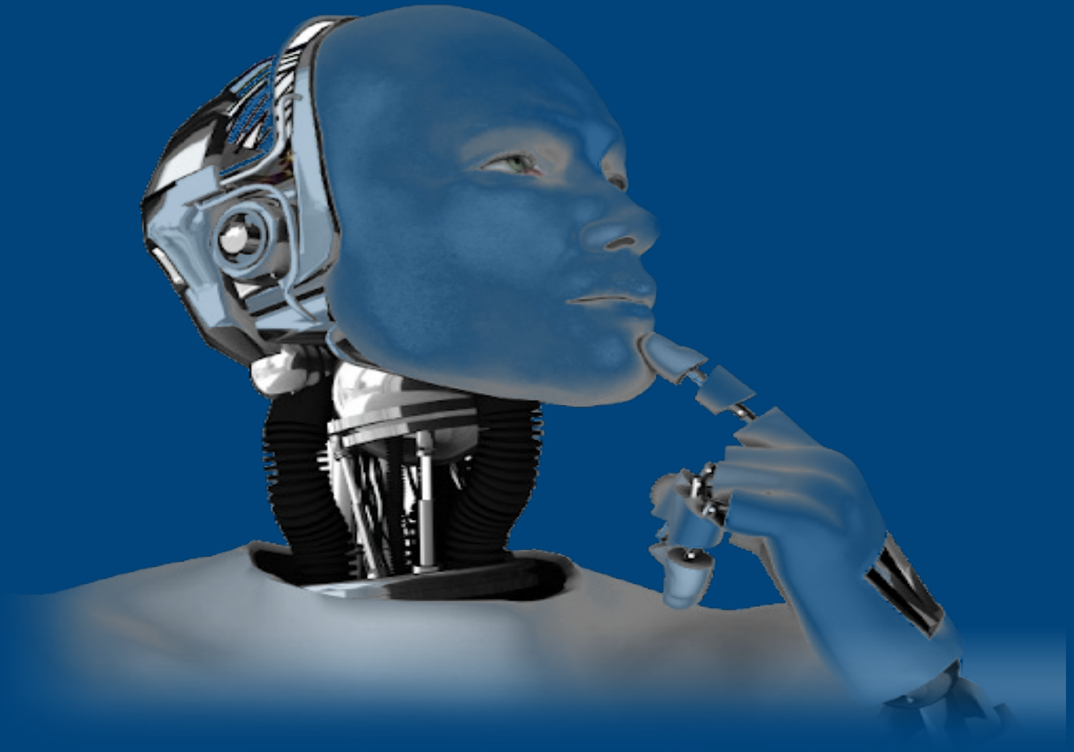


可信机器学习

Trustworthy Machine Learning

Lecturer: Dr. Xingjun Ma
Tutor: Dr. Zichan Ruan

School of Computer Science, Fudan University

Autumn, 2022

Course Info

Dr Xingjun Ma
Instructor
xingjunma@fudan.edu.cn

Dr Zichan Ruan
Tutor
zichanr@fudan.edu.cn

Time&Palce: Class 11-13, 18:30pm – 21:05pm
Wednesday, Weekly (Except National Holiday)
江湾校区， JA203

Course page: https://trustworthymachinelearning.github.io/

Office: D5025, X2, 交叉2号楼D5025

Office Hours: Tuesday Afternoon

Personal page: https://xingjunma.com

Syllabus

- Week 1: Intro, Basics of Machine Learning
- Week 2: Explainablity and Robustness to Common Corruptions
- Week 3: Adversarial Examples, Attacks and Explainations
- Week 4: Adversarial Defense (Part I), Adversarial Example Detection
- Week 5: Adversarial Defense (Part II), Early Defense Methods, Adversarial Training
- Week 6: Adversarial Defense (Part III), Certifiable Adversarial Defense
- Week 7: Data Poisoning Attack and Defense
- Week 8: Backdoor Attack and Defense
- Week 9: Data Leakage and Model Stealing
- Week 10: Differential Privacy
- Week 11: Federated Learning
- Week 12: Machine Learning Fairness
- Week 13: Data Manipulation and Deepfakes
- Week 14: Model Intellectual Property Protection
- Wee 15: Guest Lectures on Research Frontiers
- Week 16: Project Report
- Week 17: Project Report

Assessment

考核形式* Assessment Criteria	权重 Percentage	评定标准 Assessment Standard
出勤 Attendance	10%	全勤10分，缺席1次扣1分
课堂表现 Participation	0%	
作业/实验/实践 Assignment(s)	20%	基于Kaggle的课堂对抗攻防赛（20%）
课程论文 Course Paper	60%	学生自选研究题目，解决一个可信机器学习问题，设计自己的方法并与基线方法比较。 40分以上：选题新颖，方法创新，具备学术价值和现实意义、写作规范，行文流畅。 30分以上：选题合理，观点明确，思路清晰，方法具有一定的创新。 30分以下：背景知识缺乏了解，选题、方法设计、分析不能达到基本要求。
开卷考试 Open-book exam	0%	
闭卷考试 Close-book exam	0%	
其他 Other(s)	10%	开源社区贡献（10%），包括但不限于收集各研究方向的论文、设计开源示例、整合并复现各研究方向的基线方法、建设开源社区等。

Assessment

◆ 基于Kaggle的课堂对抗攻防赛 （**占比20%**）

- 计划第5-6周发布，可能会提前
- 请同学们自行寻找计算资源（GPU）
- 比赛内容：
  - ✓ 对抗攻击一个鲁棒训练的模型
  - ✓ 数据集为CIFAR-10
  - ✓ 衡量攻击成功率和效率，各占50%

■ 得分：按排名进行评分，**第一名100分**，**最后一名50分**

◆ 自选研究题目 （**占比60%**）

- 有4-5个备选题目，第10周左右发布
- **需要组队：博士1-2人、硕士2-3人**
- **需要做实验**
- **需要写报告**（英文报告加分）
- **需要课堂作展示**，**每个组5分钟**

■ 得分：结合创新性、报告质量、展示质量三个方面综合评分

没卡的同学建议使用Google Colab：https://colab.research.google.com/

Textbook

- 自编教材《人工智能数据与模型安全》
  - 由Fudan Vision and Learning Lab编写
  - 未经允许不能分享给课外人员
  - 教材还在优化中，部分章节缺失
  - 同学可参与到教材的优化中来（算开源贡献）：发现错误、改正错误，至少需要完成一个二级章节（2、4、5、6、7）中的三级章节（例如：5.3），章节由老师来制定
  - 教材优化的同学不多于10人

