Generative Model

Autoencoder (AE) and Generative adversarial network (GAN)

Weijian Deng

Overview

- Thanks for giving me the opportunity to talk about the generative model

- This talk covers
  - Supervised learning vs. unsupervised learning
  - Generative model
  - Autoencoder (sparse AE); generative adversarial network (GAN)

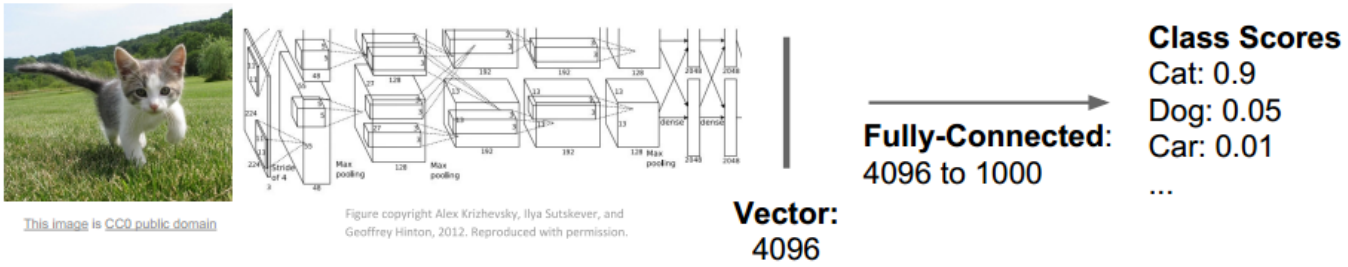
Supervised learning vs. Unsupervised learning

Supervised learning

Data: (x, y)
X is sample, y is label

Goal: learn a function to map x □y

Image classification



cat

A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS 2012

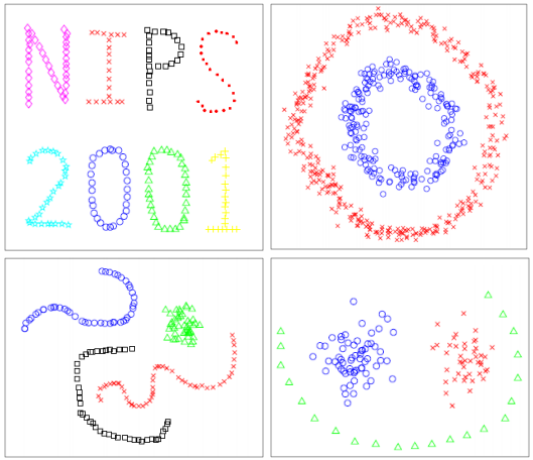
Supervised learning vs. Unsupervised learning

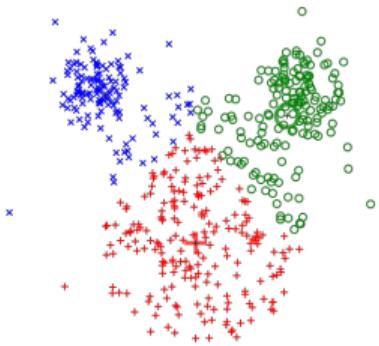
Unsupervised learning

Data: x
X is sample, no label

Goal: learn underlying hidden structure of data

Clustering





A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In NIPS 2002.

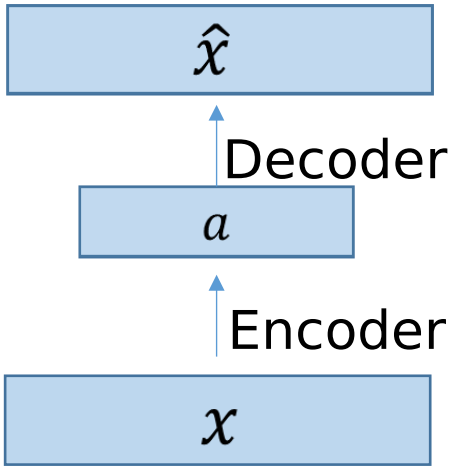
Supervised learning vs. Unsupervised learning

Unsupervised learning

Data: x
X is sample, no label

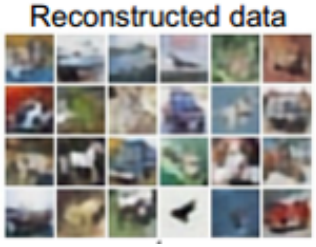
Goal: learn underlying hidden structure of data

Autoencoder



Reconstructed Input data

Input data



loss function: $||x - \hat{x}||^2$

Supervised learning vs. Unsupervised learning

Supervised learning

Data: (x, y)
X is sample, y is label

Goal: learn a function to map x □y

Example: image classification

Unsupervised learning

Data: x
X is sample, **no label**

Goal: learn underlying hidden structure of data

Example: clustering, Autoencoder

Supervised learning vs. Unsupervised learning

Supervised learning

Data: (x, y)
X is sample, y is label

Goal: learn a function to map x □y

Example: image classification

Unsupervised learning

understand structure of visual world

Data: x
X is sample, **no label**

Goal: learn underlying hidden structure of data

Example: clustering, Autoencoder

Generative model

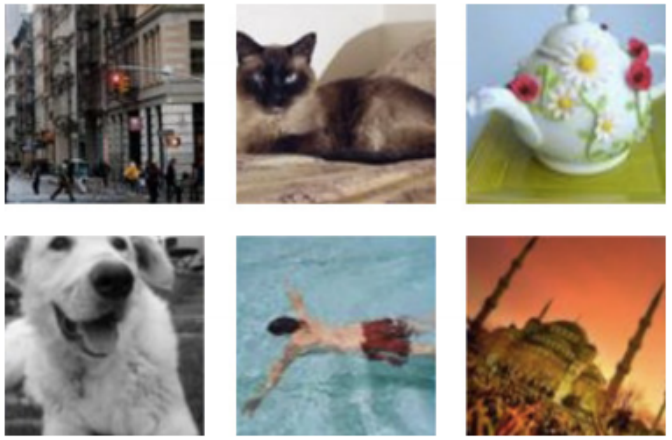
What I cannot create, I do not understand
—Richard Feynman

