CENG 506 Deep Learning

Lecture 12 – Generative Models

Slides were prepared using the course material of
Stanford's CNN Course (CS231n by Fei-Fei, Johnson, Yeung)

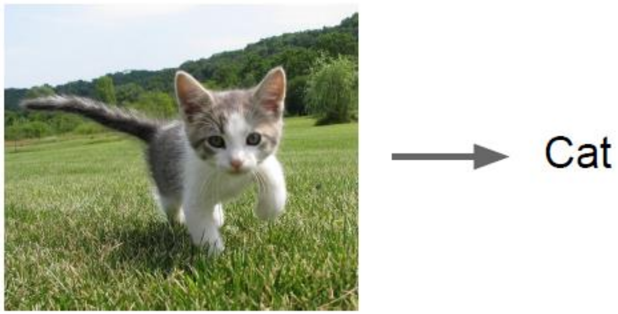
Supervised vs. Unsupervised Learning

Supervised 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.



Classification

Semantic Segmentation

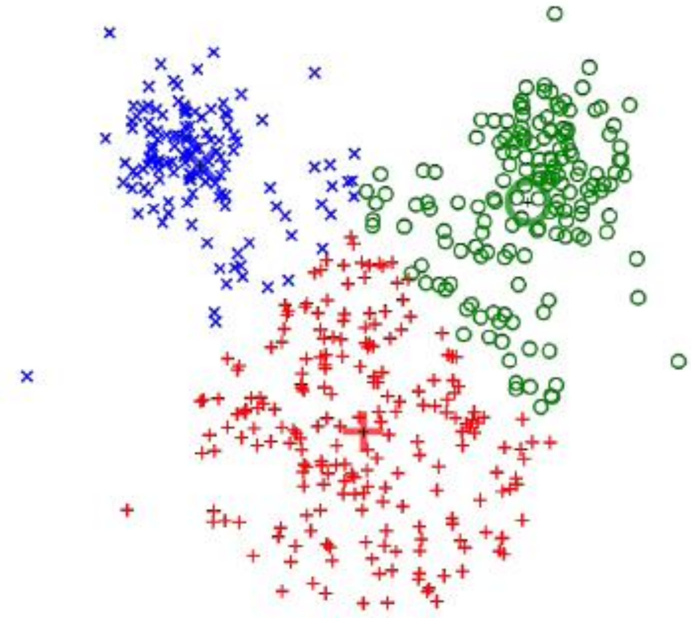
Supervised vs. Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

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

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



K-means clustering

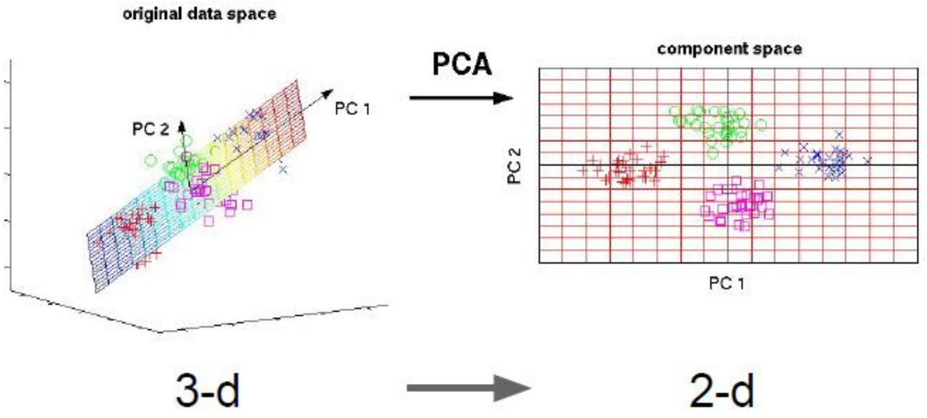
Supervised vs. Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

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

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



Principal Component Analysis
(Dimensionality reduction)

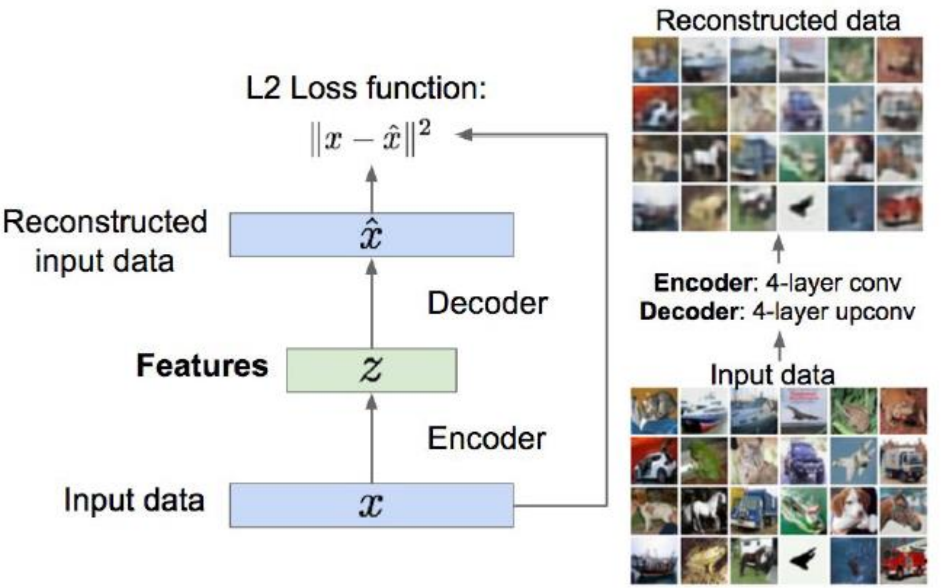
Supervised vs. Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

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

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



Autoencoders
(Feature learning)

Supervised vs. Unsupervised Learning

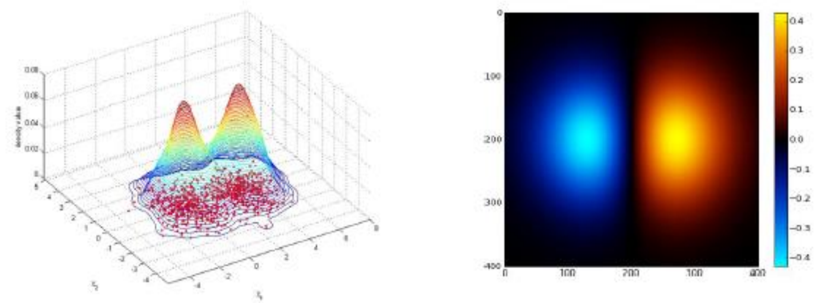
Unsupervised Learning

Data: x
Just data, no labels!

