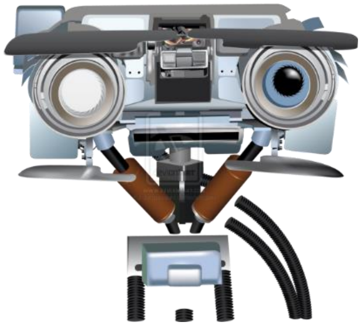
Announcements

Reminder: pset5 self-grading form and pset6 out, due Thursday 11/19 11:59pm Boston Time

- Class challenge out Thursday (will discuss in class)
- Lab this week: Q-Learning and GANs



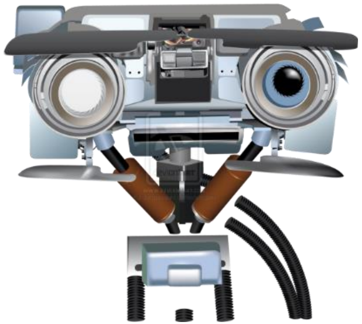
Generative Adversarial Networks (GANs)

Unsupervised Learning IV

Slides from lectures by Fei-Fei Li, Justin Johnson & Serena Yeung, Roger Grosse

Today

- Supervised vs Unsupervised Learning (recap)
- Density Models
- Generative Adversarial Networks (GANs)
- Cycle-GANs



Supervised vs Unsupervised Learning

Recap

Generative Adversarial Networks (GANs)

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.

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

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.

DOG, **DOG**, **CAT**

Object Detection

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.

GRASS, CAT, TREE, SKY

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.

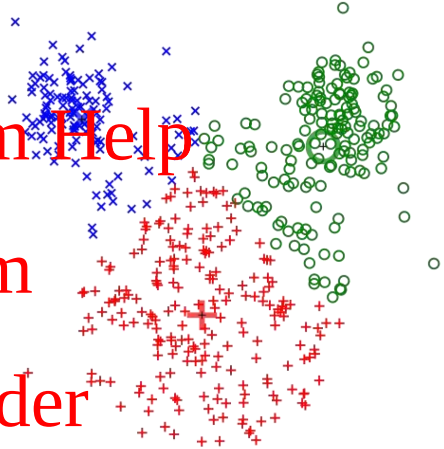
Supervised vs Unsupervised Learning

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

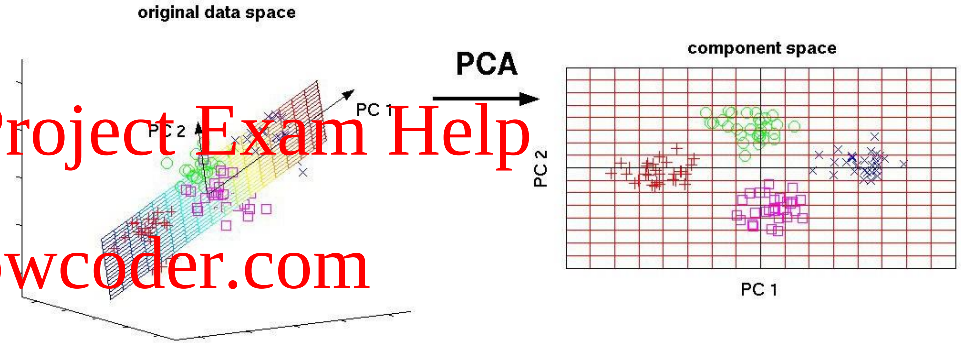
Supervised vs Unsupervised Learning

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.



3-d → 2-d

Principal Component Analysis
(Dimensionality reduction)

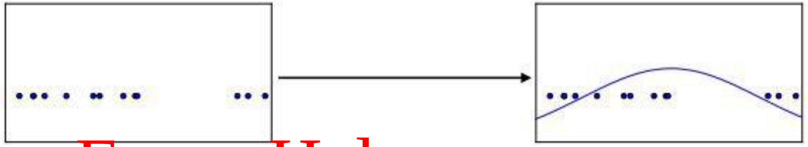
Supervised vs Unsupervised Learning

Unsupervised Learning

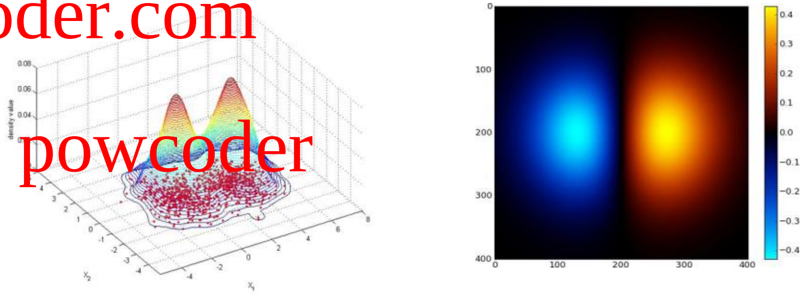
Data: x
Just data, no labels!

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

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



1-d density estimation



2-d density estimation

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.

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.

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.

Unsupervised Learning

Training data is cheap

Data: x
Just data, no labels!