Generative model

Real images distribution

$P_{data}(x)$

Real images distribution



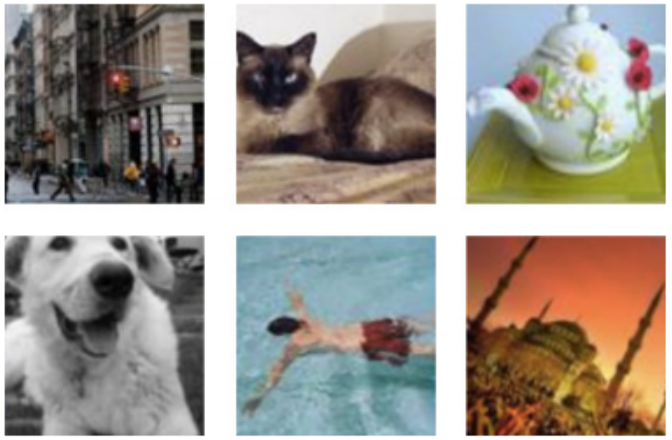
generated images distribution

$P_G(x)$

generated images distribution

We want to learn $P_G(x)$ similar to $P_{data}(x)$

Generative model



Real images distribution

generated images distribution

Real images distribution
$P_{data}(x)$

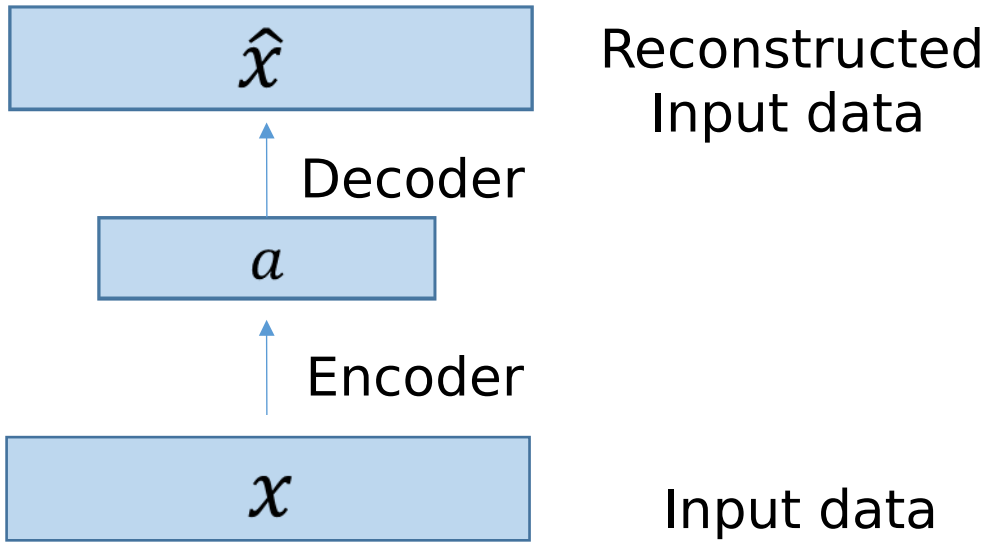
generated images distribution
$P_G(x)$

How to learn $P_G(x)$?

Autoencoder

Autoencoder

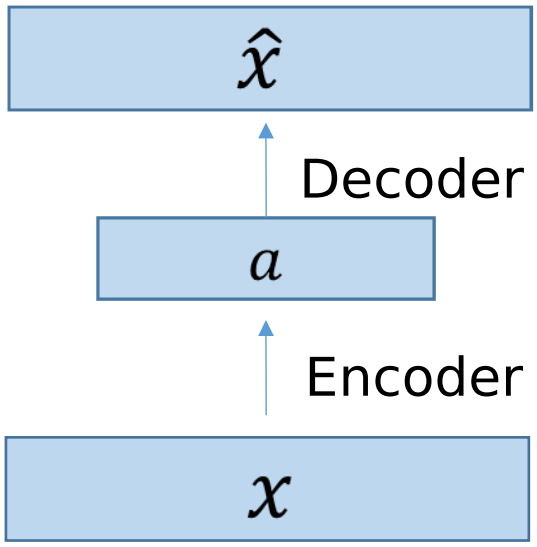
Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



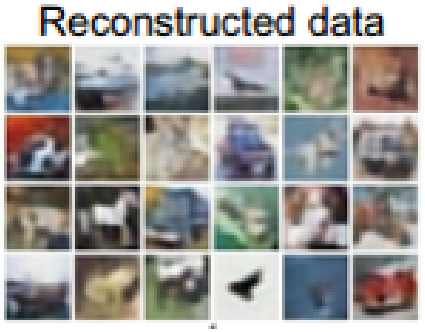
Autoencoder

Autoencoder

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



Reconstructed
Input data



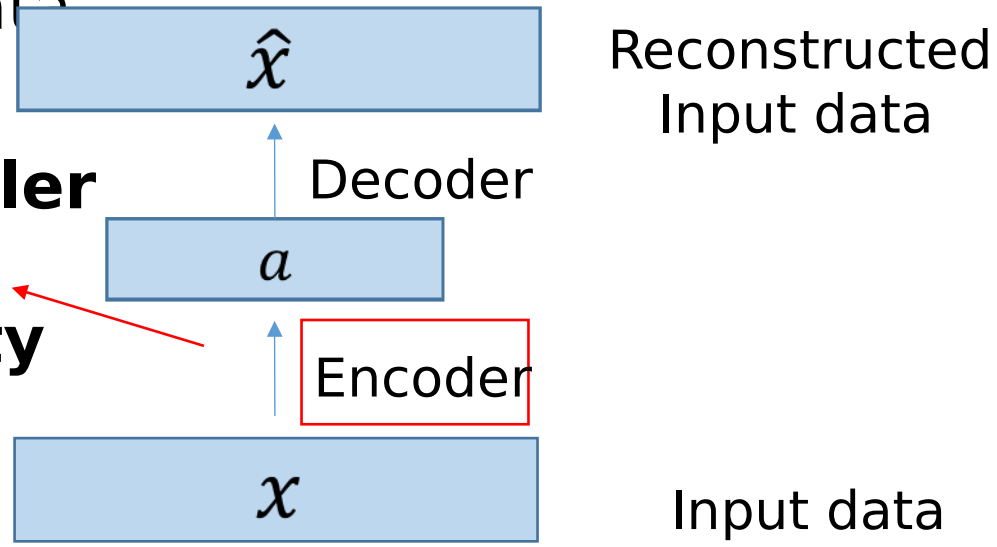
Input data

Autoencoder

Autoencoder

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

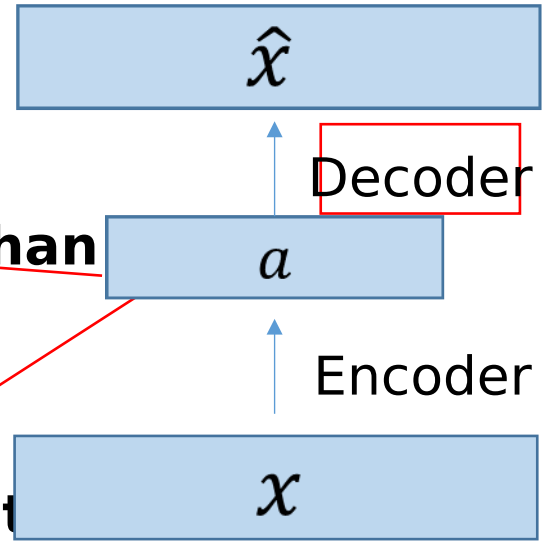
a usually smaller than x (dimensionality reduction)