下载链接: https://pan.baidu.com/s/1kybxud_tz0xshWpMEORAhg?pwd=tauu 提取码: tauu

Week 1: Machine Learning Basics

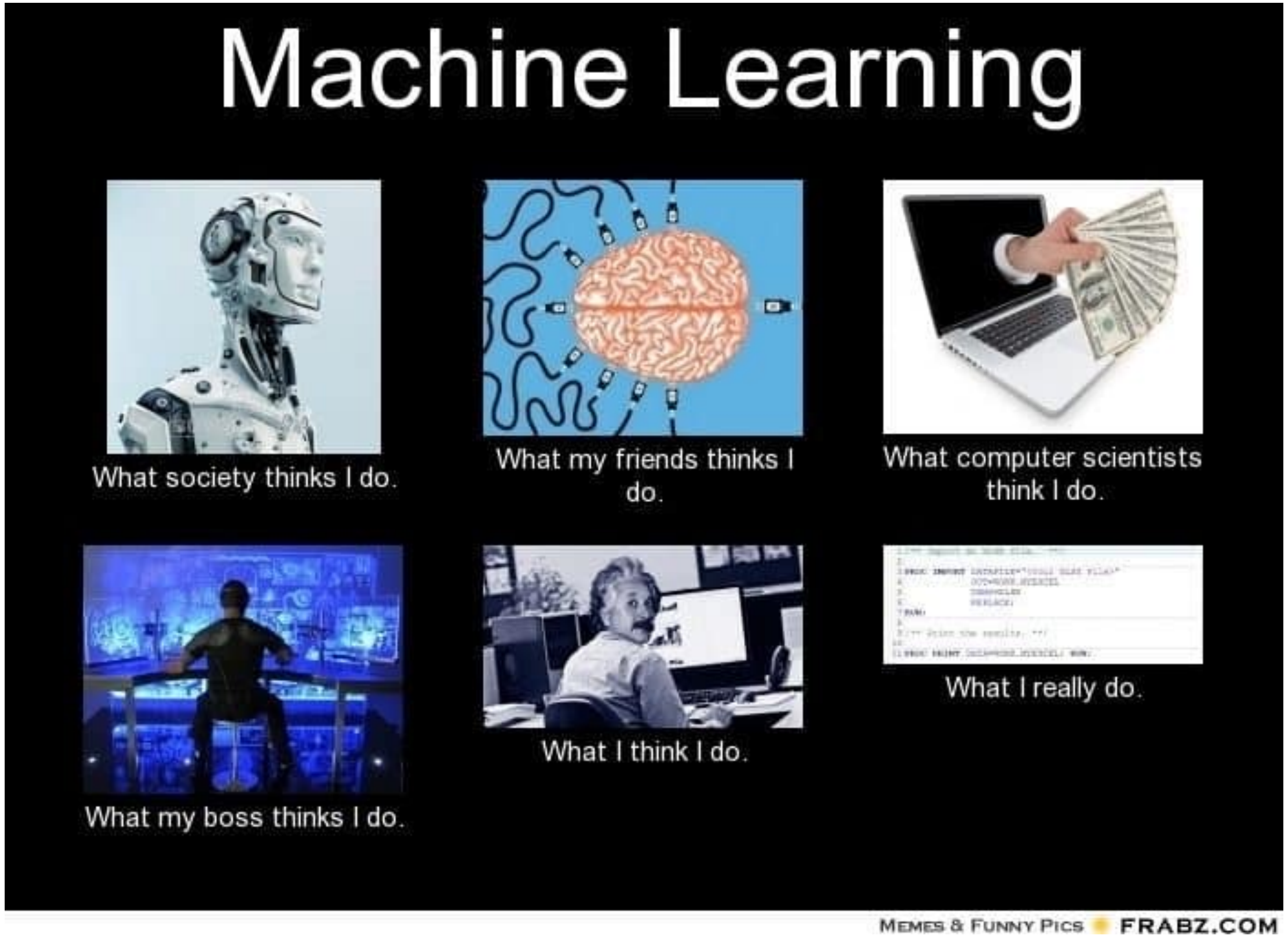
1. What is Machine Learning

2. Machine Learning Paradigms

3. Loss Functions

4. Optimization Methods

What Is Machine Learning



https://carlolepelaars.nl/2018/10/15/100daysofmlcode-summary/

What Is Machine Learning

‘Cat’

‘Dog’

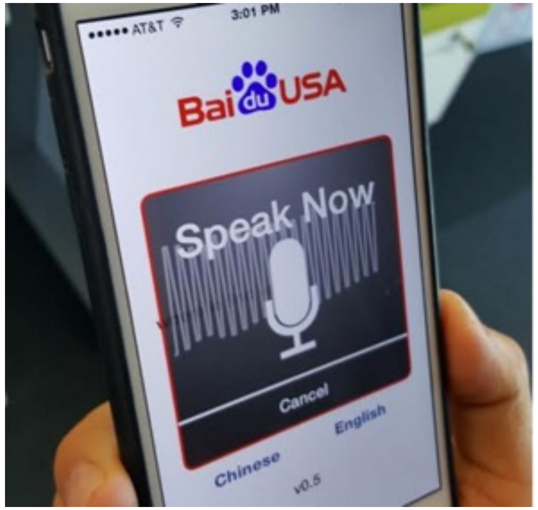
https://www.image-net.org/

What Is Machine Learning

Million-scale Image Recognition

https://www.image-net.org/

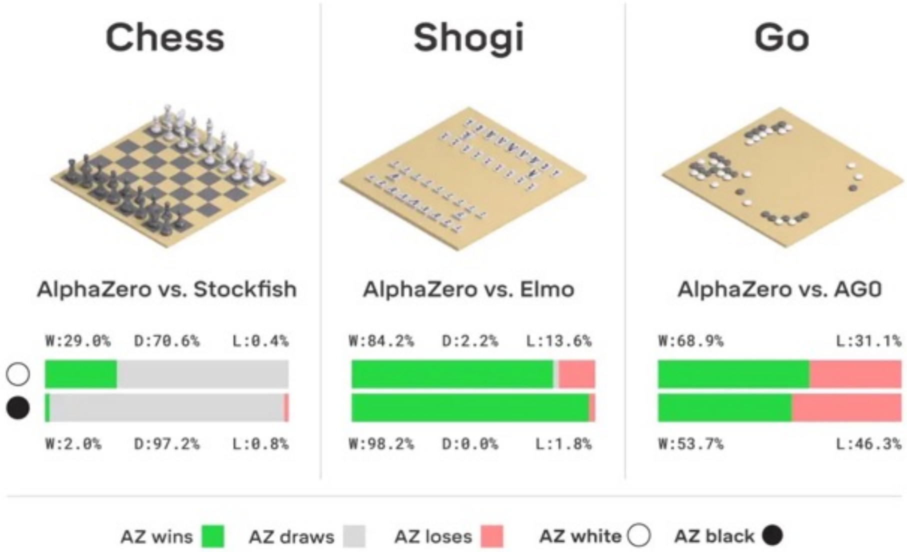
What Is Machine Learning



Speech Recognition

https://machinelearning.apple.com/research/hey-siri;

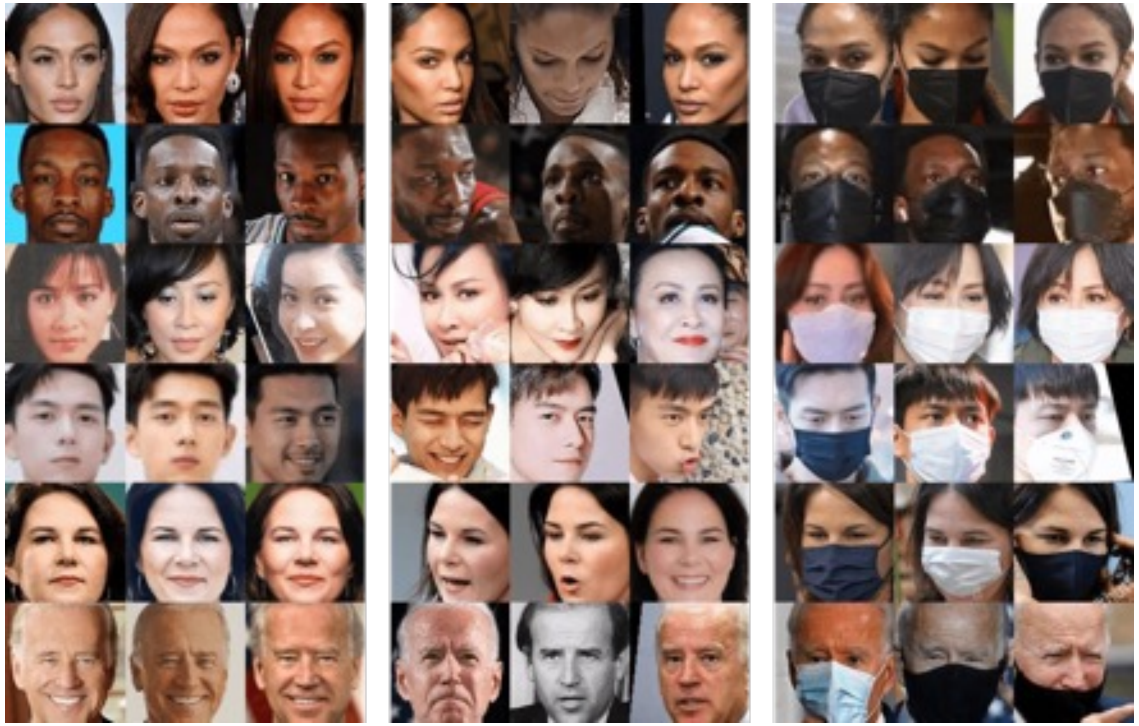
What Is Machine Learning



Strategy Games

https://www.deepmind.com/research/highlighted-research/alphago;
https://www.deepmind.com/blog/alphazero-shedding-new-light-on-chess-shogi-and-go

