
MACHINE LEARNING

CHAPTER 0: INTRODUCTION

AlphaGo: 2016

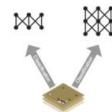
AlphaGo vs. 李世石



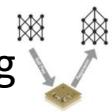
AlphaGo vs. 柯洁



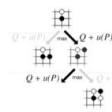
Supervised learning



Reinforcement learning



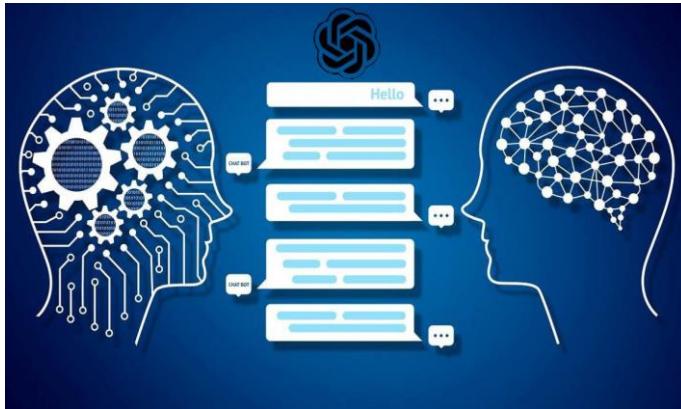
Priori knowledge



Better network **structure**

Enhance the role of **Reinforcement learning**

ChatGPT: 2023



Generative

VS



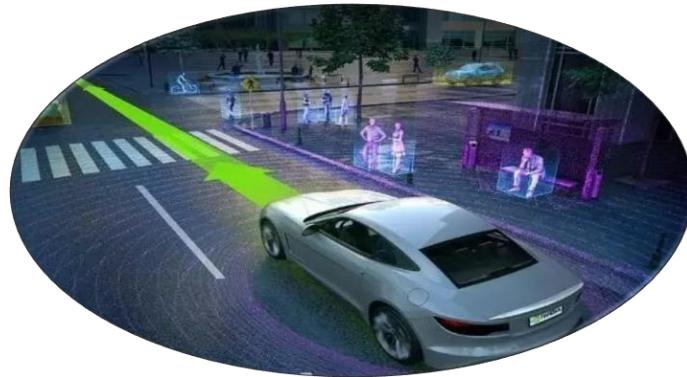
Recommended

- AI-powered large language model (**Strong AI**)
 - Generating human-like text in response to input
 - Question-answering, composition, coding, text completion, and conversation
 - Trained on a large corpus of text data and to be widespread
-

ML Applications



AI Medicine



Autonomous Driving



AI Robotics



Human-Machine Interface

Autonomous Driving

2014 Google



2016 Tesla



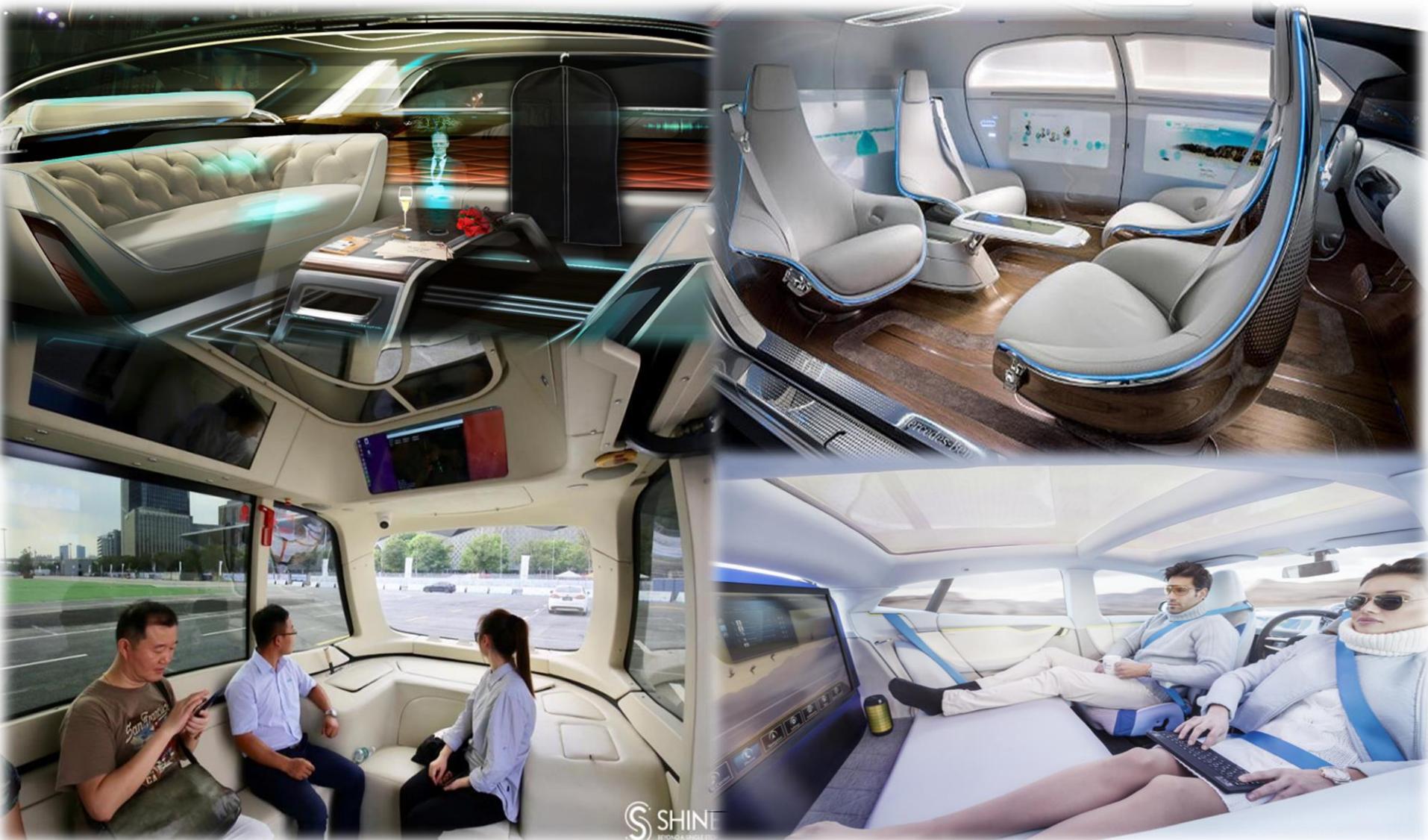
2017 apollo



Future of Autonomous Driving



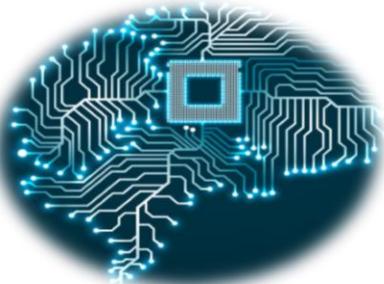
Personal Mobile Space



New Life Styles



Human Evolution



- 人机一体
- 脑机芯片

Contact Information

Instructor:	Qi Hao
E-mail:	hao.q@sustc.edu.cn
Office:	South Tower Room 415
Office Hours:	M 2:00-4:00pm
Available other times by appointment or the open door policy	
Phone:	186-6495-7027
QQ:	415904617 2023机器学习
Web:	http://hqlab.isus.tech/teaching/CS405
BB:	Machine Learning Fall 2023
OJ:	http://oj.isus.tech/

Class Schedule

- **Lectures:** T 8:00 am – 9:50 am Business School Room 101/206
- **Study:** M 7:00 pm – 8:50 pm Business School Room 206
- **Grading policy:**

Final Exam (in-class):	20%	Midterm Exam (take-home):	10%
Assignments (8~12 times):	15%	Quizzes (<=10 times):	10%
Lab Projects :	25%	Final Projects (4 per group):	20%
Bonus Credits:	<=5%		

90~93: A-	94~97: A	98~100: A+
80~82: B-	83~86: B	87~89: B+
70~72: C-	73~76: C	77~79: C+
60~62: D-	63~66: D	67~69: D+

Textbook and Lecture Notes

Textbooks:

- [1] Pattern Recognition and Machine Learning, by Christopher M. Bishop, 2006 Springer
- [2] Machine Learning in Action, by Peter Harrington, 2012, Manning

Other books:

- [1] 机器学习, 周志华
- [2] Dive in Deep Learning, by Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola
- [3] Reinforcement Learning: An Introduction, by Richard S. Sutton
- [4] The Elements of Statistical Learning, by Trevor Hstie, Rober Tibshirani, Jerome Friedman

Paper reading:

- [1] Ghahramani Z. Probabilistic machine learning and artificial intelligence, Nature, 2015
- [2] Lecun Y, Bengio Y, Hinton G. Deep learning, Nature, 2015
- [3] Littman M L. Reinforcement learning improves behavior from evaluative feedback, Nature, 2015

Lecture notes:

<http://hqlab.isus.tech/teaching/CS405>

Other Resources

Assignment platform: bb.sustech.edu.cn

Textbook resource: <https://www.microsoft.com/en-us/research/people/cmbishop/#prml-book>

Textbook Python codes :

<https://github.com/ctgk/PRML>

Teaching Objectives

- Fundamental knowledge about machine learning and pattern recognition, from Bayesian approaches to deep learning frameworks through lectures, quizzes and exercises
- Machine learning system development methods in Python based platforms (numpy, sciki-learn, pytorch) through labs and projects
- Model-based and data-driven machine learning system design and integration skills through the final project, literature surveys and reports

Lecture Schedule

Section 0	Course Introduction	
Section 1	Preliminary	(HW1)
Section 2	Probability Distributions	(HW2)
Section 3	Linear Regression and Classification	(HW3)
Section 4*	Dimension Reduction and Feature Selection	(HW4)
Section 5	Neural Networks	(HW5)
Section 6	Sparse Kernel Machine	(HW6)
Section 7	Clustering and EM learning	(HW7)
<i>-- Midterm Exam --</i>		
Section 8*	Ensemble Learning	(HW8)
Section 9	Hidden Markov Models	(HW9)
Section 10*	Bayesian Networks	(HW10)
Section 11	Markov Decision Process	(HW11)
Section 12*	Reinforcement Learning	(HW12)
<i>-- Final Exam --</i>		

* means learning by yourselves

Lab Schedule

Section 0 Lab Introduction

Section 1 Preliminary

Section 2 Bayes

Section 3 Regression and Classification

--*Final Project Proposal*--

Section 4 Decision Tree

Section 5 Random Forest (Ensemble Learning)

Section 6 KNN and Support Vector Machine

Section 7 K-Mean and EM Clustering

Section 8 Neural Network (I)

Section 9 Neural Network (II)

Section 10 Neural Network (III)

Section 11* Reinforcement Learning

--*Final Project Report*--

Final Project Examples

- [1] Reinforcement learning based planning using a self-driving car simulator
 - [2] Segmentation of 2D/3D measurements for self-driving applications
 - [3] Detection and recognition of traffic signs for self-driving applications
 - [4] Detection and tracking of 2D/3D objects for self-driving applications
 - [5] Federated learning for model fusion of networked vehicle applications
 - [6] GNN for self-driving data augmentation
-

Bonus Credits

- AI companies
- Survey Papers
- Attendance
- Bonus Credits



Plagiarism

From Spring 2018, the plagiarism policy applied by the Computer Science and Engineering department is the following:

- * If an assignment is found to be plagiarized, the first time the score of the assignment will be 0.
- The second time the score of the course will be 0.

As it may be difficult when two assignments are identical or nearly identical who actually wrote it, the policy will apply to BOTH students, unless one confesses having copied without the knowledge of the other.

What is OK, and what isn't OK

It's OK

- to work on an assignment with a friend, and think together about the program structure, share ideas and even the global logic. At the time of actually writing the code, you should write it alone.
- to use in an assignment a piece of code found on the web, as long as you indicate in a comment where it was found and don't claim it as your own work.
- to help friends debug their programs (you'll probably learn a lot yourself by doing so).
- to show your code to friends to explain the logic, as long as the friends write their code on their own later.

It's NOT OK

- **to take the code of a friend, make a few cosmetic changes (comments, some variable names) and pass it as your own work.**
-

Make a Promise to Keep

Sign

the “Student Commitment for Assignments”

Keep

the promise during the whole semester!

Learning Objectives

1. What is the history of machine learning?
 2. What are the most important functionalities of machine learning?
 3. What are the major technical challenges for developing machine learning systems?
 4. What are the most useful tools for developing machine learning systems?
 5. What are the most popular software and hardware platforms for developing machine learning systems?
 6. What are the most promising applications for machine learning?
-

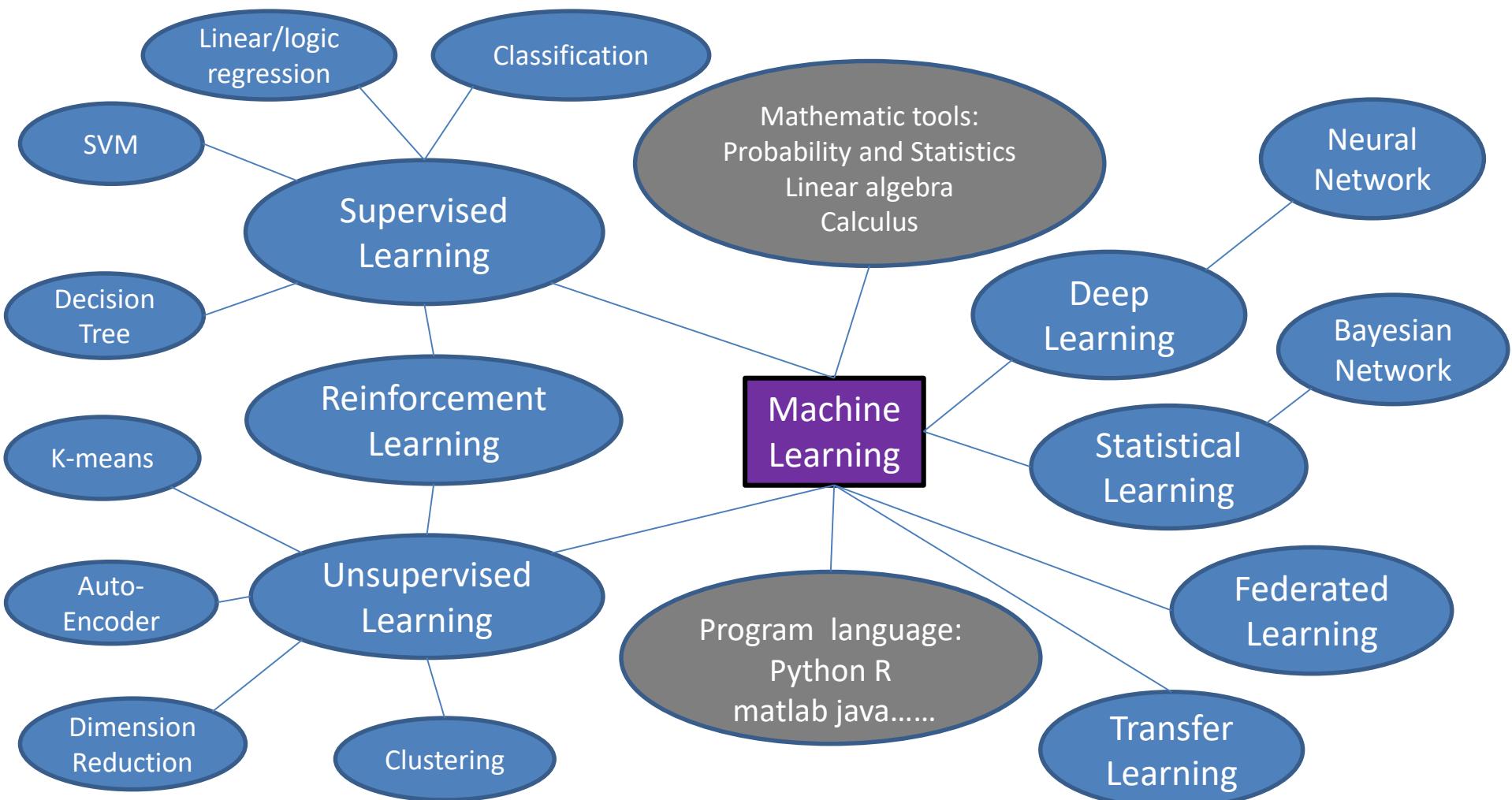
Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