Holy grail: Solve unsupervised learning => understand structure of the data

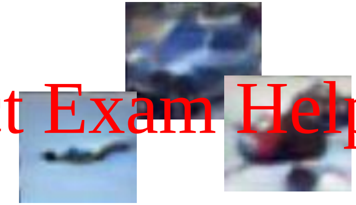
Goal: Learn some underlying hidden *structure* of the data

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

Generative Models

Given training data, generate new samples from same distribution

Training data ~ $p_{data}(x)$

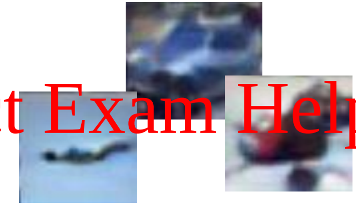


Generated samples ~ $p_{model}(x)$

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

Generative Models

Given training data, generate new samples from same distribution



Training data ~ $p_{data}(x)$

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

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

Addresses density estimation, a core problem in unsupervised learning

Several flavors:

- Explicit density estimation: explicitly define and solve for $p_{model}(x)$
- Implicit density estimation: learn model that can sample from $p_{model}(x)$ w/o explicitly defining it

Why Generative Models?

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



- Generative models of time-series data can be used for simulation and planning (reinforcement learning applications!)
- Training generative models can also enable inference of latent representations that can be useful as general features

Taxonomy of Generative Models

Generative models

Explicit density

Implicit density

Direct

GAN

Tractable density

Fully Visible Belief Nets

Approximate density

Markov Chain

GSN

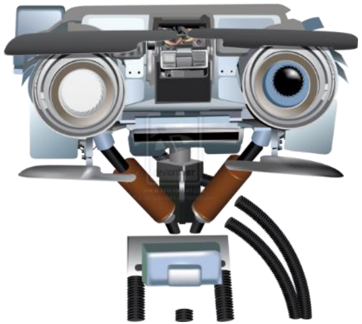
Variational

Variational Autoencoder

Markov Chain

Boltzmann Machine

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



Generative Adversarial Networks

Generative Adversarial Networks (GANs)

Overview

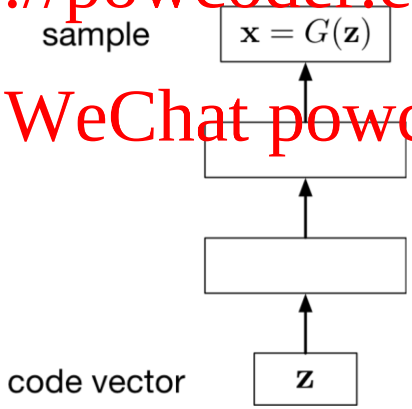
Four modern approaches to generative modeling:

- Generative adversarial networks (today)
- Reversible architectures
- Autoregressive models
- Variational autoencoders

All four approaches have different pros and cons.

