# Generative Modeling

## How to Use Deep Neural Networks to Generate a Cat

Denis Derkach, Artem Ryzhikov, Maxim Artemev

Laboratory for methods of big data analysis

Spring 2022

# In this Lecture

- What’s a generative model.
- What it does.
- What are the main components.

# Generative Modeling

# This X Does Not Exist!

### This Person Does Not Exist

The site that started it all, with the name that says it all. Created using a style-based generative adversarial network (StyleGAN), this website had the tech community buzzing with excitement and intrigue and inspired many more sites.

Created by Phillip Wang.

### This Cat Does Not Exist

These purr-fect GAN-made cats will freshen your feeline-gs and make you wish you could reach through your screen and cuddle them. Once in a while the cats have visual deformities due to imperfections in the model – beware, they can cause nightmares.

Created by Ryan Hoover.

### This Rental Does Not Exist

Why bother trying to look for the perfect home when you can create one instead? Just find a listing you like, buy some land, build it, and then enjoy the rest of your life.

Created by Christopher Schmidt.

https://thisxdoesnotexist.com/

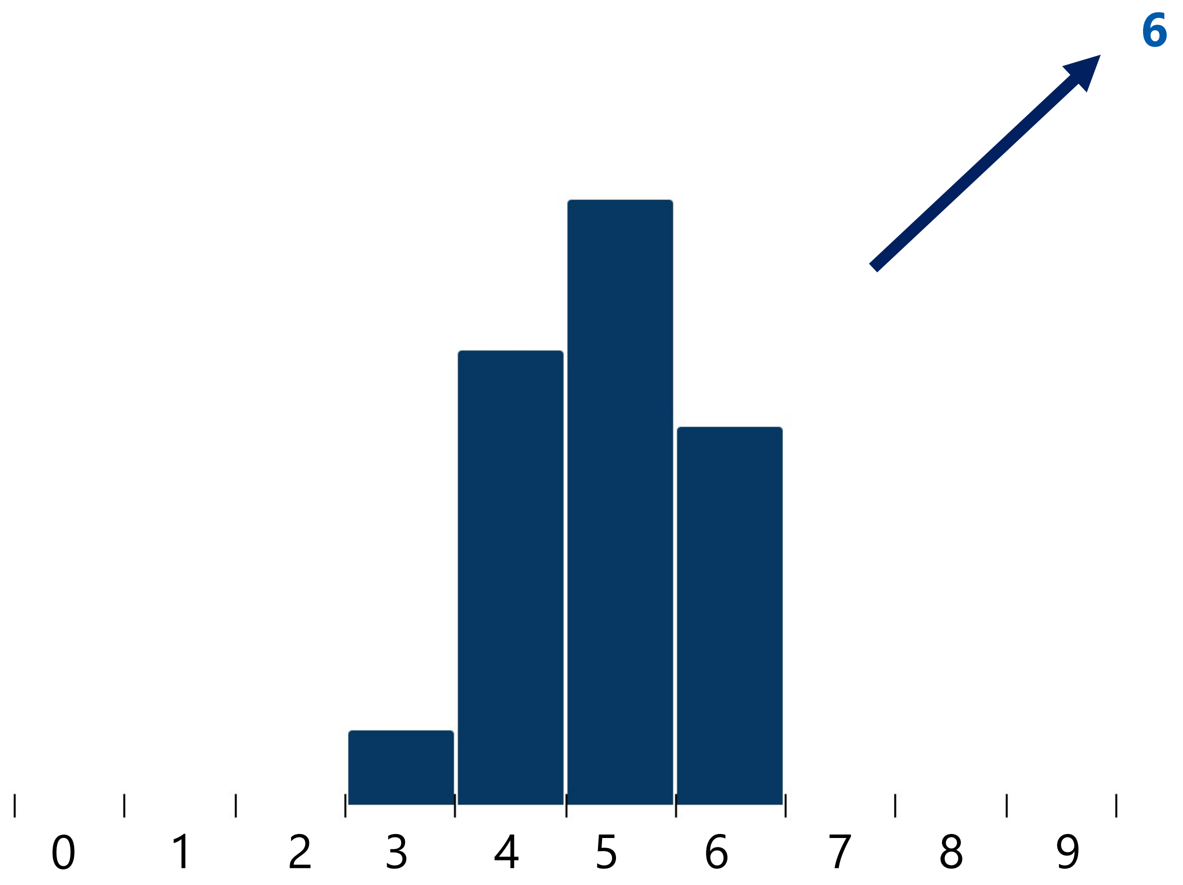
# What is Generative Modeling

# Random Number Generation

- We have sample with numbers:

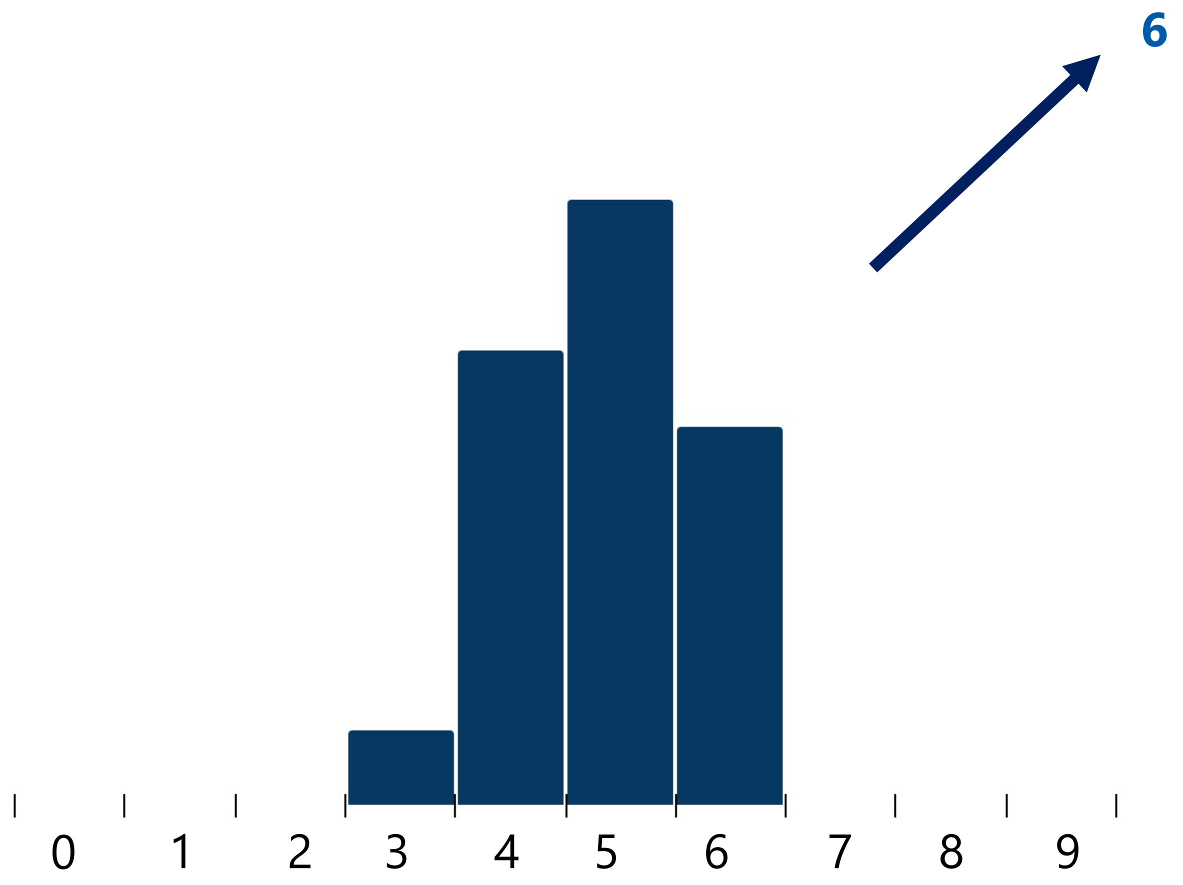
  **3; 5; 4; 4; 4; 4; 5 ; 6 ; 5 ; 4 ; 5; 4; 5; 6; 5; 6; 5; 5; 6; 6**

- Want to create a new number alike.



# How we did it?

- Assume there is a probability density $p_{true}(x)$.
- Try to estimate $p_{true}(x)$ using data and obtain $p_{data}(x)$.
- Sample from $p_{data}(x)$.