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

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

1-d density estimation



2-d density estimation

Generative Models

Given training data, generate new samples from same distribution. This addresses density estimation.

Training data ~ $p_{data}(x)$ Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Generative model is the opposite of discriminative model which learns $p(y|x)$, for instance classification boundaries.

Why Generative Models?

- Realistic samples for artwork, super-resolution, colorization, etc.


- Generative models of time-series data can be used for simulation/planning.

Generative Models

Two main streams in density estimation:

1) Explicit density estimation:
   explicitly define and solve for $p_{model}(x)$
2) Implicit density estimation:
   learn model that can sample from $p_{model}(x)$
   without explicitly defining it

Generative Adversarial Networks (GANs)

- GANs: An implicit way. We just want ability to sample.
- We take game-theoretic approach: learn to generate from training distribution through 2-player game

Some Background: Autoencoders

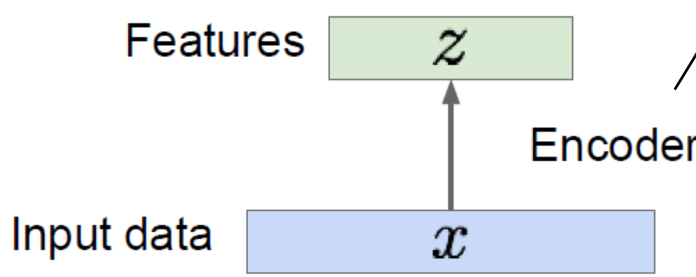
Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

z usually smaller than **x** (dimensionality reduction)

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN

Q: Why dimensionality reduction?

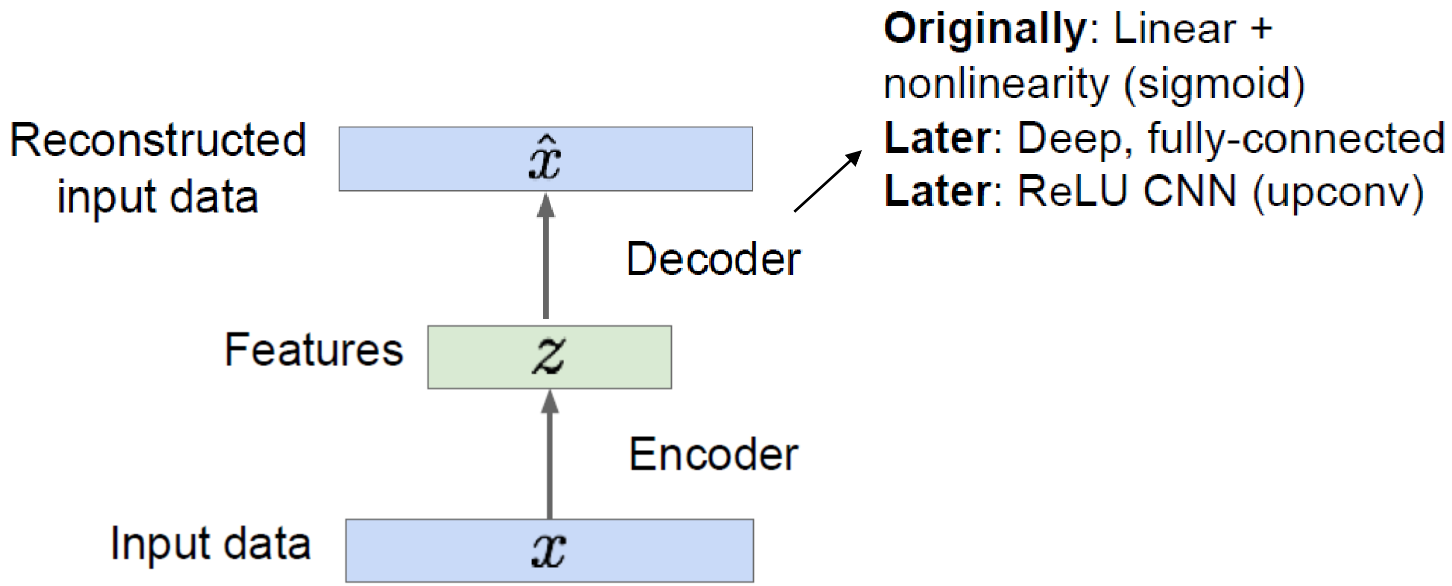
A: Want features to capture meaningful factors of variation in data



Some Background: Autoencoders

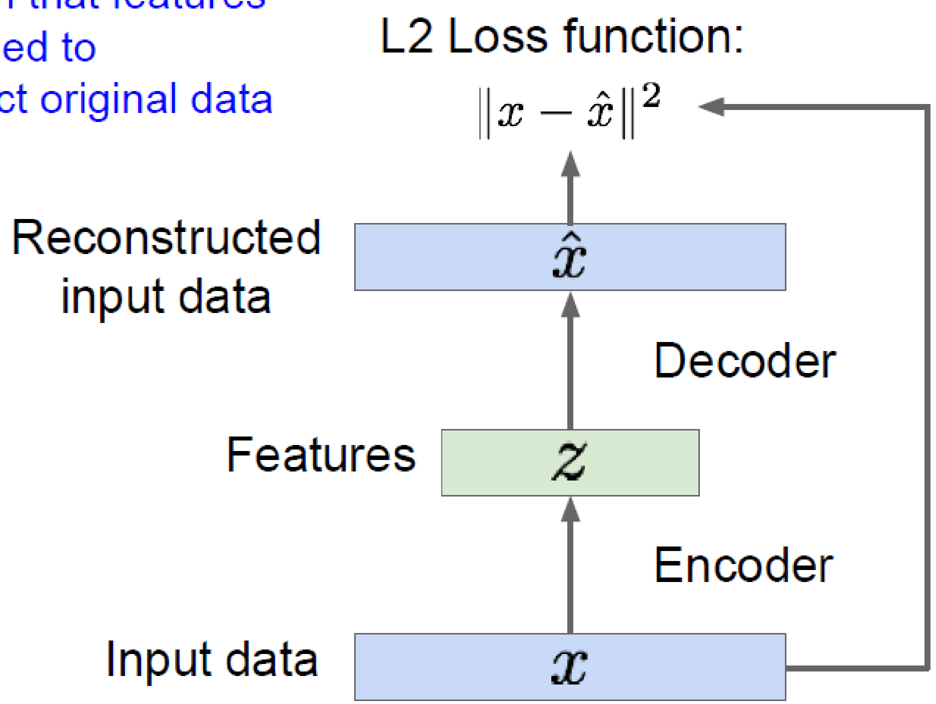
Q. How to learn this feature representation?

