Generative models

Vladislav Goncharenko

girafe ai

MIPT, spring 2024

Outline

- Problem statement: probabilistic view
- Generative models overview
- Autoencoders (not generative models!)
- Variational Autoencoders (VAE)
    - Conditional VAE
- Generative Adversarial Networks
    - Conditional GAN

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)

x is data, y is label

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

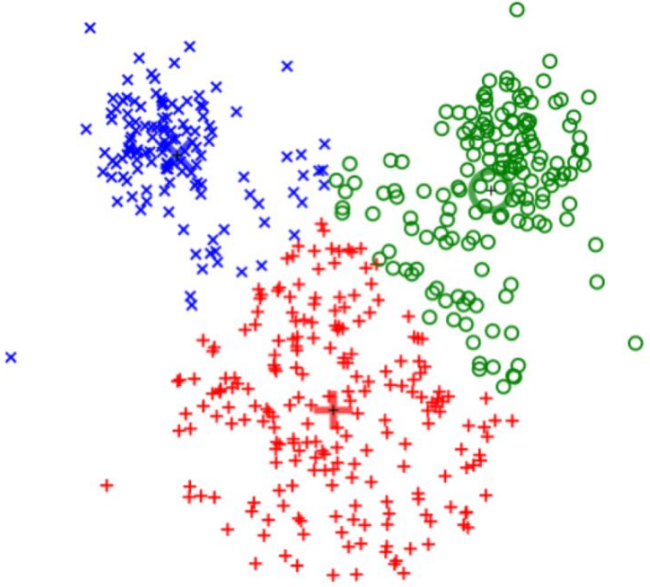
Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Classification

Cat

Supervised vs Unsupervised Learning

Clustering
(e.g. K-Means)



Unsupervised Learning

Data: x

Just data, no labels!

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

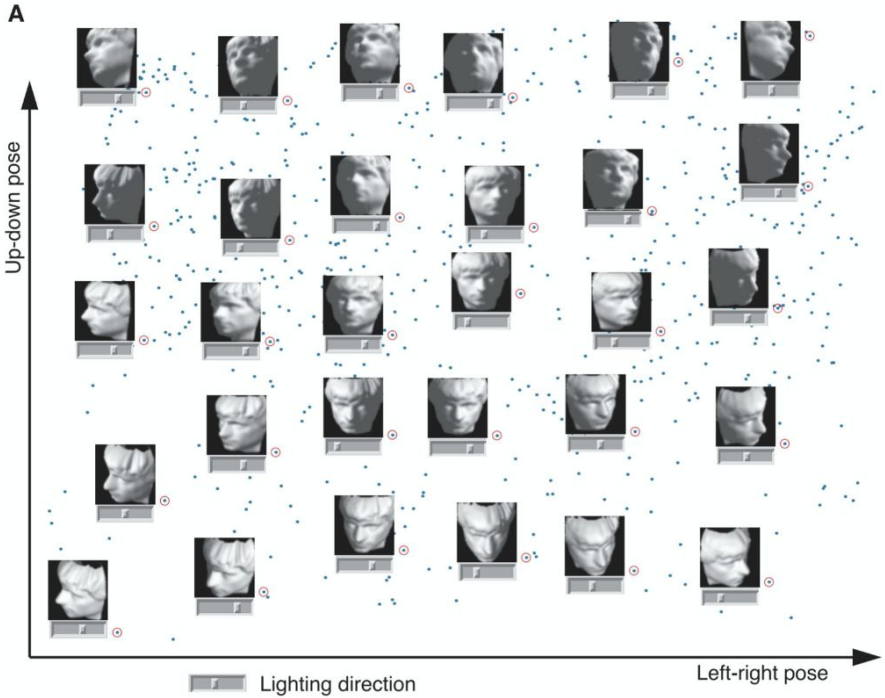
Examples: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Manifold assumption

The data lie approximately on a surface (called manifold) of usually much lower dimension than the input space

So problem dimensionality could be (non-)linearly reduced or other tasks solved

Sometimes dimensionality of manifold is referred as intrinsic dimension (see this article)



Tenenbaum, de Silva, Langford
A Global Geometric Framework for Nonlinear Dimensionality Reduction

Latent space

Latent (embedding) space describes data in coordinates more relevant to humans' reason and often allows useful linear operations:

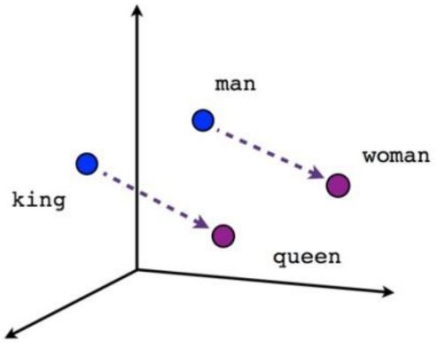
- Interpolation (A)
- Extrapolation (B)
- Analogy (C)

This process is also called embedding space 'walking'

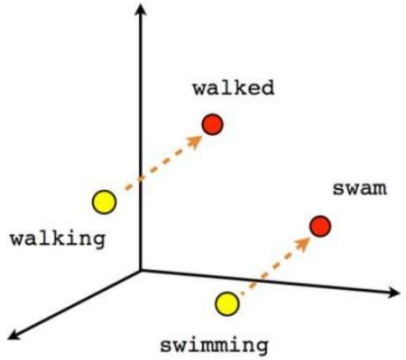
Latent space example

Word2vec is a method to embed words from text corpus into linear space

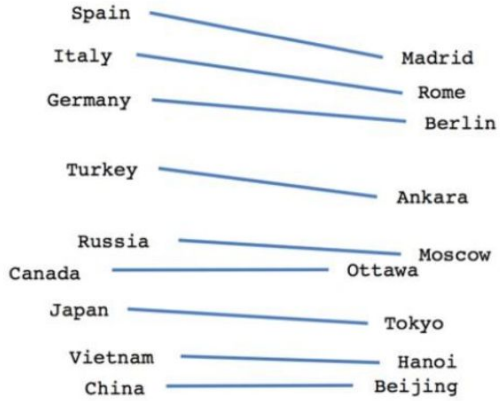
Read more: manifold assumption, assessing assumption



Male-Female



Verb tense



Country-Capital

Discriminative vs Generative Models

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

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)

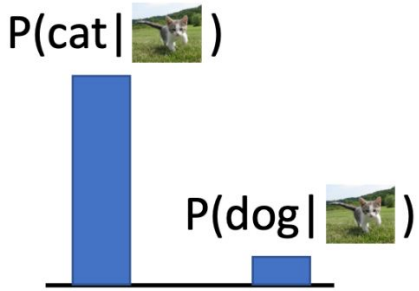
Data: x

Label: y

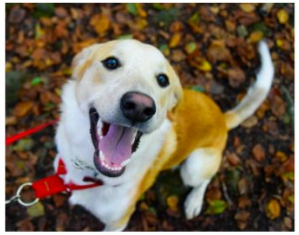
Cat

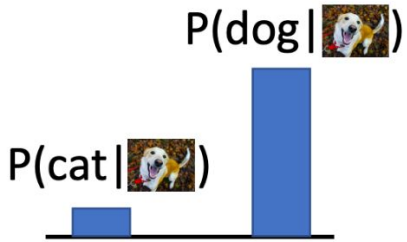
Discriminative vs Generative Models

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



Generative Model: Learn a probability distribution p(x)





Conditional Generative Model: Learn p(x|y)

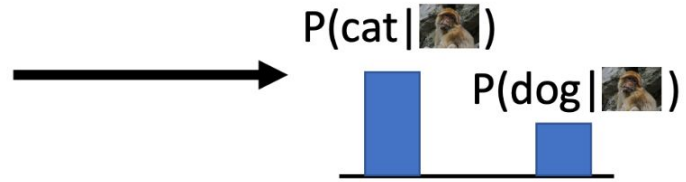
Discriminative model: the possible labels for each input "compete" for probability mass. But no competition between **images**

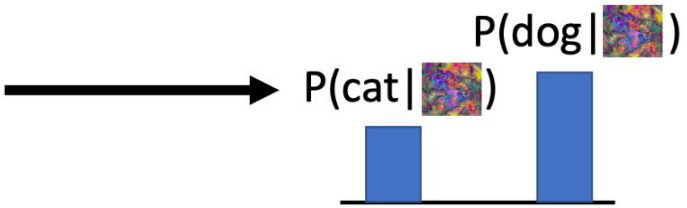
Discriminative vs Generative Models

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

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)





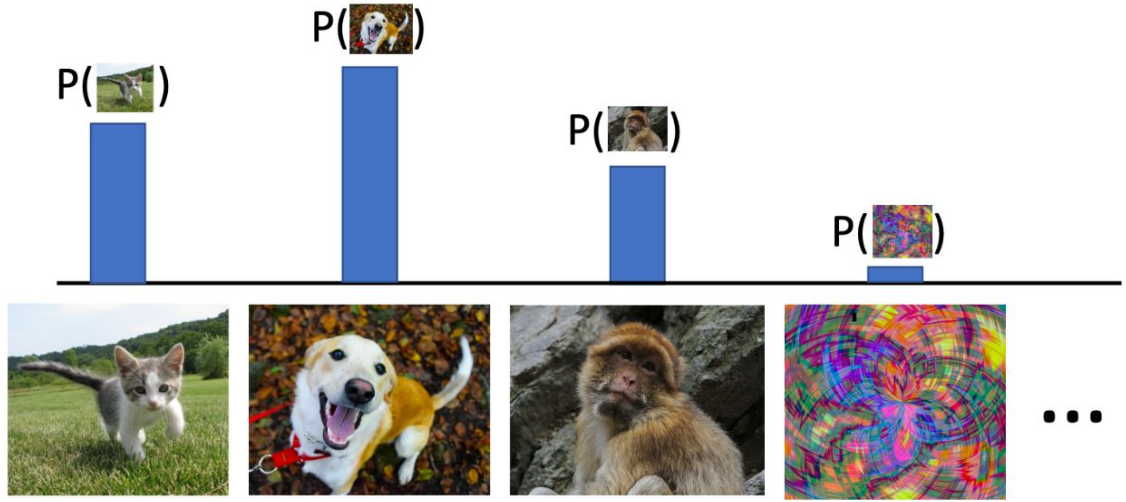
Discriminative model: No way for the model to handle unreasonable inputs; it must give label distributions for all images

Discriminative vs Generative Models

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

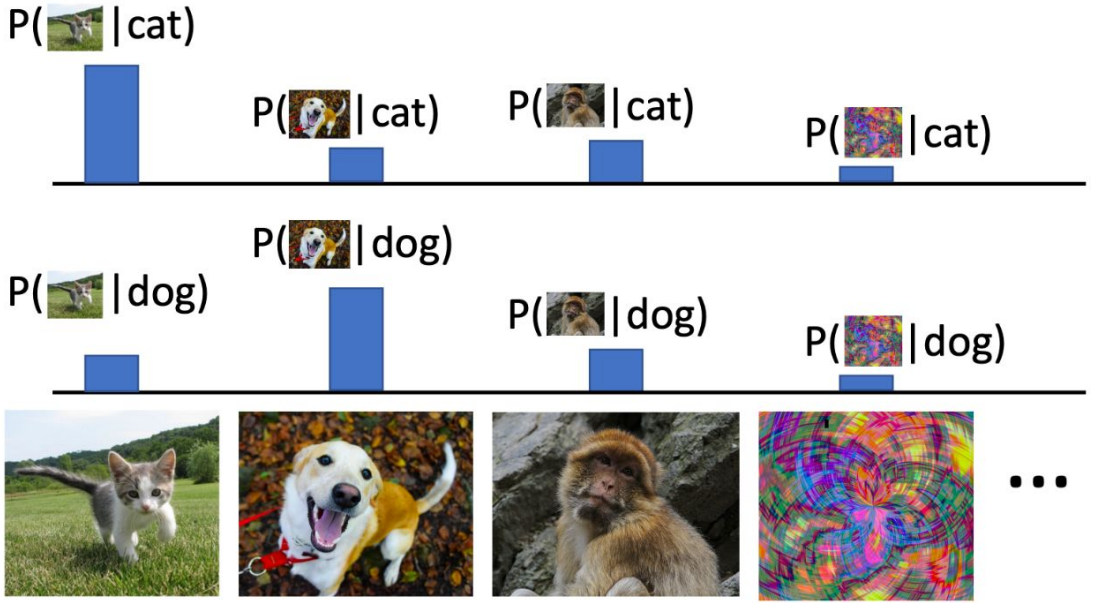
Model can “reject” unreasonable inputs by assigning them small values

Discriminative vs Generative Models

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

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)



Conditional Generative Model: Each possible label induces a competition among all images

Discriminative vs Generative Models

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:**

$$P(x \mid y) = \frac{P(y \mid x)}{P(y)} P(x)$$

Conditional Generative Model

Discriminative Model

Prior over labels

(Unconditional) Generative Model

We can build a conditional generative model from other components!