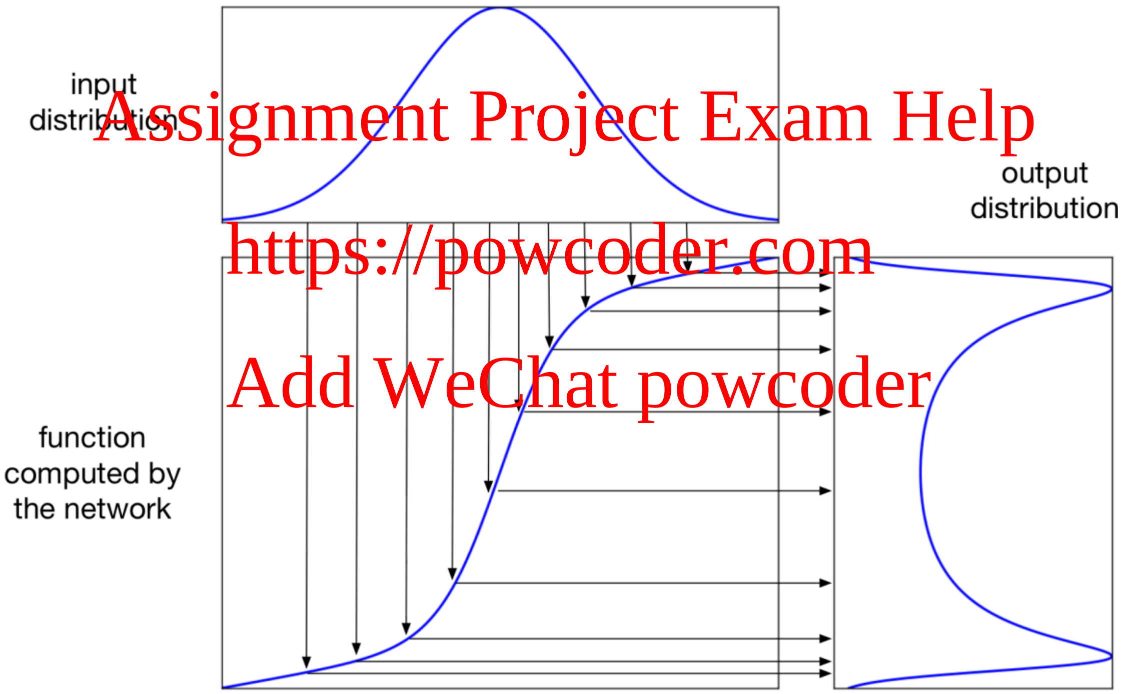
Implicit Generative Models

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

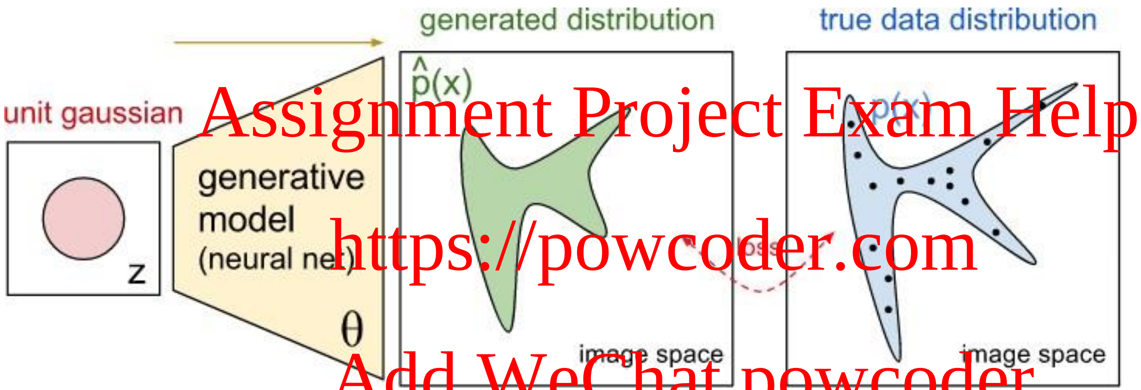


Implicit Generative Models

A 1-dimensional example:

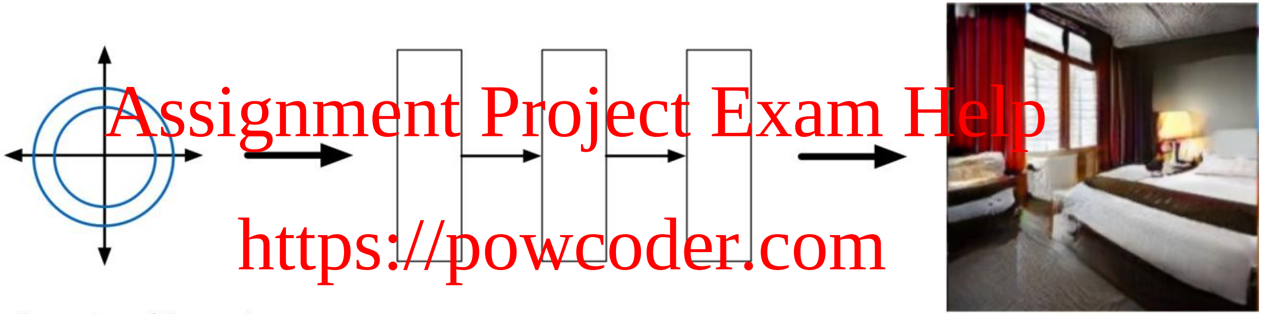


Implicit Generative Models



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

Implicit Generative Models



Each dimension of the code vector is sampled independently from a simple distribution, e.g. Gaussian or uniform.

This is fed to a (deterministic) generator network.

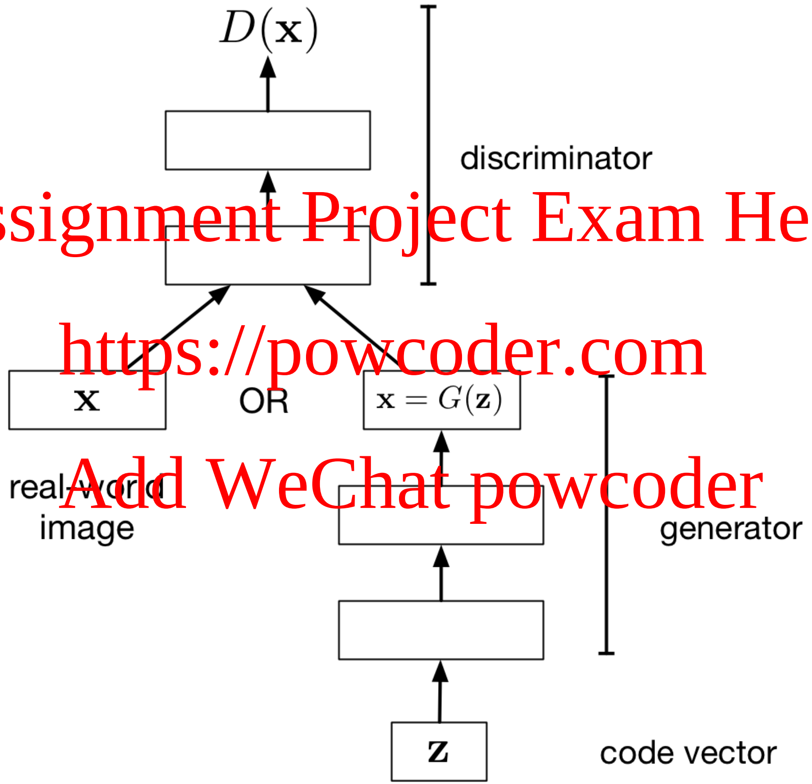
The network outputs an image.

