深度学习

明静思

华东师范大学统计学院

课程内容

1. 机器学习基础
2. 全连接神经网络
3. 卷积神经网络
4. 网络优化与正则化
5. 深度生成模型
6. 循环神经网络

考核方式：

课堂表现（10%）

作业（40%）

期末大作业（50%）

学习资料

- 邱锡鹏. 神经网络与深度学习. 机械工业出版社, 2020. https://nndl.github.io/
- Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola. "*Dive into deep learning*." arXiv preprint arXiv:2106.11342 (2021). https://zh.d2l.ai （中英文版）
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

第一章 机器学习基础

人工智能、机器学习与深度学习

人工智能

- **人工智能（Artificial Intelligence，AI）**就是要让机器的行为看起来就像是人所表现出的智能行为一样。

“计算机控制” + “智能行为”

感知	• 计算机视觉、语音信息处理
学习	• 监督学习、无监督学习、强化学习
认知	• 知识表示、自然语言理解、推理、决策

机器学习

- 机器学习是人工智能的一个重要分支，并逐渐成为推动人工智能发展的关键因素。
- **机器学习（Machine Learning，ML）**是指从有限的观测数据中学习出具有一般性的规律，并利用这些规律对未知数据进行预测的方法。

任务：判断信用卡用户是否会发生逾期

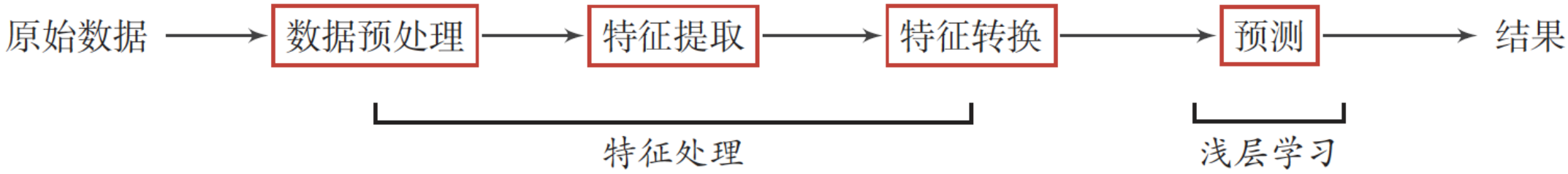
训练数据：信用卡用户的特征（年龄、性别、工作、信用卡额度、存款、是否有住房贷款等）和逾期状态

机器学习算法：学习用户特征与逾期状态之间的模型

预测：使用模型预测新用户未来的逾期情况

传统的机器学习方法

- 在实际任务中使用机器学习模型一般会包含以下几个步骤：



特征提取 → 颜色、纹理、形状…… → 预测 → 狗

语义鸿沟

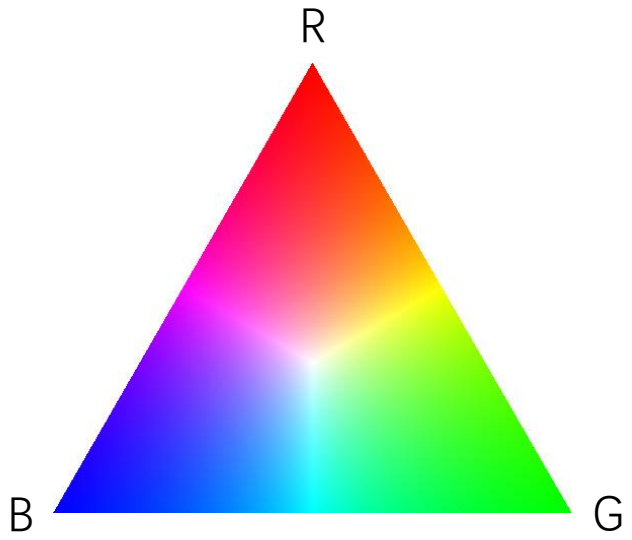
- 基于内容的图像检索系统中广泛存在“语义鸿沟”问题。
- 语义鸿沟问题是指输入数据的底层视觉特征和高层语义信息之间的不一致性和差异性。

表示学习

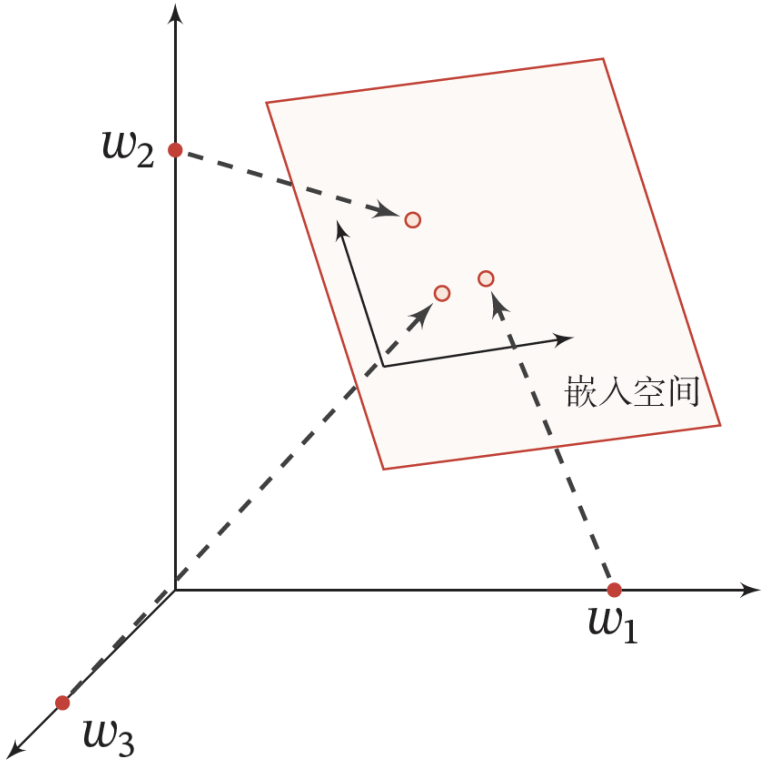
- 为了提高机器学习系统的准确率，需要将输入信息转换为有效的**特征**，或者更一般性地称为**表示**。
- **表示学习**：自动学习出有效的特征，并提高最终机器学习模型性能的算法。
- 一般而言，一个好的表示具有以下几个特点：
  - 应该具有很强的表示能力，即同样大小的向量可以表示更多信息。
  - 应该使后续的学习任务变得简单，即需要包含更高层的语义信息。
  - 应该具有一般性，是任务或领域独立的。

局部表示和分布式表示

颜色	局部表示	分布式表示
琥珀色	$[1, 0, 0, 0]^T$	$[1.00, 0.75, 0.00]^T$
天蓝色	$[0, 1, 0, 0]^T$	$[0.00, 0.5, 1.00]^T$
中国红	$[0, 0, 1, 0]^T$	$[0.67, 0.22, 0.12]^T$
咖啡色	$[0, 0, 0, 1]^T$	$[0.44, 0.31\ 0.22]^T$