What can we do with a generative model?

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

→ Assign labels to data
Feature learning (supervised)

Generative Model:
Learn a probability distribution p(x)

→ Detect outliers
Feature learning (unsupervised)
Sample to **generate** new data

Conditional Generative Model: Learn p(x|y)

→ Assign labels, while rejecting outliers!
Generate new data conditioned on input labels

Taxonomy of Generative Models

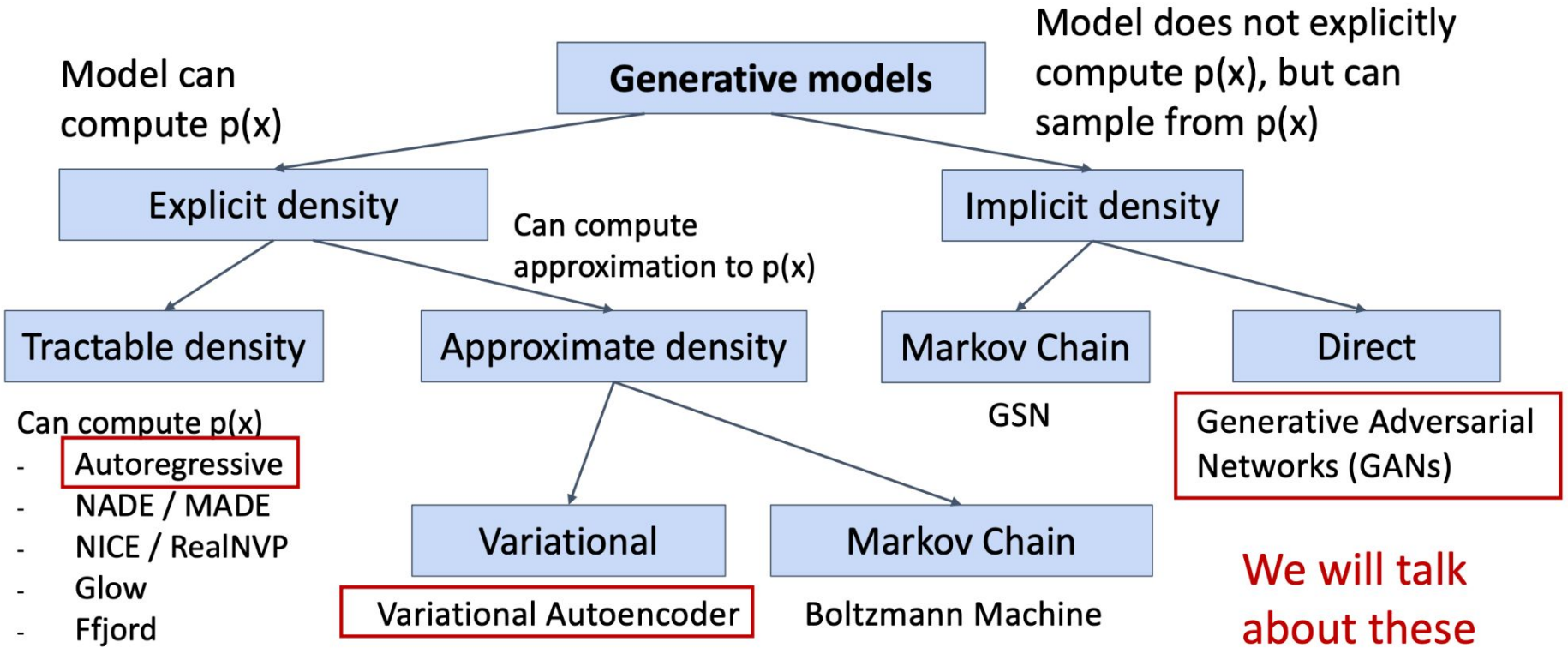


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

Autoencoders

Denote **z** as encoded with encoder E input **x**

$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers **x** from latent representation

$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

Optimal parameters learned w.r.t. loss function L

$$[\boldsymbol{\theta}_E, \boldsymbol{\theta}_D] = \arg\min L(\hat{\mathbf{x}}, \mathbf{x})$$

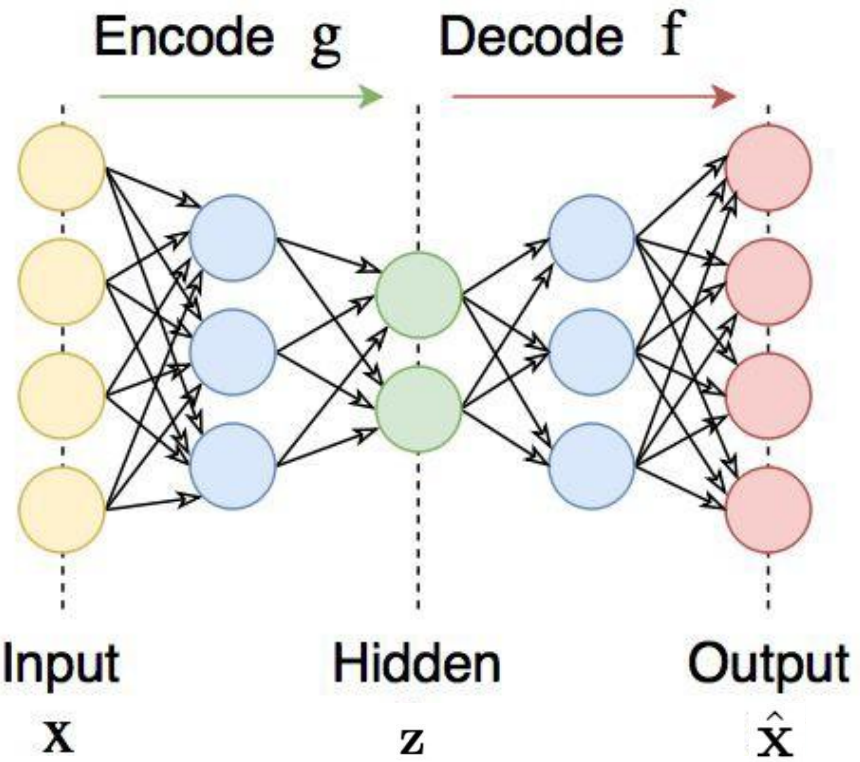


Image source: [Habr post on autoencoders and GANs]

Autoencoders

Denote **z** as encoded with encoder E input **x**

$$\mathbf{z} = E(\mathbf{x}, \boldsymbol{\theta}_E)$$

Decoder D recovers **x** from latent representation

$$\hat{\mathbf{x}} = D(\mathbf{z}, \boldsymbol{\theta}_D)$$

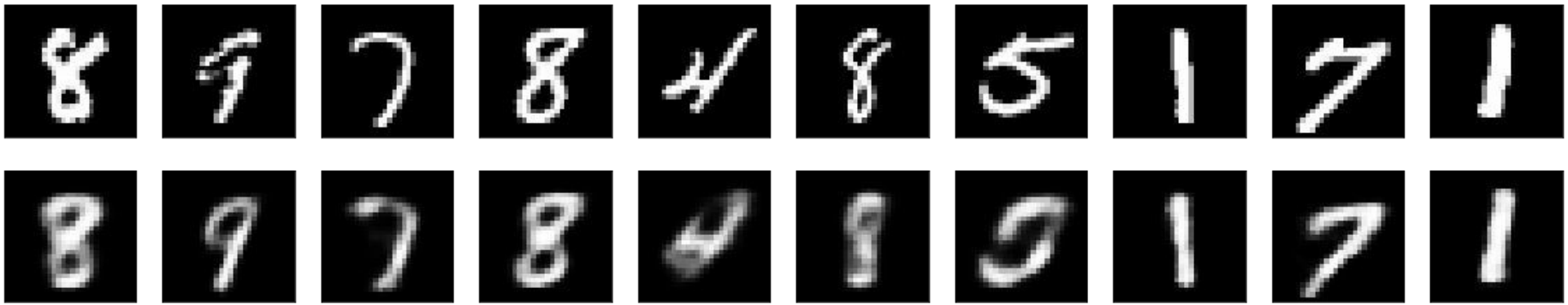
Simple example: PCA

Optimal parameters learned w.r.t. loss function L

$$[\boldsymbol{\theta}_E, \boldsymbol{\theta}_D] = \arg\min L(\hat{\mathbf{x}}, \mathbf{x})$$

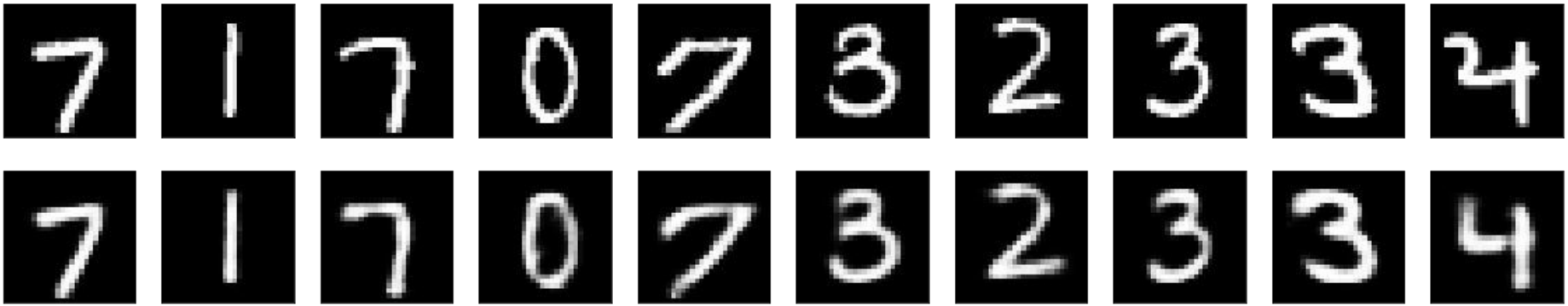
Image source: Habr post on autoencoders and GANs

PCA performance on MNIST



16 components

Convolutional performance on MNIST

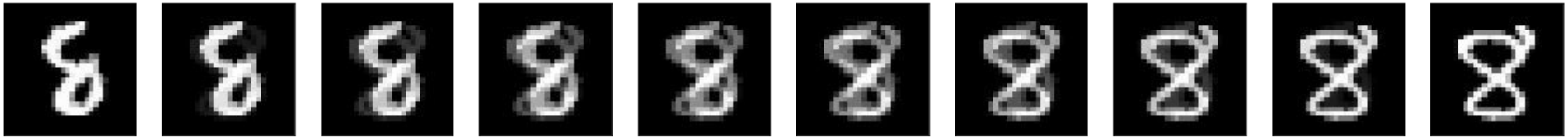


7 x 7 latent space

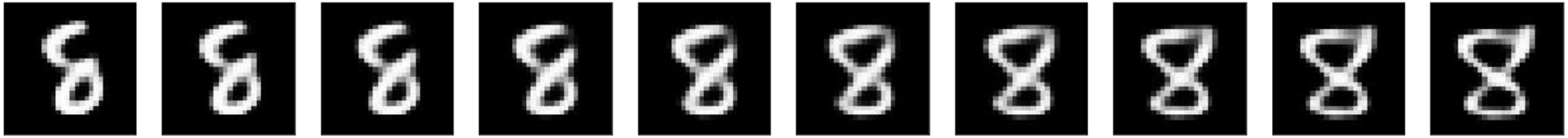
Homotopy between samples

10 steps between samples

- In original feature space (28 x 28):



- In latent space (7 x 7):



