深度学习应用与工程实践
2.机器学习
2. Machine Learning Types

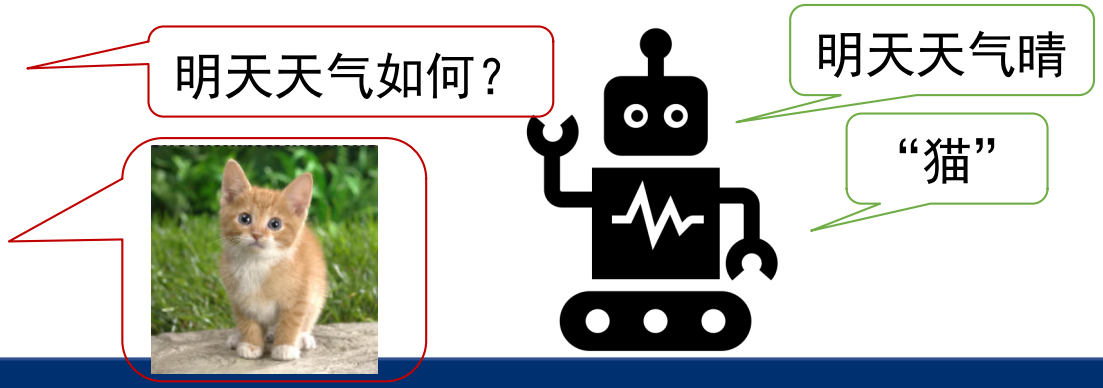
李冰
副研究员，硕士生导师
交叉科学研究院
首都师范大学

这节课

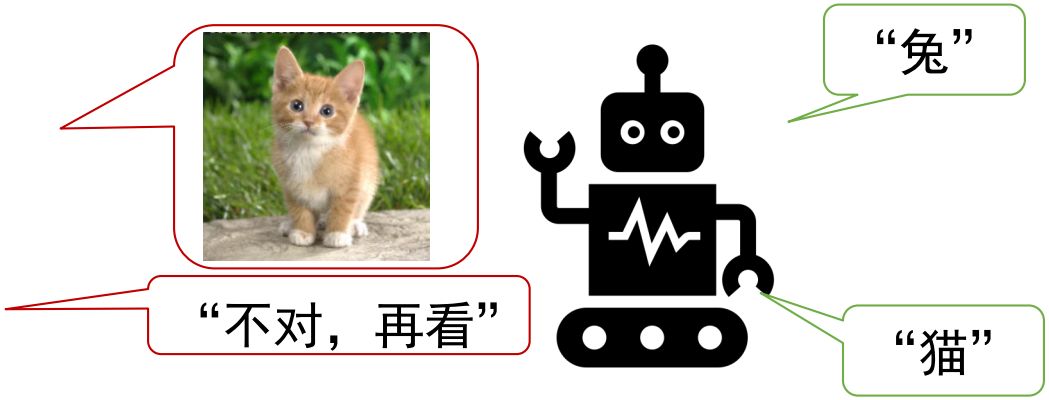
- 机器学习概述
  - 学习类型
- 深度学习
- 不同任务中深度学习的应用

机器学习

- 机器学习是人工智能 (AI) 和计算机科学的分支
  - 使用数据和算法来模仿人类学习的方式。
- 深度学习是机器学习的子类。
  - 机器学习依赖于人工干预进行学习。
    - 人类专家确定一组特征，以了解数据输入之间的差异，通常需要更为结构化的数据以进行学习。
  - 深度学习
    - 自动执行数据中特征提取，消除人工干预，使用更大的数据集。



机器学习

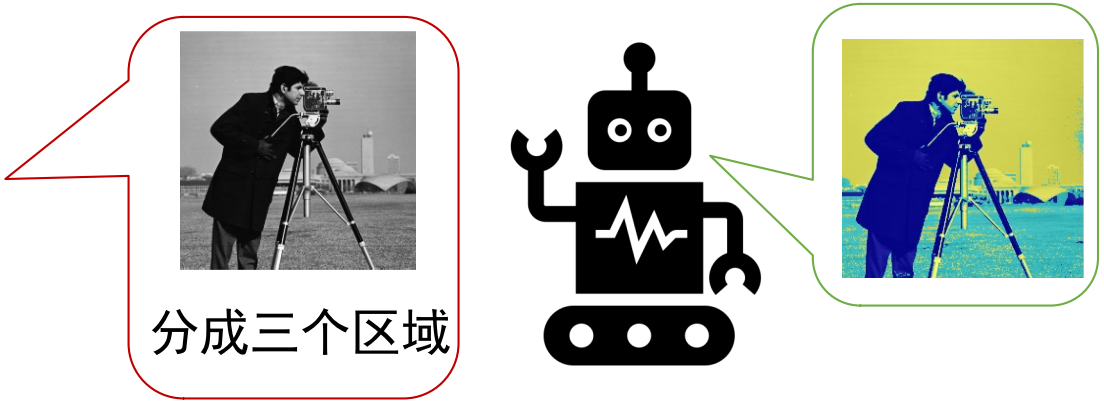


➢ 数据有正确标注(label)

监督式学习（也称为监督式机器学习）

- 使用标签化数据集训练算法，以准确分类数据或预测结果。
- 输入数据进入模型后，该方法会调整权重，直到模型拟合。
- 方法包括神经网络、朴素贝叶斯、**线性回归**、**逻辑回归**、随机森林、**支持向量机 (SVM)** 等。

机器学习

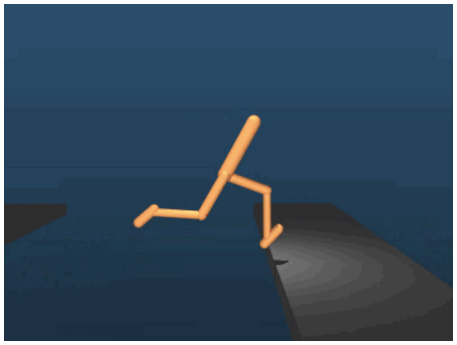


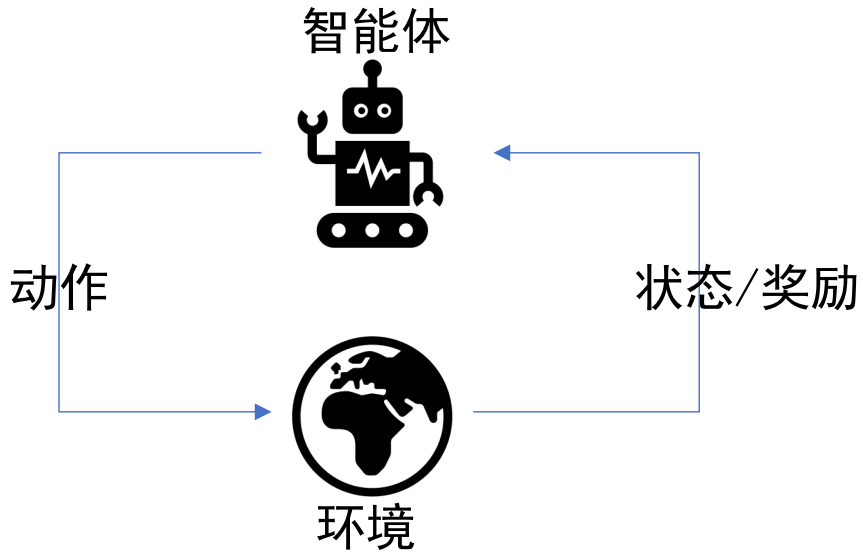
- 数据没有标注(label)

非监督式学习

- 分析未标签化数据集并形成聚类，发现隐藏的模式或数据分组
- 发现信息的相似性和差异，无需人工干预。探索性数据分析、交叉销售策略、客户细分、图像和模式识别的理想解决方案。
- 该方法还通过降维过程，减少模型中特征的数量；主要成分分析 (PCA) 和奇异值分解 (SVD) 是两种常见的方法。
- 在无监督学习中使用的其他算法包括神经网络、k-平均值聚类、概率聚类方法等

机器学习



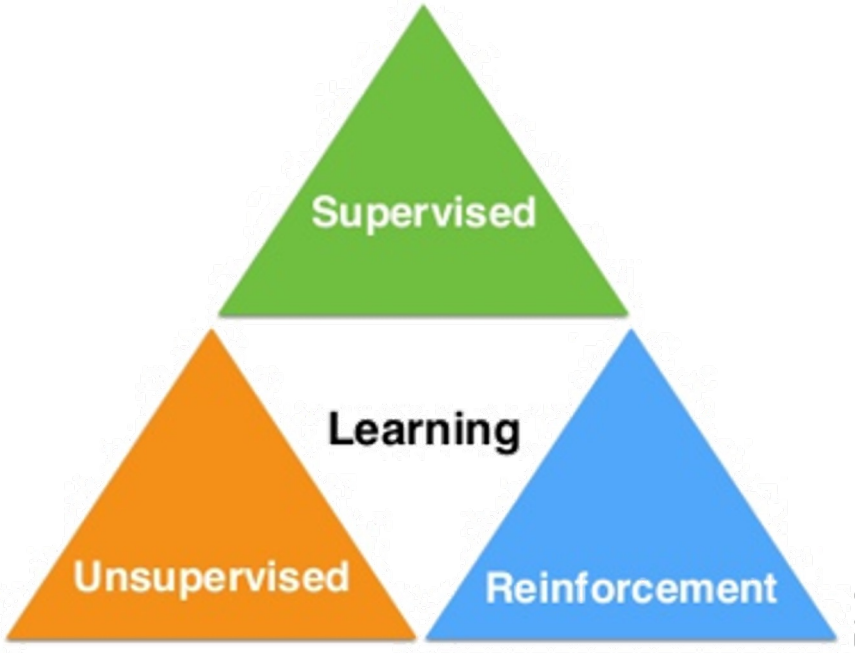


强化机器学习

- 通过不断试错进行学习，提出最佳建议或者最佳策略
- 策略性游戏，围棋、跳棋等

机器学习(Machine Learning)

- 一类从数据中自动分析获得规律，并利用规律对未知数据进行预测的算法。
- 监督学习:
  - f(输入) → 目标 (连续数值/离散量)
- 非监督学习:
  - f(输入) → 相似度/可能性
- 强化学习:
  - f(状态, 行为) → 价值

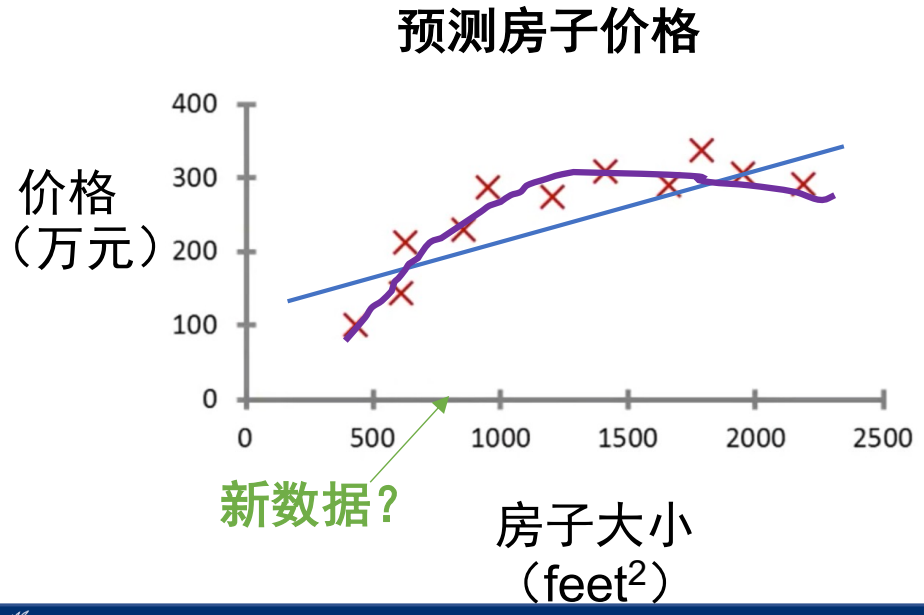


Credit: 台湾大学 李宏毅教授

监督学习 Supervised learning:

- 函数: f(输入) → 目标
  - 连续具体数值: 回归 Regression
  - 离散: 分类 Classification

回归问题



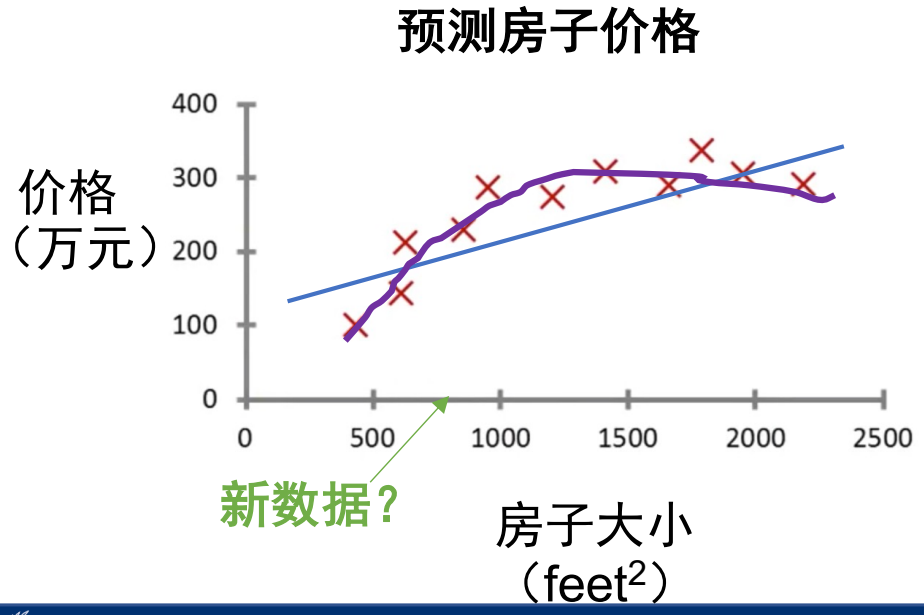
数据：**房子面积**，房子价格
目标：有一定面积的房子，价格是多少？

特征	目标值
2104	460
1416	232
1534	315
852	178

监督学习 Supervised learning:

- 函数: f(输入) → 目标
  - 连续具体数值: 回归 Regression
  - 离散: 分类 Classification

回归问题



数据：**房子面积**，房子价格
目标：有一定面积的房子，价格是多少？

特征	目标值
2104	460
1416	232
1534	315
852	178

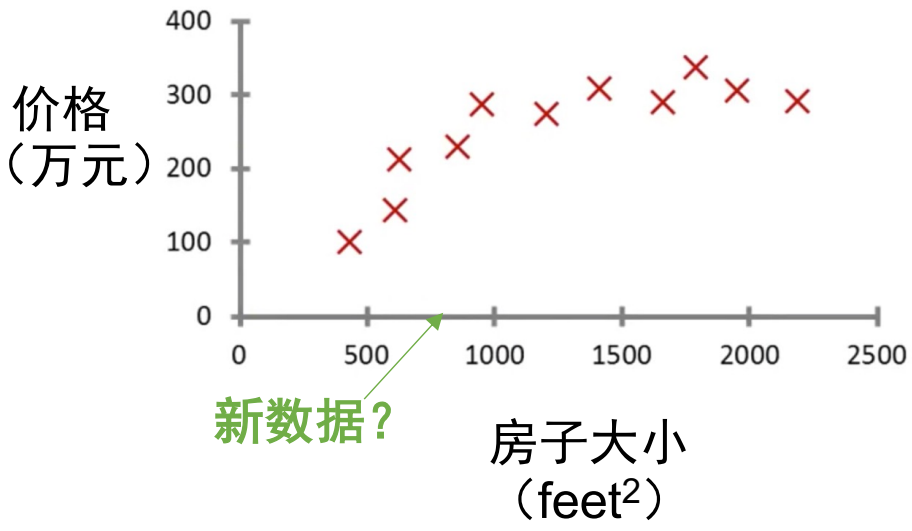
$f(x) = wx + b$

线性回归

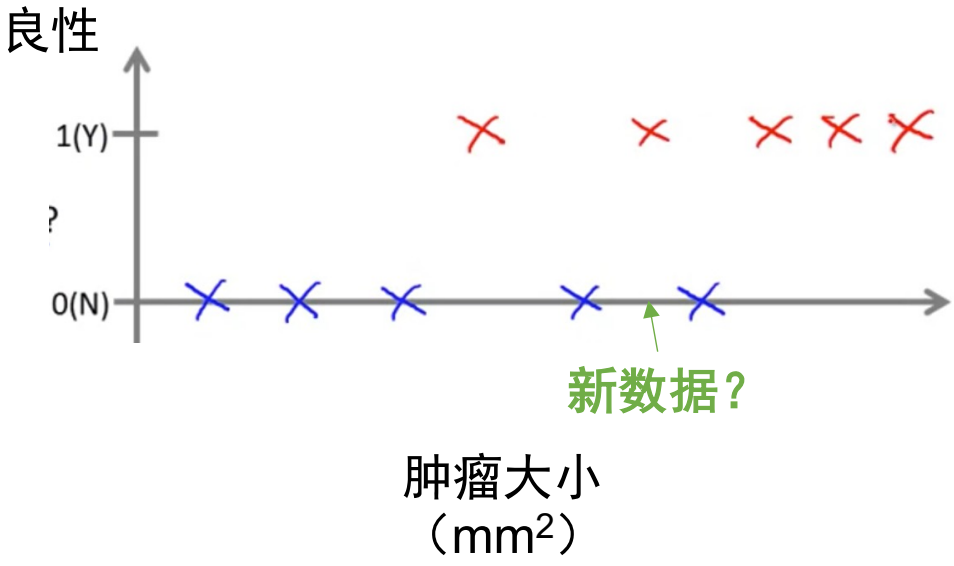
监督学习 Supervised learning:

- 函数: f(输入) → 目标
  - 连续具体数值: 回归 Regression
  - 离散: 分类 Classification

线性回归

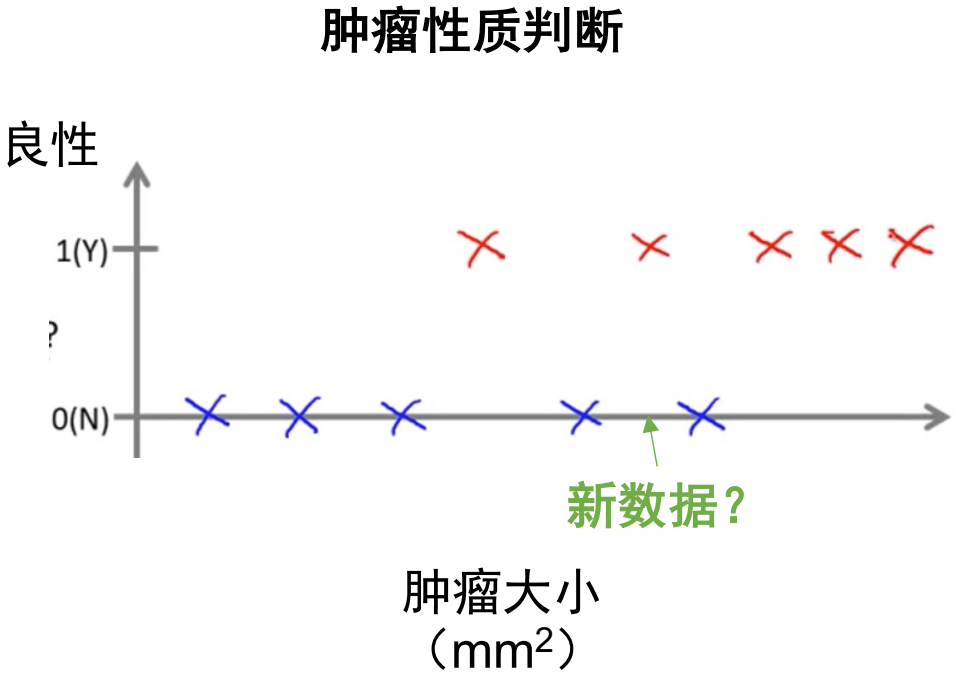


分类



监督学习 Supervised learning:

- 函数: f(输入) → 目标
  - 连续具体数值: 回归 Regression
  - 离散: 分类 Classification

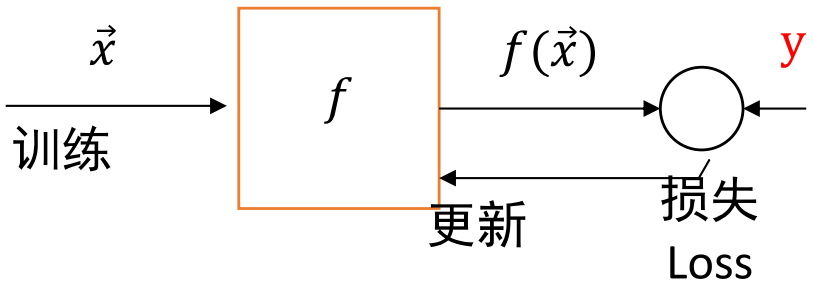


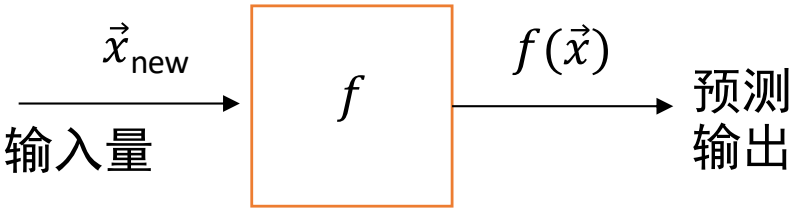
数据：**肿瘤大小**，恶性/良性
目标：肿瘤大小，对肿瘤分类

特征	类别
2	0
3.2	1
4	0
1.2	0

回归 Regression

- 通过数据找出输入到输出的映射关系
- 利用找到的这个关系进行预测
- 输出是连续值。



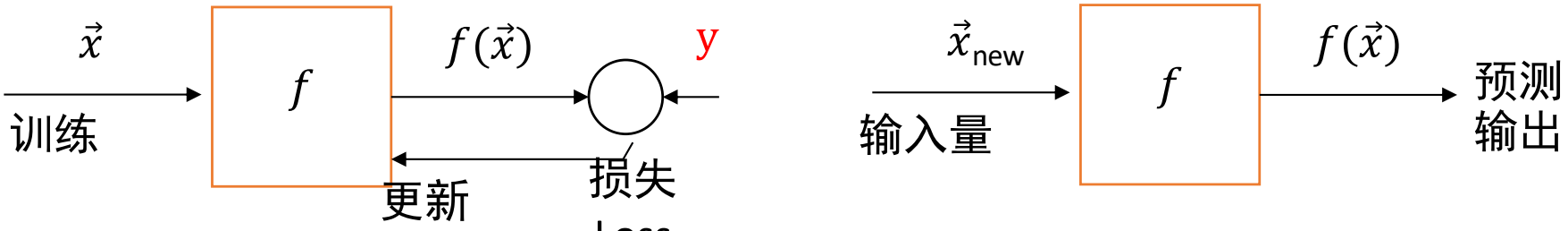


训练数据集
f(x)：学习者
x： 输入训练样本，样本有多个特征。
模型的输出y： 预测
损失Loss： 预测值与真实值之间差异

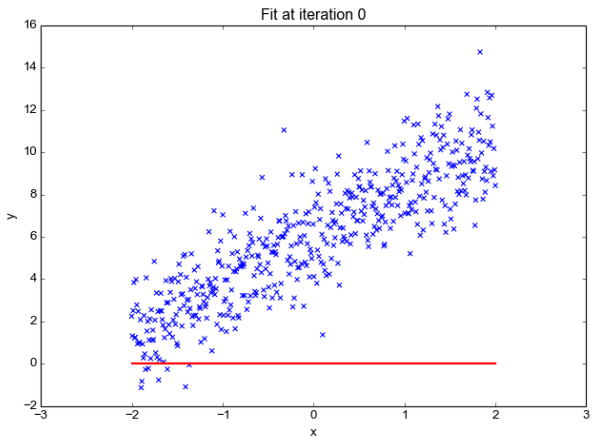
新的数据x_{new}
学习者得到预测输出

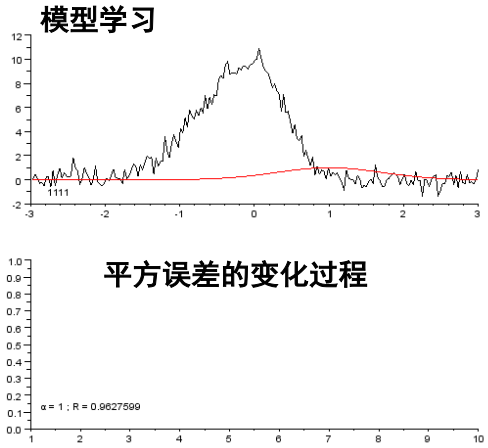
回归 Regression

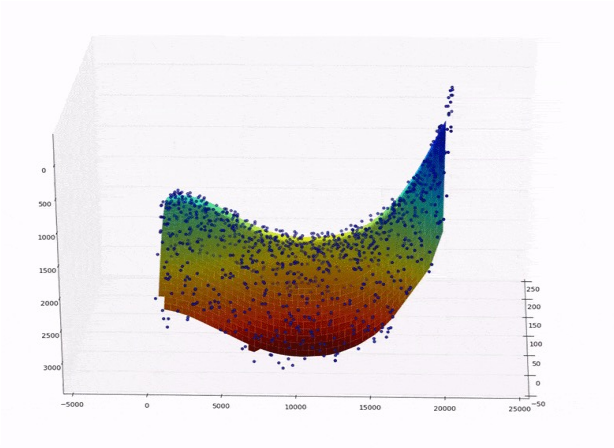
- 通过数据找出输入到输出的映射关系
- 利用找到的这个关系进行预测
- 输出是连续值。



- 线性回归
- 非线性回归
- 支持向量回归



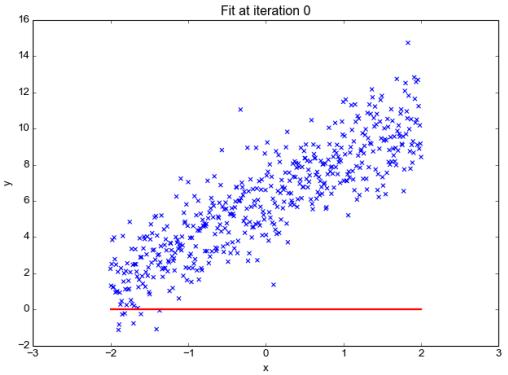




Source of image: https://towardsdatascience.com/linear-regression-the-easier-way-6f941aa471ea; https://commons.wikimedia.org/wiki/File:Regression_pic_assymetrique.gif; http://www.semspirit.com/artificial-intelligence/machine-learning/regression/support-vector-regression/support-vector-regression-in-python/

传统回归分析方法

- 线性回归
  - 学得一个通过属性进行线性组合进行预测的函数。



假设数据只有一个属性

$$\vec{x} = [x_1 \quad x_2 \quad \dots \quad x_n]$$

$$\mathrm{f}(x_i) = w x_i \ (i = 0 \ \dots n)$$

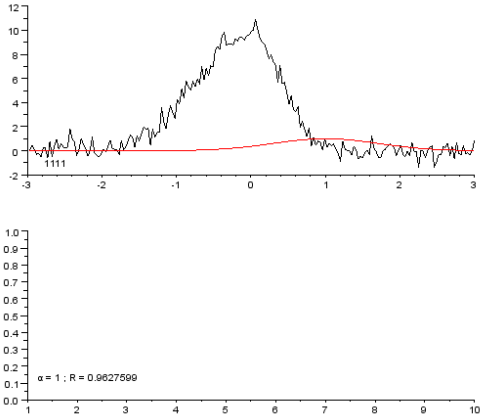
均方误差 $\boldsymbol{\omega} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \sum_{for\ all\ data} \|\vec{x}\boldsymbol{\omega} - y\|_2^2$

线性回归中，最小二乘法是试图找到一条曲线，使所有样本到直线上的欧式距离之和最小。
线性回归模型的最小二乘”参数估计”

$$E = \sum_{for\ all\ data} \|\vec{x}\boldsymbol{\omega} - y\|_2^2 \qquad \frac{\partial E}{\partial w}$$

传统回归分析方法

- 多元线性回归
  - 样本用多个属性描述



$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ {x_1}^2 & {x_2}^2 & \dots & {x_n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ {x_1}^{p-1} & {x_2}^{p-1} & \dots & {x_n}^{p-1} \end{bmatrix}$$

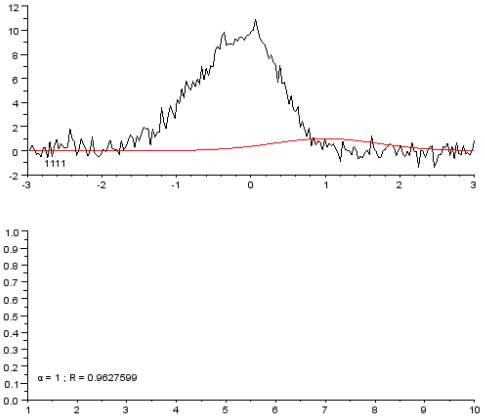
$$\boldsymbol{\omega} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \sum_{for\ all\ data} \|\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{\omega} - \boldsymbol{y}\|_2^2, (\boldsymbol{\omega} \in \mathbb{R}^p)$$

当$\boldsymbol{X}^T\boldsymbol{X}$为满秩矩阵或正定矩阵时

$$\omega = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

传统回归分析方法

- 多元线性回归
  - 样本用多个属性描述



$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ {x_1}^2 & {x_2}^2 & \dots & {x_n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ {x_1}^{p-1} & {x_2}^{p-1} & \dots & {x_n}^{p-1} \end{bmatrix}$$