Framework

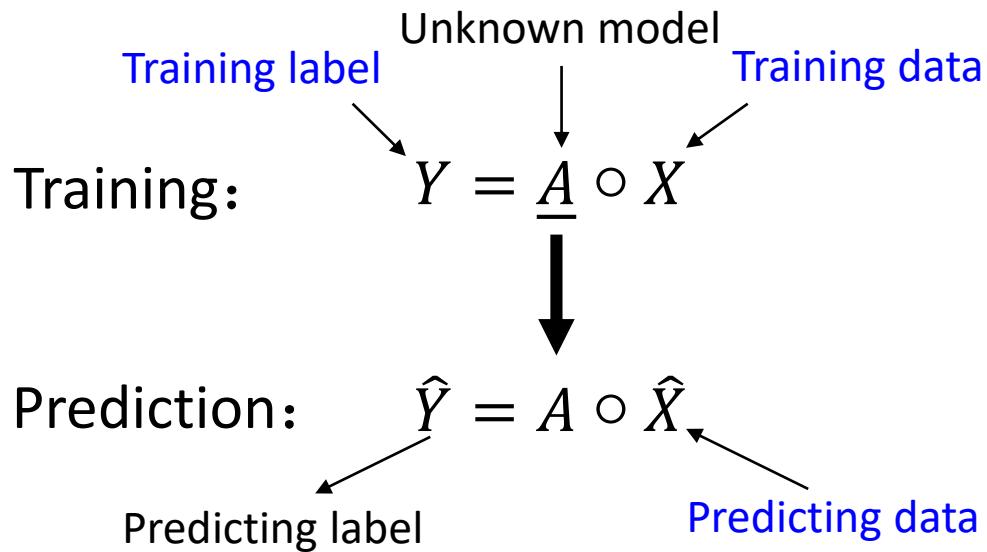


Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

Problem Statement

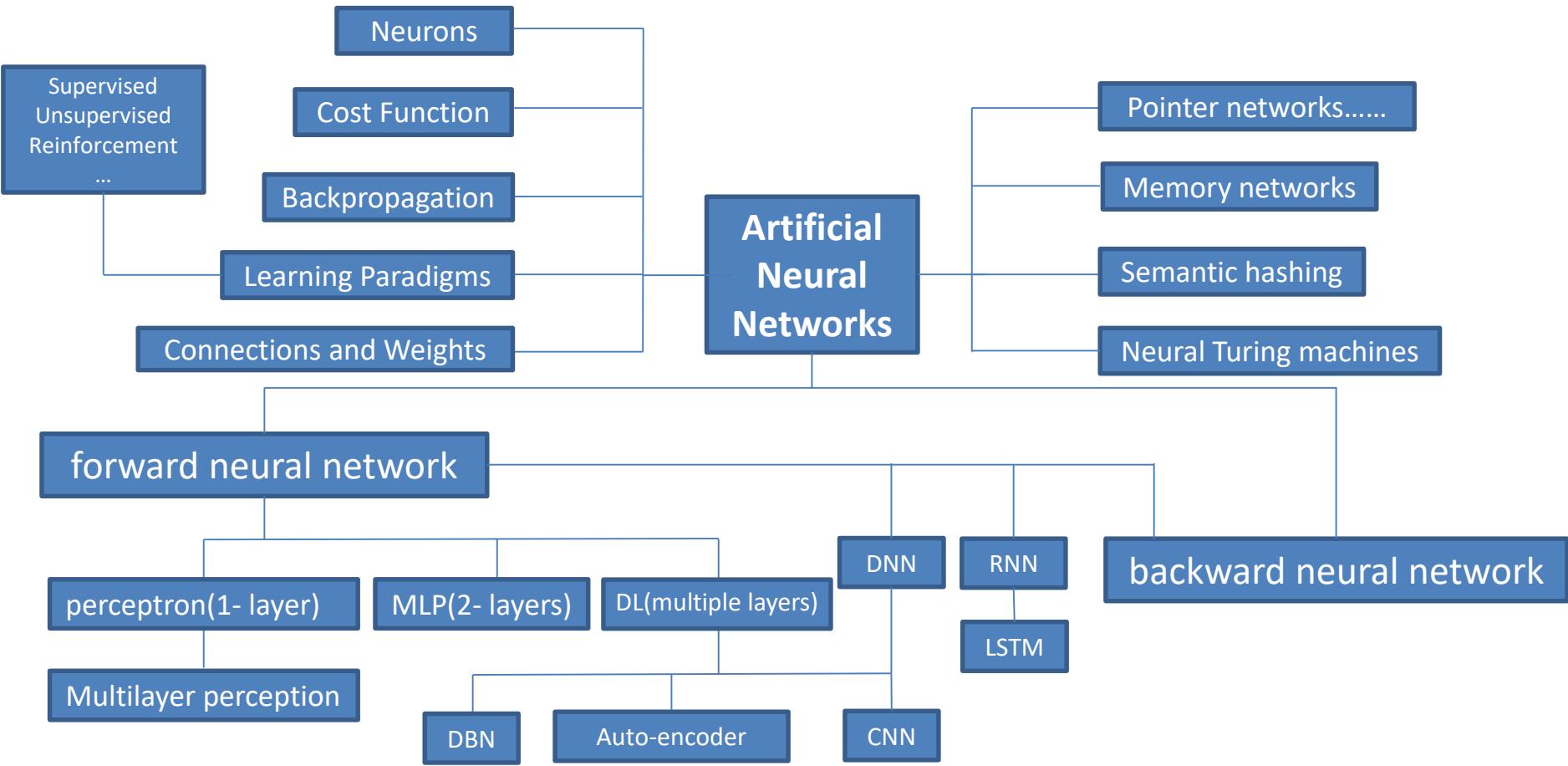
- Problem: Predict the label \hat{Y} and data \hat{X} with training set (X, Y) ?



$\begin{cases} Y \text{ and } X \text{ is known: supervised learning} \\ Y \text{ or } X \text{ is unknown: unsupervised learning} \end{cases}$ $\begin{cases} Y, \hat{Y} \text{ are continuous: Regression} \\ Y, \hat{Y} \text{ are discrete: classification} \end{cases}$

Y is known and $\text{Dim}(Y) > \text{Dim}(X)$: dimensionality reduction

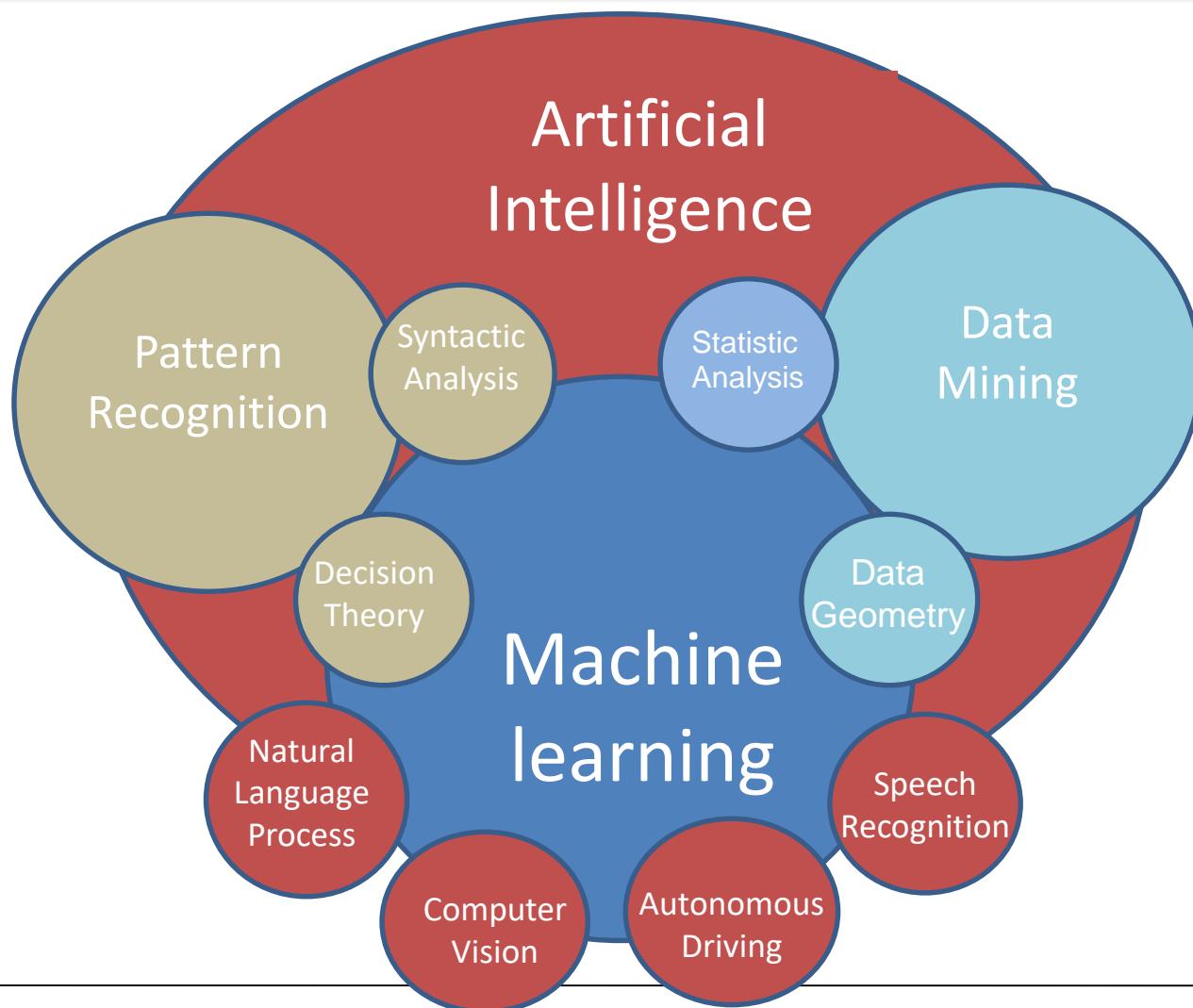
Neural Network Models



Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

The Whole Picture

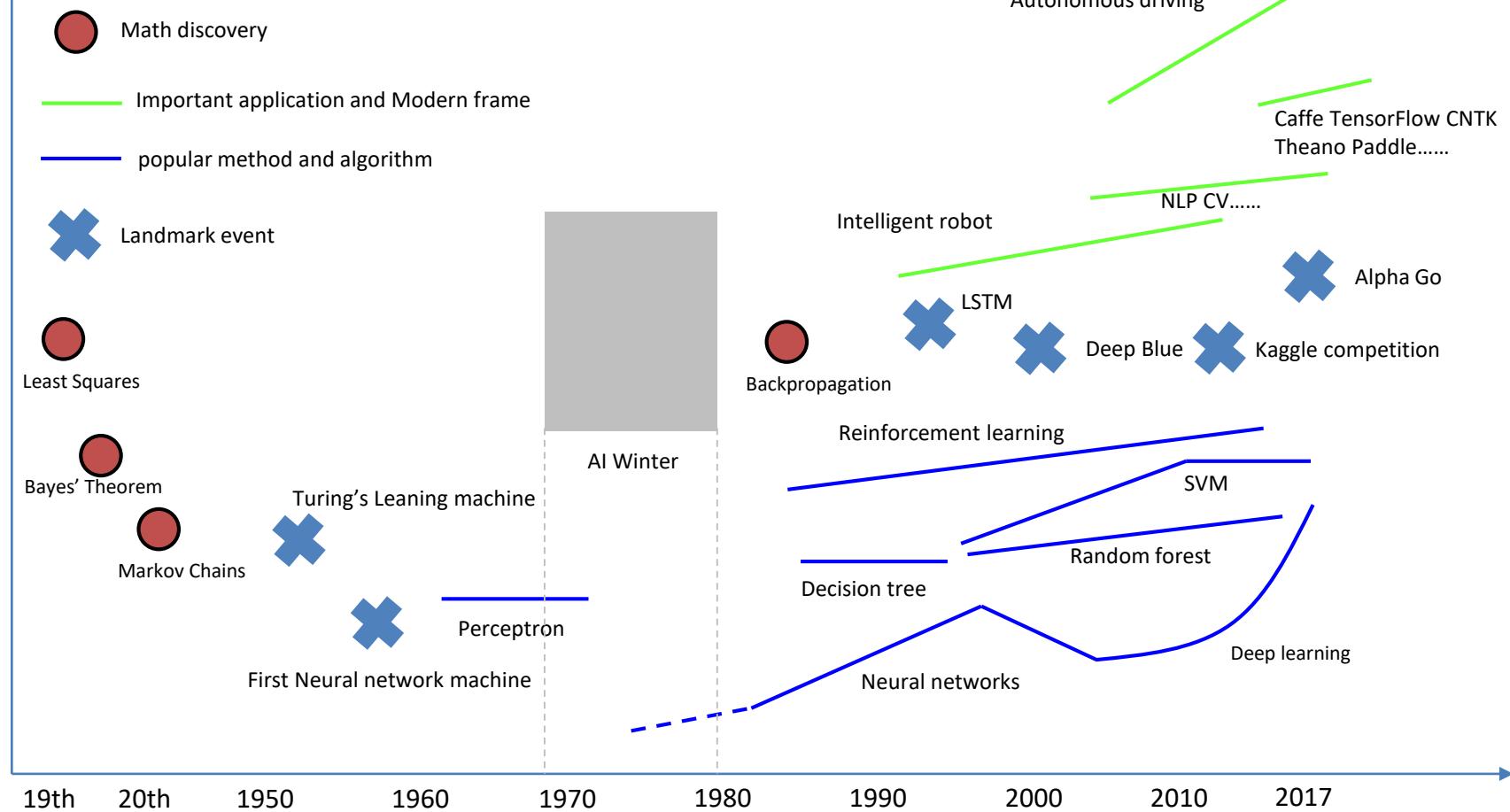


Outlines

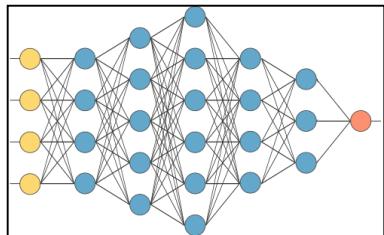
- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

History

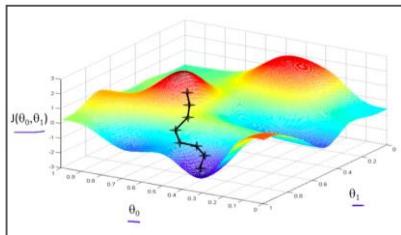
Popularity and degree of application



Deep ML vs Conventional ML



model



optimization

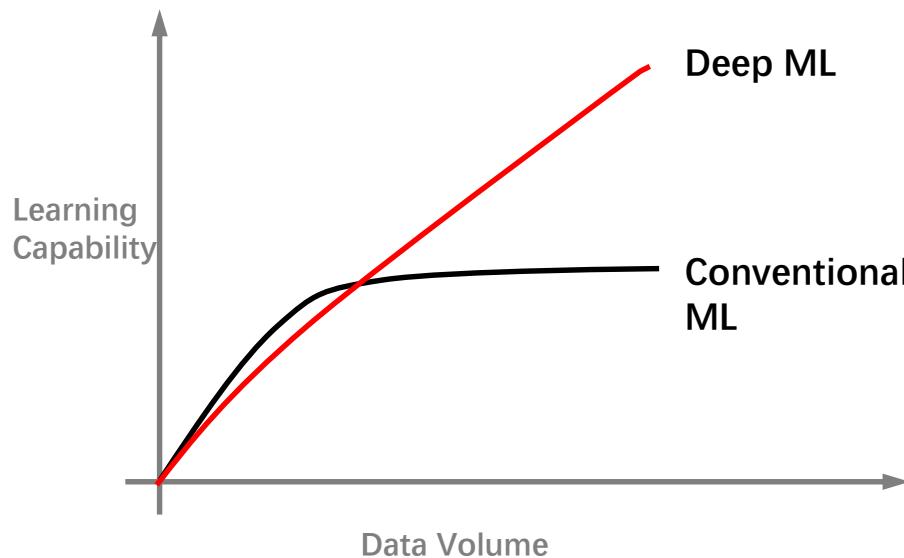


data



platform

CNN
RNN
LSTM
Auto Encoder
GAN
RL & FL



GPU & TPU
Cloud Computing
Data Visualization
GPU Service
TensorFlow
Caffe & Pytorch

Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

Machine learning

- **Machine Learning**—minimization of some loss function for generalizing data sets with models.