Latent space structure

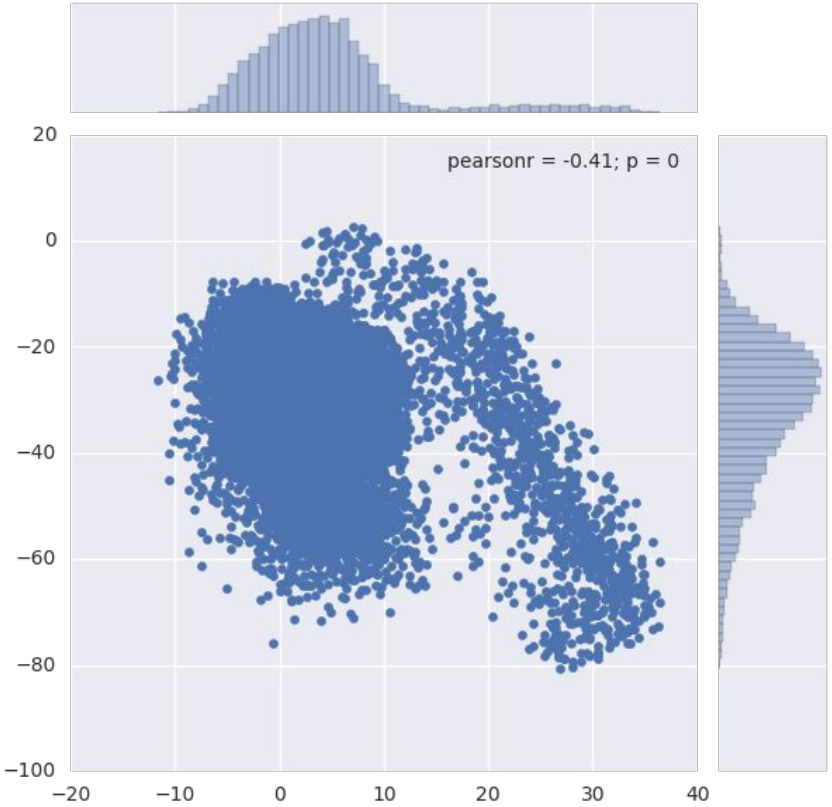


Image source: [Habr post on autoencoders and GANs]

VAE intuition

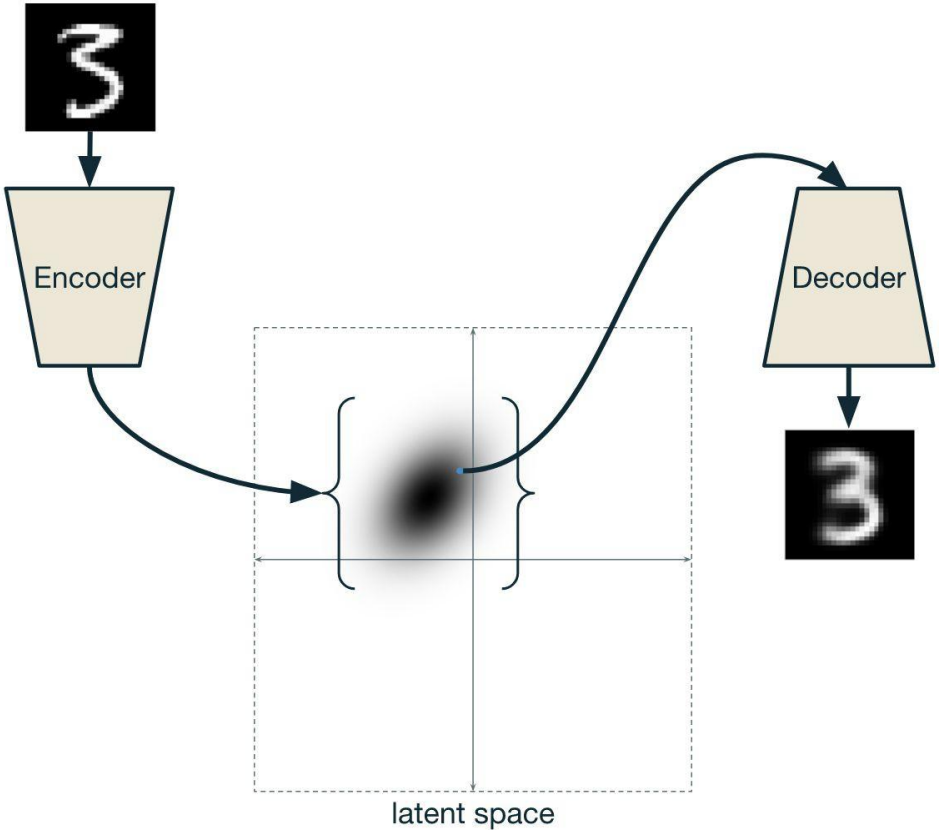


Image source: Conditional Variational Autoencoders by Isaac Dykeman, 2016

KL divergence

Denote distributions $Q(z)$ and $P(z|X)$.

Kullback–Leibler divergence is defined as

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

KL divergence

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

Applying the Bayes rule:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

KL divergence

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

Applying the Bayes rule:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

$$\log P(X) - \mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z)\|P(z)\right]$$

KL divergence

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

Applying the Bayes rule:

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

$$\log P(X) - \mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z)\|P(z)\right]$$

$$\log P(X) - \mathcal{D}\left[Q(z|X)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X)\|P(z)\right]$$

This equation is the core of Variational Autoencoders

Structure of the latent space

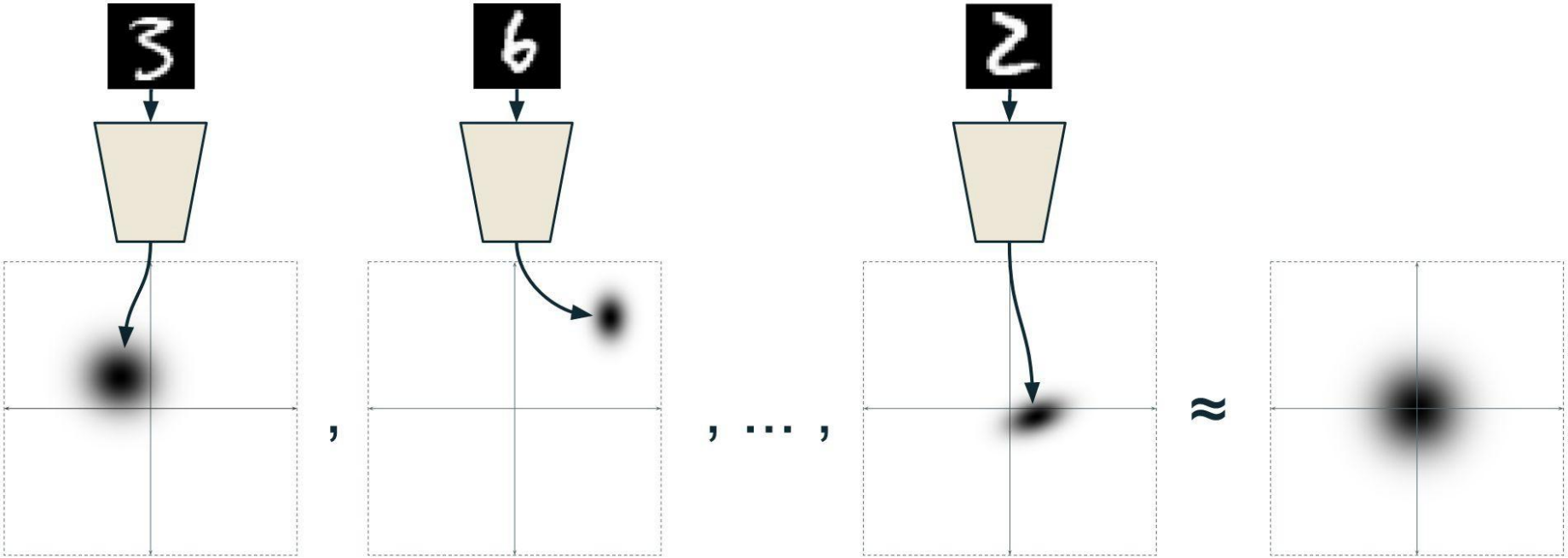


Image source: Conditional Variational Autoencoders by Isaac Dykeman, 2016

VAE so far

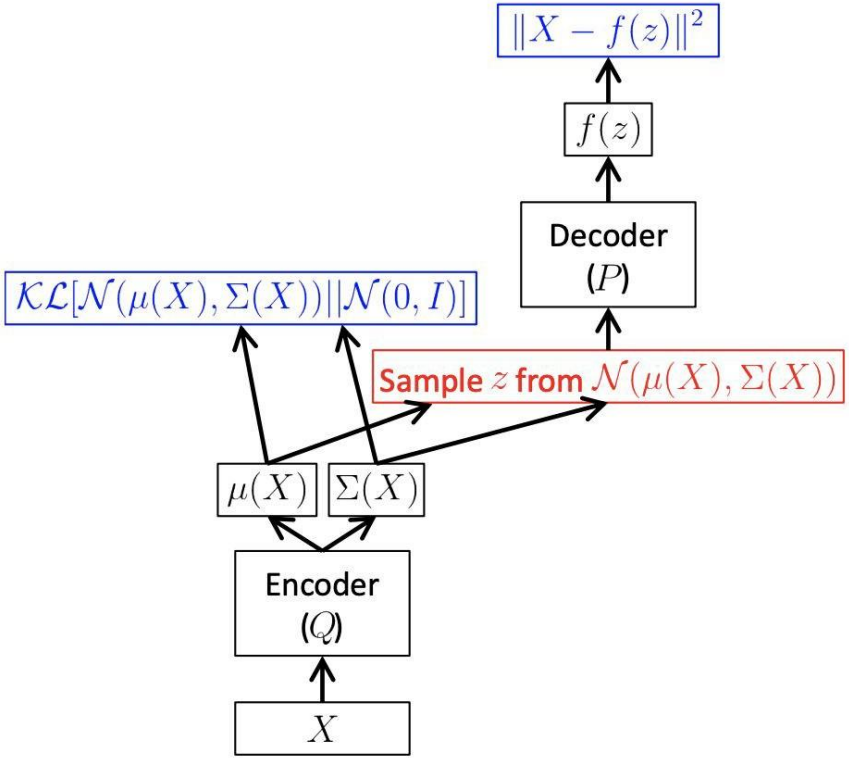


Image source: Tutorial on Variational Autoencoders by Carl Doersch, 2016

VAE so far

$$\mathcal{D}[\mathcal{N}(\mu(X),\Sigma(X))\|\mathcal{N}(0,I)] = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma(X)\right) + \left(\mu(X)\right)^{\top}\left(\mu(X)\right) - k - \log\det\left(\Sigma(X)\right)\right)$$

Try to derive it by yourself

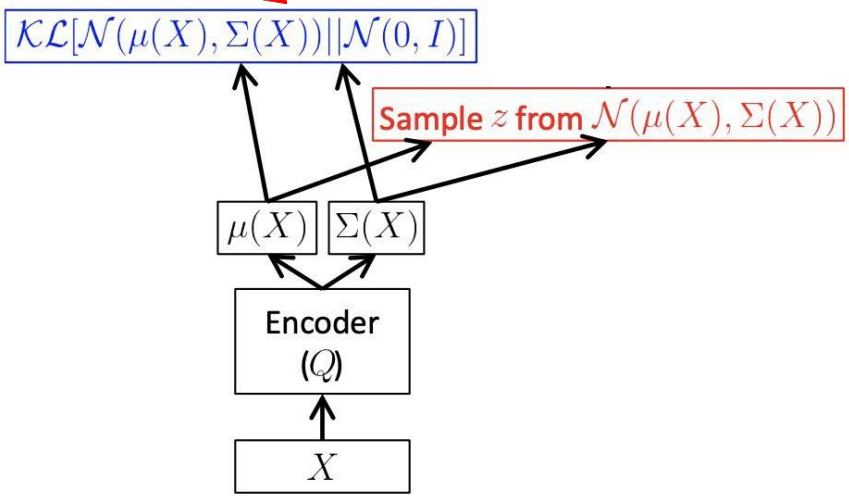
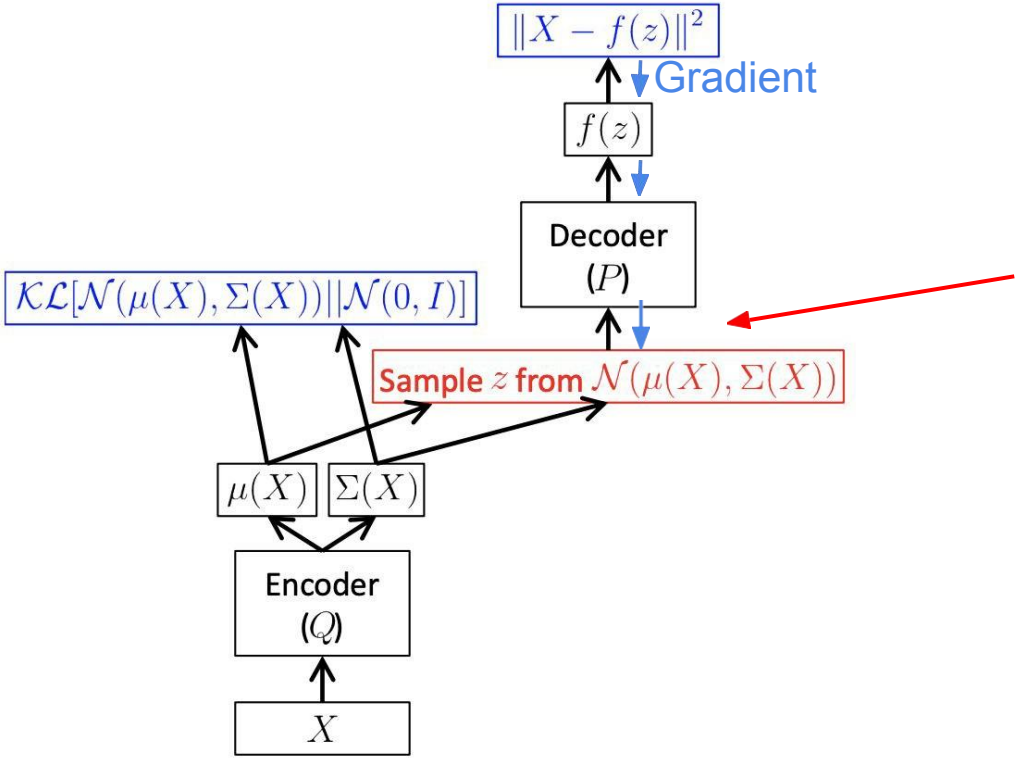