What Is Machine Learning



Million-scale Facial Recognition

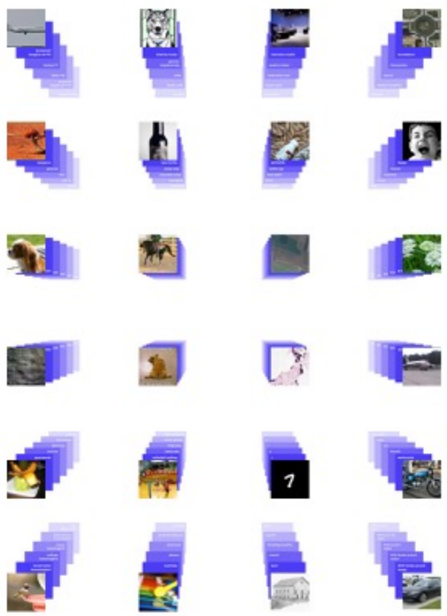
https://www.face-benchmark.org/

What Is Machine Learning

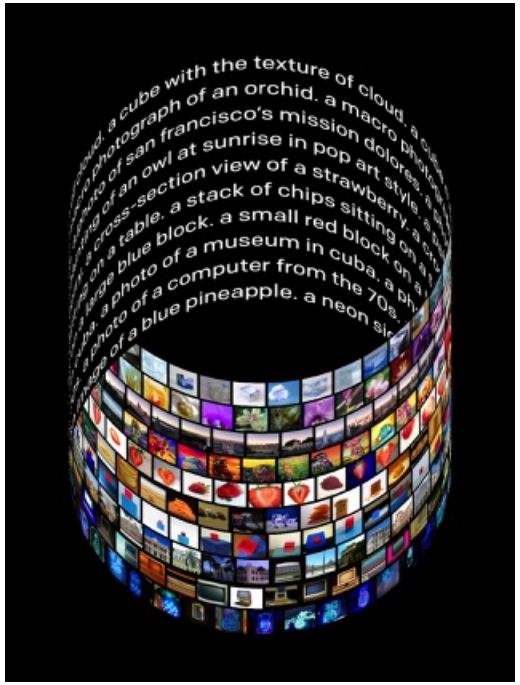
Large-scale Visual-Speech Learning

https://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrs3.html

What Is Machine Learning



CLIP: Connecting Text and Images



DALL·E: Creating Images from Text

https://openai.com/research/

What Is Machine Learning

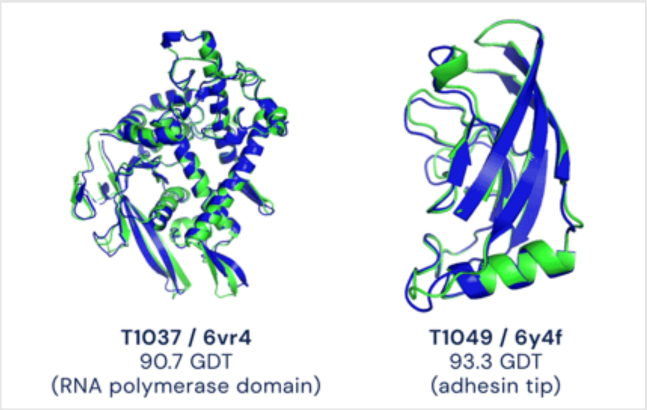
https://www.deepmind.com/blog/alphafold-reveals-the-structure-of-the-protein-universe

Machine Learning Is Everywhere

智慧教育

智慧医疗

自动驾驶

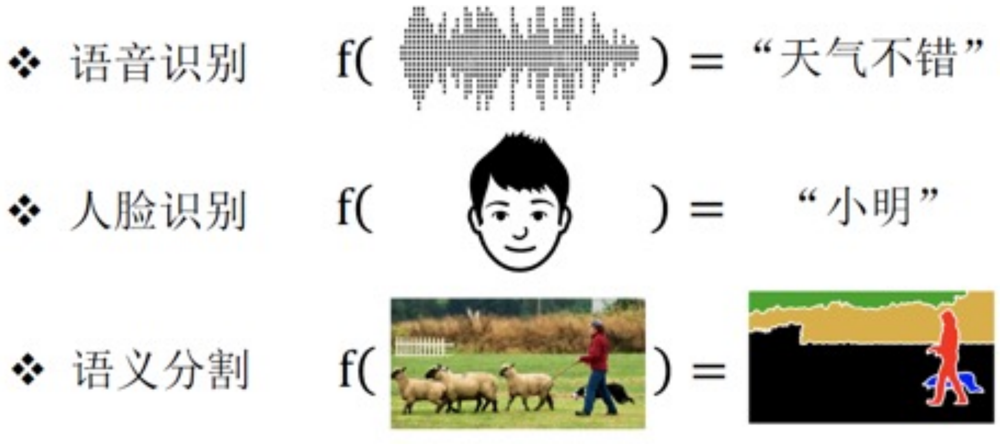


生物信息

智能制造

智慧金融

Elements of Machine Learning



Learning Patterns From A Given Dataset Using An Algorithm

Data describes the problem

Model describes the brain of the machine

Algorithm describes the learning mechanism

Hardware accelerates the learning

机器学习四要素：数据、模型、算法、算力

Elements of Machine Learning

Data ⟷ Knowledge

Model ⟷ Brain

Algorithm ⟷ Mind

Hardware ⟷ Body

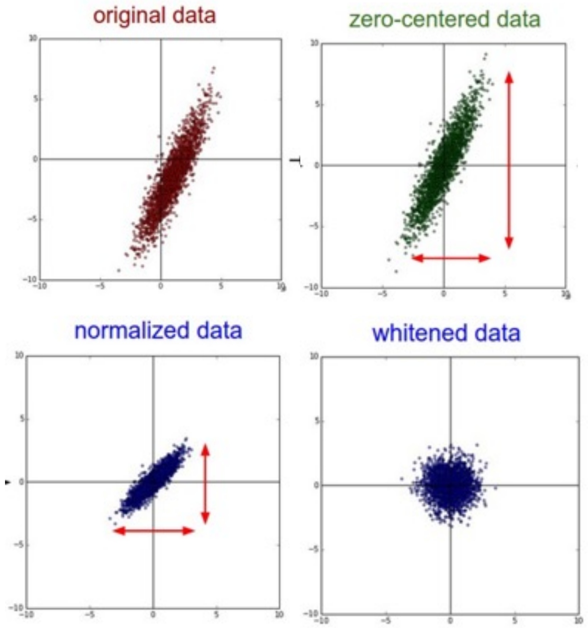
10 Questions of Machine Learning

1. What is the task?
2. What is the objective?
3. What is the data?
4. How much data do we have?
5. What is the model?
6. What are the inputs and outputs?
7. What needs to be learned?
8. How is the model trained?
9. How is the model tested?
10. How is the model deployed?