This sort of architecture sounded preposterous to many of us, but amazingly, it works.

Generative Adversarial Networks

- 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 Networks



Generative Adversarial Networks

- Let D denote the discriminator's predicted probability of being data
- Discriminator's cost function: cross-entropy loss for task of classifying real vs. fake images

$$J_D = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[-\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}}[-\log(1 - D(G(\mathbf{z})))]$$

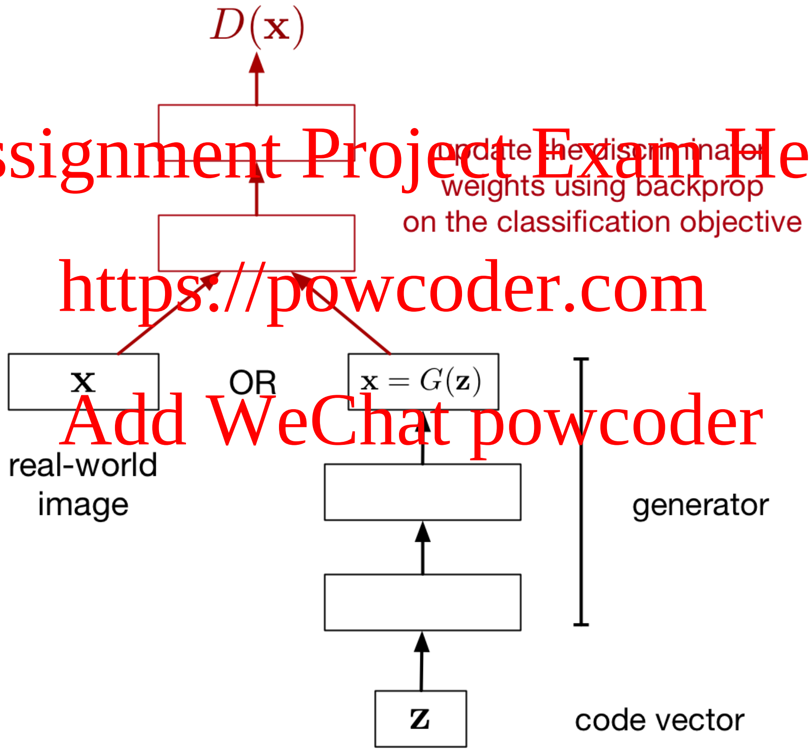
- One possible cost function for the generator: the opposite of the discriminator's

$$\begin{aligned} J_G &= -J_D \\ &= \text{const} + \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{z})))] \end{aligned}$$

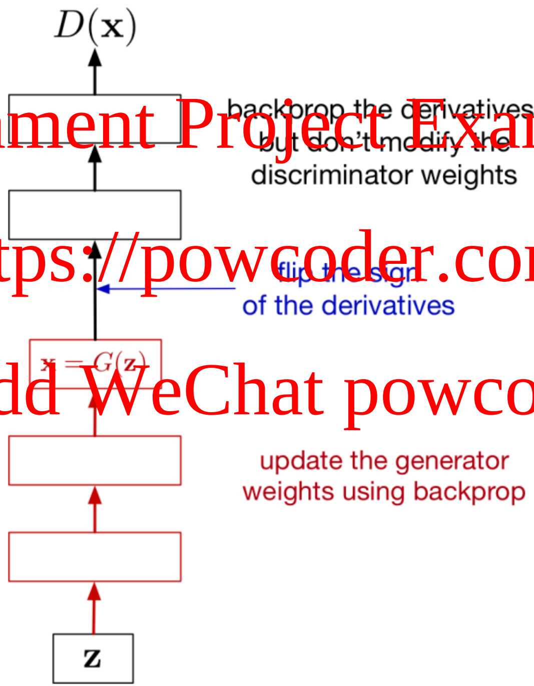
- This is called the minimax formulation, since the generator and discriminator are playing a zero-sum game against each other:

$$\max_G \min_D J_D$$

Updating the discriminator:



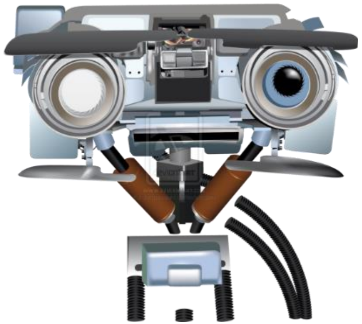
Updating the generator:



Generative Adversarial Networks

- Since GANs were introduced in 2014, there have been hundreds of papers introducing various architectures and training methods.

- Most modern architectures are based on the Deep Convolutional GAN (DC-GAN), where the generator and discriminator are both conv nets.
- GAN Zoo: https://github.com/hindupuravinash/the-gan-zoo
  - Good source of horrible puns (VEEGAN, Checkhov GAN, etc.)



