Intro to ML

Dr. Sci., Prof. **Evgeny Burnaev**

Head of Applied AI Center, Skoltech
Head of Research Group, AIRI

Skoltech

Outline

- **Motivation**
- Supervised Learning
- Neural Networks
- Unsupervised Learning and Generative Modeling
- What’s next

Artificial Intelligence

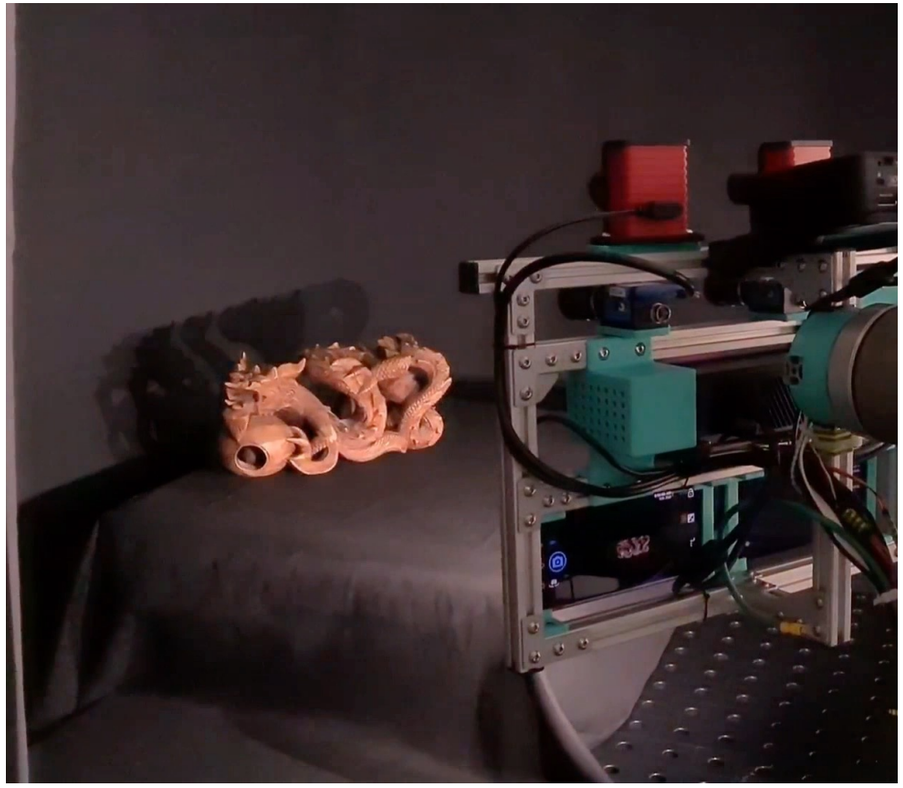
★ What is Artificial Intelligence ?

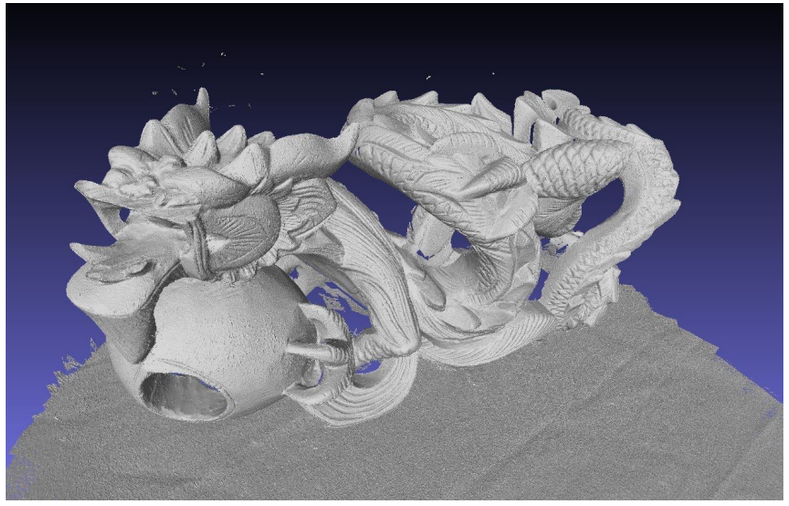
Why everyone is talking about AI now: sometimes it works better than a human

In image classification task

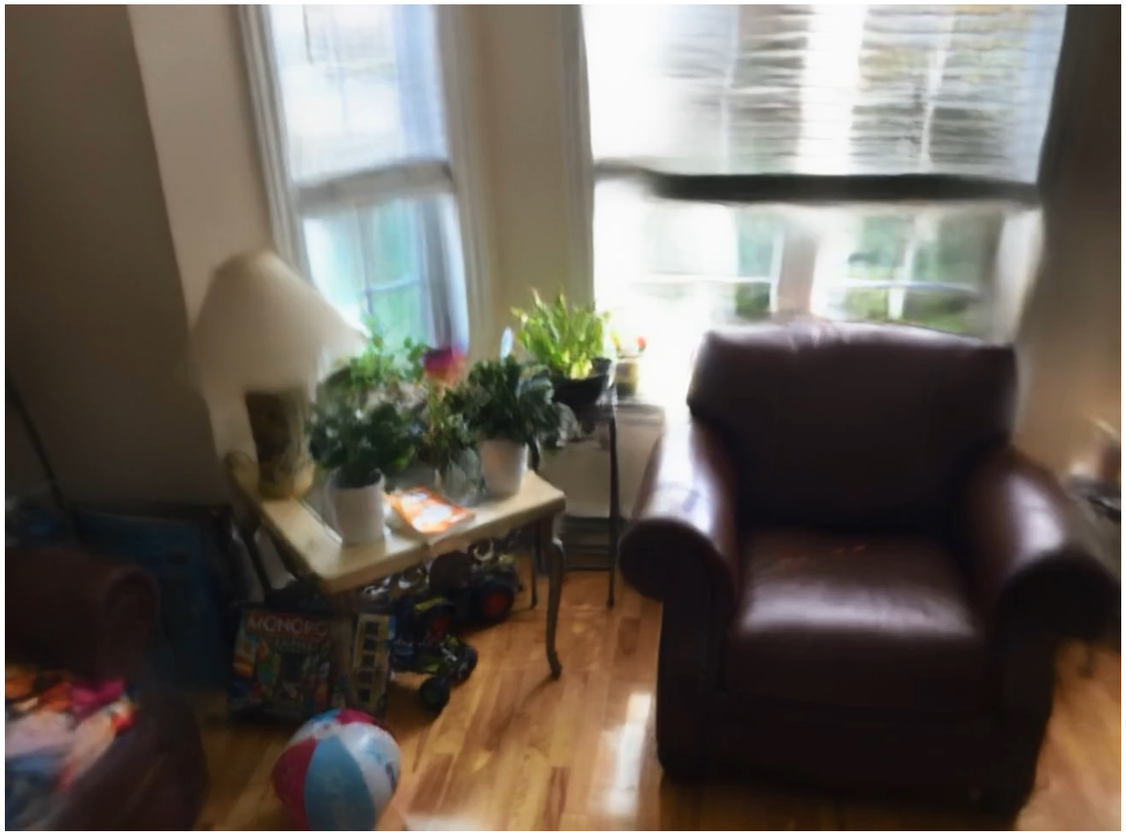
- human error is **5.1%**
- AI error is **3.57%**

Digital twins





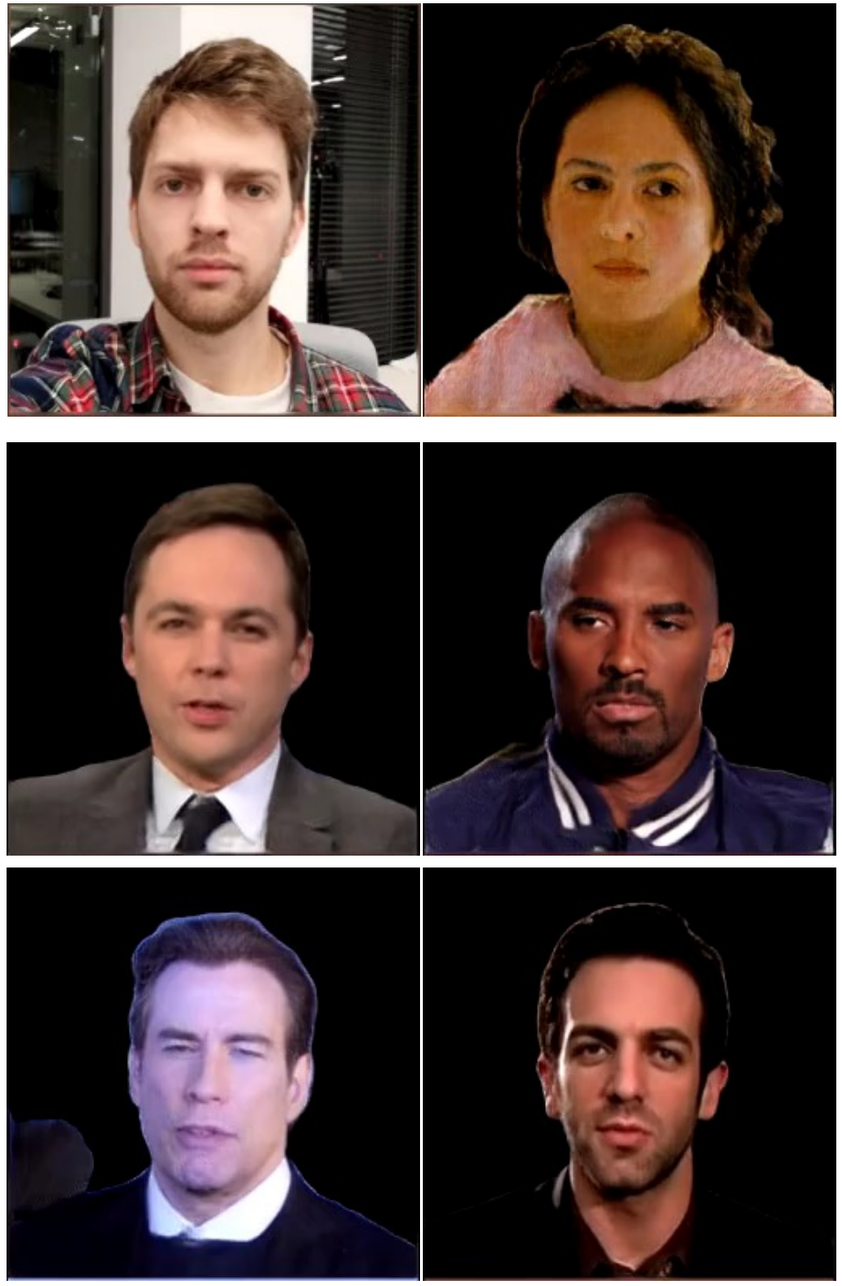
Neuro-rendering



Modeling 3D scenes from images with high realism

[Aliev et al. ECCV 2020]

Neuro avatars

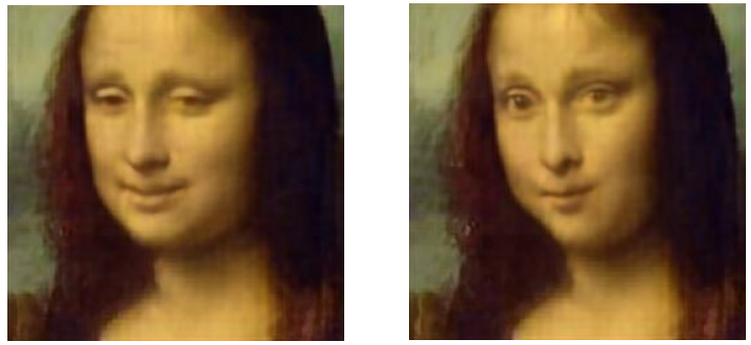


[Burkov et al. CVPR2020]

[Grigorev et al. CVPR2021]

Human image synthesis and deepfakes

[Karras et al. ICLR 2018]



[Zakharov et al. ICCV 2019]

General structure

- ARTIFICIAL INTELLIGENCE

AI is the broadest term, applying to any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning)

- MACHINE LEARNING

The subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience. The category includes deep learning

- DEEP LEARNING

The subset of machine learning composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data

Data Analysis

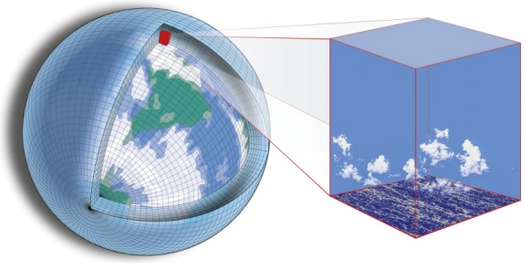
Data Mining. ML algorithms are often used for pattern mining and extraction

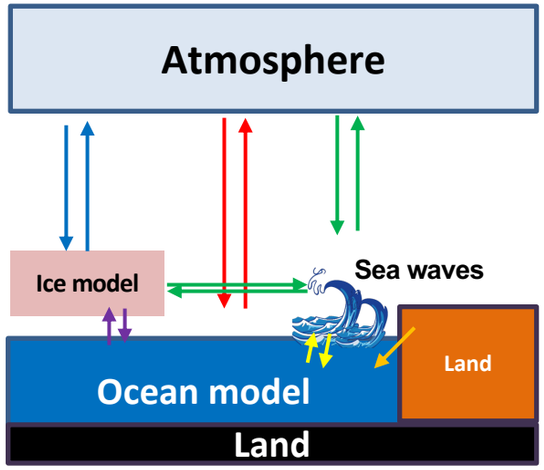
Sea Ice Regional Forecasting

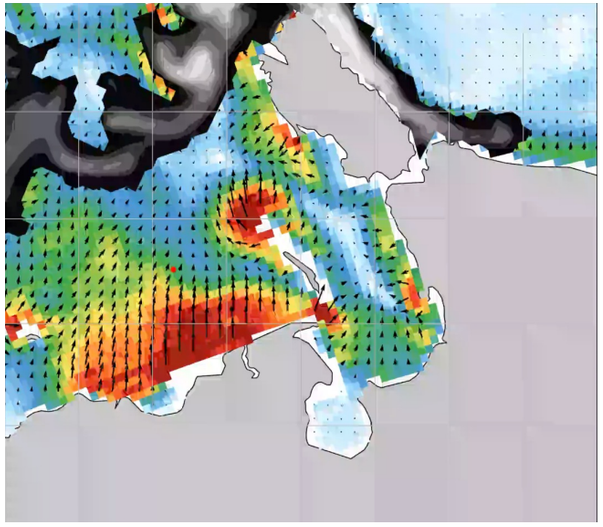
Challenge: year-round navigation along the Northern Sea Route requires reliable and efficient navigation systems

Subproblems:
- ✓ weather forecasting
- ✓ sea current forecasting
- ✓ sea ice forecasting
- ✓ multi-agent system for navigation and optimization of ship logistics in the Arctic

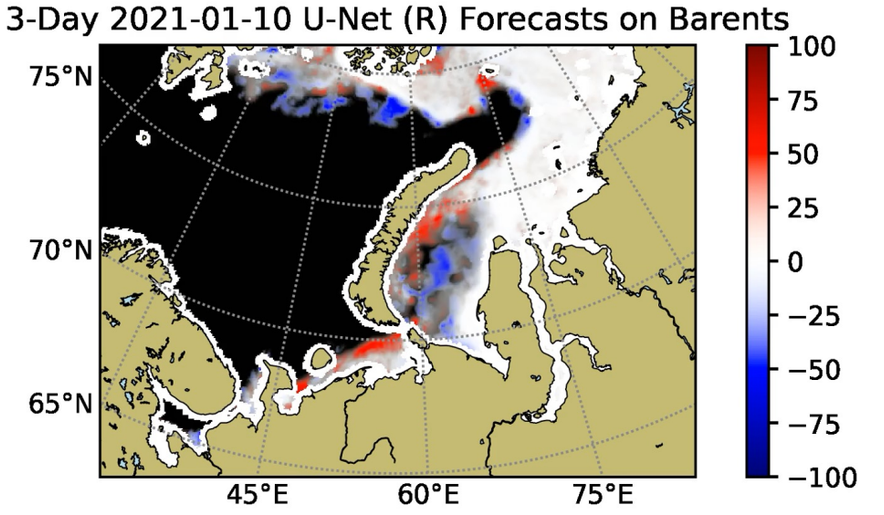
Technologies:
- ✓ Operational data collection system (remote sensing, weather stations, buoys, etc.)
- ✓ Ocean physical model
- ✓ AI for
  - ➢ Fast and accurate forecasting
  - ➢ Assimilation of measurement data (remote sensing, ships, buoys, weather stations)
  - ➢ Accounting for risks and uncertainties
  - ➢ Simulation of subgrid processes
- ✓ Final coupling of phys. models and data-driven models for accurate and reliable prediction of sea currents and sea ice conditions







