Lecture 20: Generative Models, Part 2

Admin: A4

A4 due yesterday, many people still working

Admin: A5

A5 Released last night

Recurrent networks, image captioning, Transformers

Due Tuesday April 12th at 11:59pm ET

Justin Johnson Lecture 20 - 3 March 30, 2022

Admin: Project Proposal

If you want to propose your own project:

Need to submit a project proposal by tomorrow, 4/1 on Piazza

Justin Johnson Lecture 20 - 4 March 30, 2022

Last Time: Supervised vs Unsupervised Learning

Supervised Learning

Unsupervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn a *function* to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Data: x

Just data, no labels!

Goal: Learn some underlying hidden *structure* of the data

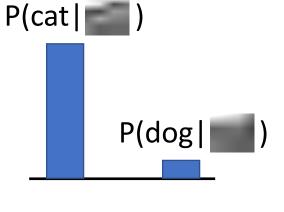
Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Discriminative Model:

Learn a probability distribution p(y|x)







Generative Model:

Learn a probability distribution p(x)

Density Function

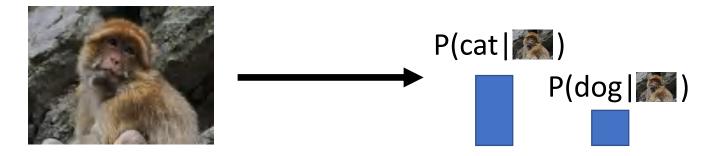
p(x) assigns a positive number to each possible x; higher numbers mean x is more likely Density functions are **normalized**:

$$\int_X p(x)dx = 1$$

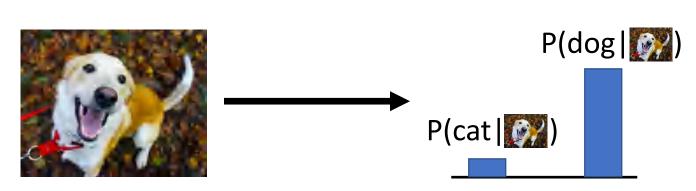
Different values of x compete for density

Conditional Generative Model: Learn p(x|y)

Discriminative Model: Learn a probability distribution p(y|x)



Generative Model: Learn a probability distribution p(x)



Conditional Generative Model: Learn p(x|y)

Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

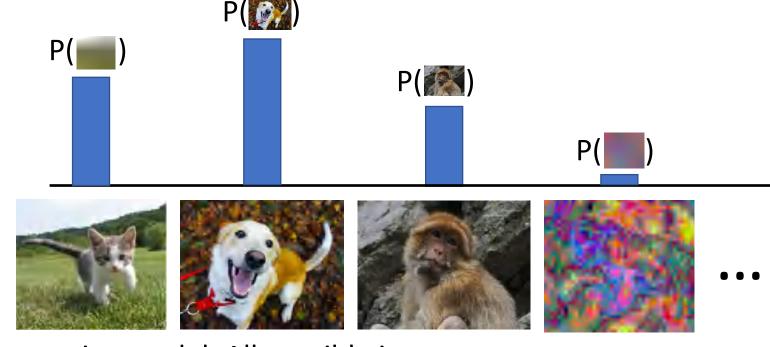
Monkey image is CCO Public Doma

Discriminative Model:

Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)



Generative model: All possible images compete with each other for probability mass

Requires deep image understanding! Is a dog more likely to sit or stand? How about 3-legged dog vs 3-armed monkey?

Justin Johnson Lecture 20 - 8 March 30, 2022

Discriminative Model:

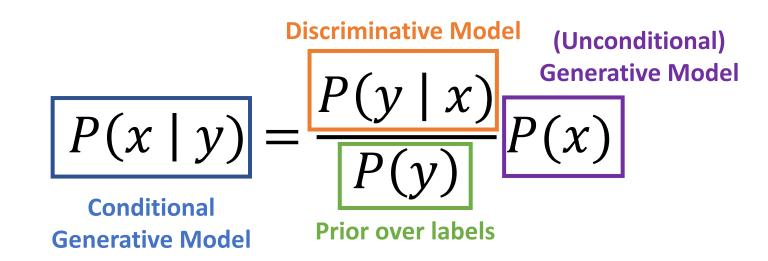
Learn a probability distribution p(y|x)

Generative Model:

Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

Recall Bayes' Rule:



We can build a conditional generative model from other components!

Last Time: Taxonomy of Generative Models

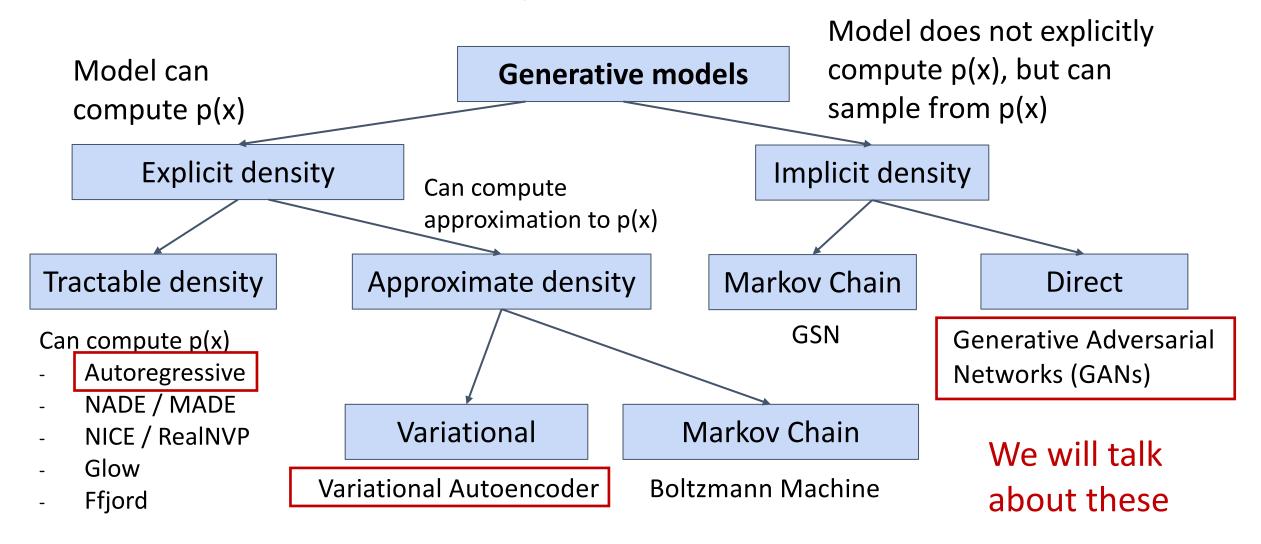


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Justin Johnson Lecture 20 - 10 March 30, 2022

Last Time: Autoregressive Models

Explicit Density Function

$$p(x) = p(x_1, x_2, x_3, ..., x_T)$$

$$= p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2) ...$$

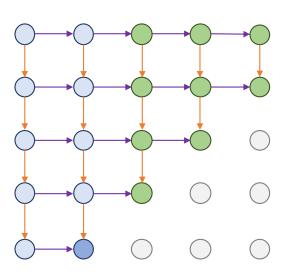
$$= \prod_{t=1}^{T} p(x_t | x_1, ..., x_{t-1})$$

Train by maximizing log-likelihood of training data

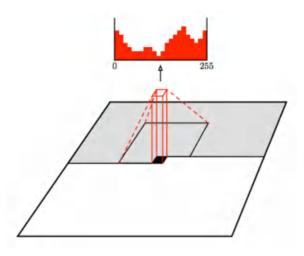
Van den Oord et al, "Pixel Recurrent Neural Networks", ICML 2016

Van den Oord et al, "Conditional Image Generation with PixelCNN Decoders", NeurIPS 2016

PixelRNN



PixelCNN

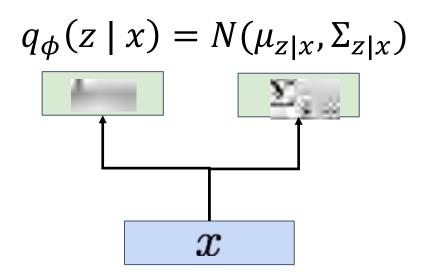


Last Time: Variational Autoencoders

Jointly train **encoder** q and **decoder** p to maximize the **variational lower bound** on the data likelihood

$$\log p_{\theta}(x) \ge E_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] - D_{KL} \left(q_{\phi}(z|x), p(z) \right)$$

Encoder Network



Decoder Network

$$p_{\theta}(x \mid z) = N(\mu_{x\mid z}, \Sigma_{x\mid z})$$

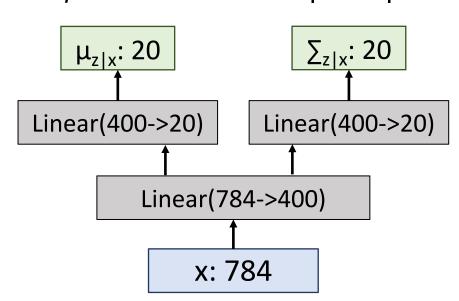
Example: Fully-Connected VAE

x: 28x28 image, flattened to 784-dim vector

z: 20-dim vector

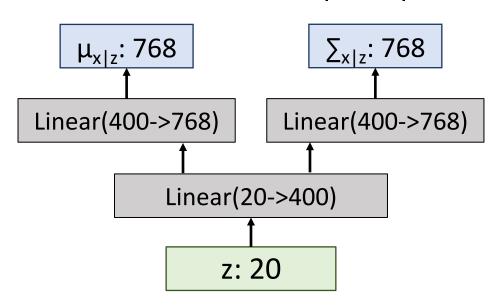
Encoder Network

$$q_{\phi}(z \mid x) = N(\mu_{z|x}, \Sigma_{z|x})$$



Decoder Network

$$p_{\theta}(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$



Justin Johnson Lecture 20 - 13 March 30, 2022

Train by maximizing the variational lower bound

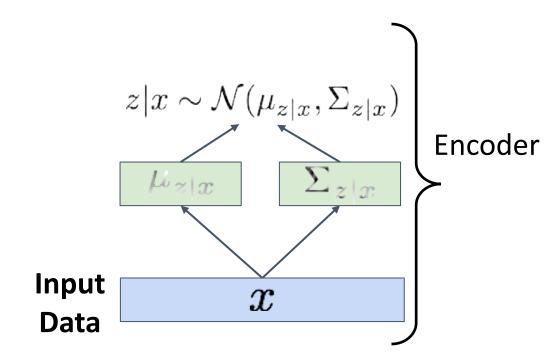
$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$



Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

1. Run input data through **encoder** to get a distribution over latent codes

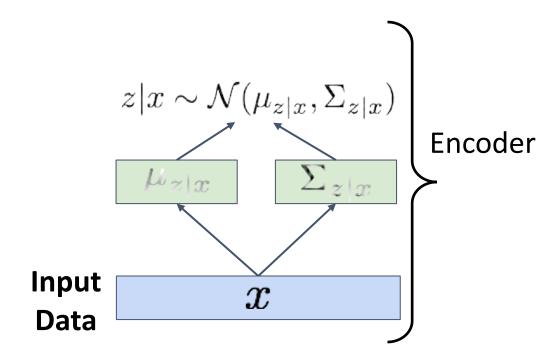


Justin Johnson Lecture 20 - 15 March 30, 2022

Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}(q_{\phi}(z|x), p(z))$$

- Run input data through encoder to get a distribution over latent codes
- 2. Encoder output should match the prior p(z)!



Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

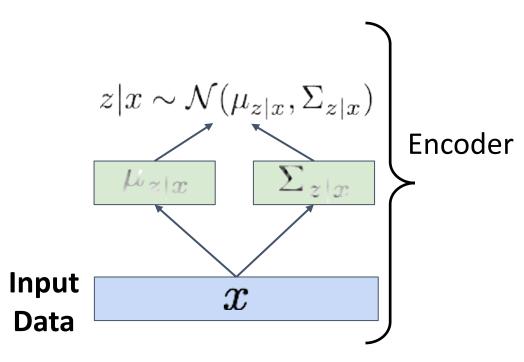
- Run input data through encoder to get a distribution over latent codes
- 2. Encoder output should match the prior p(z)!

$$-D_{KL}(q_{\phi}(z|x), p(z)) = \int_{Z} q_{\phi}(z|x) \log \frac{p(z)}{q_{\phi}(z|x)} dz$$

$$= \int_{Z} N(z; \mu_{z|x}, \Sigma_{z|x}) \log \frac{N(z; 0, I)}{N(z; \mu_{z|x}, \Sigma_{z|x})} dz$$

$$= \frac{1}{2} \sum_{i=1}^{J} \left(1 + \log \left(\left(\Sigma_{z|x} \right)_{j}^{2} \right) - \left(\mu_{z|x} \right)_{j}^{2} - \left(\Sigma_{z|x} \right)_{j}^{2} \right)$$

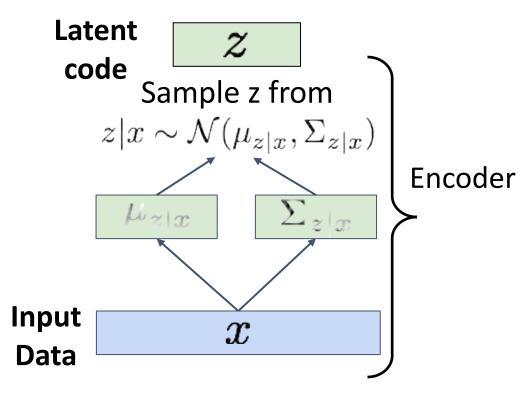
Closed form solution when q_{ϕ} is diagonal Gaussian and p is unit Gaussian! (Assume z has dimension J)



Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

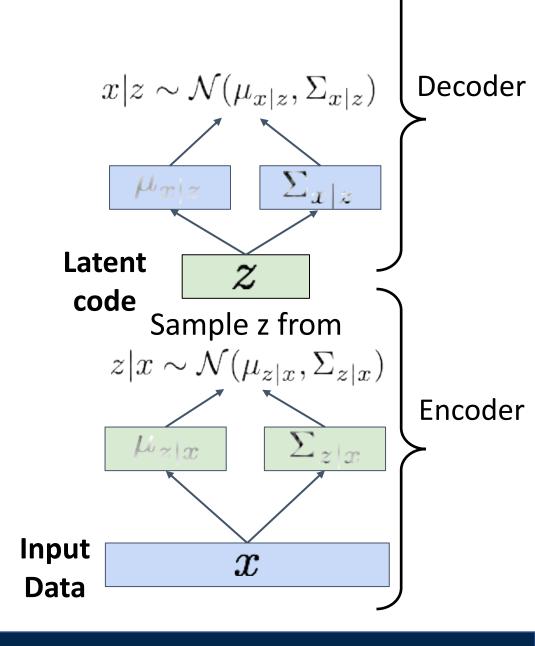
- 1. Run input data through **encoder** to get a distribution over latent codes
- 2. Encoder output should match the prior p(z)!
- 3. Sample code z from encoder output



Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

- Run input data through encoder to get a distribution over latent codes
- 2. Encoder output should match the prior p(z)!
- 3. Sample code z from encoder output
- Run sampled code through decoder to get a distribution over data samples

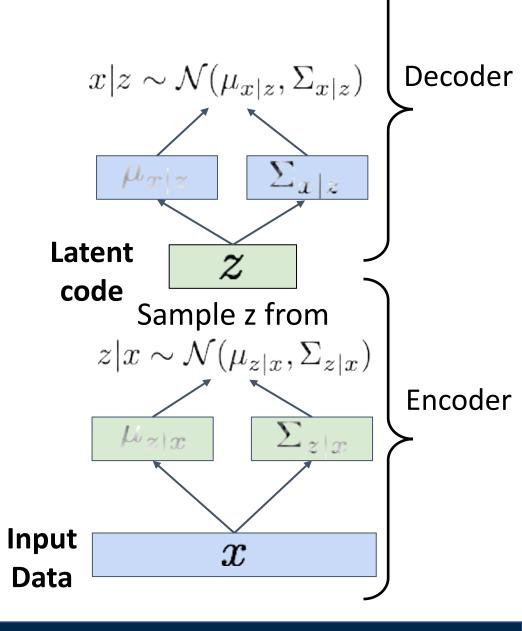


Justin Johnson Lecture 20 - 19 March 30, 2022

Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}(q_{\phi}(z|x), p(z))$$

- 1. Run input data through **encoder** to get a distribution over latent codes
- 2. Encoder output should match the prior p(z)!
- 3. Sample code z from encoder output
- Run sampled code through decoder to get a distribution over data samples
- 5. Original input data should be likely under the distribution output from (4)!