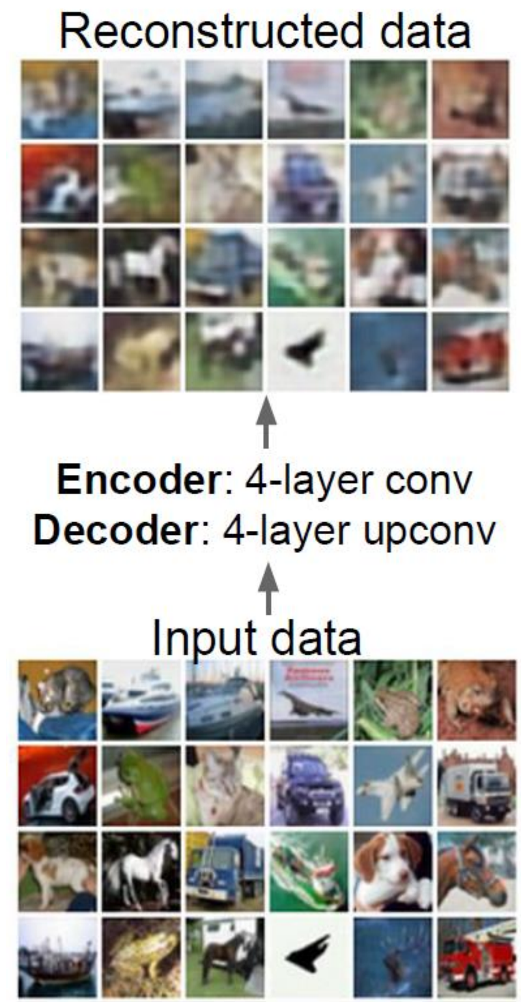
A. Train such that features can be used to reconstruct original data ("Autoencoding" - encoding itself)



Some Background: Autoencoders

Train such that features can be used to reconstruct original data





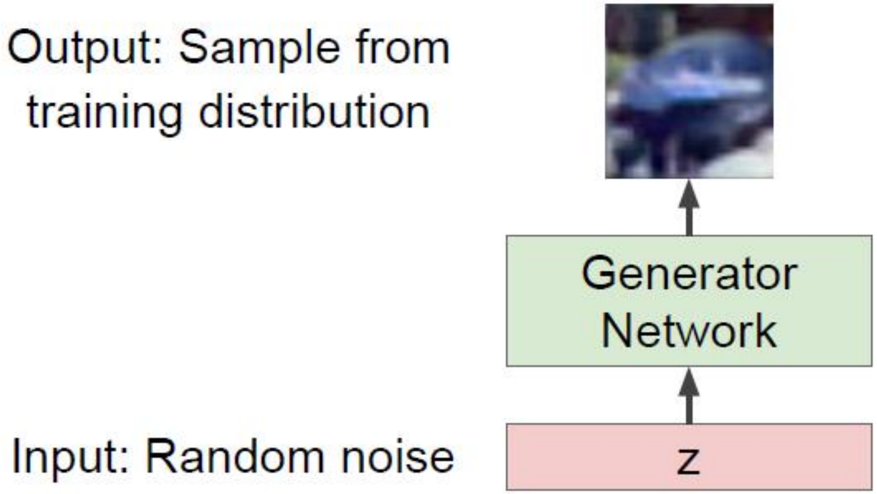
Generative Adversarial Networks (GANs)

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!

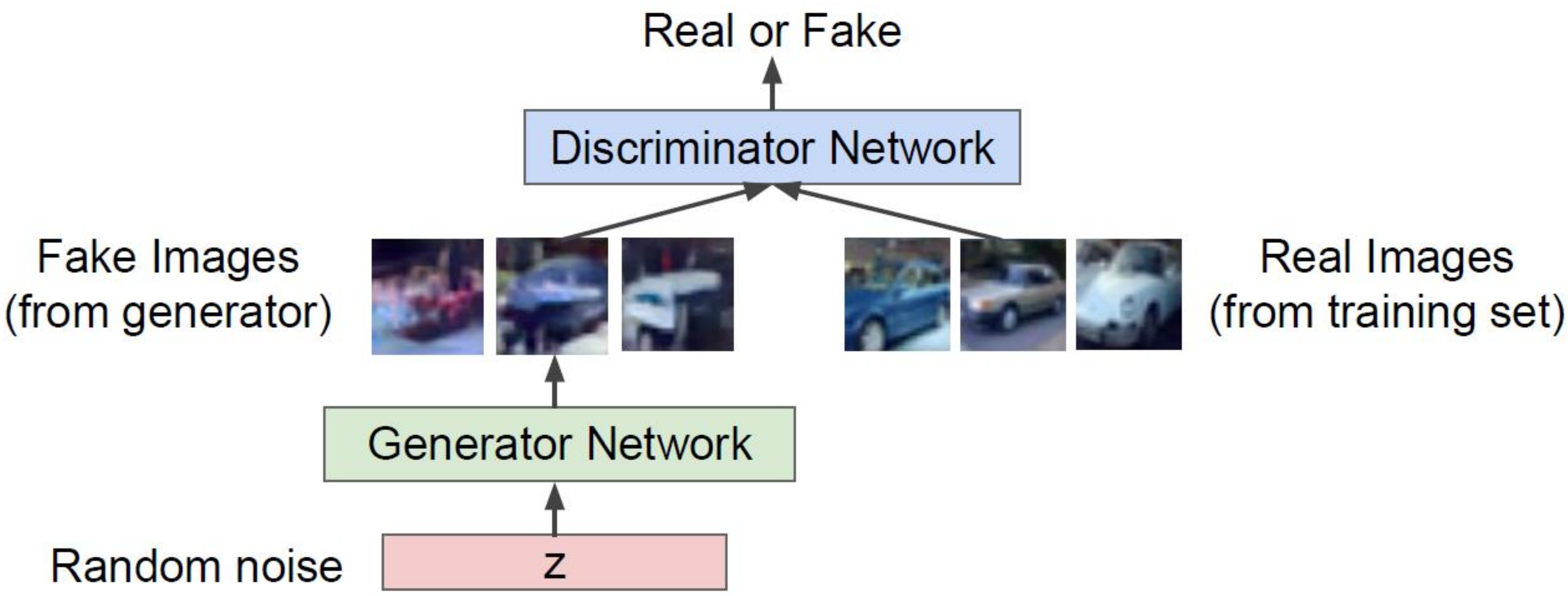


Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Training GANs: Two-player game

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images



Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Training GANs: Two-player game

Train Generator network and Discriminator network jointly in **minimax game.**

Minimax objective function:

x: real data
G(z): generated fake data
Discriminator outputs between (0,1)

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for G(z)}}) \right]$$

- Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)
- Note: max value of log D(x) is zero when D(x)=1,
  min value of log D(x) is –inf when D(x)=0.