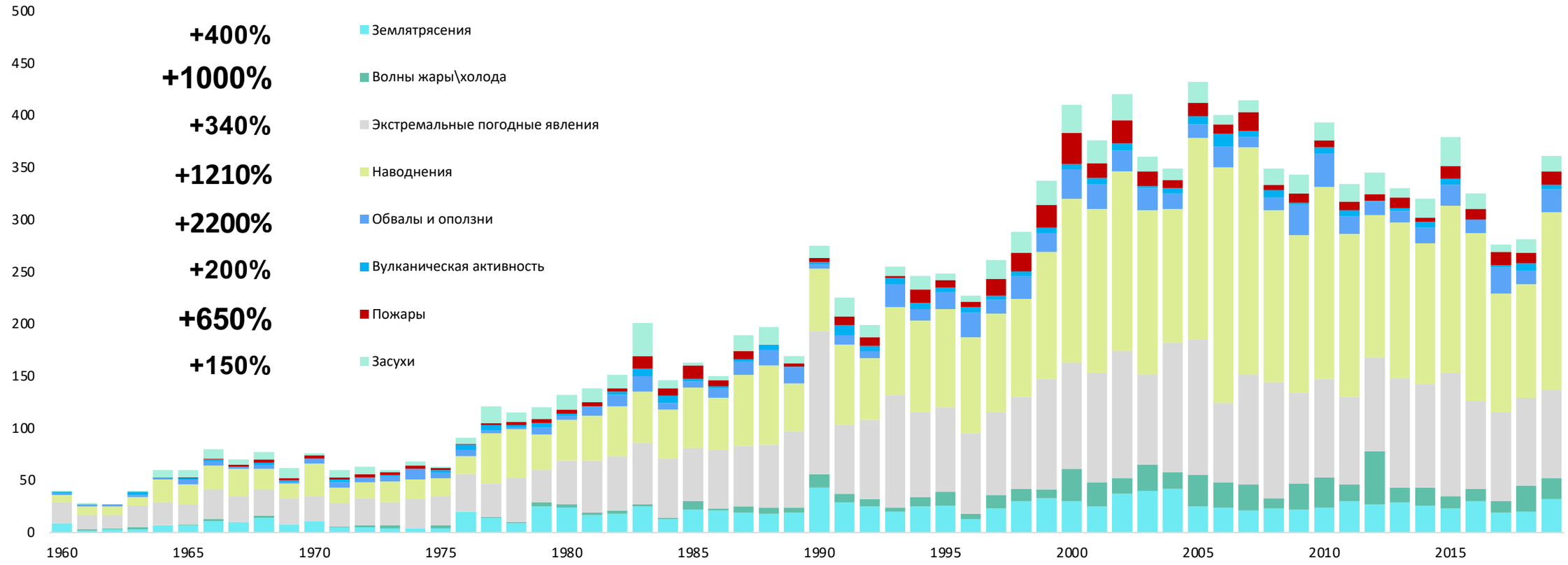
Ocean physical model



3-Day-Ahead Operational Sea Ice Regional Forecasting based on AI processing of remote sensing data

E-risks: forecast horizon is 5 – 30 years

The number of incidents of physical climate risks over the past 50 years (1970-2020) has increased 4.5 times



Source: Our World in Data, 2020

General scheme of damage assessment from E-risk

Losses = Risk of an Incident x Vulnerability of a Company

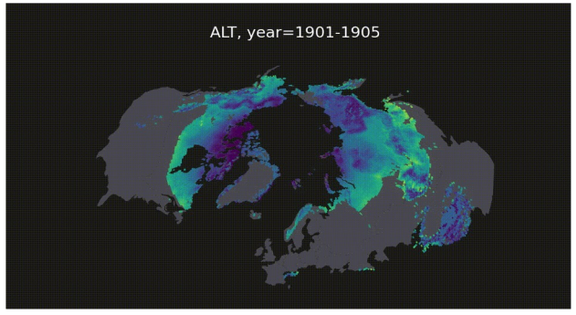
Machine Learning
- Historical data
- Locality of a forecast
- Time-space models (RNN-CNN architectures)

Data-driven analysis
- Insurance data
- Remote sensing data + ML
- Transactional analytics

Example of E-risk: Permafrost Melting

- **Permafrost modeling**
  - A mixture of Physics (Heat equation) and ML (model correction)
- **Importance**
  - 65% of the country land, major export resources (gas, oil, metals)
  - access infrastructure reliability, both short-term and long-term

Gas emission crater in YaNAO[2]

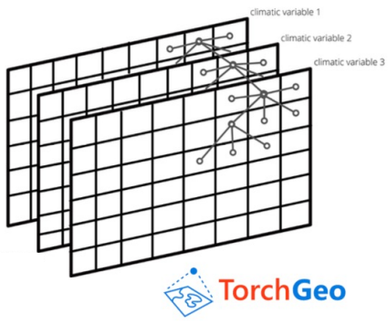


Kudryavtsev model results example

Damaged building caused by instability in foundation, Chersky settlement[2]

Natural Disasters Modeling: Extreme Rain, Wind, Temperature

- **Extreme Events are Hard to Predict:**
  A mixture of **Machine Learning (ML) and Probabilistic Modeling (PM)** works the best



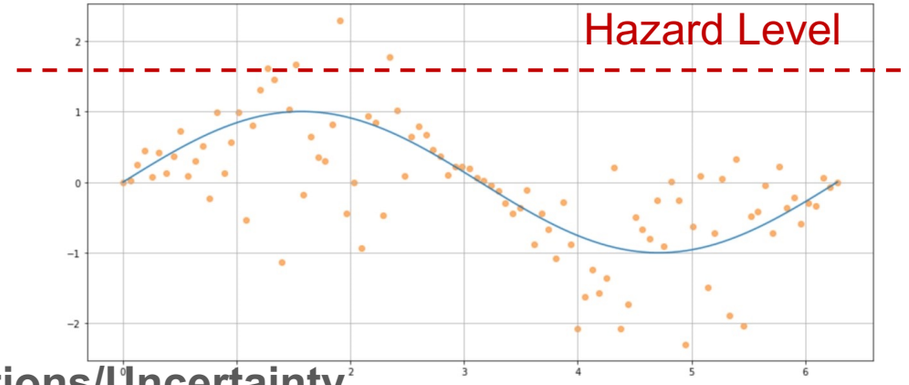
X = climate (CMIP 5/CMIP 6, re-analysis)

$$y = f(x) + \xi(x)$$

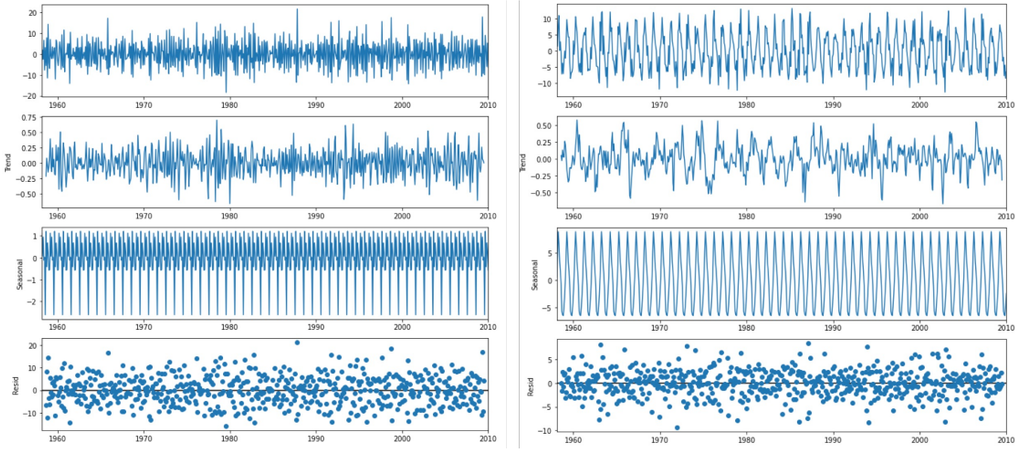
Observed value

Trend/Average value

Noise/Fluctuations/Uncertainty



- **Noise and Trend are meaningful:**
  pure ML and pure PM fail
  Extreme rains, snow, temperate anomalies can be predicted in this way



Precipitation (left) and Temperature (right) prediction

Math. ML Tasks

- **Dimensionality Reduction**: lower-dimensional features, preserving some properties of data
- **Regression**: predict some real-valued output variable for some input parameters (ship fuel consumption depending on weather conditions, route, etc.)
- **Classification**: set a label for each object (e.g. image classification)
- **Clustering**: partition objects into some "homogeneous" groups (e.g. divide documents into groups with similar topics)
- **Ranking**: rank objects according to some metric

Outline

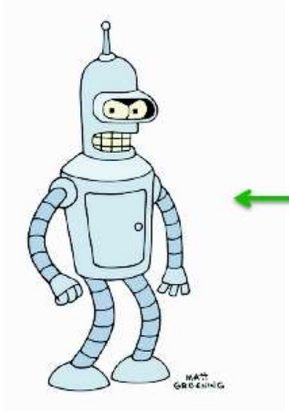
- Motivation
- **Supervised Learning**
- Neural Networks
- Unsupervised Learning and Generative Modeling
- What’s next

(Passive) Supervised Learning

raw unlabeled data

$x_1, x_2, x_3, \ldots$

random sample

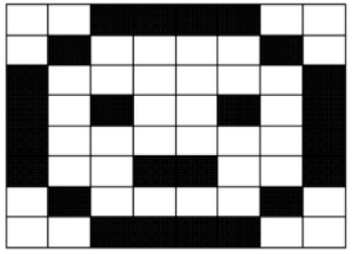


labeled training instances

$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$

supervised learner
induces a classifier

expert / oracle
analyzes experiments
to determine labels

O	x(O)	Y
		Person Bird House

Supervised learning

- **Training data**: sample S_m of size m drawn $i.i.d.$ according to distribution D on $X \times Y$

$$S_m = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

- **Empirical error** for $f \in F$ and sample S_m

$$L(f) = \frac{1}{m} \sum_{i=1}^{m} l(f(\mathbf{x}_i), y_i)$$

- **Generalization error**: for $f \in F$

$$L^*(f) = \mathbb{E}_{(\mathbf{x},y)\sim D}[l(f(\mathbf{x}), y)]$$

Supervised learning

- Let us select a hypothesis set $F = \{f_\theta, \theta \in \Theta\}$

- Find hypothesis $f \in F$ minimizing empirical error

$$L(\theta) = \frac{1}{m} \sum_{i=1}^{m} l(f_\theta(\mathbf{x}_i), y_i) \to \min_{\theta \in \Theta}$$

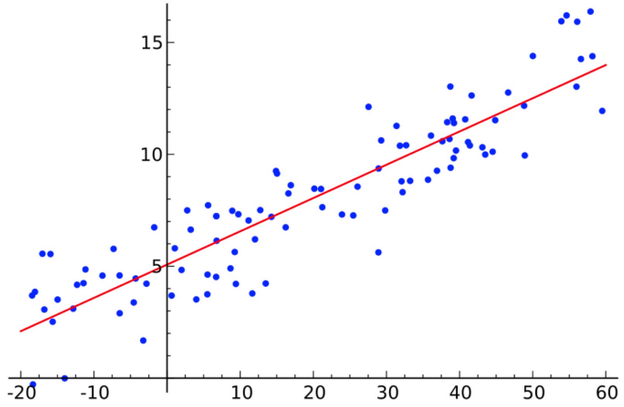
ML pipeline

1. Decompose an applied problem

2. Define
   - ✓ Features for object description
   - ✓ Method/Function class
   - ✓ Loss function
   - ✓ Validation approach

Supervised Learning: Regression

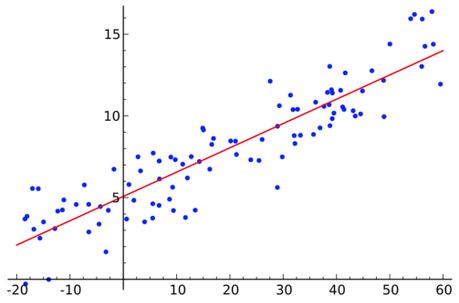
- **Loss function**: $l : Y \times Y \to \mathbb{R}_+$ a measure of closeness, e.g. $l(y, y') = (y - y')^2$ or $l(y, y') = |y - y'|^p$ for some $p \geq 1$



- Hypothesis set: linear functions

$$F = \{\mathbf{x} \to \mathbf{w} \cdot \mathbf{x}^\top + b : \mathbf{w} \in \mathbb{R}^{1 \times d}, b \in \mathbb{R}\}$$

Supervised Learning: Linear Regression



- **Optimization problem**: empirical risk minimization

$$L(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{w} \cdot \mathbf{x}_i^\top + b - y_i\right)^2 \to \min_{\mathbf{w}, b}$$

- Rewrite objective function as $F(\mathbf{W}) = \frac{1}{m} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|^2$, where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & 1 \\ \vdots & \vdots \\ \mathbf{x}_m & 1 \end{bmatrix} \in \mathbb{R}^{m \times (d+1)}, \ \mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix}, \ \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

- **Solution**:

$$\mathbf{W} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} \ \text{ if } \ \mathbf{X}^\top \mathbf{X} \ \text{ invertible}$$

Supervised Learning: Linear Regression

- **Optimization problem**:

$$L(\mathbf{w}, b) = \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i^\top + b - y_i)^2 + \lambda \|\mathbf{w}\|^2 \to \min_{\mathbf{w}, b},$$

where $\lambda \geq 0$ is a regularization parameter

- **Solution**:

$$\mathbf{W} = \underbrace{(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}}_{\text{always invertible!}} \mathbf{X}^\top \mathbf{Y}$$

Supervised Learning: Linear Regression

- **Dual solution**: thus we get that

$$\mathbf{W} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{Y} = \mathbf{X}^\top \underbrace{(\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1} \mathbf{Y}}_{\text{new variable } \boldsymbol{\alpha}}$$

- With

$$\boldsymbol{\alpha} = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1} \mathbf{Y},$$