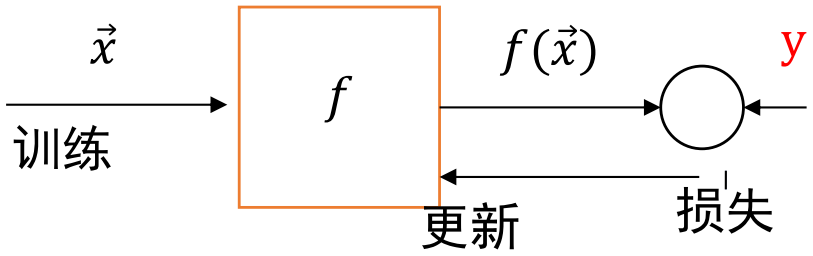
$$\boldsymbol{\omega} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \sum_{for\ all\ data} \|\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{\omega} - \boldsymbol{y}\|_2^2, (\boldsymbol{\omega} \in \mathbb{R}^p)$$

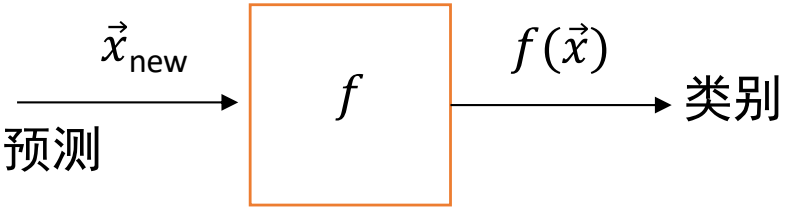
p： 数据点的个数，n： 数据维度。

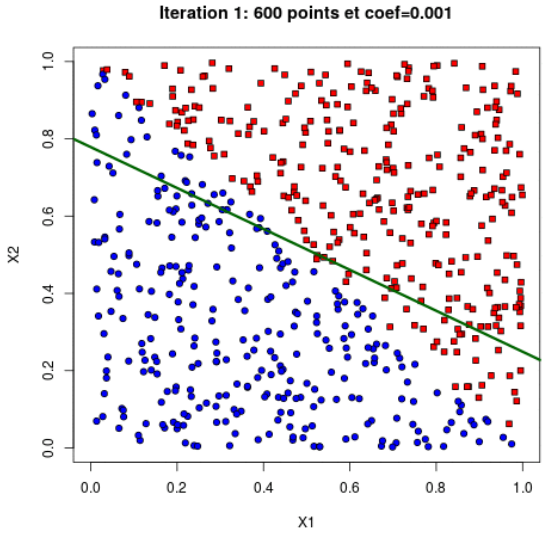
- 当p>>n的时候，不做任何其他假设或者限制的话，学习问题基本上是没法进行的。
- p 越大，通常会导致模型越复杂，但是反过来 n 有很小，于是就会出现很严重的 overfitting 问题。
- 如果p>n 的话，矩阵X将会不是满秩的，将会有无穷多个解。如果我们从所有可行解里随机选一个的话，很可能并不是真正好的解。

分类

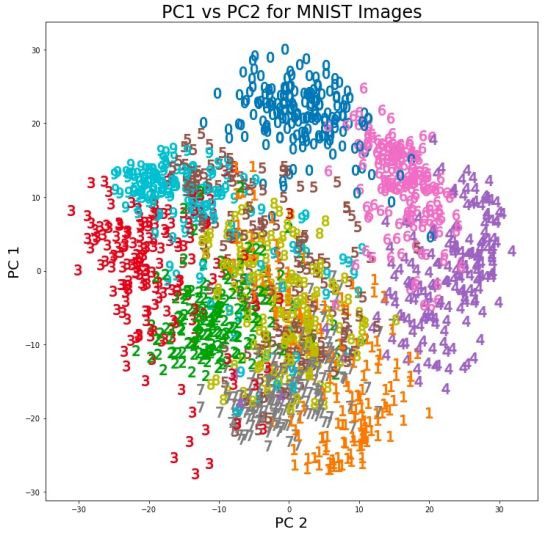
- 与回归问题相似。
- 唯一的区别：标签y 和输出是离散的.







线性

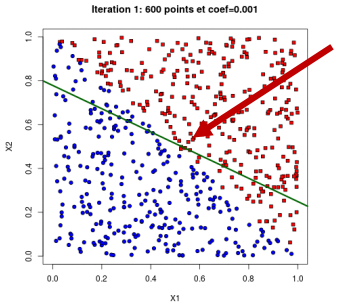


非线性、多分类

Source of Image: http://www-lisic.univ-littoral.fr/~teytaud/files/Cours/Apprentissage/perceptron/; https://www.mathworks.com/matlabcentral/fileexchange/52003-viewboundary

传统分类方法

决策边界

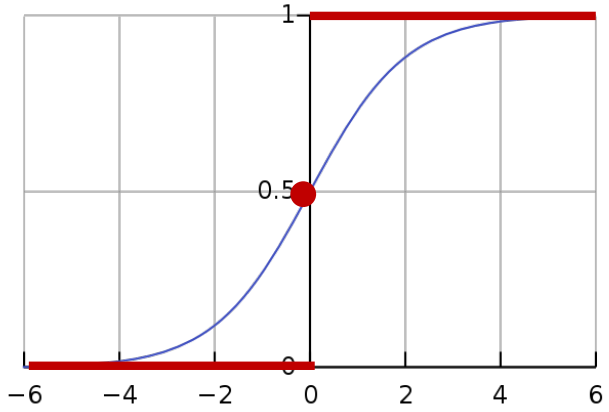


- 二分类任务，输出标记$y \in \{0,1\}$.
- Logistic回归 (对数几率回归)
  - 决策边界
  - 作用在n维空间，将不同样本分开的平面或曲面在对数几率回归中 $\vec{x}\boldsymbol{\omega} = 0$

$$y = \begin{cases} 1, \omega X > 0 \\ 0.5 \quad \omega X = 0 \\ 0, \omega X < 0 \end{cases}$$

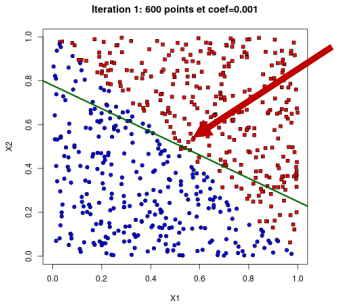
但，不可导，不连续

阶跃函数



传统分类方法

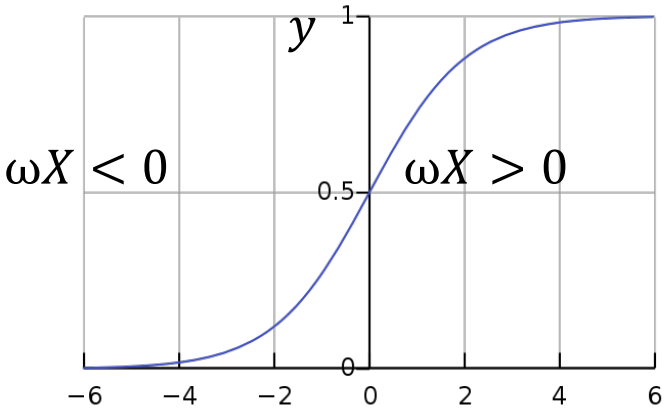
决策边界



- 二分类任务，输出标记$y \in \{0,1\}$.
- Logistic回归 (对数几率回归)
  - 决策边界
  - 作用在n维空间，将不同样本分开的平面或曲面在对数几率回归中 $\vec{x}\boldsymbol{\omega} = 0$

Sigmoid 函数

$$y = \frac{1}{1 + e^{-\omega X}}$$



$$\omega X = \ln\left(\frac{y}{1-y}\right)$$

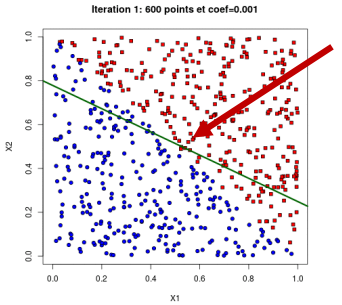
$\frac{y}{1-y}$ **几率（odds）**

y 表示 样本视为正例的可能性

$1-y$ 表示样本为反例的可能性

传统分类方法

决策边界



- Logistic回归 (对数几率回归)

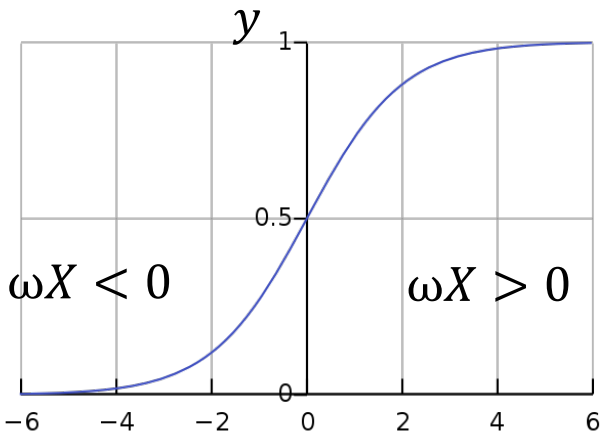
Sigmoid 函数 $y = \dfrac{1}{1+e^{-\omega X}}$

$$\omega X = \ln\left(\frac{y}{1-y}\right)$$

y 看作后验概率估计 $P(y=1|x)$

$$P(y=1|x) = \frac{1}{1+e^{\omega X}}$$

$$P(y=0|x) = 1 - \frac{1}{1+e^{\omega X}}$$



用极大似然法，来估计w。
最大化对数似然

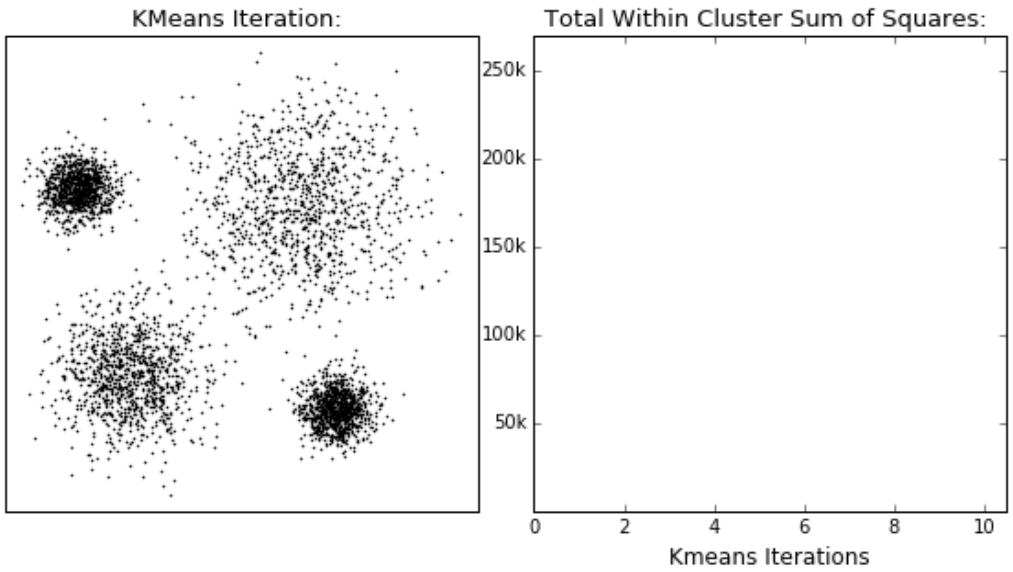
$$\sum_{i=1} \ln P(y_i|\boldsymbol{x}_i;\ \boldsymbol{w})$$

非监督学习

- 函数: f(输入) -> 相似性
  - 分布中采样；
  - 分布中去噪
  - 数据中的相关样本分类
- 聚类算法

聚类

K-均值算法



1.随机地选择k个对象，每个对象初始地代表了一个簇的中心；

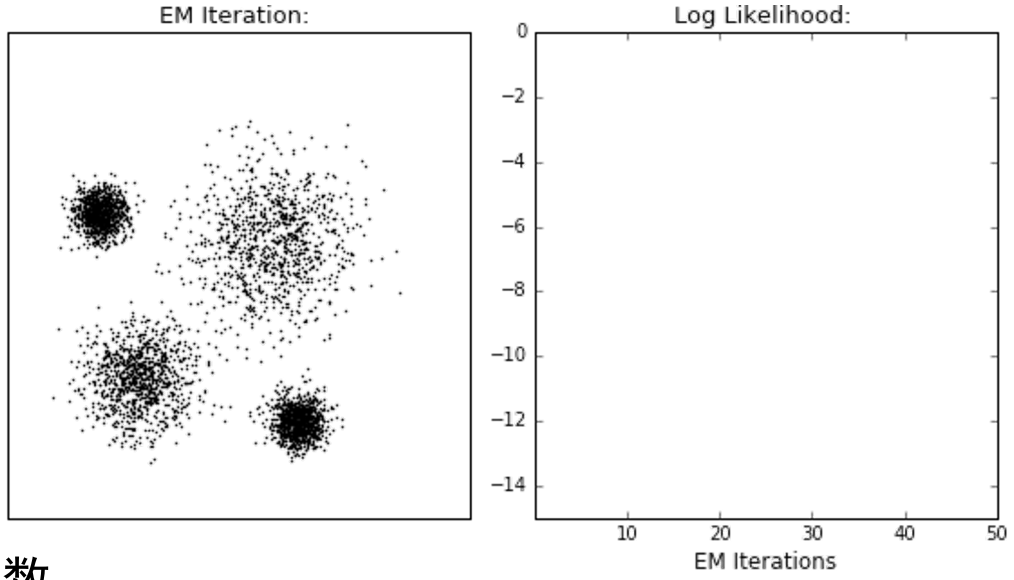
2. 对剩余的每个对象，根据其与各簇中心的距离，将它赋给最近的簇；

3. 重新计算每个簇的平均值，更新为新的簇中心；

4. 不断重复2、3，直到收敛（每个Cluster中样本数目变化不大)

Source of image: https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/

聚类



高斯混合模型

Gaussian mixture model (GMM)

$$p_{\mathcal{M}}(x) = \sum_{i=1}^{k} \alpha_i \cdot p(x \mid \mu_i, \Sigma_i)$$

高斯分布的n维随机向量x的概率密度函数

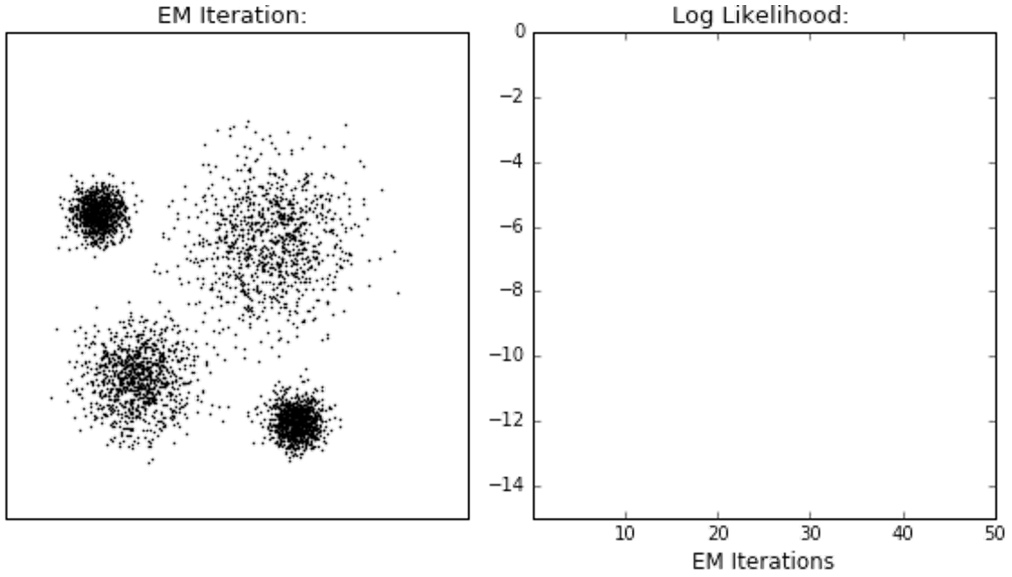
$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