- **Datasets** —annotated, indexed, organized

- **Models** —tree, distance, probabilistic, graph, bio-inspired

- **Optimization** —algorithms can minimize the loss.

Datasets

- Collection
 - Storage
 - Annotation
 - Indexing
 - Organization
 - Access
-

Simulators

- Data visualization
- Generate training data
- Algorithm evaluation



Benchmark Metrics

- System functionalities
- System scalability
- System robustness
- System efficiency

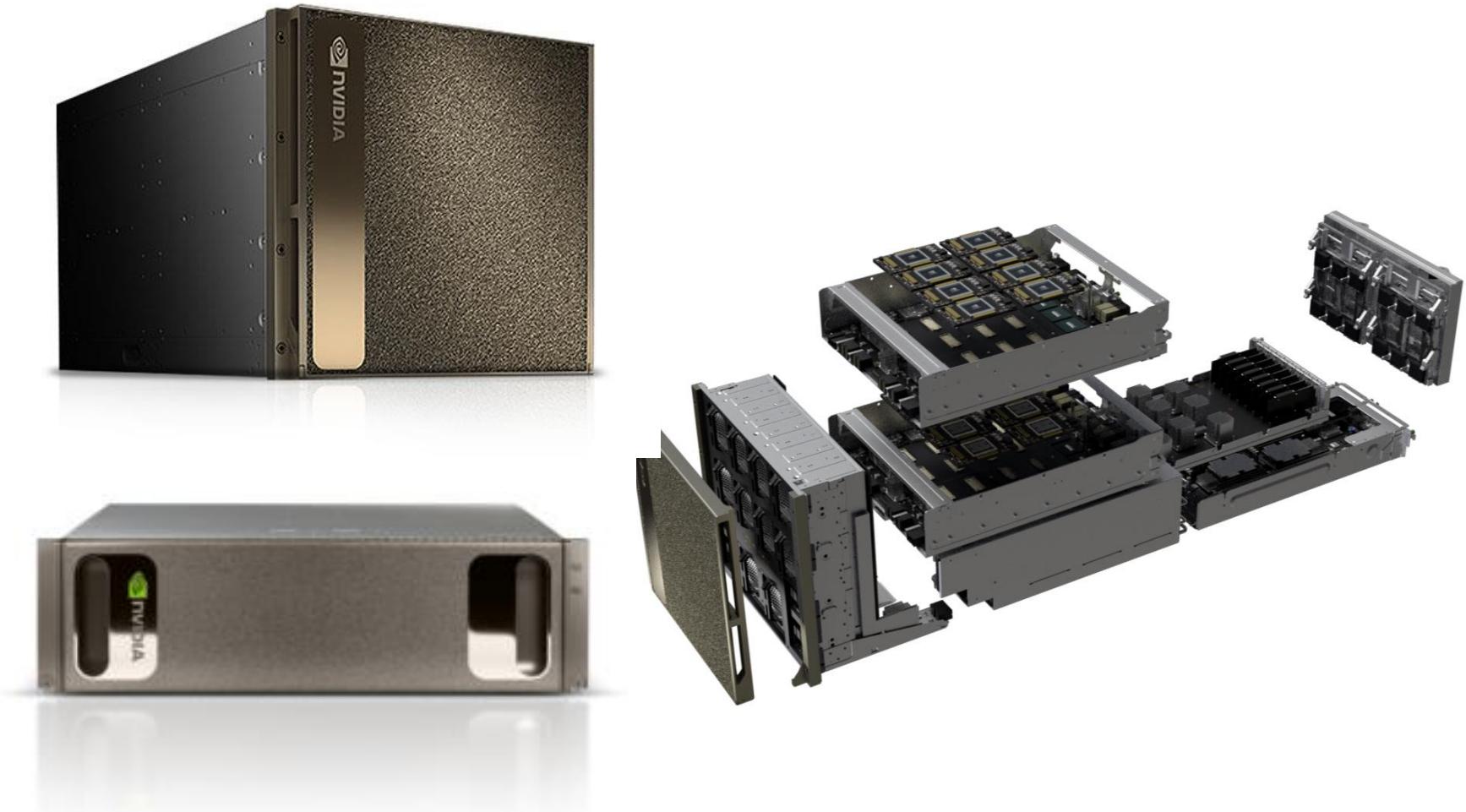
Models

- Tree Models
- Distance-based Models
- Probabilistic Models
- Neural Network Models
- Graph-based Models

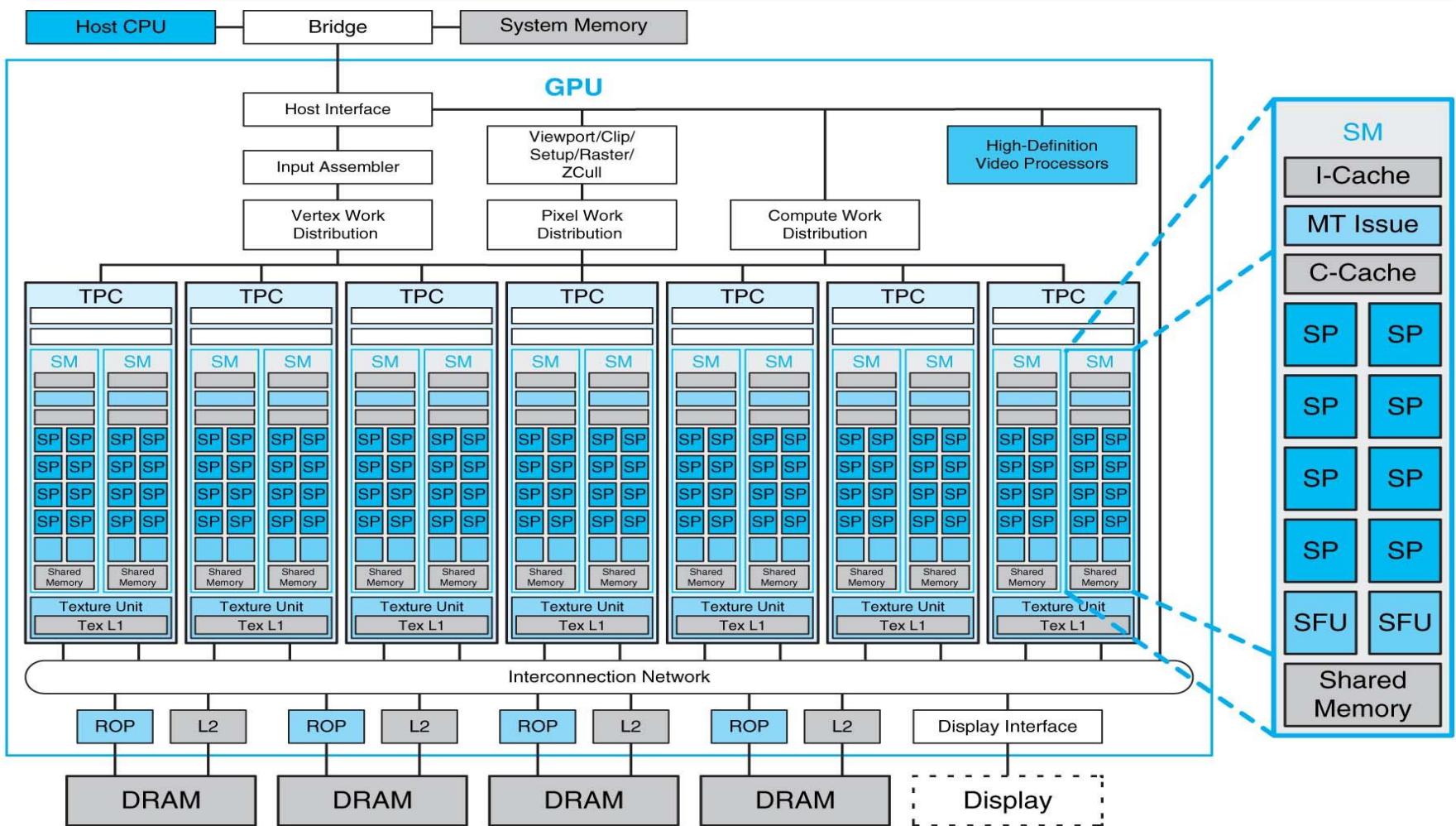
Models

- Boosting
 - Mixed Models
 - Ensemble Learning
-

Hardware Platform (GPU Server)



Hardware Platform (GPU)



构筑业界最强AI算力平台，极简易用、极致性能

行业应用

智慧城市、制造、能源、交通、金融、运营商、教育等更多行业应用



ModelArts



HiAI Service



第三方平台



全流程开发工具链



FusionDirector/SmartKit



昇腾社区 hiascend.com

应用使能

MindX 昇腾应用使能



MindX DL
深度学习使能



MindX Edge
智能边缘使能



ModelZoo
优选模型库



MindX SDK
行业SDK

AI框架

[M]^s MindSpore

最佳匹配昇腾AI处理器算力的全场景AI计算框架

TensorFlow/PyTorch等第三方框架

可基于第三方框架开发的模型进行二次开发、训练和推理

CANN

统一异构计算架构，释放昇腾硬件澎湃算力

异构计算架构

系列硬件



Atlas系列硬件打造人工智能算力平台基石

Atlas训练系列硬件



Atlas 300T训练卡
单卡算力**业界领先**

320 TFLOPS FP16



Atlas 800
训练服务器



Atlas 900 PoD
Atlas 900 AI集群

Atlas推理系列硬件



Atlas 300I 推理卡



Atlas 800
推理服务器

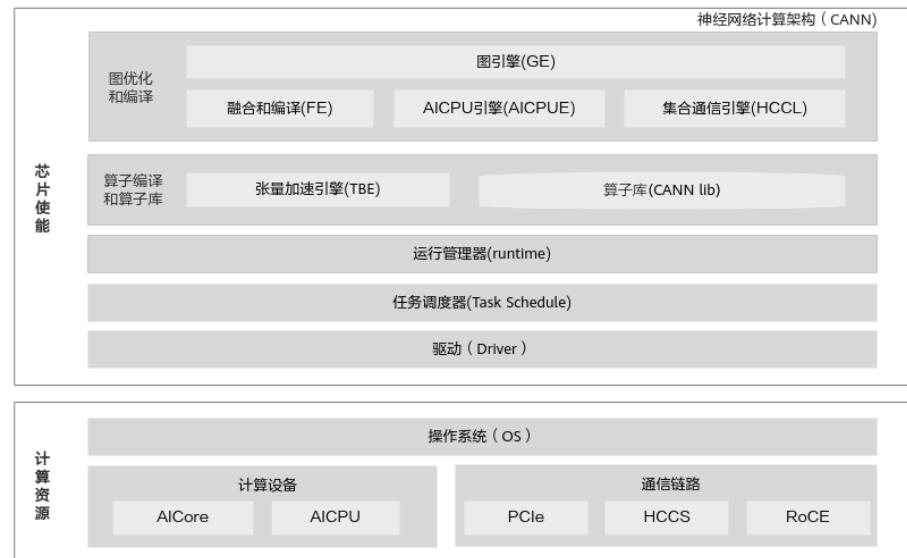


Atlas 200
AI加速模块
Atlas 500
智能小站



Atlas 200DK
Atlas 500 Pro
智能边缘服务器

异构计算架构CANN，软硬协同充分释放澎湃算力



[M]^s 开源AI框架MindSpore，构建端边云全场景生态



简单的开发体验

帮助开发者实现网络自动切分，只需串行表达就能实现并行训练，降低门槛，简化开发流程。



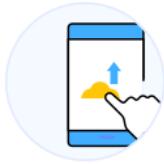
灵活的调试模式

具备训练过程静态执行和动态调试能力，开发者通过变更一行代码即可切换模式，快速在线定位问题。



充分发挥硬件潜能

最佳匹配昇腾处理器，最大程度地发挥硬件能力，帮助开发者缩短训练时间，提升推理性能。



全场景快速部署

支持云、边缘和手机上的快速部署，实现更好的资源利用和隐私保护，让开发者专注于AI应用的创造。



全自动并行

静态图自动混合并行
训练**性能提升40%**

动态图优化
性能**超越业界60%**



全场景协同

云端分布式推理
边缘AI加速
超轻量IoT设备推理



全流程极简

第三方框架转换工具
业务快速迁移

开发者生态

51万+ 2300+

下载量

社区贡献者

开放AI应用使能套件MindX，加速人工智能应用创新

MindX：昇腾应用使能

MindX DL

深度学习使能

MindX Edge

智能边缘使能

MindX SDK

行业应用开发套件

ModelZoo

250+ 预训练模型

MindX SDK：沉淀行业知识，使能行业应用 极简开发



2人月

传统应用开发方式

2人天

基于SDK开发方式

已支撑 20+ 场景化解决方案高效开发

华为松山湖产线

PCB板质检

友达光电

切片AOI检测

南瑞继远

变电站
火警检测
变电站
人员着装检测





一站式开发环境MindStudio，打造高效、便捷的全流程开发工具链

MindStudio是一套基于IntelliJ框架的开发工具平台。提供了应用开发、调试、模型转换功能，同时还提供了网络移植、优化和分析等功能，为用户开发应用程序带来了极大的便利。

The screenshot shows the IntelliJ-based MindStudio IDE interface. The left sidebar displays a project structure for 'MyOperator1' with files like 'main.cpp' and 'op_runner.cpp'. The main window contains two code editors. The top editor shows code related to device handling and operator descriptions. The bottom editor shows code for defining input tensors. At the bottom, there are tabs for Output, Log, Problems, TODO, Profiling, File Transfer, Remote Terminal, and Terminal.

```
7 * but WITHOUT ANY WARRANTY; without even the implied warranty of
8 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
9 */
10 #include <...>
11
12 bool g_isDevice = false;
13 int deviceId = 0;
14
15 OperatorDesc CreateOpDesc()
16 {
17     std::string opType = "GatherV2";
18
19     std::vector<int64_t> shape1{100};
20     std::vector<int64_t> shape2{10};
21     std::vector<int64_t> shape3{1};
22     std::vector<int64_t> shape4{10};
23
24     OperatorDesc opDesc(opType);
25
26     opDesc.AddInputTensorDesc(ACL_INT32, shape1.size(), shape1.data(), ACL_FORMAT_ND);
27     opDesc.AddInputTensorDesc(ACL_INT32, shape2.size(), shape2.data(), ACL_FORMAT_ND);
28     opDesc.AddInputTensorDesc(ACL_INT32, shape3.size(), shape3.data(), ACL_FORMAT_ND);
29
30     OperatorDesc opDesc(opType);
31
32     opDesc.AddInputTensorDesc(ACL_INT32, shape1.size(), shape1.data(), ACL_FORMAT_ND);
33     opDesc.AddInputTensorDesc(ACL_INT32, shape2.size(), shape2.data(), ACL_FORMAT_ND);
34     opDesc.AddInputTensorDesc(ACL_INT32, shape3.size(), shape3.data(), ACL_FORMAT_ND);
35
36 }
```

- 训练脚本转换
- 模型转换
- 精度比对
- Profiling性能分析
- System Profiling工具
- AI Core Error分析工具

应用开发

- 基于MindX SDK开发应用
- 基于新工程开发应用
- 应用工程调试

模型开发

- 查询模型
 - 模型可视化
 - 模型训练
-
- 算子开发
 - 查询算子
 - 开发流程
 - 算子分析
 - 工程创建
 - TBE算子开发 (TensorFlow)
 - TBE算子开发 (MindSpore)
 - AI CPU算子开发 (PyTorch)
 - AI CPU算子开发 (TensorFlow)

Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

Machine learning and Optimization

- **Machine Learning**—minimization of some loss function for generalizing data sets with models.