嵌入



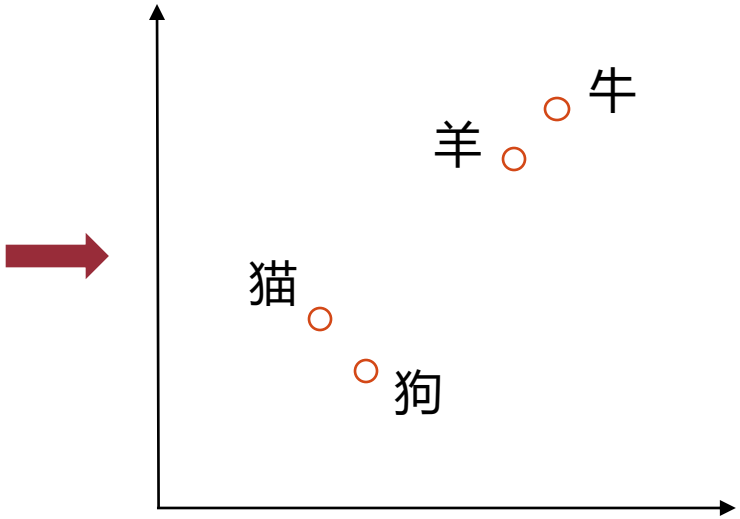
嵌入（Embedding）

猫：$[1, 0, 0, 0]$

狗：$[0, 1, 0, 0]$

牛：$[0, 0, 1, 0]$

羊：$[0, 0, 0, 1]$

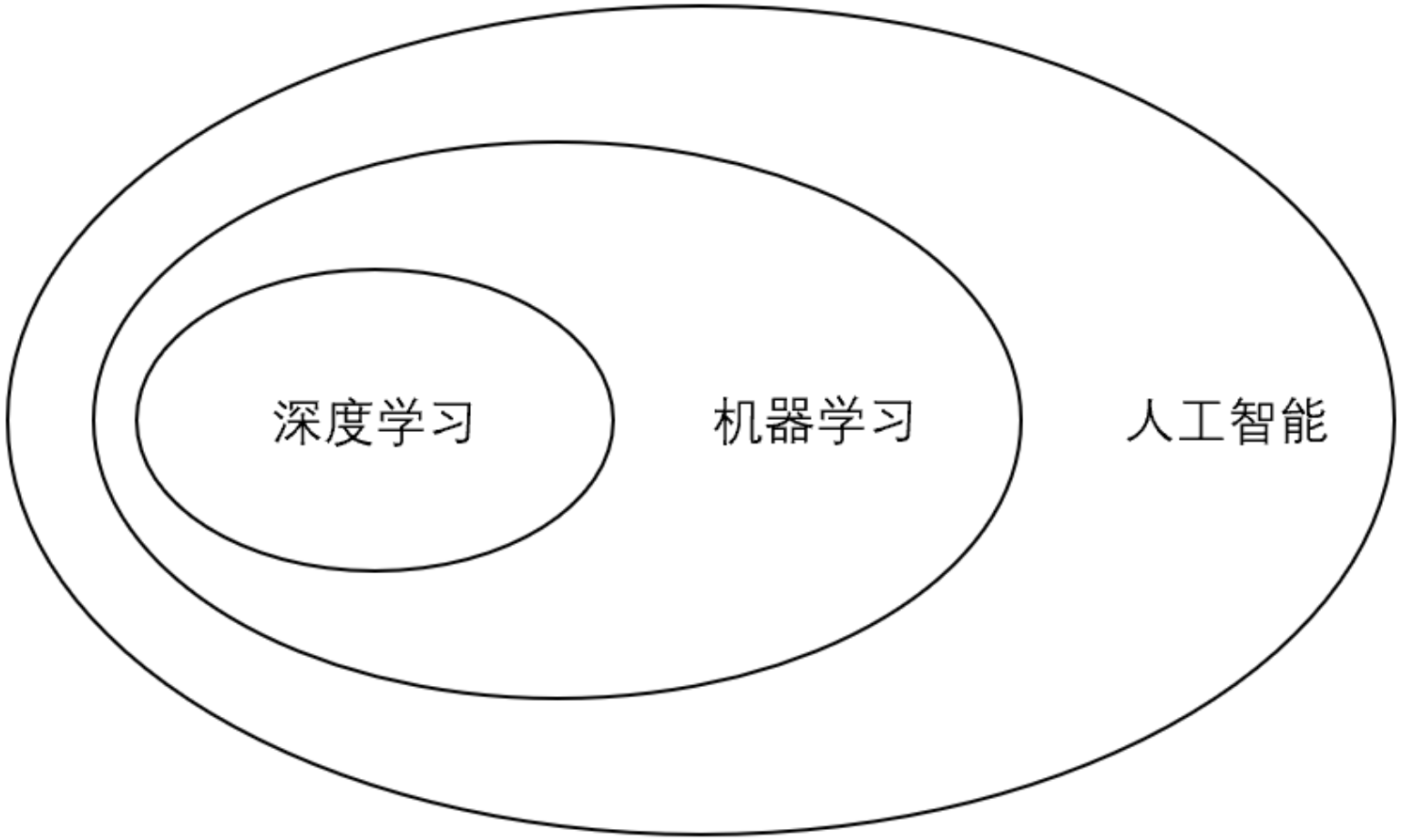


词嵌入

深度学习

- **深度学习（Deep Learning）**是一个复杂的机器学习算法，通过构建具有一定“深度”的模型，让模型来自动学习好的特征表示，从而最终提升预测或识别的准确性。
- 深度学习具备自动提取抽象特征的能力。

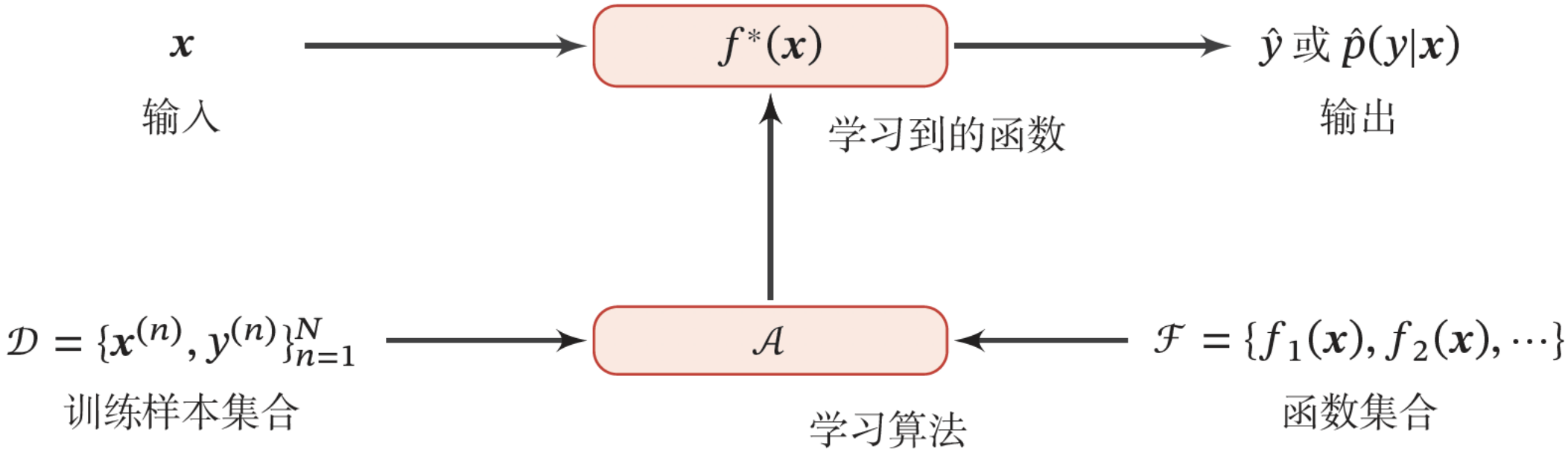
人工智能、机器学习与深度学习的关系



机器学习

模型 + 学习准则 + 优化方法

- 对一个预测任务，输入特征向量为 $\boldsymbol{x}$，输出标签为 y，我们选择一个函数集合 $\mathcal{F}$，通过学习算法 $\mathcal{A}$ 和一组训练样本 $\mathcal{D}$，从 $\mathcal{F}$ 中学习到函数 $f^*(\boldsymbol{x})$，这样对新的输入 $\boldsymbol{x}$，就可以用函数 $f^*(\boldsymbol{x})$ 进行预测。



模型

- 对于样本空间中的样本 $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$，机器学习的目标是找到一个模型来近似 $\boldsymbol{x}$ 和 y 之间的真实映射函数 $g(\boldsymbol{x})$ 或真实条件概率分布 $p_r(y|\boldsymbol{x})$。
- 假设一个函数集合 $\mathcal{F}$，称为假设空间，然后通过观测其在训练集 $\mathcal{D}$ 上的特性，从中选择一个理想的假设 $f^* \in \mathcal{F}$。
- 假设空间 $\mathcal{F}$ 通常为一个参数化的函数族

$$\mathcal{F} = \{f(\boldsymbol{x};\theta) | \theta \in \mathbb{R}^D\},$$

其中 $f(\boldsymbol{x};\theta)$ 是参数为 θ 的函数，也称为模型，D 为参数的数量。

机器学习——模型

- **线性模型**

$$f(\boldsymbol{x};\theta) = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b,$$

其中参数 θ 包含了权重向量 $\boldsymbol{w}$ 和偏置 b。

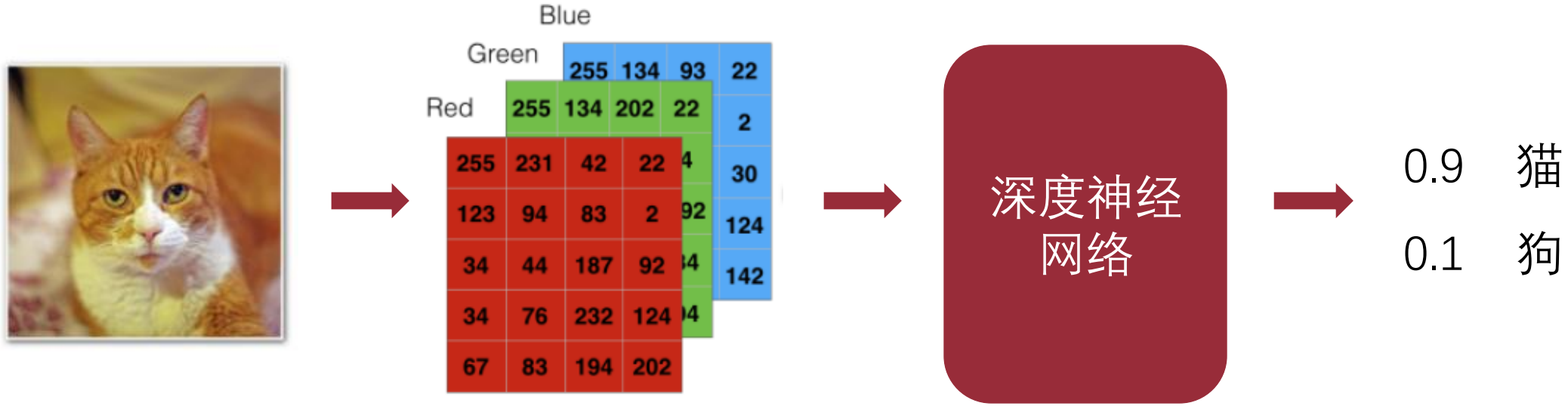
- **非线性模型**

$$f(\boldsymbol{x};\theta) = \boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x}) + b,$$