Autoencoder

Autoencoder

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



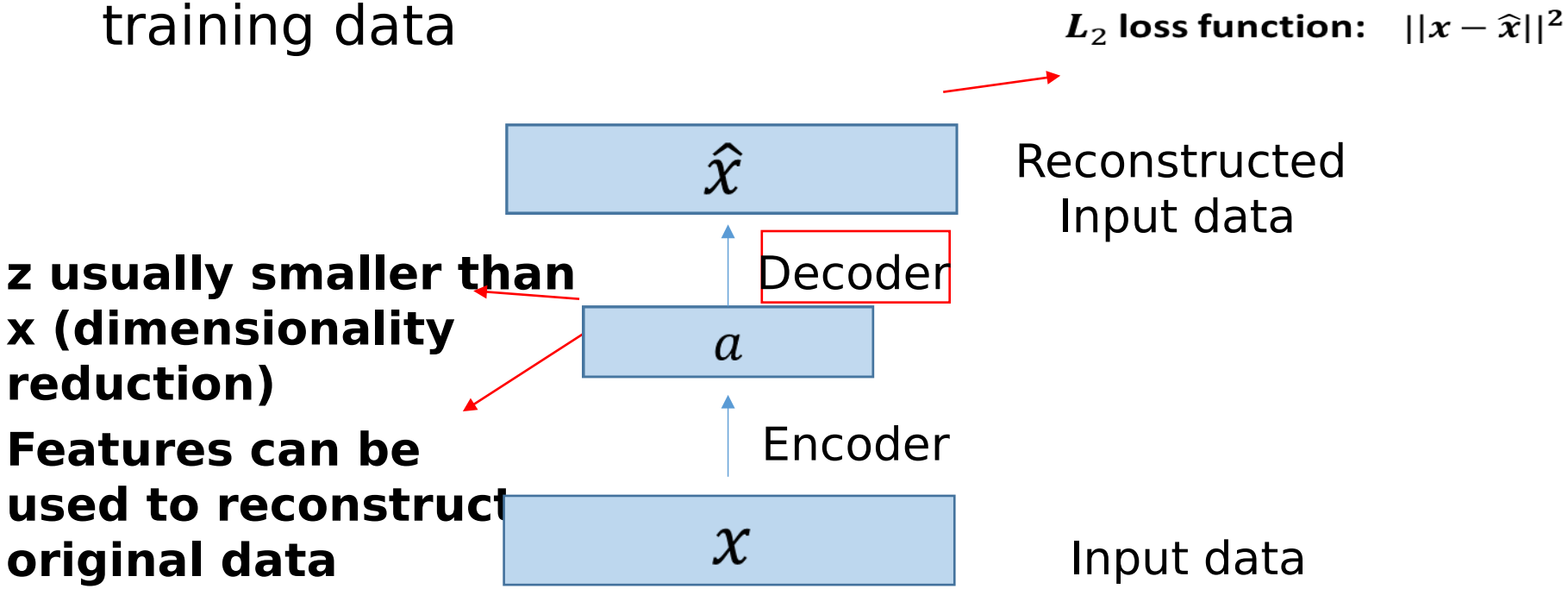
z usually smaller than x (dimensionality reduction)