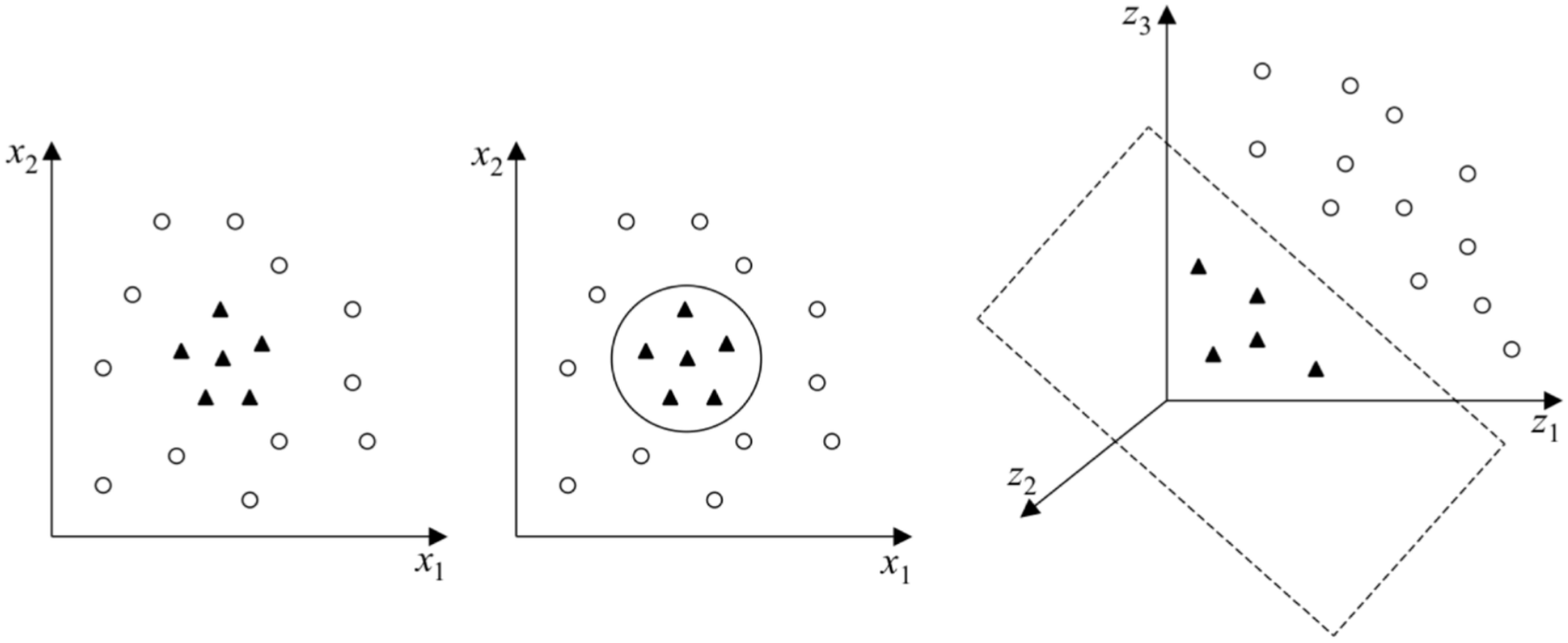
  we can represent $\mathbf{W}$ as

$$\mathbf{W} = \mathbf{X}^\top \boldsymbol{\alpha} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i^\top,$$

- We can use dual representation of the solution

$$\widehat{f}(\mathbf{x}) = \mathbf{x} \cdot \mathbf{W} = \sum_{i=1}^{m} \alpha_i (\mathbf{x} \cdot \mathbf{x}_i^\top)$$

Kernels



For $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, let $\Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$

Kernels

- **Idea**:
  - Define $K : X \times X \to \mathbb{R}$ called kernel, such that
  $$\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')^\top = K(\mathbf{x}, \mathbf{x}')$$
  - K is often interpreted as a similarity measure

$$\begin{aligned} K(\mathbf{x}', \mathbf{x}) &= \Phi(\mathbf{x}') \cdot \Phi(\mathbf{x})^\top \quad \text{[dot product of features]} \\ &= x_1^2 (x_1')^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 (x_2')^2 \\ &= (x_1 x_1' + x_2 x_2')^2 = (\mathbf{x}' \cdot \mathbf{x}^\top)^2 \end{aligned}$$

- **Gaussian kernels**:
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right), \ \sigma \neq 0$$

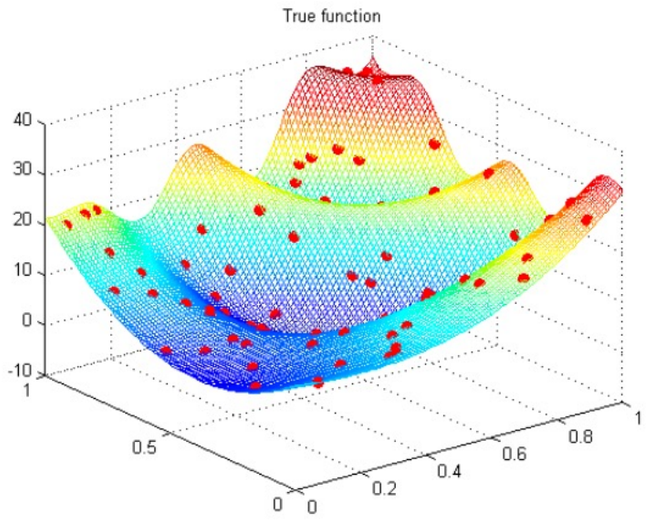
Supervised Learning: Kernel Ridge regression

- Usual linear ridge regression in dual representation

$$\widehat{f}(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i (\mathbf{x} \cdot \mathbf{x}_i^\top)$$

with

$$\boldsymbol{\alpha} = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1} \mathbf{Y}$$



- Kernel ridge regression

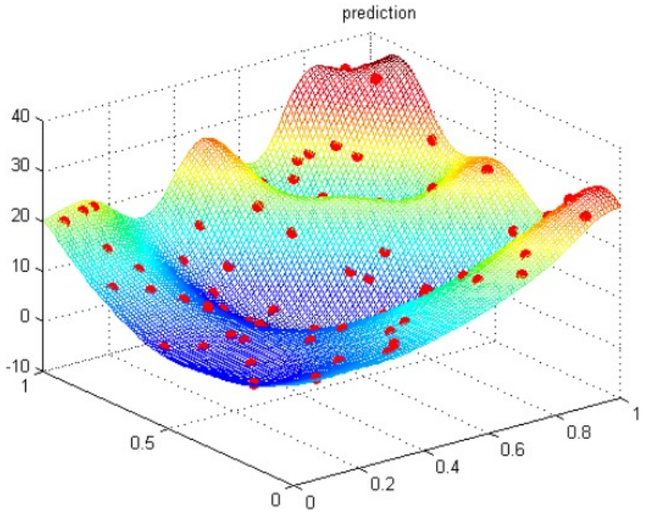
$$\widehat{f}(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)^\top) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

with

$$\boldsymbol{\alpha} = (\Phi(\mathbf{X}) \cdot \Phi(\mathbf{X})^\top + \lambda \mathbf{I})^{-1} \mathbf{Y} = (\mathrm{K} + \lambda \mathbf{I})^{-1} \mathbf{Y},$$

where

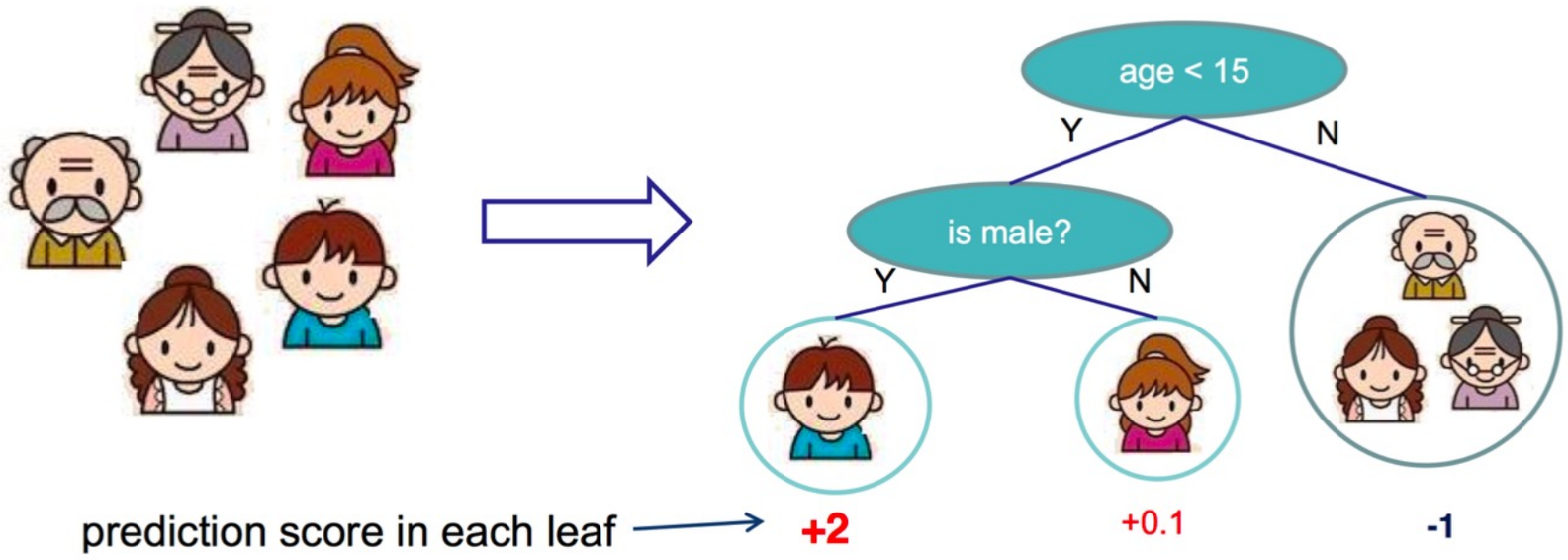
$$\mathrm{K} = \{\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)^\top\}_{i,j=1}^{m} = \{K(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^{m}$$



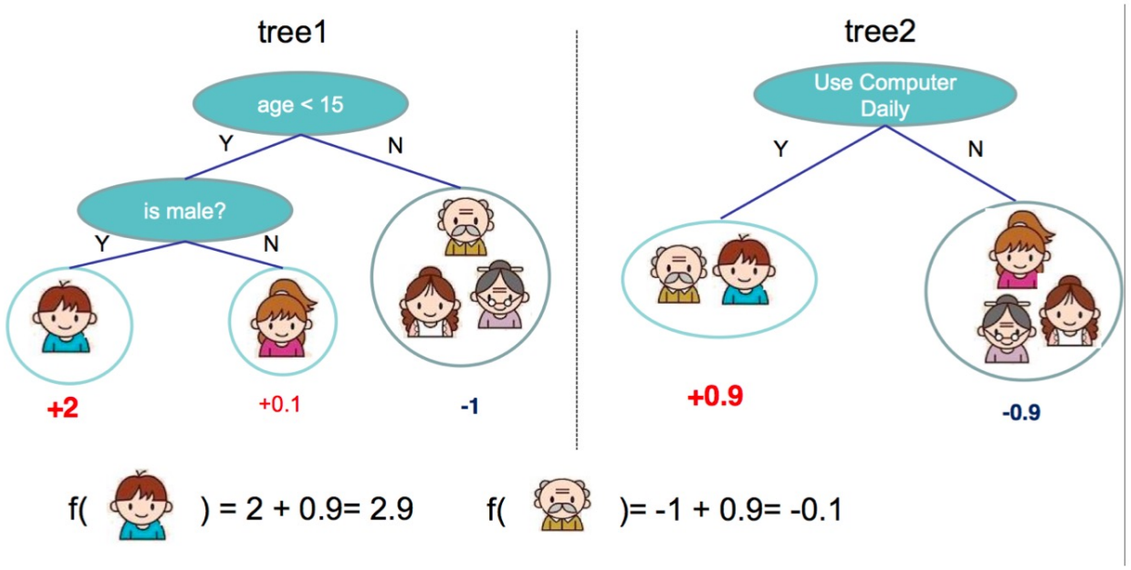
Supervised Learning: Decision Trees

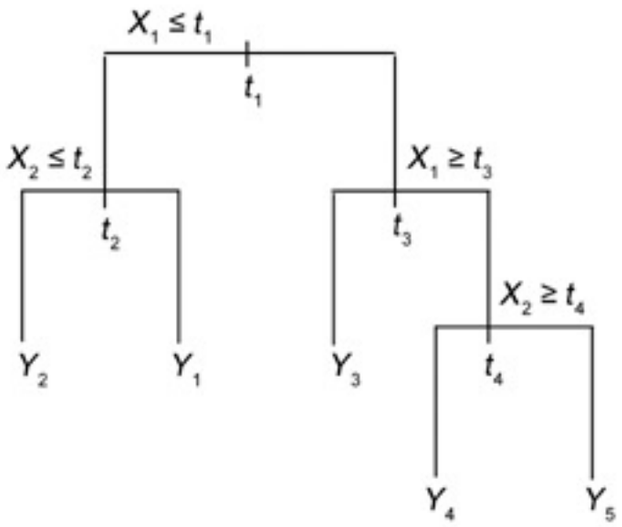
- Classification and Regression Trees:
  - Decision rules
  - Contains one score in each leaf value

Input: age, gender, occupation,... $\Rightarrow$ Does the person like computer games?

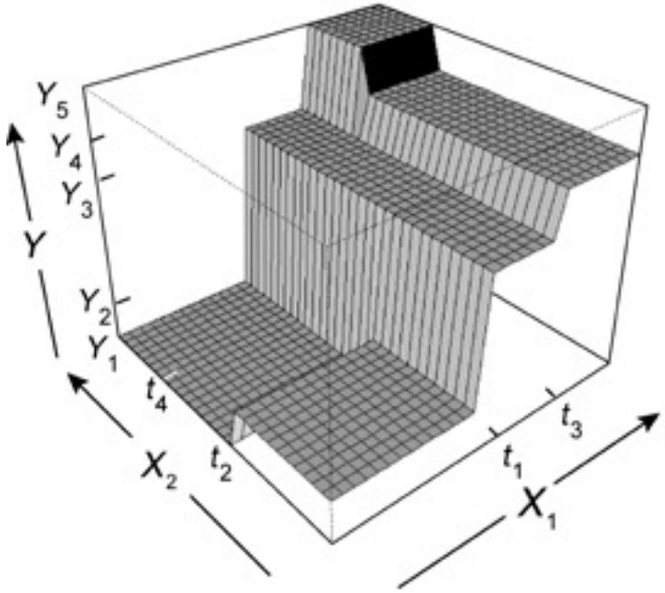


Supervised Learning: Tree Ensembles





Prediction is a sum of scores predicted by each of the tree



- Model: we have T trees

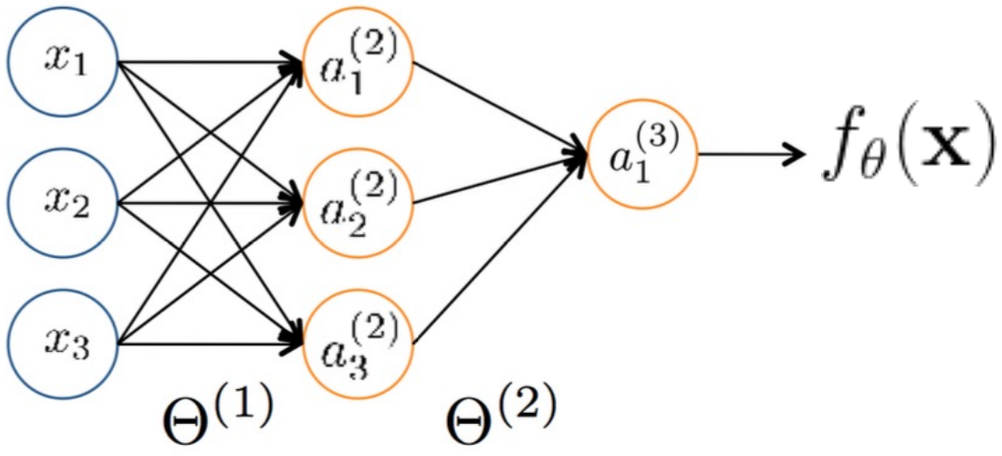
$$\widehat{f}_T(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x}), \;\; f_t(\mathbf{x}) \in F,$$

  where F is a space of functions, containing all regression trees
- Parameters: structure of each tree, and the score in the leaf

Outline

- Motivation
- Supervised Learning
- **Neural Networks**
- Unsupervised Learning and Generative Modeling
- What’s next