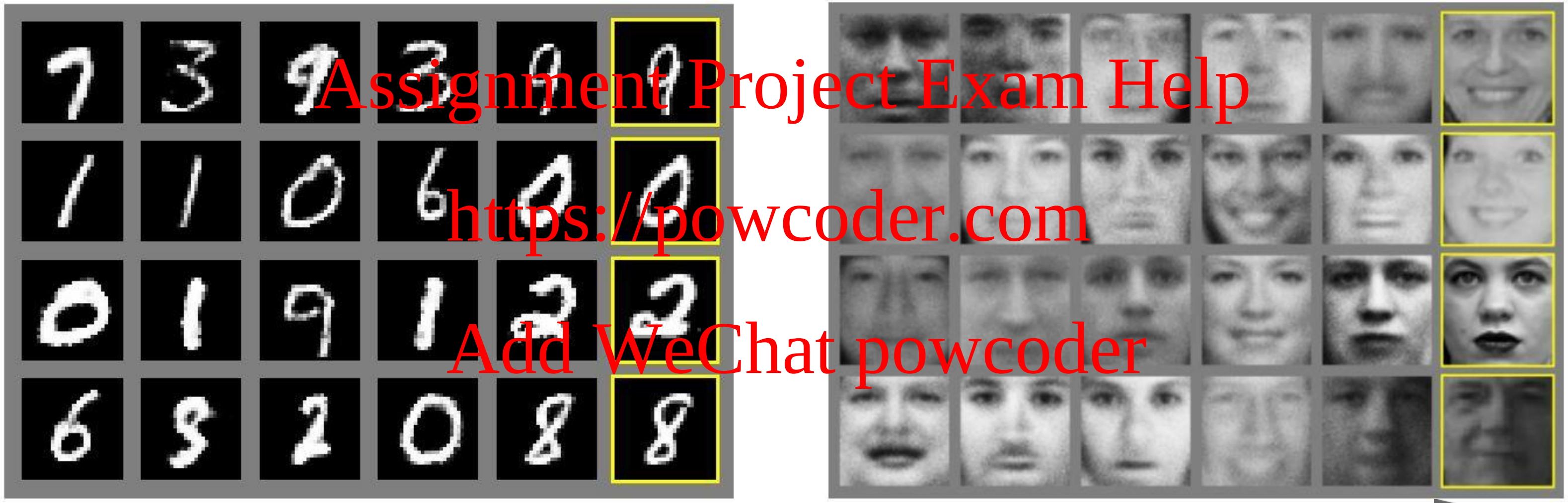
GANs: Application to Image Generation

Generative Adversarial Networks (GANs)

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

Generative Adversarial Nets

Generated samples

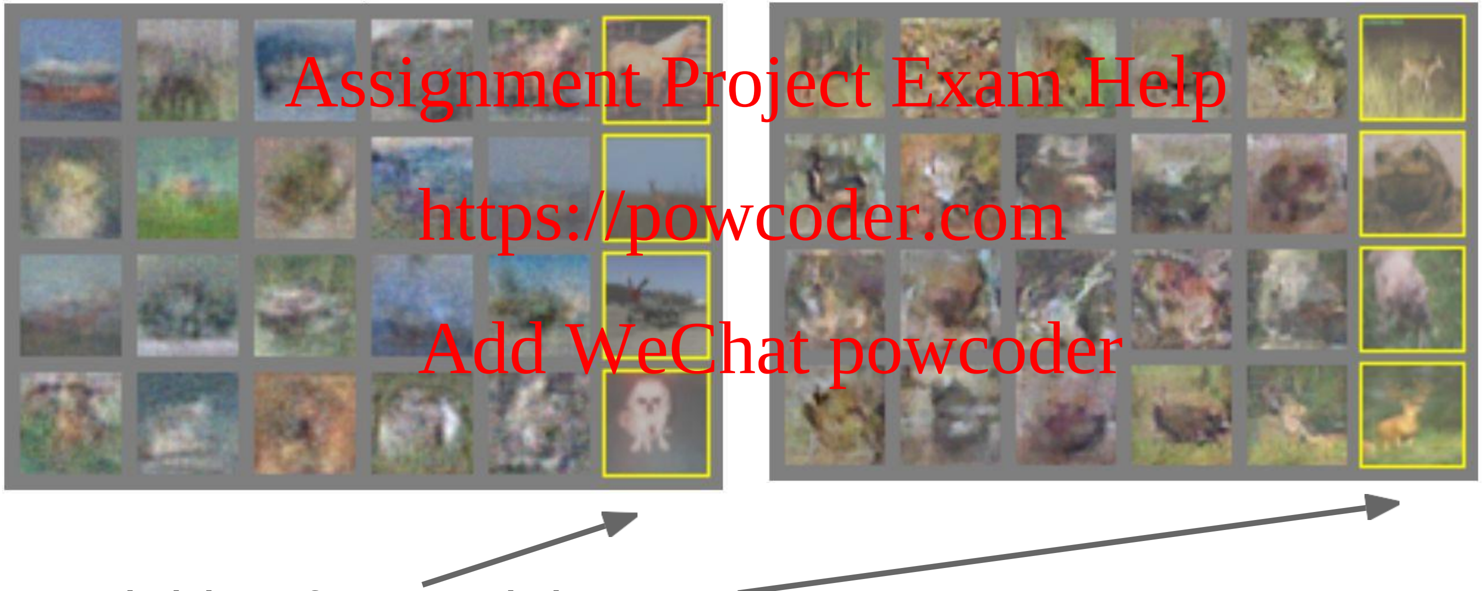


Nearest neighbor from training set

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

Generative Adversarial Nets

Generated samples (CIFAR-10)