Features can be used to reconstruct original data

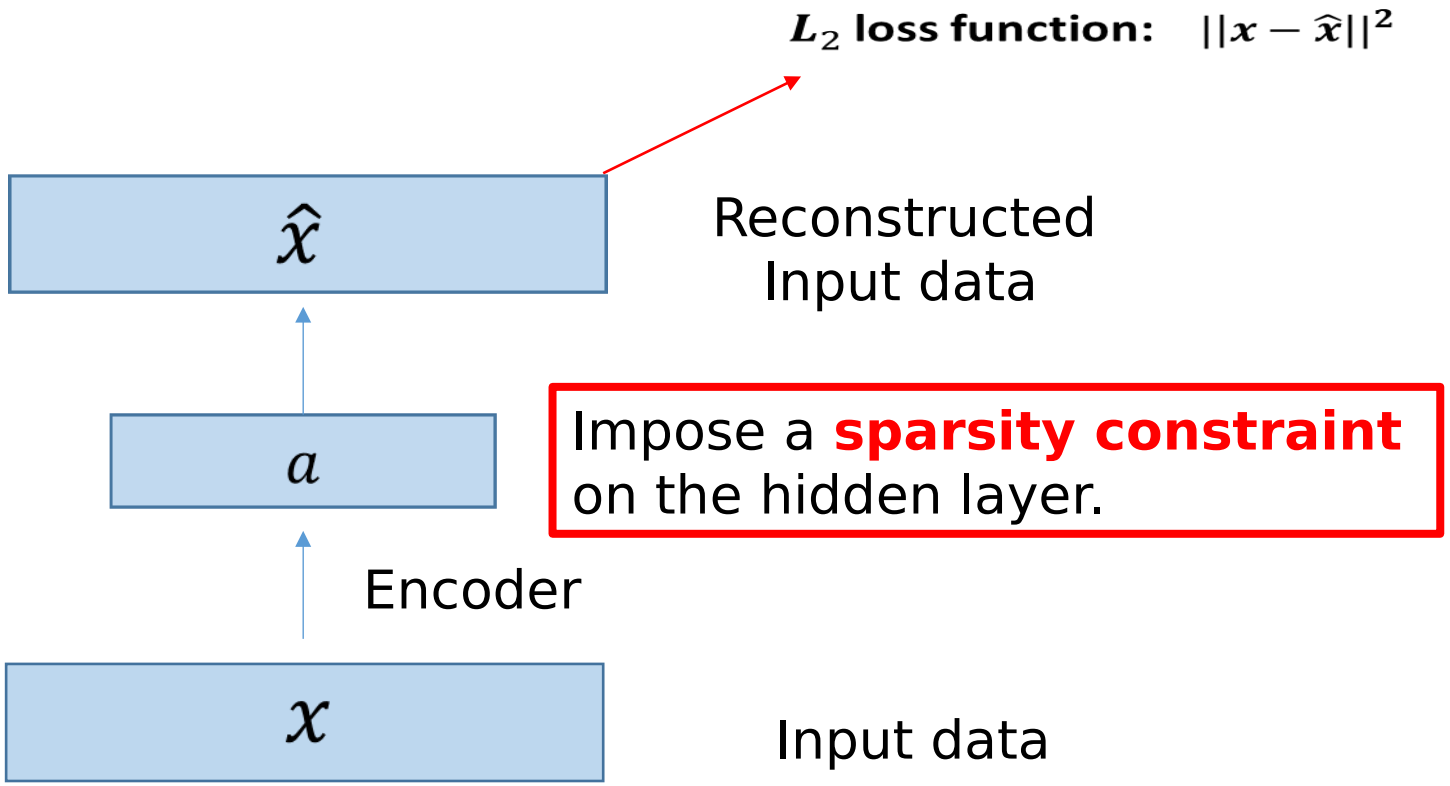
Autoencoder

Autoencoder

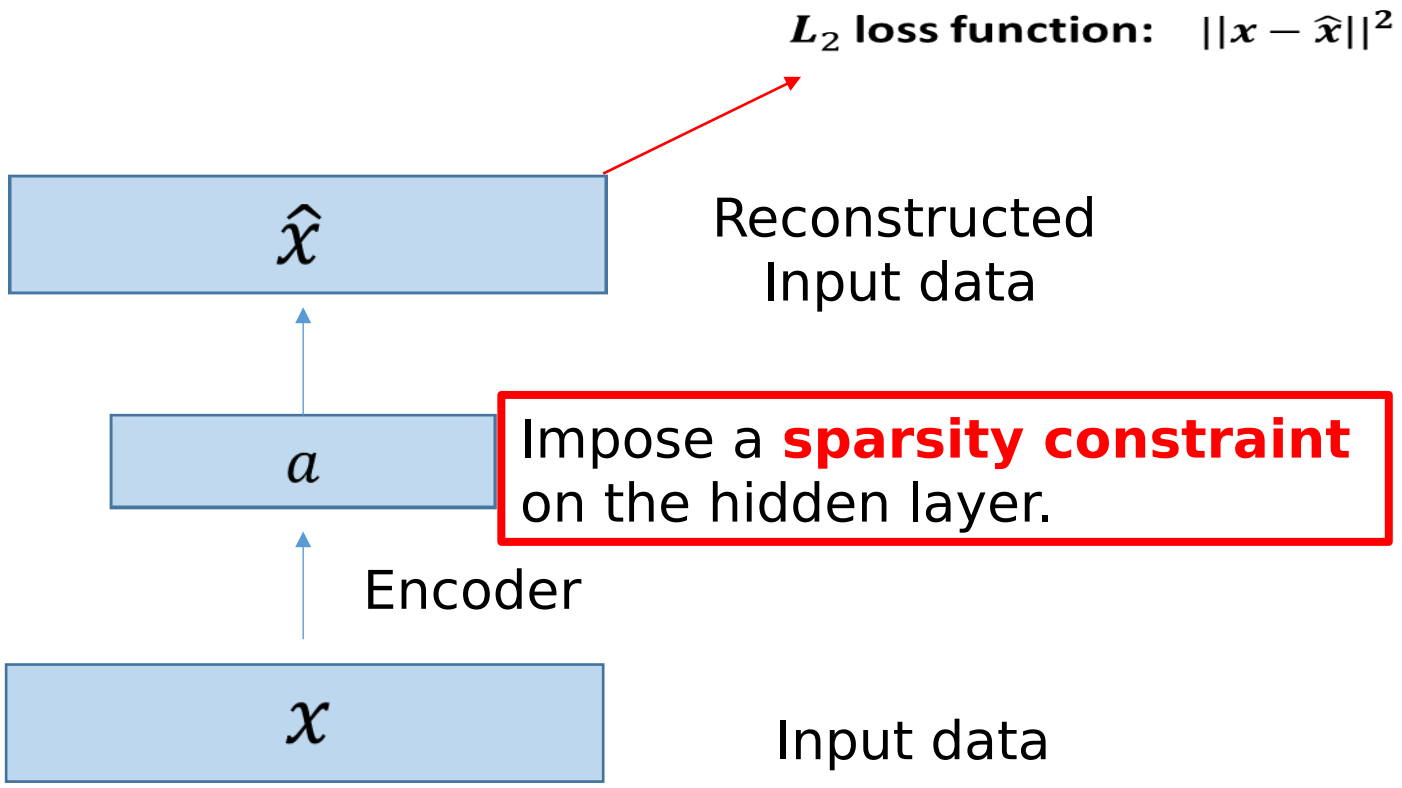
Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



Sparse Autoencoder

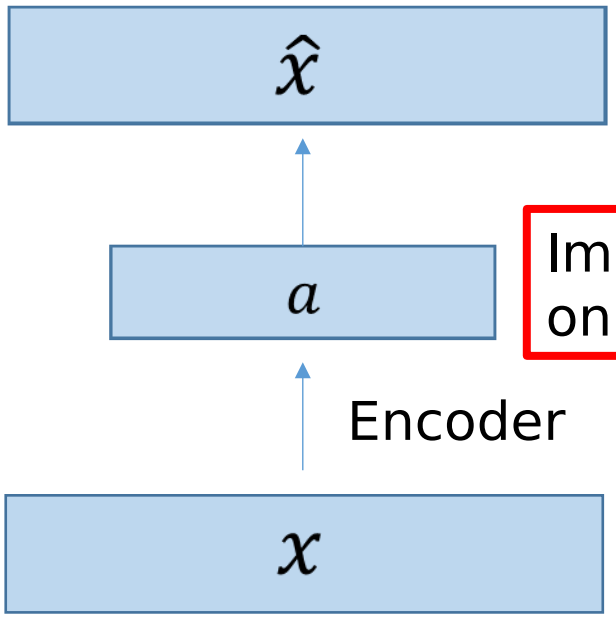


Sparse Autoencoder



We would like the **average activation** of each hidden neuron to be a small value (0.05).

Sparse Autoencoder



Reconstructed Input data

Impose a **sparsity constraint** on the hidden layer.

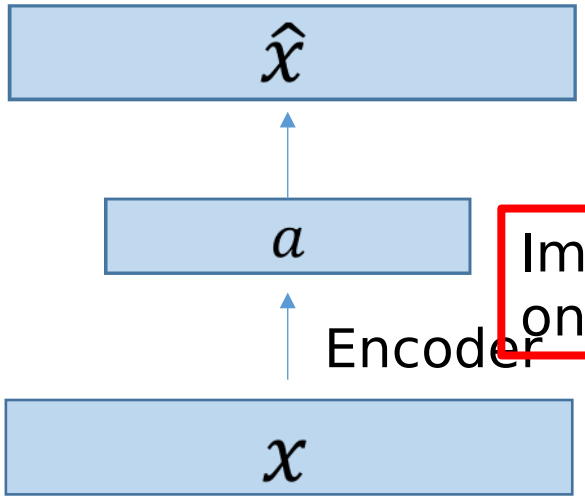
Encoder

Input data

$\hat{p}_j = \frac{1}{m}\sum_{i=1}^{m} a_j(f(x^{(i)}))$ a_j denotes the activation of hidden unit j; we have **m** sample

$$a_j(f(x^{(i)})) = Sigmoid(W_j{}^T x^{(i)} + b)$$

Sparse Autoencoder



$\hat{p}_j = \frac{1}{m}\sum_{i=1}^{m} a_j(f(x^{(i)}))$ a_j denotes the activation of hidden unit j; we have **m** sample

$$\sum_{j=1}^{s} KL(p||\hat{p}_j)$$ ***s*** is the number of neurons in the hidden layer; **P** = 0.05

$KL(p||\hat{p}_j) = \text{p}\log\frac{p}{\hat{p}_j} + (1-\text{p})\log\frac{1-p}{1-\hat{p}_j}$ is the Kullback-Leibler (KL) divergence.