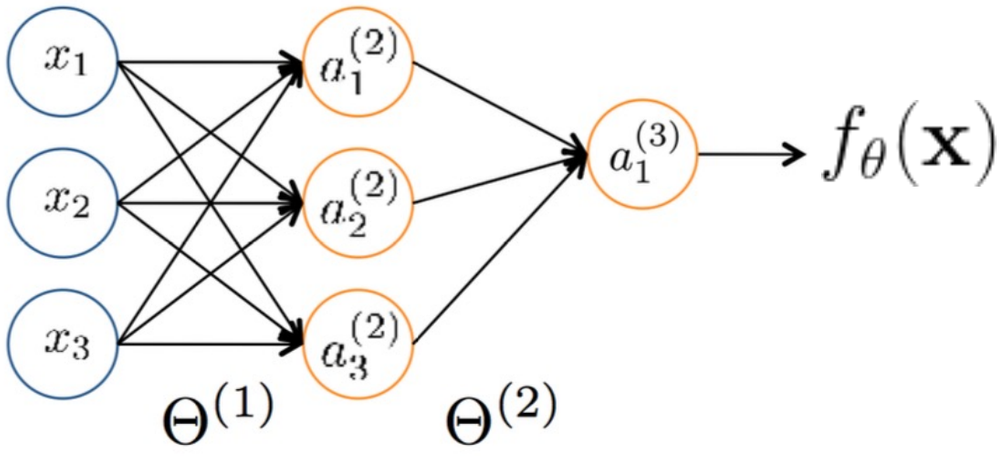
Neural Networks



Feed-forward Steps:

- $\mathbf{z}^{(2)} = \Theta^{(1)}\mathbf{x}$
- $\mathbf{a}^{(2)} = \sigma(\mathbf{z}^{(2)}),\ \sigma(t) = \max(0, t)$ (ReLU)
- $\mathbf{z}^{(3)} = \Theta^{(2)}\mathbf{a}^{(2)}$
- $f_\theta(\mathbf{x}) = \mathbf{a}^{(3)} = p(\mathbf{z}^3),\ p(t) = \frac{e^t}{1+e^t}$

Deep Networks



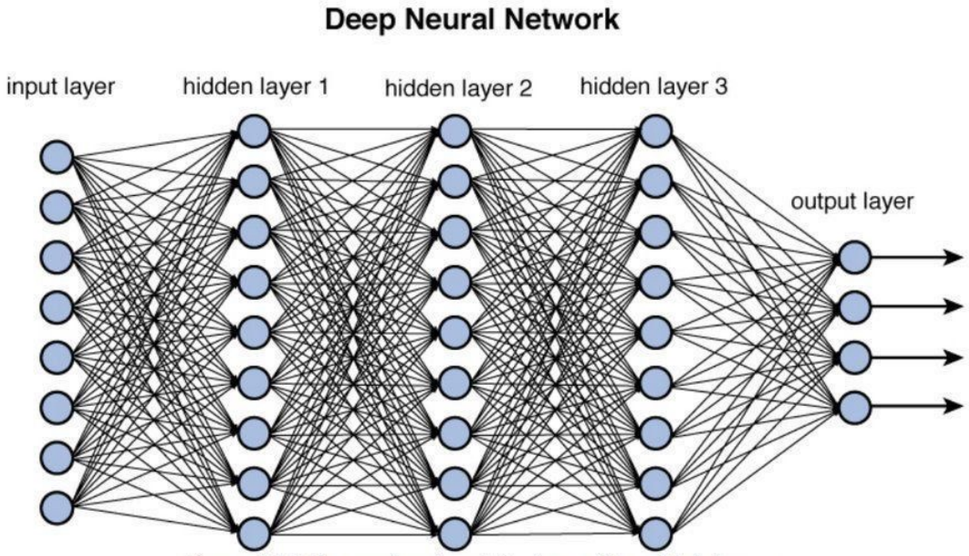


Figure 12.2 Deep network architecture with multiple layers.

Feed-forward Steps:

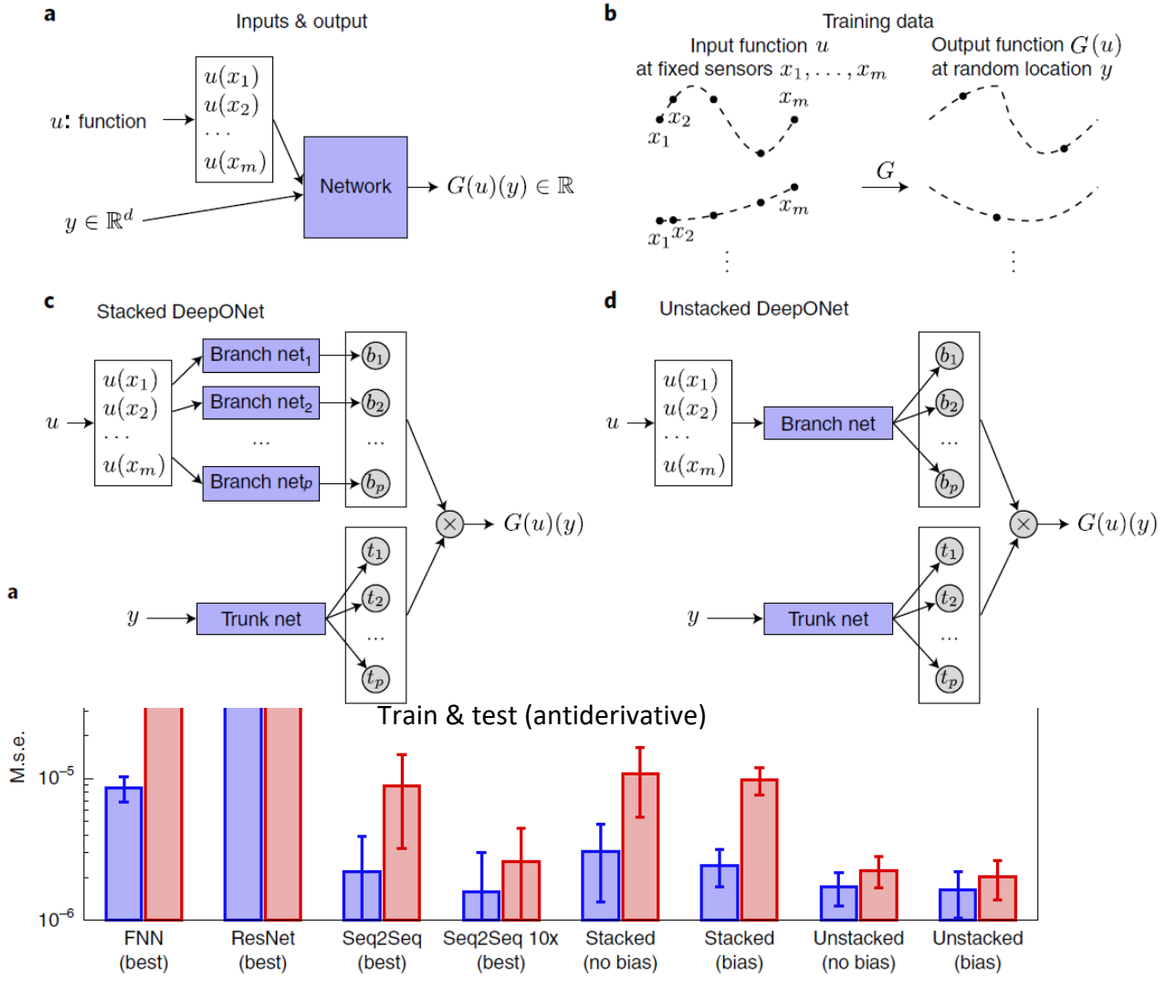
- $\mathbf{z}^{(2)} = \Theta^{(1)}\mathbf{x}$
- $\mathbf{a}^{(2)} = \sigma(\mathbf{z}^{(2)}),\ \sigma(t) = \max(0, t)$ (ReLU)
- $\mathbf{z}^{(3)} = \Theta^{(2)}\mathbf{a}^{(2)}$
- $f_\theta(\mathbf{x}) = \mathbf{a}^{(3)} = p(\mathbf{z}^3),\ p(t) = \frac{e^t}{1+e^t}$

Universal Operator Approximator Theorem

Theorem 1 (Universal Approximation Theorem for Operator). *Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p and m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$ and $j = 1, \ldots, m$, such that*

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{\text{branch}} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{\text{trunk}} \right| < \epsilon \tag{1}$$

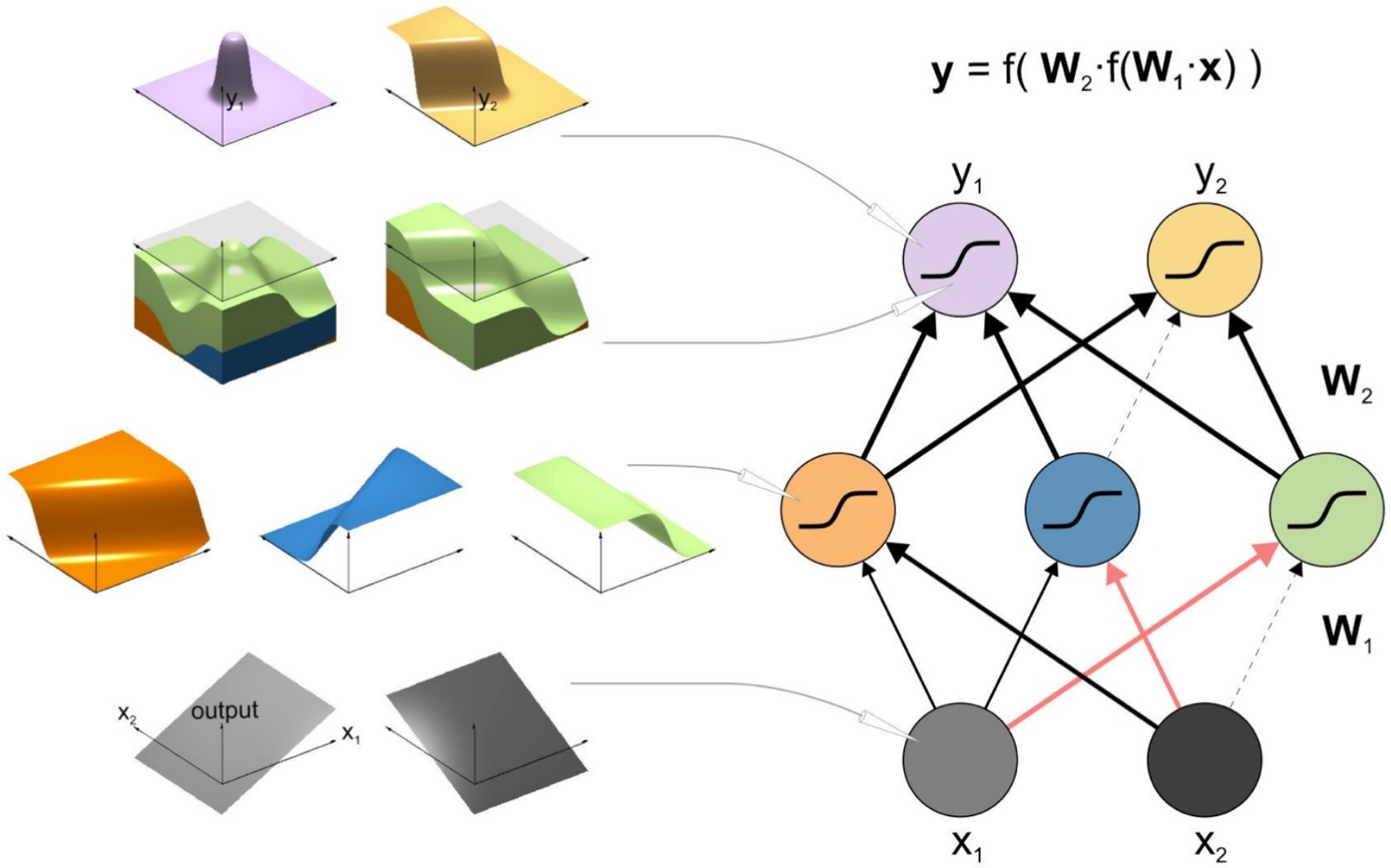
holds for all $u \in V$ and $y \in K_2$. Here, $C(K)$ is the Banach space of all continuous functions defined on K with norm $\| f \|_{C(K)} = \max_{x \in K} |f(x)|$.

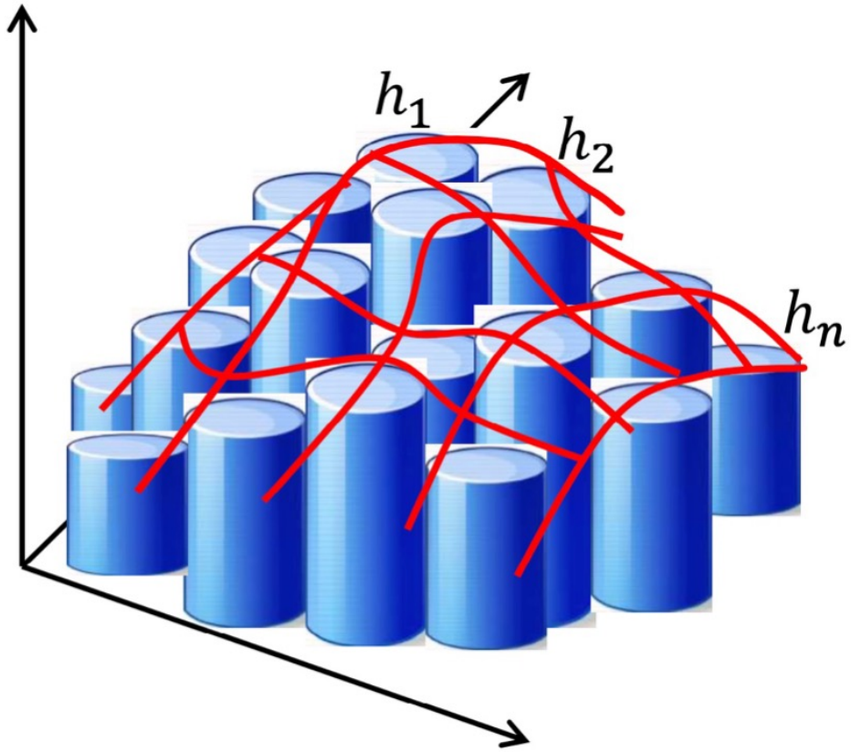


Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks, 6(4), 911-917.

Lu, Lu, et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators." Nature Machine Intelligence 3.3 (2021): 218-229.