Image source: Tutorial on Variational Autoencoders by Carl Doersch, 2016

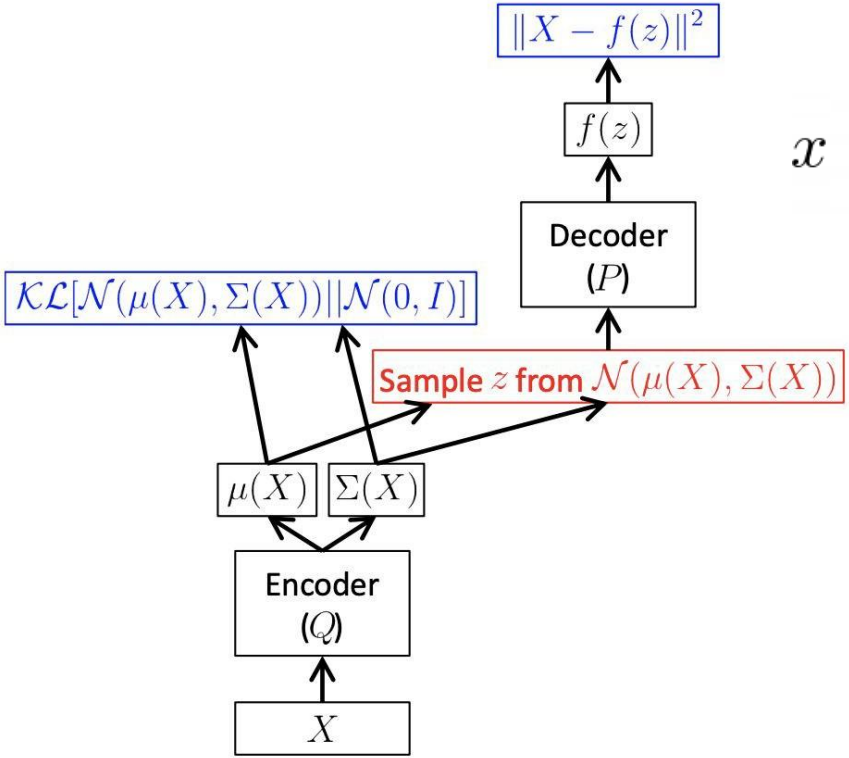
VAE so far



Gradient flow is broken!

Image source: Tutorial on Variational Autoencoders by Carl Doersch, 2016

Reparametrization trick



$$x \sim \mathcal{N}(\mu, \sigma^2), \quad z = \frac{x - \mu}{\sigma} \sim \mathcal{N}(0, 1)$$

$$\Rightarrow x = \sigma(z + \mu) \sim \mathcal{N}(\mu, \sigma^2)$$

Image source: Tutorial on Variational Autoencoders by Carl Doersch, 2016

Reparametrization trick

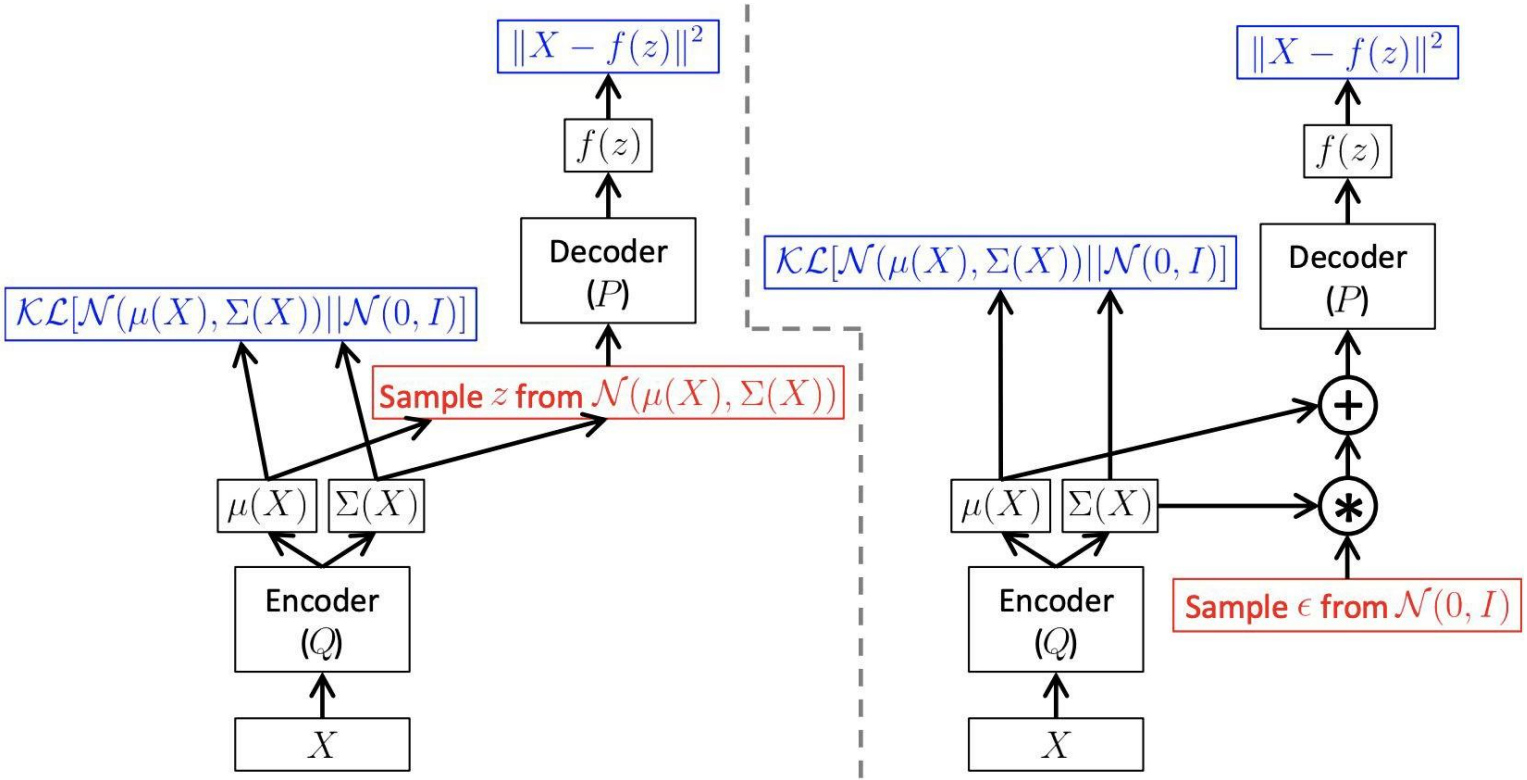
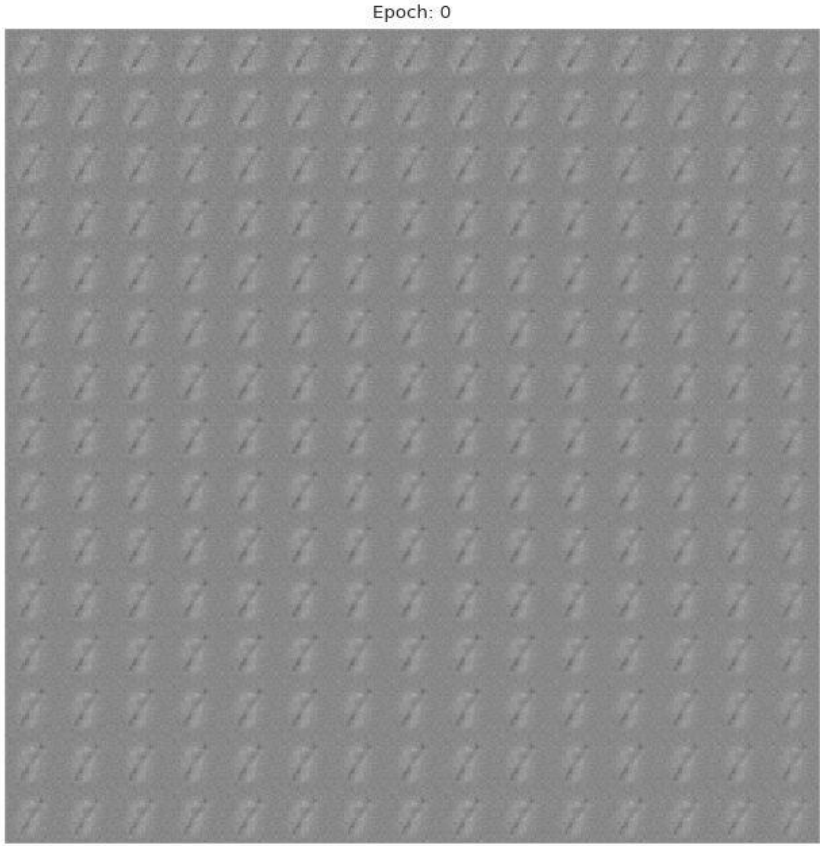


Image source: [Tutorial on Variational Autoencoders by Carl Doersch, 2016]