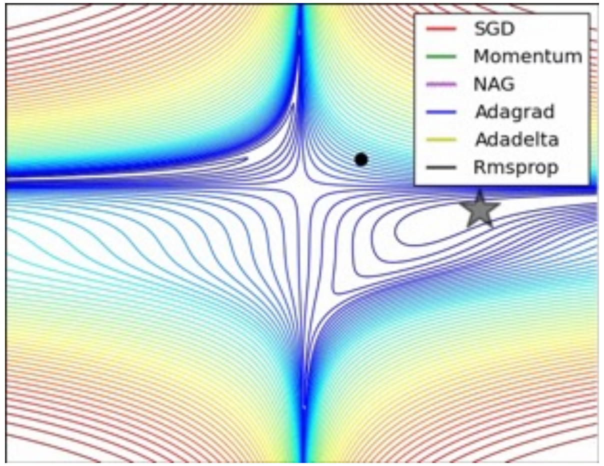
1. Problem definition
2. Learning objective
3. Training/Test data
4. Scale of learning
5. Model Architecture
6. Function Family
7. Features/Representations
8. Training Method
9. Evaluation Metrics
10. Generalization

Machine Learning Pipeline

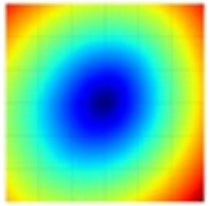
setup the input



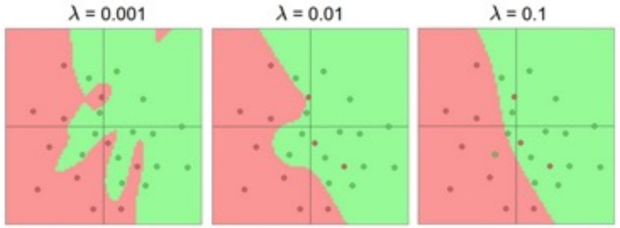
setup the optimiser



setup the loss



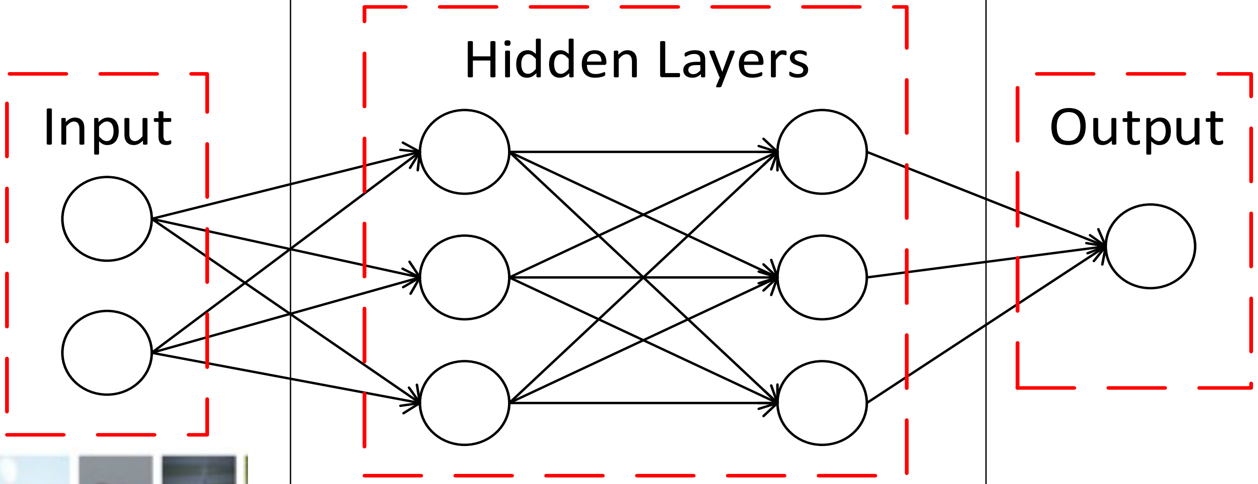
landscape of a loss function, it varies w.r.t. data, the function itself



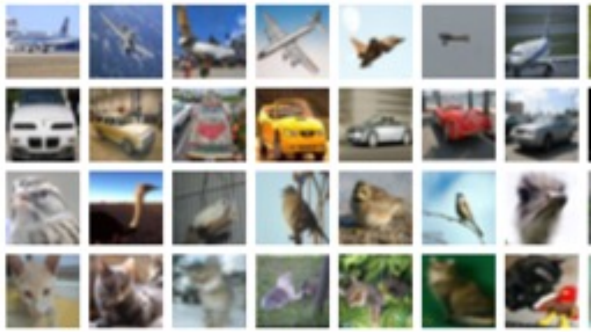
regularization makes decision region smoother

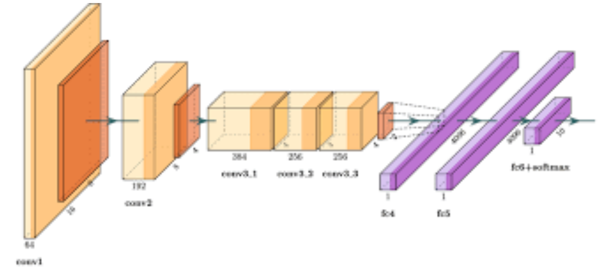
Machine Learning Pipeline

setup the input | setup the optimiser | setup the loss



$$\min_{\theta} \mathbb{E}_D[L(f_\theta(x_i), y_i)]$$





Cross entropy:

$$L(f_\theta(x_i), y_i) = \sum_{j=1}^{C} y_{ij} \log f_\theta^j(x_i)$$

Machine Learning Concepts

Data

Training data
Test data
Samples
IID/Non-IID
Domain
Feature
Representation
Noise
Corruptions
…

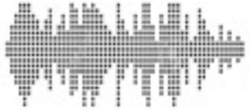
Model

SVM/RF/LR
DNN
RNN
CNN
FWN
Layers, neurons, blocks, module
Activations, logits, probabilities
Model capacity, parameters
…

Algorithm

Learning method
Standard learning
Curriculum learning
Supervised learning
Unsupervised learning
Reinforcement learning
Continual learning
Self-supervised learning
Representation learning
Contrastive learning
…

Learning Is Optimizing

- 语音识别 f() = “天气不错”
- 人脸识别 f() = “小明”
- 语义分割 f() = 

Learning is the process of empirical risk minimization (ERM)

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f_{\theta}(\boldsymbol{x}_i), y_i)$$

Mapping function: $Y = f(X)$

Hypothesis space: $\mathcal{F} = \{f | Y = f_{\theta}(X), \theta \in R^m\}$

Expected risk: $R_{exp}(f) = \mathbb{E}_P[\mathcal{L}(Y, f(X))] = \int_{X \times Y} \mathcal{L}(f(\boldsymbol{x}), y) P(\boldsymbol{x}, y) d\boldsymbol{x} dy$

Empirical risk: $R_{emp}(f) = \mathbb{E}_{(\boldsymbol{x},y) \in D} \mathcal{L}(f(\boldsymbol{x}), y) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i), y_i)$

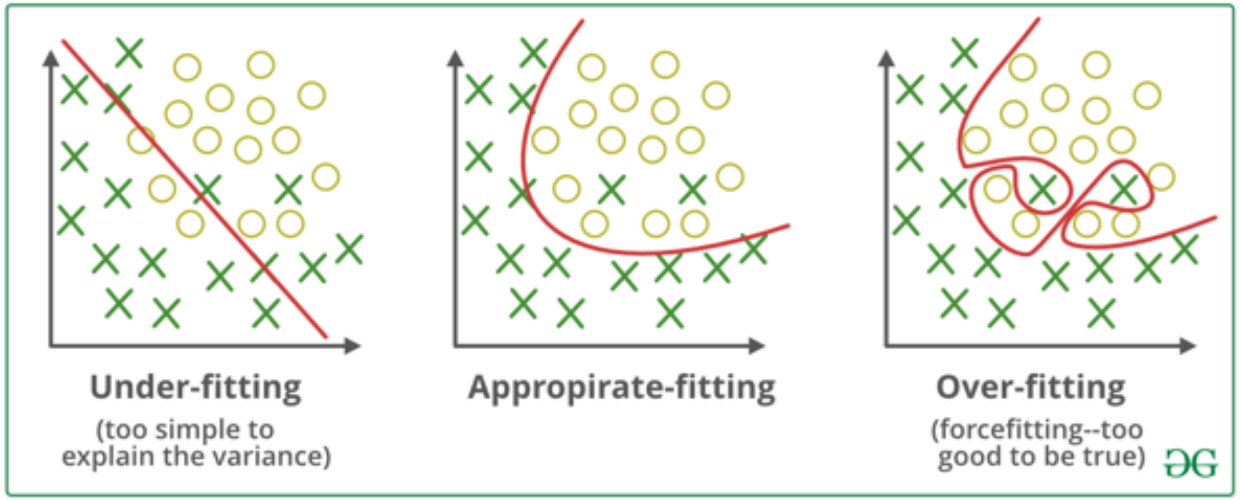
```
Input -> X
Output -> Y

f(X) => mapping function
Y = f(X)
```