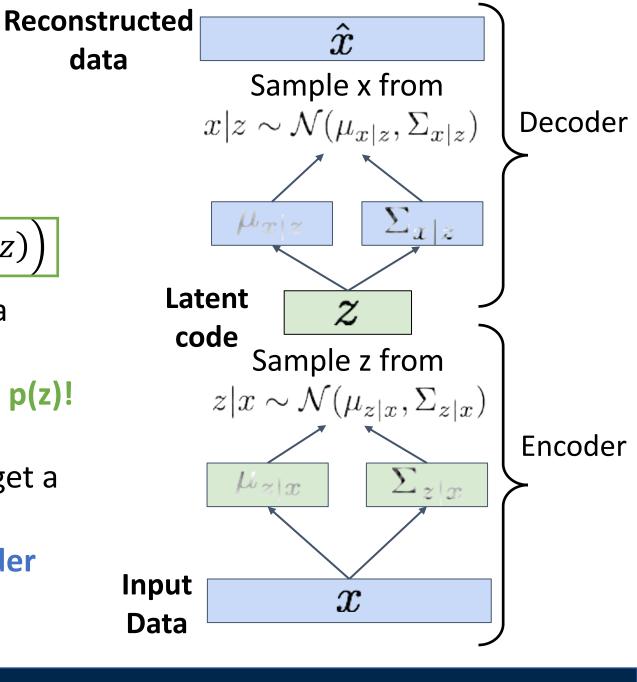


Justin Johnson Lecture 20 - 20 March 30, 2022

Train by maximizing the variational lower bound

$$E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}(q_{\phi}(z|x), p(z))$$

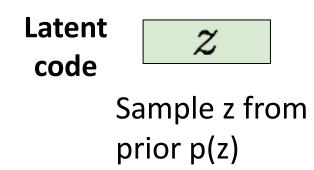
- Run input data through encoder to get a distribution over latent codes
- 2. Encoder output should match the prior p(z)!
- 3. Sample code z from encoder output
- 4. Run sampled code through **decoder** to get a distribution over data samples
- 5. Original input data should be likely under the distribution output from (4)!
- 6. Can sample a reconstruction from (4)



Justin Johnson Lecture 20 - 21 March 30, 2022

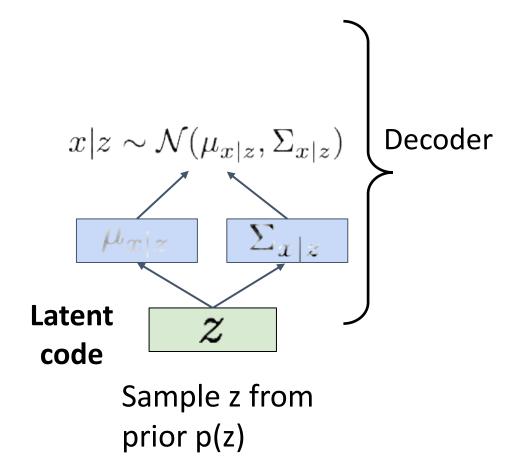
After training we can generate new data!

Sample z from prior p(z)



After training we can generate new data!

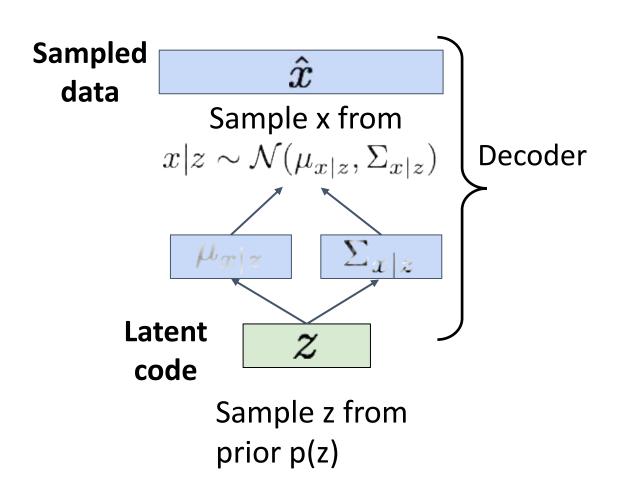
- 1. Sample z from prior p(z)
- Run sampled z through decoder to get distribution over data x



Justin Johnson Lecture 20 - 23 March 30, 2022

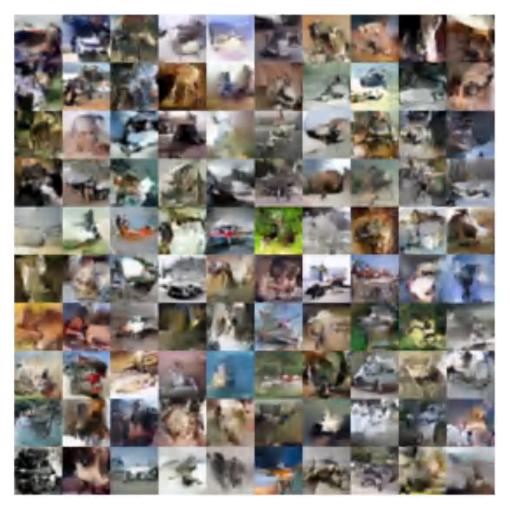
After training we can generate new data!

- 1. Sample z from prior p(z)
- Run sampled z through decoder to get distribution over data x
- 3. Sample from distribution in (2) to generate data

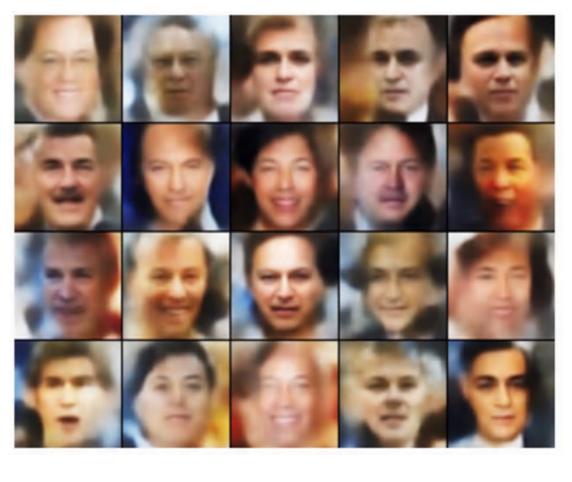


Justin Johnson Lecture 20 - 24 March 30, 2022

32x32 CIFAR-10



Labeled Faces in the Wild



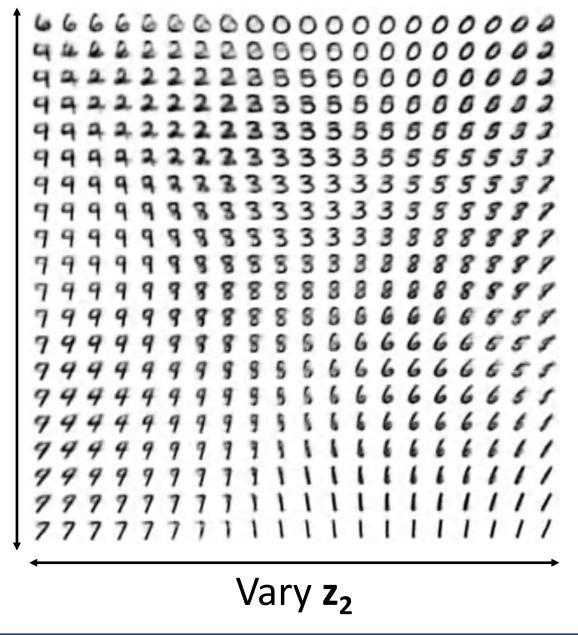
Figures from (L) Dirk Kingma et al. 2016; (R) Anders Larsen et al. 2017.

Justin Johnson Lecture 20 - 25 March 30, 2022

The diagonal prior on p(z) causes dimensions of z to be independent

"Disentangling factors of variation"

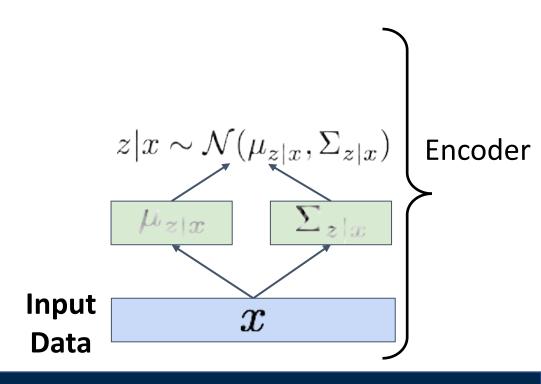
Vary z₁



Kingma and Welling, Auto-Encoding Variational Beyes, ICLR 2014

After training we can edit images

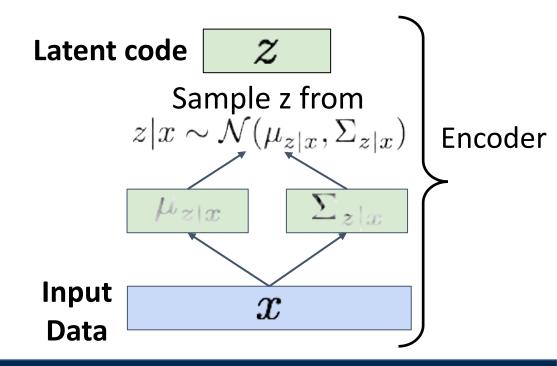
1. Run input data through **encoder** to get a distribution over latent codes



Justin Johnson Lecture 20 - 27 March 30, 2022

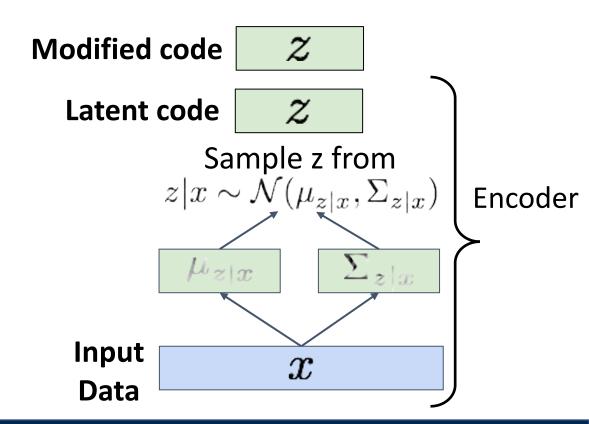
After training we can edit images

- Run input data through encoder to get a distribution over latent codes
- 2. Sample code z from encoder output



After training we can edit images

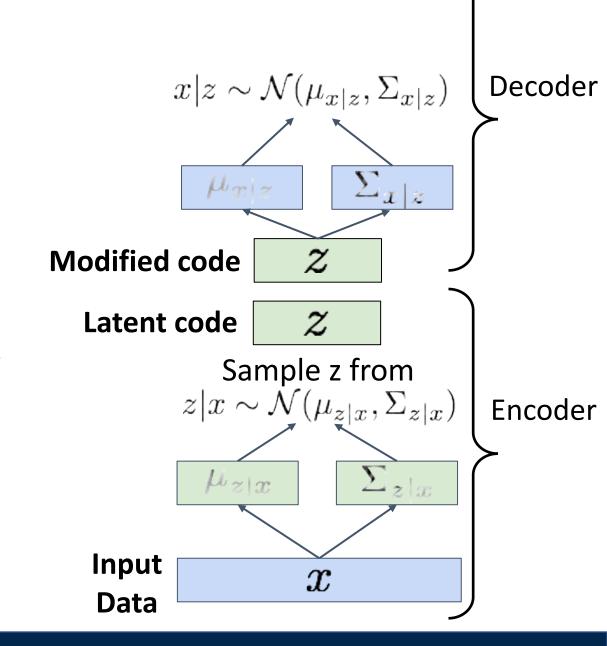
- Run input data through encoder to get a distribution over latent codes
- 2. Sample code z from encoder output
- 3. Modify some dimensions of sampled code



Justin Johnson Lecture 20 - 29 March 30, 2022

After training we can edit images

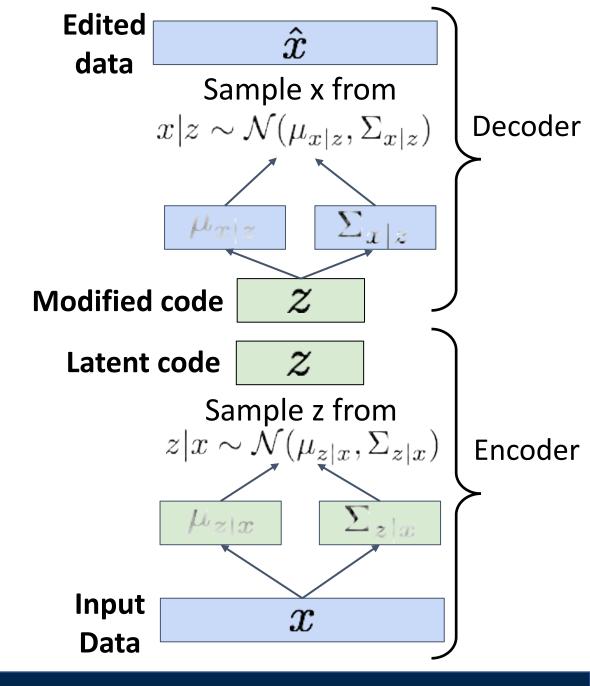
- Run input data through encoder to get a distribution over latent codes
- 2. Sample code z from encoder output
- 3. Modify some dimensions of sampled code
- Run modified z through decoder to get a distribution over data sample