(KL-divergence is a standard function for measuring how different two distributions are)

Generative adversarial network (GAN)

GAN is the most important upcoming breakthrough in deep learning.
—Yan LeCun

Generative adversarial network (GAN)

Discriminative Model
D

1

0

Real image
from real data

Fake image
from ***G***

Generative
Model G

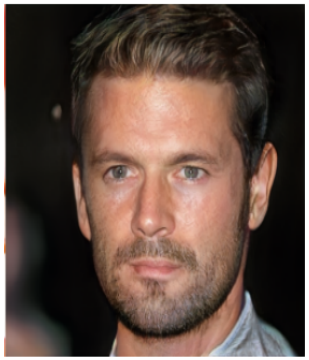
p(z)

random noise *z*

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

Generative adversarial network (GAN)

Which of the followings are generated/ fake images ?



A

B

C

D

Karras et al, 2017

Generative adversarial network (GAN)

Which of the followings are generated/ fake images ?

A

B

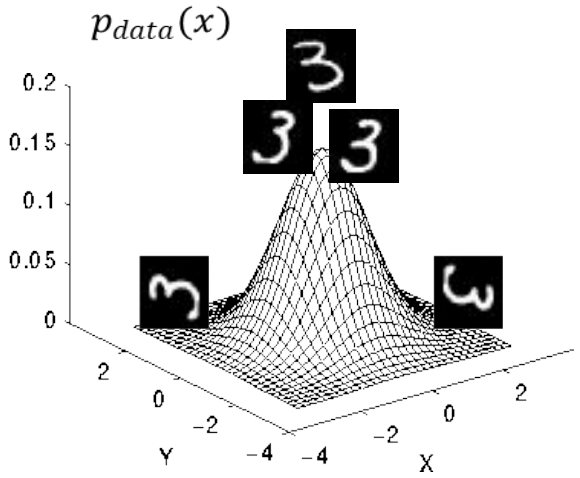
C

D

Andrew etal. Large scale gan training for high fidelity natural image synthesis. Arxiv 2018

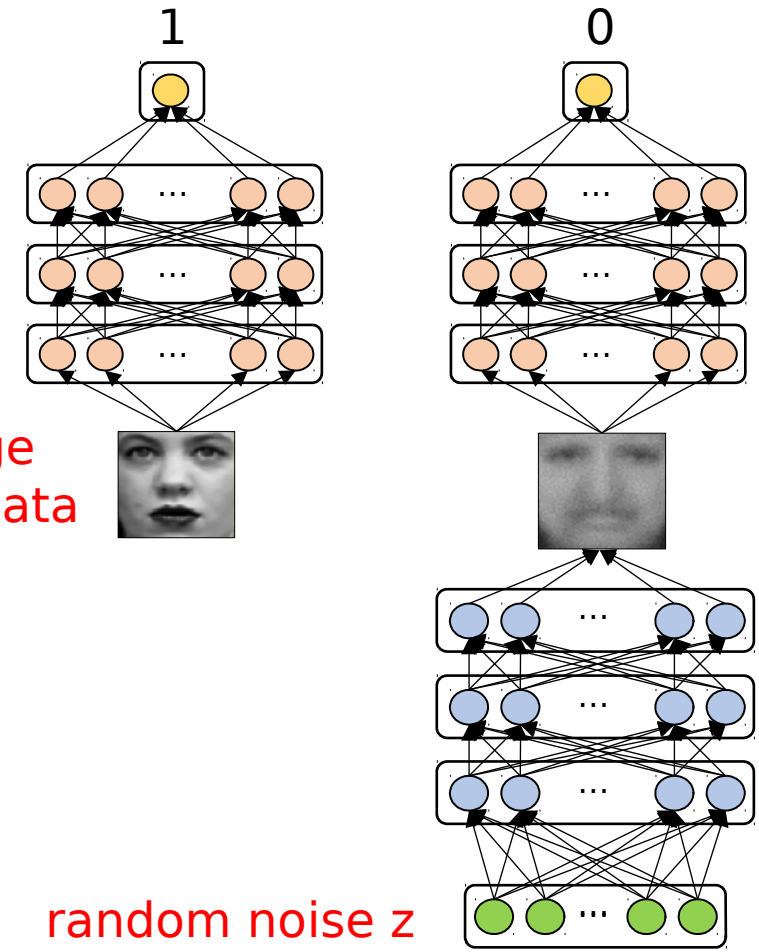
Generative?

- Data:  …
- Discriminative model: $p_D(x;\ \theta_D)$
- Generative model: $p_G(x;\theta_G)$
  - True data distribution: $p_{data}(x)$
  - Train $p_G(x) \approx p_{data}(x)$



Generative adversarial network (GAN)

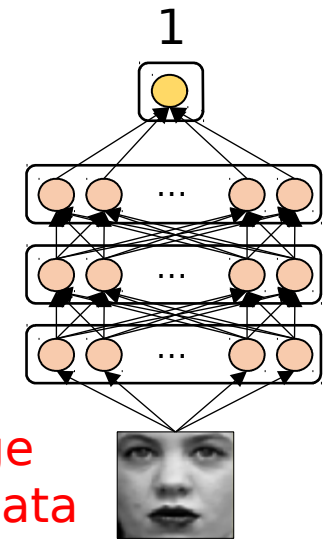
Discriminative Model
D



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

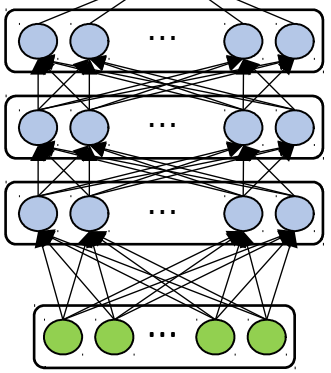
Generative adversarial network (GAN)

Discriminative Model
D



- **Try to classify the image x**
  - D(x)=1 when x from real data
  - D(x)=0 when x from G (generator)