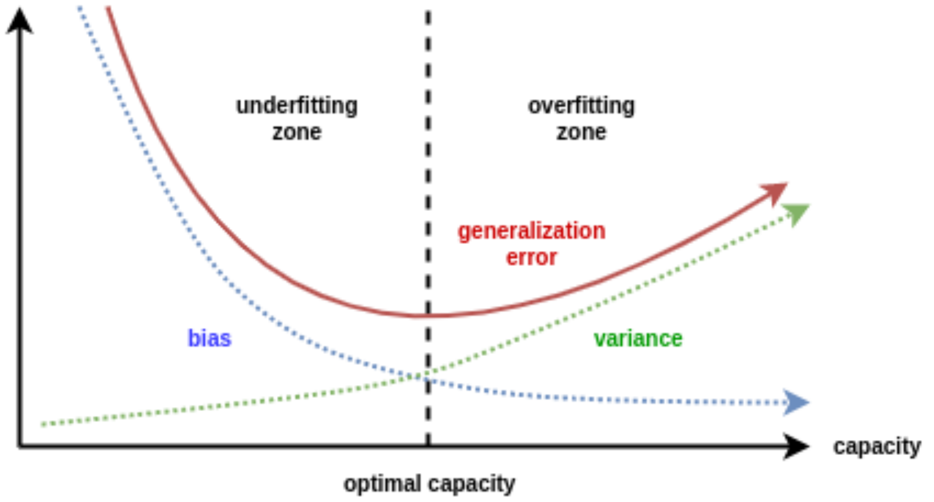
Fitting, Overfitting, Underfitting

Bias: assumptions made by a model to make learning easier — Training Error

Variance: difference between training and test error — Test Error –Training Error — Generalization gap

Generalization error = expected loss = test error = Bias + Variance





https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

Regularization

One solution to the **Overfitting** problem

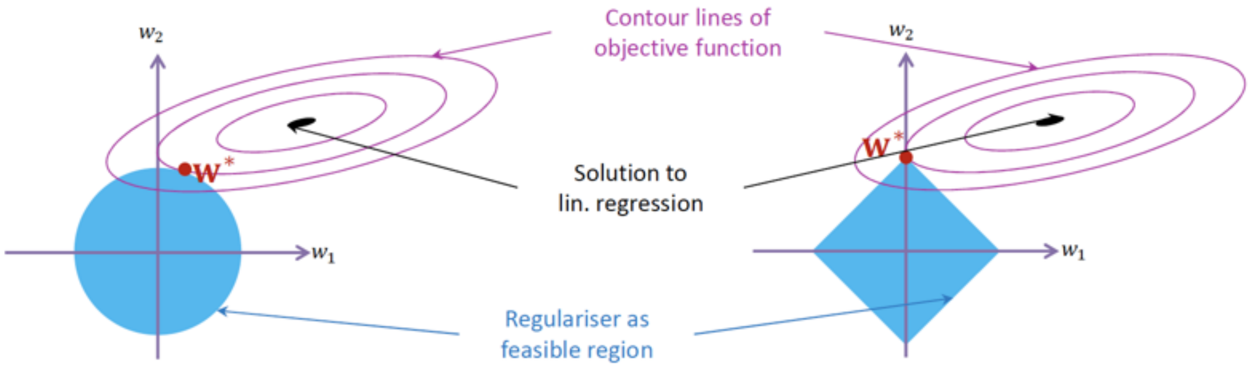
结构风险最小化
Structural Risk Minimization

$$R_{srm}(f) = R_{emp} + \lambda \cdot \Omega(\theta)$$

$$L_1 : \Omega(\theta) = ||\theta||_1 = \sum_i |\theta_i|$$

$$L_2 : \Omega(\theta) = ||\theta||_2 = \sum_i \theta_i^2$$

$$\min_{\mathbf{w}} \sum_i (y_i - \mathbf{X}_{i\cdot}\mathbf{w})^2 \text{ s.t. } \|\mathbf{w}\|_2 \le \lambda$$

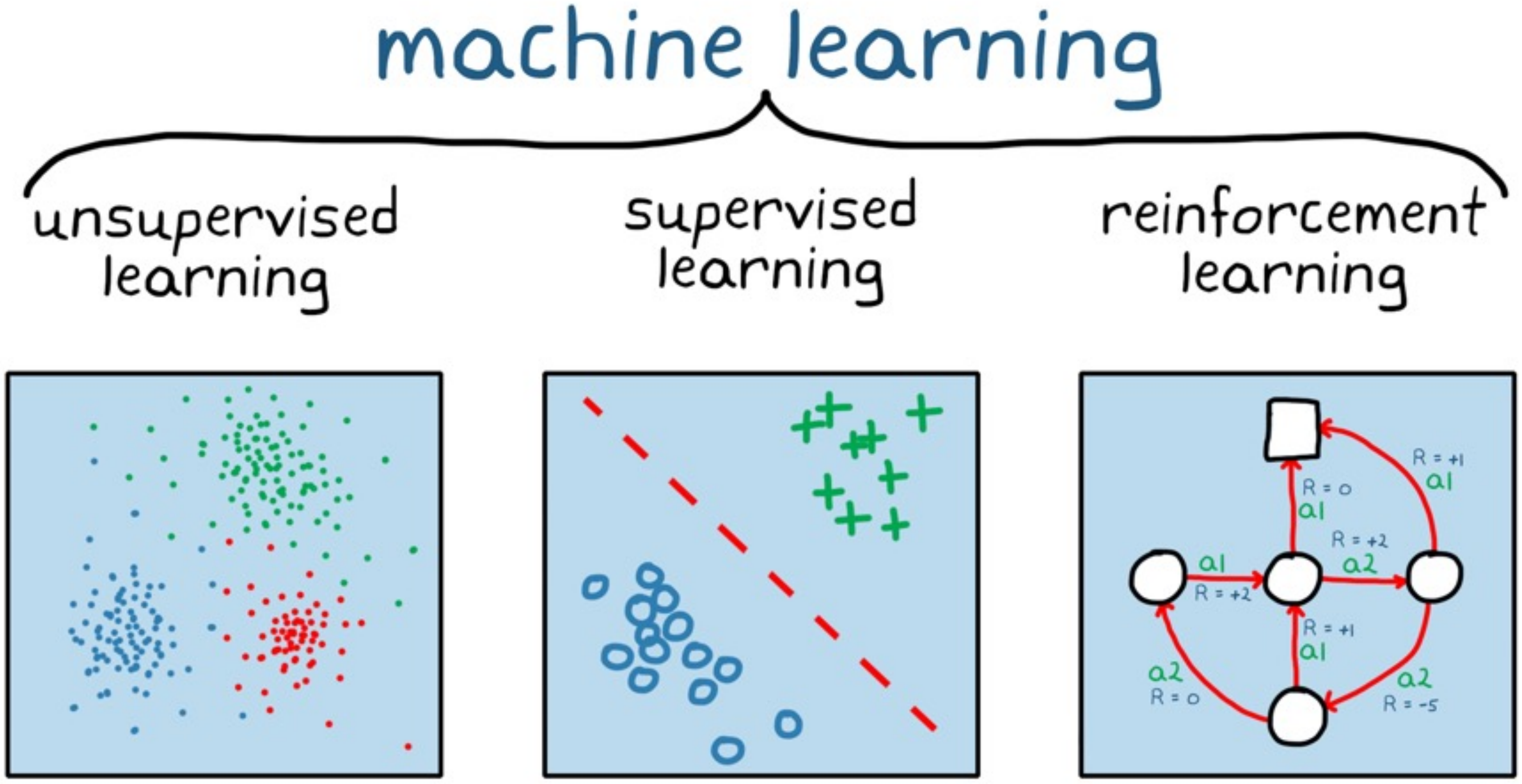


RIDGE REGRESSION ($\|\mathbf{w}\|_2$)