μ是n维均值向量Σ是n×n的协方差矩阵， μ_i与Σ_i是第i个高斯混合成分的参数，而$\alpha_i>0$为相应的“混合系数”(mixture coefficient), $\sum_{i=1}^{k} \alpha_i = 1$

给定聚类簇数k，通过给定的数据集，以某一种参数估计的方法，推导出每一个混合成分的参数:均值向量μ,协方差矩阵Σ和混合系数α

每一个多元高斯分布成分即对应于聚类后的一个簇。

聚类



高斯混合模型

Gaussian mixture model (GMM)

$$p_{\mathcal{M}}(x) = \sum_{i=1}^{k} \alpha_i \cdot p(x \mid \mu_i, \Sigma_i)$$

最大化对数似然

$$L = log \prod_{j=1}^{m} p(x) = \sum_{j=1}^{m} log \sum_{i=1}^{k} (\alpha_i \cdot p(x \mid \mu_i, \Sigma_i))$$

$$\frac{\partial L}{\partial \mu_i} = \sum_{j=1}^{m} \underbrace{\frac{\alpha_i \cdot p(x_j \mid u_i, \Sigma_i)}{\sum_{i=1}^{k} \alpha_l \cdot p(x_j \mid \mu_l, \Sigma_l)}}_{\gamma_{ji}} (x_j - \mu_i) = 0 \qquad \mu_i = \sum_{j=1}^{m} \frac{\gamma_{ji} \cdot x_j}{\sum_{j=1}^{m} \gamma_{ji}} \qquad \Sigma_i = \sum_{j=1}^{m} \frac{\gamma_{ji}(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{m} \gamma_{ji}}$$

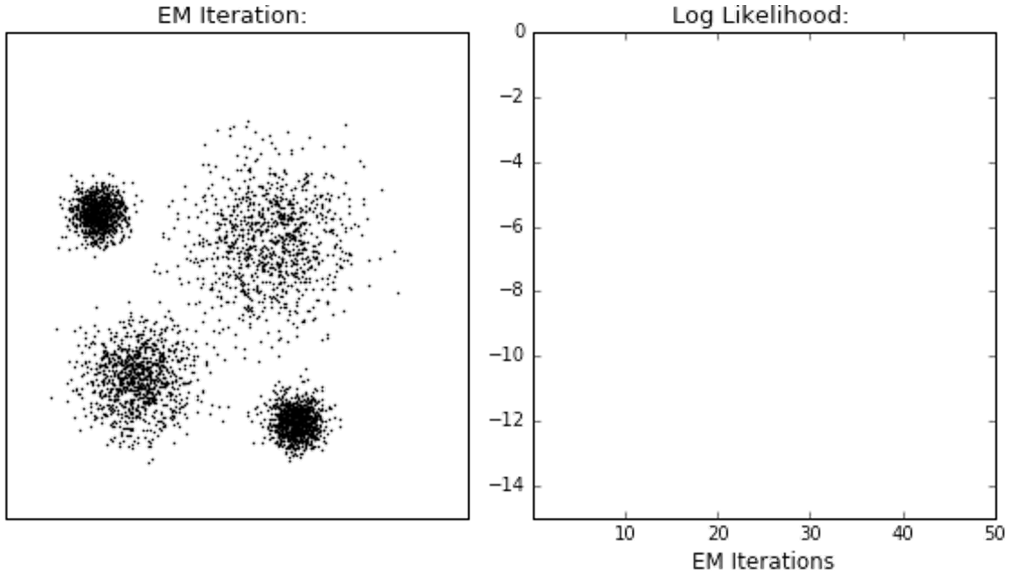
$$L + \lambda(\sum_{i=1}^{k} \alpha_i - 1) \qquad \sum_{j=1}^{m} \frac{\alpha_i \cdot p(x_j \mid u_i, \Sigma_i)}{\sum_{i=1}^{k} \alpha_l \cdot p(x_j \mid \mu_l, \Sigma_l)} + \lambda = 0 \qquad \alpha_i = \frac{1}{m} \sum_{j=1}^{m} \gamma_{ji}$$

聚类

高斯混合模型

Gaussian mixture model (GMM)

$$p_{\mathcal{M}}(x) = \sum_{i=1}^{k} \alpha_i \cdot p(x \mid \mu_i, \Sigma_i)$$



1. 选择簇的数量（与K-Means类似）并随机初始化每个簇的高斯分布参数（均值和方差）。

2. 给定每个簇的高斯分布，计算每个数据点属于每个簇的概率。 $\frac{\alpha_i \cdot p(x_j \mid u_i, \Sigma_i)}{\sum_{i=1}^{k} \alpha_l \cdot p(x_j \mid \mu_l, \Sigma_l)}$

一个点越靠近高斯分布的中心就越可能属于该簇。

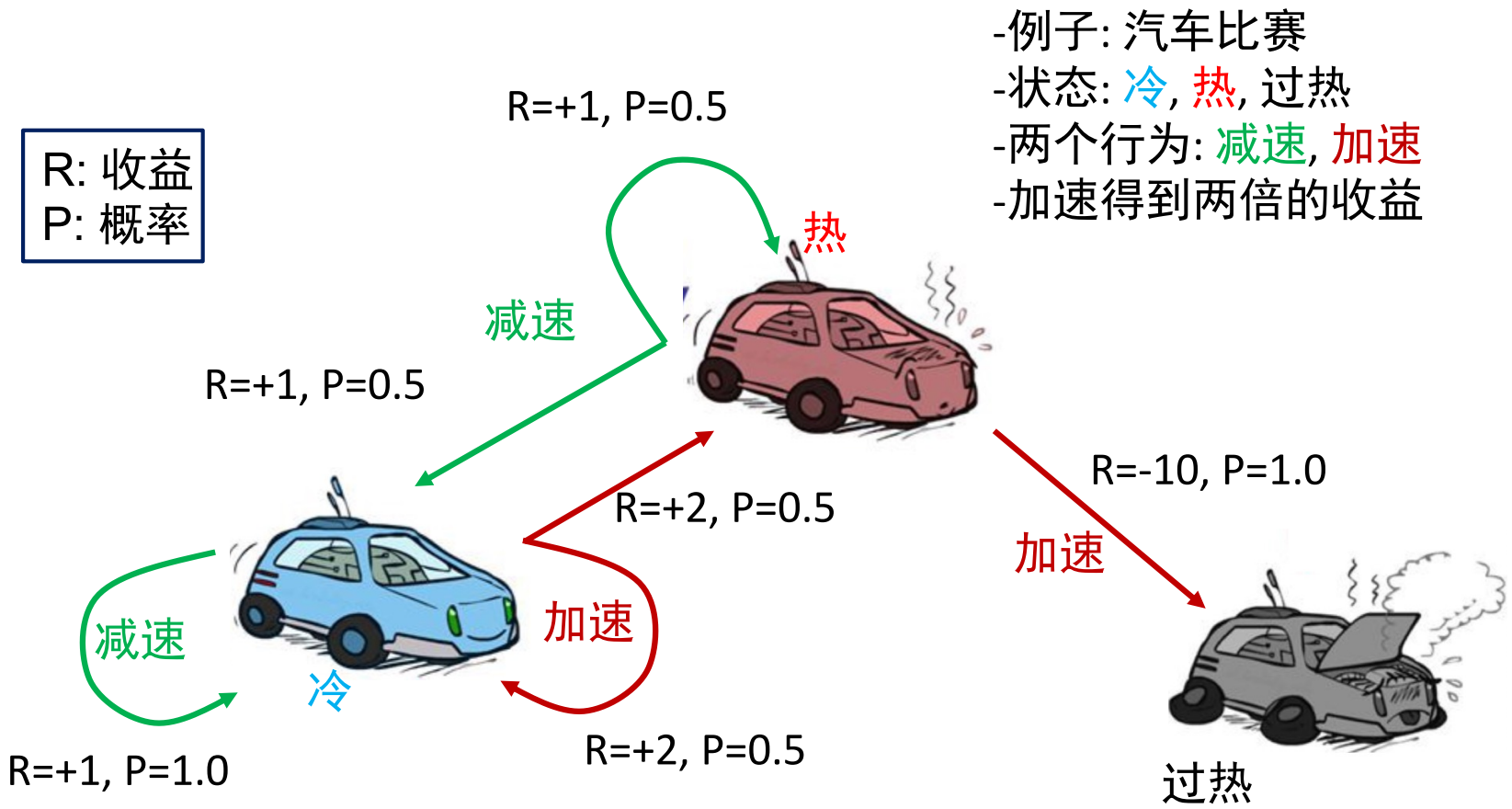
3. 根据均值，协方差的定义以及2步求出的后验概率，更新均值向量，协方差矩阵式，和混合系数； $\Sigma_i = \sum_{j=1}^{m} \frac{\gamma_{ji}(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{m} \gamma_{ji}}$ $\mu_i = \sum_{j=1}^{m} \frac{\gamma_{ji} \cdot x_j}{\sum_{j=1}^{m} \gamma_{ji}}$ $\alpha_i = \frac{1}{m} \sum_{j=1}^{m} \gamma_{ji}$

4. 重复迭代2和3直到在迭代中的变化不大。

5. 对于每一个样本点，根据贝叶斯定理计算出其属于每一个簇的后验概率，并将样本划分到后验概率最大的簇上去

强化学习

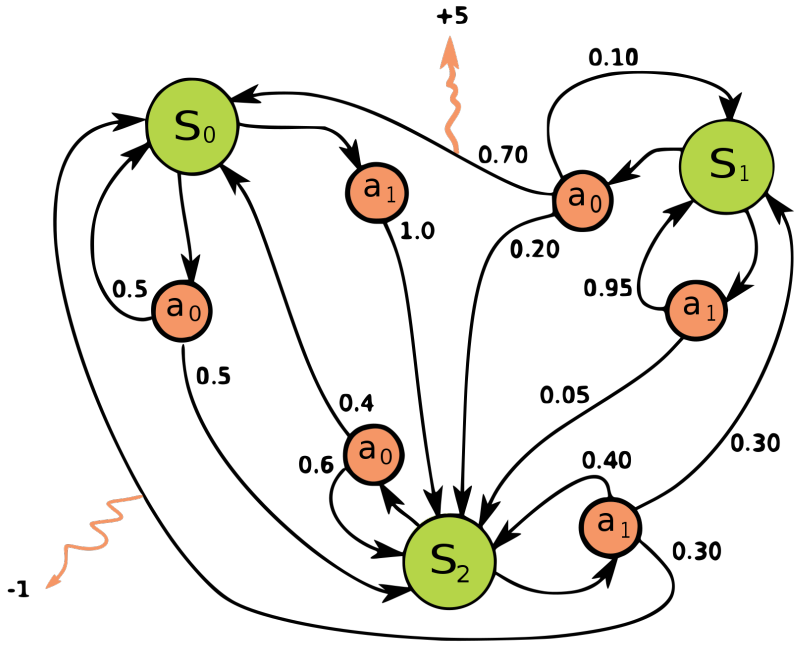
- 函数: f(状态, 行动) ->反馈值Reward Value
- 基本的强化学习模拟**马尔科夫决策过程**.



Source of image: https://medium.com/@sanchittanwar75/markov-chains-and-markov-decision-process-e91cda7fa8f2

传统强化学习

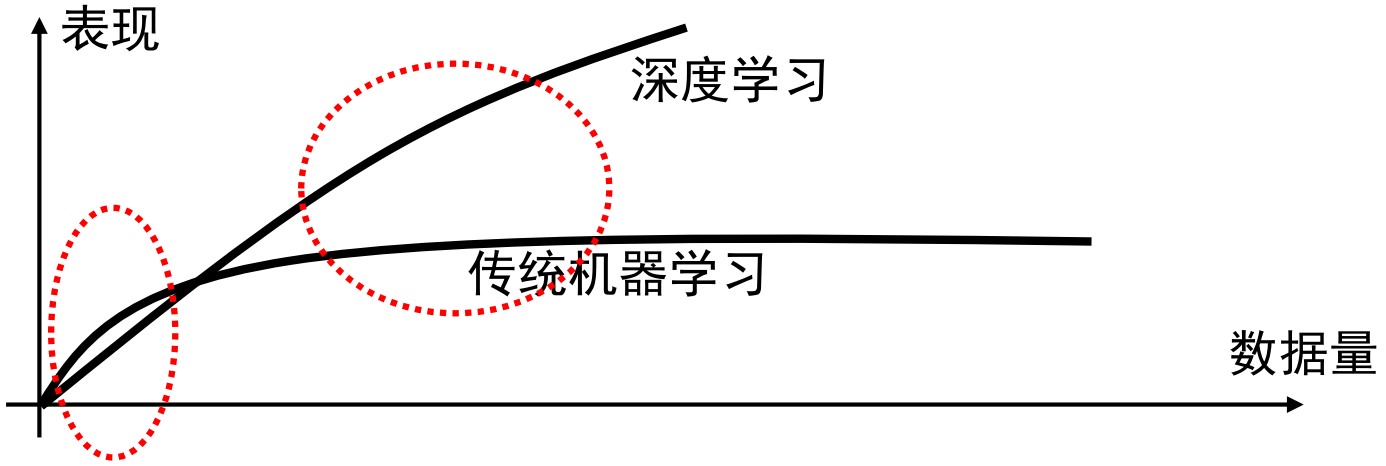
- 马尔科夫决策过程
  - 从环境中不断尝试学到一个策略；根据这个策略，在状态x下，我们能够知道下一步要执行的动作。
  - 目标是选择一个策略使得随机奖励的累积最大。
  - $\sum_{t=0}^{\infty} \gamma^t R_{at}(S_t, S_{t+1})$



Source of image: https://en.wikipedia.org/wiki/Markov_decision_process

为什么需要深度神经网络

- 在现实世界。
  - 输入数据是高维的
  - 映射函数复杂
  - 传统方法通常会失败
- 深度神经网络能够拟合高度复杂的功能。



Source of image: https://becominghuman.ai/what-is-deep-learning-and-why-you-need-it-9e2fc0f0e61b