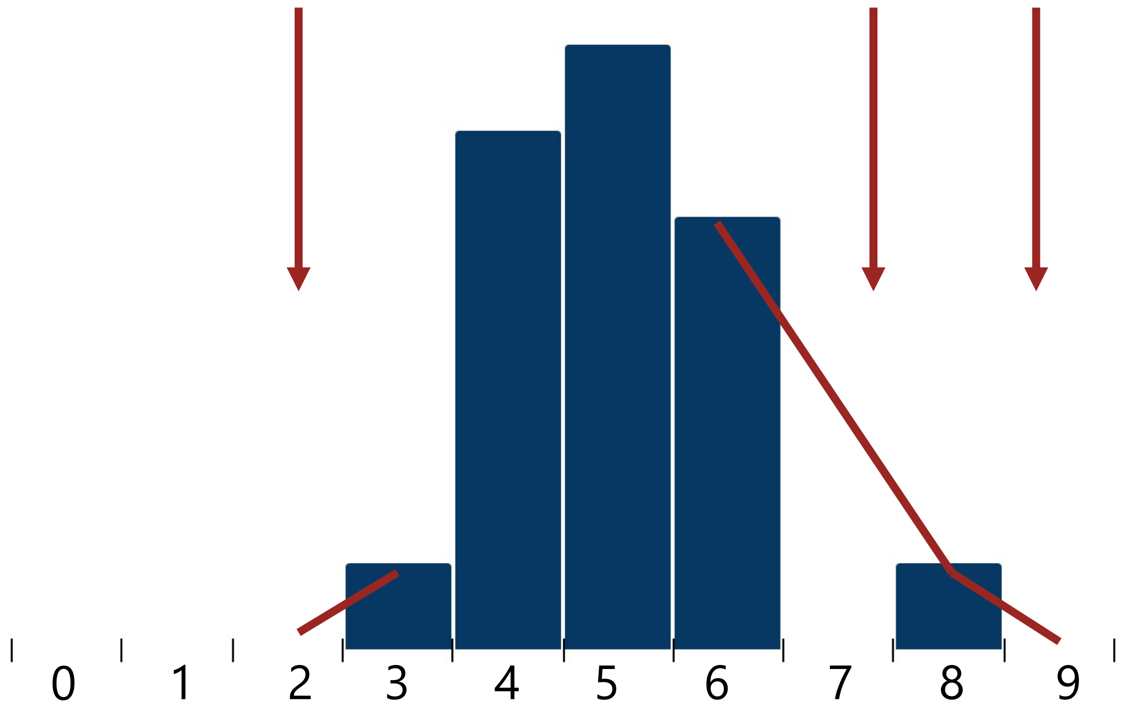


# Random Number Generation

- We have **different** sample with numbers:

  **3; 5; 4; 4; 4; 4; 5 ; 6 ; 8 ; 4 ; 5; 4; 5; 6; 5; 6; 5; 5; 6; 5**

- Want to create a new number alike.

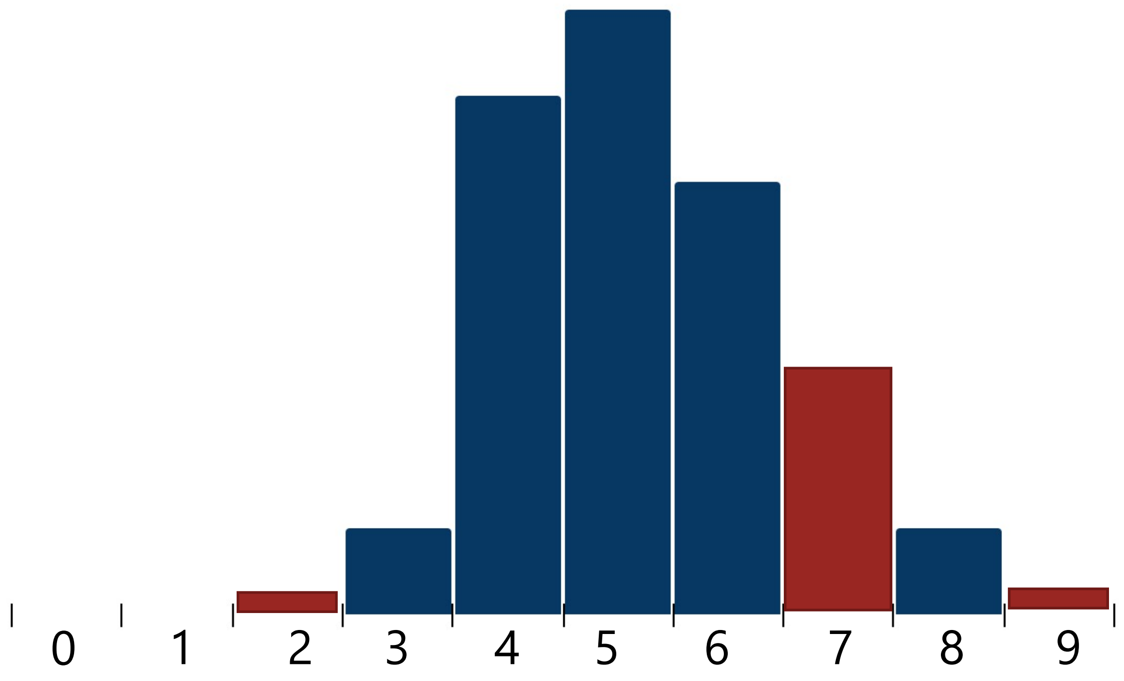


# Random Number Generation

- We have **different** sample with numbers:

  **3; 5; 4; 4; 4; 4; 5 ; 6 ; 8 ; 4 ; 5; 4; 5; 6; 5; 6; 5; 5; 6; 5**

- Want to create a new number alike.

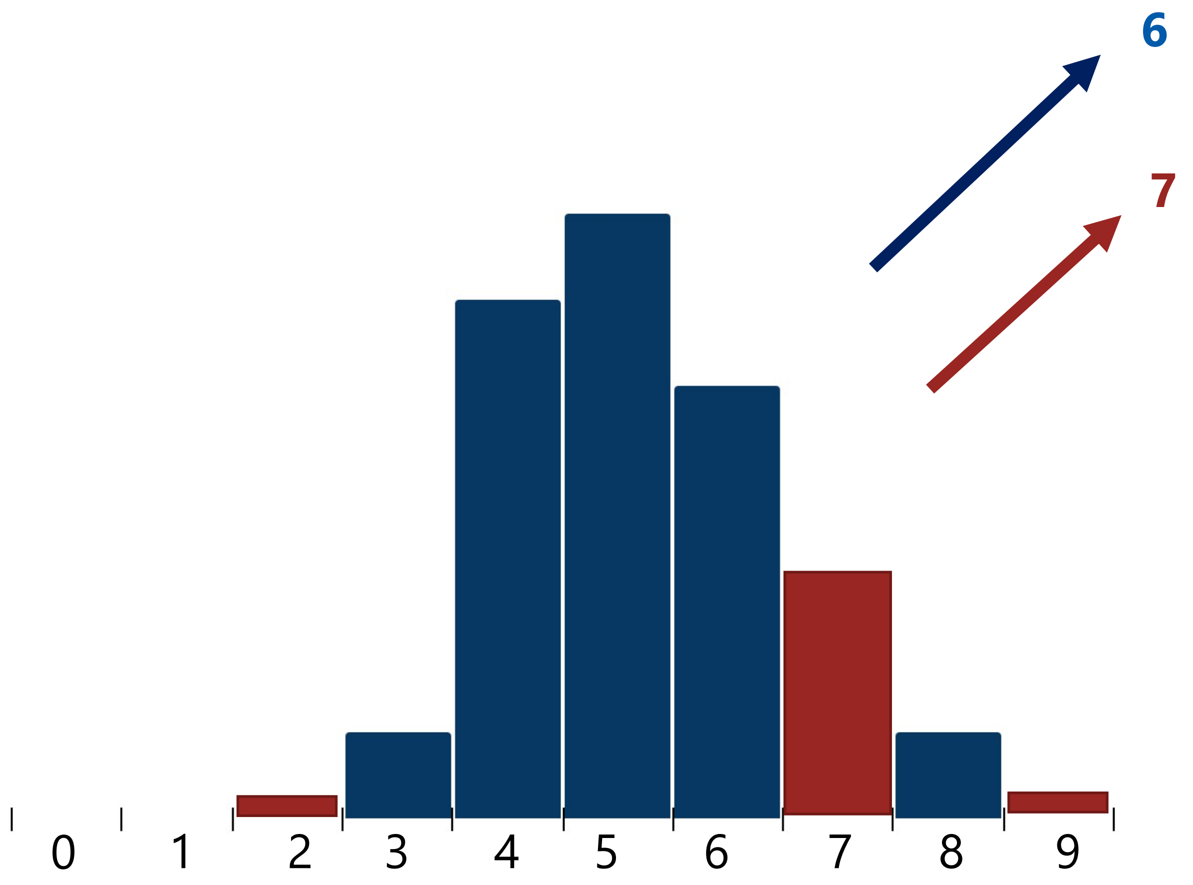


# Random Number Generation

- We have **different** sample with numbers:

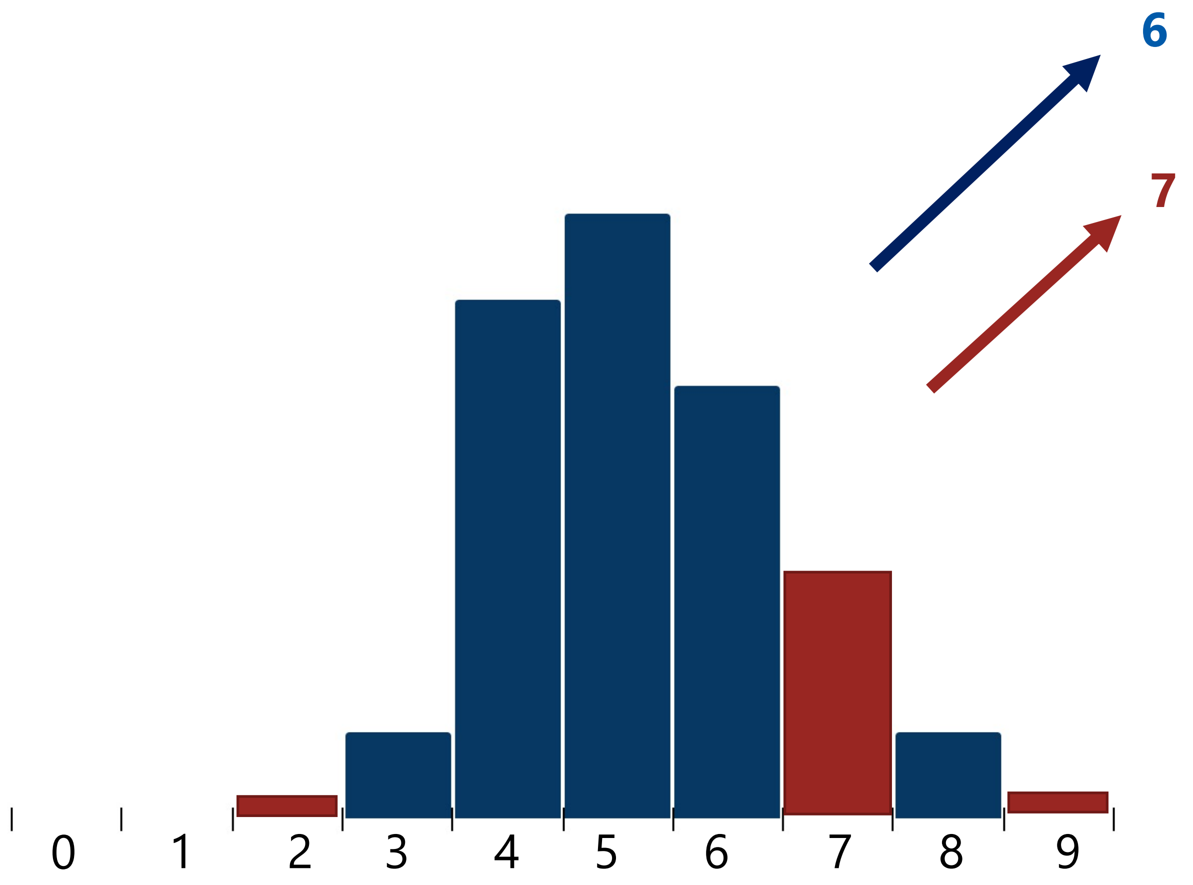
  **3; 5; 4; 4; 4; 4; 5 ; 6 ; 8 ; 4 ; 5; 4; 5; 6; 5; 6; 5; 5; 6; 5**

- Want to create a new number alike.



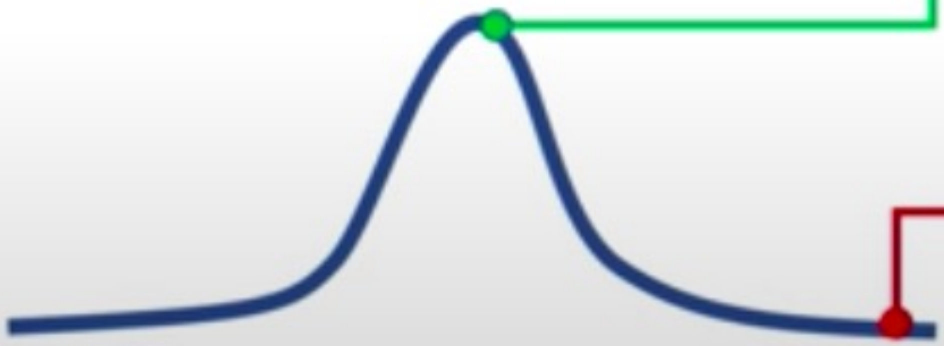
# Random Number Generation

- Assume there is a probability density $p_{true}(x)$.
- **Choose interpolation model.**
- Try to estimate $p_{true}(x)$ using data and obtain $p_{data}(x)$.
- Sample from $p_{data}(x)$.



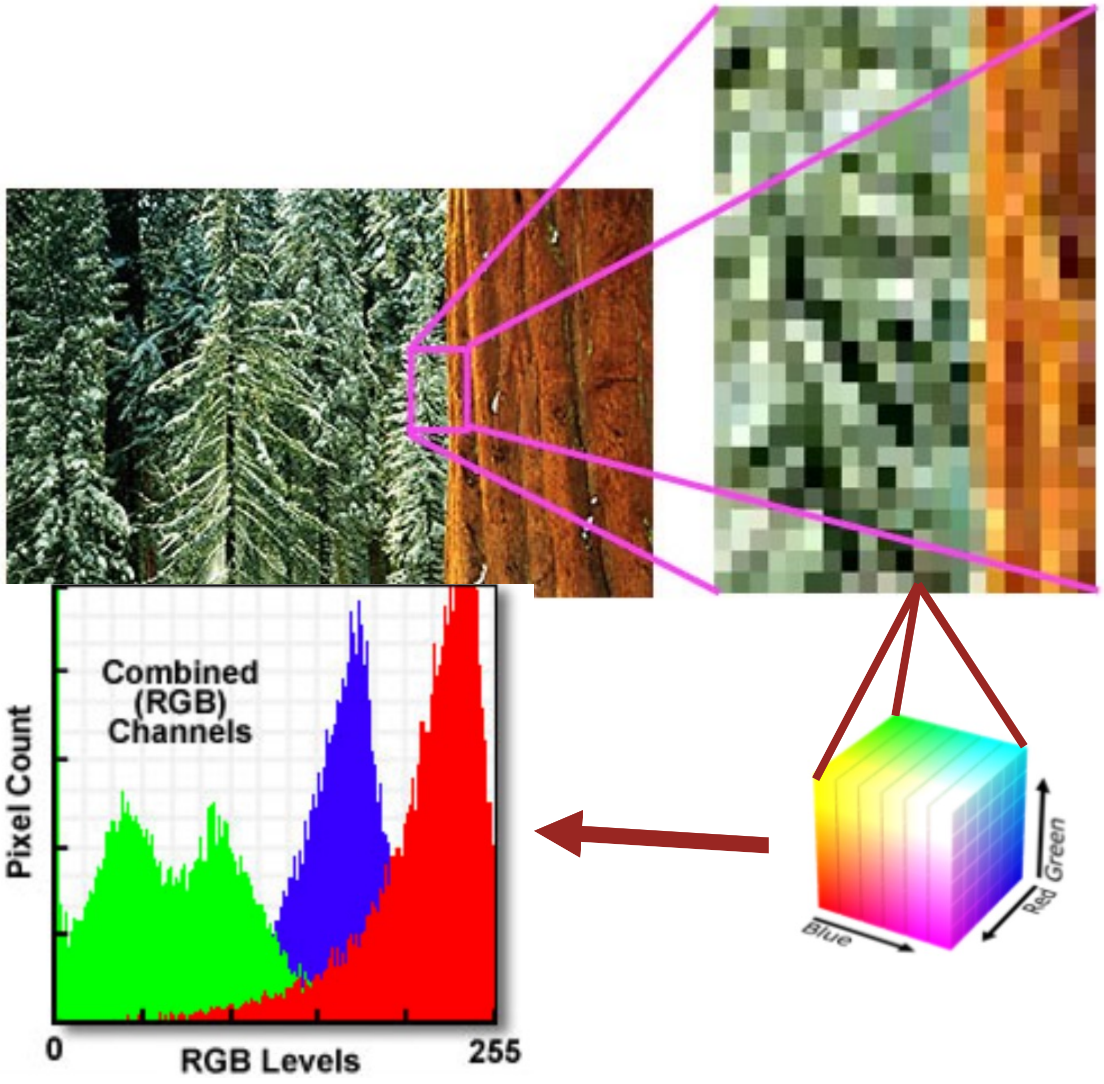
# Case Study: Anomaly Detection

- **Problem:** How can we detect when we encounter something new or rare?
- **Strategy:** Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more!




http://introtodeeplearning.com/

# More Complicated Case: Figures




- Figure consists of pixels.
- One can use this representation.
- Each pixel is encoded by 3 colours.
- **Multi-modal distribution**.
- **Multidimensional problem**.