广义的非线性模型可以写为多个非线性基函数 $\phi(\boldsymbol{x})$ 的线性组合，参数 θ 包含了权重向量 $\boldsymbol{w}$ 和偏置 b。

深度学习——模型

- 大多数的深度学习可以看成一个高度复杂的非线性回归模型。给定一个输入 $\boldsymbol{X}$，就可以给出一个预测值 $\hat{y} = f(\boldsymbol{X}; \theta)$，其中 θ 是待估参数。在实际应用中，构建的神经网络极其复杂，因此 θ 的维度也非常高。



深度学习——模型

- 深度学习采用的模型主要是神经网络模型。
- 广义的非线性模型可以写为多个非线性基函数 $\phi(\boldsymbol{x})$ 的线性组合

$$f(\boldsymbol{x};\theta) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b$$

- 如果 $\phi(\boldsymbol{x})$ 本身为可学习的基函数，比如

$$\phi_k(\boldsymbol{x}) = h\left(\boldsymbol{w}_k^T\phi'(\boldsymbol{x}) + b_k\right), \forall 1 \leq k \leq K$$

其中 $h(\cdot)$ 为非线性函数，$\phi'(\boldsymbol{x})$ 为另一组基函数，$\boldsymbol{w}_k$ 和 b_k 为可学习的参数，则 $f(\boldsymbol{x};\theta)$ 就等价于神经网络模型。

学习准则

- 一个好的模型 $f(\boldsymbol{x}, \theta^*)$ 应该在所有 $(\boldsymbol{x}, y)$ 的可能取值上都与真实映射函数 $y = g(\boldsymbol{x})$ 或与真实条件概率分布 $p_r(y|\boldsymbol{x})$ 一致。
- 模型 $f(\boldsymbol{x}; \theta)$ 的好坏可以通过期望风险 $\mathcal{R}(\theta)$ 来衡量

$$\mathcal{R}(\theta) = \mathbb{E}_{(\boldsymbol{x},y)\sim p_r(y|\boldsymbol{x})}[\mathcal{L}(y, f(\boldsymbol{x}; \theta))]$$

其中 $p_r(\boldsymbol{x}, y)$ 为真实的数据分布，$\mathcal{L}(y, f(\boldsymbol{x}; \theta))$ 为损失函数，用来量化两个变量之间的差异。

损失函数

- **0-1损失函数**

$$\mathcal{L}\big(y, f(\boldsymbol{x};\theta)\big) = \begin{cases} 0 & \text{if } y = f(\boldsymbol{x};\theta) \\ 1 & \text{if } y \neq f(\boldsymbol{x};\theta) \end{cases} = I\big(y \neq f(\boldsymbol{x};\theta)\big)$$

  - 不连续且导数为0，难以优化

- **平方损失函数**

$$\mathcal{L}\big(y, f(\boldsymbol{x};\theta)\big) = \frac{1}{2}\big(y - f(\boldsymbol{x};\theta)\big)^2$$

  - 用在预测标签 y 为实数值的任务中，一般不适用于分类问题。

损失函数

- **交叉熵损失函数**：一般用于分类问题
  - 假设样本的标签 $y \in \{1, \cdots, C\}$ 为离散的类别，模型 $f(\boldsymbol{x}; \theta) \in [0, 1]^C$ 的输出为类别标签的条件概率分布 $p(y = c|\boldsymbol{x}; \theta) = f_c(\boldsymbol{x}; \theta)$。
  - 用一个 C 维的one-hot向量 $\boldsymbol{y}$ 来表示样本标签，看作样本标签的真实条件概率分布 $p_r(\boldsymbol{y}|\boldsymbol{x})$。
  - 用交叉熵衡量两个概率分布的差异

$$\mathcal{L}\big(\boldsymbol{y}, f(\boldsymbol{x}; \theta)\big) = -\boldsymbol{y}^T \log f(\boldsymbol{x}; \theta) = -\sum_{c=1}^{C} y_c \log f_c(\boldsymbol{x}; \theta) = -\log f_y(\boldsymbol{x}; \theta)$$

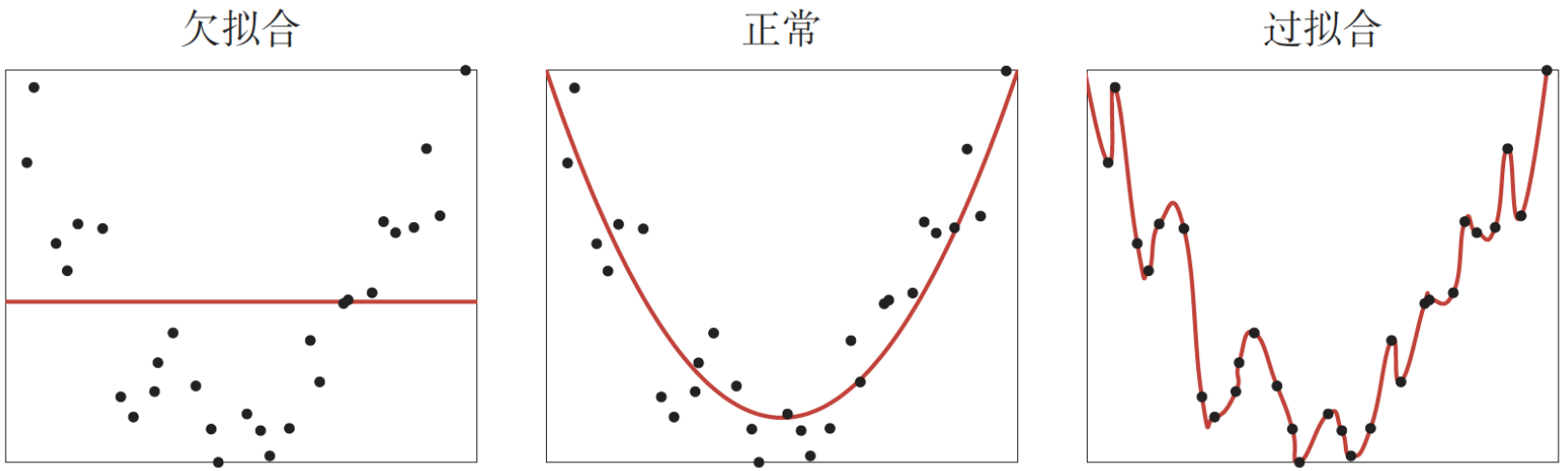
风险最小化准则

- 经验风险：在训练集 $\mathcal{D}$ 上的平均损失

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}\left(y^{(n)}, f\left(\boldsymbol{x}^{(n)};\theta\right)\right)$$