Justin Johnson Lecture 20 - 30 March 30, 2022

After training we can edit images

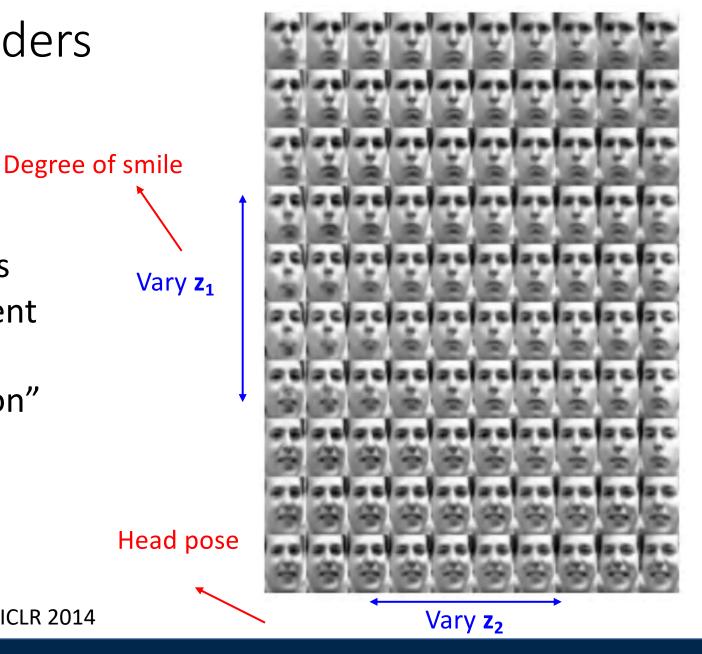
- 1. Run input data through **encoder** to get a distribution over latent codes
- 2. Sample code z from encoder output
- 3. Modify some dimensions of sampled code
- Run modified z through decoder to get a distribution over data samples
- 5. Sample new data from (4)



Justin Johnson Lecture 20 - 31 March 30, 2022

The diagonal prior on p(z) causes dimensions of z to be independent

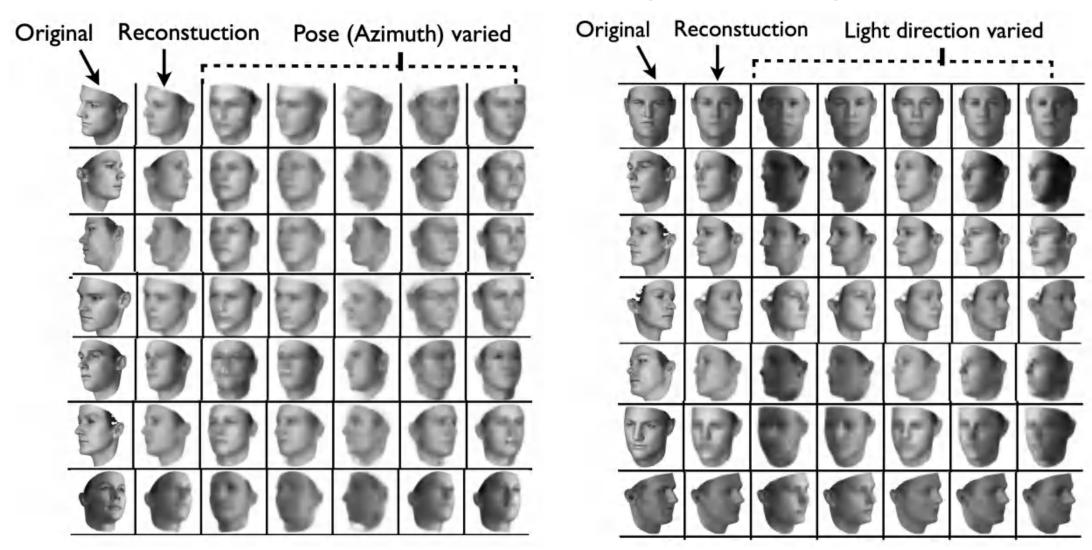
"Disentangling factors of variation"



Kingma and Welling, Auto-Encoding Variational Beyes, ICLR 2014

Justin Johnson Lecture 20 - 32 March 30, 2022

Variational Autoencoders: Image Editing



Kulkarni et al, "Deep Convolutional Inverse Graphics Networks", NeurIPS 2014

Justin Johnson Lecture 20 - 33 March 30, 2022

Variational Autoencoder: Summary

Probabilistic spin to traditional autoencoders => allows generating data

Defines an intractable density => derive and optimize a (variational) lower bound

Pros:

- Principled approach to generative models
- Allows inference of q(z|x), can be useful feature representation for other tasks

Cons:

- Maximizes lower bound of likelihood: okay, but not as good evaluation as PixelRNN/PixelCNN
- Samples blurrier and lower quality compared to state-of-the-art (GANs)

Active areas of research:

- More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian, e.g., Gaussian Mixture Models (GMMs)
- Incorporating structure in latent variables, e.g., Categorical Distributions

So far: Two types of generative models

Autoregressive models

- Directly maximize p(data)
- High-quality generated images
- Slow to generate images
- No explicit latent codes

Variational models

- Maximize lower-bound on p(data)
- Generated images often blurry
- Very fast to generate images
- Learn rich latent codes

Justin Johnson Lecture 20 - 35 March 30, 2022

So far: Two types of generative models

Autoregressive models

- Directly maximize p(data)
- High-quality generated images
- Slow to generate images
- No explicit latent codes

Variational models

- Maximize lower-bound on p(data)
- Generated images often blurry
- Very fast to generate images
- Learn rich latent codes

Can we combine them and get the best of both worlds?

Justin Johnson Lecture 20 - 36 March 30, 2022

Combining VAE + Autoregressive: Vector-Quantized Variational Autoencoder (VQ-VAE2)

Train a VAE-like model to generate multiscale grids of latent codes

VQ-VAE Encoder and Decoder Training

Top Level

Encoder

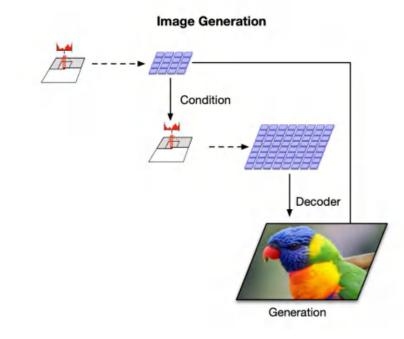
Decoder

Decoder

Decoder

Decoder

Use a multiscale PixelCNN to sample in latent code space

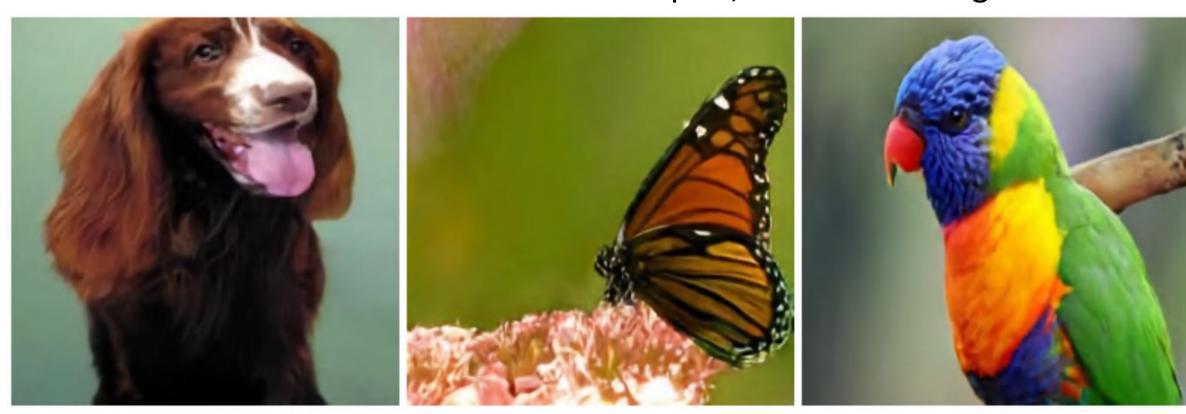


Razavi et al, "Generating Diverse High-Fidelity Images with VQ-VAE-2", NeurIPS 2019

Reconstruction

Justin Johnson Lecture 20 - 37 March 30, 2022

256 x 256 class-conditional samples, trained on ImageNet



Razavi et al, "Generating Diverse High-Fidelity Images with VQ-VAE-2", NeurIPS 2019

256 x 256 class-conditional samples, trained on ImageNet

Redshank

Pekinese

Papillon

Drake

Spotted Salamander



Razavi et al, "Generating Diverse High-Fidelity Images with VQ-VAE-2", NeurIPS 2019

Justin Johnson Lecture 20 - 39 March 30, 2022

1024 x 1024 generated faces, trained on FFHQ



Razavi et al, "Generating Diverse High-Fidelity Images with VQ-VAE-2", NeurIPS 2019

1024 x 1024 generated faces, trained on FFHQ





Razavi et al, "Generating Diverse High-Fidelity Images with VQ-VAE-2", NeurIPS 2019

Generative Models So Far:

Autoregressive Models directly maximize likelihood of training data:

$$p_{\theta}(x) = \prod_{i=1}^{N} p_{\theta}(x_i|x_1,...,x_{i-1})$$

Justin Johnson Lecture 20 - 42 March 30, 2022

Generative Models So Far:

Autoregressive Models directly maximize likelihood of training data:

$$p_{\theta}(x) = \prod_{i=1}^{N} p_{\theta}(x_i|x_1,...,x_{i-1})$$

Variational Autoencoders introduce a latent z, and maximize a lower bound:

$$p_{\theta}(x) = \int_{Z} p_{\theta}(x|z)p(z)dz \ge E_{z \sim q_{\phi}(Z|X)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

Justin Johnson Lecture 20 - 43 March 30, 2022

Generative Models So Far:

Autoregressive Models directly maximize likelihood of training data:

$$p_{\theta}(x) = \prod_{i=1}^{N} p_{\theta}(x_i|x_1,...,x_{i-1})$$

Variational Autoencoders introduce a latent z, and maximize a lower bound:

$$p_{\theta}(x) = \int_{Z} p_{\theta}(x|z)p(z)dz \ge E_{z \sim q_{\phi}(Z|X)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

Generative Adversarial Networks give up on modeling p(x), but allow us to draw samples from p(x)

Justin Johnson Lecture 20 - 44 March 30, 2022

Setup: Assume we have data x_i drawn from distribution $p_{data}(x)$. Want to sample from p_{data} .

Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Justin Johnson Lecture 20 - 45 March 30, 2022

Setup: Assume we have data x_i drawn from distribution $p_{data}(x)$. Want to sample from p_{data} .

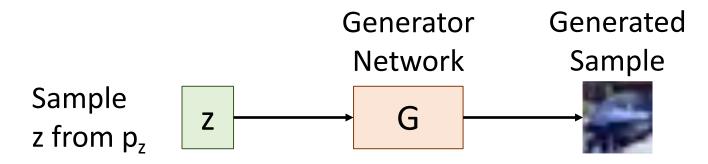
Idea: Introduce a latent variable z with simple prior p(z).

Sample $z \sim p(z)$ and pass to a **Generator Network** x = G(z)

Then x is a sample from the **Generator distribution** p_G . Want $p_G = p_{data}!$

Setup: Assume we have data x_i drawn from distribution $p_{data}(x)$. Want to sample from p_{data} .

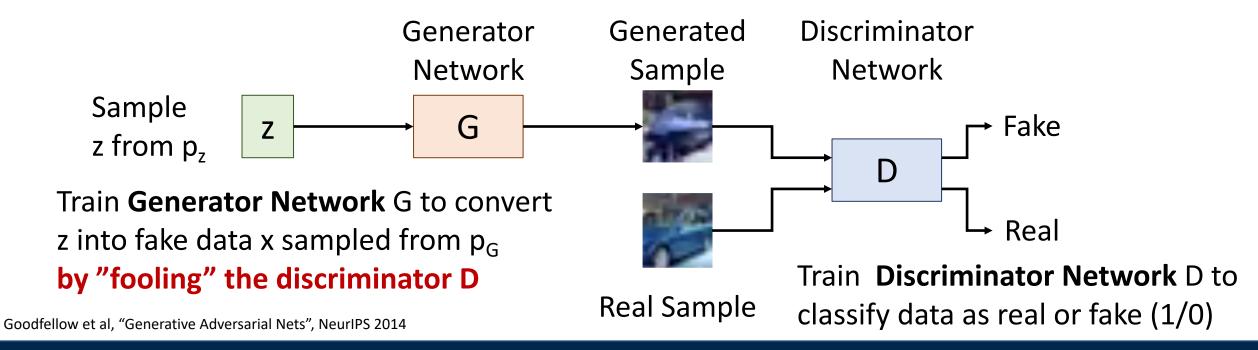
Idea: Introduce a latent variable z with simple prior p(z). Sample $z \sim p(z)$ and pass to a **Generator Network** x = G(z) Then x is a sample from the **Generator distribution** p_G. Want p_G = p_{data}!



Train **Generator Network** G to convert z into fake data x sampled from p_G

Setup: Assume we have data x_i drawn from distribution $p_{data}(x)$. Want to sample from p_{data} .

Idea: Introduce a latent variable z with simple prior p(z). Sample $z \sim p(z)$ and pass to a **Generator Network** x = G(z) Then x is a sample from the **Generator distribution** p_G. Want p_G = p_{data}!



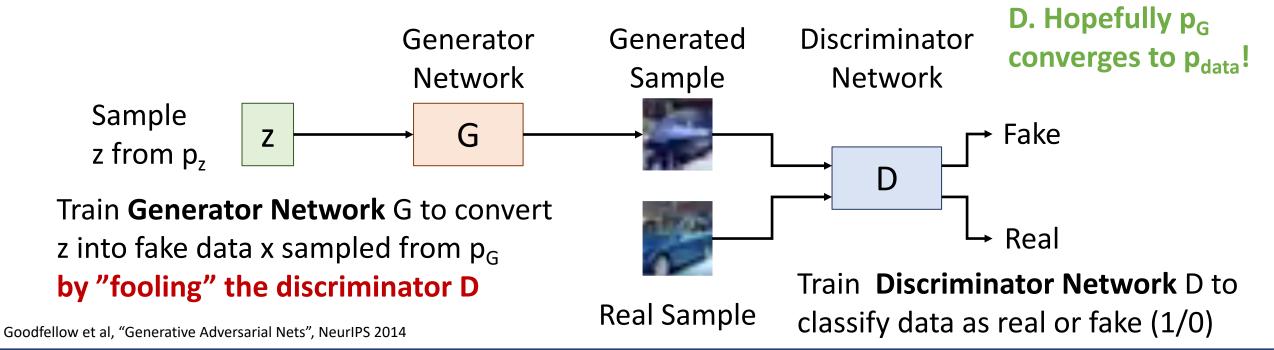
Justin Johnson Lecture 20 - 48 March 30, 2022

Setup: Assume we have data x_i drawn from distribution $p_{data}(x)$. Want to sample from p_{data} .