# Number of Parameters

- Handwritten digits dataset.
- Only black and white pixels.
- Number of pixels 28X28.
- Number of possible states:

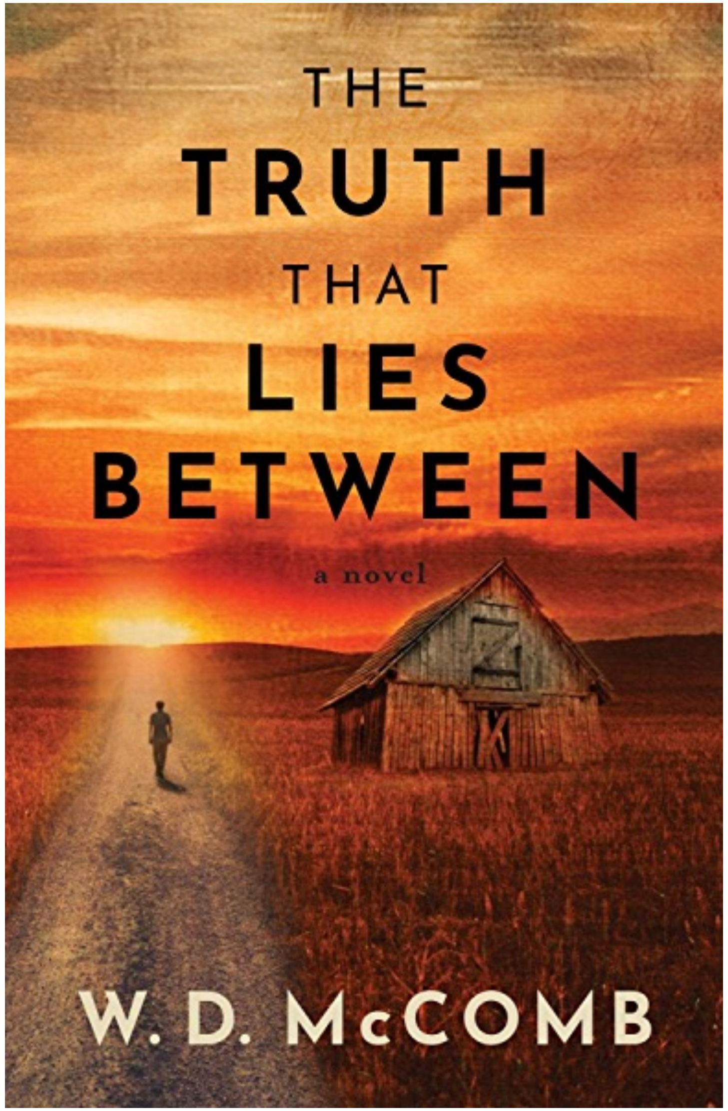
  $2x2x2x....x2 = 2^n.$

- **Number of parameters:**

  $\mathbf{2^n - 1.}$

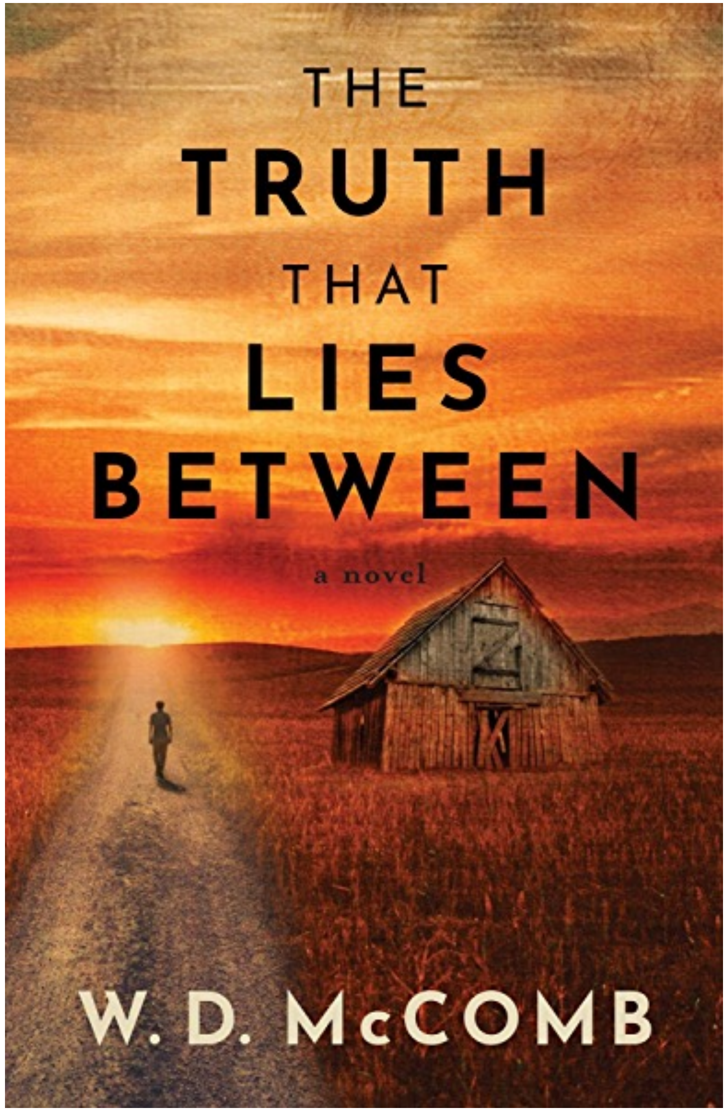
- **For Independent pixels:**

  **n.**



# Generative model: Final Touch

- Assume there is a probability density $p_{true}(x)$.
- Choose interpolation model.
- **Reduce number of dimensions.**
- Try to estimate $p_{true}(x)$ using data and obtain $p_{data}(x)$.
- Sample from $p_{data}(x)$.




Amazon

# Generative model: Problem Statement

Three major tasks, given a generative model $f$ from a class of models $\mathcal{F}$ :

- **Estimation**: find the $f$ in $\mathcal{F}$ that best matches observed data.
- **Evaluate Likelihood**: compute $f(z)$ for a given $z$.
- **Sampling**: drawing from $f$.

S. Nowozin et al. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization

# Generative model vs Discriminative model

**Discriminative models**

- learn $\mathbb{P}(y|x)$
- Directly characterizes the decision boundary between classes only
- Examples: Logistic Regression, SVM, etc

**Generative models**

- learn $\mathbb{P}(x|y)$ (and eventually $\mathbb{P}(y,x)$)
- Characterize how data is generated (distribution of individual class)
- Examples: Naive Bayes, HMM, etc.

https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

# Chapter outcome

- Generative modeling is a distinct task in machine learning.
- Mathematically, it aims to reconstruct the probability density, from which the given dataset was sampled.

# Early Generative Models

# First ideas

For parametric model.

- **Inversion sampling**. For $x$ with CDF $F_X(x)$ :