Fake image
from ***G***

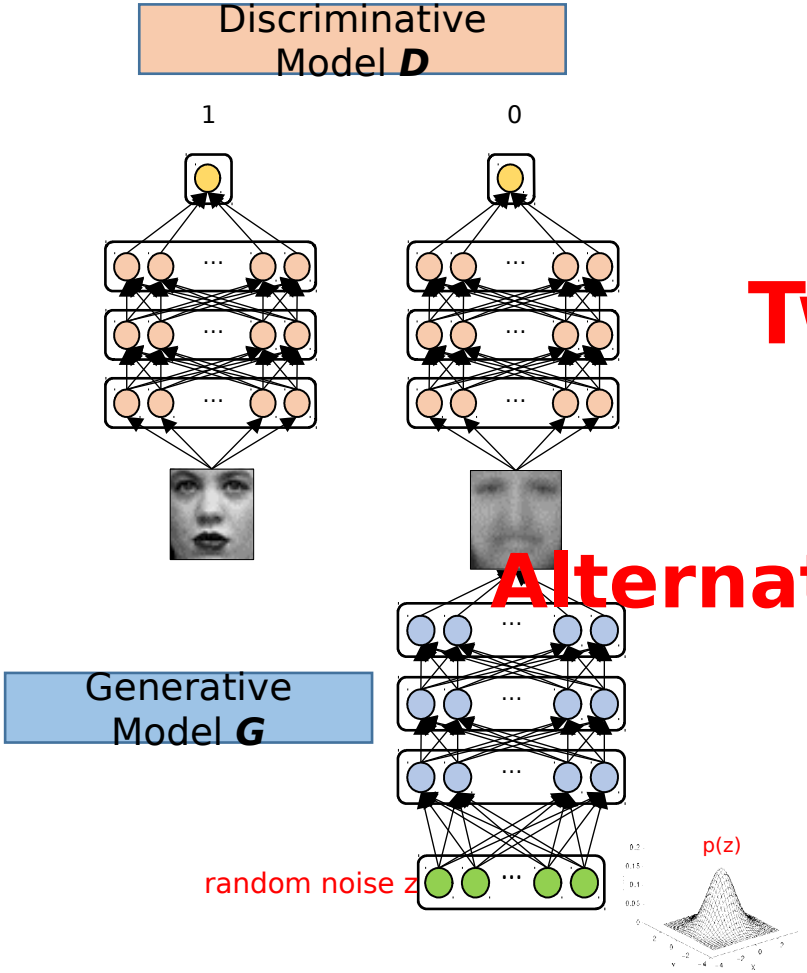


- **Try to fool the discriminator by generating real-looking images**

Generative adversarial network (GAN)

$$\min_G \max_D V(D,G)$$

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



Two-player game

Alternately optimize G and D

Generative adversarial network (GAN)

$$\min_G \max_D V(D,G)$$

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

Step 1: Optimize D

- Fixed G, maximize V(D):

$$\max_D V_G(D) = \max_D \left[\mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_{z(z)}}[\log(1 - D(G(z)))]\right]$$

- From sample $x^{(i)}$, $z^{(i)}$

$$\max_D \left[\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]\right]$$

- Binary Classification (logistic loss):
  - Sample from data: label=1
  - Sample from generator: label = 0

Generative adversarial network (GAN)

$$\min_G \max_D V(D,G)$$

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

Step 2: Optimize G

- Fixed D, minimize $V_D(G)$:

$$\min_G V_D(G) = \min\left[\mathbb{E}_{z\sim p_{z(z)}}[\log(1-D(G(z)))]\right]$$

Try to make

Try to make $D(G(z)) = 1$

Generative adversarial network (GAN)

$$\min_G \max_D V(D,G)$$

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

Step 2: Optimize G

- Fixed D, minimize $V_D(G)$:

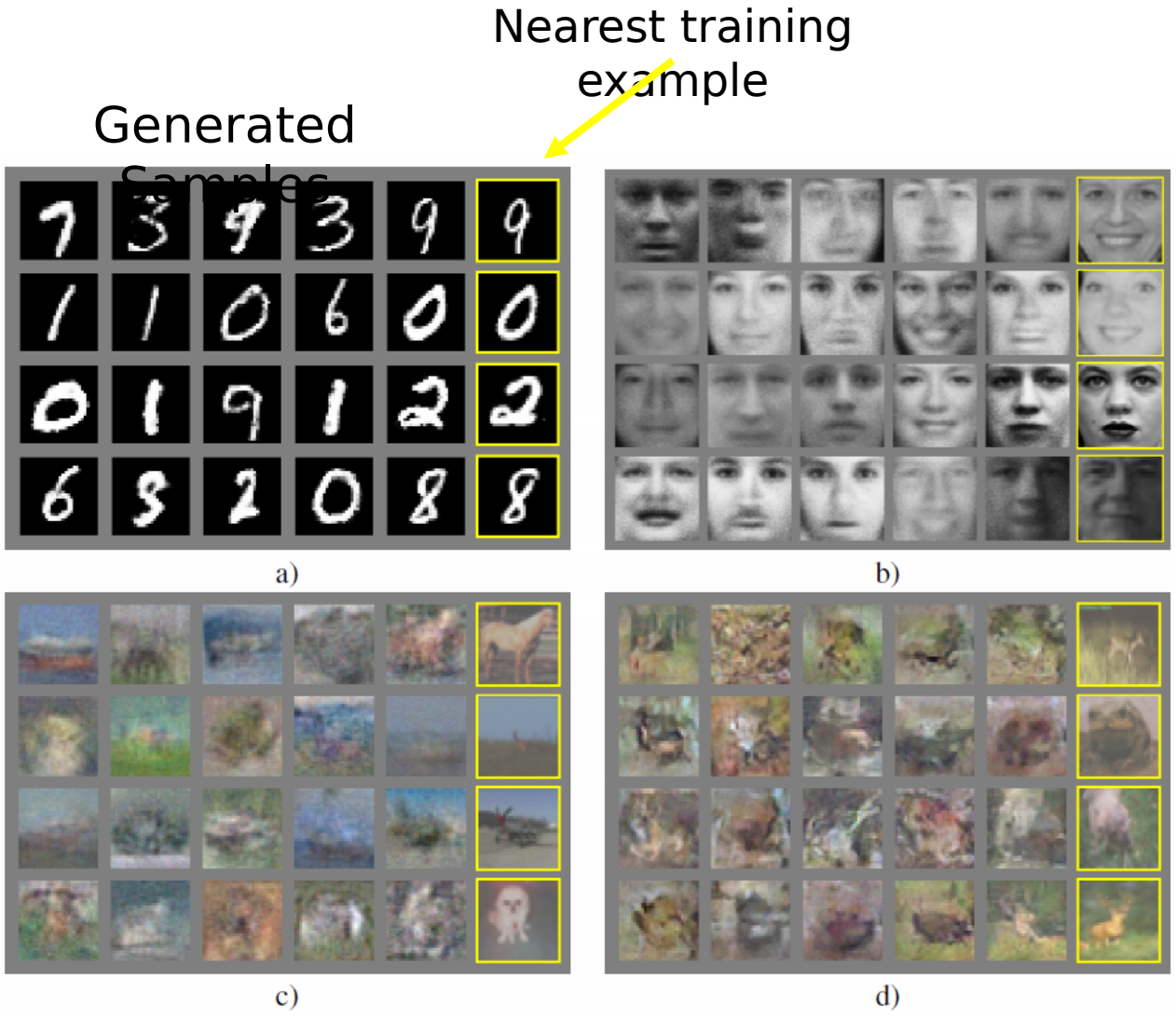
$$\min_G V_D(G) = \min\left[\mathbb{E}_{z\sim p_{z(z)}}[\log(1 - D(G(z)))]\right]$$