LASSO ($\|\mathbf{w}\|_1$)

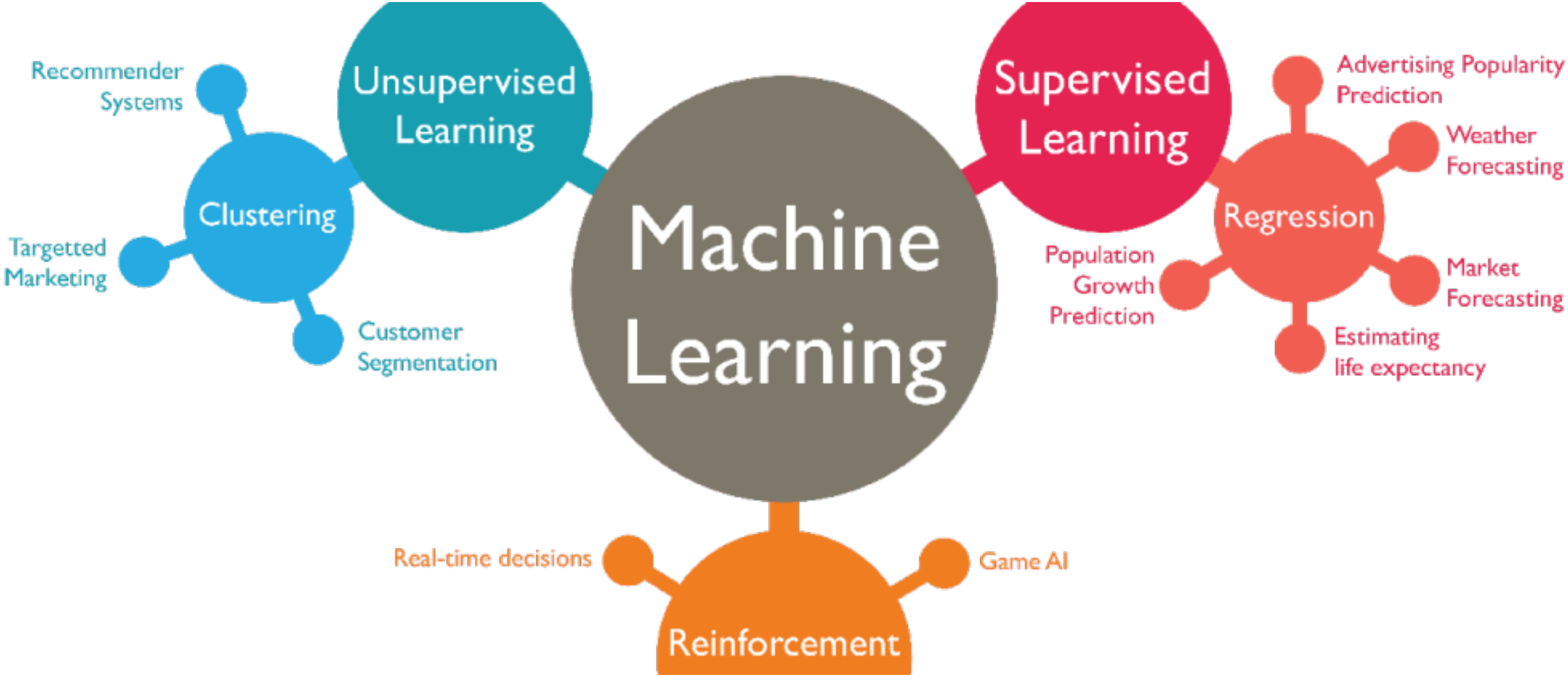
L_1-regularisation encourages solutions $\mathbf{w}^*$ to sit on axes
→ $\mathbf{w}^*$ will have components equal zero → $\mathbf{w}^*$ will be sparse!

Learning Paradigms



https://ww2.mathworks.cn/discovery/reinforcement-learning.html

Learning Paradigms



https://dev.to/afozbek/supervised-learning-vs-unsupervised-learning-4b65

Supervised Learning

'dog'

'cat'

$$\min_{\theta} \mathbb{E}_{(x,y)\in D} \mathcal{L}(f(\boldsymbol{x}), y) \qquad D = \{\boldsymbol{x}_i, \boxed{y_i}\}_{i=1}^{n}$$

Unsupervised Learning

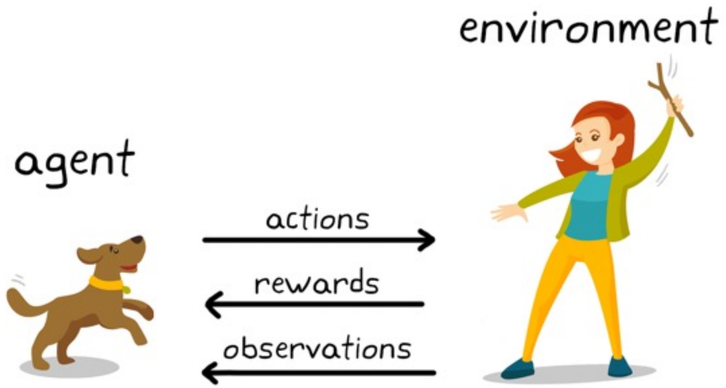
Step 1: $A(X) \rightarrow f$

Step 2: $f(x \in X^*) \rightarrow t$

$$\boxed{D = \{\boldsymbol{x}_i\}_{i=1}^{n}}$$

https://towardsdatascience.com/unsupervised-learning-algorithms-cheat-sheet-d391a39de44a

Reinforcement Learning



History: $\mathrm{H_t} = A_1, O_1, R_1, \ldots, A_t, O_t, R_t$

State: $S_\mathrm{t} = f(H_t)$ $\quad S_t^e \quad S_t^a \quad S_t$

Markov State: $\mathbb{P}[S_{t+1} | S_t] = \mathbb{P}[S_{t+1} | S_1, \ldots, S_t]$

Policy: Deterministic： $a = \pi(s)$ Stochastic： $\pi(a|s) = \mathbb{P}[\mathrm{A}_t = a | S_t = s]$