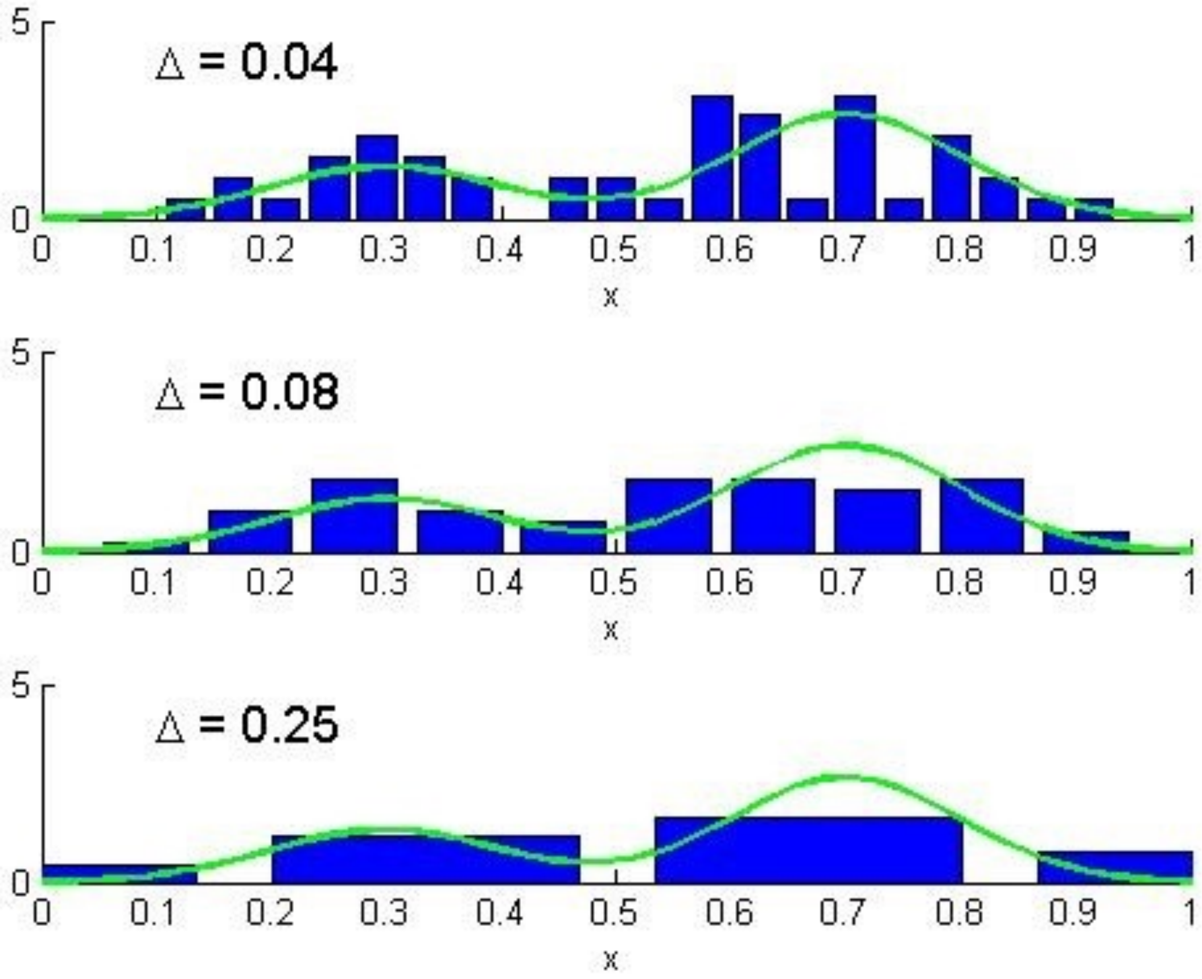
$$z \sim \mathrm{Unif}(0; 1);\ x = F_X^{-1}(z).$$

- **Works in multidimensions**. Sample successively.
  - Generate $X$ from the marginal $p_X(x) = \int p_{X,Y}(x, y)\mathrm{d}y$.
  - Generate $Y$ given $X = x$ from the conditional $p_{Y|X}(y|x) = \frac{p_{x,y}(x,y)}{p_X(x)}$.

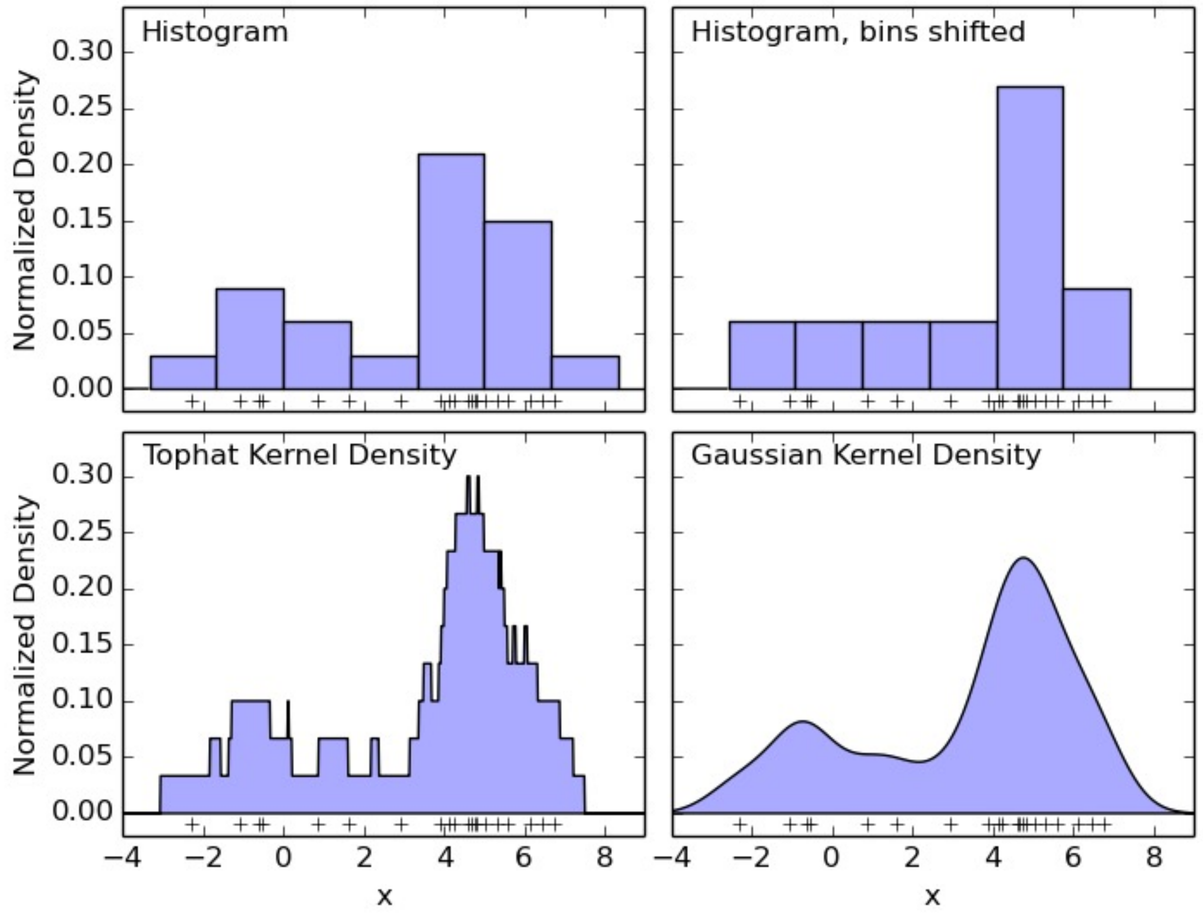
For 1D Gaussian model, the convergence is $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.

# “Non-parametric” Approaches

- Histograms can be used.
- Need to choose optimal bin size.
- Smaller bins for approximate constant estimate.
- Larger bins for less fluctuations.
- Can be chosen using empirical risk.



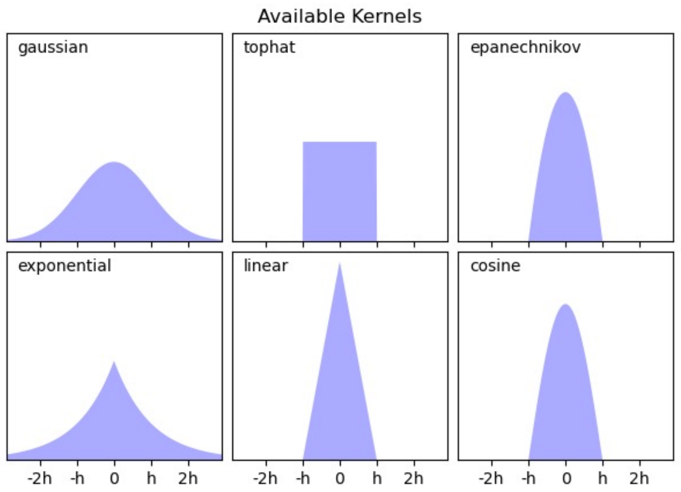
# Kernel-density estimation




- Assign **every** event a weight.
- Smooth between events.
- Kernel Density Estimation:

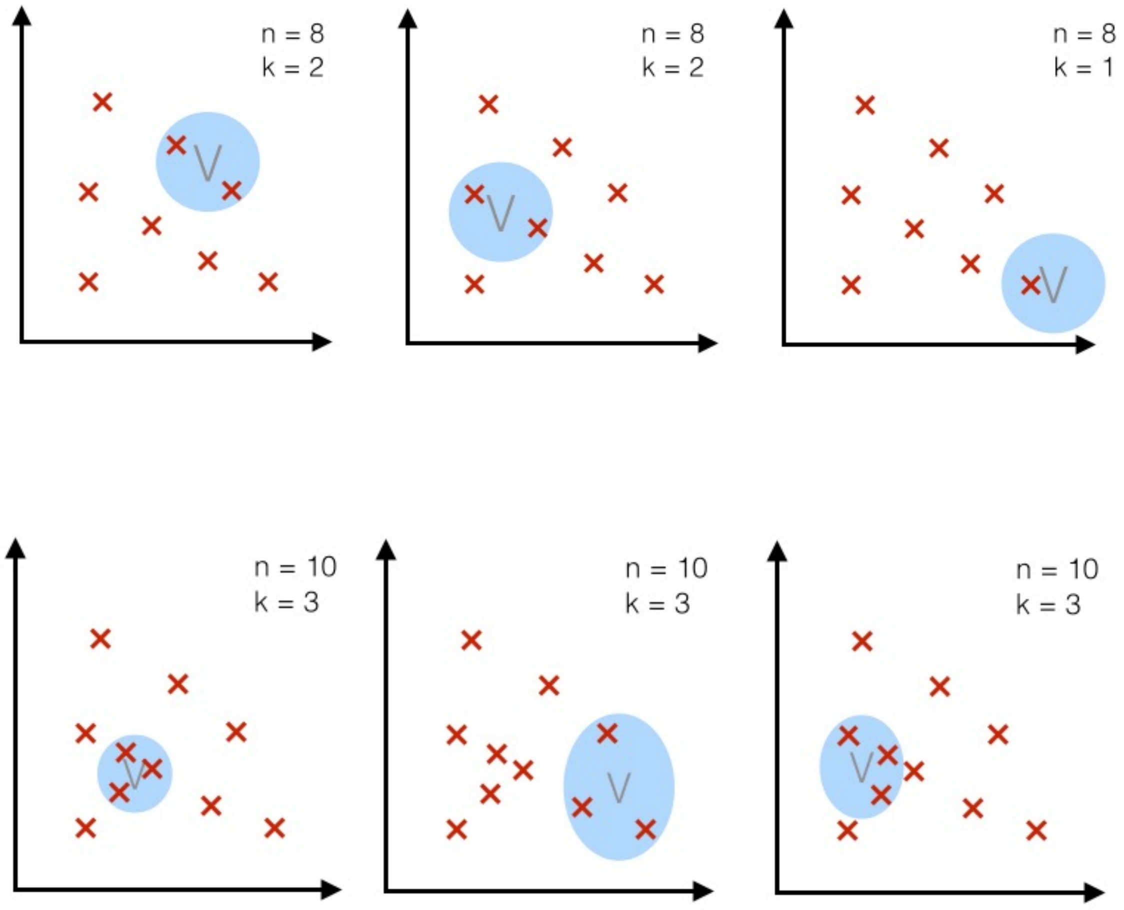
$$\hat{p}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

$K$ – some kernel, h – bandwidth.




Sci-kit

# KDE2KNN




- With fixed volume kernel outliers can lead to fluctuations in $\hat{p}(x)$.
- Vary kernel volume to cover k nearest neighbors.
- Better coverage of tails.

S. Raschka's blog

# KDE and kNN Summary

- Efficient in low dimensional estimation.
- Controllable convergence rate for bias or variance but the overall rate is similar.
- To speed up the convergence, once can attempt to find manifolds in the $d$-dimension.
- Fairly hard to sample and keep the model in memory.

| Type | Method | Convergence rate | Tuning parameter | Limitation |
|---|---|---|---|---|
| Parametric | Parametric model | $O\left(\frac{1}{\sqrt{n}}\right)$ | None | Unavoidable bias |
| | Mixture model | $O\left(\frac{1}{\sqrt{n}}\right)$ | $K$, number of mixture | Hard to compute |
| Nonparametric | Histogram | $O\left(\frac{1}{n^{1/3}}\right)$ | $b$, bin size | Lower convergence rate |
| | Kernel density estimator | $O\left(\frac{1}{n^{2/5}}\right)$ | $h$, smoothing bandwidth | |
| | K-nearest neighbor | $O\left(\frac{1}{n^{2/5}}\right)$ | $k$, number of neighbor | |
| | Basis approach | $O\left(\frac{1}{n^{2/5}}\right)$ | $M$, number of basis | |

see for example Yen Chi Chen, Learning Theory, Lec 8.

# Regressive Models

# Neural CDF Regression

For parametric model.

- **Inversion sampling**. For $x$ with CDF $F_X(x)$ :

$$z \sim \mathrm{Unif}(0;1);\; x \;=\; F_X^{-1}(z).$$

- **Idea**: use neural network to fit CDF.

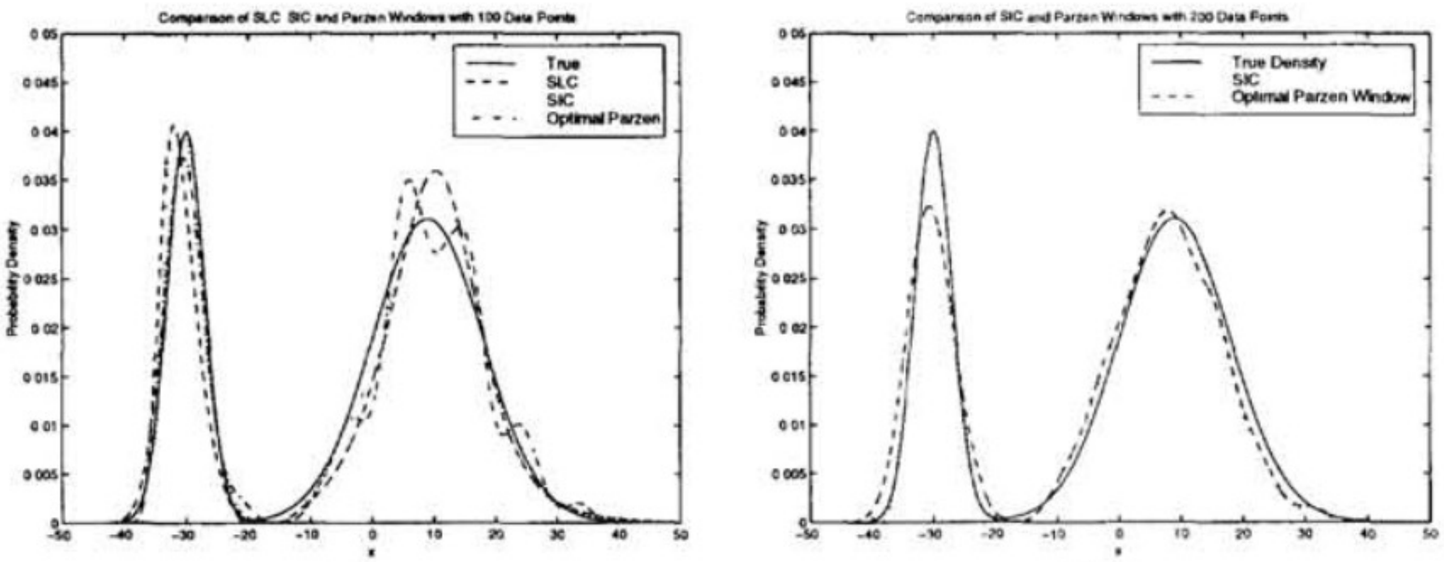
# Stochastic Learning of the Cumulative

- Let $x_1 \leq x_2 \leq \cdots \leq x_N \in \mathbb{R}$ be the data points with PDF $g(x)$ and corresponding CDF $G(x) = \int_{-\infty}^{x} g(x')dx'; G(x) \sim U[0;1]$.
- We want Neural Network: $H(x, w) = G(x)$.
- Take: random ranked variable: $u_1 \leq u_2 \leq \cdots \leq u_N \sim U[0;1]$.
- Loss:

$$L = \underbrace{\sum_{n=1}^{N} (H(x_n) - u_n)^2}_{\text{Description}} +$$

$$\underbrace{+\lambda \sum_{h=1}^{N_h} \Theta(H(y_k) - H(y_k + \Delta))(H(y_k) - H(y_k + \Delta))^2}_{\text{Monotonicity}}$$

# SLC: Discussion




(a) (b)

- Faster convergence wrt kernel methods: $\mathcal{O}\left(\frac{\log \log n}{n}\right)$.
- Multi-d densities need special treatment.
- Allows for easier sampling.