学习模式

- 监督学习
- 半监督学习
- 无监督学习
- 强化学习

带标记的数据
直接反馈
预测
应用:分类

半监督学习

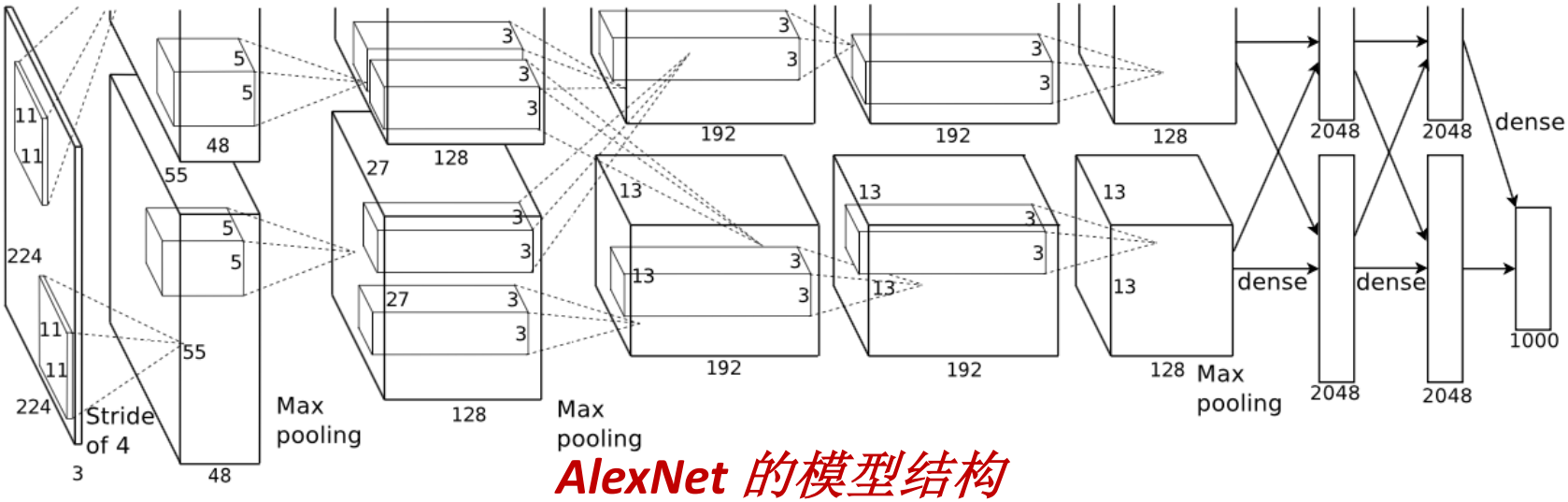
弱监督学习

数据无标记
结果无反馈
找到隐含特征
应用: 重建

决策过程
反馈系统
学习一系列行为
应用: 决策

卷积神经网络

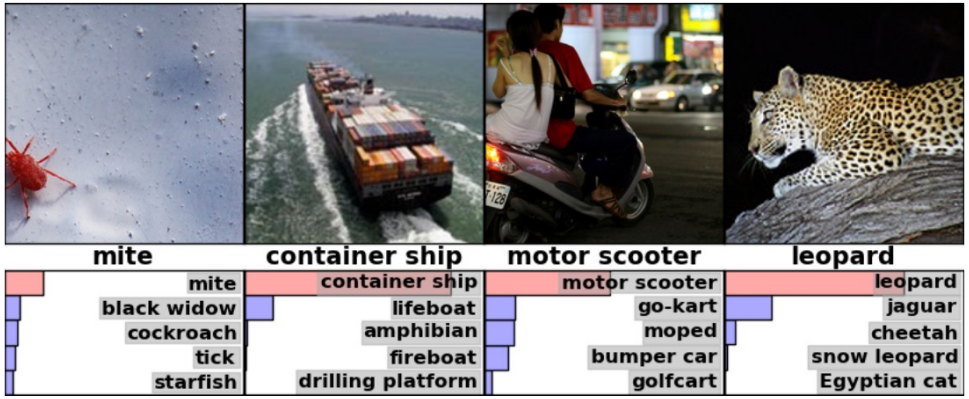
图像分类



***AlexNet* 的模型结构**

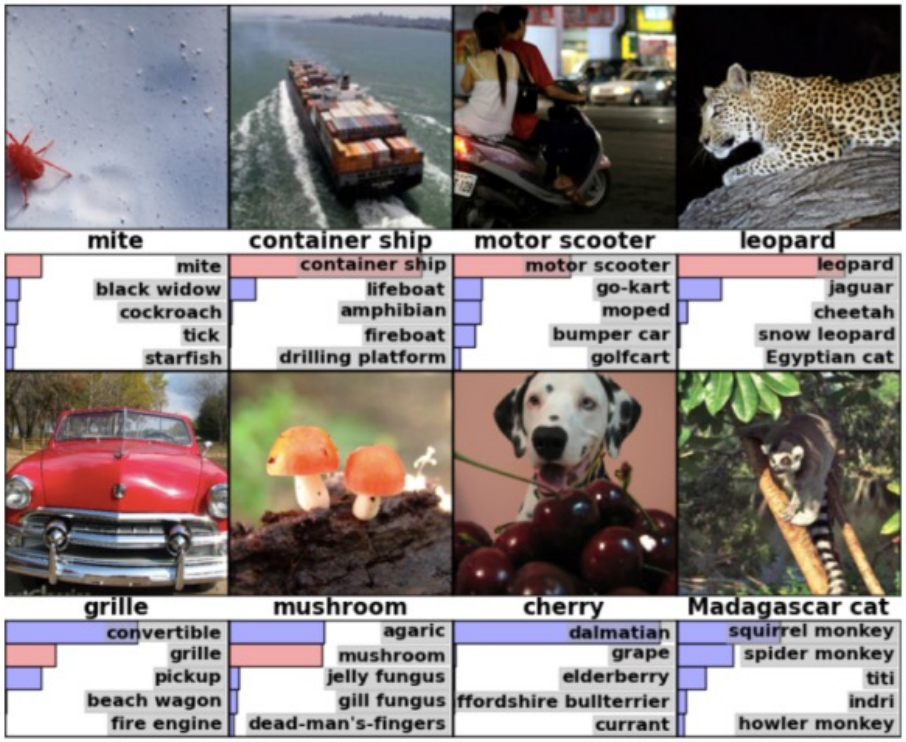
Model	Top-1 Error	Top-5 Error
Sparse coding	47.10%	28.20%
SIFT + FVs	45.70%	25.70%
AlexNet	37.50%	17.00%

性能对比



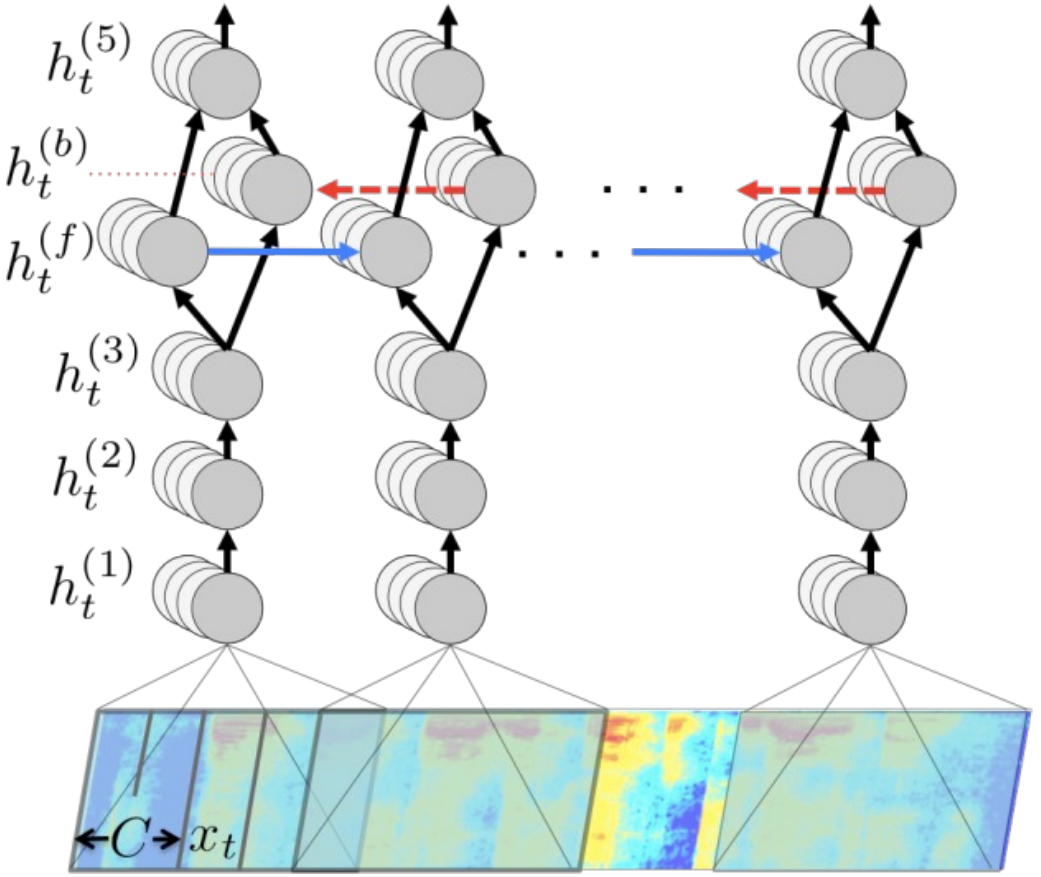
***ImageNet*样本测试结果**

监督学习



图像分类

CNN



语音识别

RNN

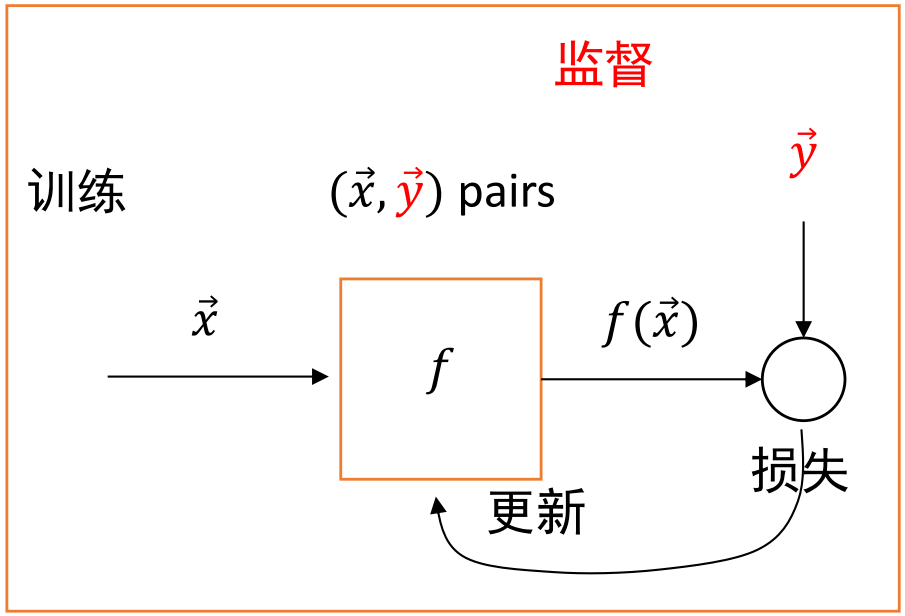
与传统机器学习方法相比

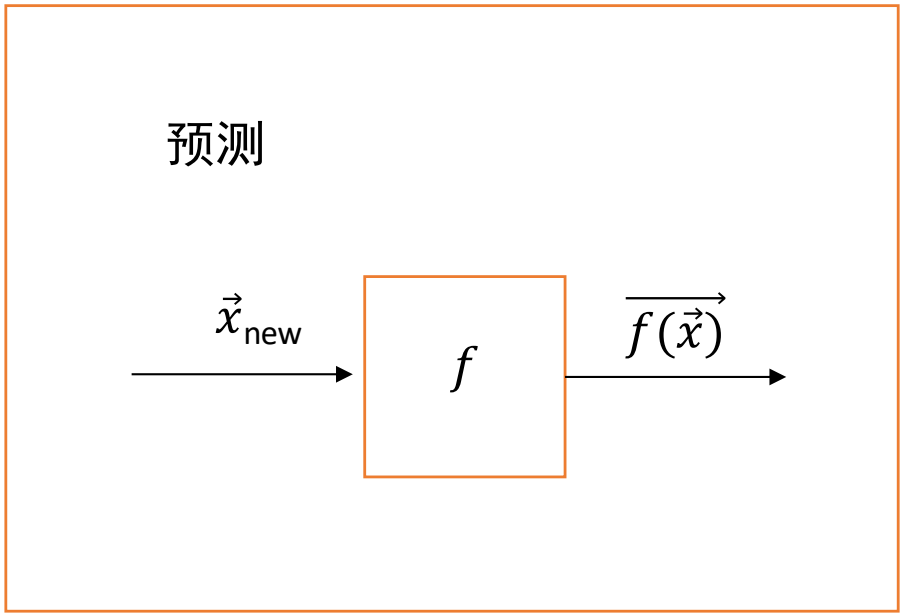
- 卷积神经网络
  - 在图像识别任务中
  - 传统机器学习方法，SIFT的最好结果：26.2%错误率.
  - ResNet-152： 3.57%错误率.
  - 人类： 5%错误率.

- 结构化学习
  - 传统机器学习方法无法解决大部分的问题.

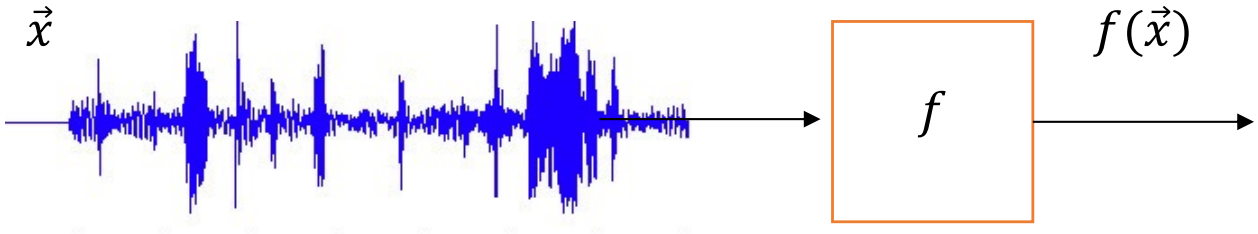
结构化学习

- 与回归和分类相似.
- 不同在于标签和输出都是结构化数据。
  - 比如: 图片, 声音, 文字…



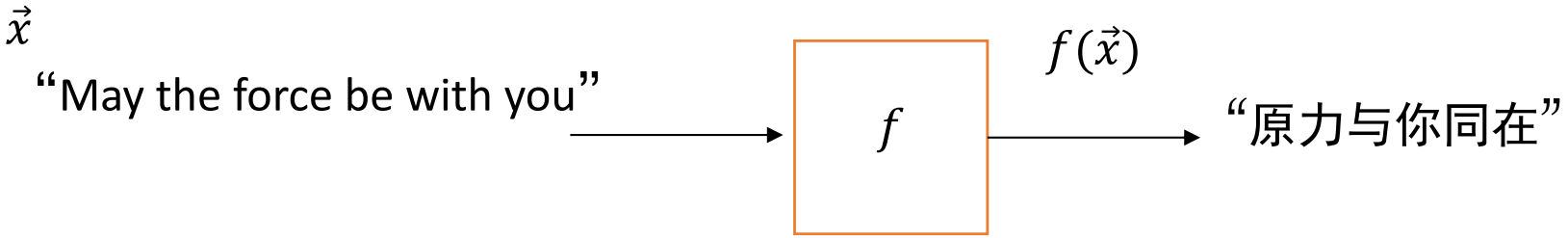


结构化学习



语音识别

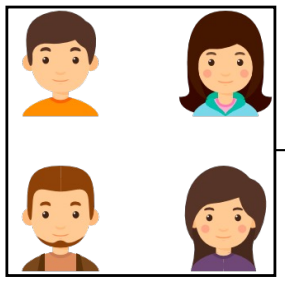
结构化数据



机器翻译

Source of image: http://blog.sina.com.cn/s/blog_4e0987310102v4bn.html

结构化学习



$\vec{x}$

f

$f(\vec{x})$

Jim

Rachel

Amy

David

面部识别

结构化数据

$\vec{x}$

f

$f(\vec{x})$

狗

分类

$\vec{x}$

f

$f(\vec{x})$

嘴里叼着飞盘的狗在草地上跑

图片标题

Source of image: https://wang970727.pixnet.net/blog/post/29313243-%5B%E7%8E%A9%5D-%E9%A3%9B%E7%9B%A4