VAE manifold



Structure of the latent space

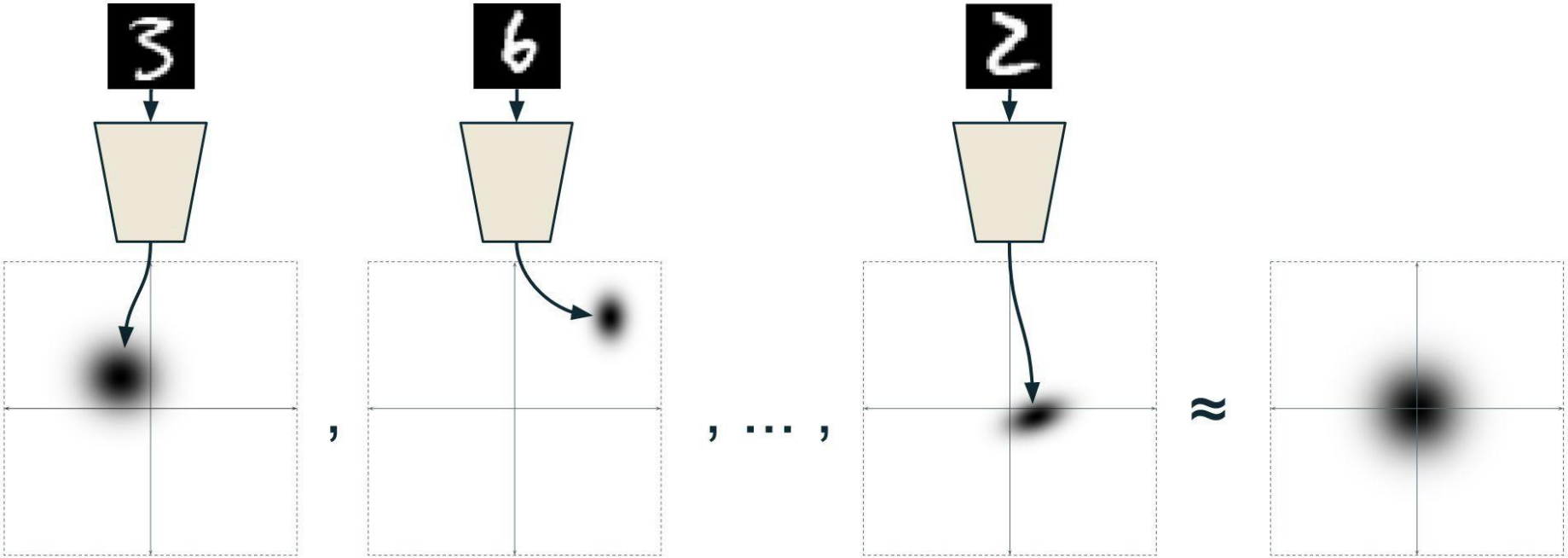
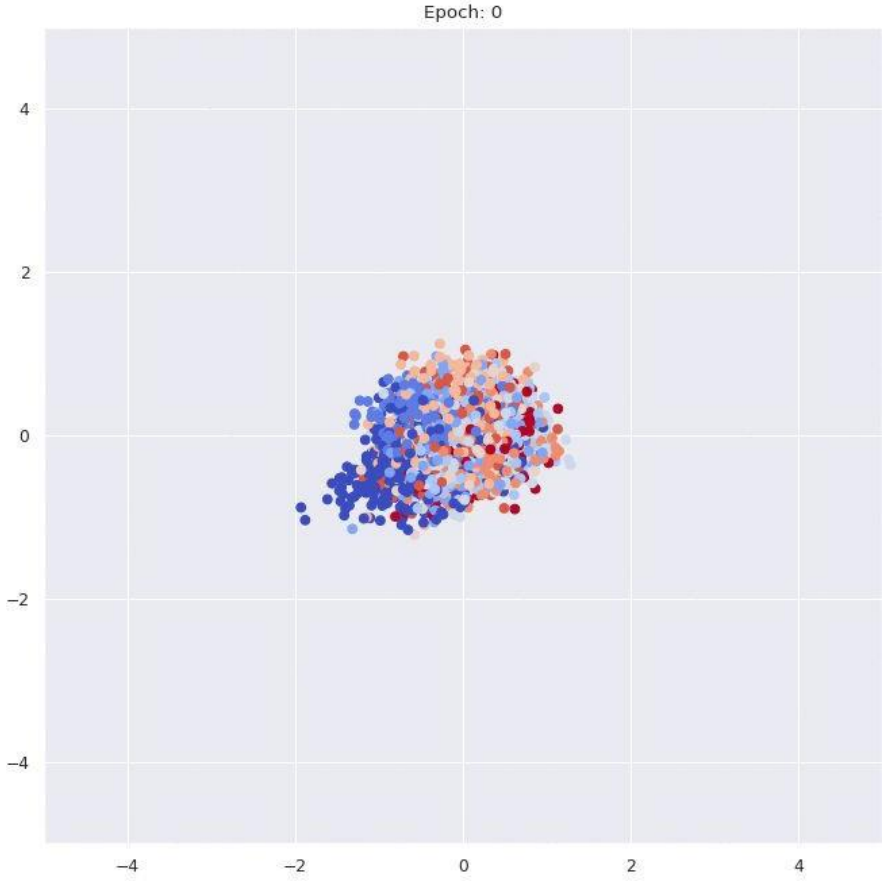


Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

VAE latent space distribution



Conditional VAE intuition

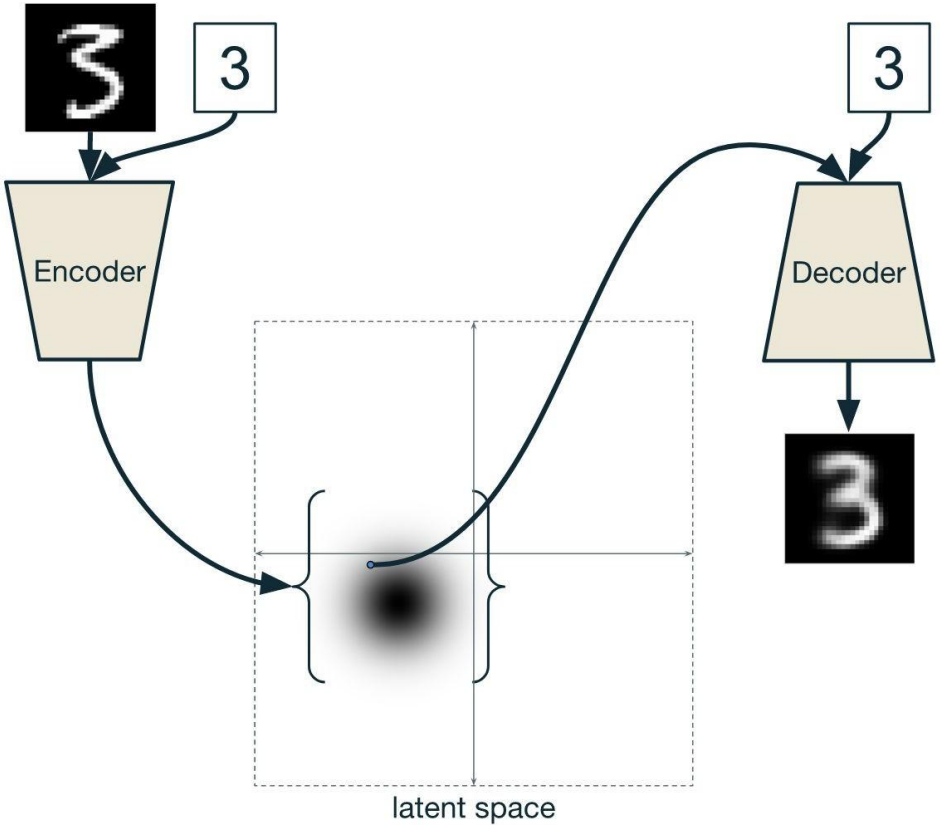


Image source: [Conditional Variational Autoencoders by Isaac Dykeman, 2016]

Conditional VAE

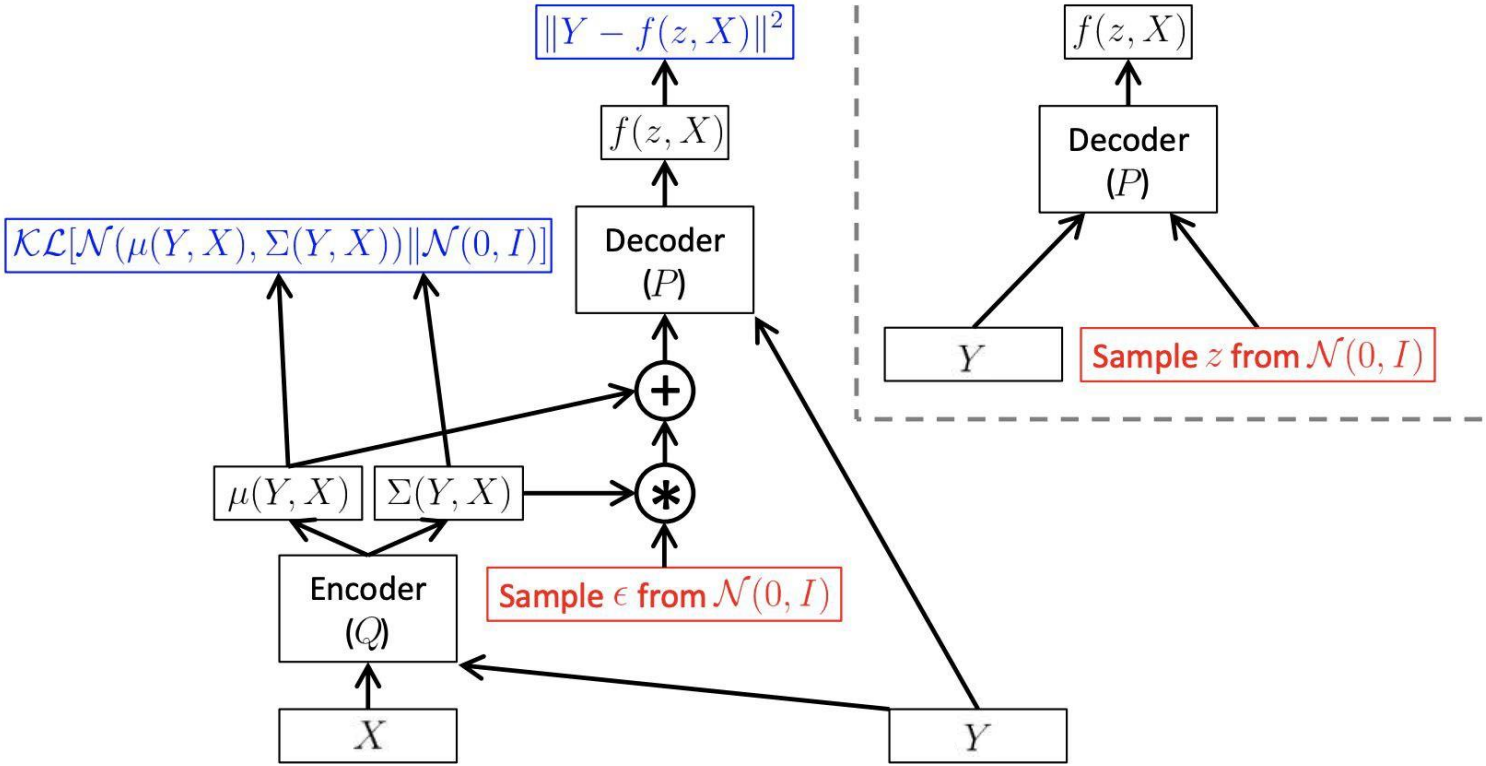
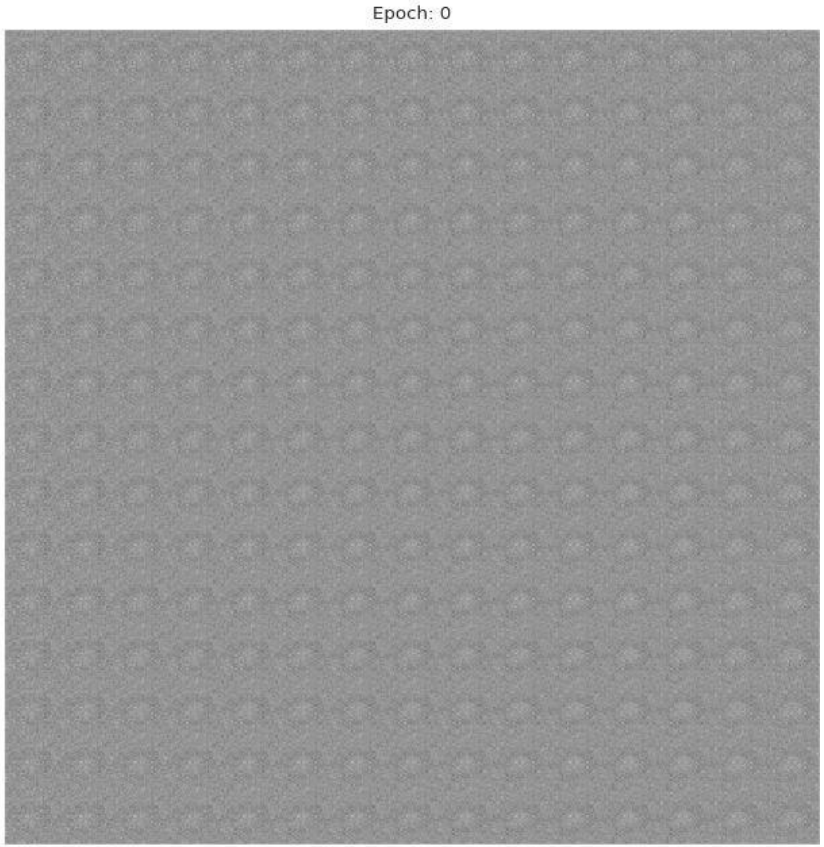
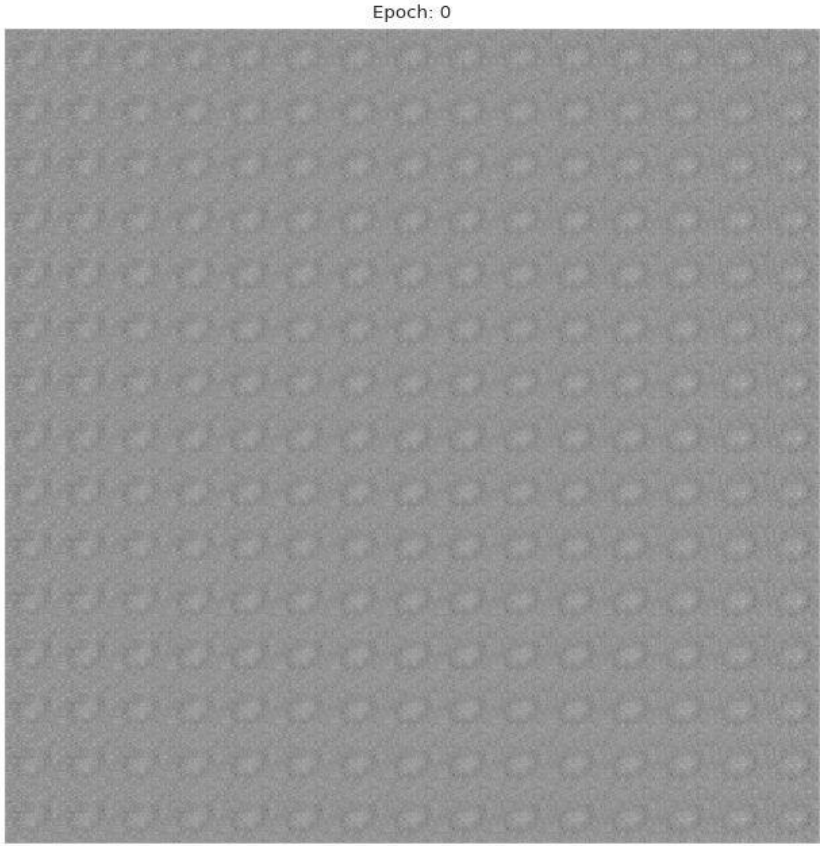