Try to make

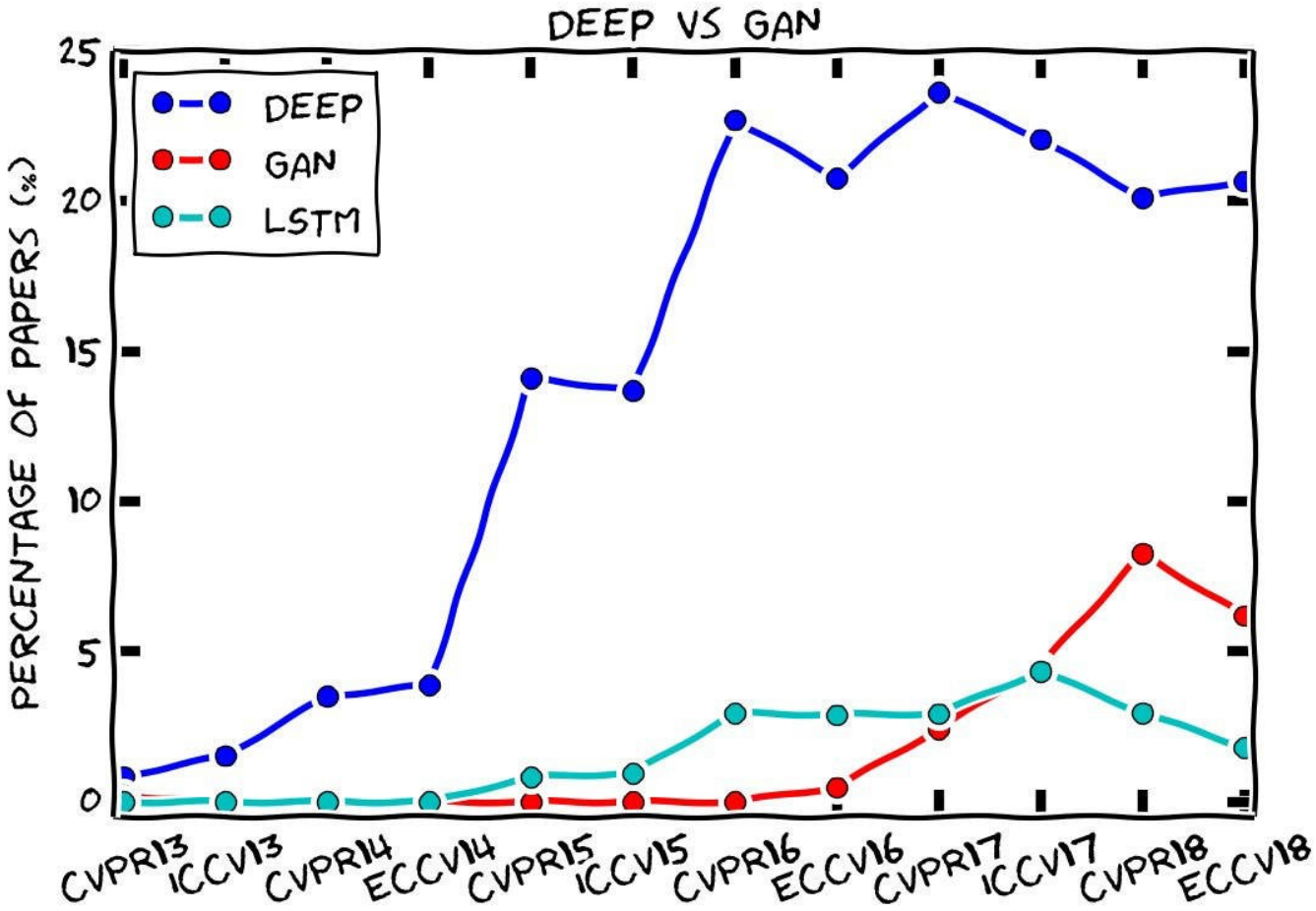
Try to make $D(G(z)) = 1$

D(G(z) =0, D wins, G fails
D(G(z) =1, D fails, G wins

Generative adversarial network (GAN)



Generative adversarial network (GAN)

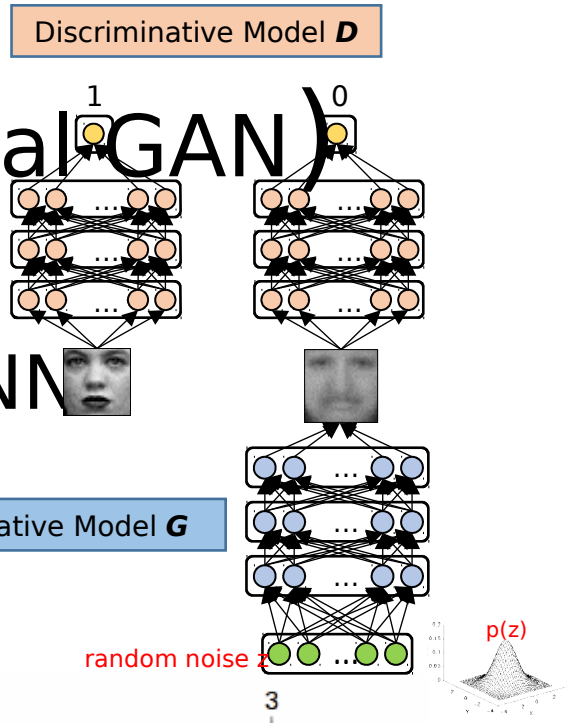


Recap

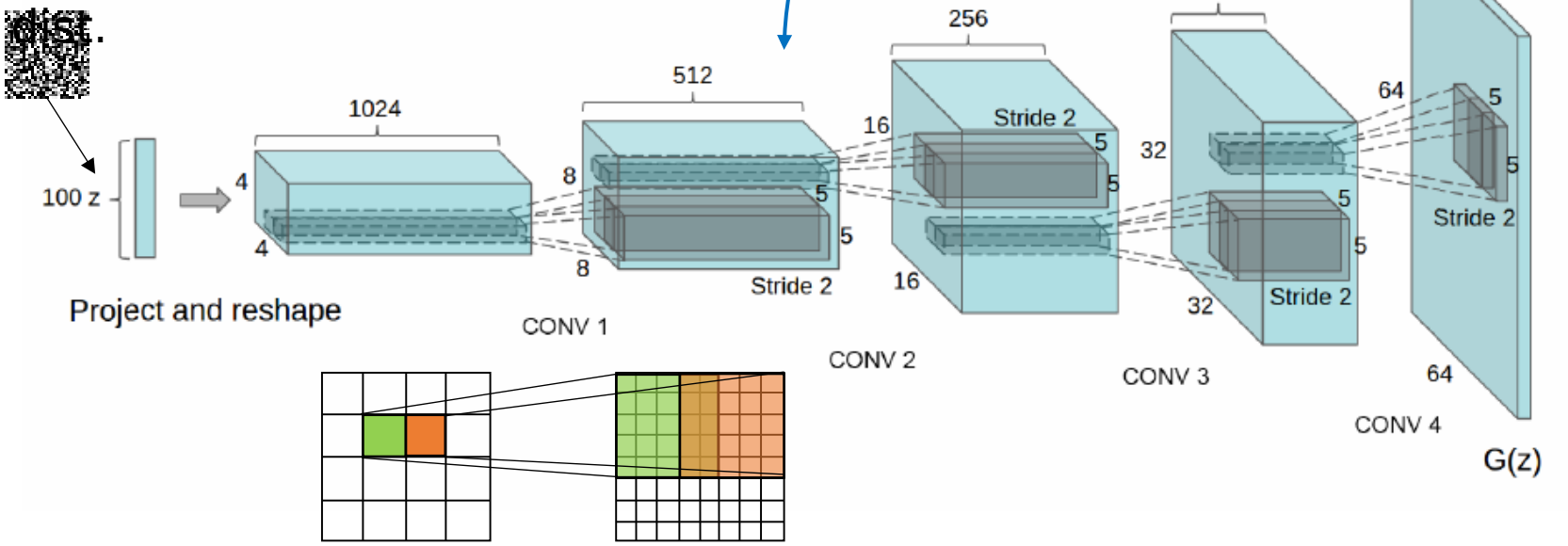
- Supervised learning vs. unsupervised learning
- Generative model
- Autoencoder (sparse AE); generative adversarial network (GAN)