Nearest neighbor from training set

GAN Samples

Celebrities:

Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation

GAN Samples

Bedrooms:

Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation

GAN Samples

Objects:

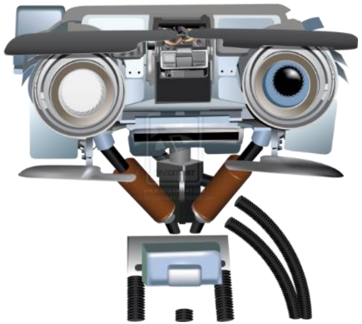
Karras et al., 2017. Progressive growing of GANs for improved quality, stability, and variation

GAN Samples

GANs revolutionized generative modeling by producing crisp, high-resolution images.

The catch: we don't know how well they're modeling the distribution.

- Can't measure the log-likelihood they assign to held-out data. Could they be memorizing training examples? (E.g., maybe they sometimes produce photos of real celebrities?)
- We have no way to tell if they are dropping important modes from the distribution.
- See Wu et al., "On the quantitative analysis of decoder-based generative models" for partial answers to these questions.



Cycle GANs

Generative Adversarial Networks (GANs)

CycleGAN

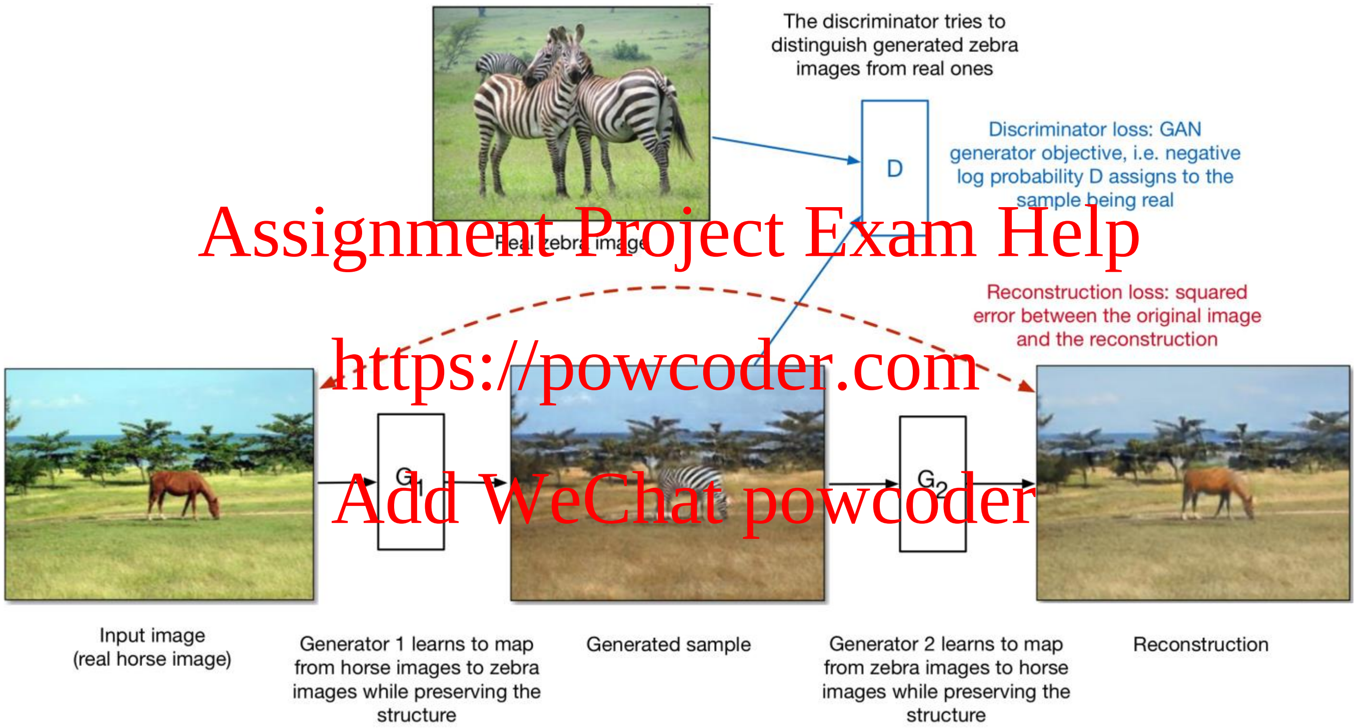
Style transfer problem: change the style of an image while preserving the content.