Training GANs: Two-player game

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Training GANs: Two-player game

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))\right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))\right]$$

2. **Instead: Gradient ascent** on generator, different objective

$$\max_{\theta_g} \mathbb{E}_{z\sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Training GANs: Two-player game

Putting it together: GAN training algorithm

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:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

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 ascending its stochastic gradient (improved objective):

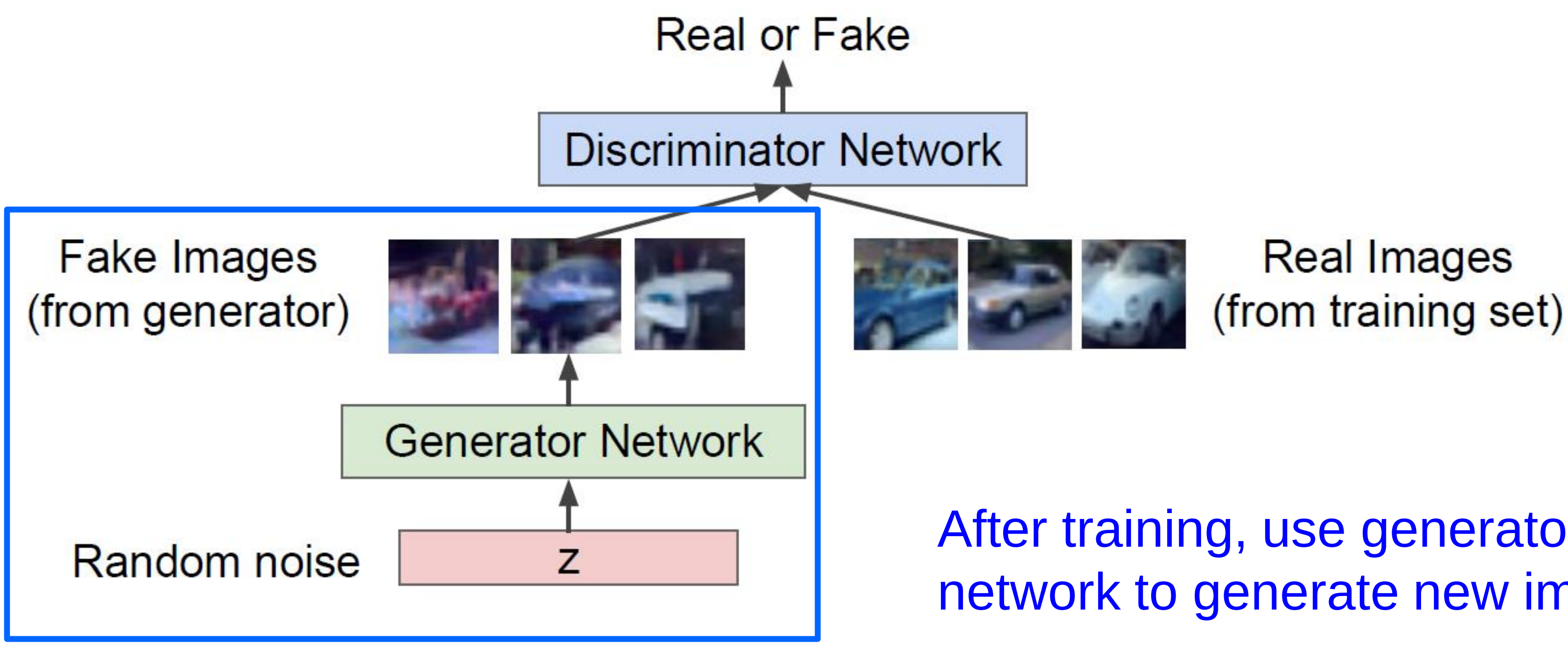
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Training GANs: Two-player game