- **Datasets** —annotated, indexed, organized

- **Models** —tree, distance, probabilistic, graph, bio-inspired

- **Optimization** —algorithms can minimize the loss.

What is optimization?

- Finding (one or more) minimizer of a function subject to constraints

$$\arg \min_x f_0(x)$$

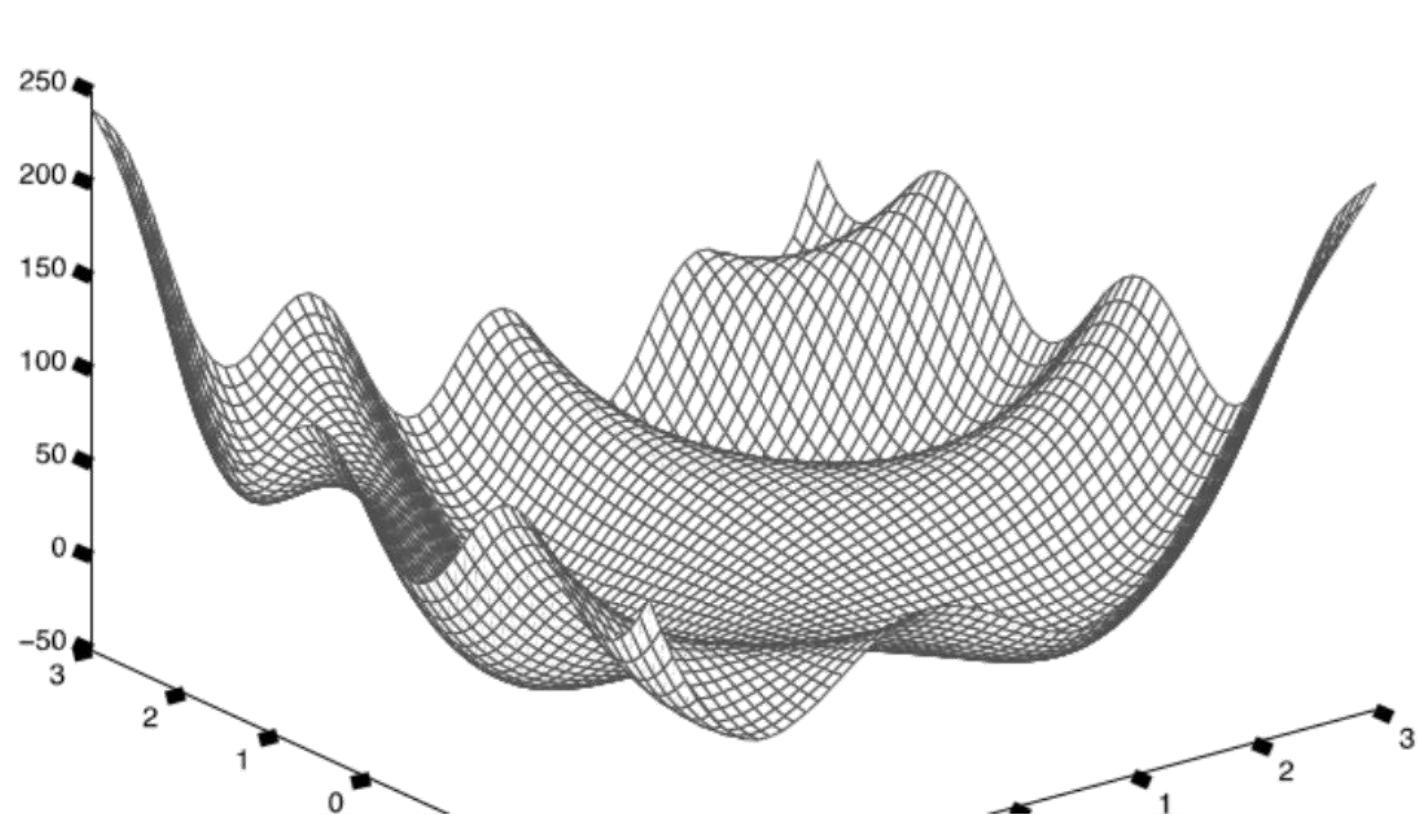
$$s.t. f_i(x) \leq 0, i = \{1, \dots, k\}$$

$$s.t. h_j(x) = 0, j = \{1, \dots, l\}$$

- Most of the machine learning problems are, in the end, optimization problems
-

General Problem

■ Minimize $f(x)$



Linear Optimization

$$Y = AX + w$$

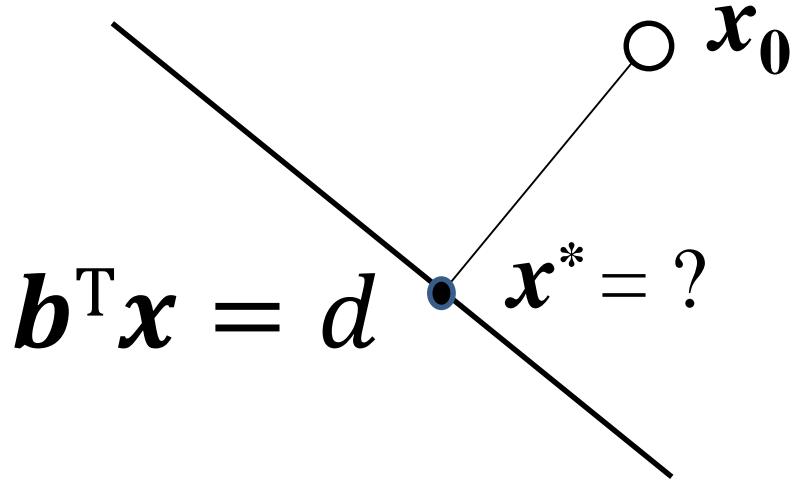
$$w \sim \mathcal{N}(0, R)$$

$$X^* = \min_X (Y - AX)^T R^{-1} (Y - AX)$$

$$\frac{\partial}{\partial X^T} (Y - AX)^T R^{-1} (Y - AX) = 0$$

$$\Rightarrow X^* = (A^T R^{-1} A)^{-1} A^T R^{-1} Y$$

Linear Optimization



$$\mathbf{x}^* = \mathbf{x}_0 - \frac{(\mathbf{b}^T \mathbf{x}_0 - d)\mathbf{b}}{\mathbf{b}^T \mathbf{b}}$$

$$\mathbf{x}^* = \min_{\mathbf{x}} (\mathbf{x} - \mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_0)$$

$$\text{s. t. } \mathbf{b}^T \mathbf{x} - d = 0$$

Nonlinear Optimization

■ Convex Optimization

- Unconstrained optimization
- Constrained optimization
- SVMs and Bayesian models

■ Non-convex Optimization

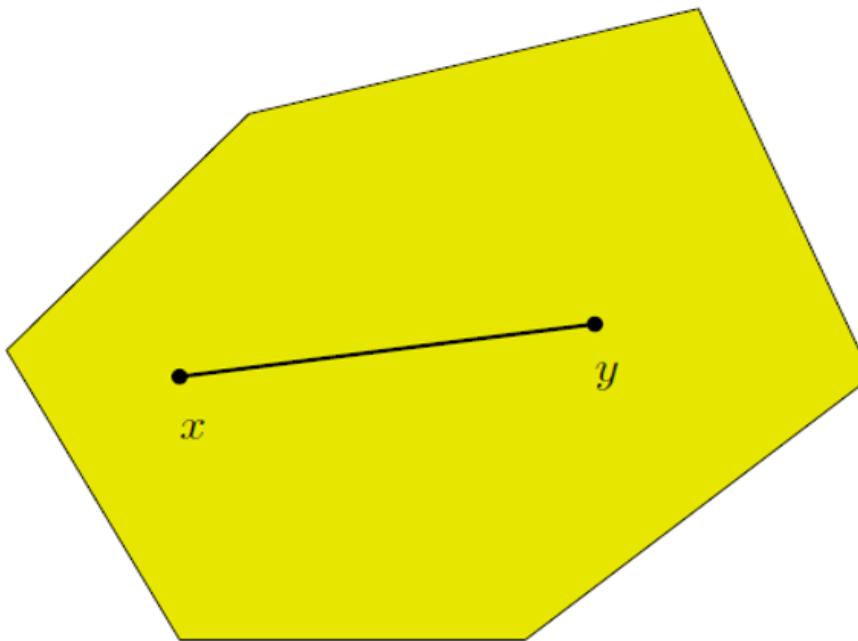
- Heuristic algorithms
 - Random search
-

What is Convex?

■ Convex sets

Def: A set $C \subseteq \mathbb{R}$ is convex if for $x, y \in C$; $a \in [0, 1]$

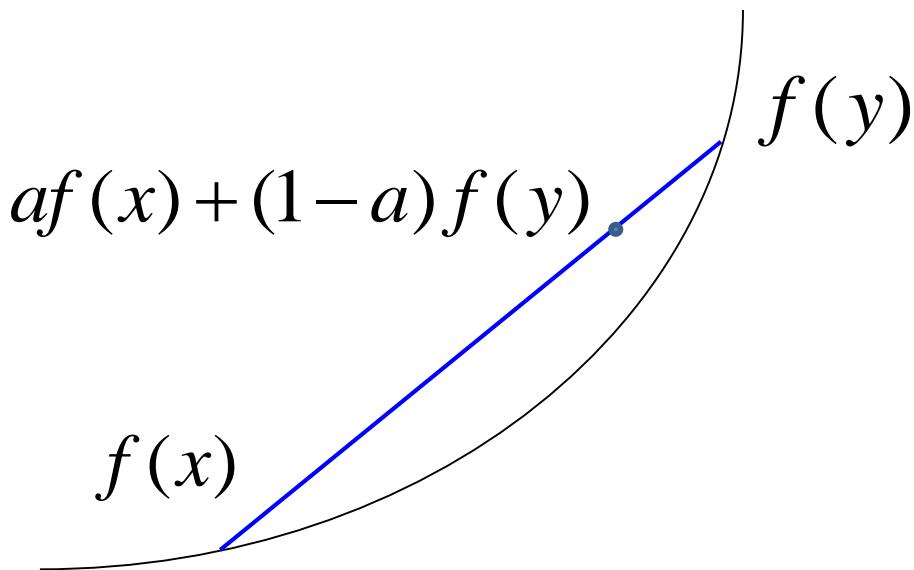
$$ax + (1 - a)y \in C$$



What is Convex?

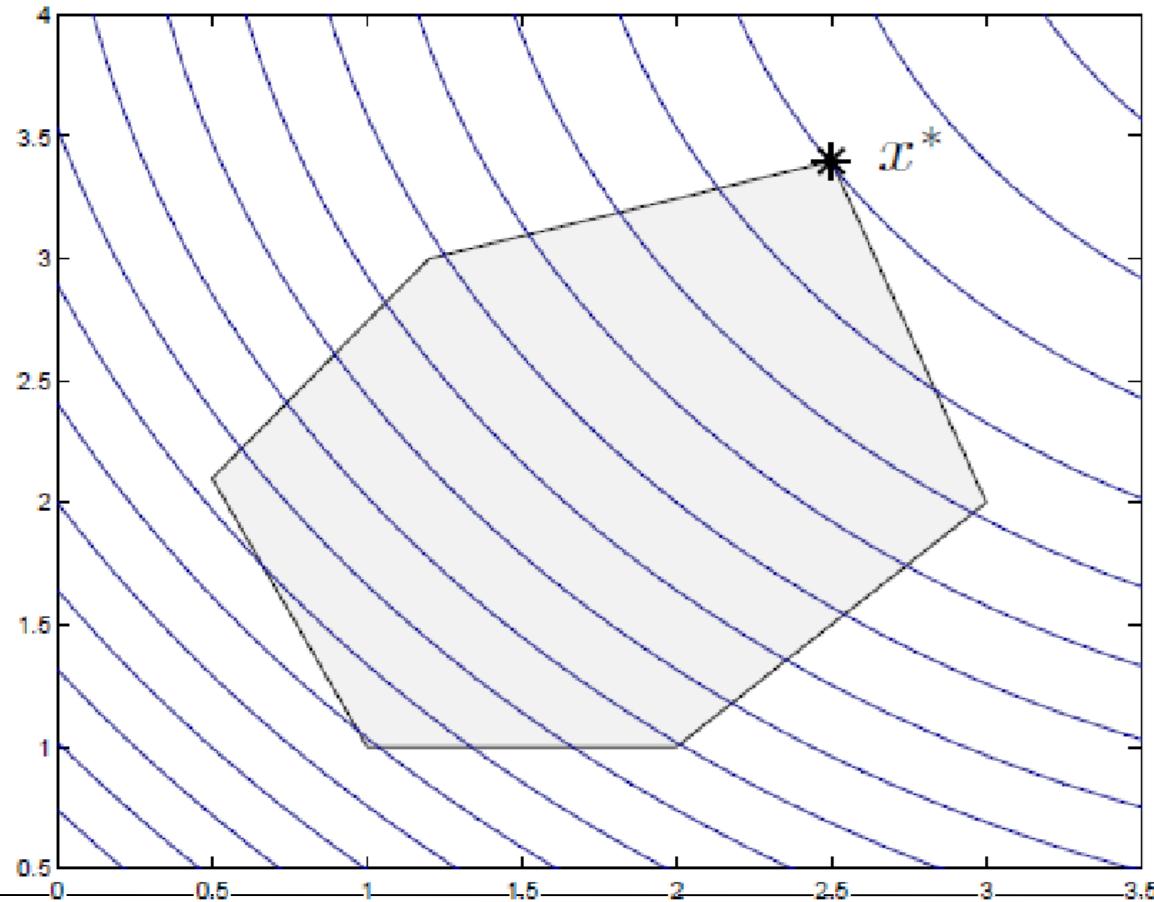
■ Convex functions

$$f(ax + (1-a)y) \leq af(x) + (1-a)f(y)$$



Convex Optimization

- Local minimizer = Global minimizer



Convex Optimization

- Unconstrained optimization
 - Gradient descent
 - Gauss-Newton's method
 - Batch learning
 - Stochastic Gradient Descent