MLPs and Universal Function Approximators



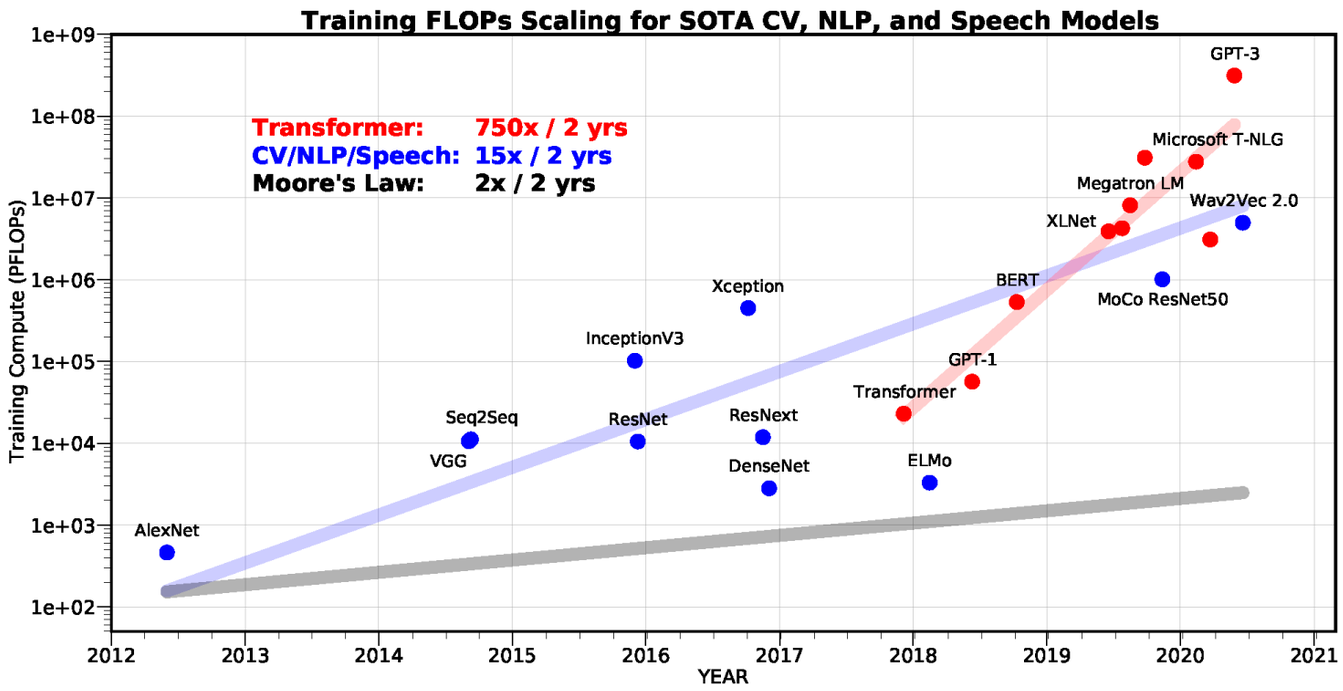


Theorem: There exists a Boolean function of $d > 2$ variables that requires at least $2^d/d$ Boolean gates, regardless of depth!

G. Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4):303–314, 1989.
K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. Neural Networks, 2(5):359–366, 1989.
Kriegeskorte N, Golan T. Neural network models and deep learning-a primer for biologists. arXiv preprint arXiv. 1902.

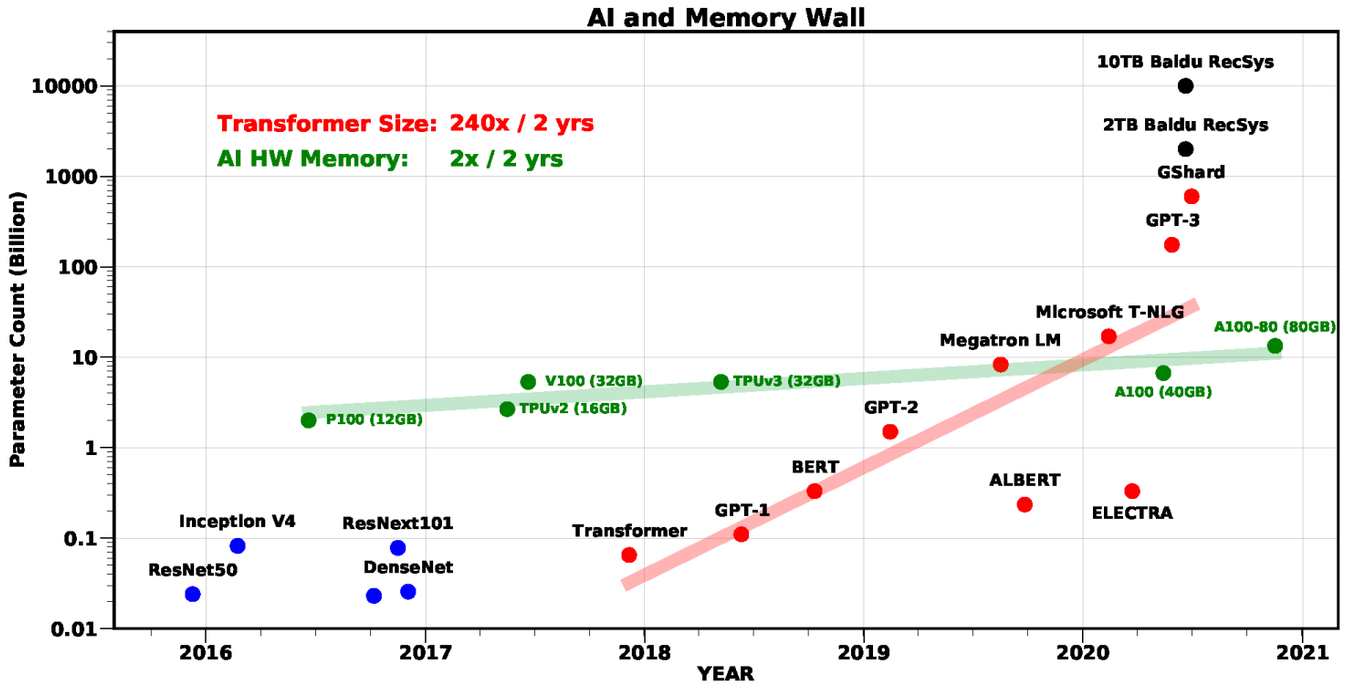
What works in practice?

Exponentially Expensive Models to Train



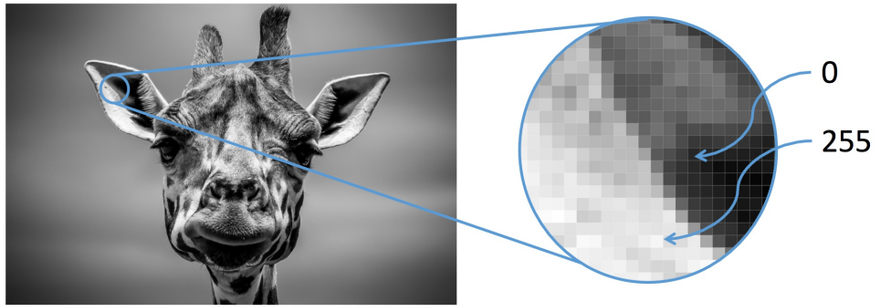
Biases are important!!!

Extremely Overparameterized Models

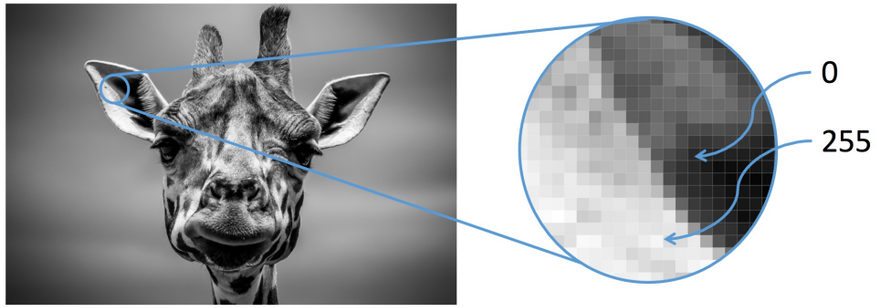


Amir Gholami, Zhewei Yao, Sehoon Kim, Michael W. Mahoney, Kurt Keutzer, AI and Memory Wall, Riselab Medium Blogpost, 2021.

Images



Convolutions



1	0	1	0
0	1	1	0
1	0	1	0
1	0	1	1

Input

1	0
0	1

Image patch (local receptive field)