Image source: [Tutorial on Variational Autoencoders by Carl Doersch, 2016]

cVAE manifold



cVAE manifold



Transferring style with VAE

Seed label is 1

Once again

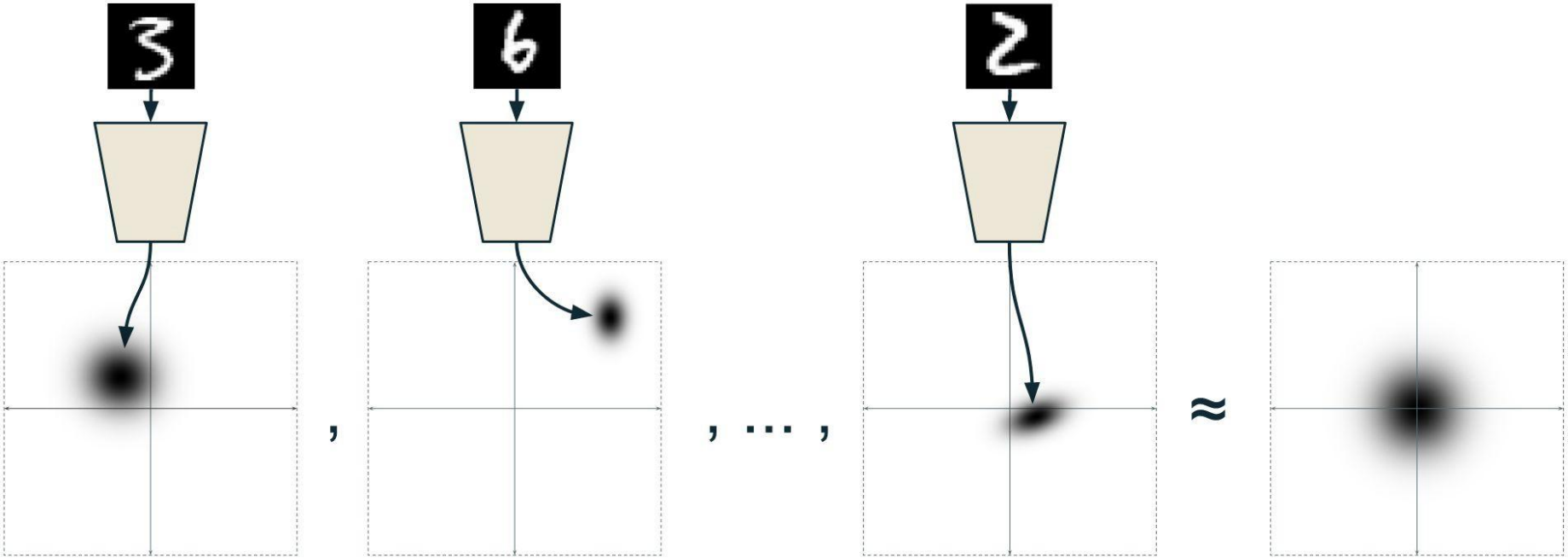
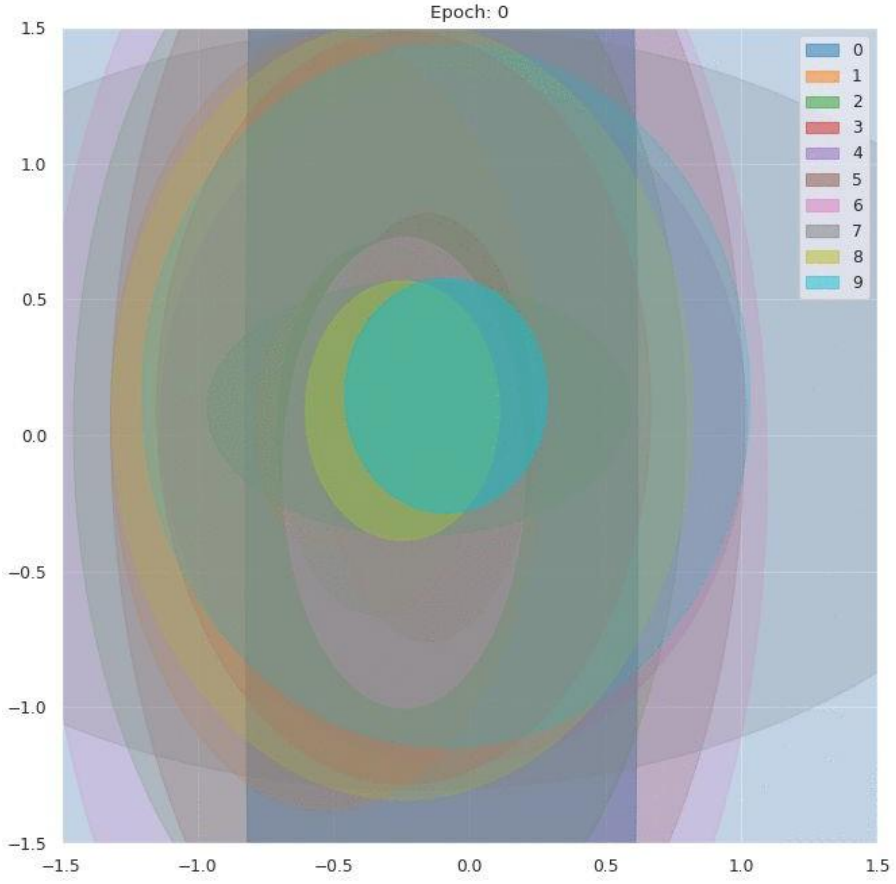


Image source: Conditional Variational Autoencoders by Isaac Dykeman, 2016

cVAE latent space distribution



GAN

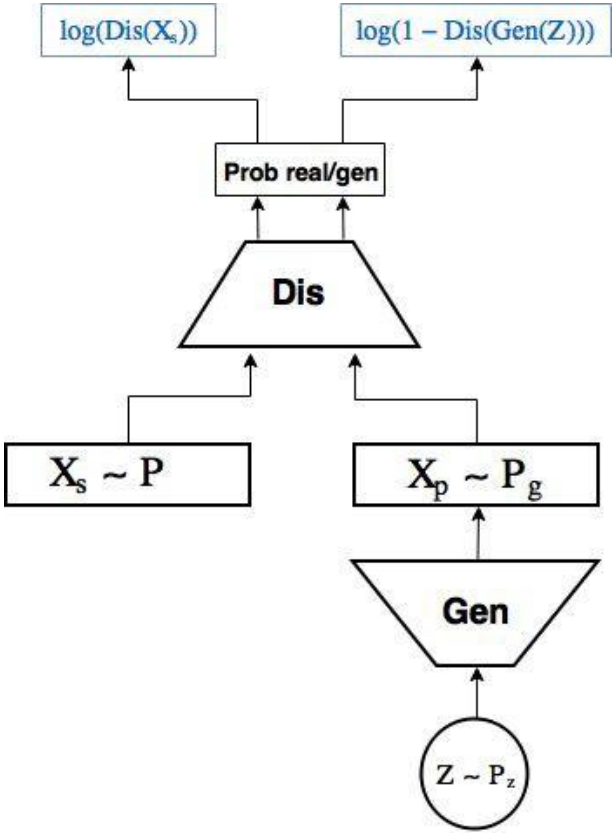
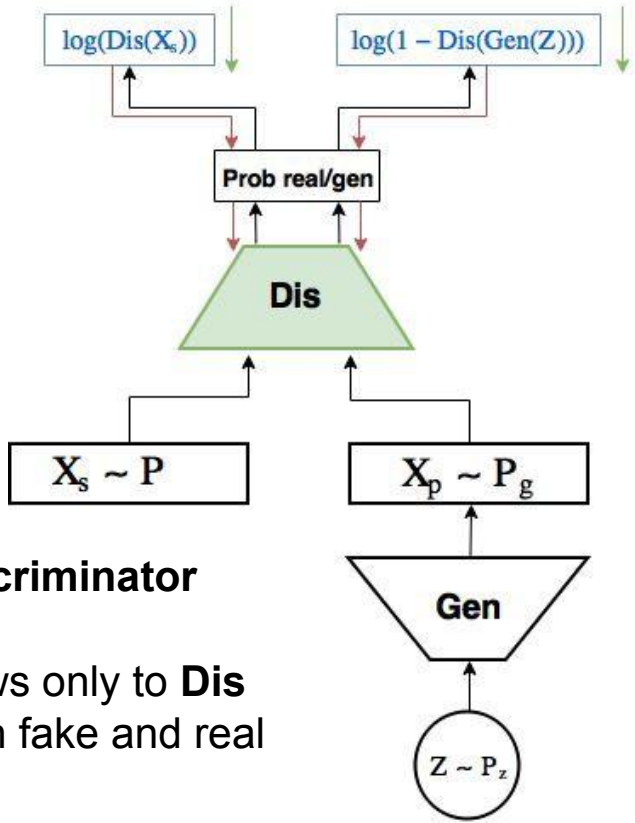
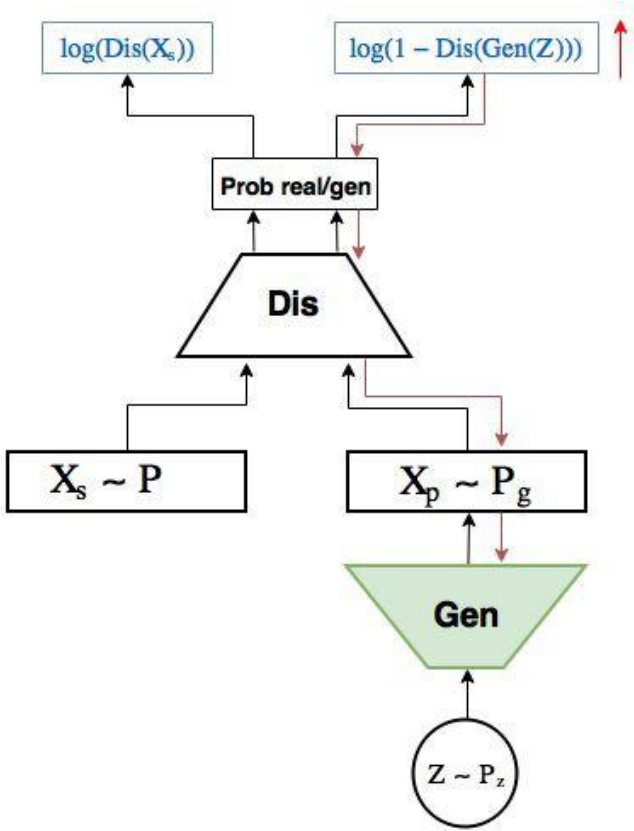


Image source: [Habr post on autoencoders and GANs]