学习模式

- 监督学习
- 半监督学习
- 无监督学习
- 强化学习

带标记的数据
直接反馈
预测
应用:分类

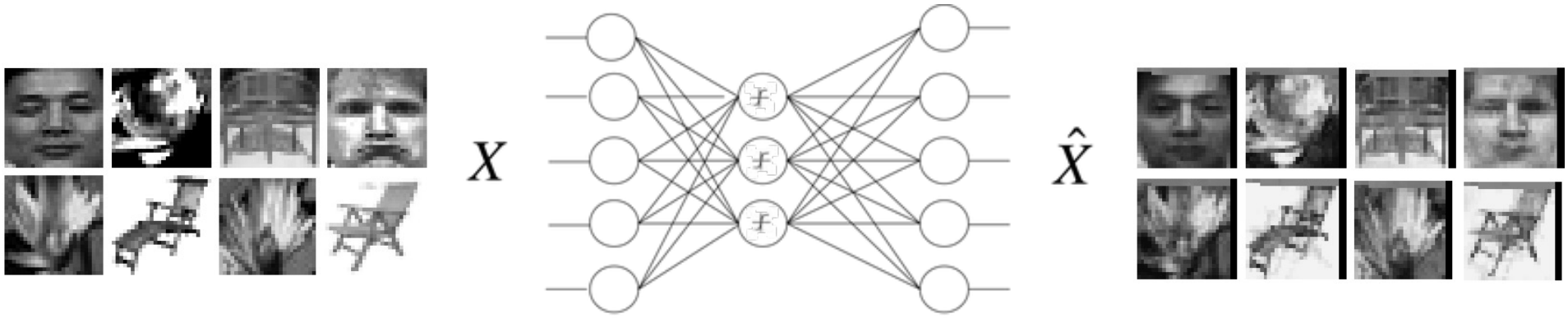
半监督学习

弱监督学习

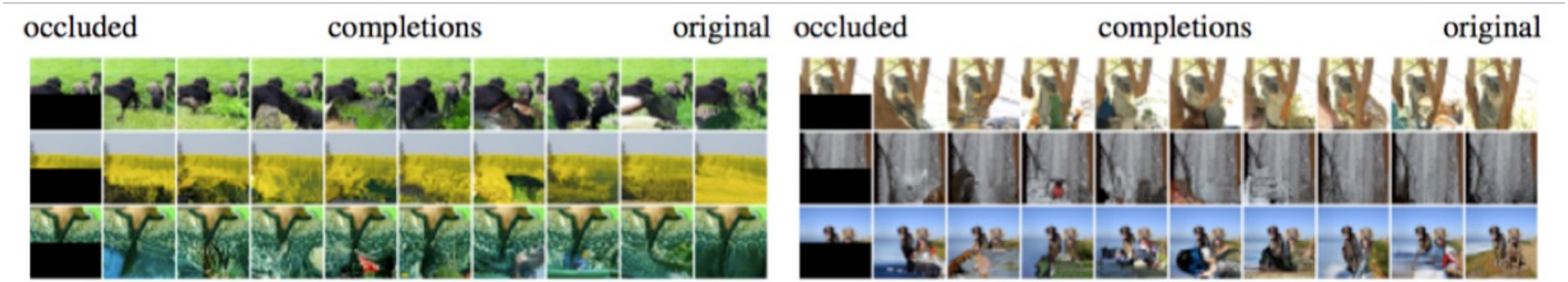
数据无标记
结果无反馈
找到隐含特征
应用: 重建

决策过程
反馈系统
学习一系列行为
应用: 决策

非监督学习



生成模型



Pixel RNN

A.Radford, M. Metz and S. Chintala:Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ICLR*, 2016
A. Oord, N. Kalchbrenner, K. Kavukcuoglu: Pixel Recurrent Neural Networks. arXiv preprint arXiv:1601.06759.

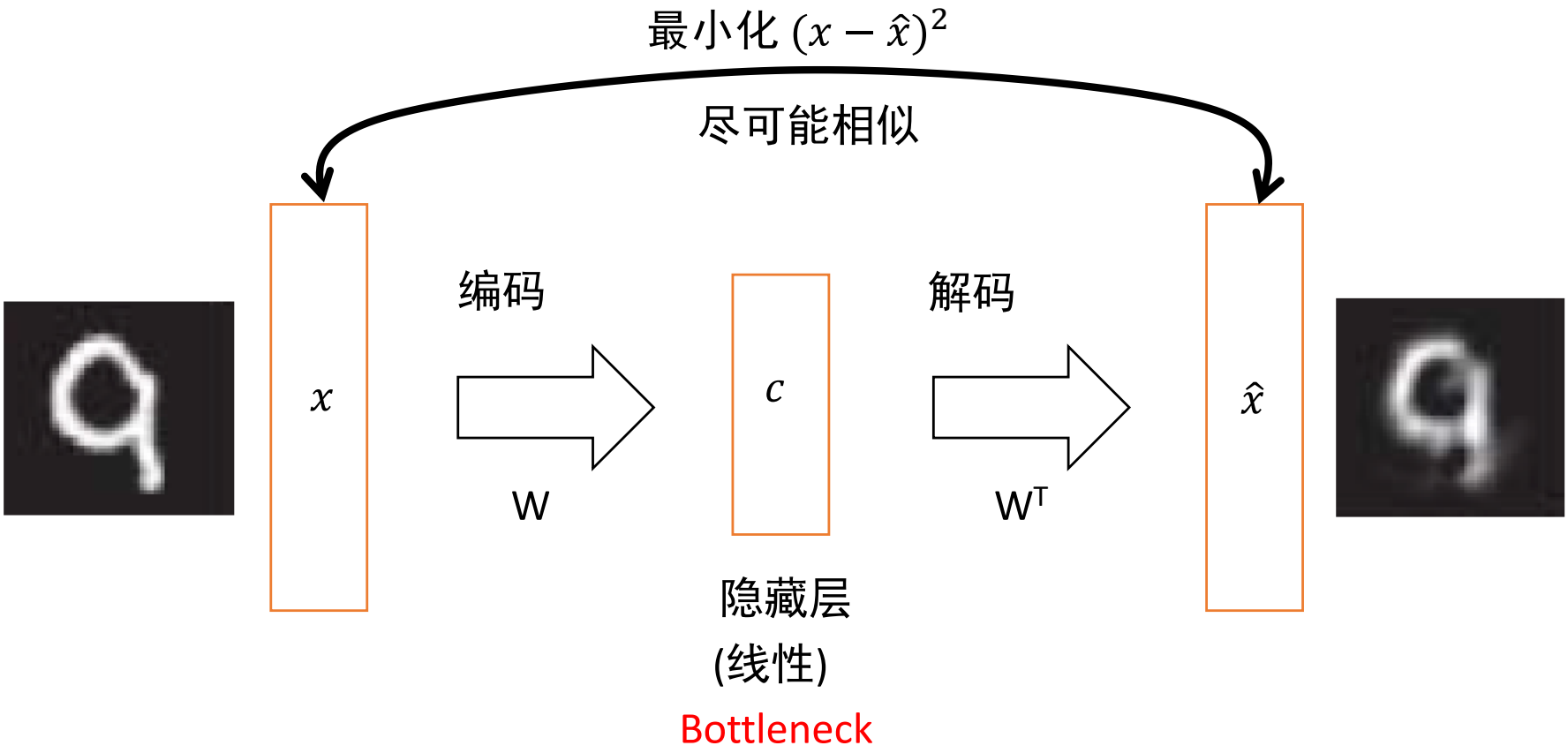
非监督学习的深度学习模型

- 生成式模型: 通过创造进行学习
  - 自动编码机 (Autoencoder, AE)
  - 可变自动编码器(Variational AE, VAE)
  - 生成对抗网络(GAN)
- 自回归：通过预测进行创造
  - OpenAI's GPT-2, 非常巨大的语言模型
  - GPT-3

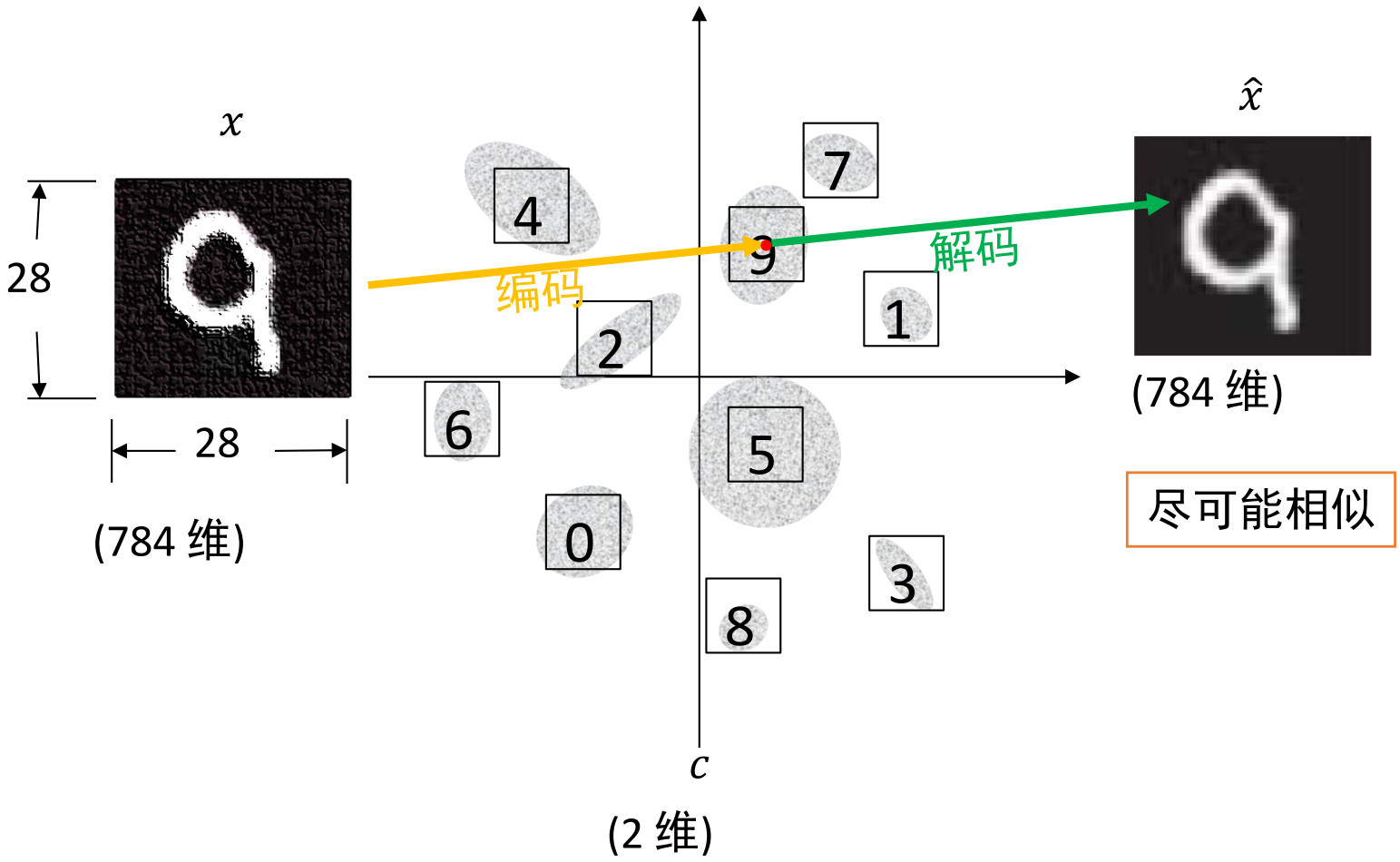
> *what I cannot create, I do not understand.*
>
> --- *Richard Feynman*, 理查德·费曼