*

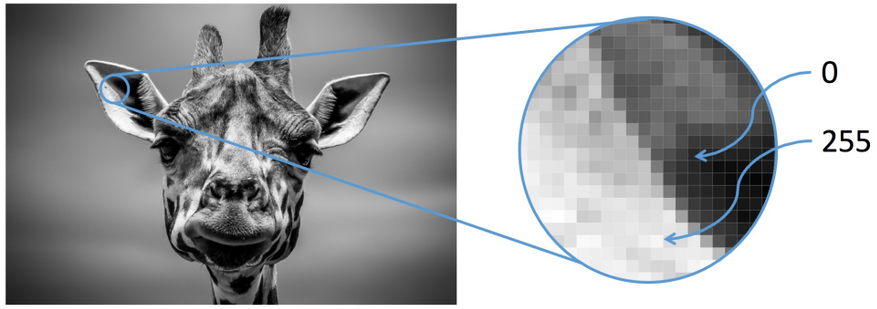
1	2
3	4

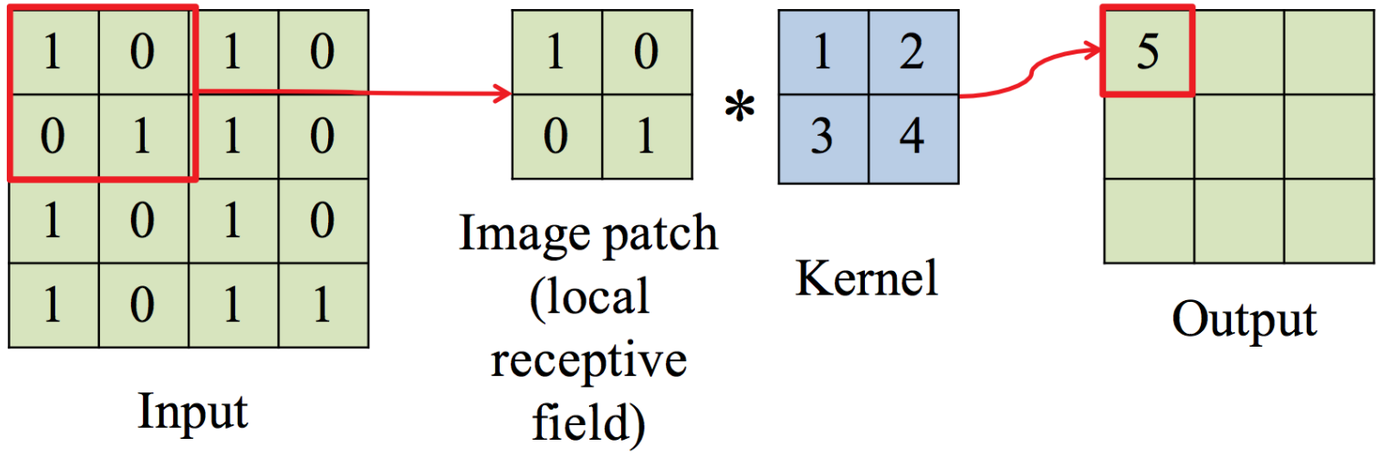
Kernel

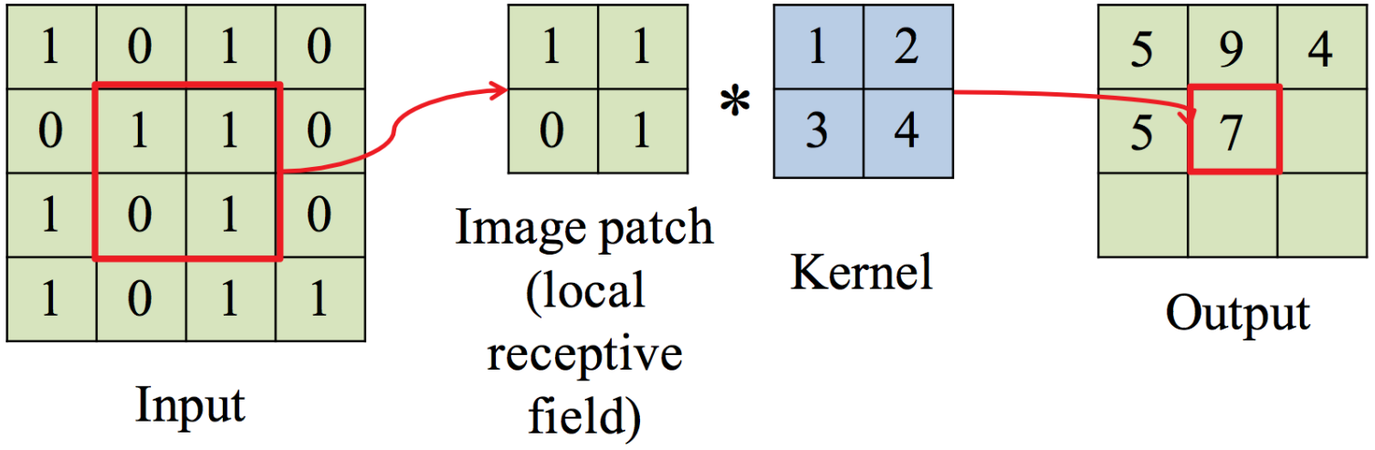
5		

Output

Convolutions







Convolutions

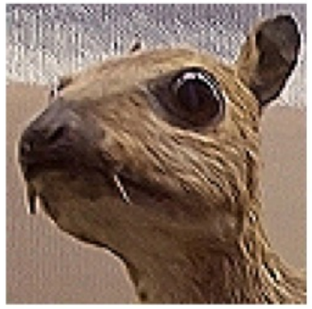
Original image

Kernel

$$* \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

Edge detection

$$* \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} =$$

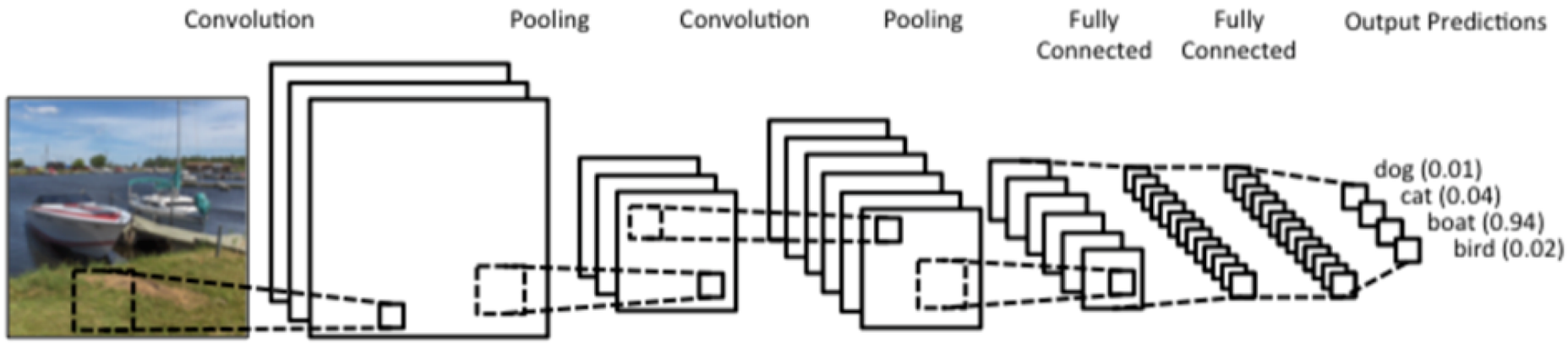


Sharpening

$$* \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

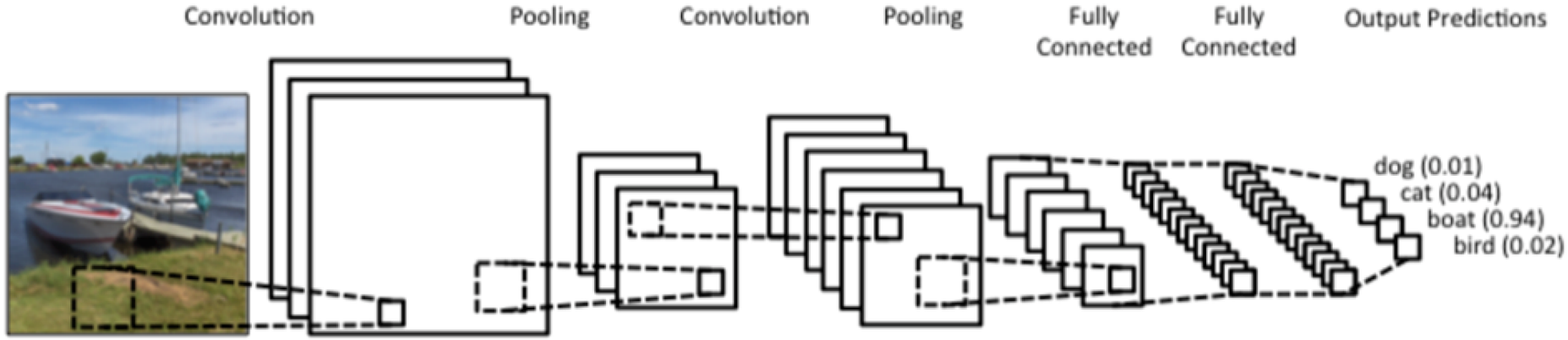
Blurring

Convolutional Neural Networks

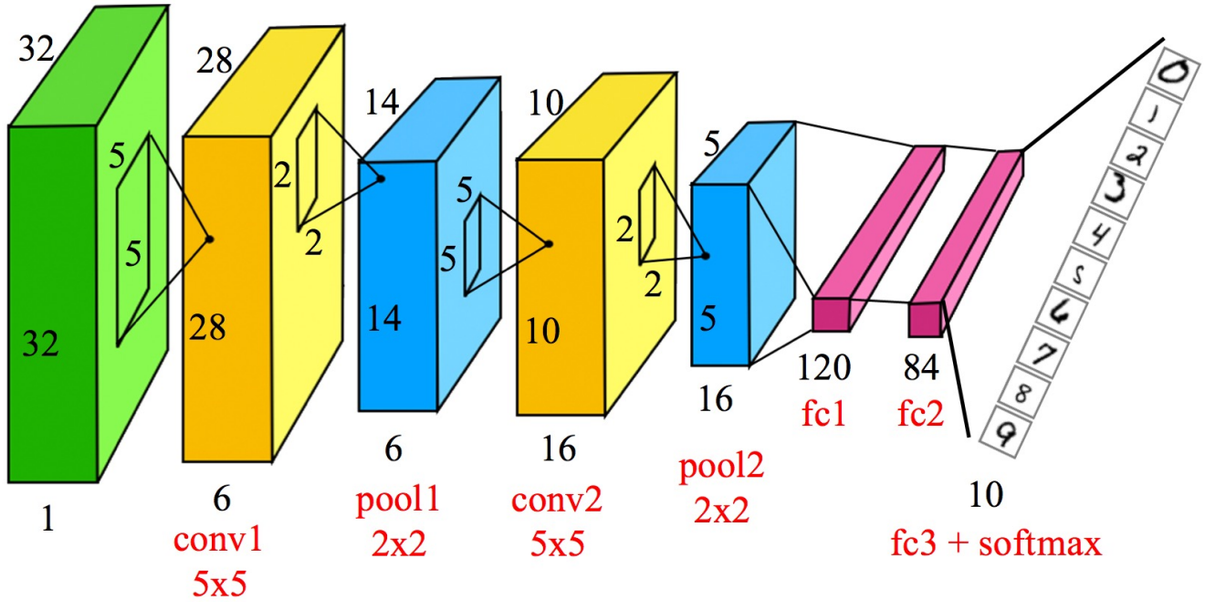


$$\sigma(t) = \max(0, t) \text{ (ReLU)}$$

Convolutional Neural Networks



$$\sigma(t) = \max(0, t) \text{ (ReLU)}$$



http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

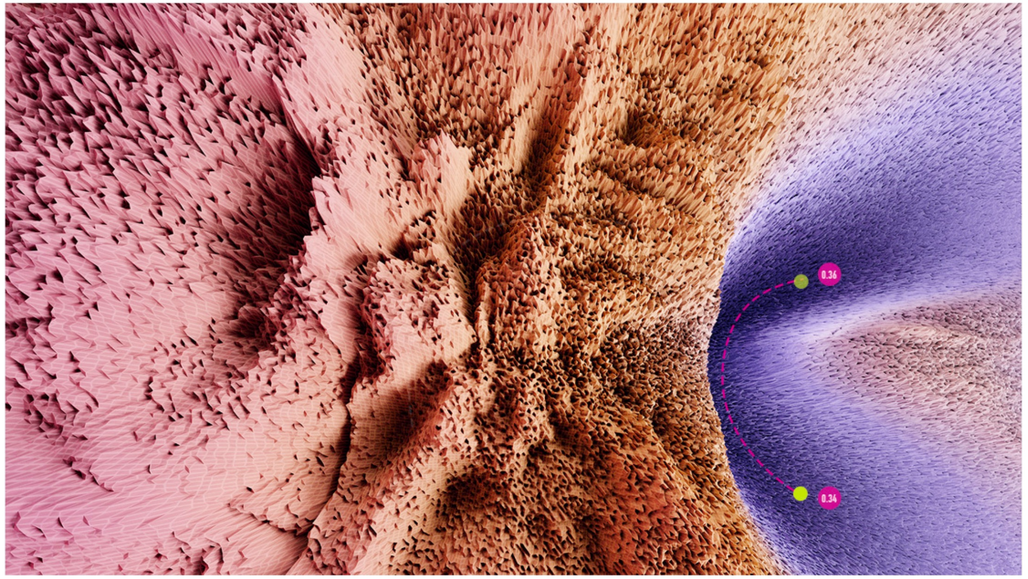
$$\sigma(t_1, \ldots, t_k) = \left\{ \frac{e^{t_1}}{\sum_j e^{t_j}}, \ldots, \frac{e^{t_k}}{\sum_j e^{t_j}} \right\}$$

Deep Learning Loss Landscapes

$$L(\theta) = \frac{1}{m}\sum_{i=1}^{m} l(f_\theta(\mathbf{x}_i), y_i),\ \theta \in \Theta$$

Video credit to losslandscape.com

Challenges



Loss surface:

- Non-convexity
- Many local minima
- Saddle points
- Flat regions

Some phenomena:

- The algorithms based on gradient descent achieve almost zero loss with Deep Neural Nets although the loss functions of DNNs are non-convex
- **Generalization**: There is no overfitting despite that the number of parameters is much bigger than the number of data points (overparametrization)

Why does DL work?

- **What does a DL system really learn?**

Probability distributions on manifolds

- **How does a DL system learn? Does it really learn or just memorize?**

Optimization in the space of all probability distributions on a manifold. A DL system both learns and memorizes

Outline

- Motivation
- Supervised Learning
- Neural Networks
- **Unsupervised Learning and Generative Modeling**
- What’s next

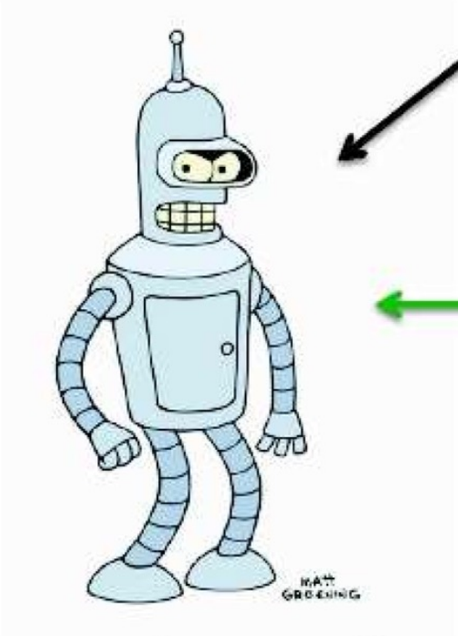
Semi-Supervised Learning

raw unlabeled data

$x_1, x_2, x_3, \ldots$

exploit the structure in unlabeled data

random sample



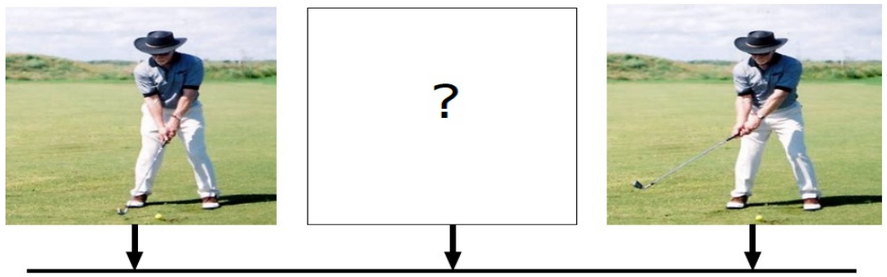
labeled training instances

$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$

semi-supervised learner
induces a classifier

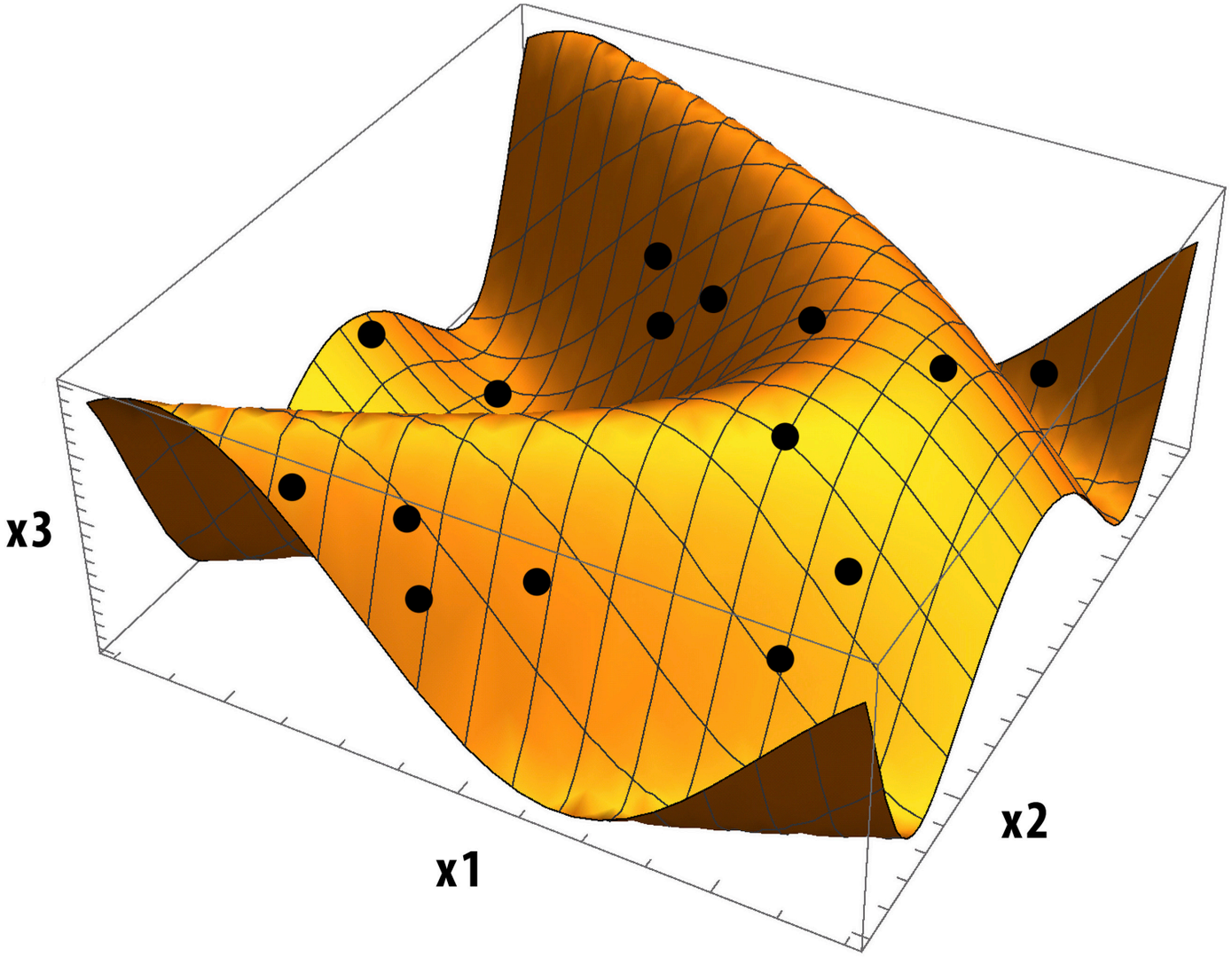
expert / oracle
analyzes experiments
to determine labels

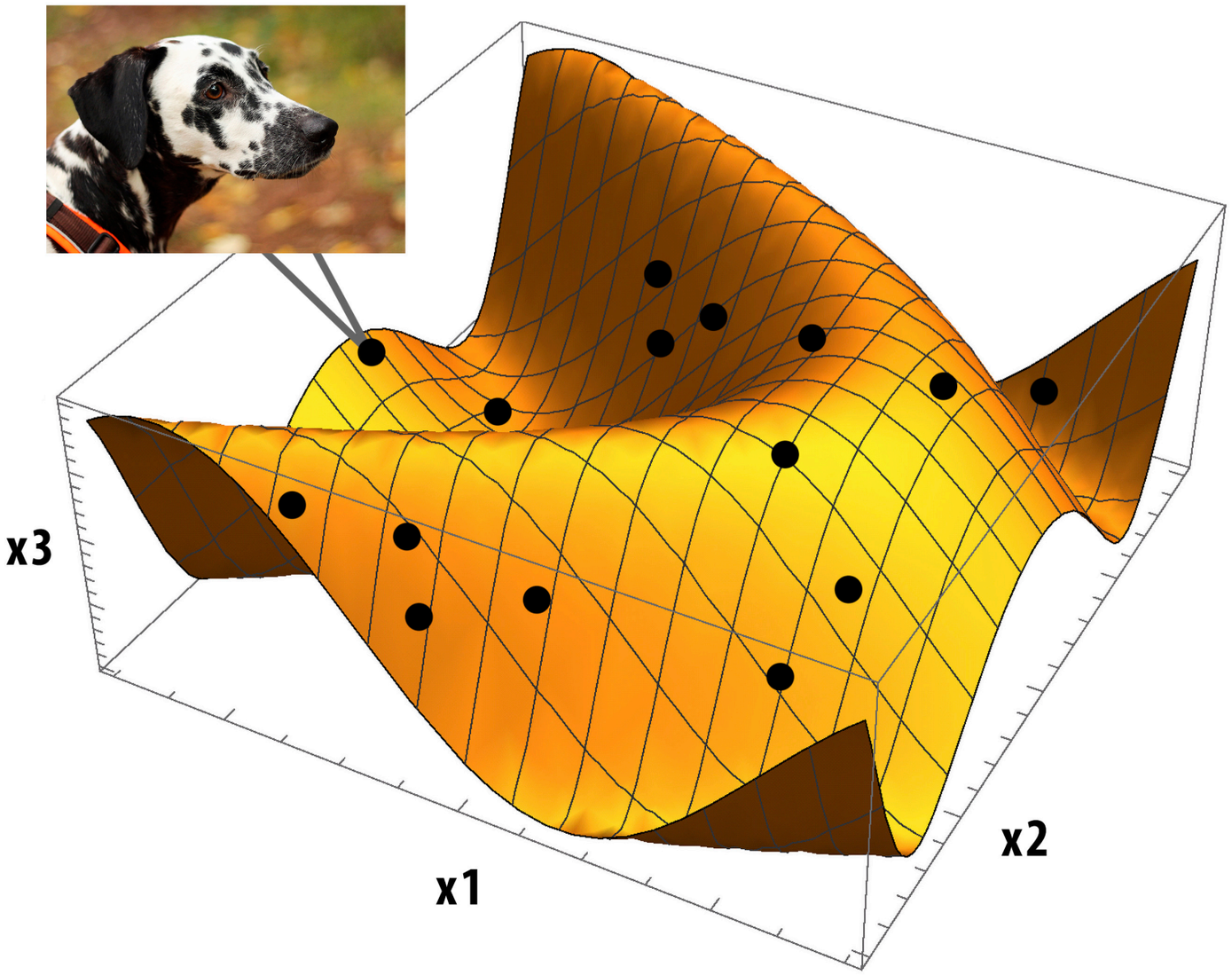
The word is not flat (nonlinear)