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images

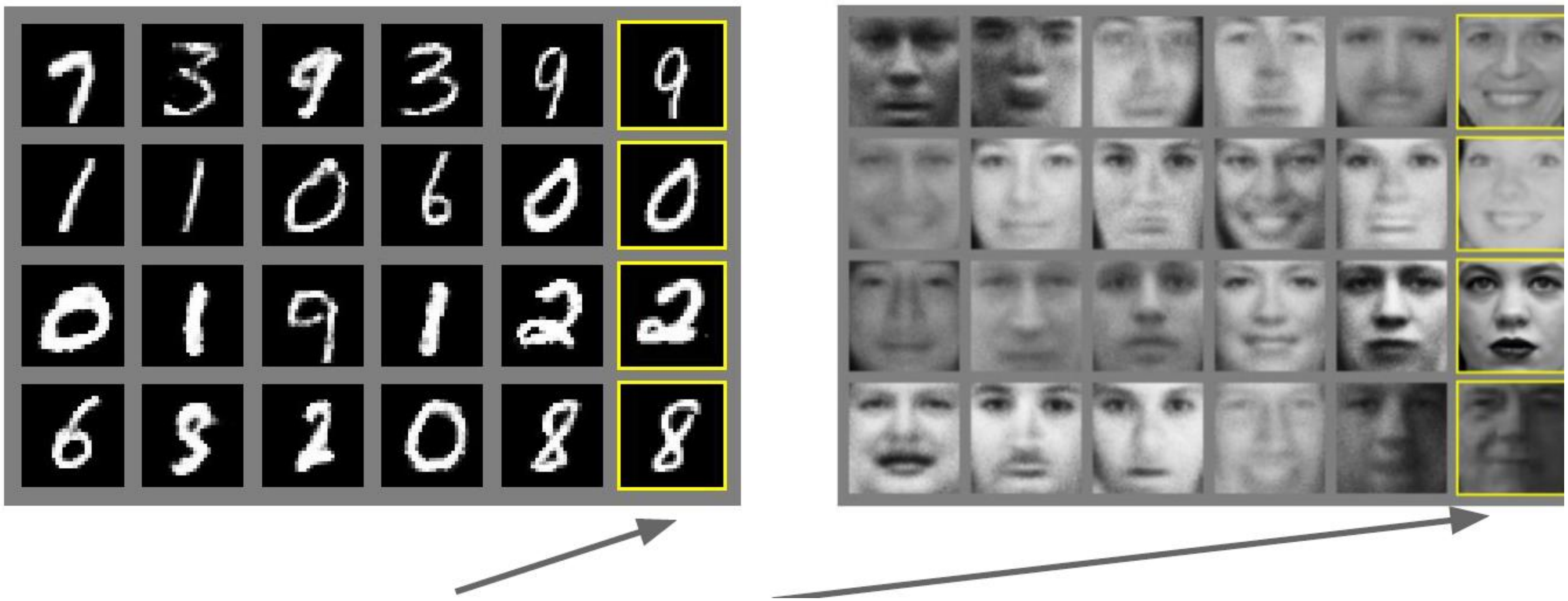


After training, use generator network to generate new images

Ian Goodfellow et al., "Generative Adversarial Nets", NIPS 2014

GAN Results

Generated samples

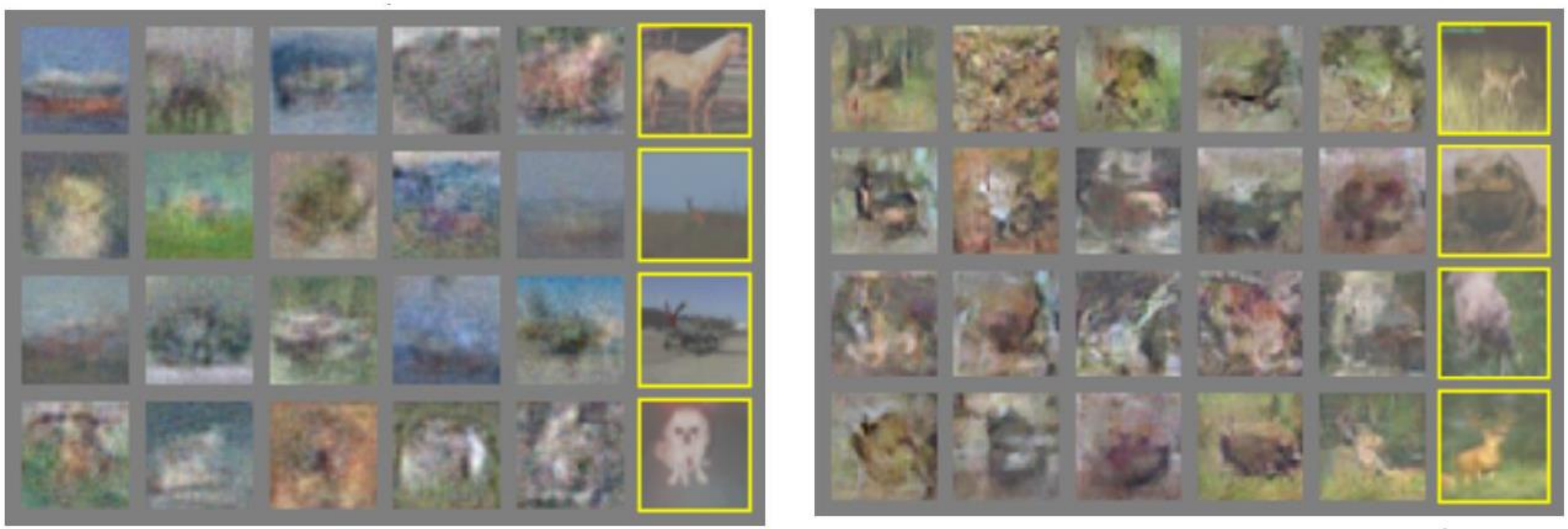


Nearest neighbor from training set for the column before
(first 5 columns are generated samples)

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

GAN Results

Generated samples (CIFAR-10)



Results not so good as of 2014

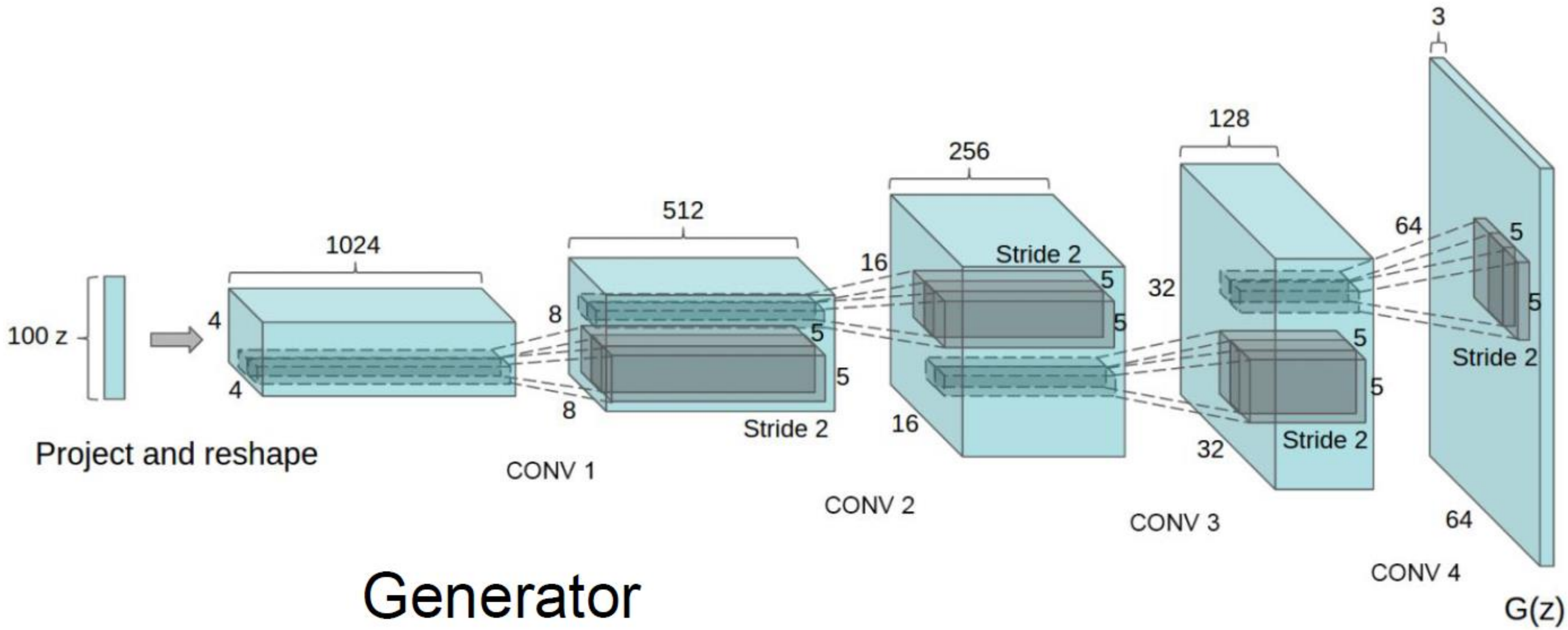
Nearest neighbor from training set for the right-most column