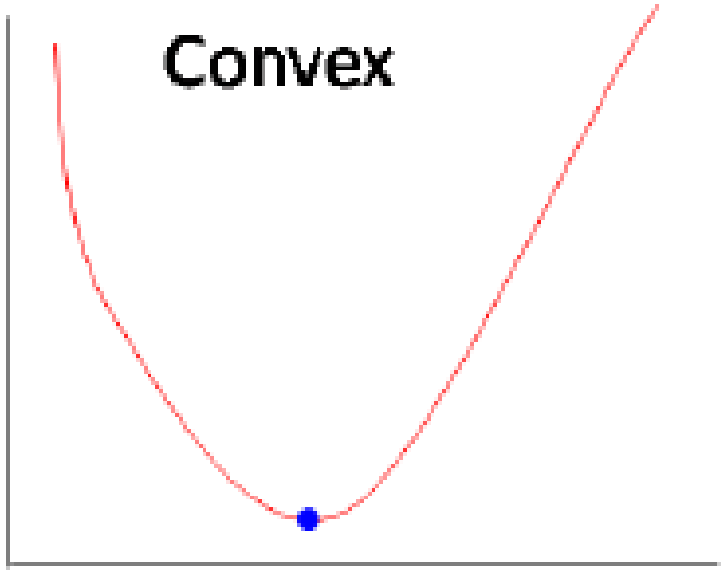
- 经验风险最小化

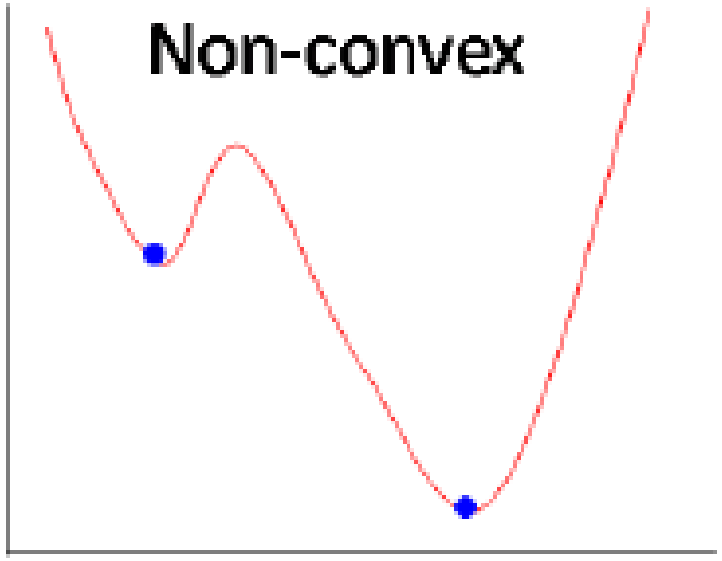
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\,\mathcal{R}_{\mathcal{D}}^{emp}(\theta)$$



优化问题

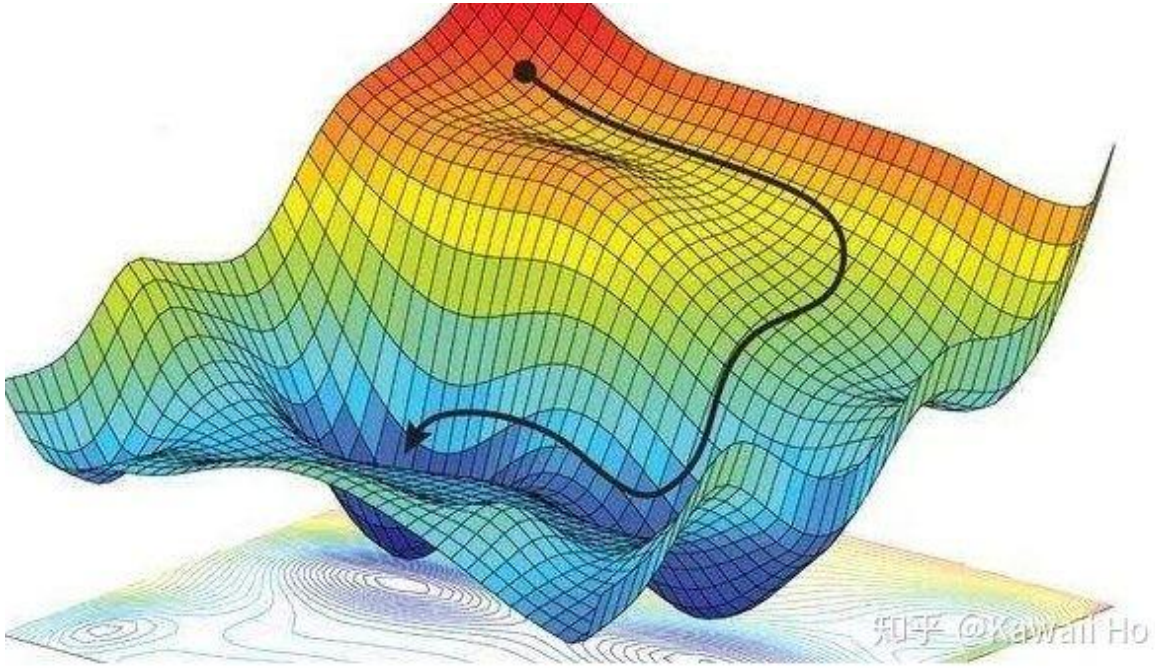
- 机器学习问题转化成为一个最优化问题





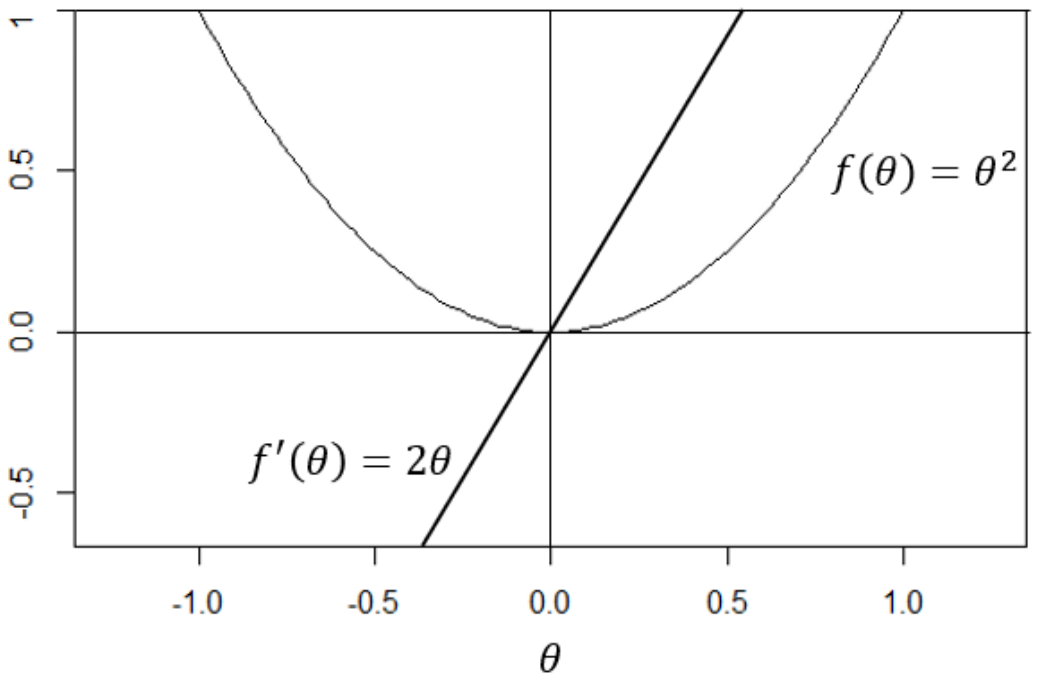
深度学习——优化算法

- 深度学习中，大多流行的优化算法通常基于梯度下降（gradient descent）。
- 梯度下降法：沿函数值下降最快的方向，改变 θ 而获得更小的 $f(\theta)$ 的技术。
- 梯度下降算法的直观理解：梯度下降就是寻找最陡峭的方向，也就是负梯度方向。