Data: Two unrelated collections of images, one for each style

CycleGAN

- If we had paired data (same content in both styles), this would be a supervised learning problem. But this is hard to find.
- The CycleGAN architecture learns to do it from unpaired data.
  - Train two different generator nets to go from style 1 to style 2, and vice versa.
  - Make sure the generated samples of style 2 are indistinguishable from real images by a discriminator net.
  - Make sure the generators are cycle-consistent: mapping from style 1 to style 2 and back again should give you almost the original image.

CycleGAN



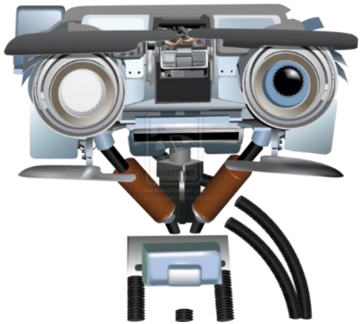
Total loss = discriminator loss + reconstruction loss

CycleGAN

Style transfer between aerial photos and maps:

CycleGAN

Style transfer between road scenes and semantic segmentations (labels of every pixel in an image by object category):



GANs: Convolutional Architectures

Generative Adversarial Networks (GANs)

Generative Adversarial Nets: Convolutional Architectures

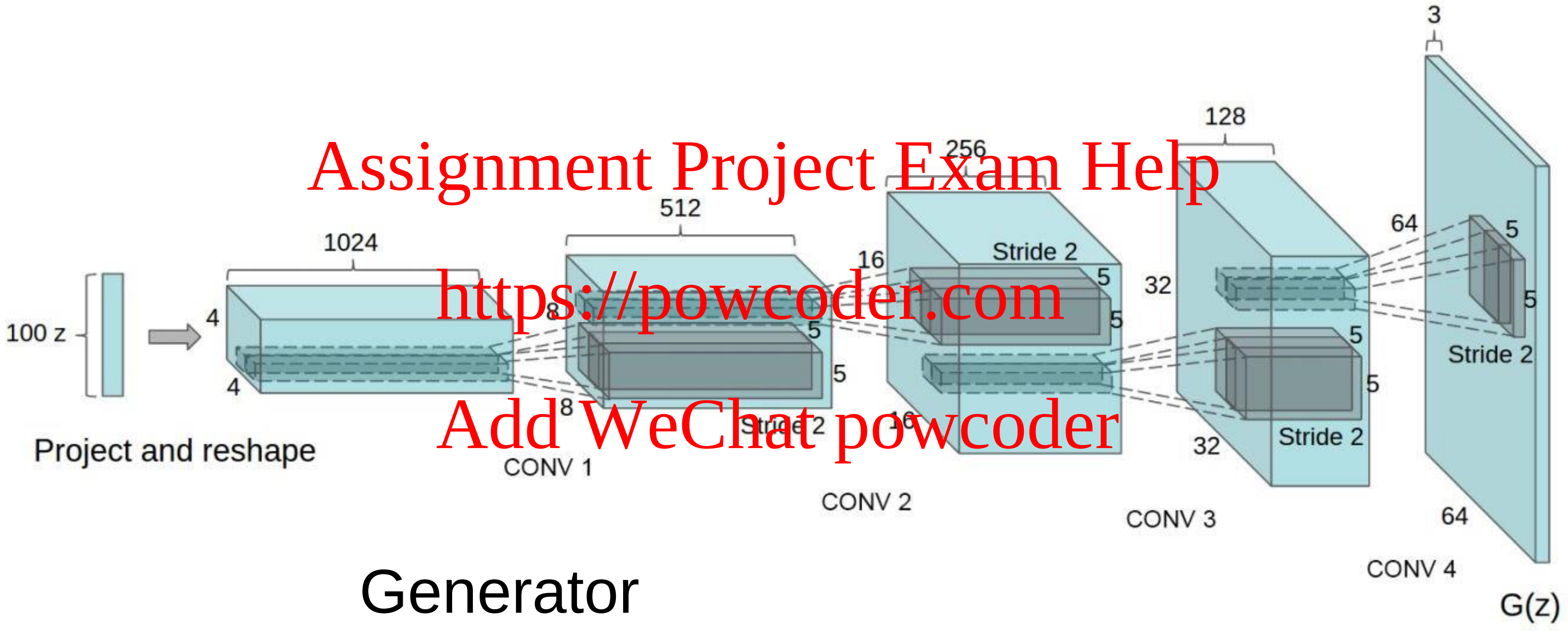
Generator is an upsampling network with fractionally-strided convolutions
Discriminator is a convolutional network

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

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

Generative Adversarial Nets: Convolutional Architectures



Generator

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

Generative Adversarial Nets: Convolutional Architectures

Samples from the model look amazing!