Idea: Introduce a latent variable z with simple prior p(z).

Sample $z \sim p(z)$ and pass to a **Generator Network** x = G(z)

Then x is a sample from the **Generator distribution** p_G . Want $p_G = p_{data}!$



Jointly train G and

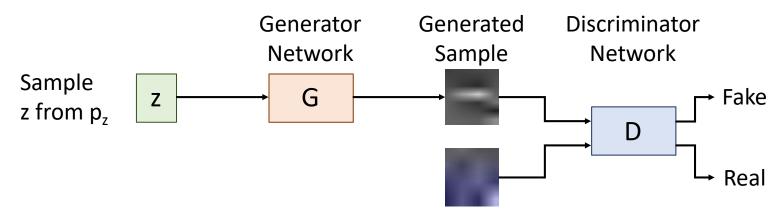
Justin Johnson Lecture 20 - 49 March 30, 2022

Jointly train generator G and discriminator D with a minimax game

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

Jointly train generator G and discriminator D with a minimax game

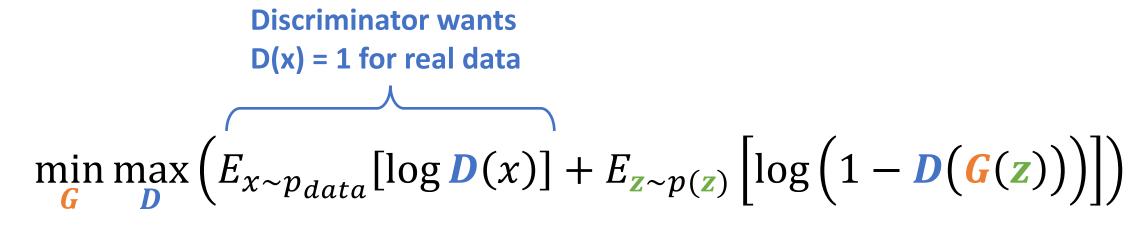
$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

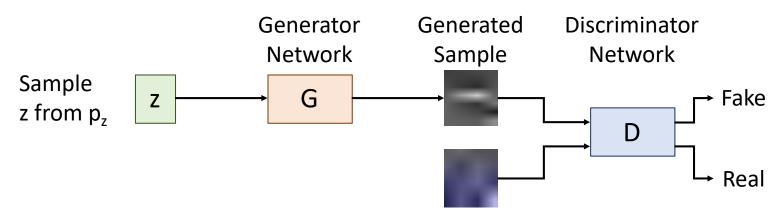


Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Justin Johnson Lecture 20 - 51 March 30, 2022

Jointly train generator G and discriminator D with a minimax game

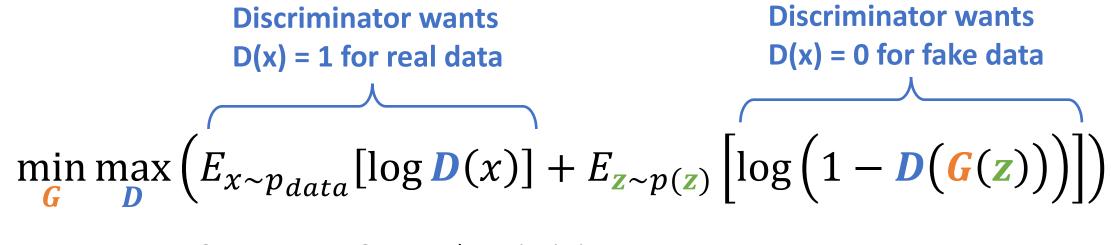


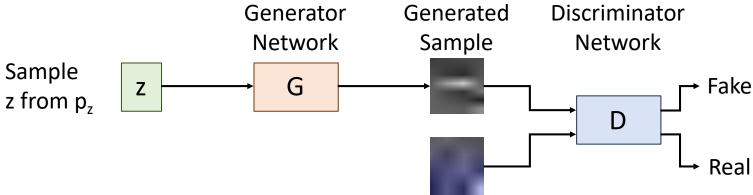


Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Justin Johnson Lecture 20 - 52 March 30, 2022

Jointly train generator G and discriminator D with a minimax game

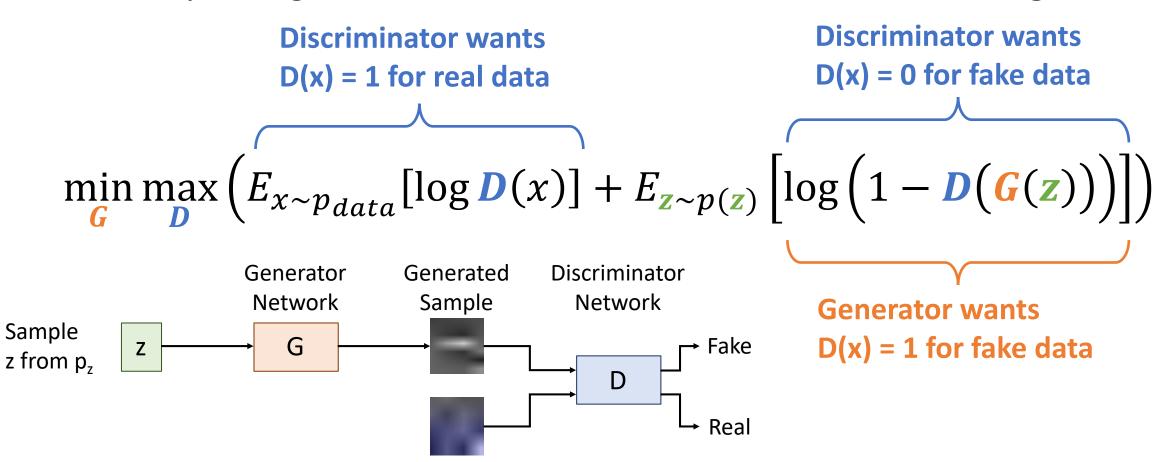




Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Justin Johnson Lecture 20 - 53 March 30, 2022

Jointly train generator G and discriminator D with a minimax game



Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Justin Johnson Lecture 20 - 54 March 30, 2022

Jointly train generator G and discriminator D with a minimax game

Train G and D using alternating gradient updates

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

Jointly train generator G and discriminator D with a minimax game

Train G and D using alternating gradient updates

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

$$= \min_{G} \max_{D} V(G, D)$$

Jointly train generator G and discriminator D with a minimax game

Train G and D using alternating gradient updates

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

$$= \min_{G} \max_{D} V(G, D)$$

For t in 1, ... T:

1. (Update D)
$$D = D + \alpha_D \frac{\partial V}{\partial D}$$

2. (Update G) $G = G - \alpha_G \frac{\partial V}{\partial G}$

2. (Update G)
$$G = G - \alpha_G \frac{\partial V}{\partial G}$$

Jointly train generator G and discriminator D with a minimax game

Train G and D using alternating gradient updates

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{G}, \mathbf{D})$$

We are not minimizing any overall loss! No training curves to look at! For t in 1, ... T:

1. (Update D)
$$D = D + \alpha_D \frac{\partial V}{\partial D}$$

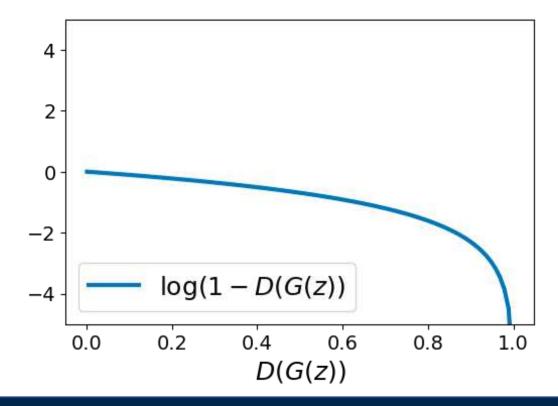
2. (Update G) $G = G - \alpha_G \frac{\partial V}{\partial G}$

2. (Update G)
$$G = G - \alpha_G \frac{\partial V}{\partial G}$$

Jointly train generator G and discriminator D with a minimax game

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

At start of training, generator is very bad and discriminator can easily tell apart real/fake, so D(G(z)) close to 0

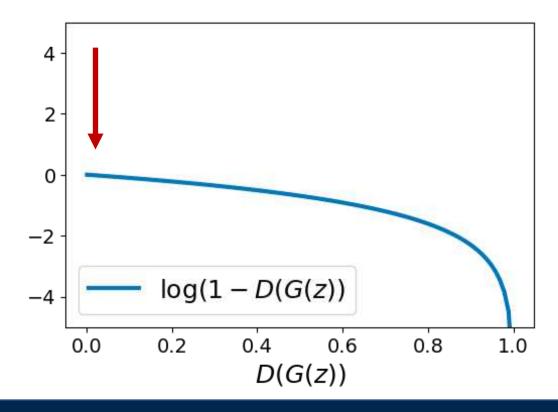


Jointly train generator G and discriminator D with a minimax game

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

At start of training, generator is very bad and discriminator can easily tell apart real/fake, so D(G(z)) close to 0

Problem: Vanishing gradients for G



Jointly train generator G and discriminator D with a minimax game

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

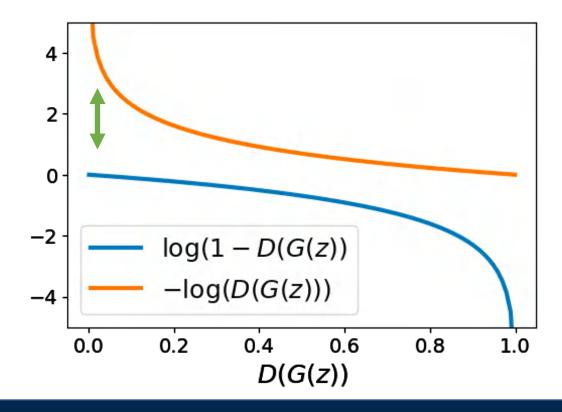
At start of training, generator is very bad and discriminator can easily tell apart real/fake, so D(G(z)) close to 0

Problem: Vanishing gradients for G

Solution: Right now G is trained to

gradients at start of training!

minimize log(1-D(G(z))). Instead, train G to minimize $-\log(D(G(z))$. Then G gets strong



Jointly train generator G and discriminator D with a minimax game

Why is this particular objective a good idea?

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} [\log \mathbf{D}(x)] + E_{\mathbf{Z} \sim p(\mathbf{Z})} \left[\log \left(1 - \mathbf{D}(\mathbf{G}(\mathbf{Z})) \right) \right] \right)$$

This minimax game achieves its global minimum when $p_G = p_{data}!$

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

(Our objective so far)

Justin Johnson Lecture 20 - 63 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} \left[\log \mathbf{D}(x) \right] + E_{x \sim p_{\mathbf{G}}} \left[\log \left(1 - \mathbf{D}(x) \right) \right] \right)$$

(Change of variables on second term)

Justin Johnson Lecture 20 - 64 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} \left[\log D(x) \right] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right)$$

$$= \min_{G} \max_{D} \int_{X} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x)\right) \right) dx$$

(Definition of expectation)

Justin Johnson Lecture 20 - 65 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} \left[\log D(x) \right] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right)$$

$$= \min_{G} \int_{X} \max_{D} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x) \right) \right) dx$$

(Push max_D inside integral)

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right)$$

$$= \min_{G} \int_{X} \max_{D} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x) \right) \right) dx$$

$$f(y) = a \log y + b \log(1 - y)$$

(Side computation to compute max)

Justin Johnson Lecture 20 - March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right)$$

$$= \min_{G} \int_{X} \max_{D} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x) \right) \right) dx$$

$$= a \log x + b \log \left(1 - y \right)$$

$$f(y) = \frac{a}{a} \log y + \frac{b}{b} \log(1 - y)$$

$$f'(y) = \frac{a}{y} - \frac{b}{1-y}$$

$$\begin{aligned} & \min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right) \\ & = \min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right) \\ & = \min_{G} \int_{X} \max_{D} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x) \right) \right) dx \\ & f(y) = a \log y + b \log (1 - y) \qquad f'(y) = 0 \iff y = \frac{a}{a + b} \text{ (local max)} \\ & f'(y) = \frac{a}{y} - \frac{b}{1 - y} \end{aligned}$$

Justin Johnson Lecture 20 - 69 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{x \sim p_{G}} [\log \left(1 - D(x) \right)] \right)$$

$$= \min_{G} \int_{X} \max_{D} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x) \right) \right) dx$$

$$f(y) = a \log y + b \log (1 - y) \qquad f'(y) = 0 \iff y = \frac{a}{a + b} \text{ (local max)}$$

$$f'(y) = \frac{a}{y} - \frac{b}{1 - y} \quad \text{Optimal Discriminator: } D_{G}^{*}(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{G}(x)}$$

Justin Johnson Lecture 20 - 70 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} \left[\log D(x) \right] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right)$$

$$= \min_{G} \int_{X} \max_{D} \left(p_{data}(x) \log D(x) + p_{G}(x) \log \left(1 - D(x) \right) \right) dx$$

Optimal Discriminator:
$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}$$

Justin Johnson Lecture 20 - 71 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \max_{\mathbf{D}} \left(E_{x \sim p_{data}} \left[\log \mathbf{D}(x) \right] + E_{x \sim p_{\mathbf{G}}} \left[\log \left(1 - \mathbf{D}(x) \right) \right] \right)$$

$$= \min_{G} \int_{Y} \left(p_{data}(x) \log D_{G}^{*}(x) + p_{G}(x) \log \left(1 - D_{G}^{*}(x) \right) \right) dx$$

Optimal Discriminator:
$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}$$

Justin Johnson Lecture 20 - 72 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \max_{D} \left(E_{x \sim p_{data}} \left[\log D(x) \right] + E_{x \sim p_{G}} \left[\log \left(1 - D(x) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \int_{X} \left(p_{data}(x) \log D_{\mathbf{G}}^{*}(x) + p_{\mathbf{G}}(x) \log \left(1 - D_{\mathbf{G}}^{*}(x) \right) \right) dx$$

$$= \min_{\mathbf{G}} \int_{X} \left(p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} + p_{\mathbf{G}}(x) \log \frac{p_{\mathbf{G}}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right) dx$$