- Constrained optimization
 - Lagrange methods
 - Bayesian methods

Convex optimization

■ Unconstrained optimization

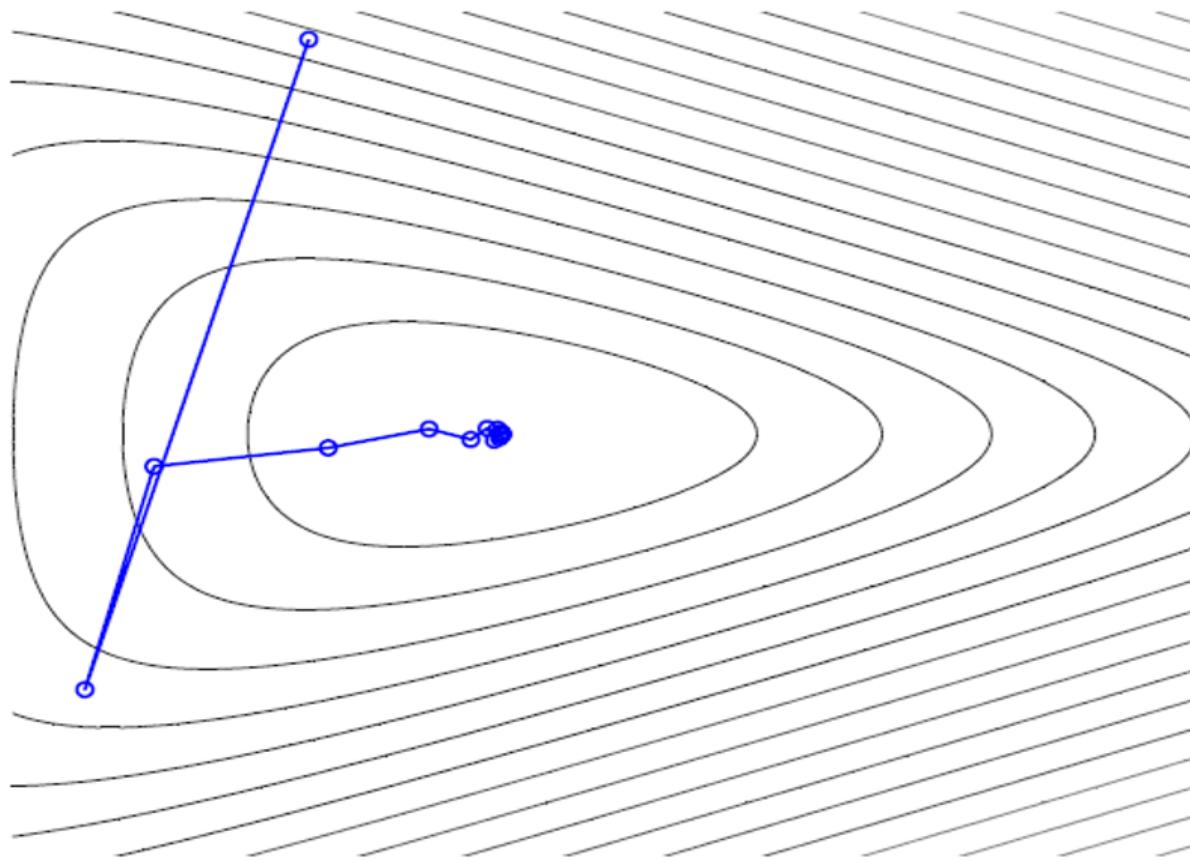
- Gradient descent
- Gauss-Newton's method
- Batch learning
- Stochastic Gradient Descent

■ Constrained optimization

- Lagrange methods
- Bayesian methods

Gradient Descent

$$f(x_{t+1}) = f(x_t) - \eta \nabla f(x_t)^T (x - x_t)$$



Gauss-Newton's Method

- Idea: use a second-order approximation to function

$$f(x + \Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x$$

- Choose Δx to minimize above:

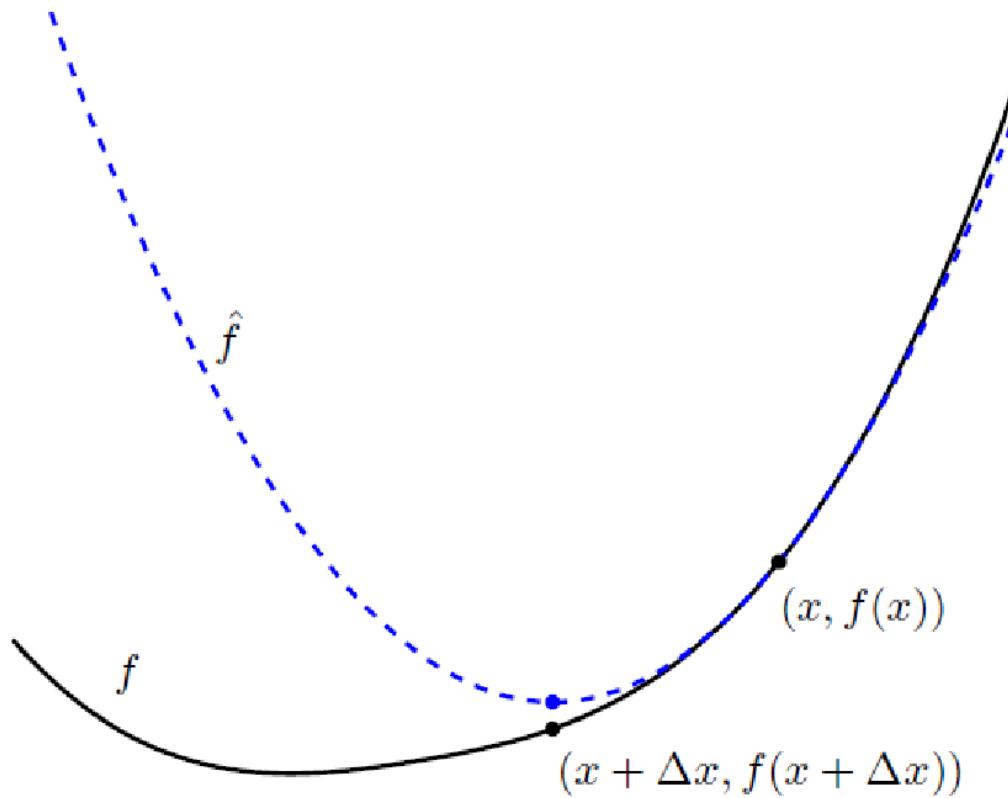
$$\Delta x = -[\nabla^2 f(x)]^{-1} \nabla f(x)$$

- This is descent direction:

$$\nabla f(x)^T \Delta x = -\nabla f(x)^T [\nabla^2 f(x)]^{-1} \nabla f(x) < 0$$

Gauss-Newton's Method

\hat{f} is 2-order approximation, f is true function.



Batch Gradient Descent

- Minimize empirical loss, assuming it's convex and unconstrained

- Gradient descent on the empirical loss
 - At each step:

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \left(\frac{1}{n} \sum_{i=1}^n \frac{\partial L(w, x_i, y_i)}{\partial w} \right)$$

- Note: at each step, gradient is the average of the gradient for all samples ($i=1, \dots, n$)
 - Very slow when n is very large

Stochastic Gradient Descent

- Alternative: compute gradient from just one (or a few samples)
- Known as stochastic gradient descent:
 - At each step,

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

(choose one sample i and compute gradient for that sample only)

Convex Optimization

■ Unconstrained optimization

- Gradient descent
- Gauss-Newton's method
- Batch learning
- Stochastic Gradient Descent

■ Constrained optimization

- Lagrange methods
 - Bayesian methods
-

Lagrange Methods

- Start with an optimization problem:

$$\arg \min_x f_0(x)$$

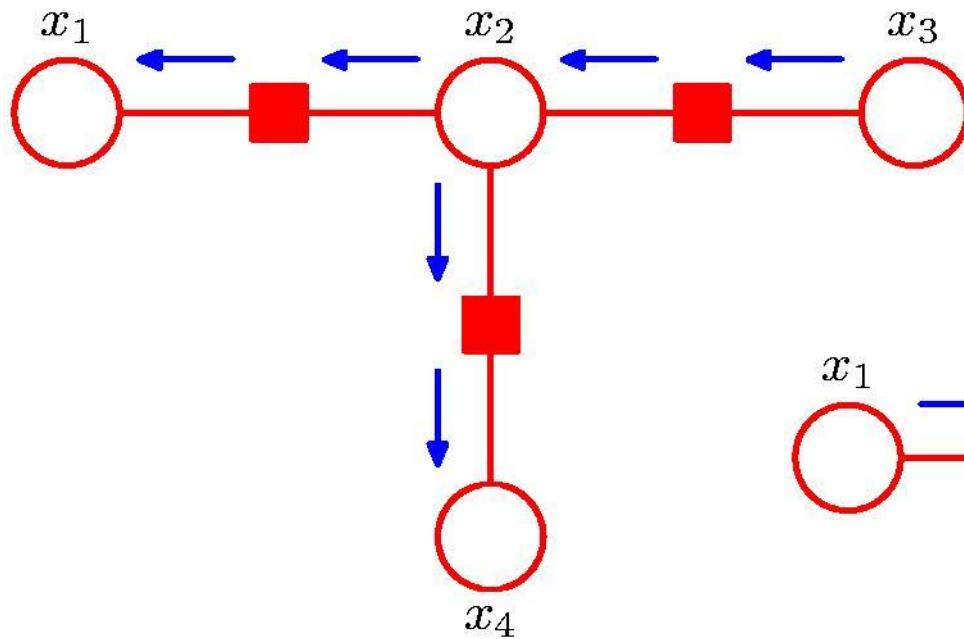
$$s.t. f_i(x) \leq 0, i = \{1, \dots, k\}$$

$$s.t. h_j(x) = 0, j = \{1, \dots, l\}$$

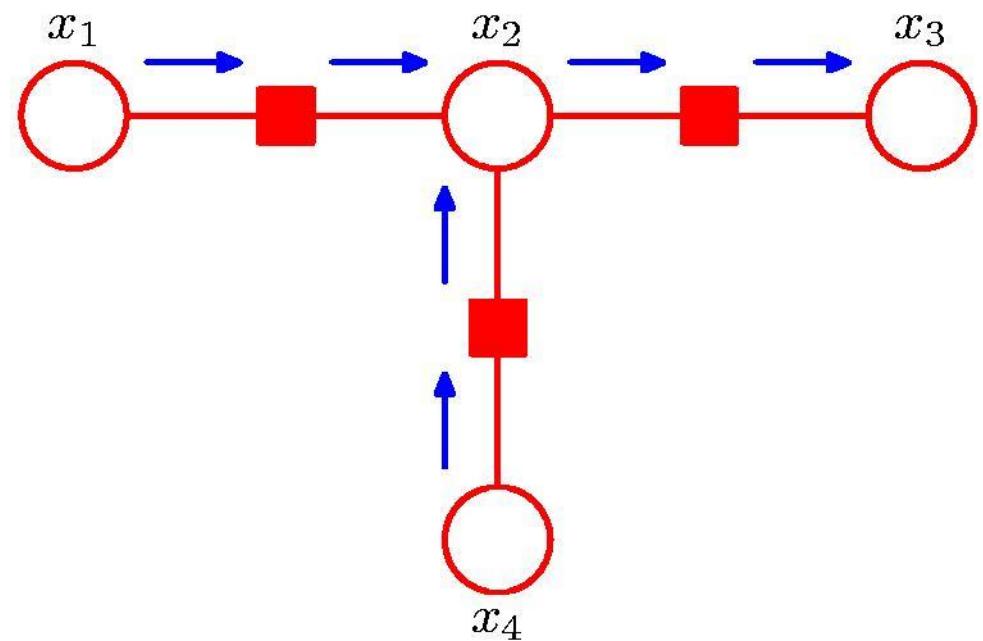
- Is equivalent to min-max optimization:

$$\arg \min_x \left[\max_{\lambda \geq 0, \gamma > 0} \left(f_0(x) + \sum_{i=1}^k \lambda_i f_i(x) + \sum_{j=1}^l \gamma_j h_j(x) \right) \right].$$

Bayesian Methods



Random variable
 factor



Convex Optimization for Machine Learning

■ Gradient Based Methods

- Neural networks

■ Lagrange Methods

- Support vector machines

■ Bayesian Methods:

- Expectation-Maximization methods (mixture models)
- Variational methods (approximate models)
- Graph optimization (belief propagation models)