梯度下降法的理解

$$f(\theta)=\theta^2$$

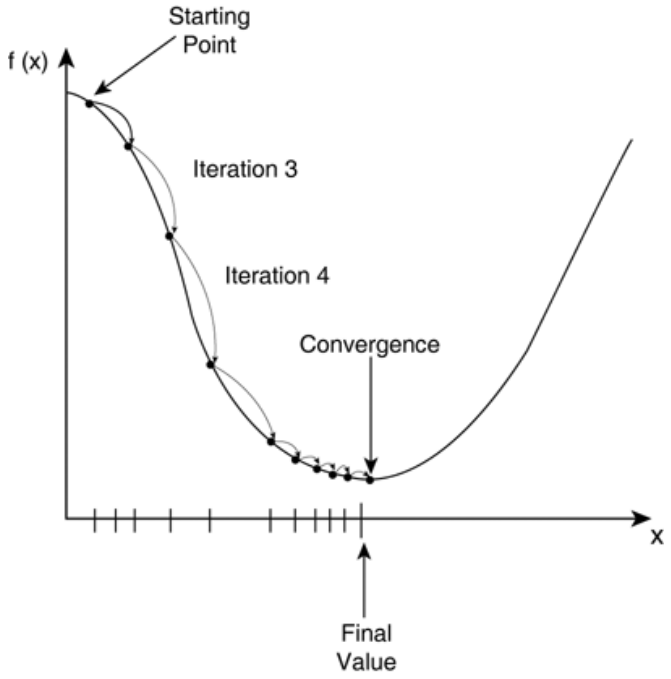


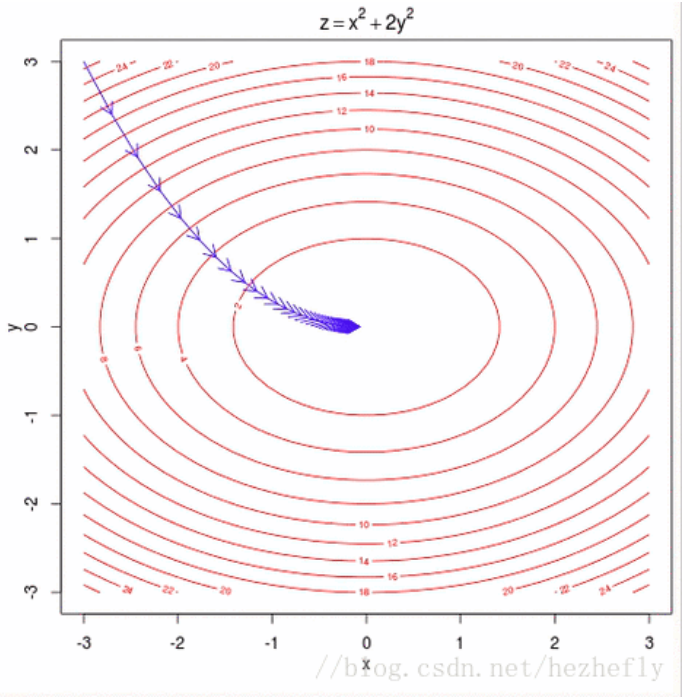
- 对于$\theta>0$，存在$f'(\theta)>0$，因此可以向左移动来减小$f(\theta)$。
- 对于$\theta<0$，存在$f'(\theta)<0$，因此可以向右移动来减小$f(\theta)$ 。
- 对于$\theta=0$，存在$f'(\theta)=0$，出现全局最小点，梯度下降到这里停止。

梯度下降法

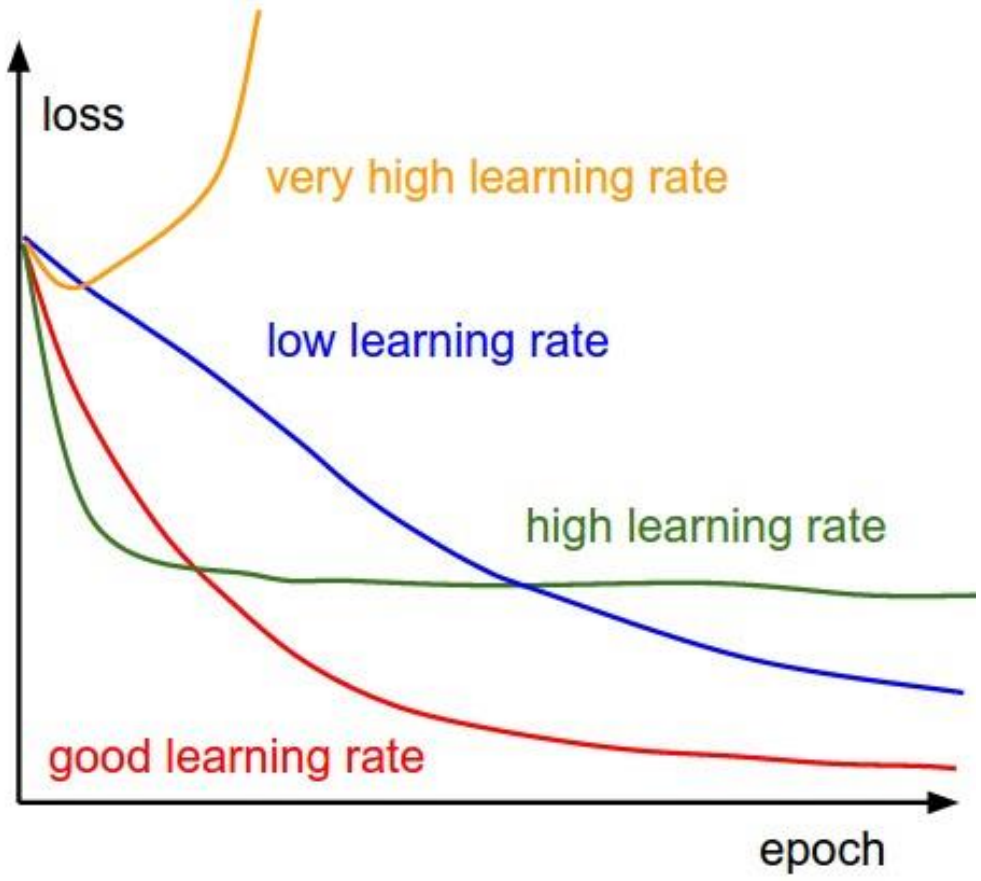
- 利用学习率α（Learning Rate）来定义每次参数更新的幅度。
- 更新参数公式：

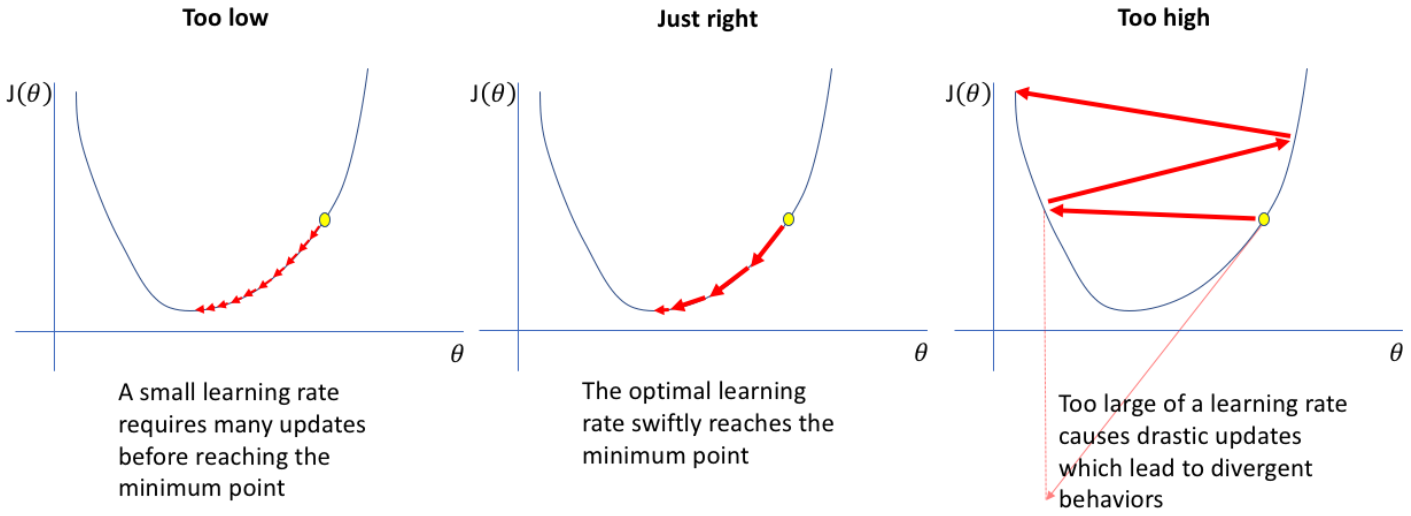
$$\theta_{n+1} = \theta_n - \alpha \frac{\partial f(\theta_n)}{\partial \theta}$$