Radford et al, ICLR 2016

Generative Adversarial Nets: Convolutional Architectures

Interpolating between random points in laten space

Radford et al, ICLR 2016

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman Neutral woman Neutral man

Samples from the model

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman | Neutral woman | Neutral man

Samples from the model

Average Z vectors, do arithmetic

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman Neutral woman Neutral man

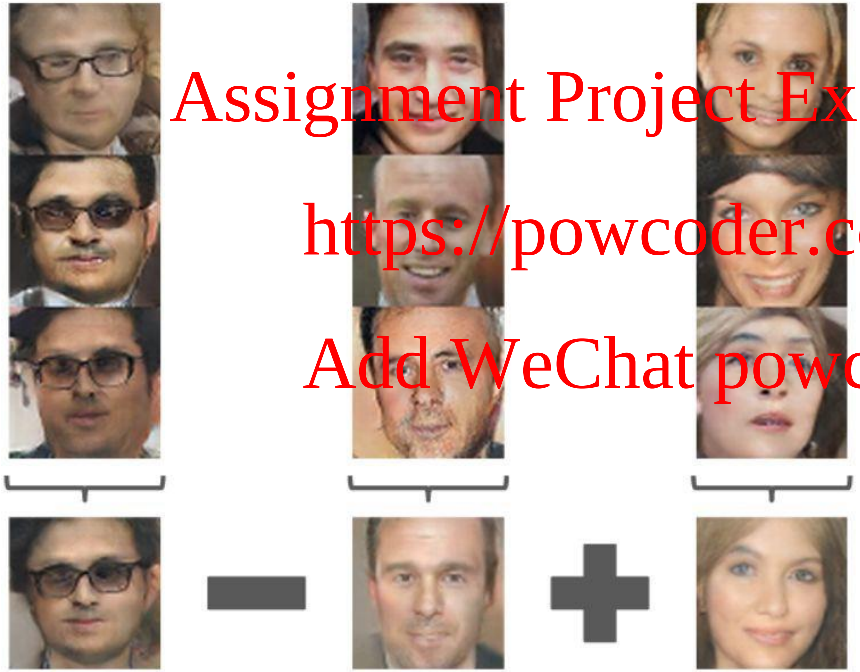
Smiling Man

Samples from the model

Average Z vectors, do arithmetic

Generative Adversarial Nets: Interpretable Vector Math

Glasses man　　No glasses man　　No glasses woman



Radford et al, ICLR 2016

Generative Adversarial Nets: Interpretable Vector Math

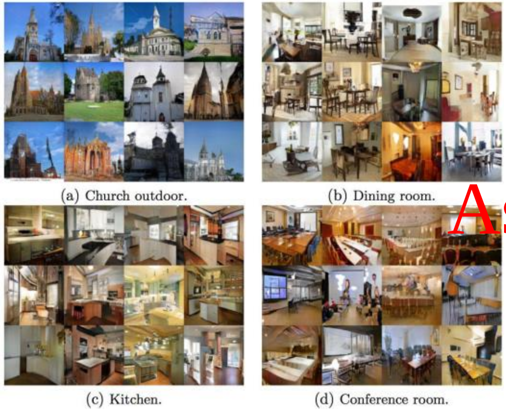
Glasses man | No glasses man | No glasses woman

Radford et al, ICLR 2016

Woman with glasses

2017: Year of the GAN

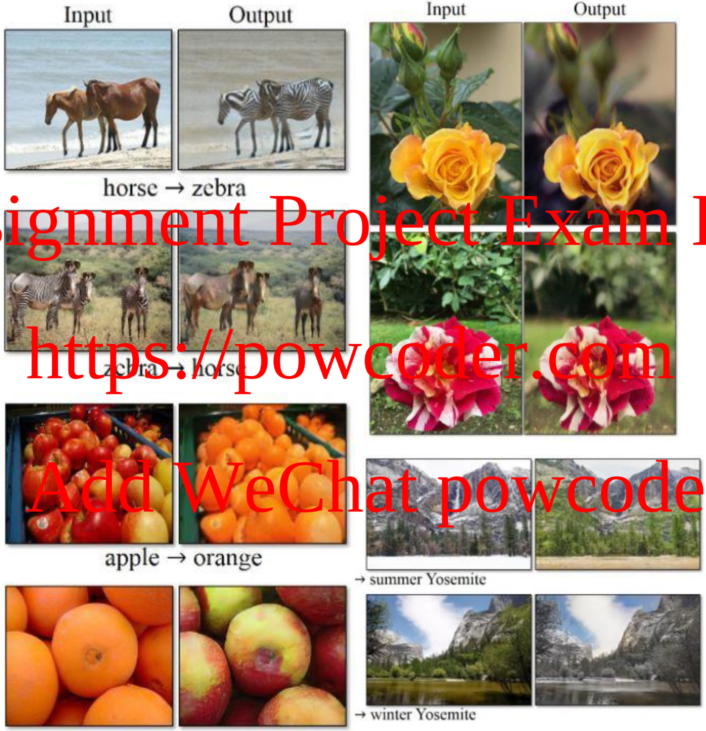
Better training and generation



LSGAN. Mao et al. 2017.

BEGAN. Bertholet et al. 2017.

Source->Target domain transfer



CycleGAN. Zhu et al. 2017.

Text -> Image Synthesis

Reed et al. 2017.

Many GAN applications

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

Next Class

Unsupervised Learning V: Semi-supervised Learning

Semi-supervised learning (SSL); self-training; co-training; clustering methods, SSSVM