Non-convex Optimization

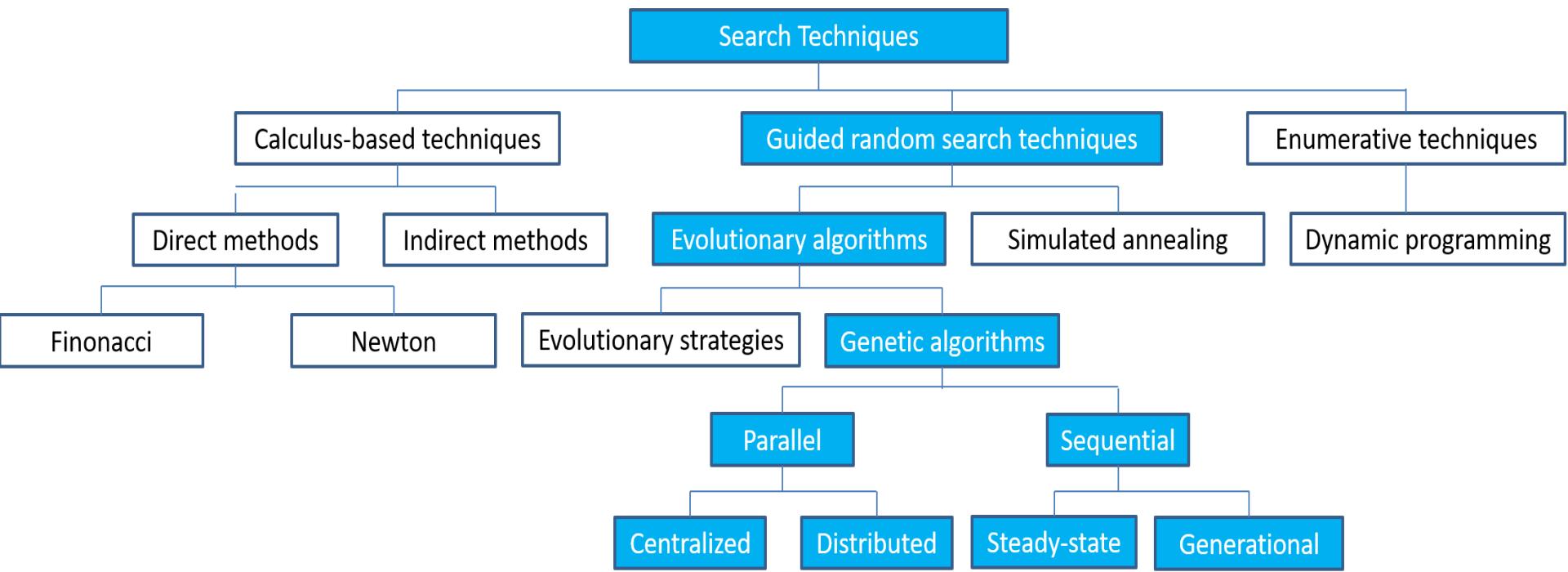
■ Convex Optimization

- Unconstrained optimization
- Constrained optimization

■ Non-convex Optimization

- Heuristic algorithms
 - Random search
-

Heuristic and Random Search



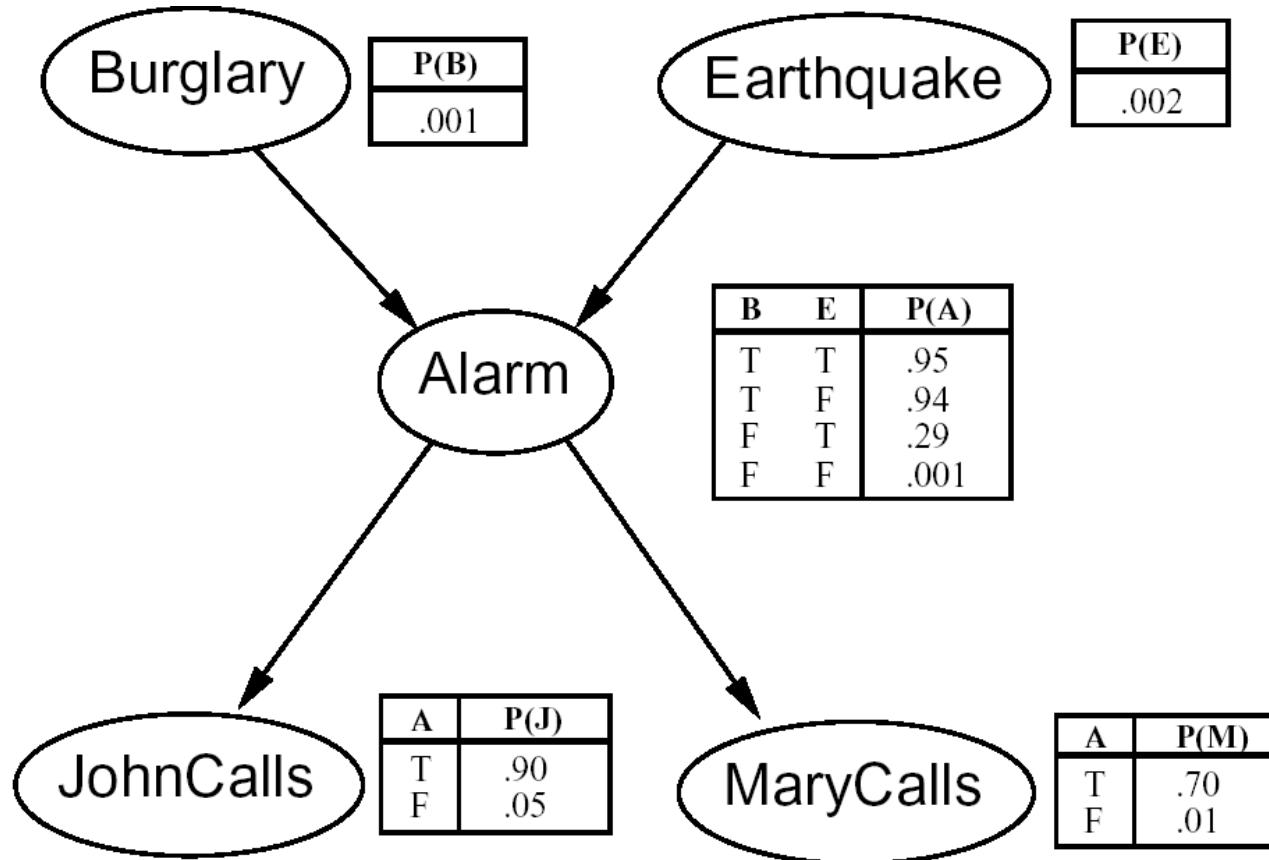
Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

Algorithms

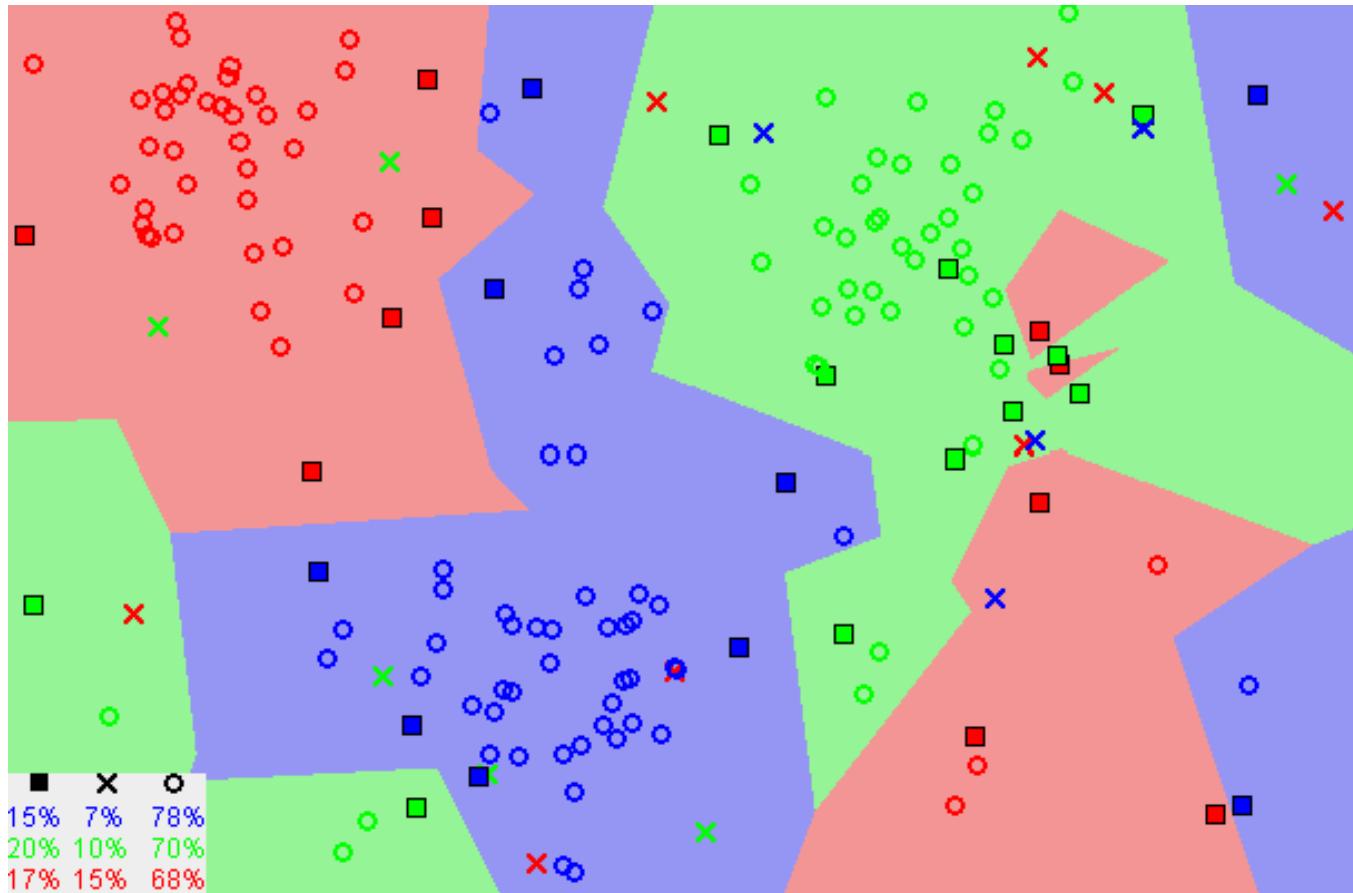
- **Bayes**
 - **KNN and K-means**
 - **Decision tree**
 - **Support Vector Machine**
 - **Boosting and Ensemble Learning**
 - **Linear Statistical Learning (PCA, ICA, NMF)**
 - **Nonlinear Statistical Learning (Manifold learning)**
 - **Deep Neural Networks**
 - **Generative Adversarial Networks**
 - **Bayesian Networks**
 - **Reinforcement Learning**
 - **Federated Learning**
-

Bayes



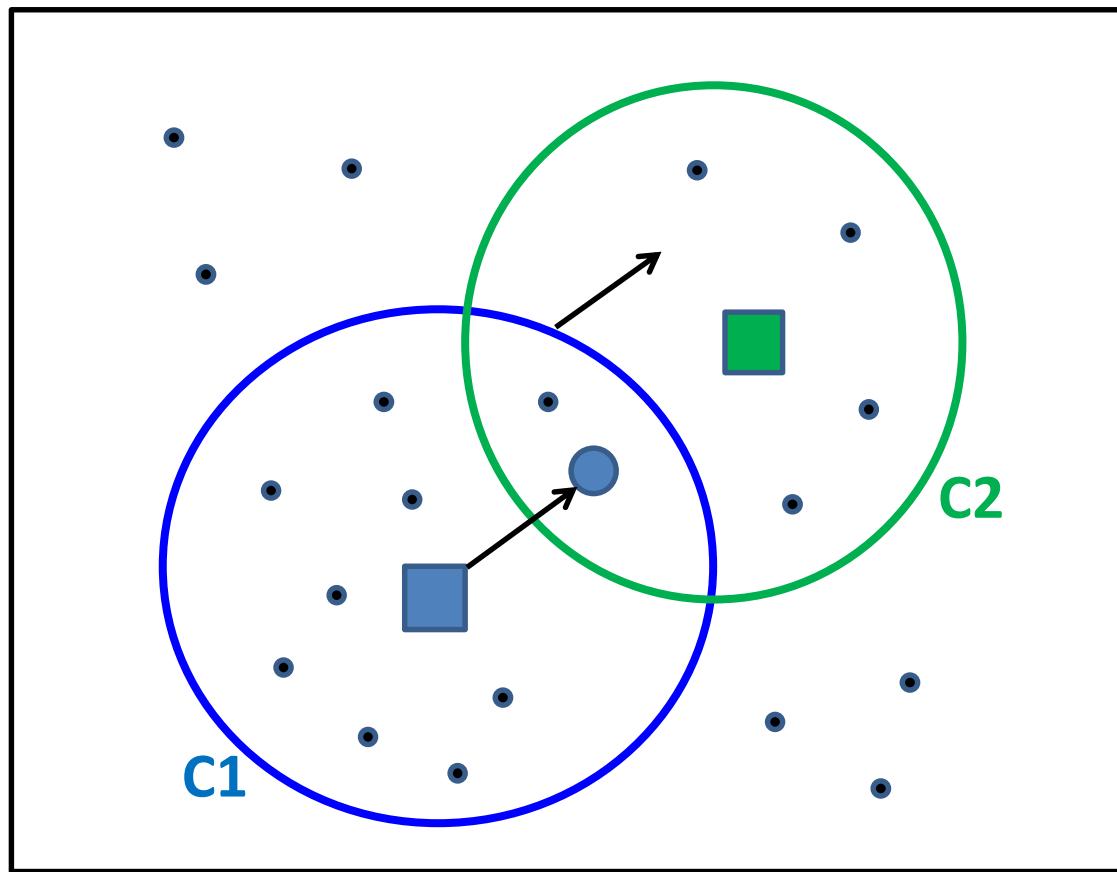
K-Nearest Neighbors

- Use training data for classification



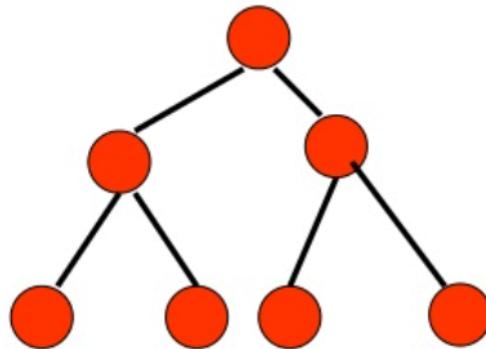
K-Means

- Mean-shift for clustering

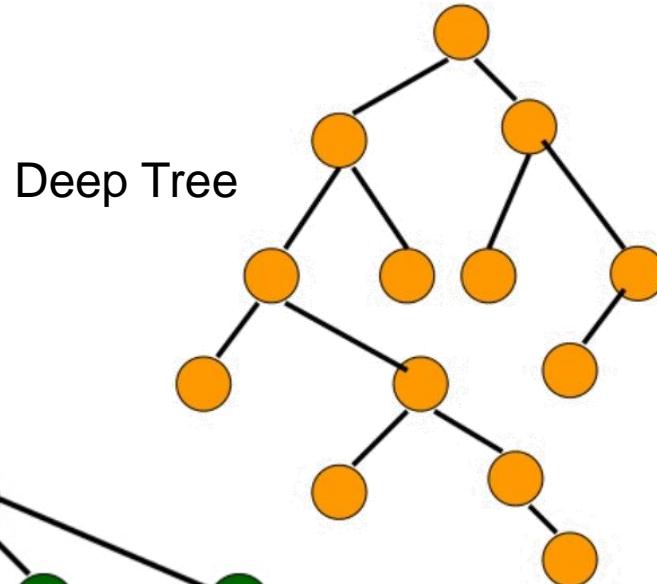


Decision Tree

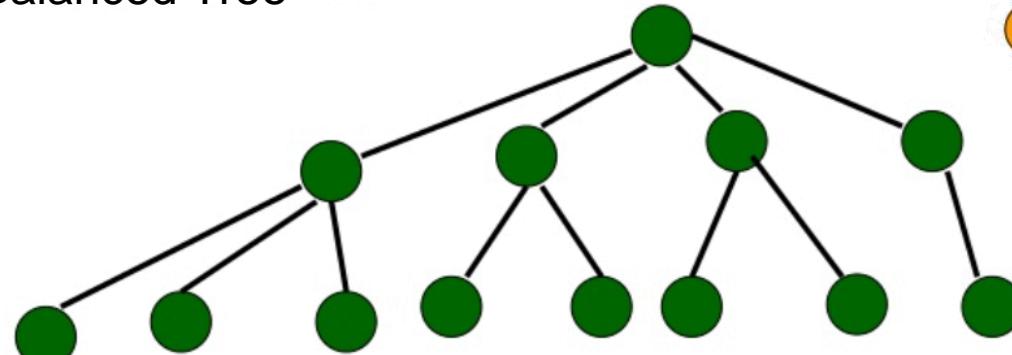
■ Types of Decision Tree:



Balanced Tree



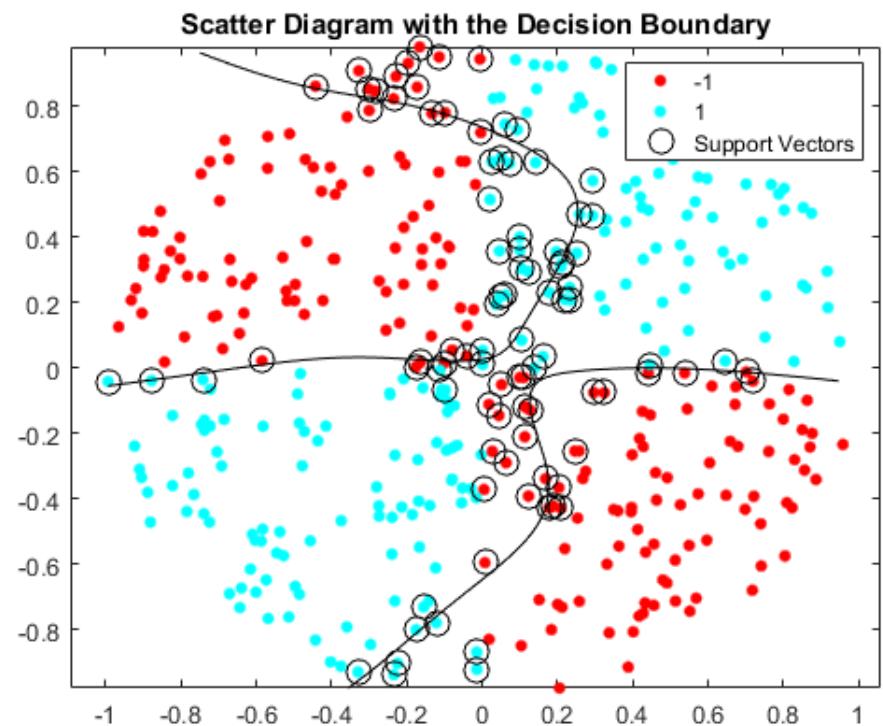
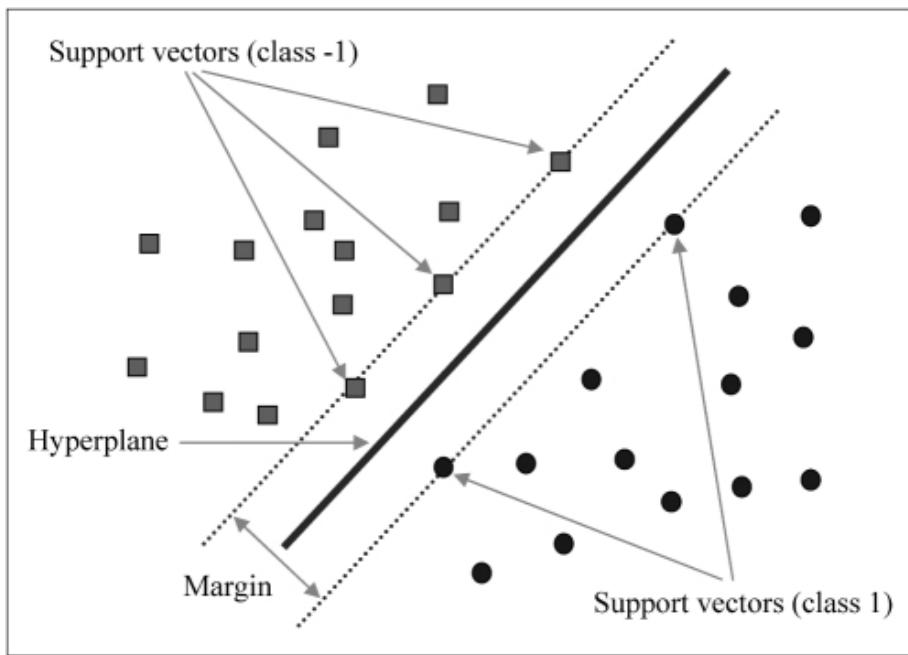
Deep Tree



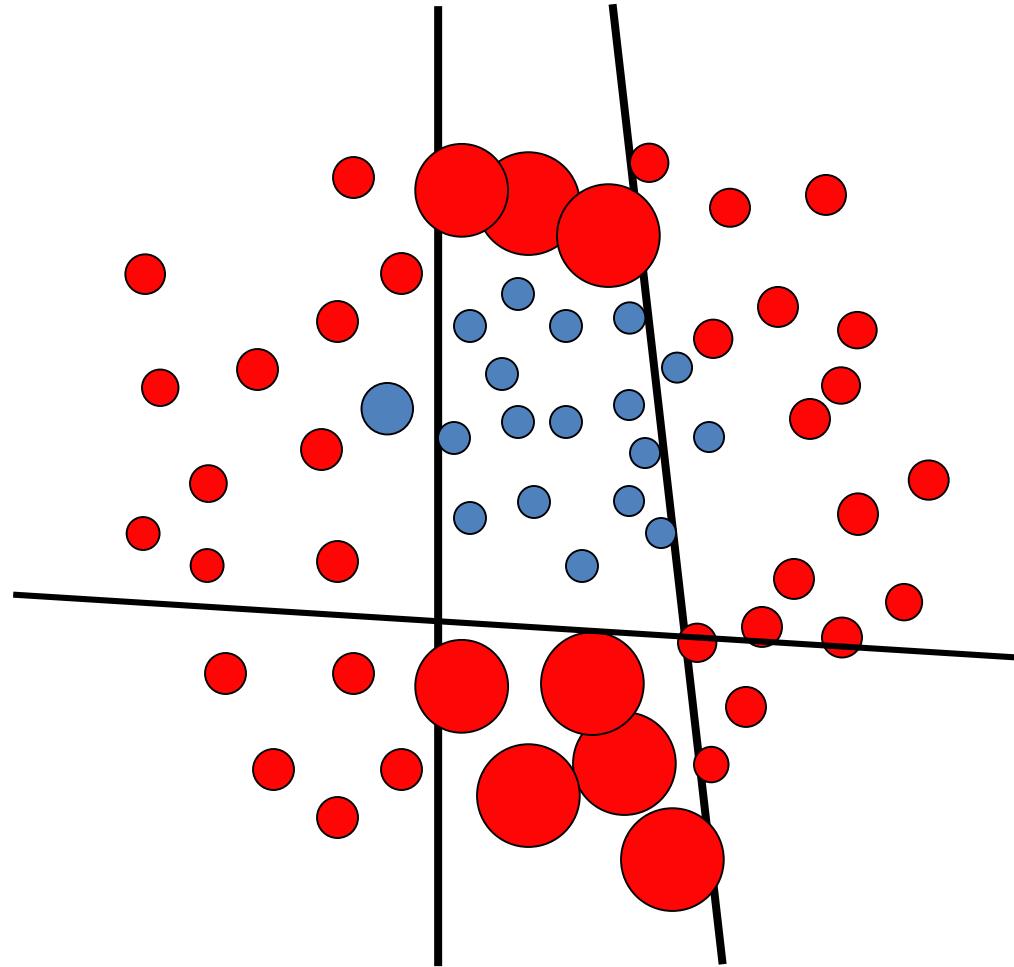
Bushy Tree

SVM

- eg. Linear SVM:
$$\begin{aligned} & \arg \min_w \sum_{i=1}^n \|w\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{s.t. } 1 - y_i x_i^T w \leq \xi_i \\ & \quad \xi_i \geq 0 \end{aligned}$$



Boosting



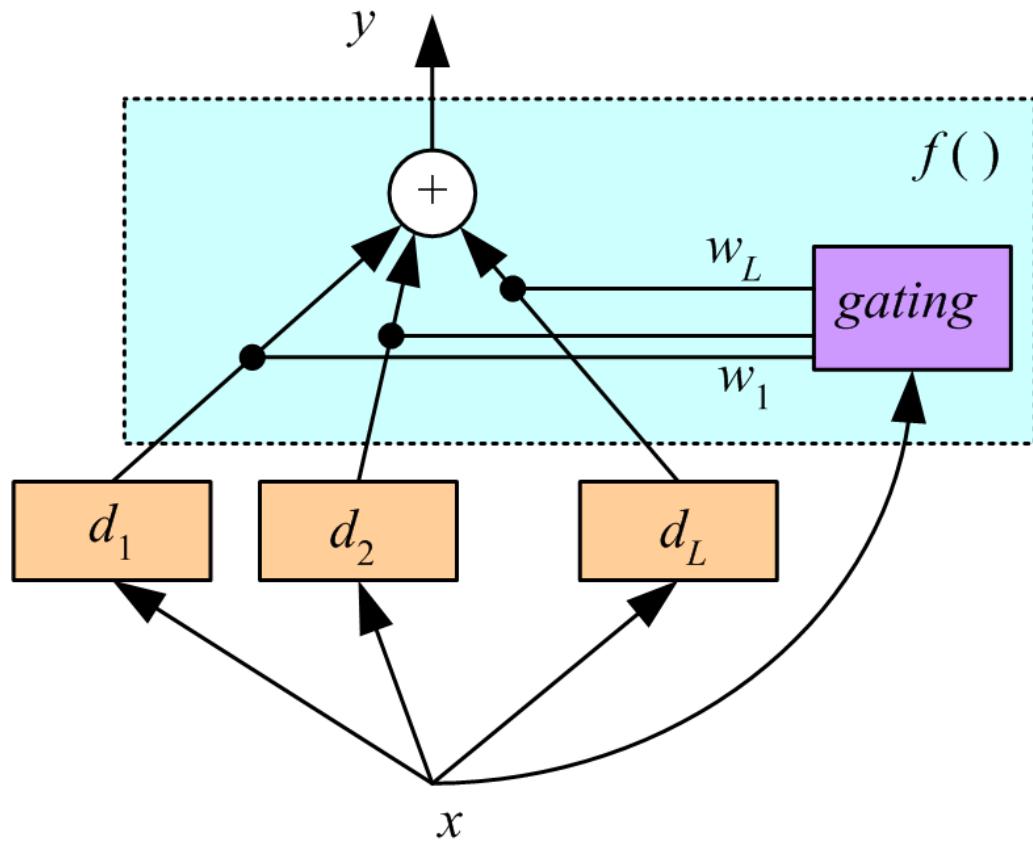
Each data point has
a class label:

$$y_t = \begin{cases} +1 (\text{red circle}) \\ -1 (\text{blue circle}) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

Ensemble Learning



$$y = \sum_{j=1}^L w_j d_j$$

Linear Statistical Learning

■ PCA

$$\begin{array}{c} \text{Data} \quad \text{Basis} \\ \searrow \quad \downarrow \quad \swarrow \\ Y = AX \quad \text{Coefficients} \\ \quad \quad \quad A_i \perp A_j \end{array}$$

■ ICA

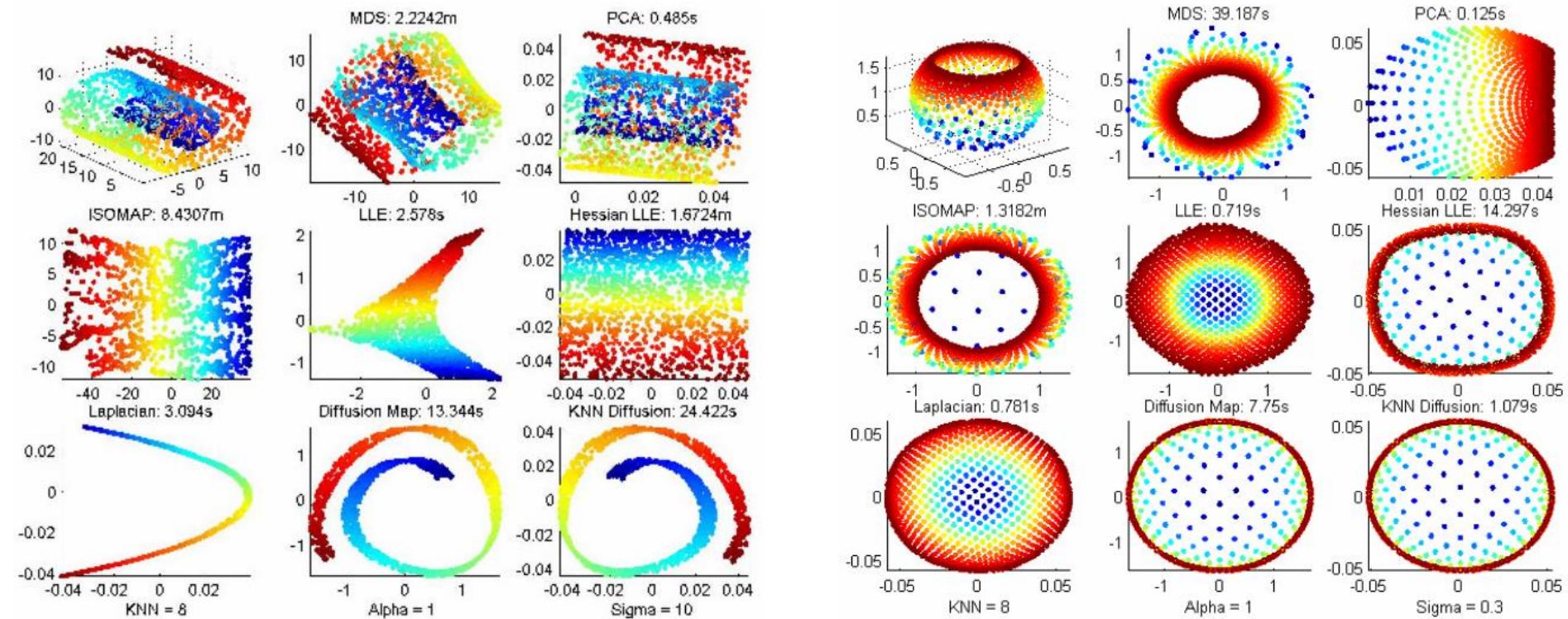
$$\begin{array}{c} \text{Data} \quad \text{Mixture Coefficients} \\ \searrow \quad \downarrow \quad \swarrow \\ Y = AX \quad \text{Components} \\ \min I(X) \end{array}$$

■ NMF

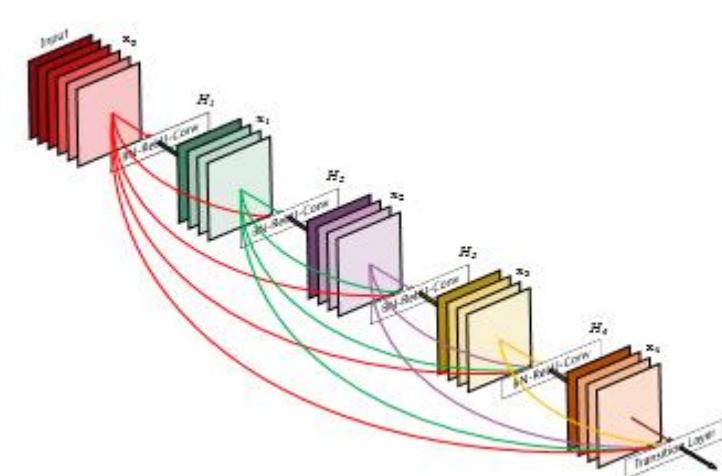
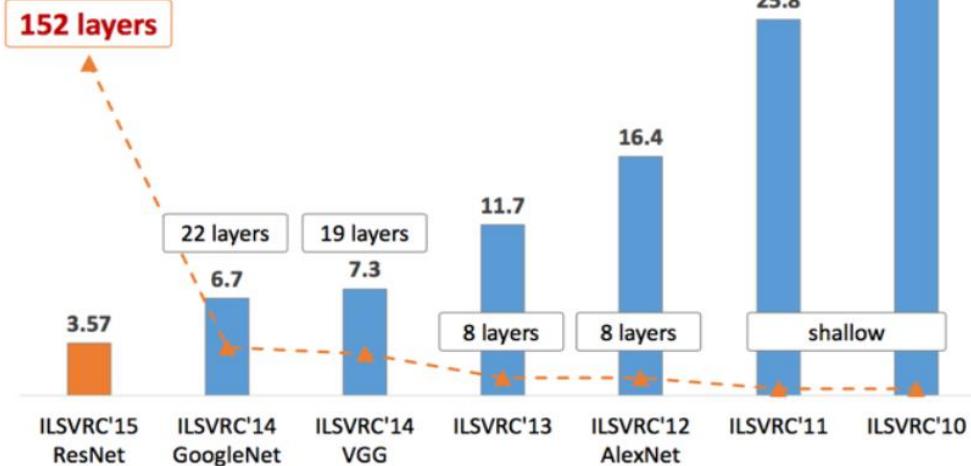
$$\begin{array}{c} \text{Data} \quad \text{Basis} \\ \searrow \quad \downarrow \quad \swarrow \\ Y = AX \quad \text{Coefficients} \\ \quad \quad \quad A, X > 0 \end{array}$$