Optimal Discriminator:
$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}$$

Justin Johnson Lecture 20 - 73 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \int_{X} \left(p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} + p_{\mathbf{G}}(x) \log \frac{p_{\mathbf{G}}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right) dx$$

Justin Johnson Lecture 20 - 74 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \int_{X} \left(p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_{G}(x)} + p_{G}(x) \log \frac{p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right) dx$$

$$= \min_{\mathbf{G}} \left(E_{x \sim p_{data}} \left[\log \frac{p_{data}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right] + E_{x \sim p_{\mathbf{G}}} \left[\log \frac{p_{\mathbf{G}}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right] \right)$$

(Definition of expectation)

Justin Johnson Lecture 20 - 75 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \int_{X} \left(p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} + p_{\mathbf{G}}(x) \log \frac{p_{\mathbf{G}}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right) dx$$

$$= \min_{\mathbf{G}} \left(E_{x \sim p_{data}} \left[\log \frac{2}{2} \frac{p_{data}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right] + E_{x \sim p_{\mathbf{G}}} \left[\log \frac{2}{2} \frac{p_{\mathbf{G}}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right] \right)$$

(Multiply by a constant)

Justin Johnson Lecture 20 - 76 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{\mathbf{G}} \int_{X} \left(p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} + p_{\mathbf{G}}(x) \log \frac{p_{\mathbf{G}}(x)}{p_{data}(x) + p_{\mathbf{G}}(x)} \right) dx$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2}{2} \frac{p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2}{2} \frac{p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

Justin Johnson Lecture 20 - 77 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

Justin Johnson Lecture 20 - 78 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

Kullback-Leibler Divergence:

$$KL(\mathbf{p}, q) = E_{x \sim \mathbf{p}} \left[\log \frac{\mathbf{p}(x)}{q(x)} \right]$$

Justin Johnson Lecture 20 - 79 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

Kullback-Leibler Divergence:

$$KL(\mathbf{p}, q) = E_{x \sim \mathbf{p}} \left[\log \frac{\mathbf{p}(x)}{q(x)} \right]$$

Justin Johnson Lecture 20 - 80 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

$$= \min_{G} \left(KL \left(p_{data}, \frac{p_{data} + p_{G}}{2} \right) + KL \left(p_{G}, \frac{p_{data} + p_{G}}{2} \right) - \log 4 \right)$$

Kullback-Leibler Divergence:

$$KL(\mathbf{p}, \mathbf{q}) = E_{x \sim \mathbf{p}} \left[\log \frac{\mathbf{p}(x)}{\mathbf{q}(x)} \right]$$

Justin Johnson Lecture 20 - 81 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

$$= \min_{\mathbf{G}} \left(KL \left(p_{data}, \frac{p_{data} + p_{\mathbf{G}}}{2} \right) + KL \left(p_{\mathbf{G}}, \frac{p_{data} + p_{\mathbf{G}}}{2} \right) - \log 4 \right)$$

Justin Johnson Lecture 20 - 82 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

$$= \min_{\mathbf{G}} \left(KL\left(p_{data}, \frac{p_{data} + p_{\mathbf{G}}}{2}\right) + KL\left(p_{\mathbf{G}}, \frac{p_{data} + p_{\mathbf{G}}}{2}\right) - \log 4 \right)$$

Jensen-Shannon Divergence:

$$JSD(\mathbf{p},q) = \frac{1}{2}KL\left(\mathbf{p},\frac{\mathbf{p}+q}{2}\right) + \frac{1}{2}KL\left(q,\frac{\mathbf{p}+q}{2}\right)$$

Justin Johnson Lecture 20 - 83 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

$$= \min_{G} \left(KL \left(p_{data}, \frac{p_{data} + p_{G}}{2} \right) + KL \left(p_{G}, \frac{p_{data} + p_{G}}{2} \right) - \log 4 \right)$$

Jensen-Shannon Divergence:

$$JSD(\mathbf{p},q) = \frac{1}{2}KL\left(\mathbf{p},\frac{\mathbf{p}+q}{2}\right) + \frac{1}{2}KL\left(q,\frac{\mathbf{p}+q}{2}\right)$$

Justin Johnson Lecture 20 - 84 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

$$= \min_{G} \left(KL \left(p_{data}, \frac{p_{data} + p_{G}}{2} \right) + KL \left(p_{G}, \frac{p_{data} + p_{G}}{2} \right) - \log 4 \right)$$

$$= \min_{G} (2 * JSD(p_{data}, p_G) - \log 4)$$

Jensen-Shannon Divergence:

$$JSD(\mathbf{p},q) = \frac{1}{2}KL\left(\mathbf{p},\frac{\mathbf{p}+q}{2}\right) + \frac{1}{2}KL\left(q,\frac{\mathbf{p}+q}{2}\right)$$

Justin Johnson Lecture 20 - 85 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} \left(E_{x \sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_{G}(x)} \right] + E_{x \sim p_{G}} \left[\log \frac{2 * p_{G}(x)}{p_{data}(x) + p_{G}(x)} \right] - \log 4 \right)$$

$$= \min_{\mathbf{G}} \left(\mathit{KL}\left(p_{data}, \frac{p_{data} + p_{\mathbf{G}}}{2}\right) + \mathit{KL}\left(p_{\mathbf{G}}, \frac{p_{data} + p_{\mathbf{G}}}{2}\right) - \log 4 \right)$$

$$= \min_{G} (2 * JSD(p_{data}, p_{G}) - \log 4)$$

JSD is always nonnegative, and zero only when the two distributions are equal! Thus $p_{data} = p_G$ is the global min, QED

Jensen-Shannon Divergence:

$$JSD(\mathbf{p},q) = \frac{1}{2}KL\left(\mathbf{p},\frac{\mathbf{p}+q}{2}\right) + \frac{1}{2}KL\left(q,\frac{\mathbf{p}+q}{2}\right)$$

Justin Johnson Lecture 20 - 86 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} (2 * JSD(p_{data}, p_G) - \log 4)$$

Justin Johnson Lecture 20 - 87 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} (2 * JSD(p_{data}, p_G) - \log 4)$$

Summary: The global minimum of the minimax game happens when:

1.
$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}$$
 (Optimal discriminator for any G)

2.
$$p_G(x) = p_{data}(x)$$
 (Optimal generator for optimal D)

Justin Johnson Lecture 20 - 88 March 30, 2022

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

$$= \min_{G} (2 * JSD(p_{data}, p_{G}) - \log 4)$$

Summary: The global minimum of the minimax game happens when:

1.
$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)}$$
 (Optimal discriminator for any G)

2.
$$p_G(x) = p_{data}(x)$$
 (Optimal generator for optimal D)

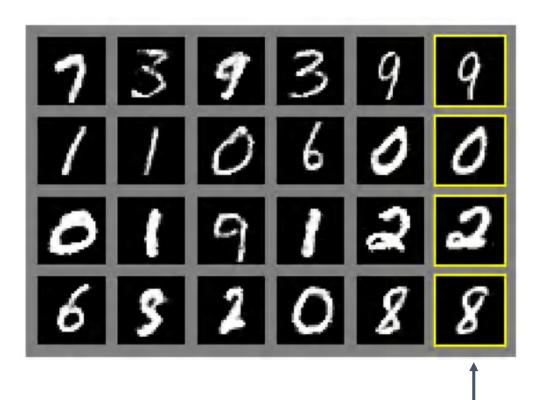
Caveats:

- 1. G and D are neural nets with fixed architecture. We don't know whether they can actually <u>represent</u> the optimal D and G.
- 2. This tells us nothing about convergence to the optimal solution

Justin Johnson Lecture 20 - 89 March 30, 2022

Generative Adversarial Networks: Results

Generated samples

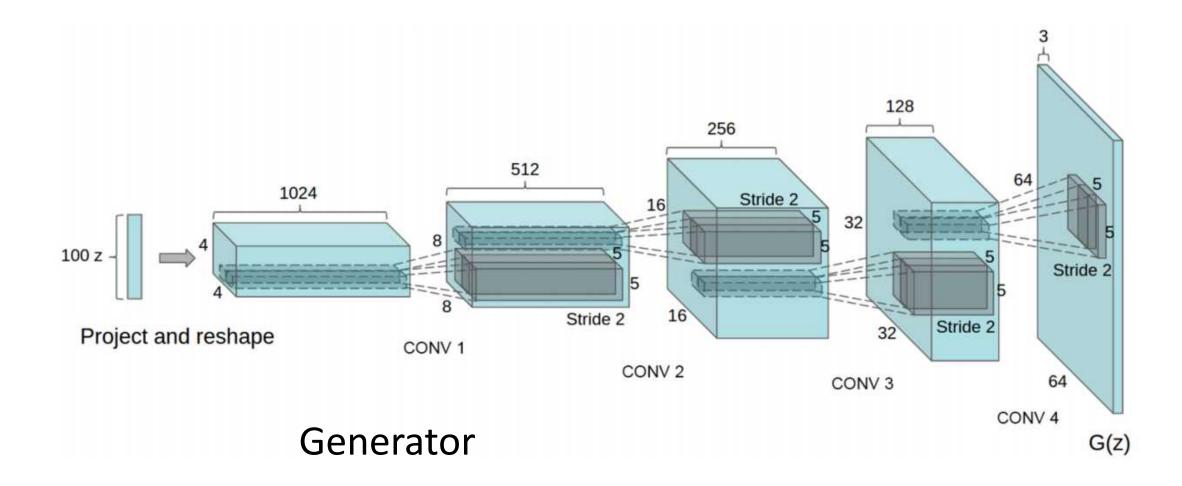




Nearest neighbor from training set

Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

Generative Adversarial Networks: DC-GAN

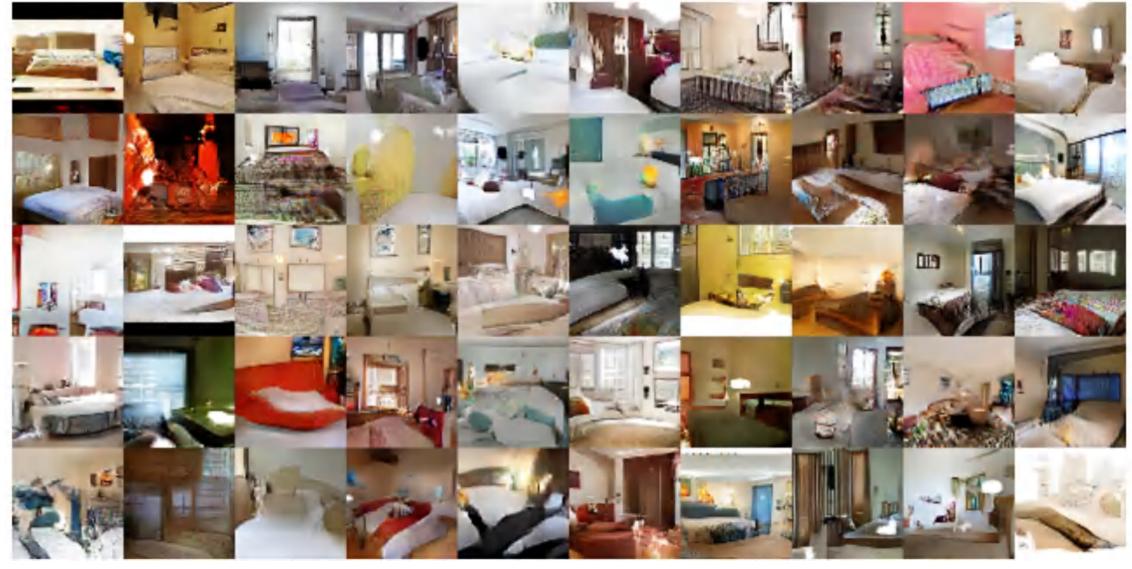


Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

Justin Johnson Lecture 20 - 91 March 30, 2022

Generative Adversarial Networks: DC-GAN

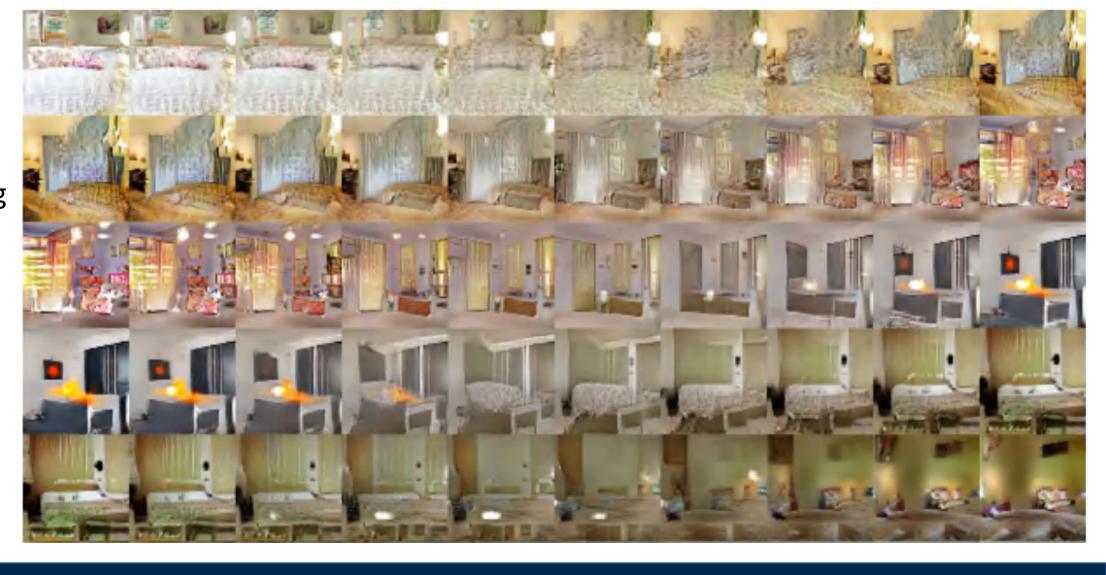
Samples from the model look much better!



