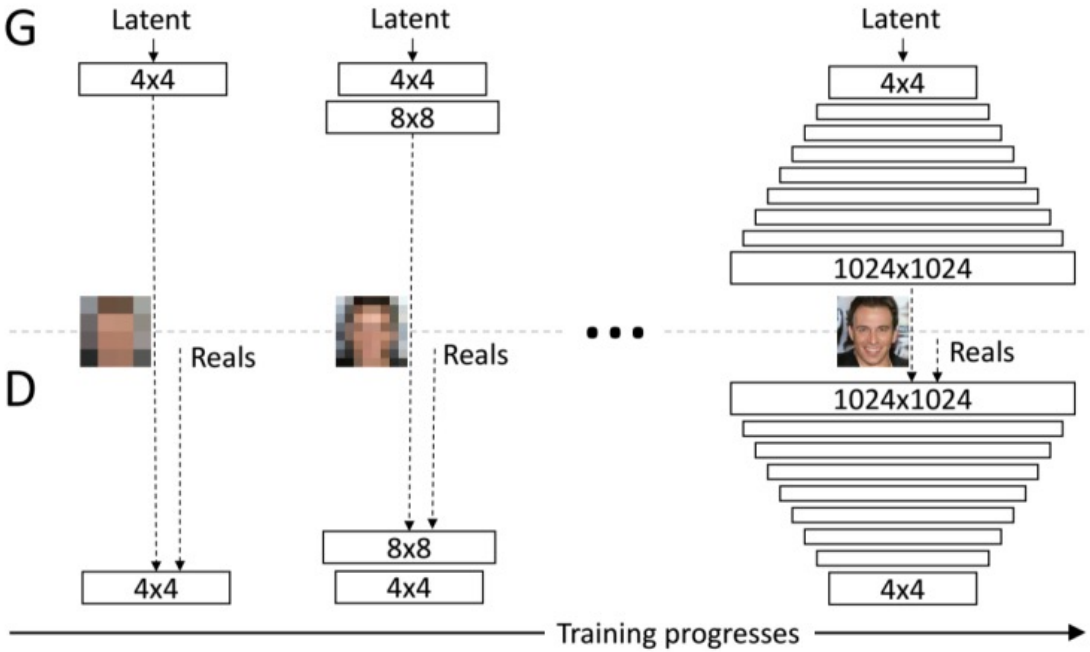
[Karras et al., 2018]

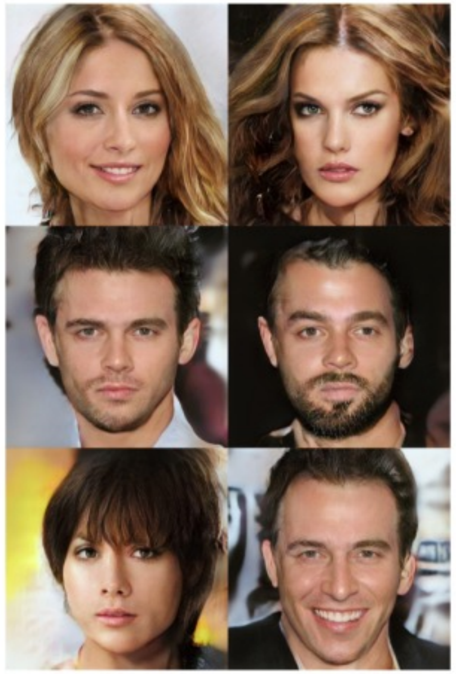
Progressive GAN

Low resolution images

add in additional layers

High resolution images





[Karras et al., 2018]

BigGAN

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling
- 2x – 4x more parameters
- 8x larger batch size
- Simple architecture changes that improve scalability

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling
- 2x – 4x more parameters
- 8x larger batch size
- Simple architecture changes that improve scalability

[Brock et al., 2018]

BigGAN

- GANs benefit dramatically from scaling
- 2x 4x more parameters
- 8x
- Sir

[Brock et al., 2018]

GANs for Text

Questions?