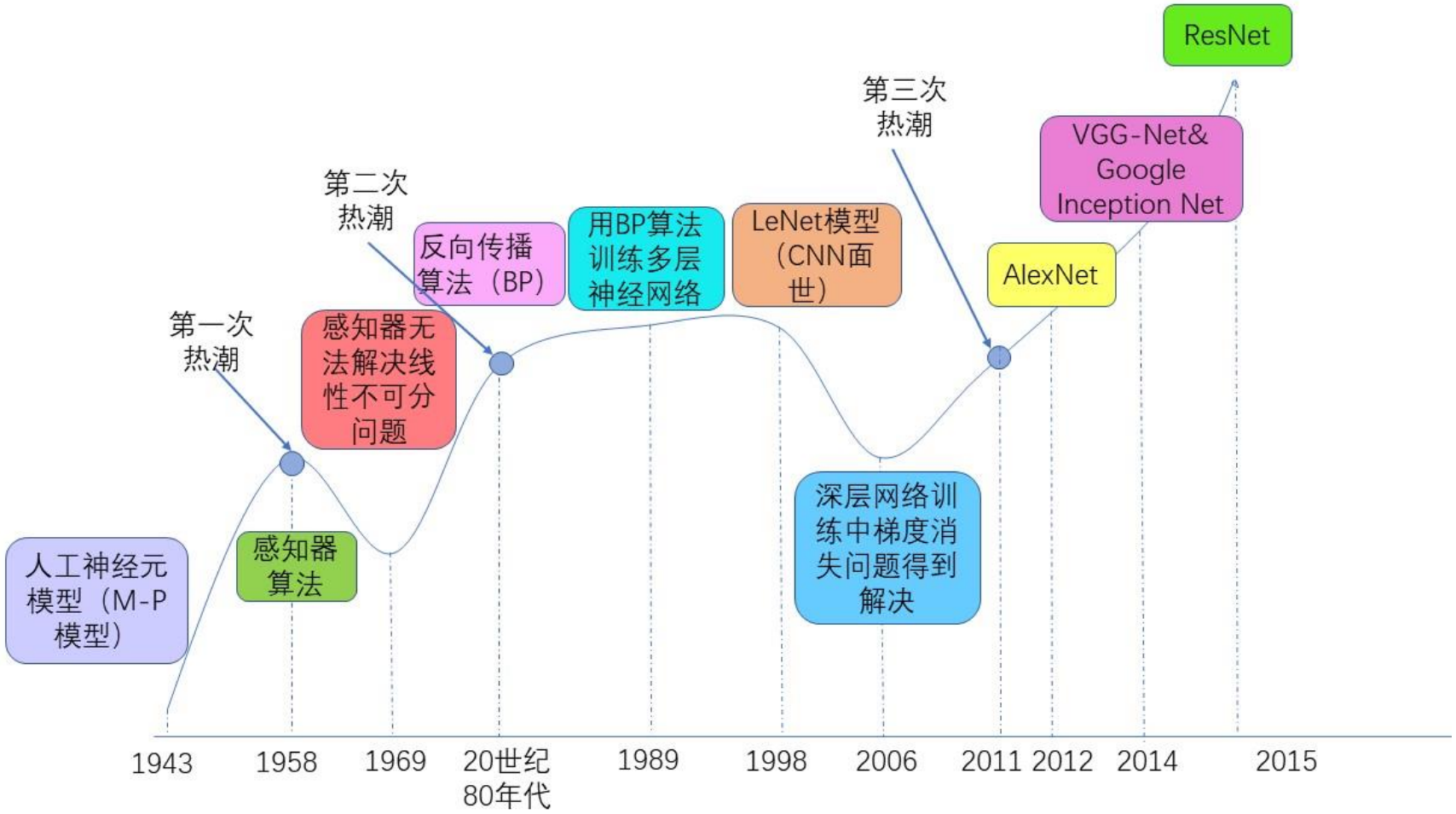


学习率是十分重要的超参数





深度学习的发展历程


ResNet
第三次
热潮
VGG-Net&
Google
Inception Net
第二次
热潮
反向传播
算法（BP）
用BP算法
训练多层
神经网络
LeNet模型
（CNN面
世）
AlexNet
第一次
热潮
感知器无
法解决线
性不可分
问题
深层网络训
练中梯度消
失问题得到
解决
人工神经元
模型（M-P
模型）
感知器
算法
1943
1958
1969
20世纪
80年代
1989
1998
2006
2011
2012
2014
2015

深度学习的发展历程

- 大数据+大规模算力

年代	数据规模	内存	每秒浮点运算
1970	100 （鸢尾花卉）	1 KB	100 KF (Intel 8080)
1980	1 K （波士顿房价）	100 KB	1 MF (Intel 80186)
1990	10 K （光学字符识别）	10 MB	10 MF (Intel 80486)
2000	10 M （网页）	100 MB	1 GF (Intel Core)
2010	10 G （广告）	1 GB	1 TF (Nvidia C2050)
2020	1 T （社交网络）	100 GB	1 PF (Nvidia DGX-2)

深度学习擅长的领域

深度学习擅长的领域

- **图像处理**
  - 图像分类、目标检测、图像分割
- **语音识别**
- **自然语言处理**
  - 机器翻译、文本分类、自动问答
- **视频处理**
- **棋牌竞技**

图像分类

Imagenet: A large-scale hierarchical image database

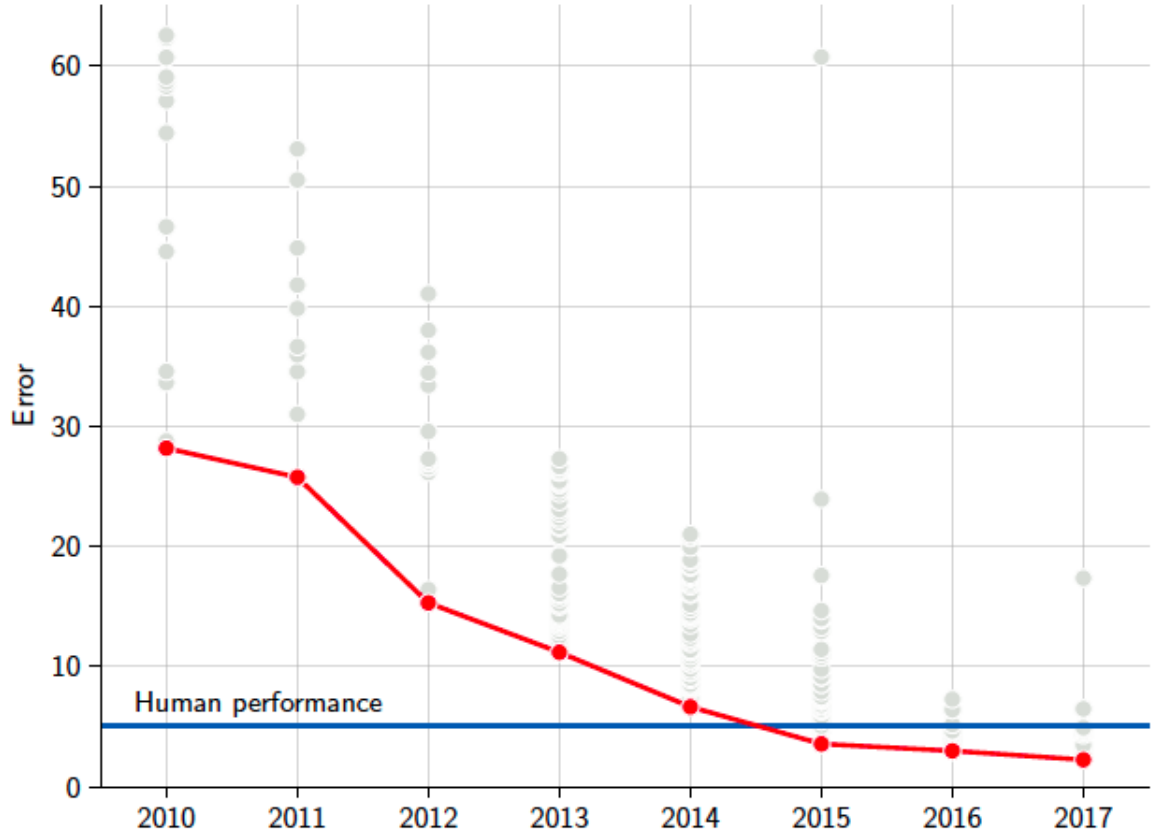
J Deng, W Dong, R Socher, LJ Li, K Li… - 2009 IEEE conference …, 2009 - ieeexplore.ieee.org

The explosion of image data on the Internet has the potential to foster more sophisticated and robust models and algorithms to index, retrieve, organize and interact with images and multimedia data. But exactly how such data can be harnessed and organized remains a …