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

GANs: Convolutional Architectures

Generator is an upsampling network with fractionally-strided convolutions

Discriminator is a convolutional network



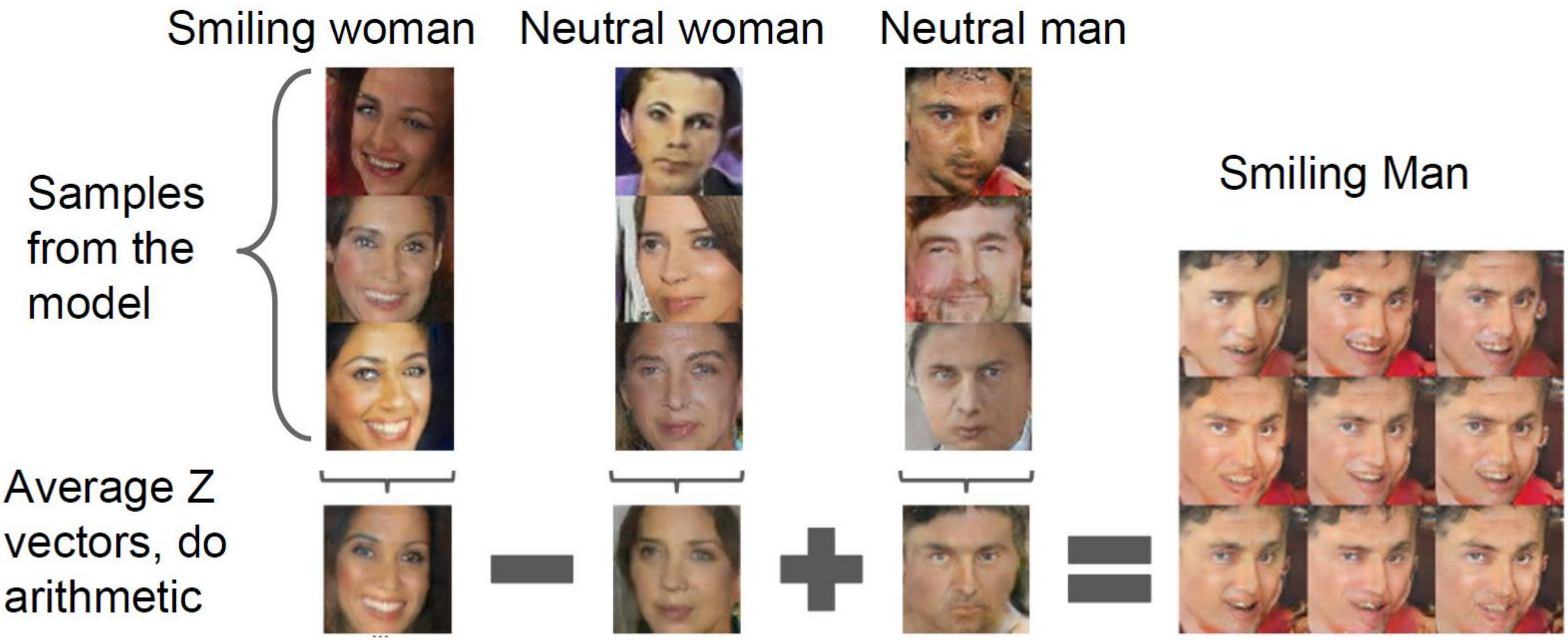
Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

GANs: Convolutional Architectures

Samples from the model look amazing!

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

GANs: Interpretable Vector Math



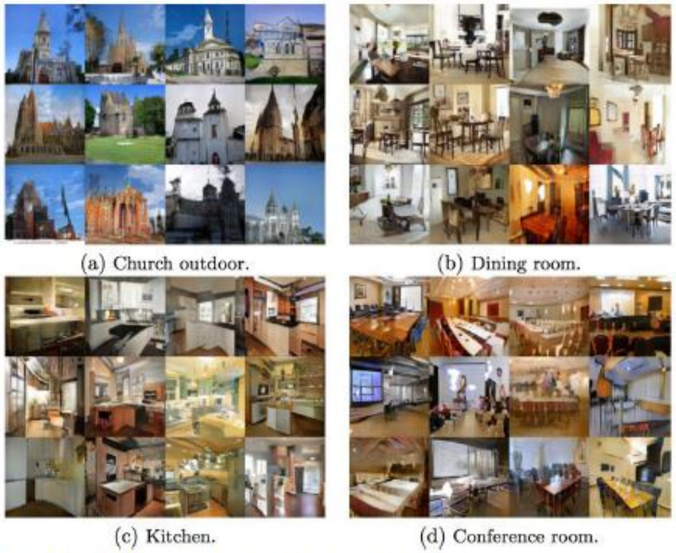
Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