Training GAN



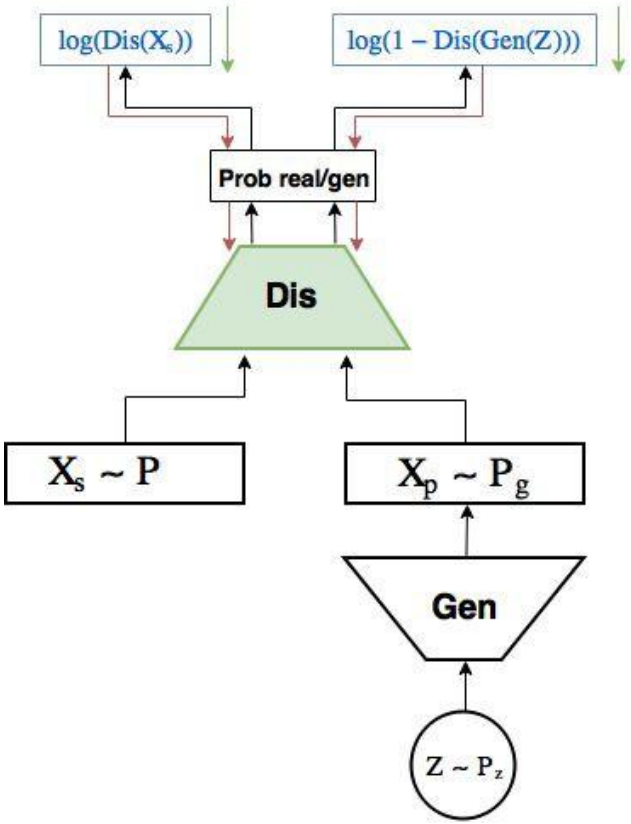


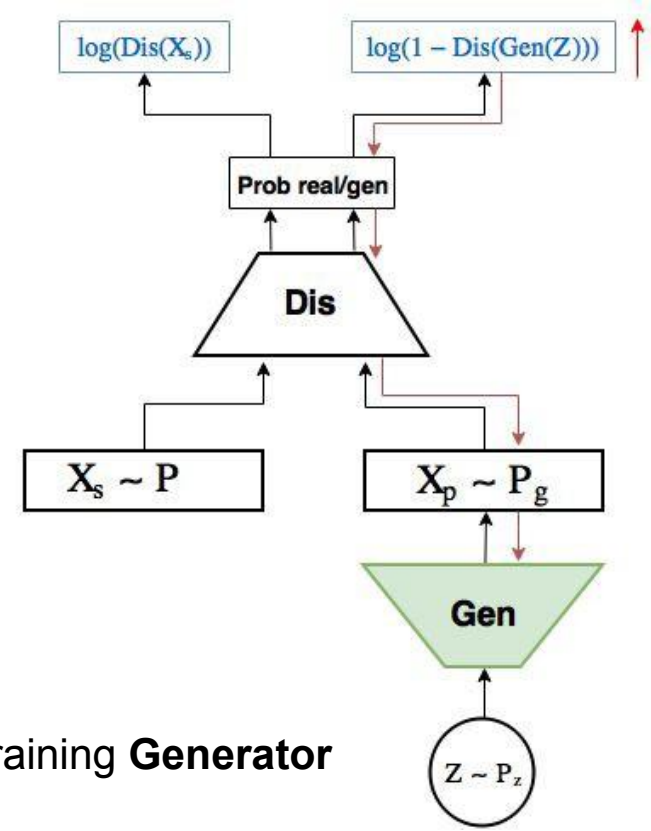
Training **Discriminator**

Gradient flows only to **Dis** to distinguish fake and real examples

Image source: [Habr post on autoencoders and GANs]

Training GAN





Training **Generator**

Gradient flows to **Gen** with **Dis** weights freezed to fool the Discriminator

Image source: [Habr post on autoencoders and GANs]

Optimization process in GAN

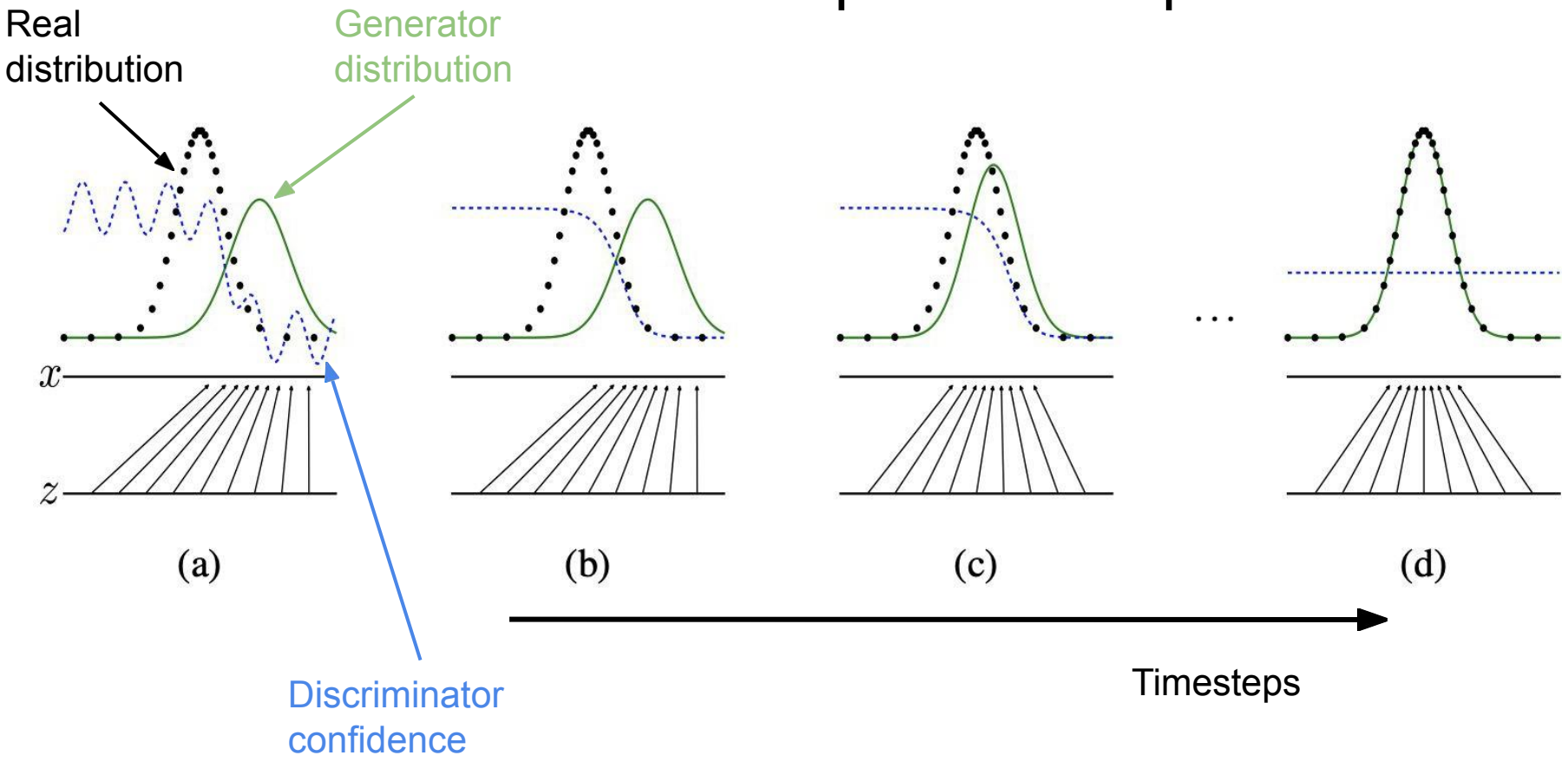


Image source: Generative Adversarial Nets, Ian J. Goodfellow et al, 2014

GAN manifold

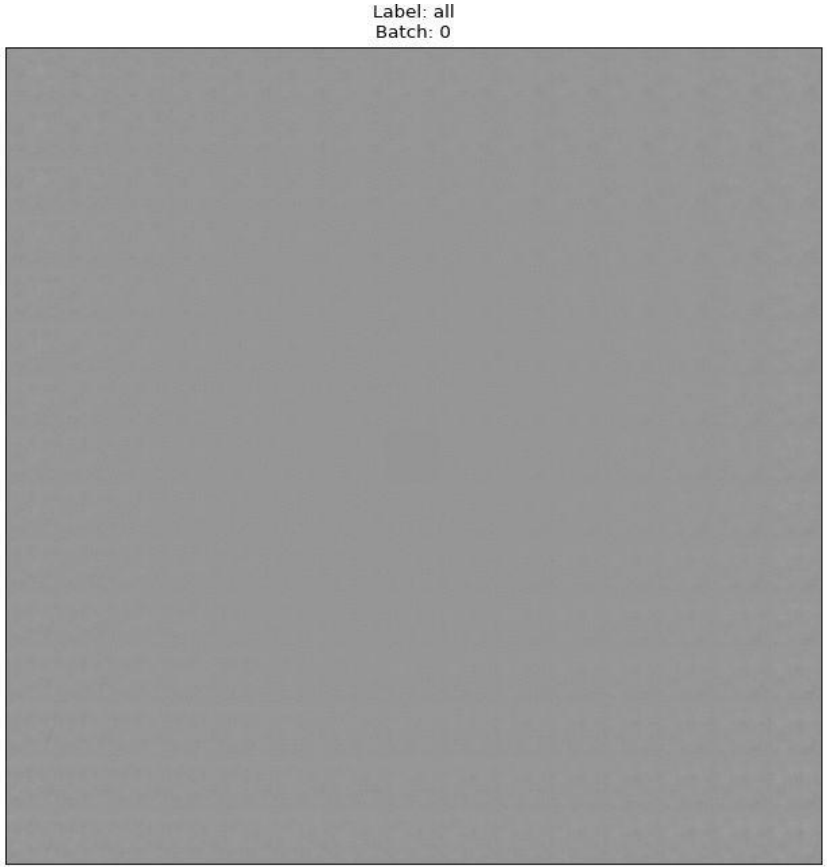


Image source: [Habr post on autoencoders and GANs]

Conditional GAN

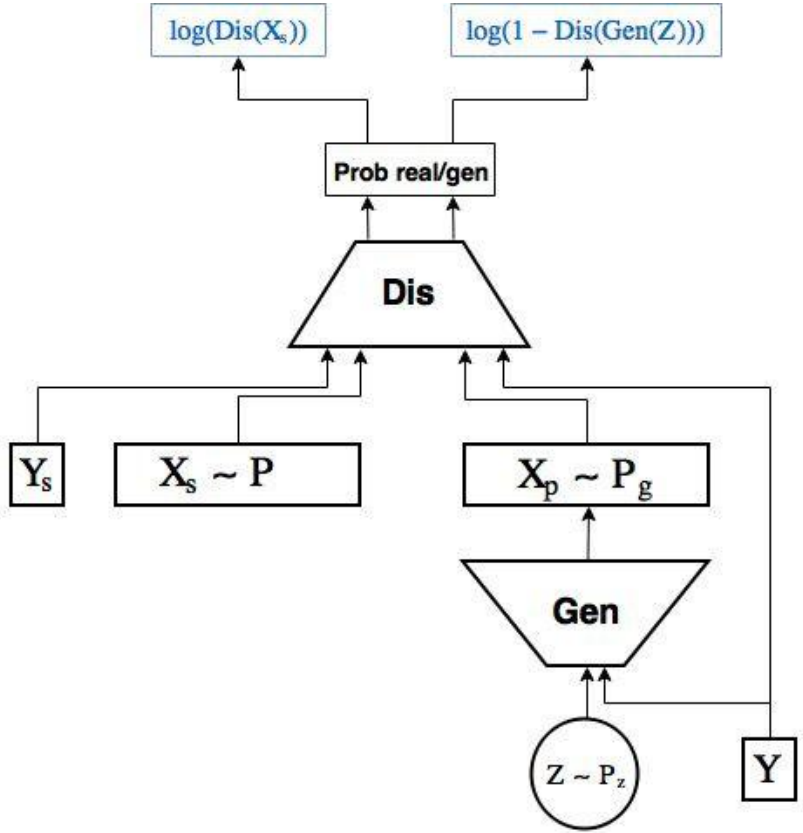
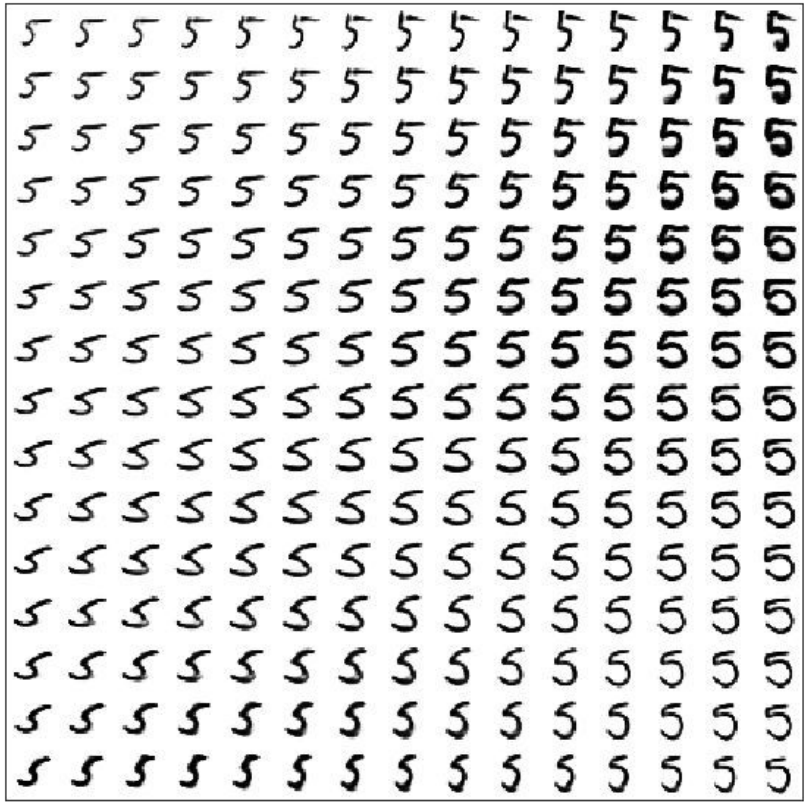