Radford et al, ICLR 2016

Generative Adversarial Networks: Interpolation

Interpolating between points in latent z space



Radford et al, ICLR 2016

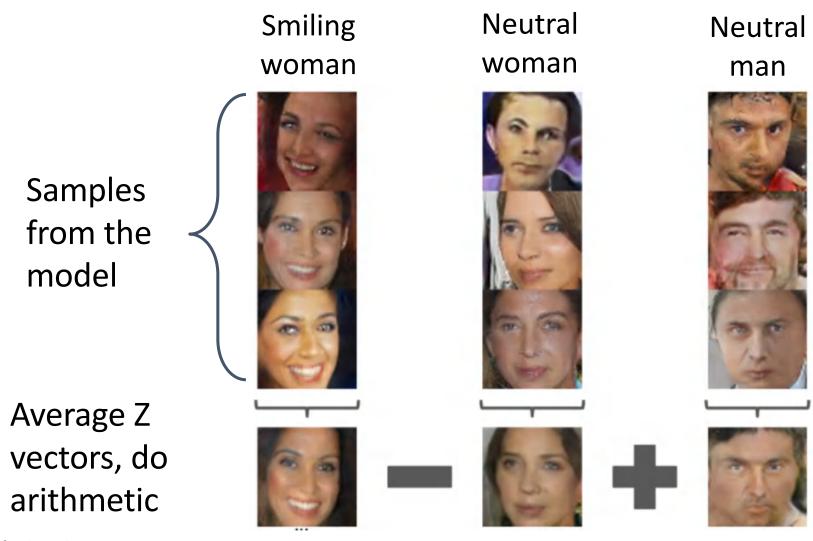
Justin Johnson Lecture 20 - 93 March 30, 2022

Smiling woman Samples from the model



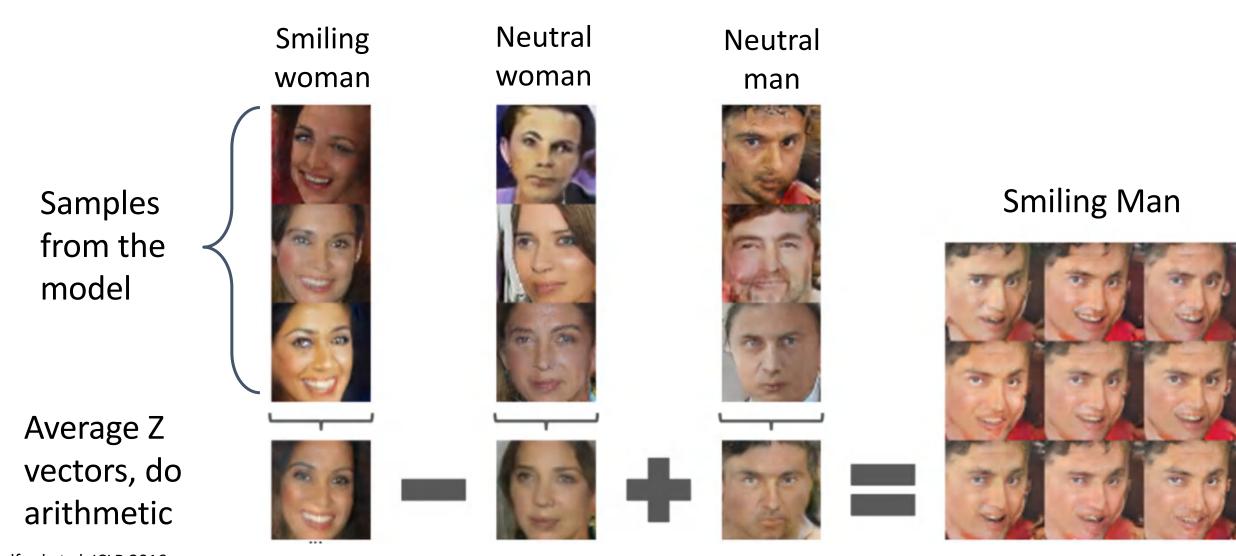


Radford et al, ICLR 2016

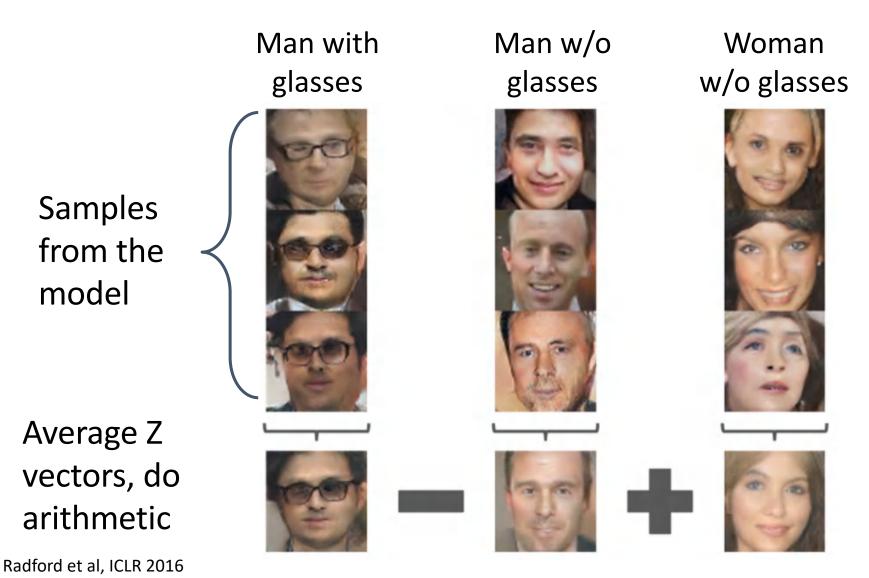


Radford et al, ICLR 2016

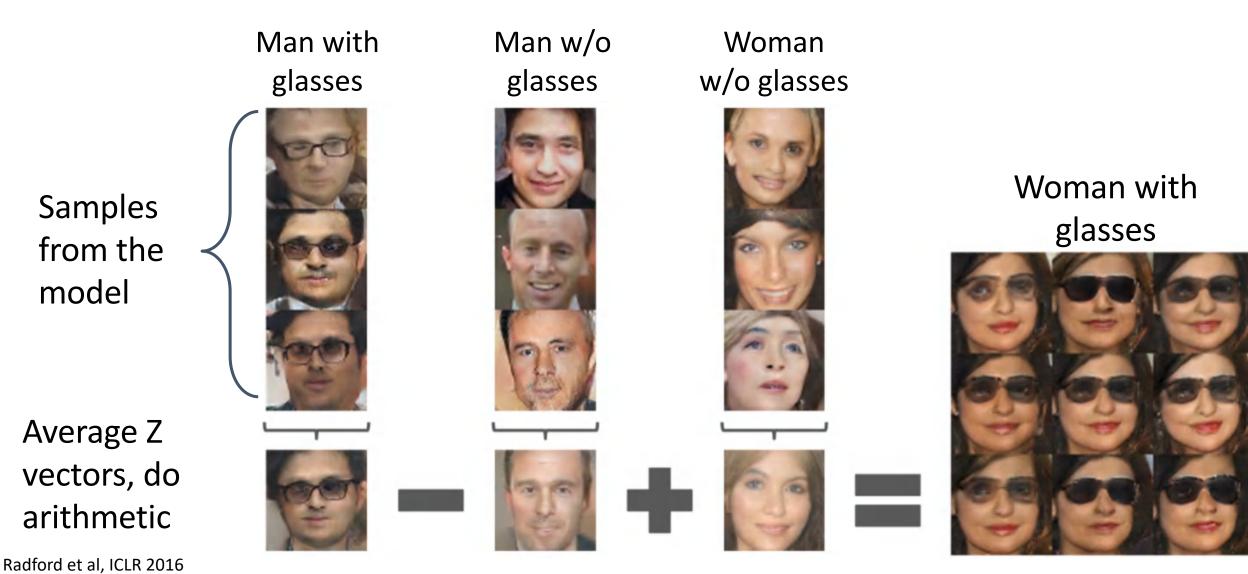
Justin Johnson Lecture 20 - 95 March 30, 2022



Radford et al, ICLR 2016

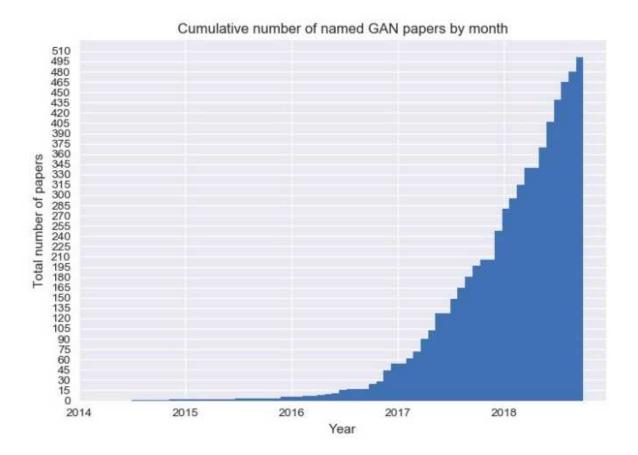


Justin Johnson Lecture 20 - 97 March 30, 2022



Justin Johnson Lecture 20 - 98 March 30, 2022

2017 to present: Explosion of GANs



https://github.com/hindupuravinash/the-gan-zoo

 30-GAM - The Principle of Space of Object - per - In Gamer - Adam and town a fall bailty * 30 WOAF - royed street part Syrum has 10 Dissel Dome from and Joseph · 30 Provider - 30 Physical Learning De Line - Stippers of Anni-Fryd Chie L Colonississis 3B-ManGAM + 3D Trimed Secretal audion Trime a Single Centiff Vive and Agreement Lawrence · ARE DAM - Are Take Adequate the and Control to Tennon street of the Te Agricultural International International . ABC-GAM - NATI- to 1971 (Securities Administrative for each 12 1 Americal Free Materials AC-DAN - Smithten I maps System. With Auntary Community (SAN). . ACRAN - Park Many 10th Continue Descriptor Incommon Many · ACGAN Covering to to sent on 400 to 5 and to 5 and on the property control of · seoAki-Onning manners Compage parmy in LAS · ACTUAL - affinal, arter Chile Laure Felor card Laurence AdSBAN - AdeQMI Reporting Universities Mildells · ABARDA GAN Continuent of Assertant County District Are Class Continued and a . Adventure - Annelsonian Adversarial Transaction Towns Brown on With Roumsidge Couled Salvey of a . ARVGAN - ARTHURNING ARTHURN IN ARTHUR ARTHURN STATEMENT * AE-GAN DE NAME SEVERANDE PROPERTY WITH GAME · All OT - Space Optional Transport for Theorythy Michael · ADDAN - Law on plant as they are by the common and Decree on the cold Held * AF-DCGAN - AT-SCRAN Amphirds Feature Deep Connubitions Deal for Engargement Correction in modes | neal trailion | . APPANA - Adoption MAP Interacts for larger argue in Martin . AM Community informative and Diverse Conventional Personnels on Advancers Informatic * ALPBAN - Linding to linear 4th ledges of United System from Attribute and Security . ALC - Albumanum Cy channel rememor temporal Aligna Aligna Anna Anna Anna Anna Anna Chaire Chaire Chaire Chairmann ann Canadanna Chairmann ARROTHER TOTAL P. AudigDAN - CauseDAN - Constraints deserved in retrocks the desired management AM- GAM - Agricultura hazarraz garant Germaniya Adversarias terris · Ambantilaid - Ambanillaid Consume mysaicium ince measurement (diliub) * AME-BAN - Viding World Chrowith Appendings and Miller Condition ARREAN - Unsuccession America Difference with Generaliza Asserbance Asserbance to Global 4 ARD - Advanced to Data Linear at Security Server belong Posterior . APT-GAM - WPG-GAM AWAY AS PRINCIPALING A REMAIN WITH GAME ARAE - surroughney fleuclinized Automorphism for Exemples Describe Structures automorphism · AHDA · Advances - Supremotions (seeing to Common Association) · ARIOAN : ARIOAN Symbolic Assessment Sonic solon Construct a resonante had + Artitlate - Artifate Arting & Symmetrical Conditional Cologonic State. * AND STANY MILITARY DESCRIPTION AND STANDARD ST AZÁ-SAN - AMERICO-AMBIE Commune Anyansoni horacost (AZA-SAN) . Attention-DAY: Attention size to the Transportation in Whit Images · ANDREAS Advance for a American Salvey Deep Course with the West formation Artiflak - ImiGAL Fig-Organic Tyri - tauge denomina with Altergram Defension strangers from the (ground) · AVES + A'mit deligencemal Vision Programmy Colombia . B-DCOAN - B-DCOAN Resignatory? Bluesses DCOAN In 1990A Include: - Decreasing appearance feels from a premise lighter believed by Parameters. . BASTAN - BACTAN TANK A JOHN MARKET WITH BARRIETS DANS Have an GAN - Green and manufactures maked browns. . Bayesurt ORty - Bayesian Core (arresto. . BCDAH - Dayward Commission Severation Assessed Names BEGAN - United time ("Incident flame alles Advances lambury) · BERM CONTRACT LONGISH ASSESSMENT MARKETING * REDAY - BESIAN payoting bag spoon generalise American Nationals . BEGAN-CS CAMBRING THEN GROWING AMERICA IS SUMMINGHAR SHOULD . British DAN - Disconspised Maliyanan Pring and acres on Commission on the Insur

 BOHH «Strary Generalize Asperse of Security Sciences (prince) · Bi-GAM - Automorranty and Smallaneous - Salving Deep Mater Malace Automates by 8 miles mergition Afford our of Nationary Address thereon & Algorithm BacycleDiff: Trouble to diagrams in age-lat-rooms "monorain increase." . MAN - Advenue of Francy Courses SMSAN - Birthing Commer | Sharp Place place with a responsible PAN . SmallSAV - Smalltaff Statement of Phone and All Company of the C Scansing by the property of the party of the BridgaDAN - Generalise Assumation Frontis Vision in New York Systems . \$5-\$A1 - BILDBUY-Septing Controlly Assertable Transport . SWIDAY DURING WASHINGTON DAY OAH Fore figure was Command community deleganglesses. · CARRIADAR CORRESPAN, Continue o marcine residence de la companya del companya de la companya de la companya del companya de la companya del la companya de Calvideti -, Calvideti - Simparing 70 man primary francoir Slowers in Musiciana. consequence Commissions with Consequence Assessing Consequence (approxi-* CAN - CAN Creative Agreement Nectoritis, Something All to Learning Apold Styles and Districting Pages Store Storens * Days DAN Company damp Symentic Replay for Symentics Advanced Million I . Carmedan - Committee Dentrolly Arms - Committee Committee · CalCAN - I heaper and and Tentimone and severy sich Calicolina femore - According . DARDAN, CONDAN, COMMAN ABSORDANCE PROPERTY for Develop Department . Carrellon - Camadon manny Cama Ingels Commune Augus and Asserta Daling · VCGGBB - Twist-Transport Country with Contract Communical Research and Specialist Security · NA DAY CONSTRUM I MADE TO MADE TO ANNAULT COROAN: Communicacy Color-Depositure: Repositive your Communicative converse. . CE-DAN - Green Converting for Februaries Critical Constitution visital Clays Expansion Committee . CSB-BAN - Compared to not resid to where Learning of Decreation Advanced Market . ICOM - CHIMNAN I CHIMNAN ANNING WHITE * VICAN - Provincially, Garagement Associated Malaysia * Children GAN - An Price Country Reposed in Committee Advanced Nationals . SCAN - CONTROL CONTROL CARD TO USE AUGUSTOSIS IN MARRIADOR CARDADOR . Dynamian summercial months away today marries have . CHINNELL CHINNELL CHINESING CHINNELL CHINNELL OF CHINNELL AND CHINA MICHIGAN · Chi-Okin - Continies Compression Description Community Statements for Community . Collis BAN - An Tour terring in Mat Seminard Whitel Godge Discretion Street Advanced St . CSGAN - CALLERY CHARGING ARTERIAN ARMITTE · Control Common Common Common States for the Printer Printer Common Summon . Conceptibility a Laurence Companitions Would Concepts with Mutual Conceptions Candidow syneight - Commission CycleGavino America Games Face I year Generalia · mercratiofiles - tienspiles decreate Atenualisms and Contracting field Contract-Right-Galler Contraction RMI - NAMe for Assured Westering Briggers Generalized · STATE OF THE PARTY OF THE PAR Confered COV 1 contracts (24 to 2000) Species from the machine on process a best in * Committee Province of Statement will be well become a contracted a second on the second . DOWNEY - TOTAL BODY AND REAL PROPERTY AND ADDRESS OF LANCE OF AN ADDRESS OF A LANCE OF AN ICE-ITAN - CIT SULS TRANSPORT CONTRIBUTE BOUNDARY OF THE SULS TO COMPANY · Cramer BAIs - The Cremer Dates on a Serution in B. Williams Drauen . CORRECTARY - CONTROL CONTROL AND ANALYSIA ANAL . STRAB-SARI - Common Resourced Surregular Assessment · Cl-GAR-Transact Association Transactor of Condition Temperatures + CHE - month storm Extratory lateral and with Contrator assumption Considera-· CDOM CLUVE OW Inthematics and the second of t * The UNIT Colors have to some think of the color of the colors of the color of the