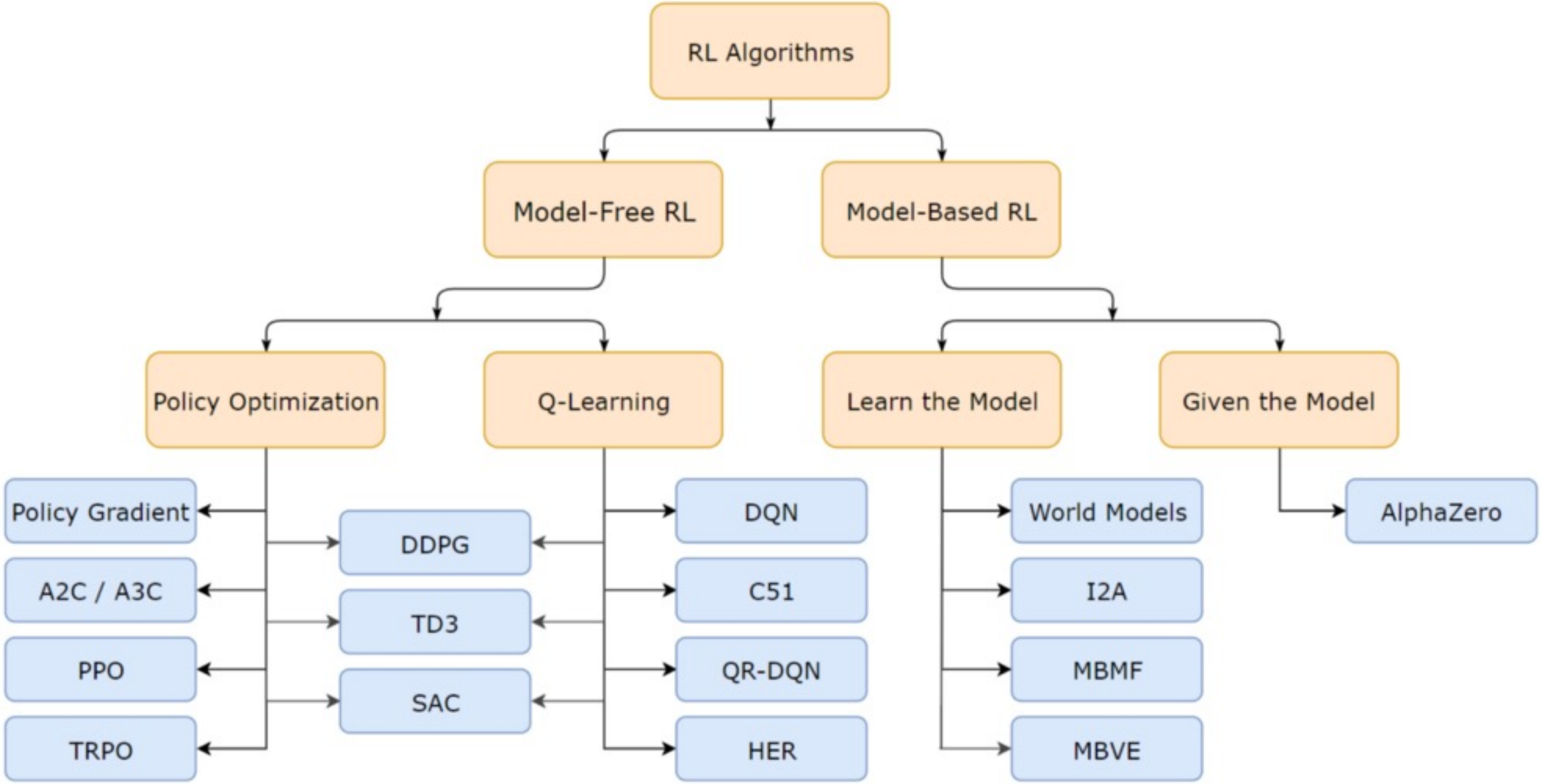
Value Function: $v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s]$

Model: $p_{ss'}^a = \mathbb{P}[S_{t+1} = \mathrm{s}', A_t = a]$

$p_s^a = \mathbb{E}[\mathrm{R}_{t+1} | \mathrm{S_t} = s, A_t = a]$

https://ww2.mathworks.cn/discovery/reinforcement-learning.html; https://towardsdatascience.com/reinforcement-learning-an-introduction-to-the-concepts-applications-and-code-ced6fbfd882d

Types of Reinforcement Learning



https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Other Popular Learning Paradigms

Transfer Learning

D_s Source Domain → D_t Target Domain

$$\min_{\theta}[\mathbb{E}_{(\boldsymbol{x},y)\in D^s}\mathcal{L}(f(\boldsymbol{x}),y)+\mathcal{L}_{dis}(g(D^s),g(D^t))]$$ Feature Transfer

$$\min_{\theta}[\mathbb{E}_{(\boldsymbol{x},y)\in D^s}\mathcal{L}(f\circ X_{s\to t}(\boldsymbol{x}),y)]$$ Sample Transfer

$$\min_{\theta\subset\theta_g\cup\theta_{h^*}}[\mathbb{E}_{(\boldsymbol{x},y)\in D^t}\mathcal{L}^*(h^*\circ g(\boldsymbol{x}),y)]$$ Model Transfer

f 为模型, g 为特征编码器，h 为任务头，θ 为模型参数，L(f(x), y)对应任务损失函数，g(D) 为数据集D 的样本特征集合，Ldis 为衡量特征集合分布差异的函数

Other Popular Learning Paradigms

Online Learning

$D_{old} = \{\boldsymbol{x}_i^{old}, y_i^{old}\}_{i=1}^{n_{old}}$

Existing Data

$D_{new} = \{\boldsymbol{x}_i^{new}, y_i^{new}\}_{i=1}^{n_{new}}$

New Data

$$\min_{\theta} [\mathbb{E}_{(\boldsymbol{x},y)\in D_{old}} \mathcal{L}(f(\boldsymbol{x}), y) + \mathbb{E}_{(\boldsymbol{x},y)\in D_{new}} \mathcal{L}(f(\boldsymbol{x}), y)]$$

Key problem: catastrophic forgetting

Other Popular Learning Paradigms

Knowledge Distillation

T

Teacher Model

S

Student Model

$$\min_{\theta_s} \mathbb{E}_{(x,y)\in D} \mathcal{L}_{sim}(S_{\theta_s}(\boldsymbol{x}), T_{\theta_t}(\boldsymbol{x}))$$

KL-divergence loss is the most commonly used distillation loss

Losses

Regression Losses

MSE: $$\mathcal{L}(f(X), Y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i))^2$$

MAE: $$\mathcal{L}(f(X), Y) = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(\boldsymbol{x}_i)|$$

Huber Loss: $$\mathcal{L}_\delta(f(\boldsymbol{x}), y) = \begin{cases} \frac{1}{2}(f(\boldsymbol{x}) - y)^2 & |f(\boldsymbol{x}) - y| < \delta \\ \delta|f(\boldsymbol{x}) - y| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

Losses

Classification Losses

Cross Entropy: $$\mathcal{L}_{CE}(y, \boldsymbol{p}) = -\sum_{c=1}^{C} \mathbb{I}(c \equiv y) \cdot \log(\boldsymbol{p}_c) = -\log(\boldsymbol{p}_y)$$

Binary Cross Entropy: $$\mathcal{L}_{BCE}(y, p) = y \cdot \log(p) + (1 - y) \cdot \log(p)$$

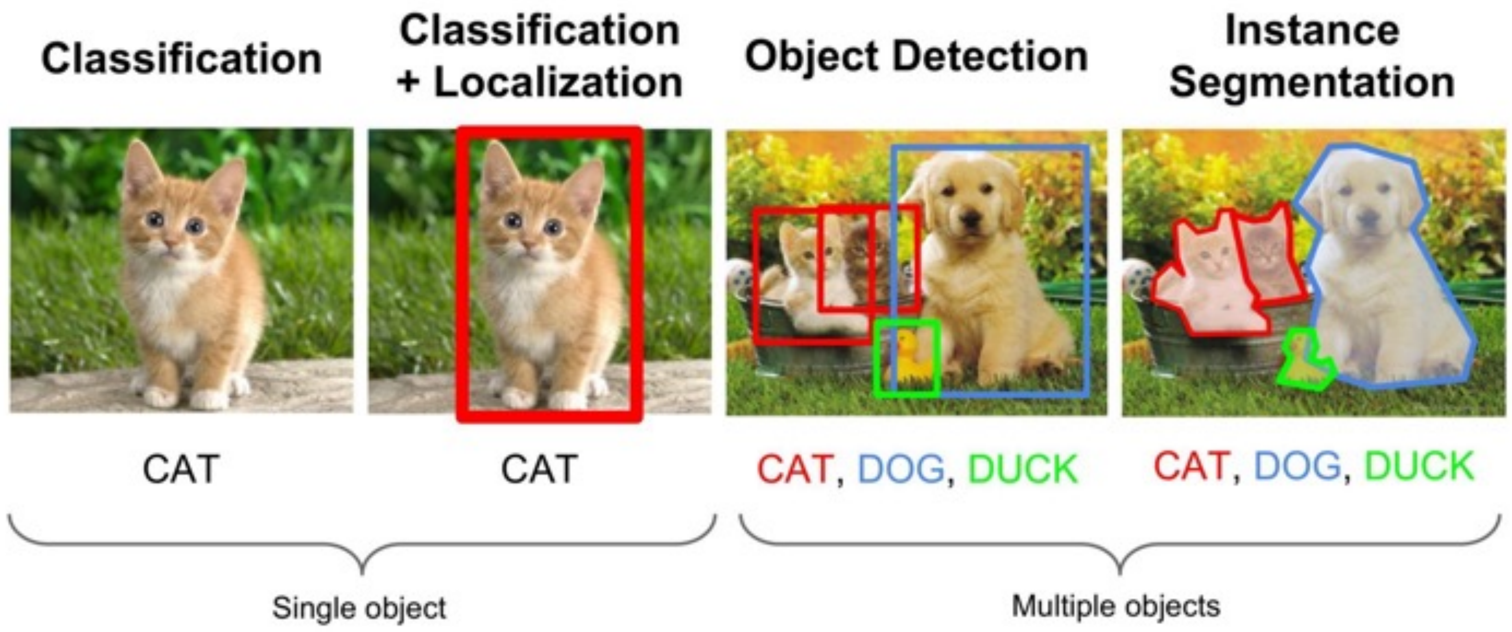
Generalized Cross Entropy: $$\mathcal{L}_q(f(\mathbf{x}; \boldsymbol{\theta}), y) = \frac{1 - f_j(\mathbf{x})^q}{q}, \quad \mathrm{q} \in (0,1]$$