DCGAN (Deep convolutional GAN)

- Replace model's network to CNN
- Example of generator G (same as D)

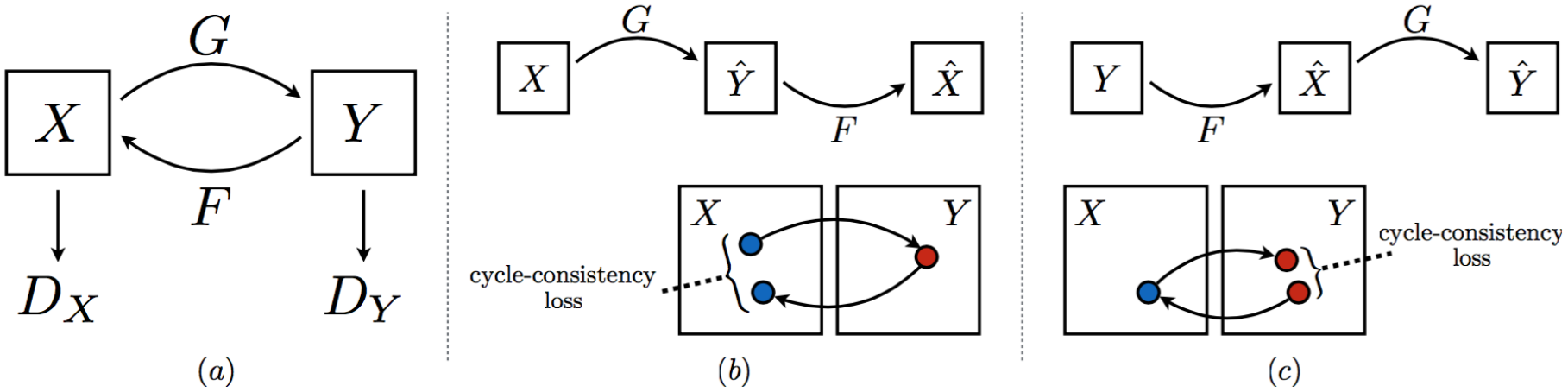


z: uniform



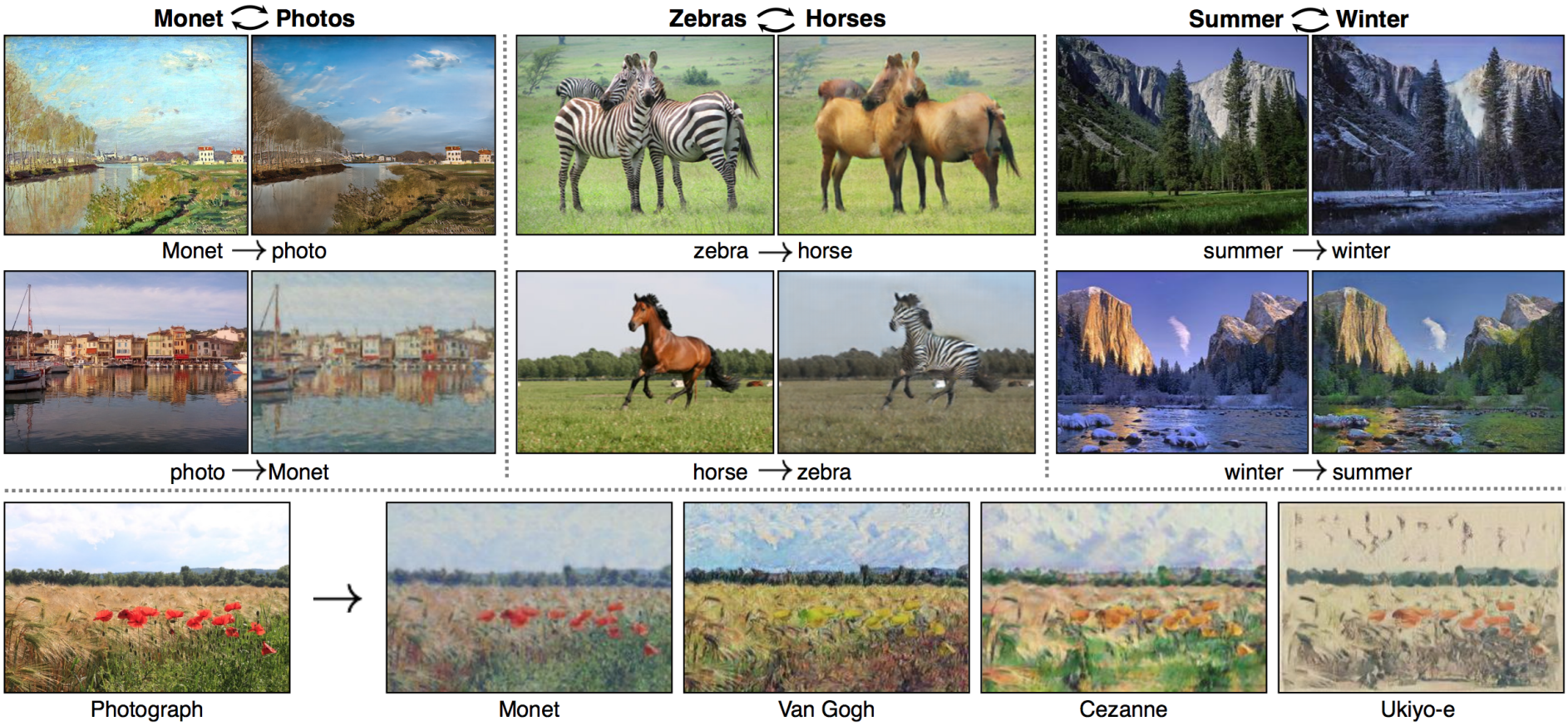
Radford A, etal. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR 2016

CycleGAN



Zhu, etal. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks In ICCV 201

CycleGAN



Zhu, etal. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks In ICCV 2

Thank you