GAN Improvements: Improved Loss Functions

Wasserstein GAN (WGAN)



Arjovsky, Chintala, and Bouttou, "Wasserstein GAN", 2017

WGAN with Gradient Penalty (WGAN-GP)



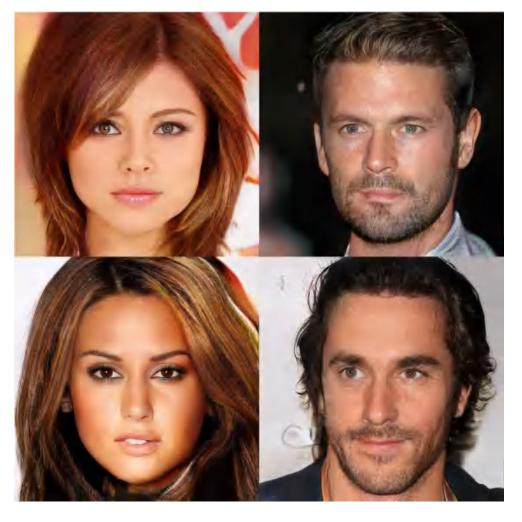
Gulrajani et al, "Improved Training of Wasserstein GANs", NeurIPS 2017

GAN Improvements: Higher Resolution

256 x 256 bedrooms



1024 x 1024 faces



Karras et al, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

Justin Johnson Lecture 20 - 101 March 30, 2022

GAN Improvements: Higher Resolution

512 x 384 cars

1024 x 1024 faces





Karras et al, "A Style-Based Generator Architecture for Generative Adversarial Networks", CVPR 2019

<u>Images</u> are licensed under <u>CC BY-NC 4.0</u>



Source: https://drive.google.com/drive/folders/1NFO7_vH0t98J13ckJYFd7kuaTkyeRJ86

StyleGAN2



Karras et al, "Analyzing and Improving the Image Quality of StyleGAN", CVPR 2020

Justin Johnson Lecture 20 - 104 March 30, 2022

Conditional GANs

Recall: Conditional Generative Models learn p(x|y) instead of p(x) Make generator and discriminator both take label y as an additional input!

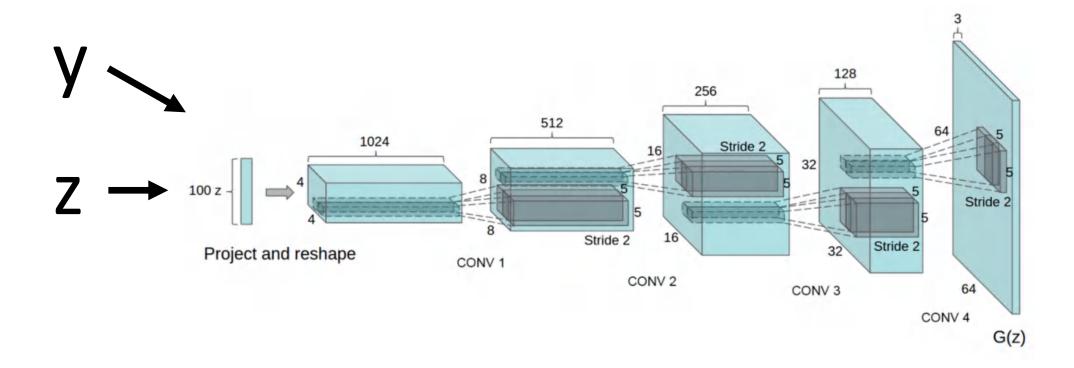


Figure credit: Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

Justin Johnson Lecture 20 - 105 March 30, 2022

Conditional GANs: Conditional Batch Normalization

Batch Normalization

$$\mu_{j} = \frac{1}{N} \sum_{i=1}^{N} x_{i,j}$$

$$\sigma_{j}^{2} = \frac{1}{N} \sum_{i=1}^{N} (x_{i,j} - \mu_{j})^{2}$$

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_{j}}{\sqrt{\sigma_{j}^{2} + \epsilon}}$$

$$y_{i,j} = \gamma_{j} \hat{x}_{i,j} + \beta_{j}$$

Learn a separate scale and shift for each different label y

Conditional Batch Normalization

$$\mu_{j} = \frac{1}{N} \sum_{i=1}^{N} x_{i,j}$$

$$\sigma_{j}^{2} = \frac{1}{N} \sum_{i=1}^{N} (x_{i,j} - \mu_{j})^{2}$$

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_{j}}{\sqrt{\sigma_{j}^{2} + \epsilon}}$$

$$y_{i,j} = \gamma_{j}^{y} \hat{x}_{i,j} + \beta_{j}^{y}$$

$$y_{i,j} = \boldsymbol{\gamma_j^y} \hat{x}_{i,j} + \boldsymbol{\beta_j^y}$$

Dumoulin et al, "A learned representation for artistic style", ICLR 2017

Conditional GANs: Spectral Normalization

Welsh springer spaniel



Fire truck





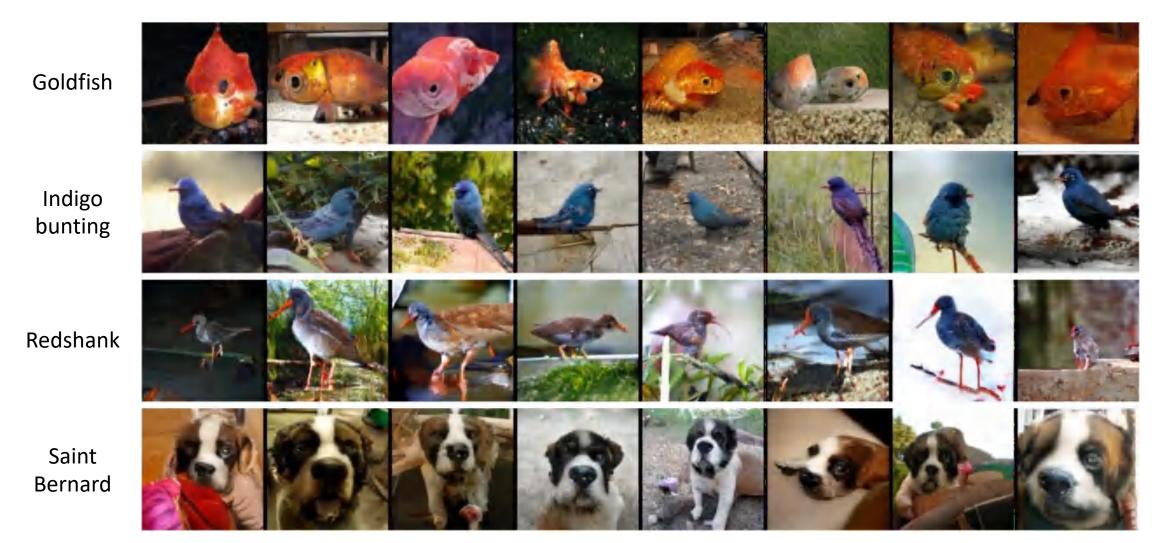


Miyato et al, "Spectral Normalization for Generative Adversarial Networks", ICLR 2018

128x128 images on ImageNet

Justin Johnson March 30, 2022 Lecture 20 - 107

Conditional GANs: Self-Attention



Zhang et al, "Self-Attention Generative Adversarial Networks", ICML 2019

128x128 images on ImageNet

Justin Johnson Lecture 20 - 108 March 30, 2022

Conditional GANs: BigGAN



Brock et al, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", ICLR 2019

512x512 images on ImageNet

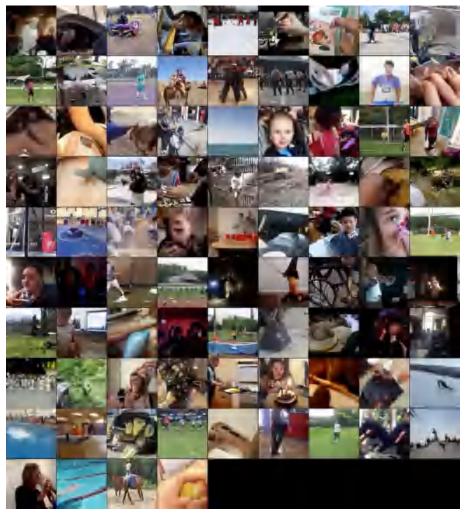
Justin Johnson Lecture 20 - 109 March 30, 2022

Generating Videos with GANs



64x64 images, 48 frames

https://drive.google.com/file/d/1FjOQYdUuxPXvS8yeOhXdPQMapUQaklLi/view

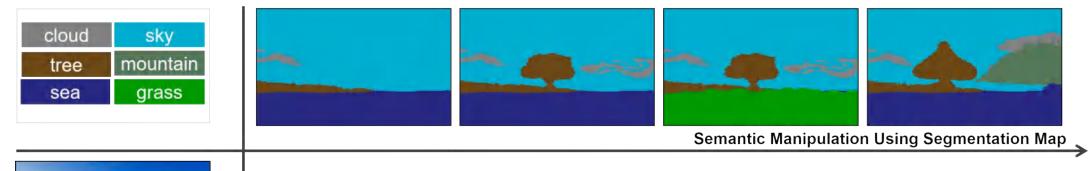


128x128 images, 12 frames

https://drive.google.com/file/d/165Yxuvvu3viOy-39LhhSDGtczbWphj_i/view

Label Map to Image

Input: Label Map



Input: Style Image



Park et al, "Semantic Image Synthesis with Spatially-Adaptive Normalization", CVPR 2019

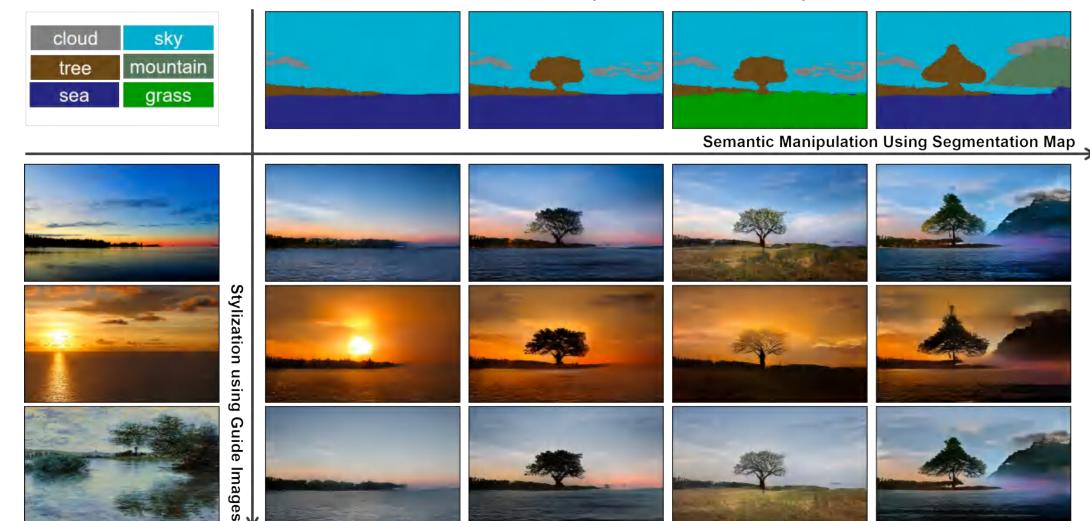
Label Map to Image

Input:

Image

Style

Input: Label Map

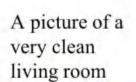


Park et al, "Semantic Image Synthesis with Spatially-Adaptive Normalization", CVPR 2019

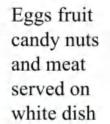
Conditioning on more than labels! Text to Image

This bird is red and brown in color, with a stubby beak The bird is short and stubby with yellow on its body A bird with a medium orange bill white body gray wings and webbed feet

This small black bird has a short, slightly curved bill and long legs



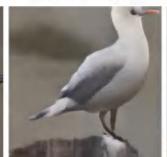
A group of people on skis stand in the snow



A street sign on a stoplight pole in the middle of a day

















Zhang et al, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks.", TPAMI 2018
Zhang et al, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks.", ICCV 2017
Reed et al, "Generative Adversarial Text-to-Image Synthesis", ICML 2016

Text to Image: DALL-E

Step 1: Train VQ-VAE (discrete grid of latent codes)

Step 2: Train autoregressive
Transformer model to predict
sequence of latent codes
(Giant model on 250M
image/text pairs)

Step 3: Given text prompt, sample new image codes; pass through VQ-VAE decoder to generate images

Ramesh et al, "Zero-Shot Text-to-Image Generation", ICML 2021

Text to Image: DALL-E

Step 1: Train VQ-VAE (discrete grid of latent codes)

Step 2: Train autoregressive
Transformer model to predict
sequence of latent codes
(Giant model on 250M
image/text pairs)