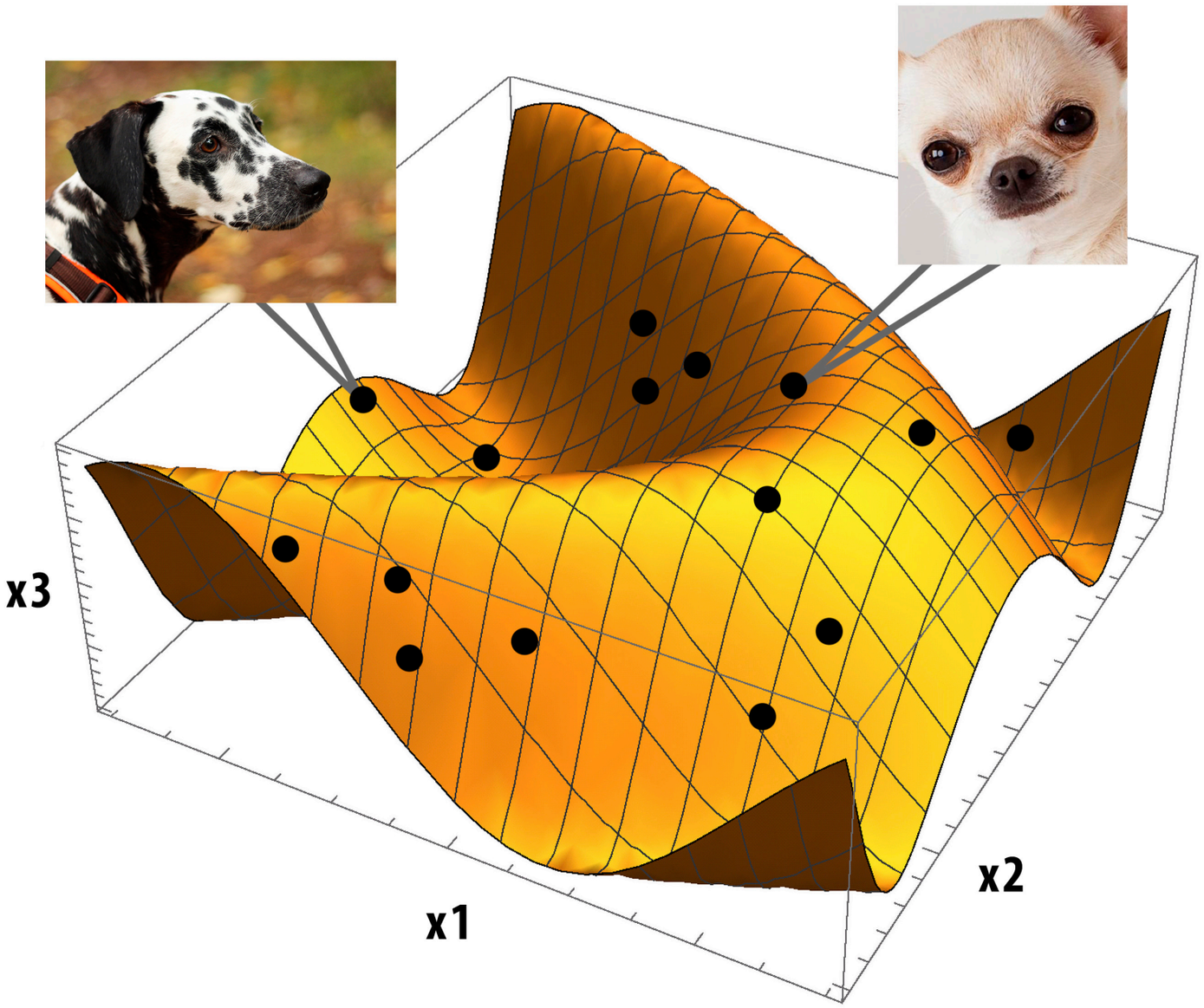


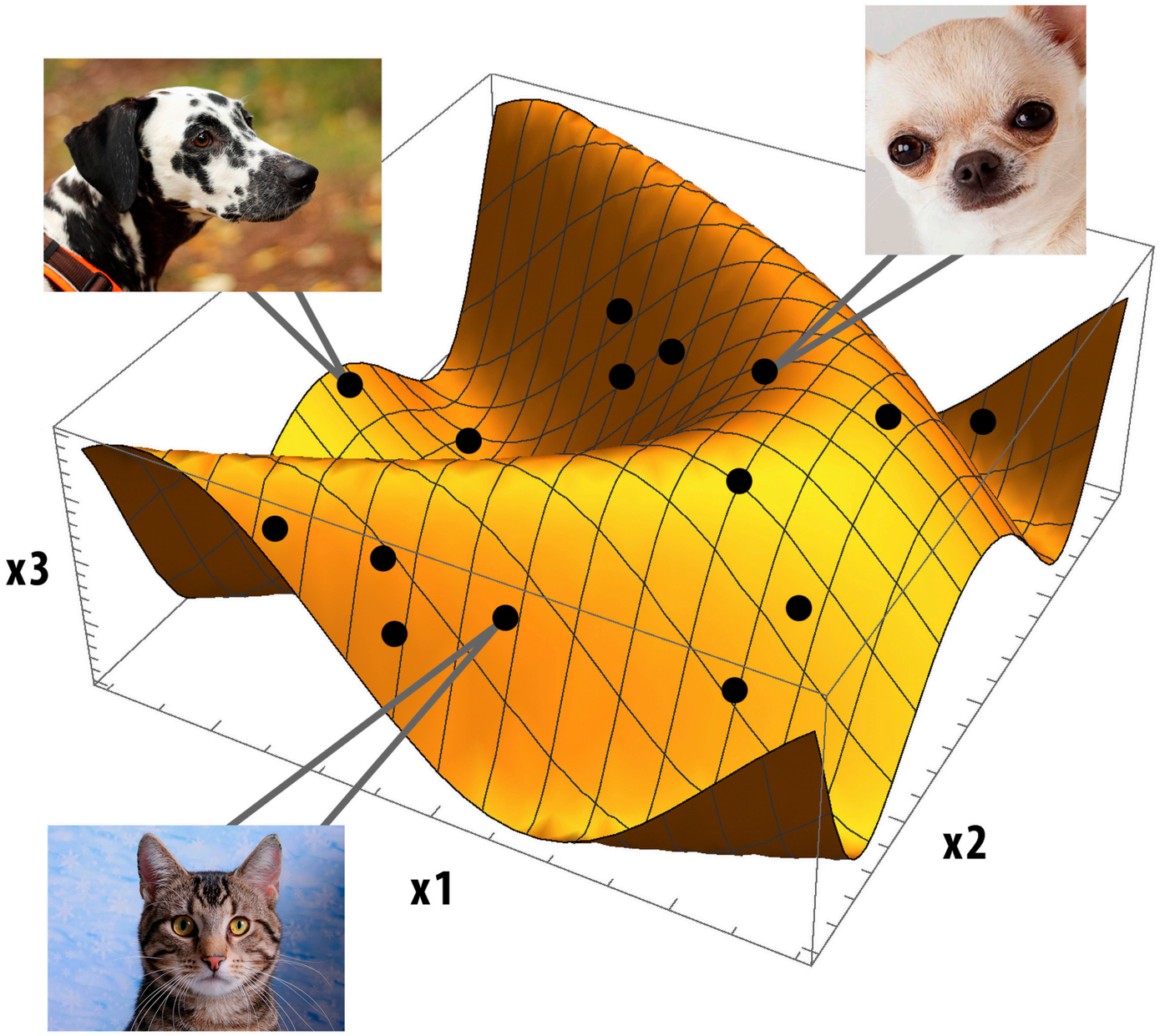
Linear interpolation

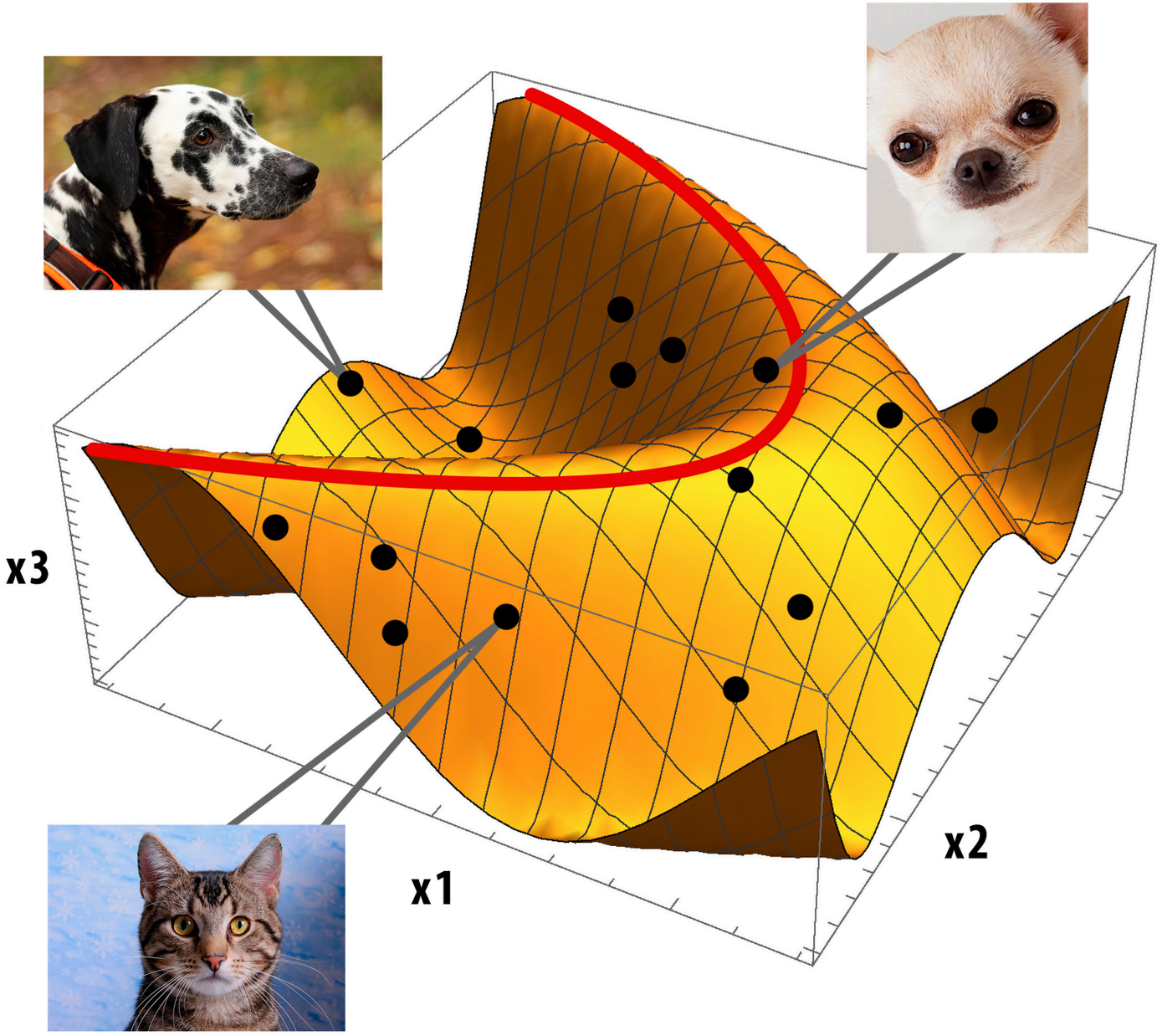
Nonlinear interpolation

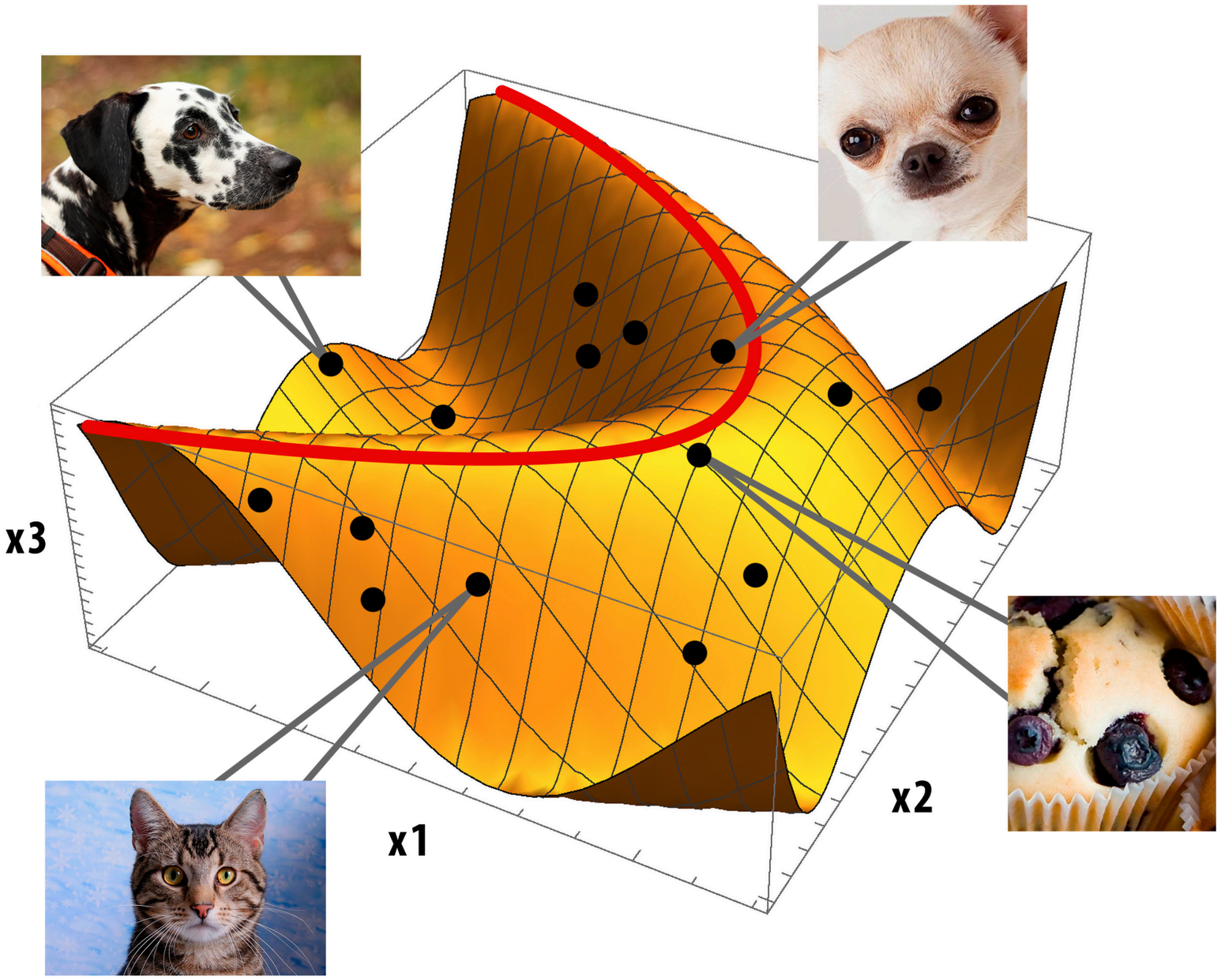

x3
x1
x2


x3
x1
x2


x3
x1
x2


x3
x1
x2


x3
x1
x2


x3
x1
x2

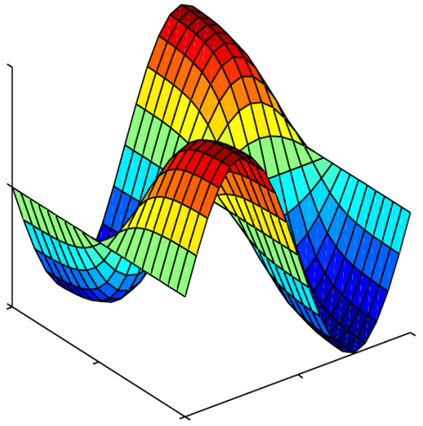
Manifold Learning

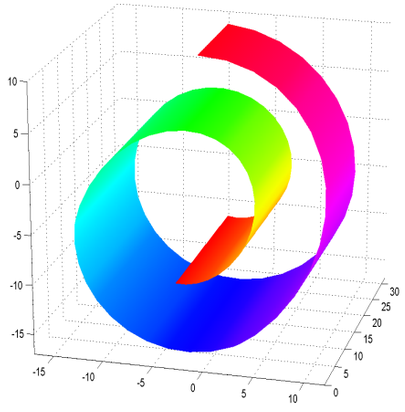
Manifold covered by a single chart (surface in $\mathbb{R}^d$)

$$\mathbf{M} = \{x = g(z) \in \mathbb{R}^d : \; z \in \mathbf{Z} \subset \mathbb{R}^s\}$$

unknown s -dimensional surface – **Data manifold**

covered by **single chart** g defined on **Coordinate space** $\mathbf{Z} \subset \mathbb{R}^s$