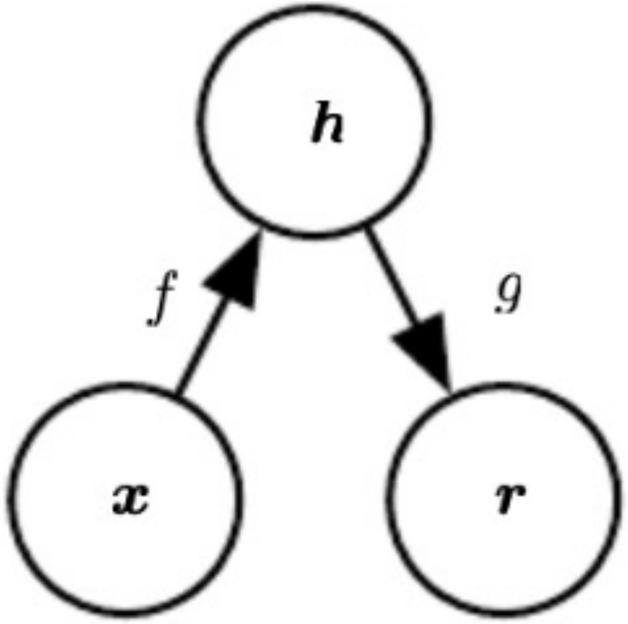
M. Magdon-Ismail et al., Neural Networks for Density Estimation, NIPS 98

# Autoencoders

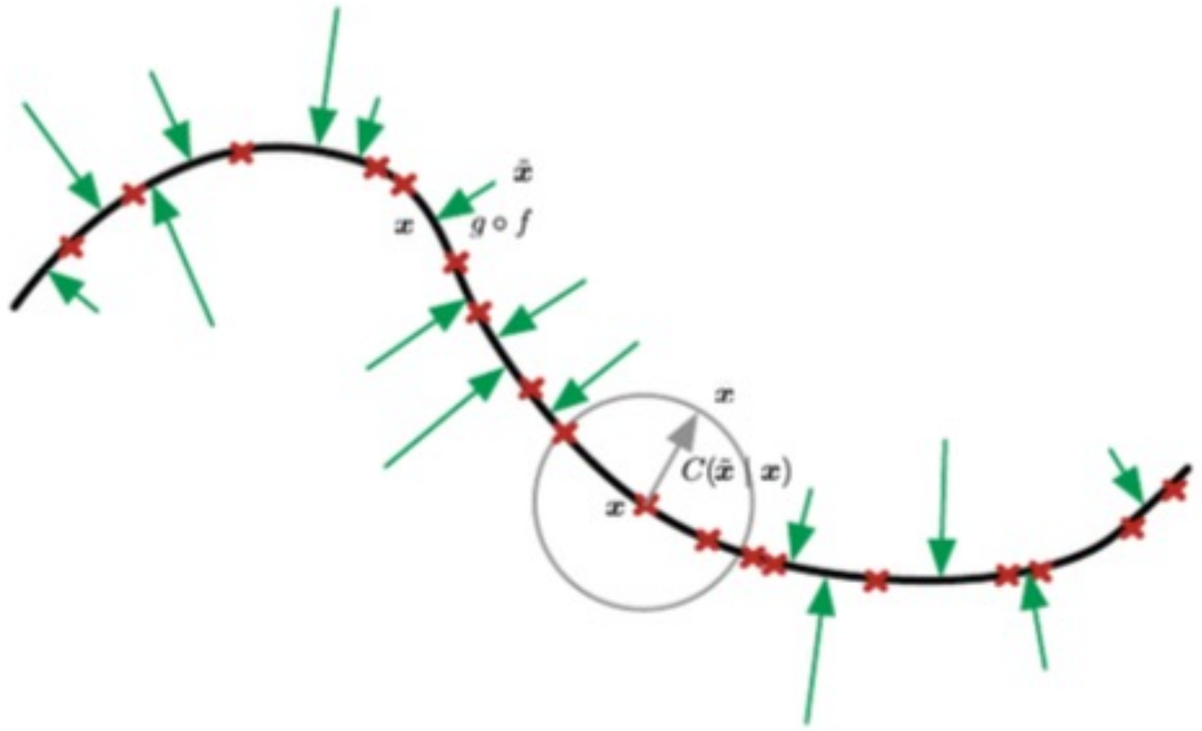
# Rationale

- Need a trick to reduce number of dimensions.
- Simplest idea: have a network with narrow hidden layer – **Autoencoder**!
  - Encoder: $h = f(x)$;
  - Decoder: $r = g(h)$.
- Want to find transformation

$$g\big(f(x)\big) = x.$$

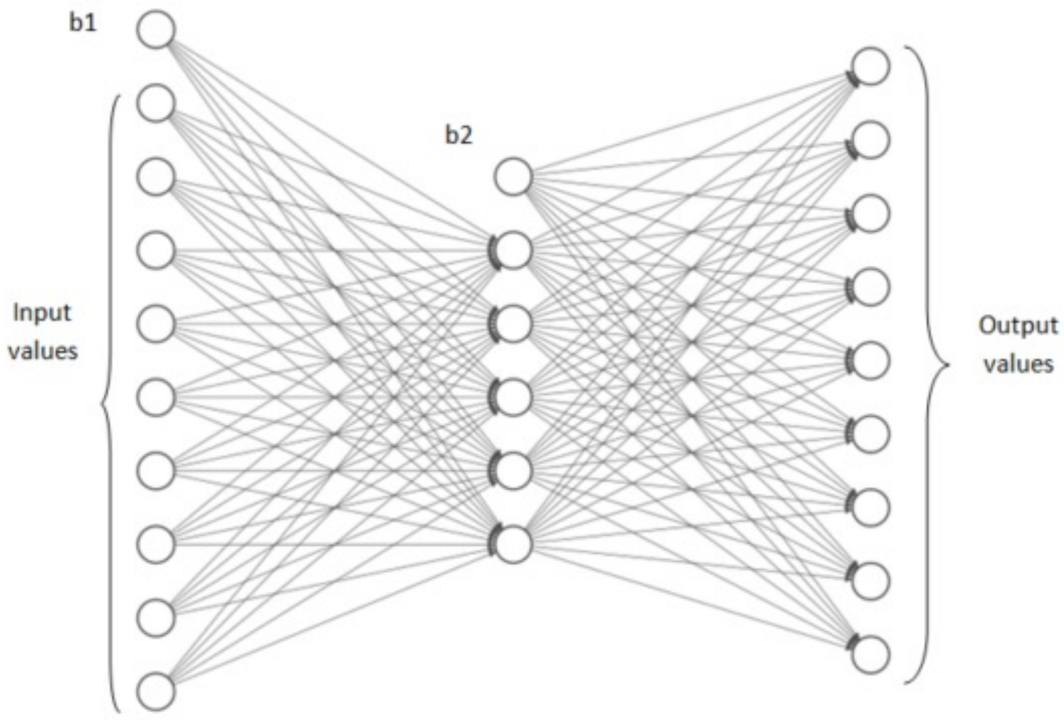


# Signal Manifold




- The true signal is always situated on a manifold inside the $R^D$ space.

# Vanilla Autoencoder




- For binary data $x \in R^D$ and a one-layer network:

$$h(x) = g(b + Wx);$$
$$\hat{x} = \sigma\big(c + Vh(x)\big).$$

- A typical loss is cross-entropy:

$$\mathcal{L}(x) = \sum -x_d \log(\hat{x}_d) - (1 - x_d)\log(1 - \hat{x}_d)$$

- Interpret $\mathcal{P}(x_d = 1) = \hat{x}_d$, and cross-entropy as negative log-probability.
- Does not produce a PDF in the end.

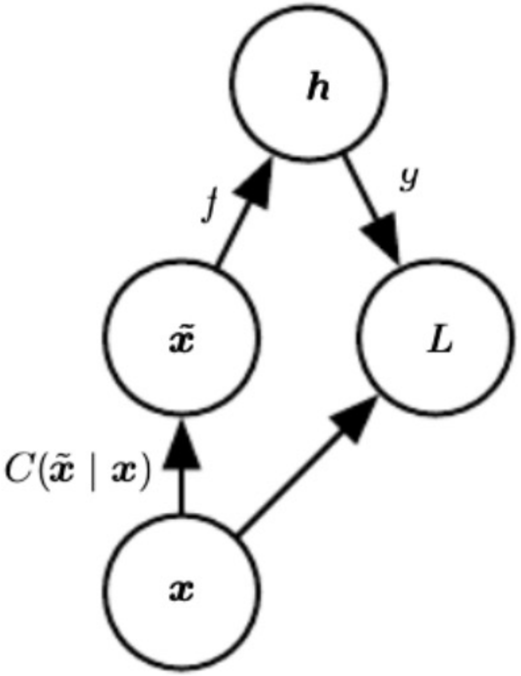
# Denoising Autoencoders as Generative model