☆ 99 Cited by 25514 Related articles All 31 versions

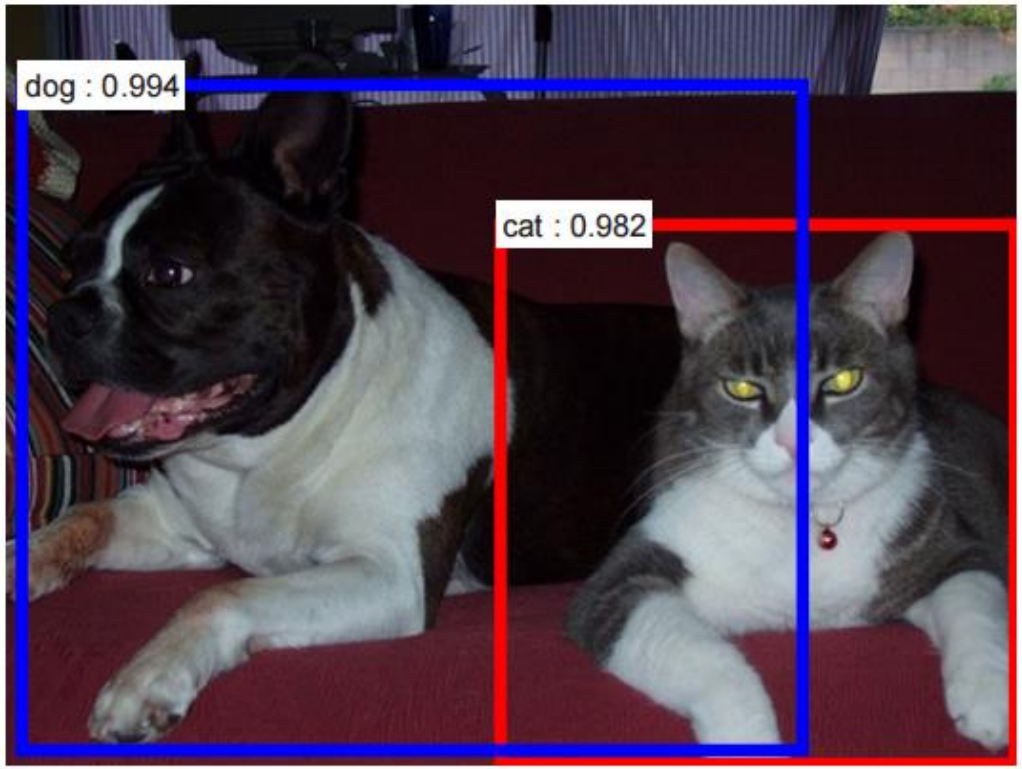
图像分类

- ImageNet 大规模视觉识别挑战（ILSVRC）

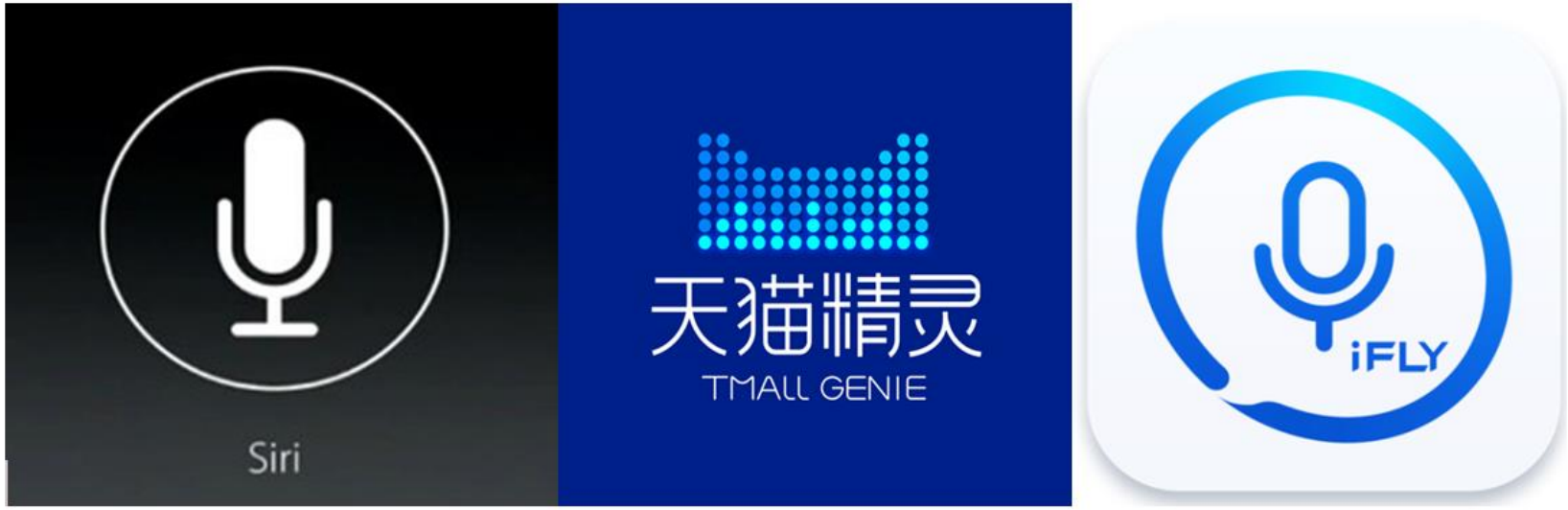


D. Gershgorn. The data that transformed AI research|and possibly the world, July 2017.

目标检测和分割

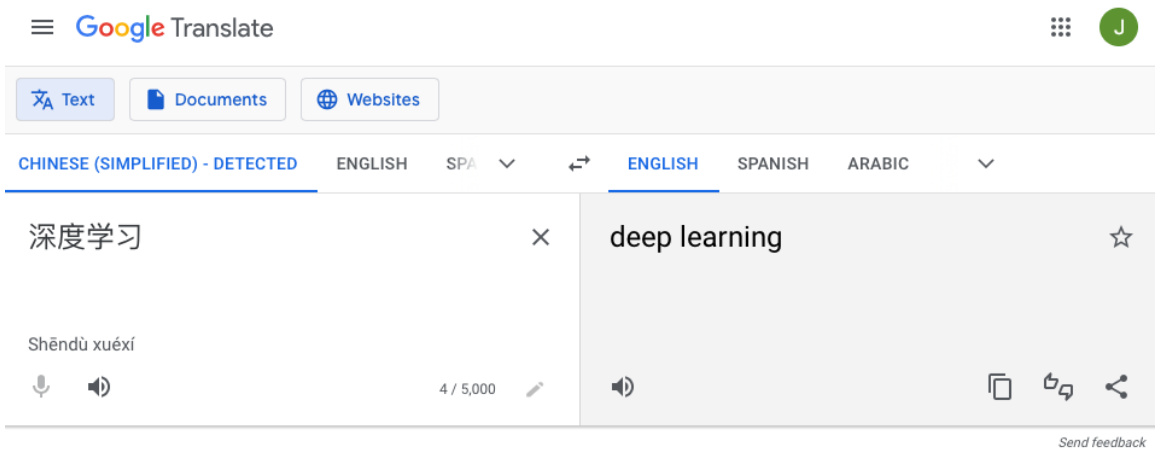


语音识别



自然语言处理

机器翻译

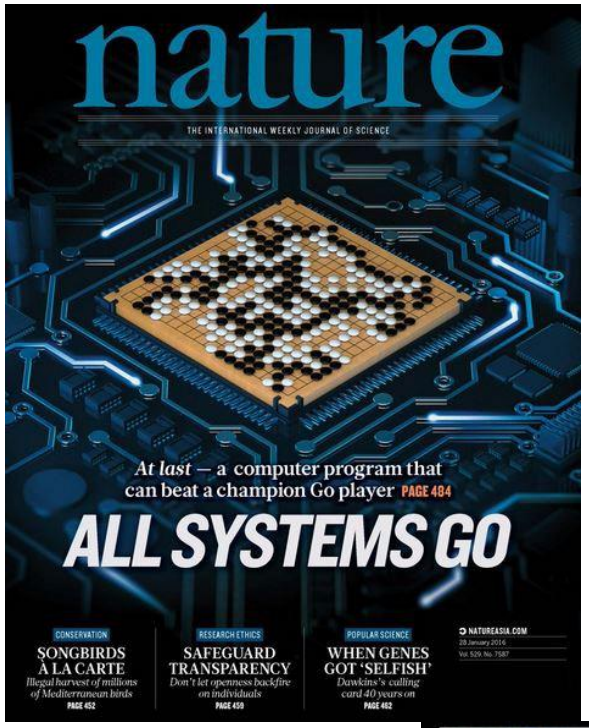


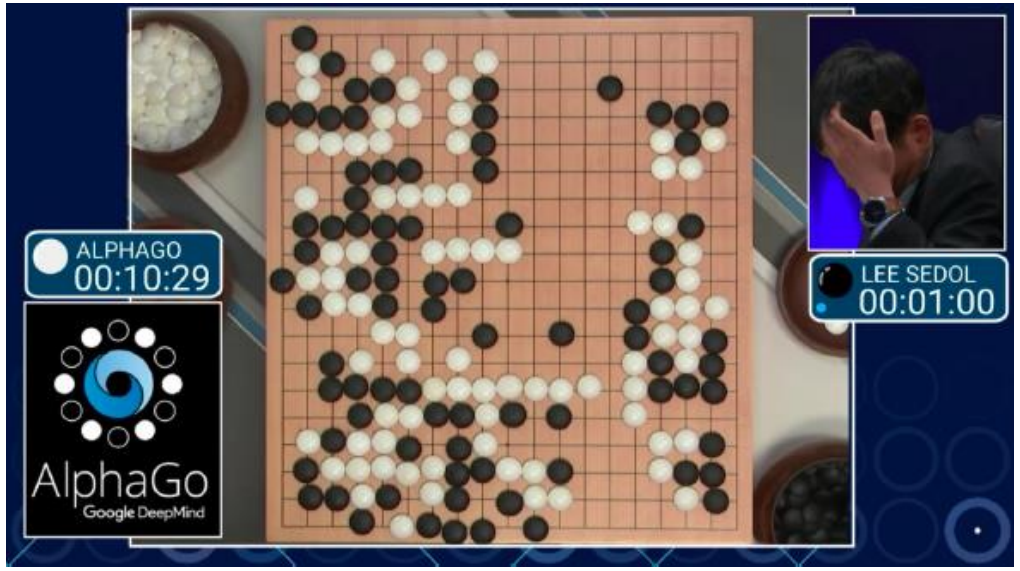
情感分析

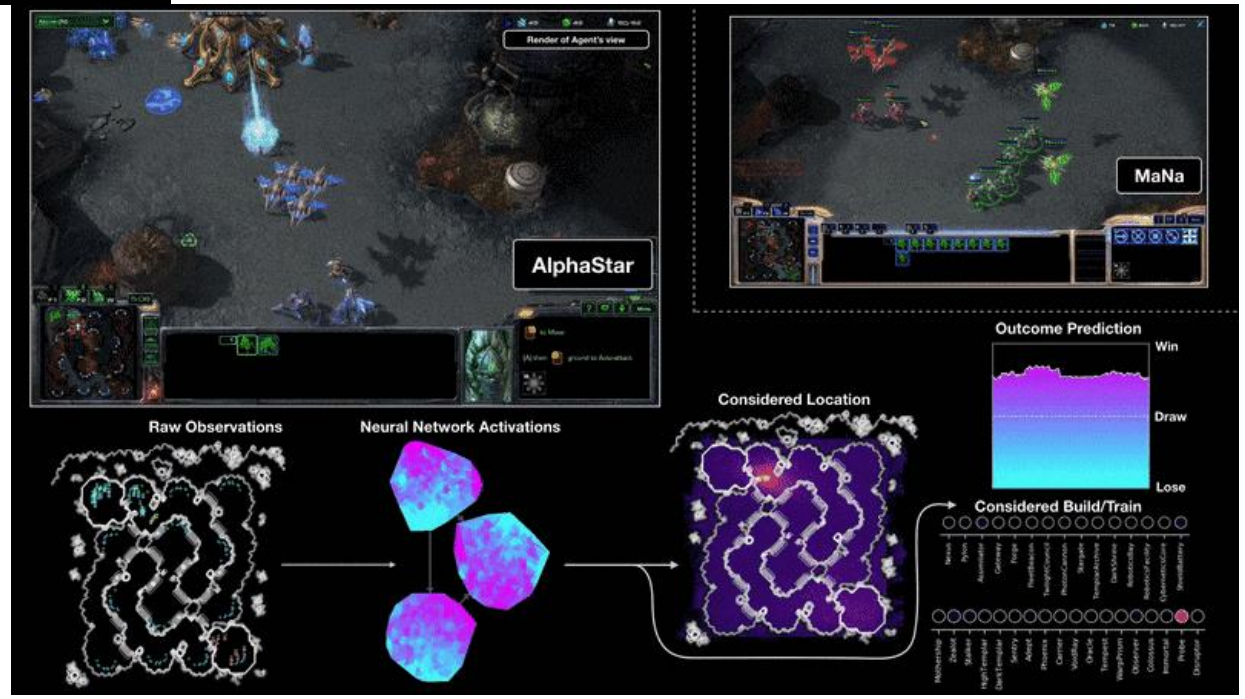
文本分类

棋牌竞技

- AlphaGo
- AlphaGo Zero
- AlphaZero
- MuZero
- AlphaStar







机器学习≈构建一个映射函数

- 图像识别

$f($  $)=$ “9”

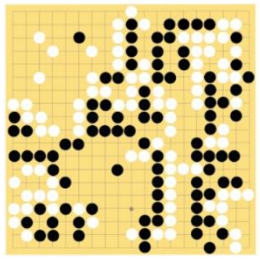
- 语音识别

$f($  $)=$ “你好”

- 机器翻译

$f($ “你好！” $)=$ “Hello!”

- 围棋

$f($  $)=$ “6-5”(落子位置)

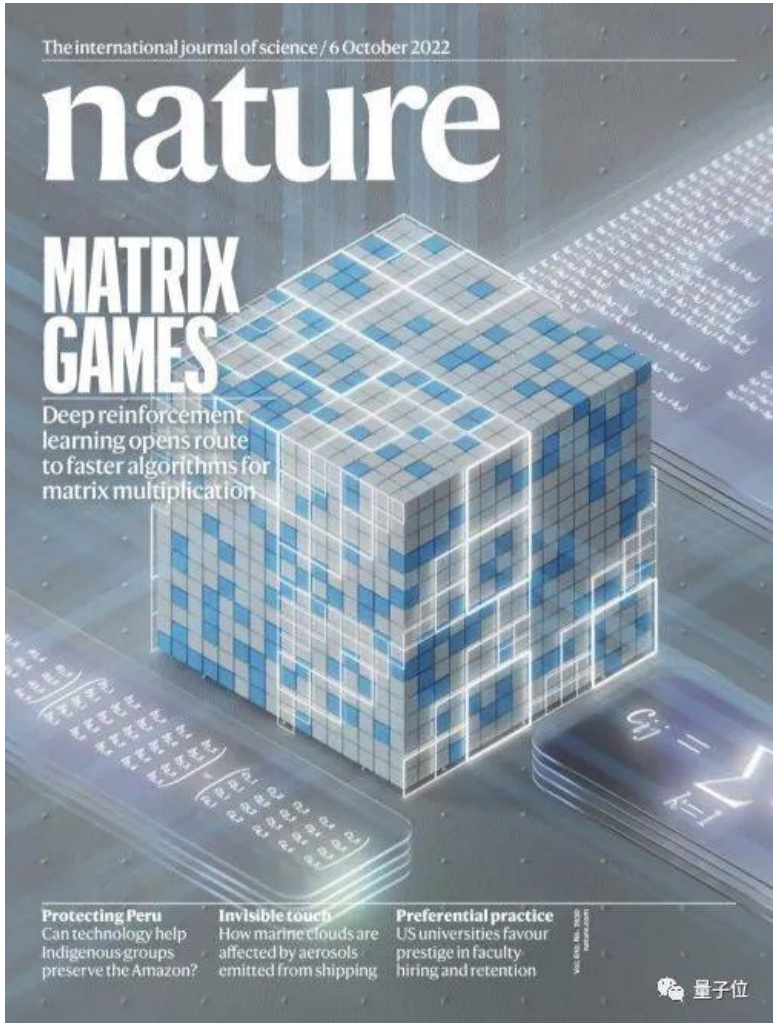
生成式人工智能（AIGC）

Stable Diffusion

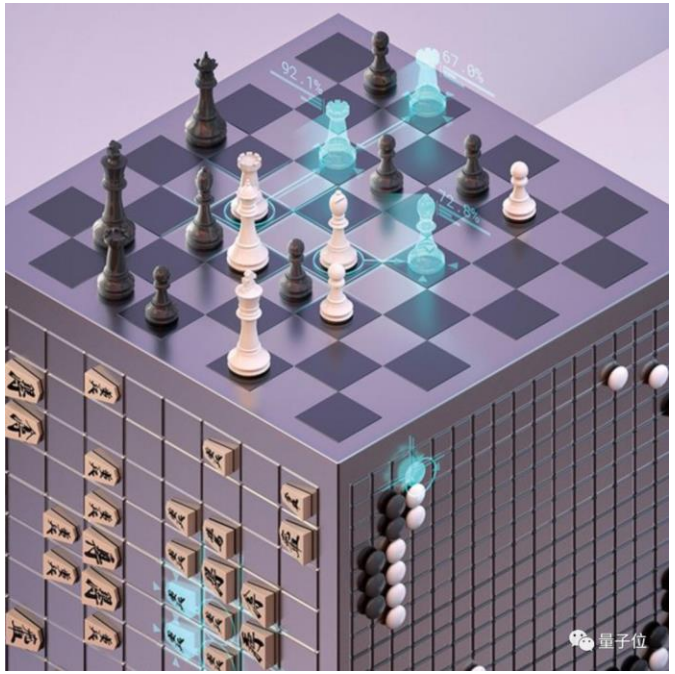
stability.ai

Sora

AlphaTensor



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$



AlphaCode

DeepMind

2022-2-2

Competition-Level Code Generation with AlphaCode

Yujia Li*, David Choi*, Junyoung Chung*, Nate Kushman*, Julian Schrittwieser*, Rémi Leblond*, Tom Eccles*, James Keeling*, Felix Gimeno*, Agustin Dal Lago*, Thomas Hubert*, Peter Choy*, Cyprien de Masson d'Autume*, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu and Oriol Vinyals

*Joint first authors

Programming is a powerful and ubiquitous problem-solving tool. Developing systems that can assist programmers or even generate programs independently could make programming more productive and accessible, yet so far incorporating innovations in AI has proven challenging. Recent large-scale language models have demonstrated an impressive ability to generate code, and are now able to complete simple programming tasks. However, these models still perform poorly when evaluated on more complex, unseen problems that require problem-solving skills beyond simply translating instructions into code. For example, competitive programming problems which require an understanding of algorithms and complex natural language remain extremely challenging. To address this gap, we introduce AlphaCode, a system for code generation that can create novel solutions to these problems that require deeper reasoning. Evaluated on recent programming competitions on the Codeforces platform, AlphaCode achieved on average a ranking of top 54.3% in programming competitions with more than 5,000 participants. We found that three key components were critical to achieve good and reliable performance: (1) an extensive and clean competitive programming dataset for training and evaluation, (2) large and efficient-to-sample transformer-based architectures, and (3) large-scale model sampling to explore the search space, followed by filtering based on program behavior to a small set of submissions.