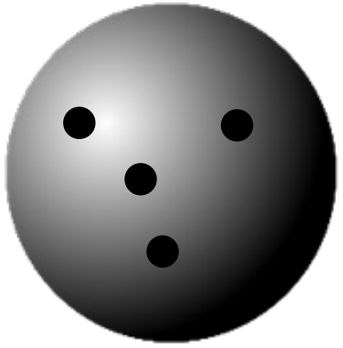




Latent Generative Model

$$\boxed{x \sim p(x|g_\theta(z)) \cdot p(z)}$$

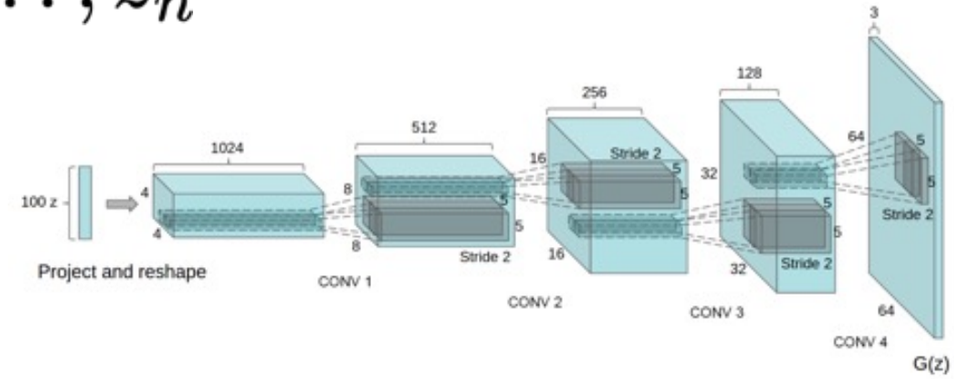
$z \sim p(z)$



g_θ

$z_1, z_2, \ldots, z_n$

$x_1, x_2, \ldots, x_n$

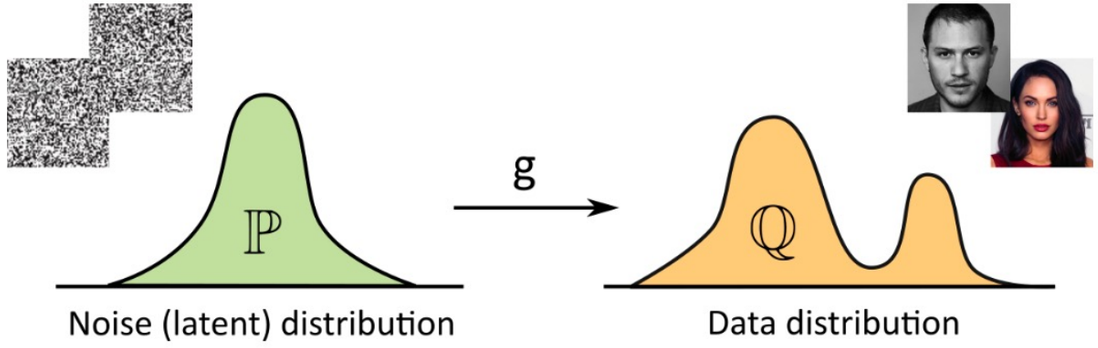


Generative Modeling tasks

Map the given distribution $\mathbb{P}$ into the given distribution $\mathbb{Q}$

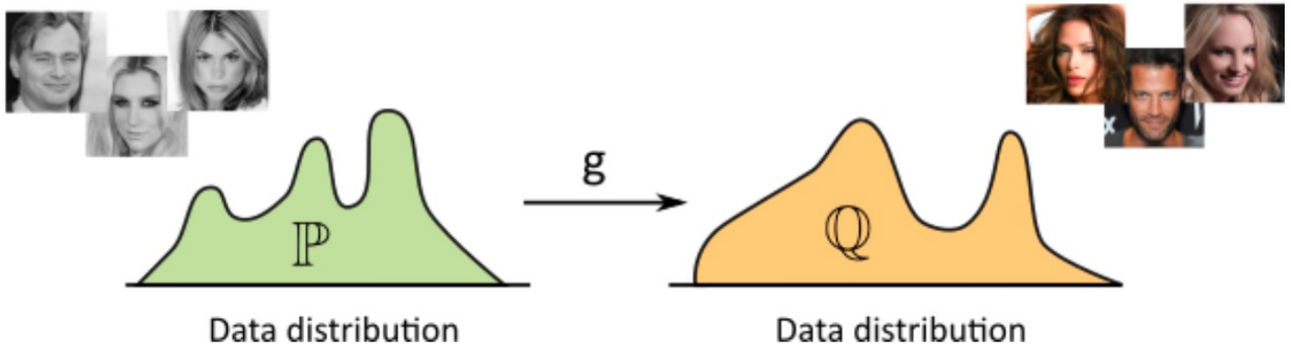
Case 1: noise $\rightarrow$ data

synthetic data generation/data manipulation



Case 2: data $\rightarrow$ data

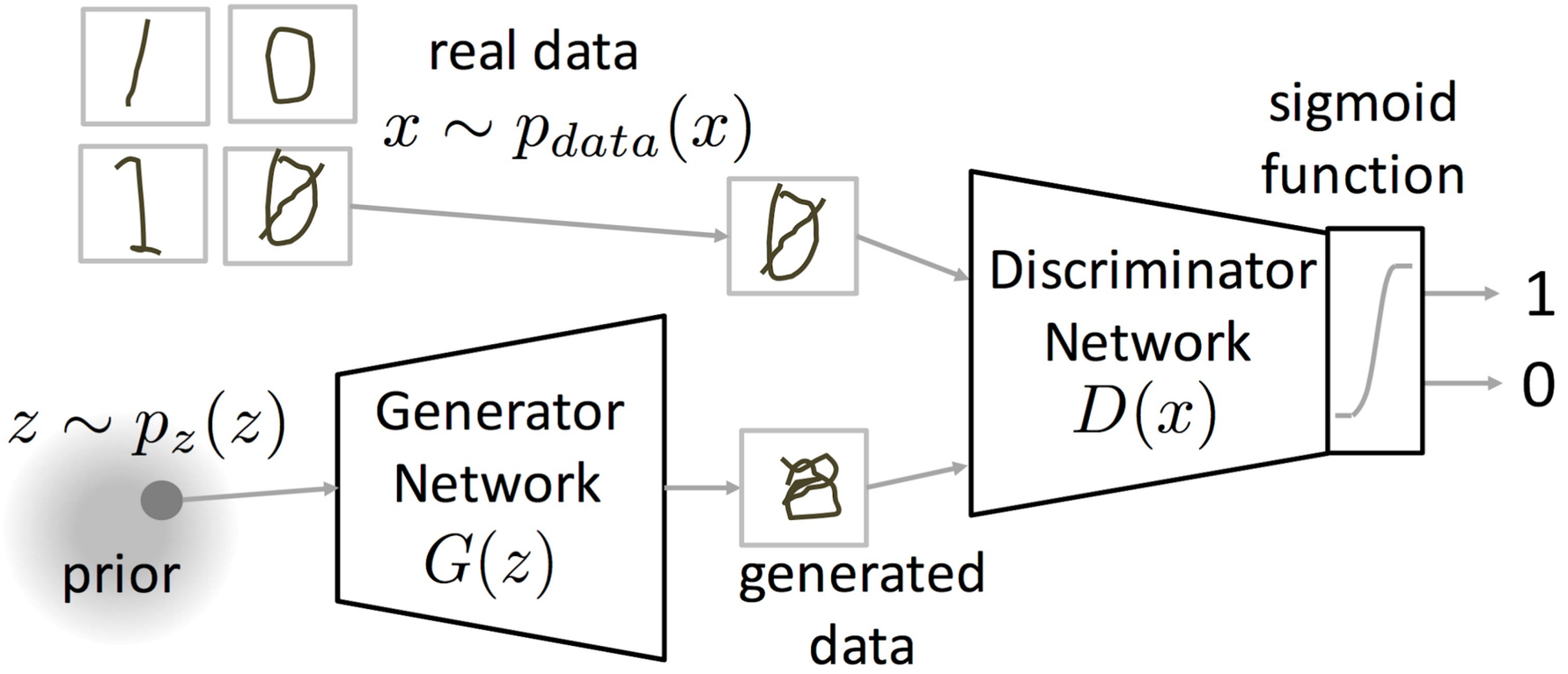
unpaired style transfer, super-resolution, domain adapatation



Generative Adversarial Network

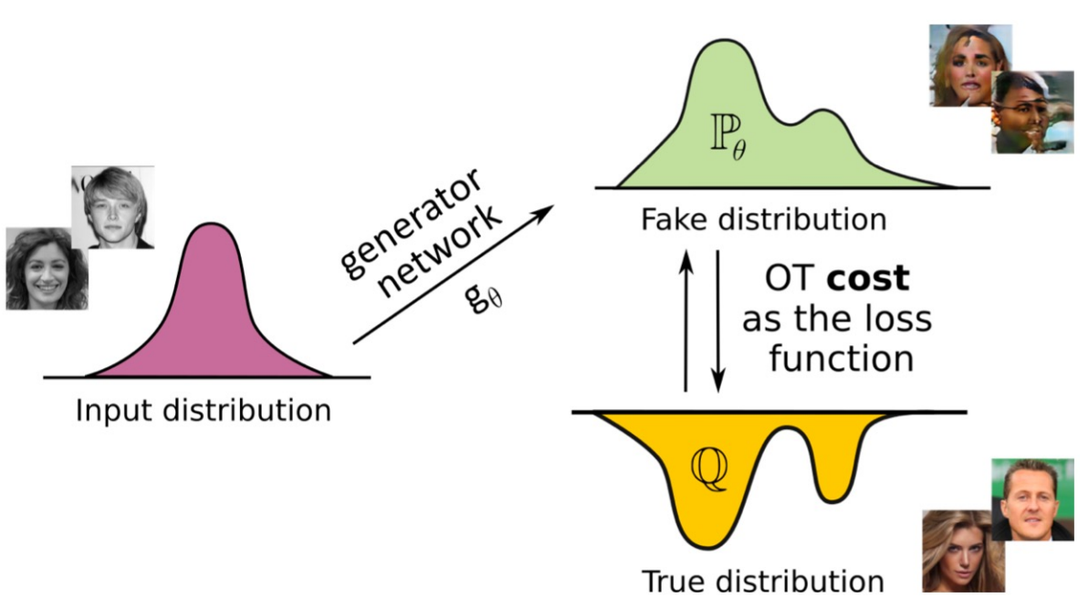
$$\min_G \max_D V(D, G)$$

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

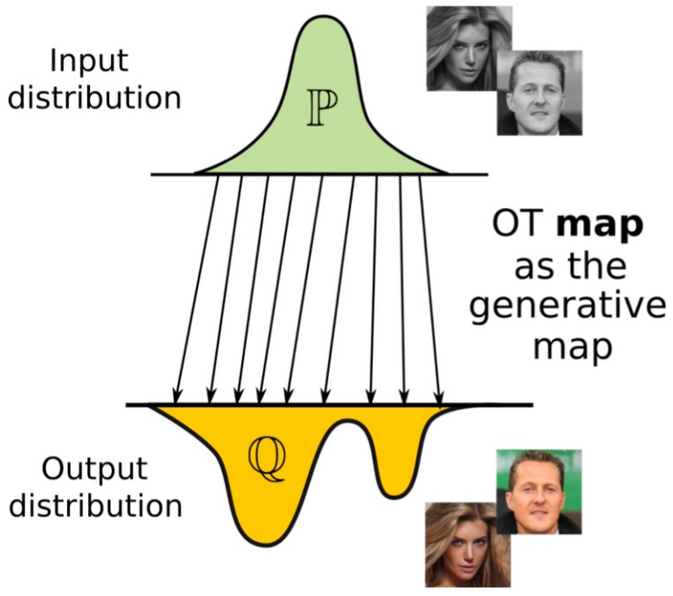


OT for Generative Modeling

OT cost as the loss (WGANs)[2]



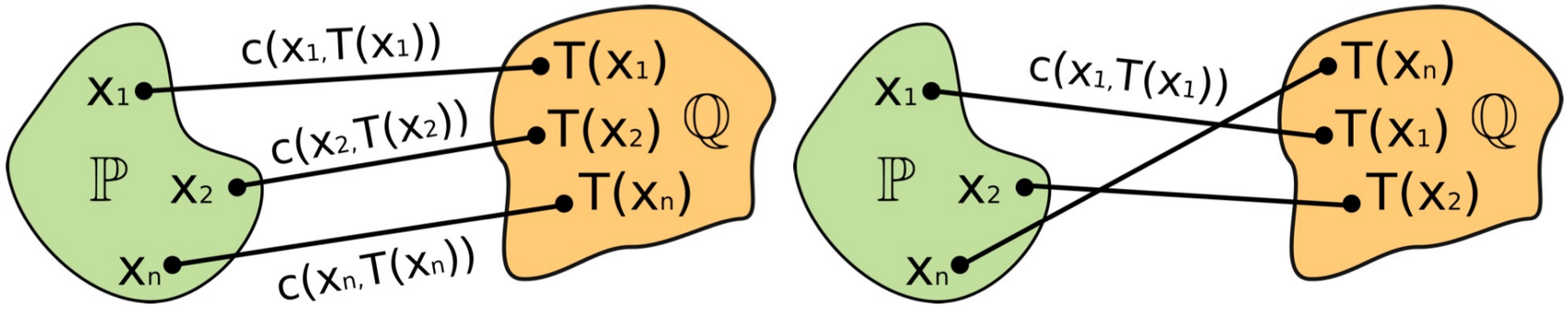
OT map



[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou (2017). "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR, pp. 214–223.

Optimal Transport

Let $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a cost function, e.g., $c(x, y) = \frac{\|x-y\|^2}{2}$.



The optimal transport **cost** between measures $\mathbb{P}$ and $\mathbb{Q}$ is

$$\mathrm{Cost}(\mathbb{P}, \mathbb{Q}) = \inf_{T \sharp \mathbb{P} = \mathbb{Q}} \int_{\mathcal{X}} c\big(x, T(x)\big) d\mathbb{P}(x).$$

The map T^* attaining the minimum is called the optimal **transport map**.

[1] Cédric Villani (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

Outline

- Motivation
- Supervised Learning
- Neural Networks
- Unsupervised Learning and Generative Modeling
- **What’s next**

ML pipeline

1. Decompose an applied problem

2. “Formulate” biases

3. Define
   - ✓ Features for object description
   - ✓ Method/Function class
   - ✓ Loss function
   - ✓ Validation approach

What’s next?

1. Lecture on Overview of Science-Informed ML

2. Seminar based on Sci-ML problem

3. Lecture on Applied Use-Cases of Sci-ML
   - ✓ Super-resolution of weather forecasts
   - ✓ Sea Ice Regional Forecasting
   - ✓ Fusion of heterogeneous data for modeling of oil-fields. Geological realism

Questions?

Active Learning