GANs: Interpretable Vector Math

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

Many GAN applications

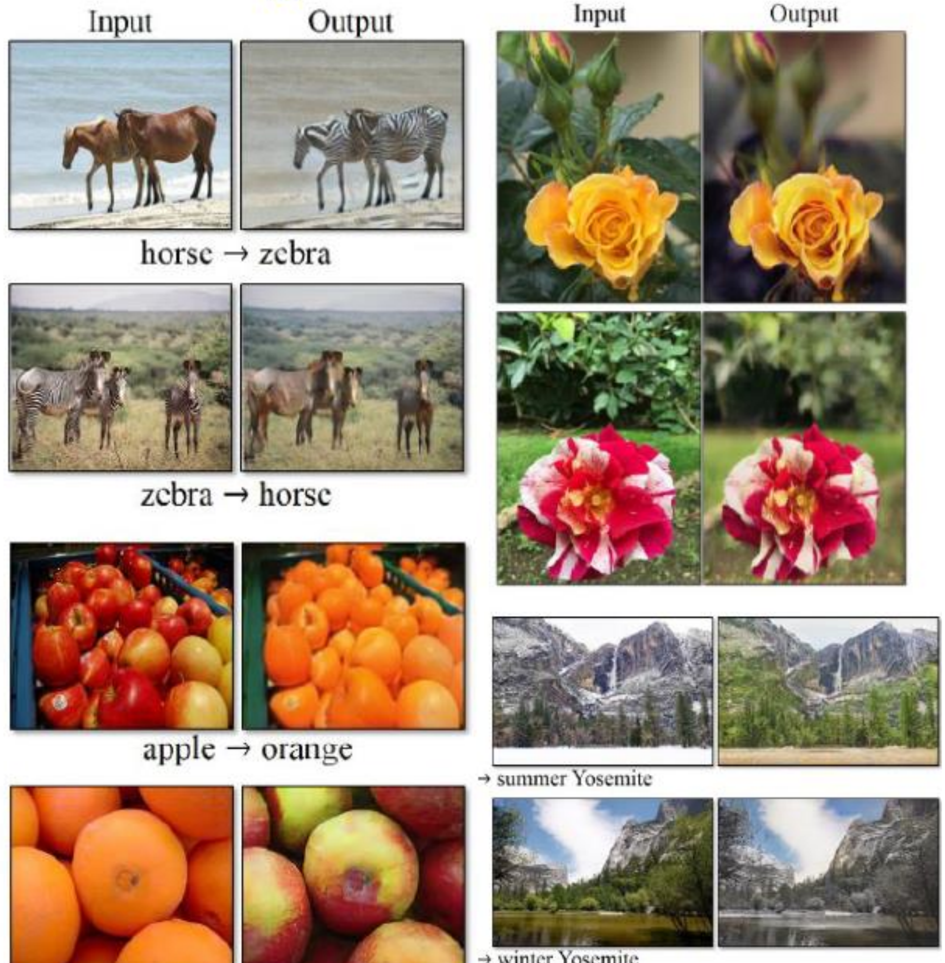
Better training and generation



LSGAN. Mao et al. 2017.

BEGAN. Bertholet et al. 2017.

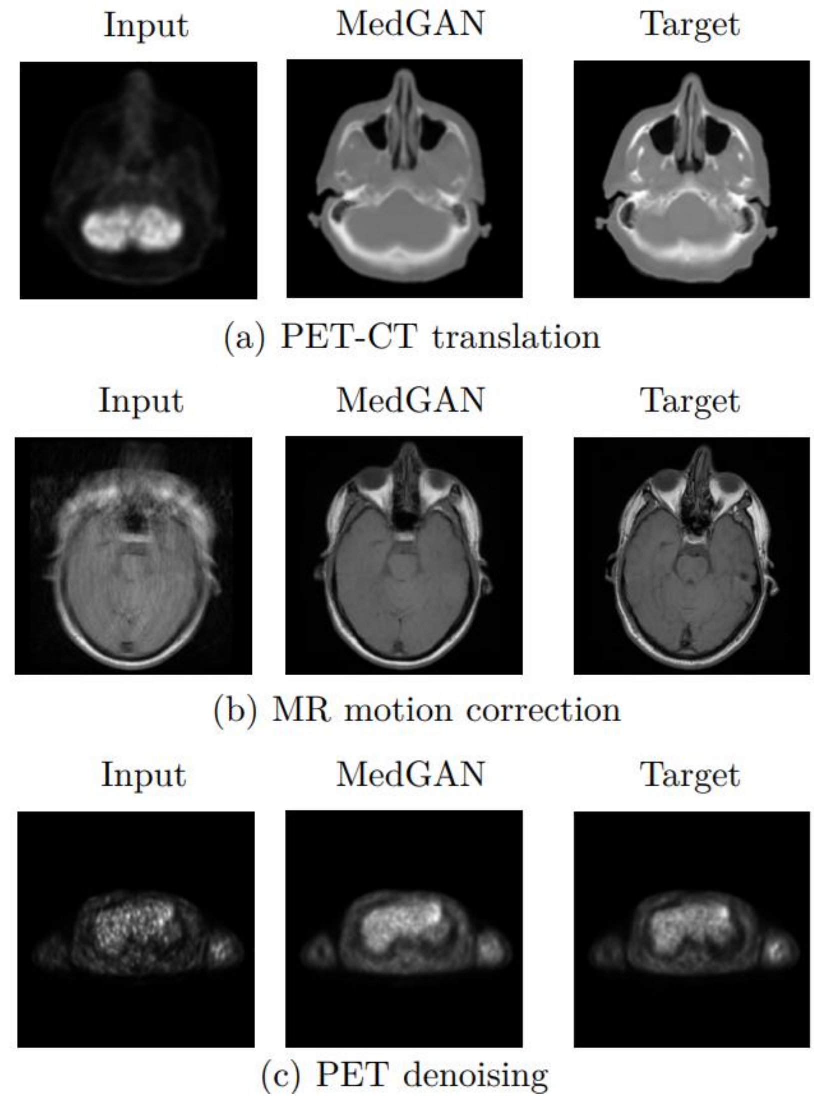
Source->Target domain transfer



CycleGAN. Zhu et al. 2017.

Many GAN applications

Pix2pix. Isola 2017. Many examples at https://phillipi.github.io/pix2pix/



(a) PET-CT translation

(b) MR motion correction

(c) PET denoising

MedGAN. Armanious et al. 2019

Many GAN applications

T. Karras, T. Aila, S. Laine, J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018. Video: https://www.youtube.com/watch?v=XOxxPcy5Gr4