Image source: [Habr post on autoencoders and GANs]

cGAN manifolds



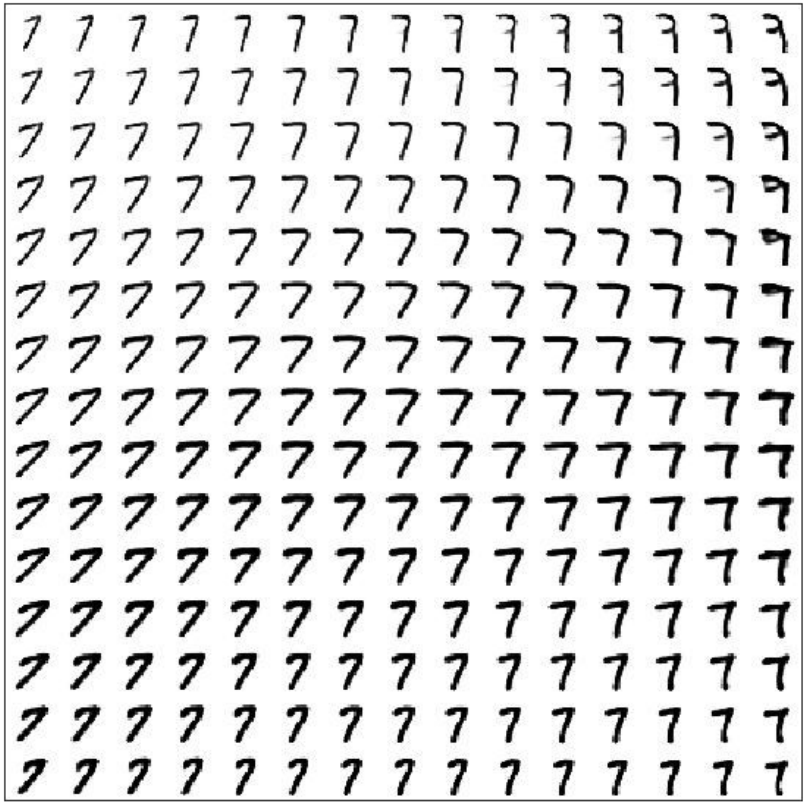


Image source: [Habr post on autoencoders and GANs]

Some more combinations

Image source: Guiness World Records blog, original video PIKOTARO - PPAP (Pen Pineapple Apple Pen)

Simple GAN

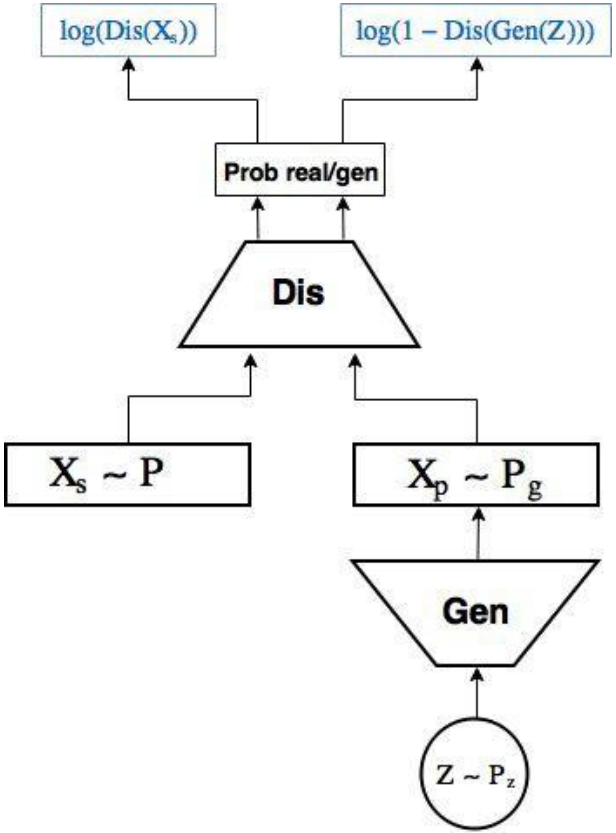
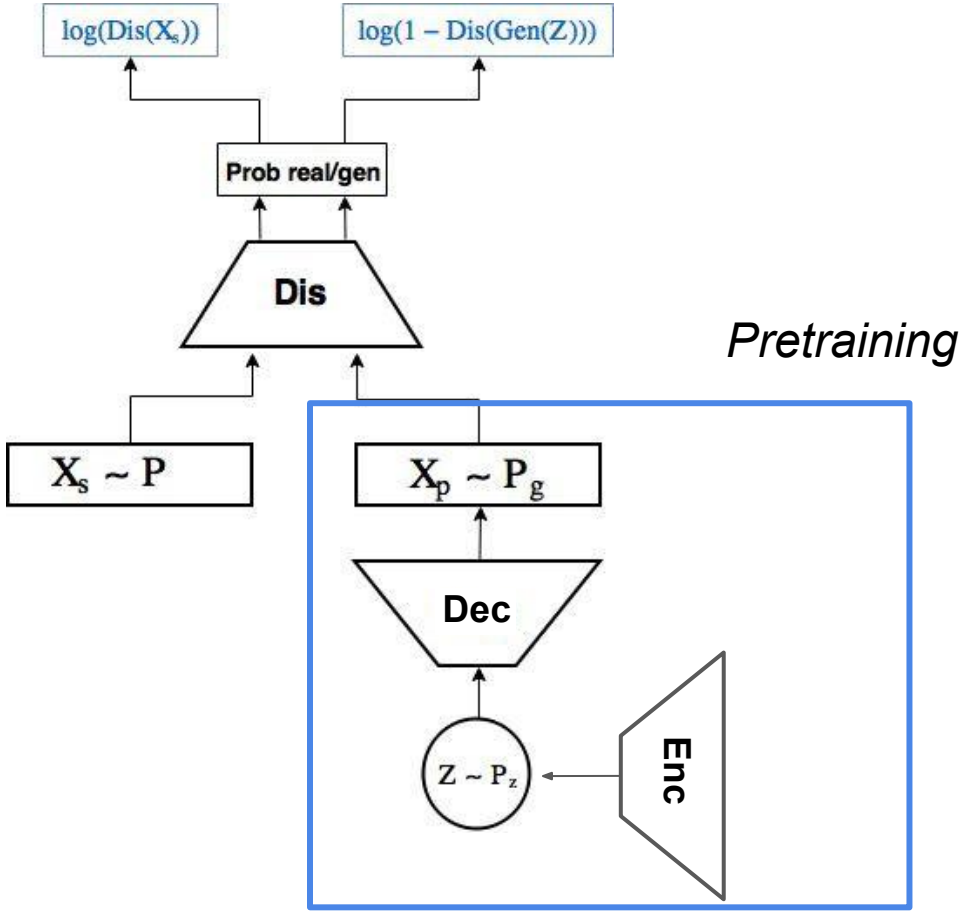


Image source: [Habr post on autoencoders and GANs]

VAE/GAN



Pretraining

Partial image source: [Habr post on autoencoders and GANs]

VAE/GAN original illustration

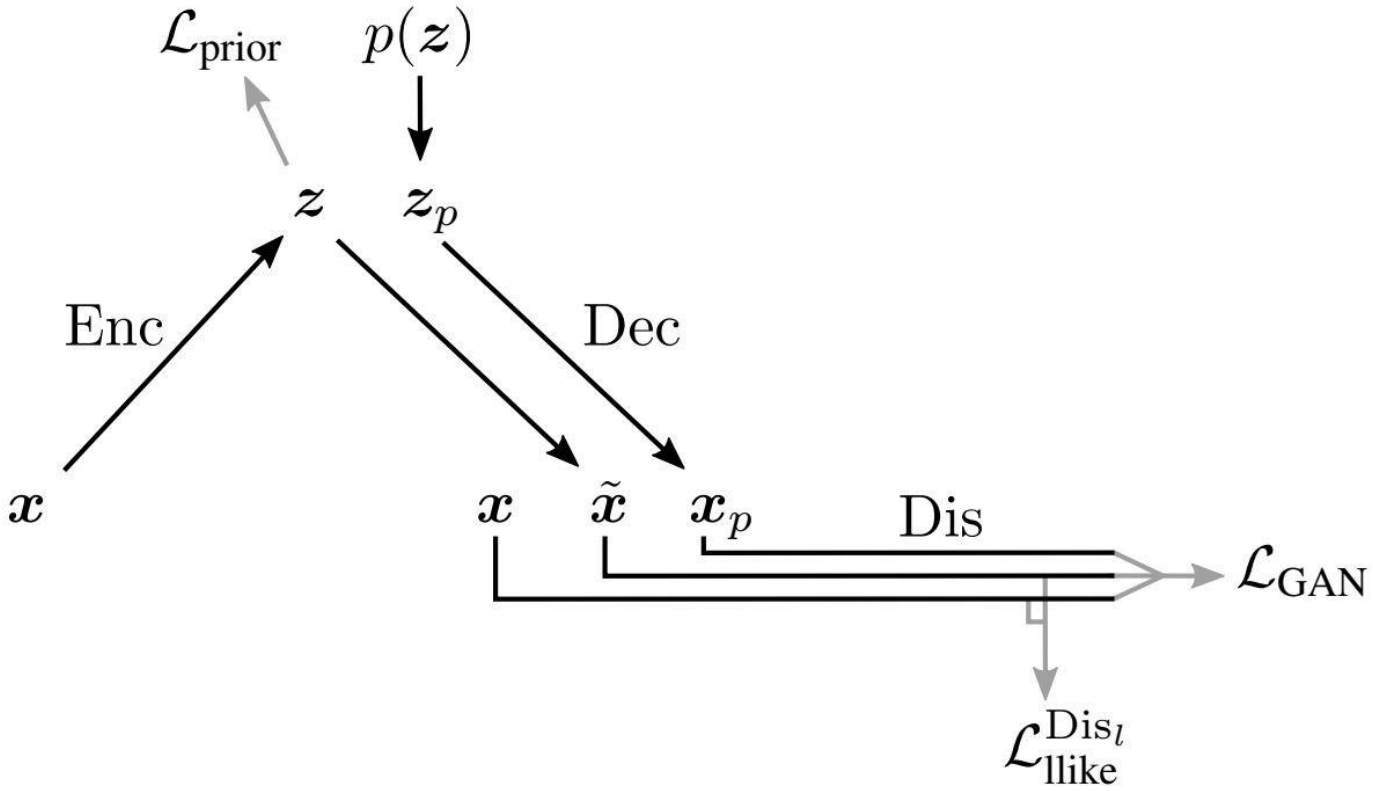


Image source: [Autoencoding beyond pixels using a learned similarity metric, Anders Boesen Lindbo Larsen et al, 2016]

Slides source

https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture19.pdf

https://web.eecs.umich.edu/~justincj/teaching/eecs498/WI2022/