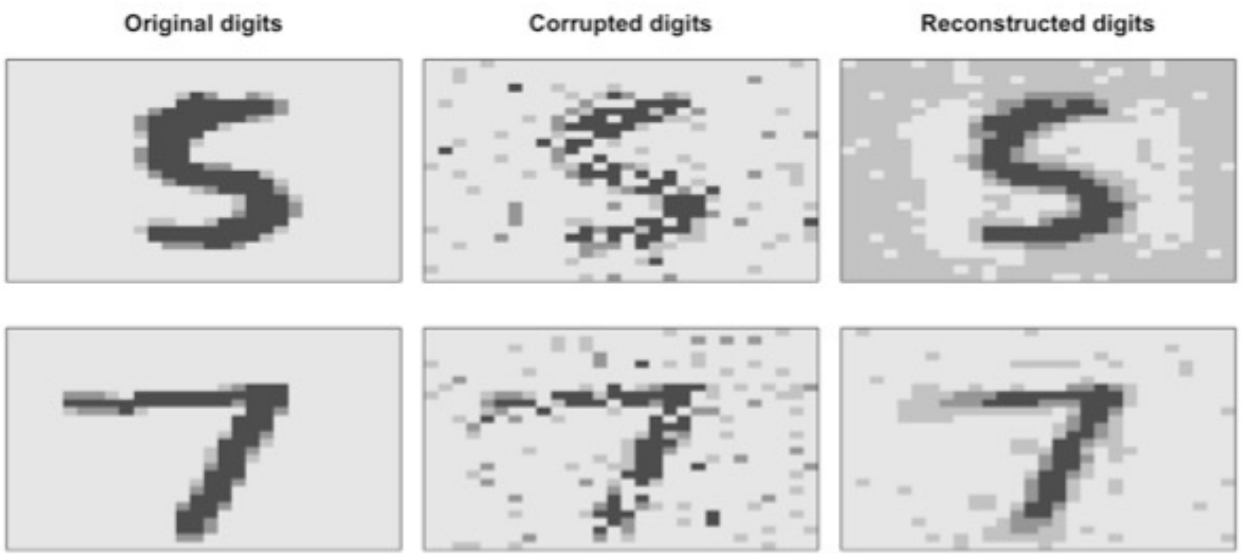
- Artificially add noise $\tilde{x} \sim C(\tilde{x}|x)$.
- Use

$$\mathcal{L}(\theta) = -E\left[\log P_\theta\left(X\middle|\tilde{X}\right)\right]$$

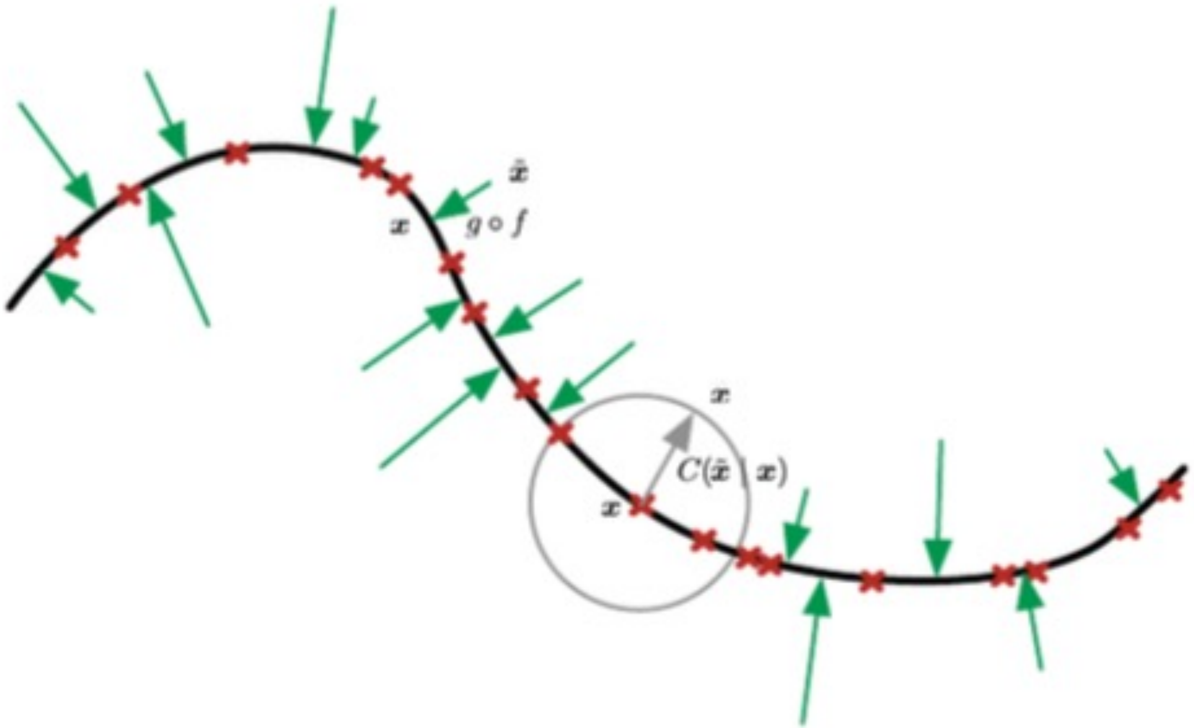
where expectation is taken over the joint data-generating distribution:

$$P(X,\tilde{X}) = P(X)C(\tilde{X}|X).$$



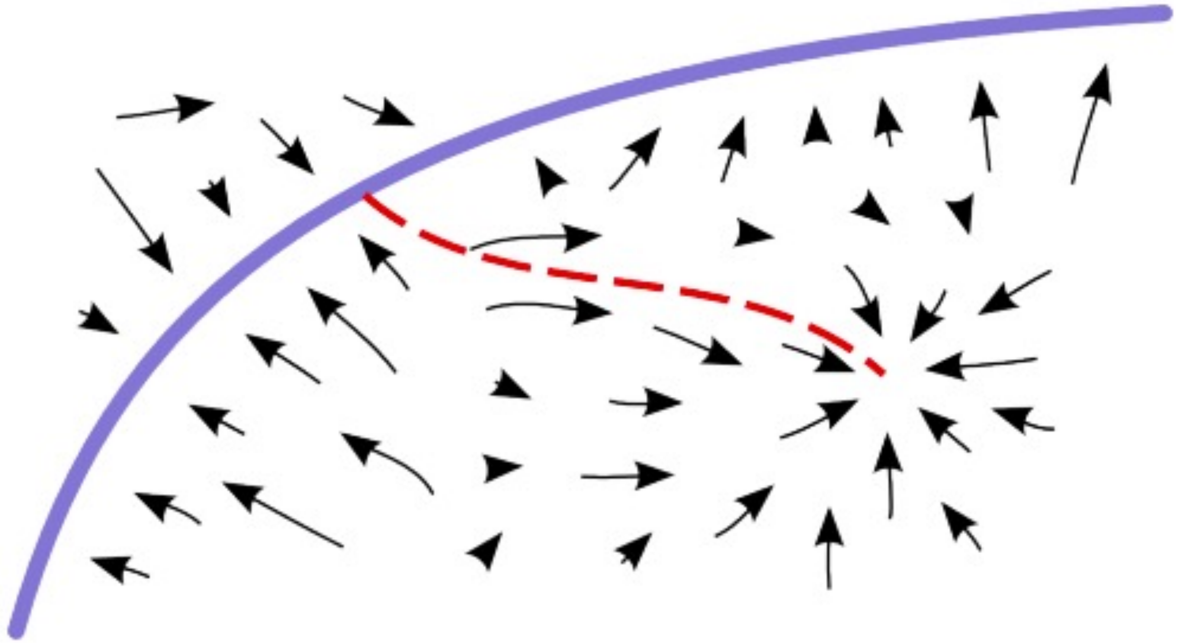



# Signal Manifold




- The true signal is always situated on a manifold inside the $R^D$ space.
- Denoising autoencoder is trained to map a corrupted data point $\tilde{x}$ back to the original data point $x$.

# Constructing Probabilistic Model



- Construct a Markov Chain:

$$X_t \sim P_\theta(X|\tilde{X}_{t-1});$$
$$X_t \sim C(\tilde{X}|X_t).$$

- Start with random point $X_0$ and sample new data points from it.
- Walk-back return algorithm
- For infinitesimal noise equivalent to KDE.

Y. Bengio et al., Generalized Denoising Auto-Encoders as Generative Models, NIPS-13

# MNIST results

Figure 4: *Successive samples generated by Markov chain associated with the trained DAEs according to the plain sampling scheme (left) and walkback sampling scheme (right). There are less "spurious" samples with the walkback algorithm.*

Y. Bengio et al., Generalized Denoising Auto-Encoders as Generative Models, NIPS-13

# AE Summary

- Autoencoders are able to find hidden representations of the observables.
- Vanilla Autoencoders are not suitable for generative modeling.
- Autoencoders can produce a sampling model with implicit PDF reconstruction.

# Final Summary

- Generative modeling is a distinct task of machine learning.
- Several pre-deep learning algorithms can produce reasonable results in the low dimensional data.
- Denoising Autoencoder is one of the first pseudo-generative models.