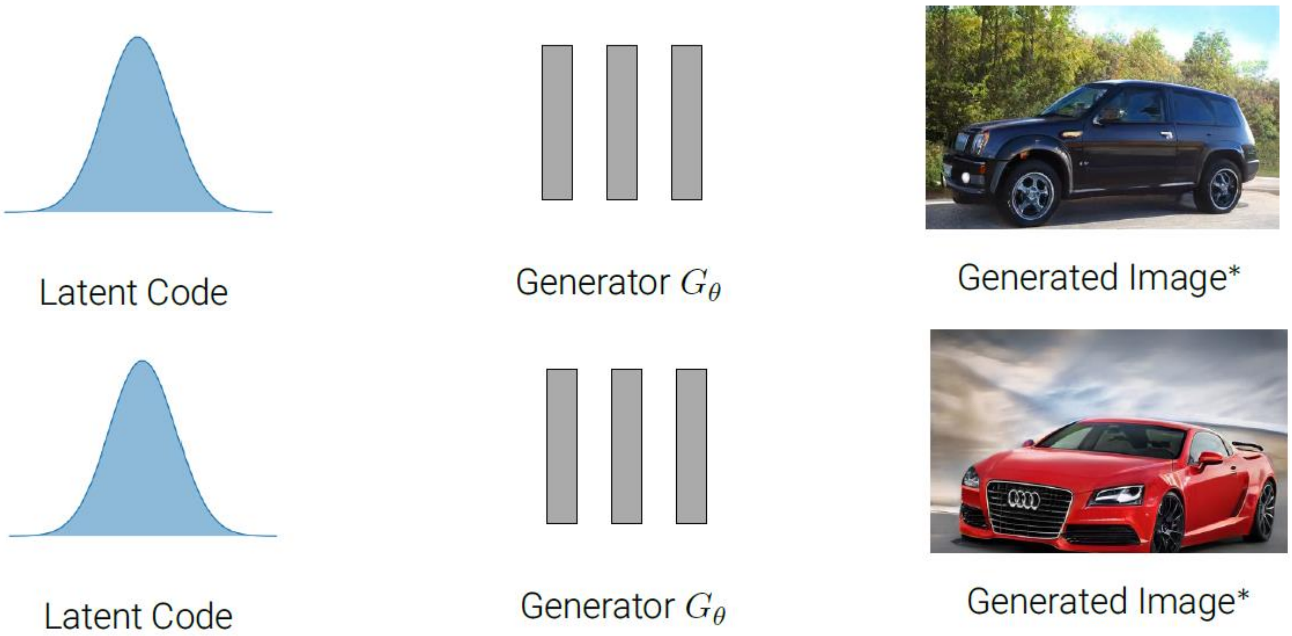
“The GAN Zoo”

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

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

Controllable GANs

Generative models are great! But..



Is the ability to sample photorealistic images all we want?

* StyleGAN2: Analyzing and Improving the Image Quality of StyleGAN, Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila, CVPR2020.

Controllable GANs

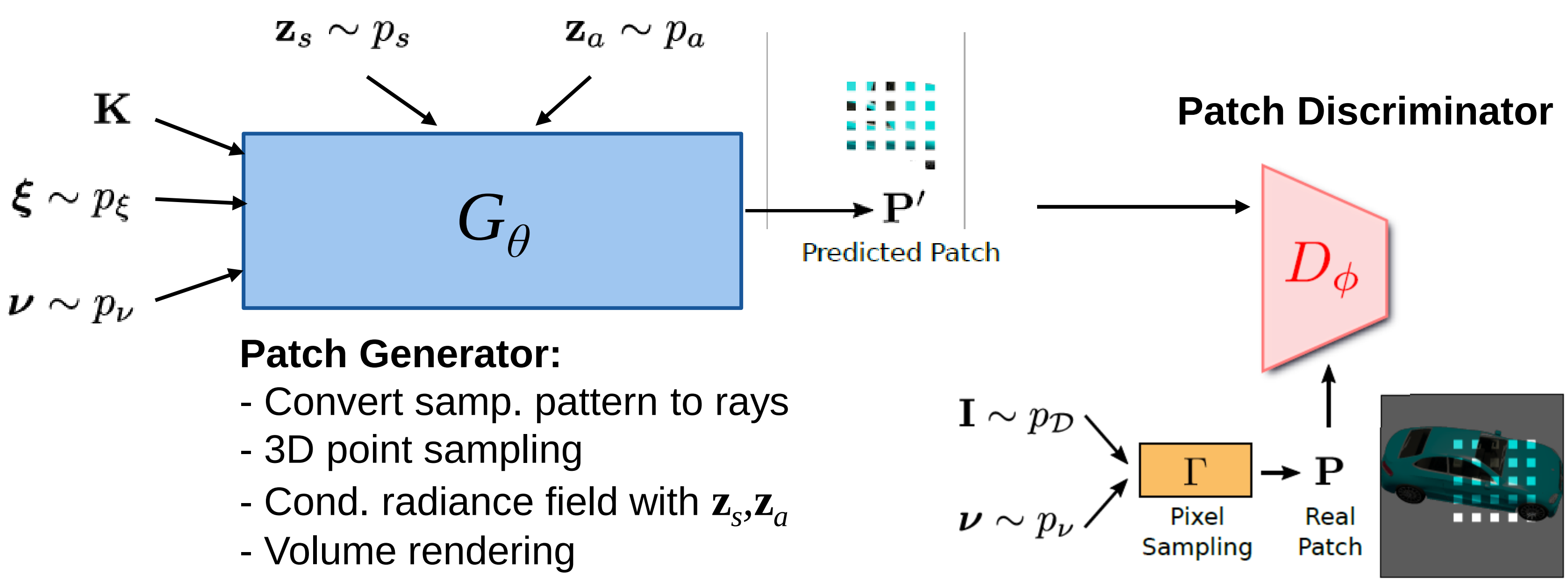
Goal: A generative model for 3D-aware image synthesis which allows us to

- Control individual objects wrt. their pose, size, and position in 3D
- Control camera viewpoint in 3D
- Train from collections of unposed images

Generative Radiance Fields*
(for 3D-Aware Image Synthesis)

Sample camera matrix $\mathbf{K}$, camera pose $\boldsymbol{\xi} \sim p_\xi$, and patch sampling pattern $\boldsymbol{\nu} \sim p_\nu$.

Sample latent shape and appearance codes $\mathbf{z}_s, \mathbf{z}_a$ and pass them to g_θ.



Patch Generator:
- Convert samp. pattern to rays
- 3D point sampling
- Cond. radiance field with $\mathbf{z}_s$,$\mathbf{z}_a$
- Volume rendering

* Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

Generative Radiance Fields*
(for 3D-Aware Image Synthesis)

Generator/discriminator for image patches of size 32x32 pixels.
Patches sampled at random scale using dilation.
Results on synthetic Carla dataset at 256x256 pixels:

Watch last 30 seconds of https://www.youtube.com/watch?v=akQf7WaCOHo

* Schwarz, Liao, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

GANs: Summary

Don’t work with an explicit density function.
Take game-theoretic approach: learn to generate from training distribution through 2-player game.

Pros:

- Beautiful, state-of-the-art samples!

Cons:

- Trickier / more unstable to train

Active areas of research:

- Better loss functions, more stable training
- GANs for all kinds of applications
- Controllable GANs
- Multi-object scene GANs