自动编码器

- 自动编码器
  - 深度神经网络来编码和解码

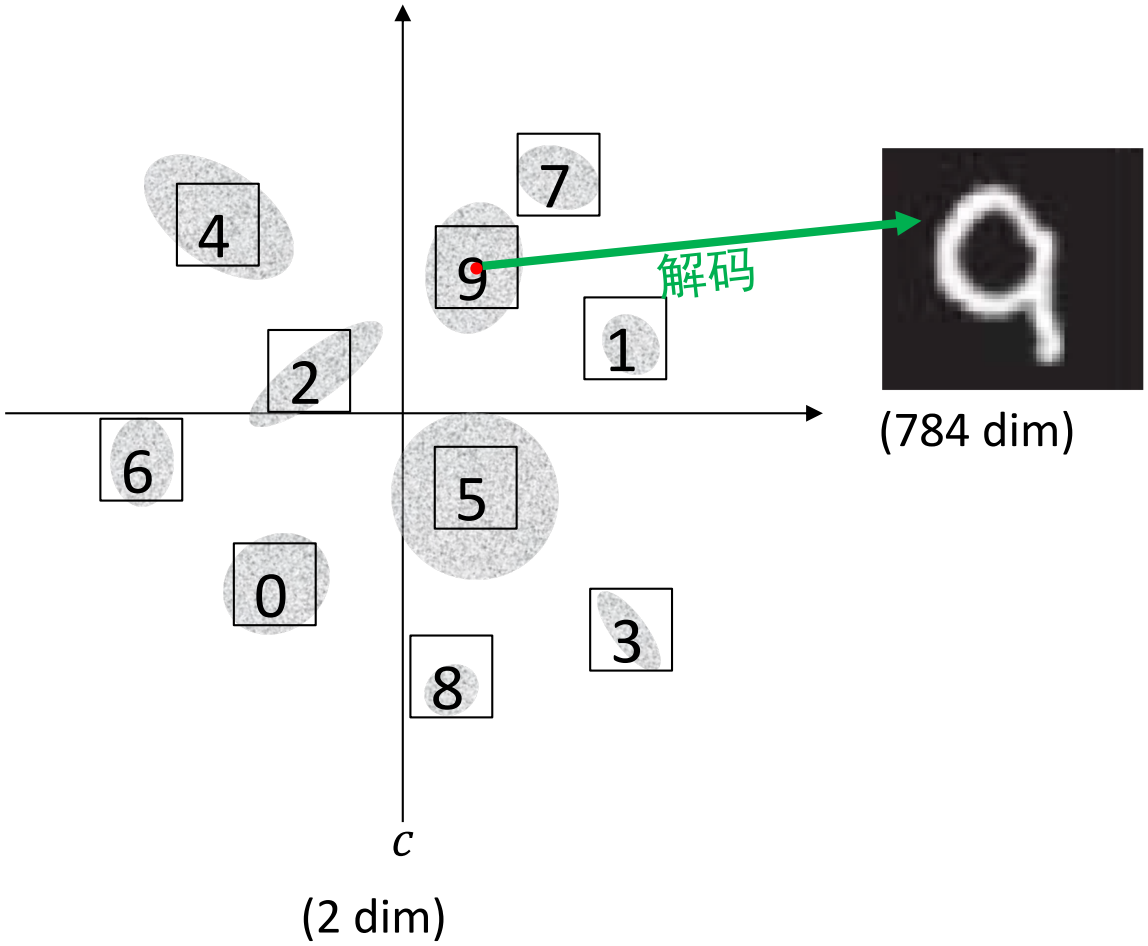


自动编码器



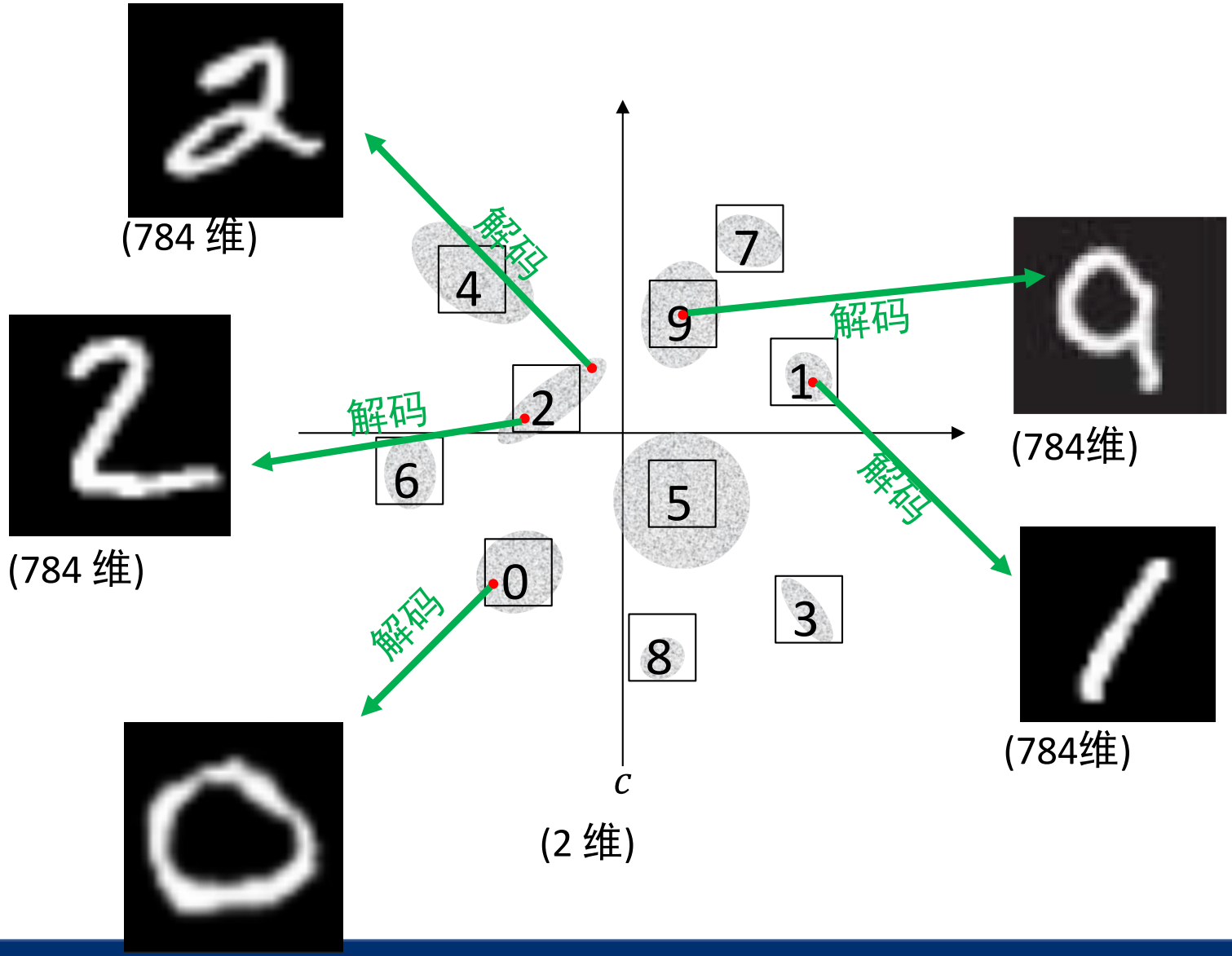
Source of images: https://zhuanlan.zhihu.com/p/25939348

自动编码器



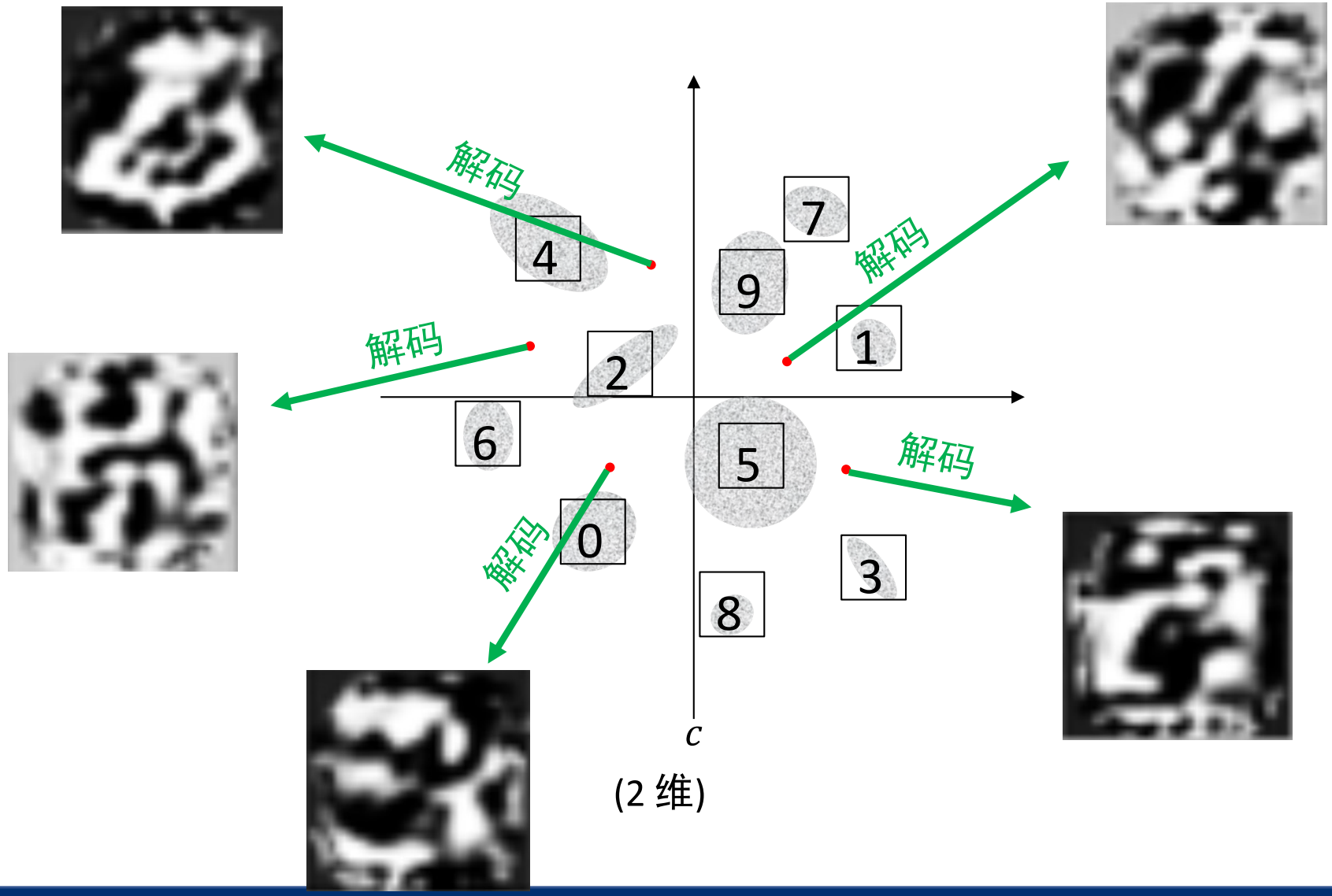
Source of images: https://zhuanlan.zhihu.com/p/25939348