Symmetric Cross Entropy: $$SCE = \alpha H(\boldsymbol{q}, \mathbf{p}) + \beta H(\hat{\mathbf{p}}, \boldsymbol{q})$$

Focal Loss: $$FL(\boldsymbol{p}_y) = -(1 - \boldsymbol{p}_y)^{\gamma} \log(\boldsymbol{p}_y), \gamma \geq 0$$

Losses

Object Detection Losses



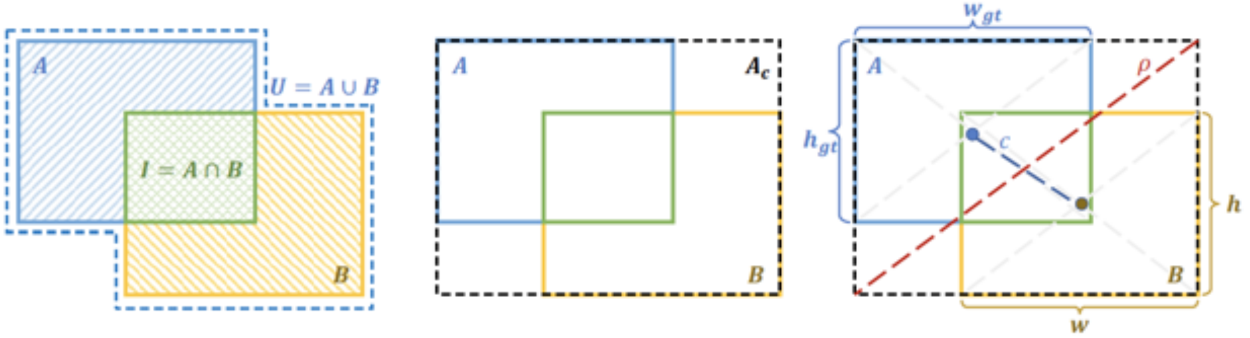
Bounding Box Regression + Classification

https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

Losses

Object Detection Losses

方法名称	*IoU 定义	损失函数
IoU-loss	$IoU = \|A \cap B\| / \|A \cup B\|$	$\mathcal{L}_{IoU} = 1 - IOU$
GIoU-loss	$GIoU = IoU - \|A_c - U\| / \|A_c\|$	$\mathcal{L}_{GIoU} = 1 - GIoU$
DIoU-loss	$DIoU = IoU - \rho^2(b, b^{gt})/c^2$	$\mathcal{L}_{DIoU} = 1 - DIoU$
CIoU-loss	$CIoU = DIoU - \beta v$ $v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2$	$\mathcal{L}_{CIoU} = 1 - CIoU$



$$\mathcal{L}_{\alpha IoU} = 1 - IoU^{\alpha} + P^{\alpha}(b, b^{gt})$$

$$\begin{cases} \mathcal{L}_{\alpha-IoU} = 1 - IoU^{\alpha} \\ \mathcal{L}_{\alpha-GIoU} = 1 - IoU^{\alpha} + \left(\frac{|A_c - U|}{|A_c|}\right)^{\alpha} \\ \mathcal{L}_{\alpha-DIoU} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(b, b^{gt})}{c^{2\alpha}} \\ \mathcal{L}_{\alpha-CIoU} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(b, b^{gt})}{c^{2\alpha}} + (\beta v)^{\alpha} \end{cases}$$

Losses

Generative Losses

- ◆ 自回归模型（Autoregressive）
- ◆ 能量模型（Energy based models）
- ◆ 流模型（Flows）
- ◆ 变分自编码器（VAE，variational
- ◆ autoencoder）
- ◆ 生成对抗网络（GAN，generative adversarial network）
- ◆ 扩散模型（Diffusion models）

方法	损失函数
GAN	$\mathcal{L}_D = -(\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))])$ $\mathcal{L}_G = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$
LSGAN	$\mathcal{L}_D = (\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[(D(\boldsymbol{x}) - 1)^2] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[(D(G(\boldsymbol{z})))^2])$ $\mathcal{L}_G = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[(D(G(\boldsymbol{z})) - 1)^2]$
WGAN	$\mathcal{L}_D = (\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[D(G(\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[D(\boldsymbol{x})])$ $\mathcal{L}_G = -\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[D(G(\boldsymbol{z}))]$ $\theta_D = \mathrm{clip}(\theta_D, -c, c)$, c 是截断参数
Hinge Loss	$\mathcal{L}_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\min(0, -1 + D(\boldsymbol{x}))]$ $-\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\min(0, -1 - D(G(\boldsymbol{z}))))]$ $\mathcal{L}_G = -\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} D(G(\boldsymbol{z}))$

Optimizers

Gradient Descent (GD)

$$\theta' = \theta - \eta\nabla_\theta = \theta - \eta\frac{1}{N}\sum_{i=1}^{N}\nabla_\theta\mathcal{L}(y_i, f(\boldsymbol{x}_i);\theta)$$

Stochastic Gradient Descent (SGD) for mini-batch based training

$$\theta' = \theta - \eta\nabla_\theta = \theta - \eta\frac{1}{N'}\sum_{i=1}^{N'}\nabla_\theta\mathcal{L}(y_i, f(\boldsymbol{x}_i);\theta)$$

SGD with Momentum

$$\boldsymbol{v}_t = \gamma\boldsymbol{v}_{t-1} + \eta\nabla_\theta J(\boldsymbol{\theta}_t)$$
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \boldsymbol{v}_t$$

SGD with Nesterov Acceleration

$$\boldsymbol{v}_t = \gamma\boldsymbol{v}_{t-1} + \eta\nabla_\theta J(\boldsymbol{\theta}_t - \gamma\boldsymbol{v}_{t-1})$$
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \boldsymbol{v}_t$$

Optimizers

AdaGrad

$$\boldsymbol{\theta}_{t+1,i} = \boldsymbol{\theta}_{t,i} - \frac{\eta}{\sqrt{\sum_t g_t^2 + \epsilon}} \cdot \boldsymbol{g}_{t_i}$$

RMSprop

$$E[\boldsymbol{g}^2]_t = \gamma E[\boldsymbol{g}^2]_{t-1} + (1-\gamma)\boldsymbol{g}_t^2$$

$$\boldsymbol{\theta}_{t+1,i} = \boldsymbol{\theta}_{t,i} - \frac{\eta}{\sqrt{E[\boldsymbol{g}^2]_{t,i} + \epsilon}} \cdot \boldsymbol{g}_{t_i}$$

Adadelta

$$\boldsymbol{\theta}_{t+1,i} = \boldsymbol{\theta}_{t,i} - \frac{\sqrt{E[\boldsymbol{\Delta\theta}^2]_{t,i} + \epsilon}}{\sqrt{E[\boldsymbol{g}^2]_{t,i} + \epsilon}} \cdot \boldsymbol{g}_{t_i}$$

Adam

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

谢谢！下周见！

Email: xingjunma@fudan.edu.cn
Personal page: www.xingjunma.com
Office: 江湾校区交叉二号楼D5025