Step 3: Given text prompt, sample new image codes; pass through VQ-VAE decoder to generate images









an illustration of a baby hedgehog in a christmas sweater walking a dog

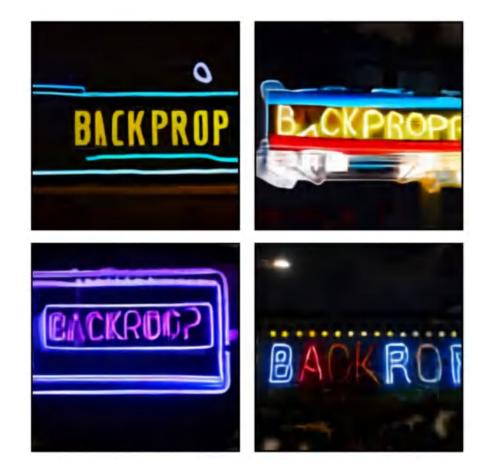
Ramesh et al, "Zero-Shot Text-to-Image Generation", ICML 2021

Text to Image: DALL-E

Step 1: Train VQ-VAE (discrete grid of latent codes)

Step 2: Train autoregressive
Transformer model to predict
sequence of latent codes
(Giant model on 250M
image/text pairs)

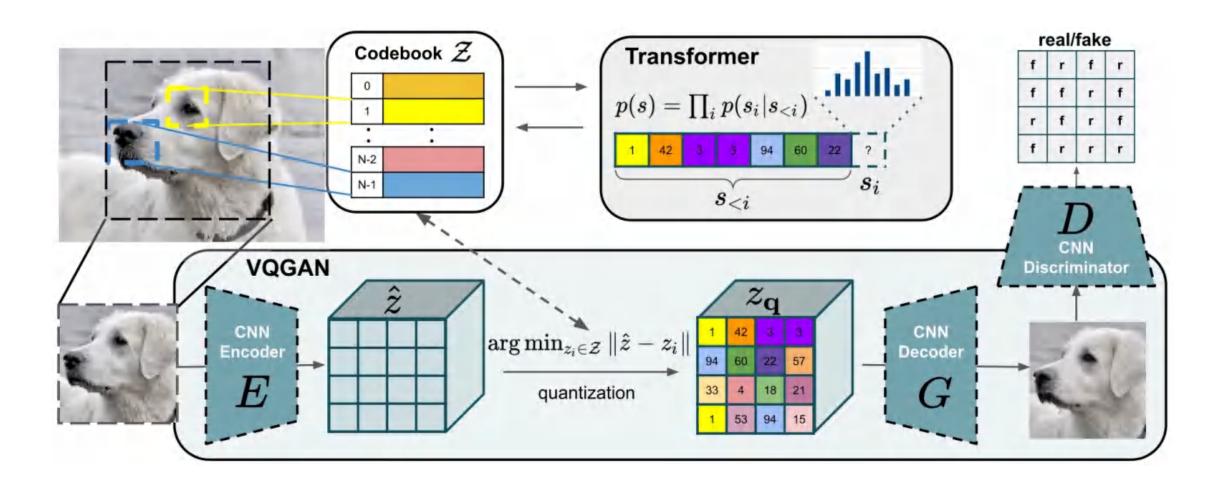
Step 3: Given text prompt, sample new image codes; pass through VQ-VAE decoder to generate images



a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

Ramesh et al, "Zero-Shot Text-to-Image Generation", ICML 2021

VQ-GAN



Esser et al, "Taming Transformers for High-Resolution Image Synthesis", CVPR 2021

Justin Johnson Lecture 20 - 117 March 30, 2022

VQ-GAN (Semantic Segmentation to Image)

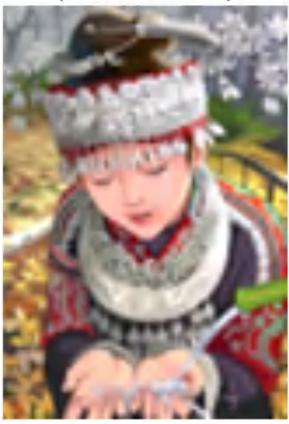


Esser et al, "Taming Transformers for High-Resolution Image Synthesis", CVPR 2021

Justin Johnson Lecture 20 - 118 March 30, 2022

Image Super-Resolution: Low-Res to High-Res

bicubic (21.59dB/0.6423)



SRResNet (23.53dB/0.7832)



SRGAN (21.15dB/0.6868)



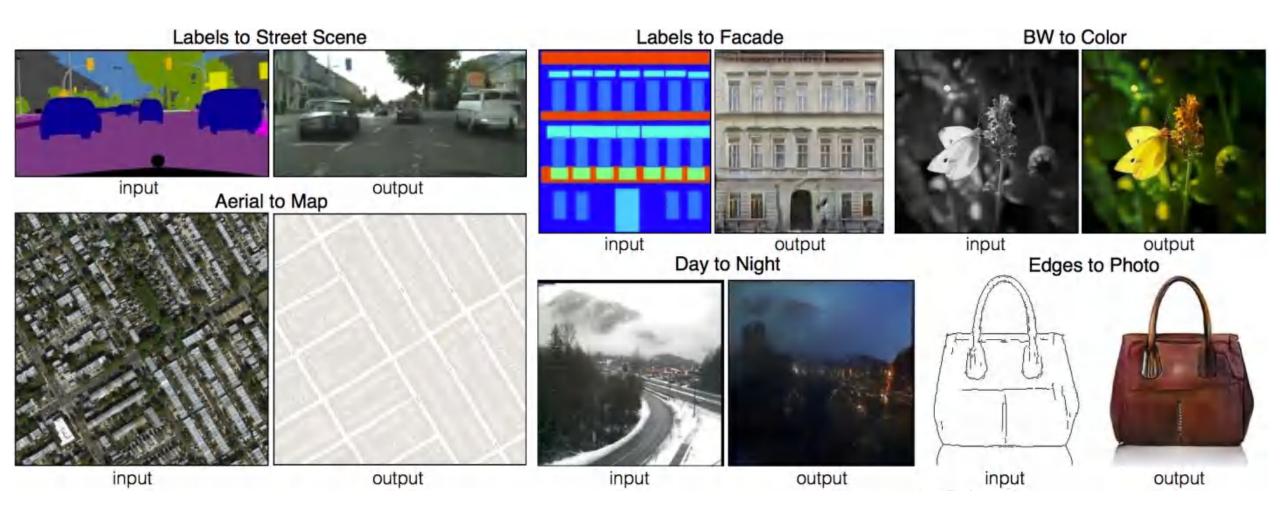
original



Ledig et al, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", CVPR 2017

Justin Johnson Lecture 20 - 119 March 30, 2022

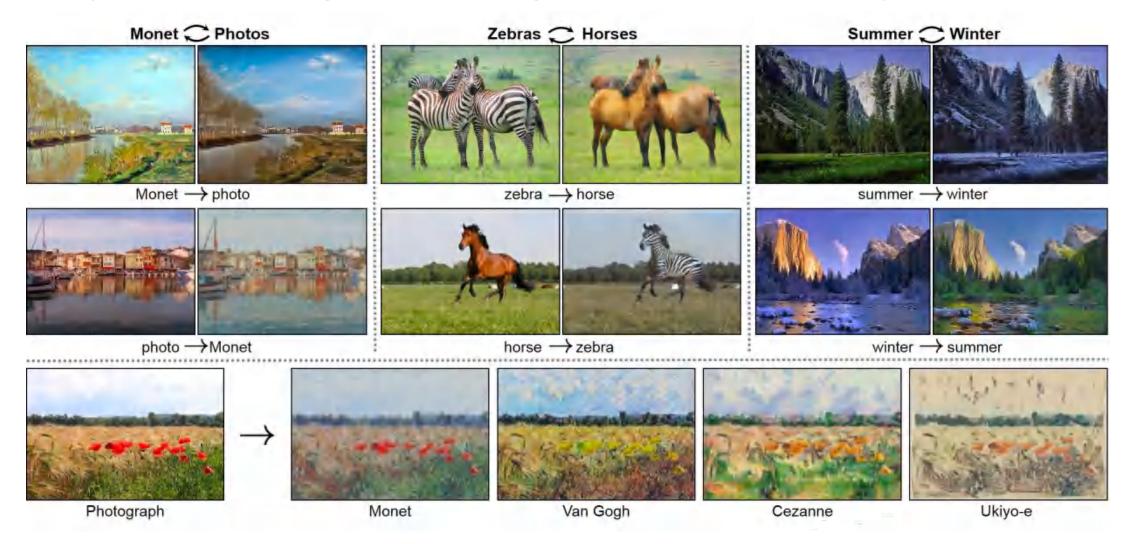
Image-to-Image Translation: Pix2Pix



Isola et al, "Image-to-Image Translation with Conditional Adversarial Nets", CVPR 2017

Justin Johnson Lecture 20 - 120 March 30, 2022

Unpaired Image-to-Image Translation: CycleGAN



Zhu et al, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", ICCV 2017

Justin Johnson Lecture 20 - 121 March 30, 2022

Unpaired Image-to-Image Translation: CycleGAN

Input Video: Horse Output Video: Zebra

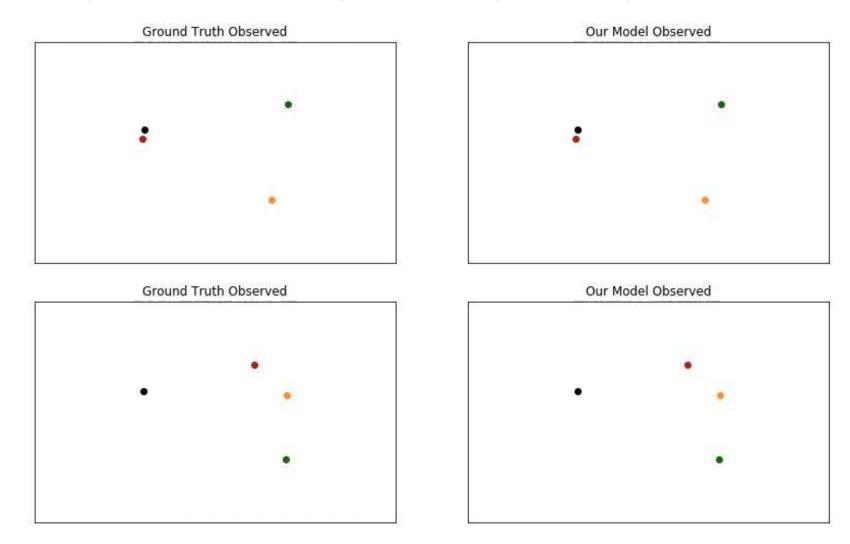


https://www.youtube.com/watch?v=9reHvktowLY

Zhu et al, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", ICCV 2017

Justin Johnson Lecture 20 - 122 March 30, 2022

GANs: Not just for images! Trajectory Prediction



Gupta, Johnson, Li, Savarese, Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks", CVPR 2018

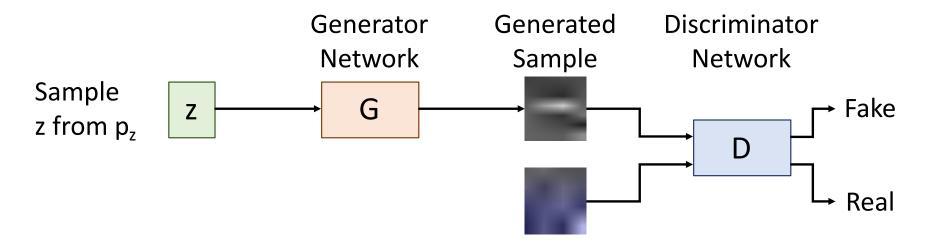
Justin Johnson Lecture 20 - 123 March 30, 2022

GAN Summary

Jointly train two networks:

Discriminator: Classify data as real or fake

Generator: Generate data that fools the discriminator



Under some assumptions, generator converges to true data distribution Many applications! Very active area of research!

Justin Johnson Lecture 20 - 124 March 30, 2022

Taxonomy of Generative Models

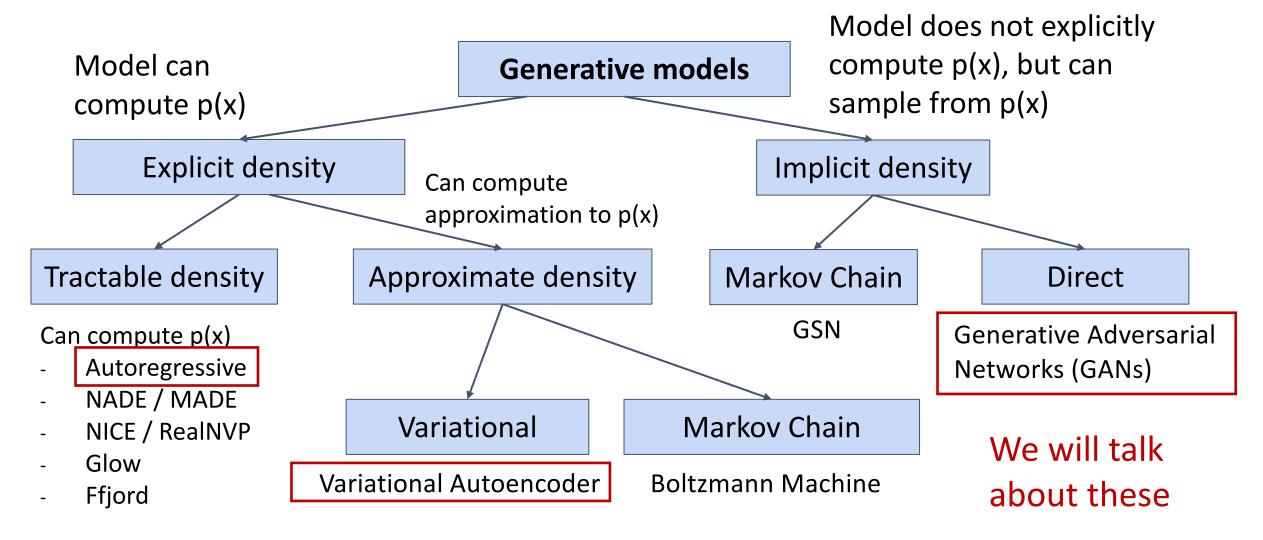


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Justin Johnson Lecture 20 - 125 March 30, 2022

Generative Models Summary

Autoregressive Models directly maximize likelihood of training data:

$$p_{\theta}(x) = \prod_{i=1}^{N} p_{\theta}(x_i|x_1, ..., x_{i-1})$$

Good image quality, can evaluate with perplexity. Slow to generate data, needs tricks to scale up.

Variational Autoencoders introduce a latent z, and maximize a lower bound:

$$p_{\theta}(x) = \int_{Z} p_{\theta}(x|z)p(z)dz \ge E_{z \sim q_{\phi}(Z|X)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

Latent z allows for powerful interpolation and editing applications.

Generative Adversarial Networks give up on modeling p(x), but allow us to draw samples from p(x). Difficult to evaluate, but best qualitative results today

Justin Johnson Lecture 20 - 126 March 30, 2022

Next Time: Visualizing Models and Generating Images