自动编码器能做



Source of images: https://zhuanlan.zhihu.com/p/25939348

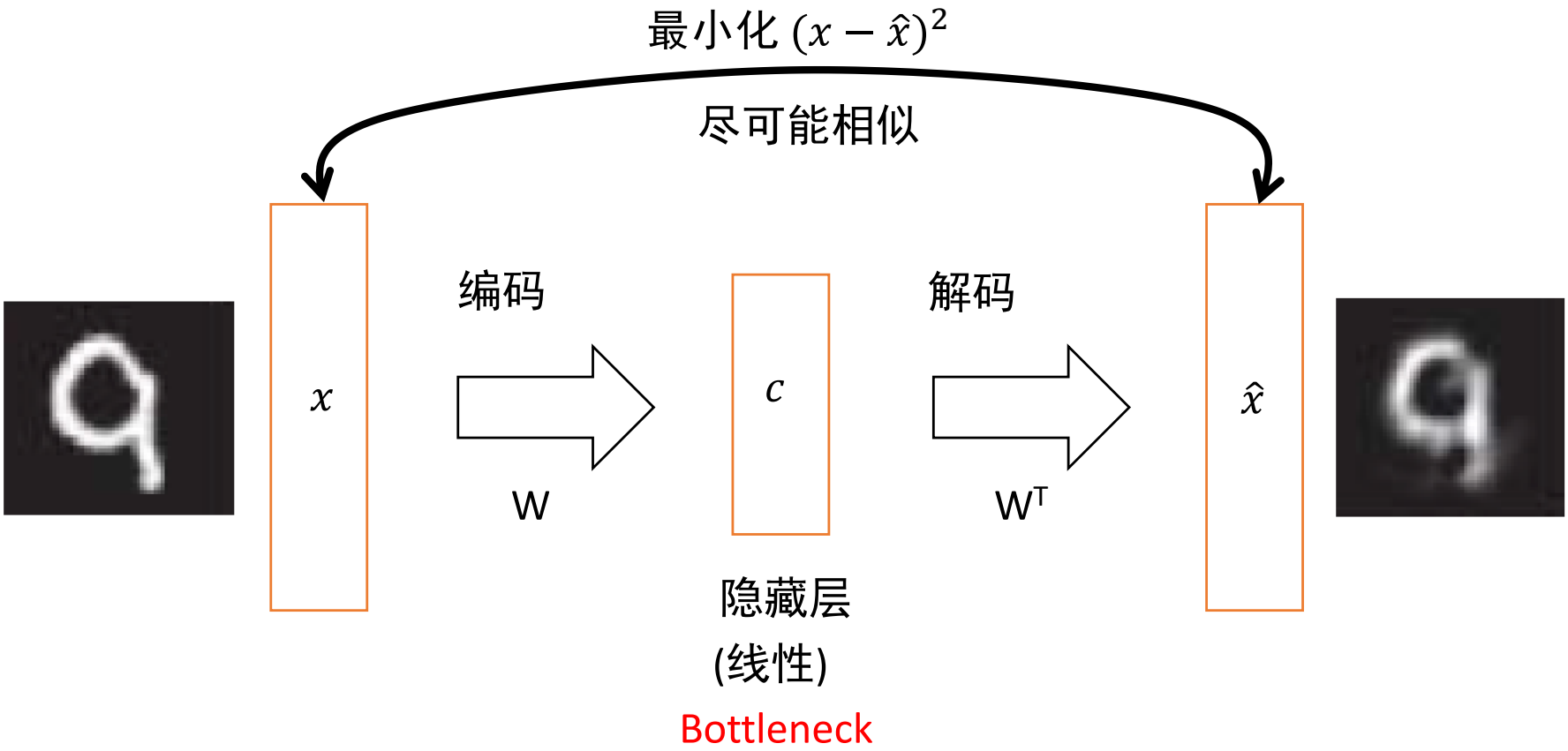
自动编码器不能做



Source of images: https://zhuanlan.zhihu.com/p/25939348

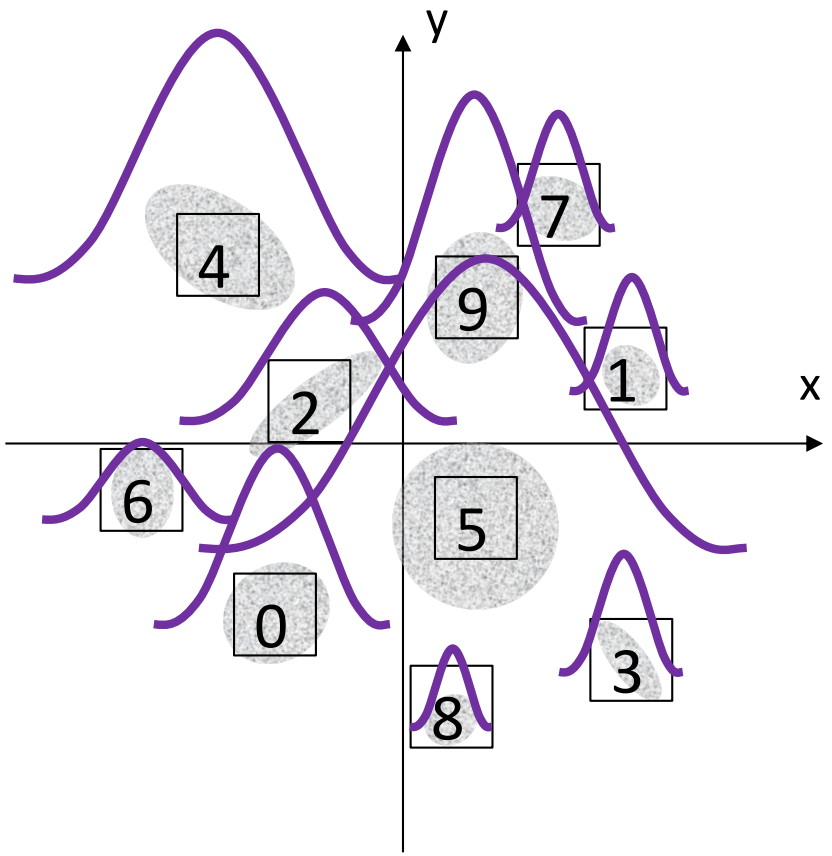
自动编码器不能用来创造

- 我们仅要求AE使原始数据更接近重建数据。
- 我们从不告诉算法在两个类别中间的数据该怎么办。



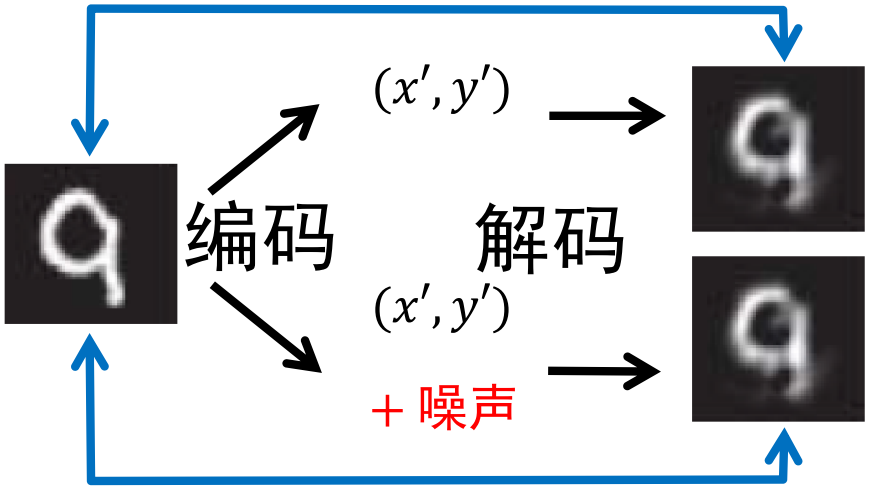
那怎么改进？

—可变自动编码器



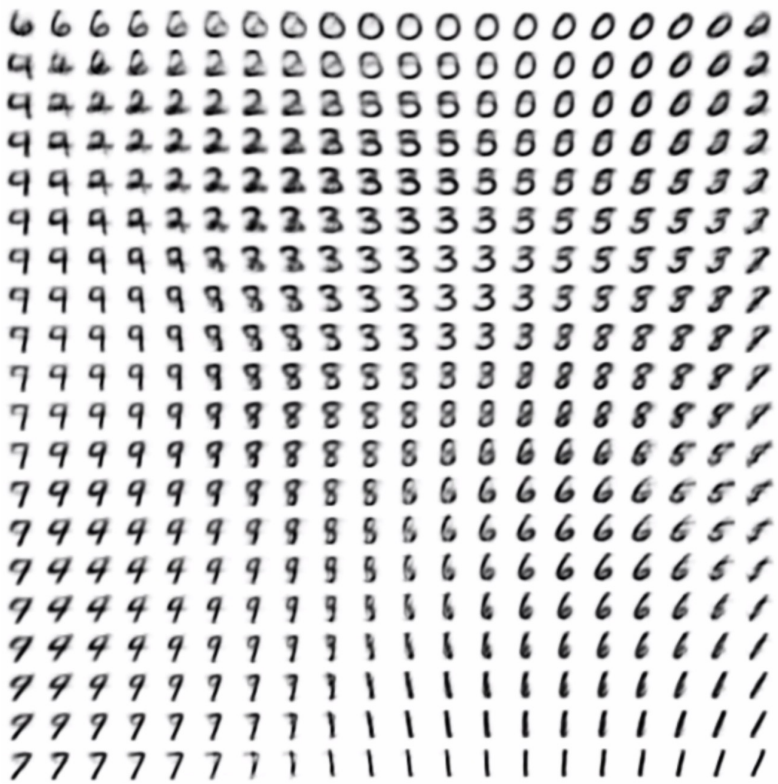
(高斯分布是一个很好的噪声选择.)

- 我们将每个原始数据点添加一些“噪声”到其附近的其他空间，并希望用概率填充整个空间
- 不仅使这两个保持接近



但仍使这两个保持接近

VAE的结果



现在：

两个数字之间的数据点看起来像数字了。

面部表情看起来不错， 但是还不够现真实。

自回归，通过预测进行创造。

- 数据分为一系列小片段，每个小片段依次预测。
- 通过连续猜测接下来会发生什么，将猜测作为输入并再次进行猜测，可以使用此类模型来生成数据。
- OpenAI's GPT-2
  - 给它一个人为的书面开始，它将一直持续下去。 人们可以用它来写新颖的故事，制作假新闻，…

但是这非常危险

程序员拯救乐坛？OpenAI用“逆天”GPT2.0搞了个AI音乐生成器

逆天的语言AI模型来了！编故事以假乱真，问答翻译写摘要都行，横扫各大语言建模任务

逆天的GPT-2居然还能写代码

OpenAI研究实验室已经发布了一套AI系统的完整版本，它用于自动生成文本，但是专家警告称该系统可能被用于恶意目的。该机构最初于今年2月发布了GPT-2程序，但由于担心该程序的完整版本会被用于传播假新闻，垃圾邮件和虚假信息，因此拒绝公布该程序的完整版本。

Artificial Intelligence Generated Content (AIGC)

AI艺术工具Midjourney创作的图像

ChatGPT是继Alpha Go后又一里程碑事件，据市场调查机构Gartner预测，2030年以人机交互为代表的AIGC市场规模将**超过万亿**人民币。

75 你会让哪些人类职业消失？

随着人工智能和机器学习技术的发展，一些重复性和高度规范化的工作，如数据输入，记账和统计，以及一些简单的客户服务工作，可能会被自动化和机器人取代。然而，这不一定意味着这些工作完全消失，也不意味着不需要人类干预或监督。实际上，人工智能可以为人类工作者创造新的机会，例如更高级别的分析和决策工作。另外，技术的发展可能会带来新的工作岗位，并补充那些被自动化取代的工作。

学习模式

- 监督学习
- 半监督学习
- 无监督学习
- 强化学习

带标记的数据
直接反馈
预测
应用:分类

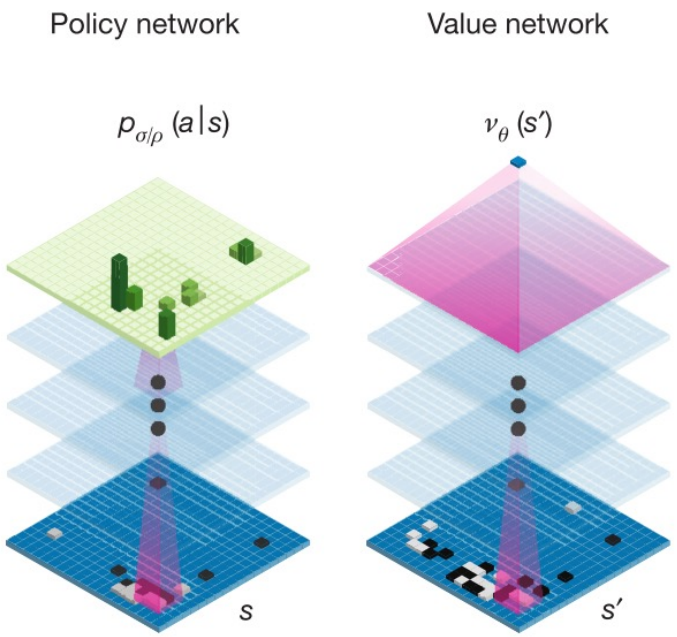
半监督学习

弱监督学习

数据无标记
结果无反馈
找到隐含特征
应用: 重建

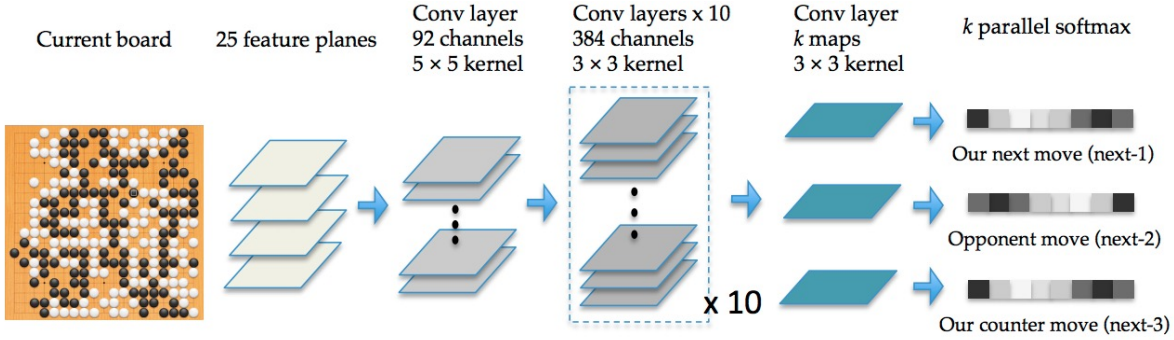
决策过程
反馈系统
学习一系列行为
应用: 决策

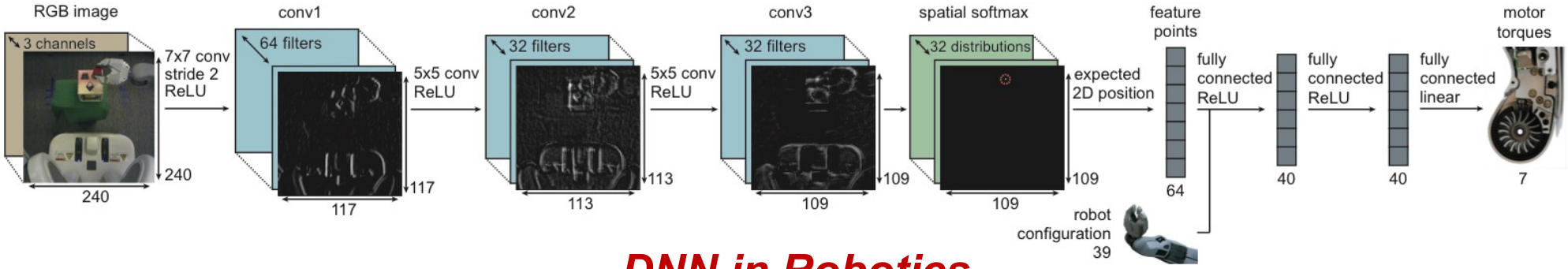
强化学习



Alpha Go

DarkForest

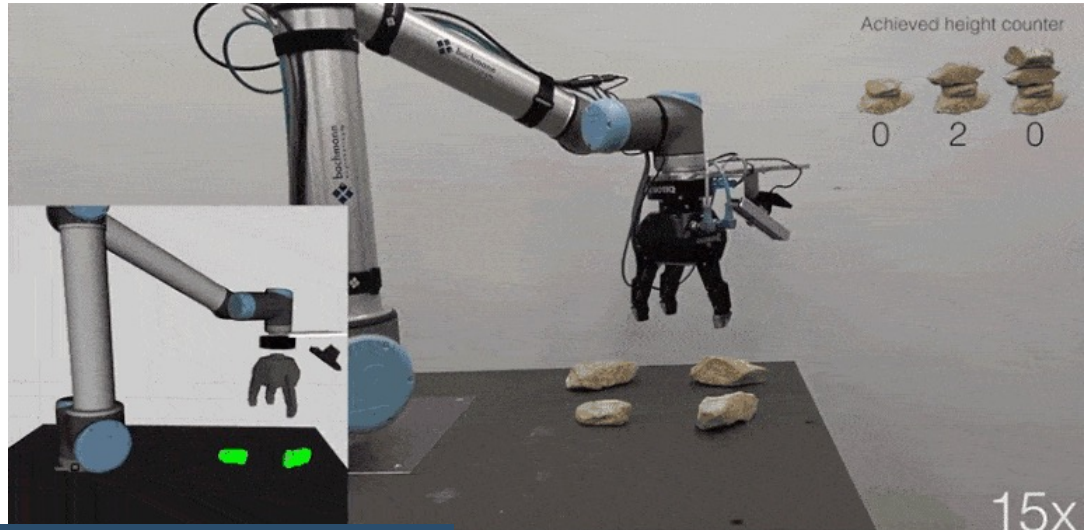




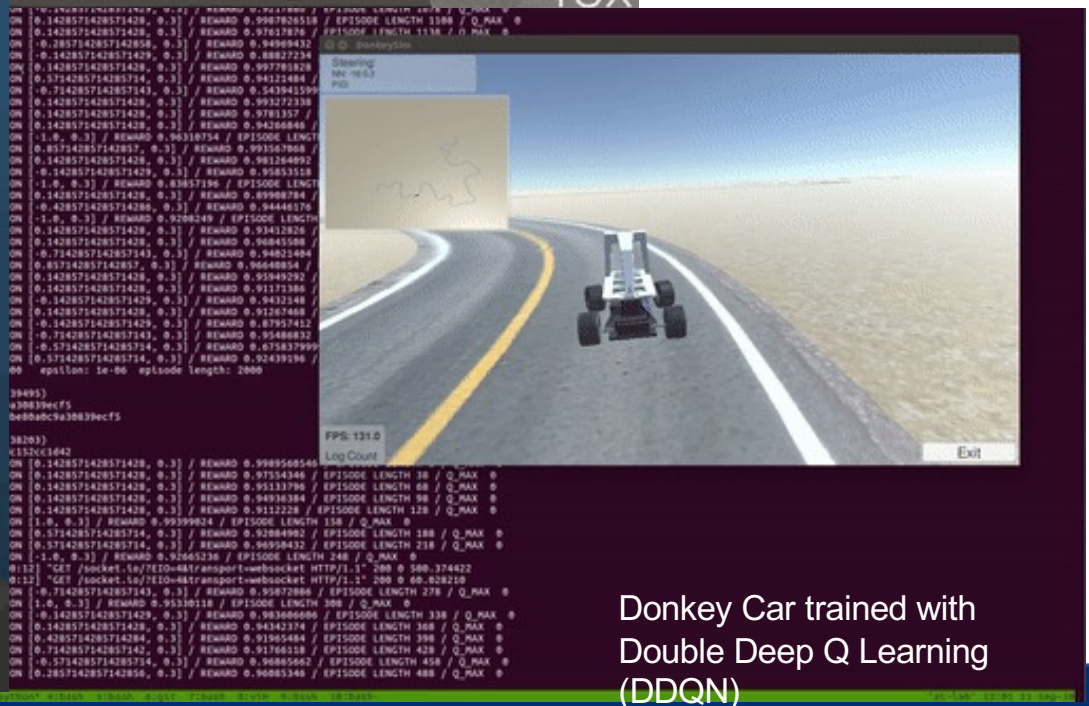
DNN in Robotics

D. Silver et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, 2016.
T. Yuandong, Z. Yan. Better Computer Go Player with Neural Network and Long-term Prediction. arXiv preprint arXiv:1511.06410.
Levine, Sergey, et al. "End-to-end training of deep visuomotor policies." arXiv preprint arXiv:1504.00702.

深度强化学习的一些例子



DeepMind
跑酷小人



Donkey Car trained with
Double Deep Q Learning
(DDQN)

Source of images: https://www.theverge.com/tldr/2017/7/10/15946542/deepmind-parkour-agent-reinforcement-learning; https://flyyufelix.github.io/2018/09/11/donkey-rl-simulation.html; https://www.freecodecamp.org/news/a-brief-introduction-to-reinforcement-learning-7799af5840db/

这节课，我们学习了

- 传统机器学习算法的简单回顾.
  - 监督/无监督/强化学习
- 为什么需要深度学习
  - 现实世界: 数据维度高，映射关系非常复杂.
  - 深度学习有这个能力去拟合.
- 深度学习的例子
  - 监督学习：卷积神经网络、结构化学习
  - 非监督学习：自动编码器、可变自动编码器、GAN、GPT-2
  - 深度强化学习