Nonlinear Statistical Learning

■ Manifold learning



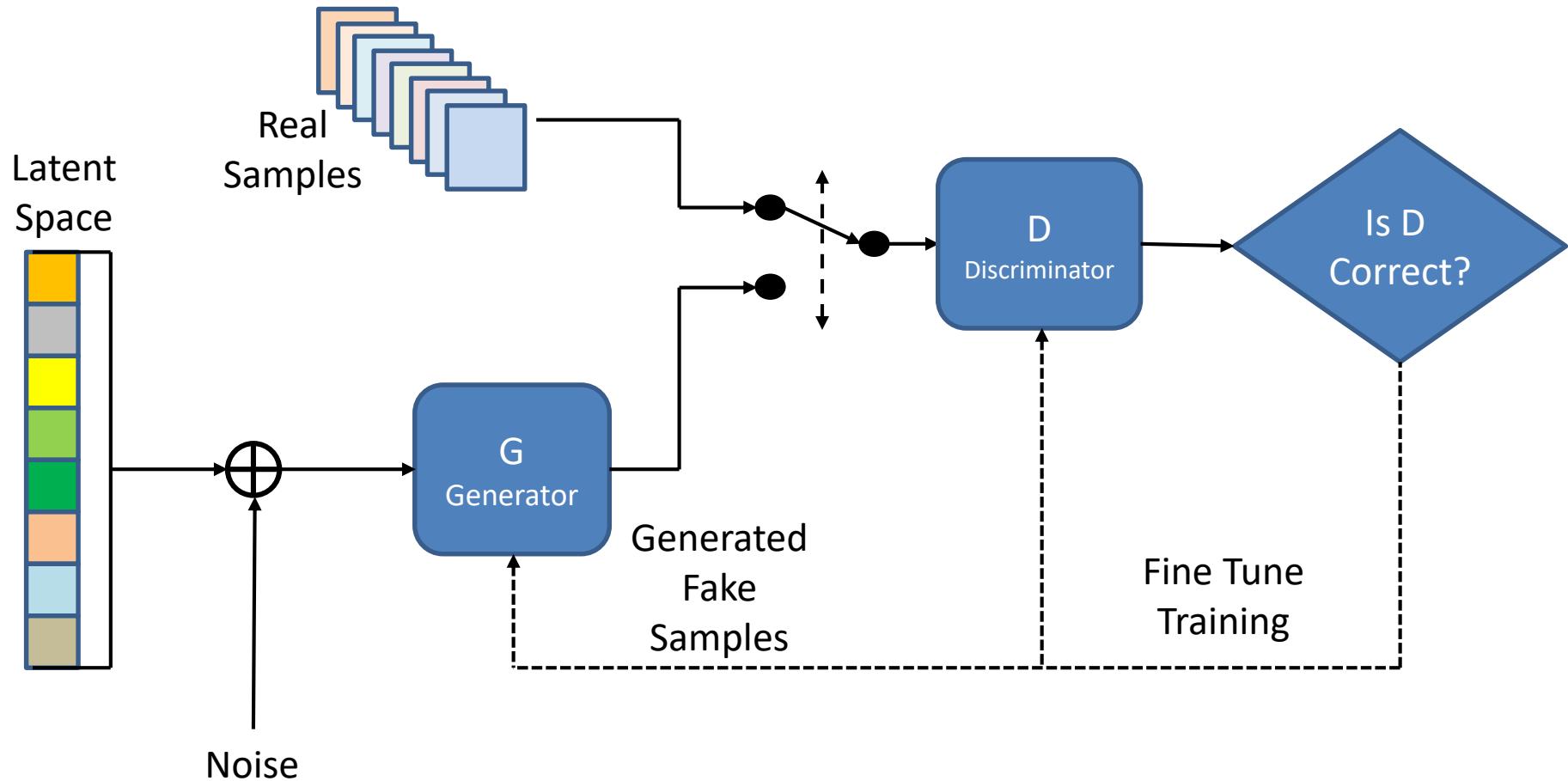
Deep Neural Networks



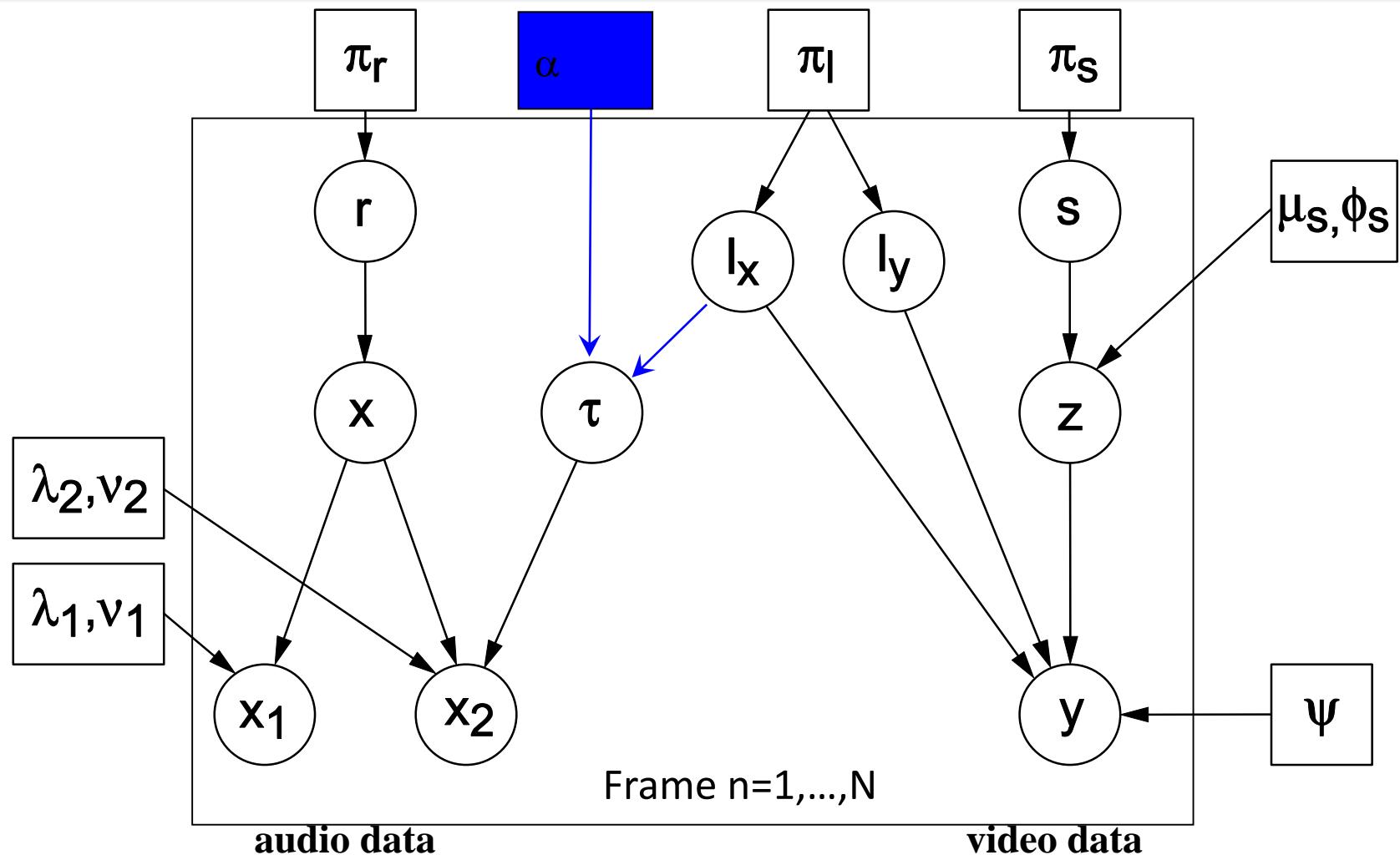
- Task: recognition
- Dataset: ILSVRC

■ Huang G, Liu Z, Weinberger K Q, et al.
Densely connected convolutional networks[J]. arXiv preprint
arXiv:1608.06993, 2016.

Generative Adversarial Networks

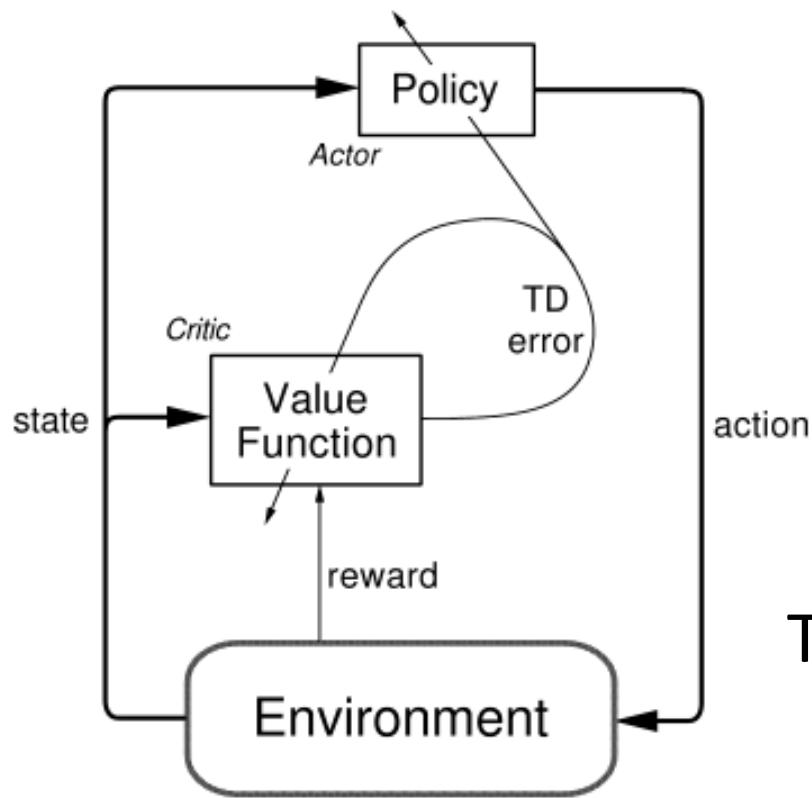


Bayesian Networks



Reinforcement Learning

- State, action, and Reward

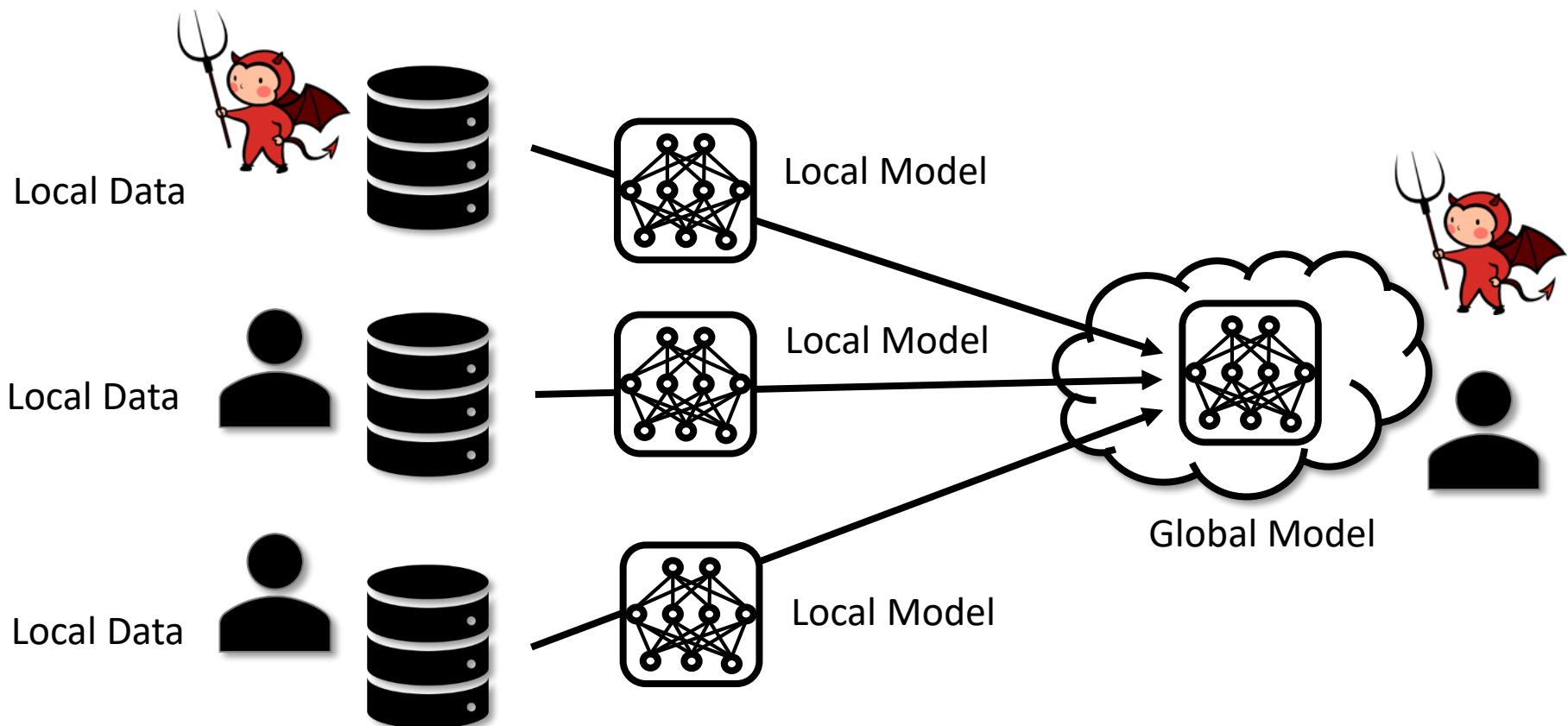


Update: Policy Function
Value Function

TD Error: Temporal Difference
between Real Reward
and Estimated Reward

Federated Learning

- Collaborative Learning



Outlines

- Framework
 - Problem Statement
 - Related Areas
 - History
 - Datasets and Learning Models
 - Optimization Methods
 - Algorithms
 - Examples
-

More Reading and Multimedia Materials

Books: 《人类简史》 《奇点将近》 《终极算法》
《人工智能时代》 《2050》 《情感机器》
《数学之美》

Movies: “Blade Runner” “AI” “Prometheus”
“Covenant” “Ex Machina” “She”
“2001: Space Odyssey” “The Matrix”
“I, Robot” “Bicentennial Man”
“Terminator”

TV Series: “West World” “Humans” “Black Mirrors”

More Course Links

Stanford Machine Learning:

<https://see.stanford.edu/Course/CS229/47>

MIT Machine Learning: <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/index.htm>

Stanford CNN for Vision: <http://cs231n.stanford.edu>

Stanford Deep Learning: <http://cs230.stanford.edu/syllabus.html>

MIT Deep Learning: <http://introtodeeplearning.com/>
