



TAIJI LABORATORY
FOR GRAVITATIONAL WAVE UNIVERSE



ICTP-AP
International Centre
for Theoretical Physics Asia-Pacific
国际理论物理中心-亚太地区



中国科学院大学
University of Chinese Academy of Sciences

引力波数据探索：编程与分析实战训练营

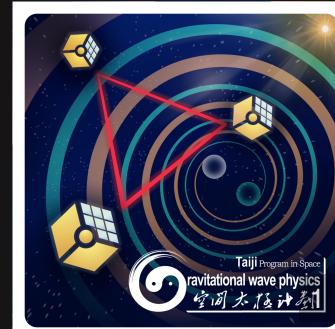
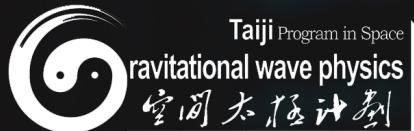
第4部分 深度学习基础

深度学习技术概述与神经网络基础

主讲老师：王赫

ICTP-AP, UCAS

2023/12/27





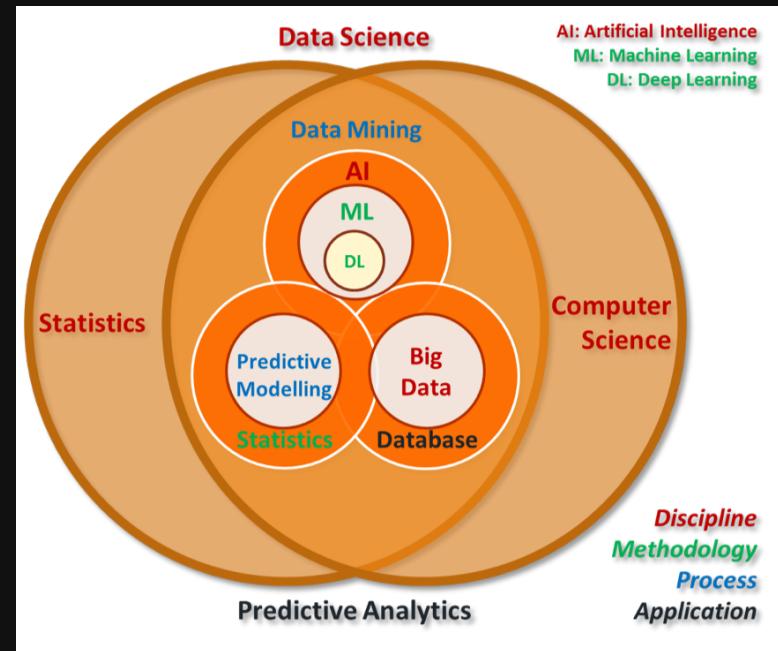
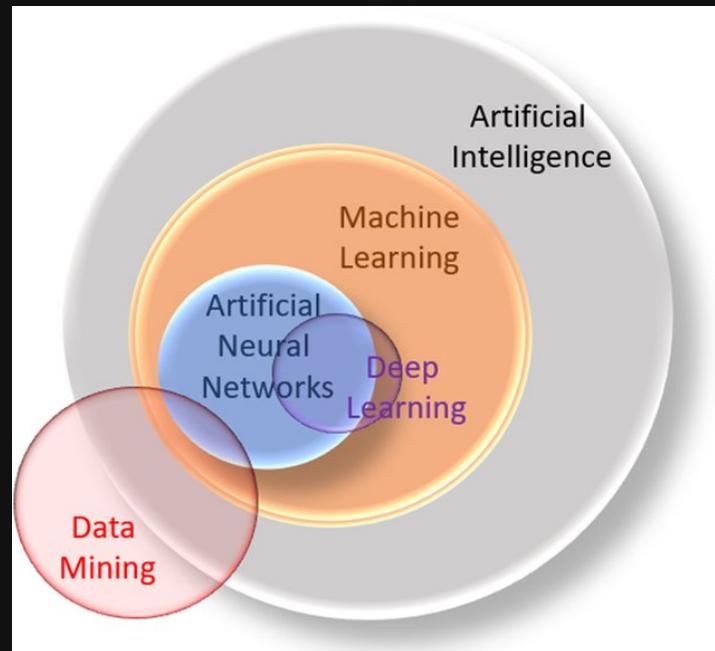
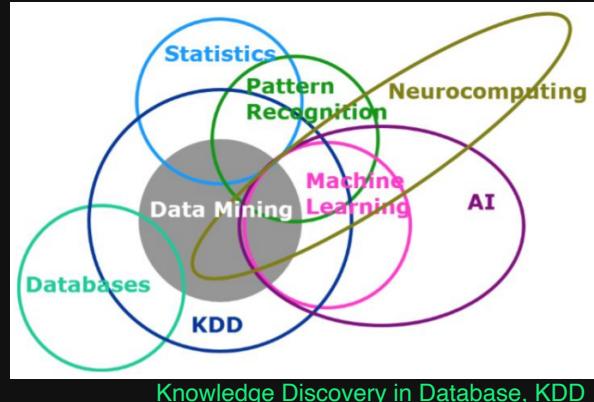
深度学习技术：概述

- 深度学习技术的起源
 - 一切的开始：感知器
- 深度学习技术的应用
- 深度学习技术的特点



深度学习技术的起源

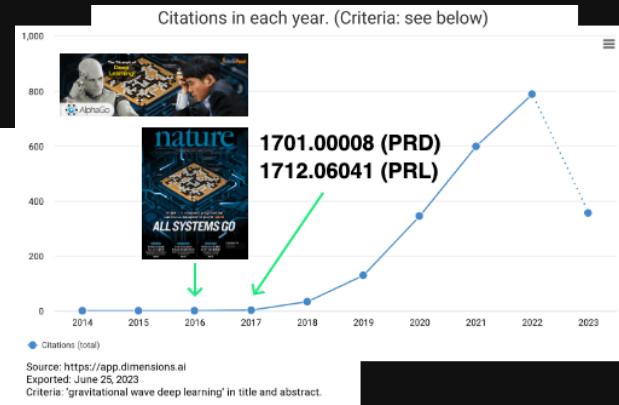
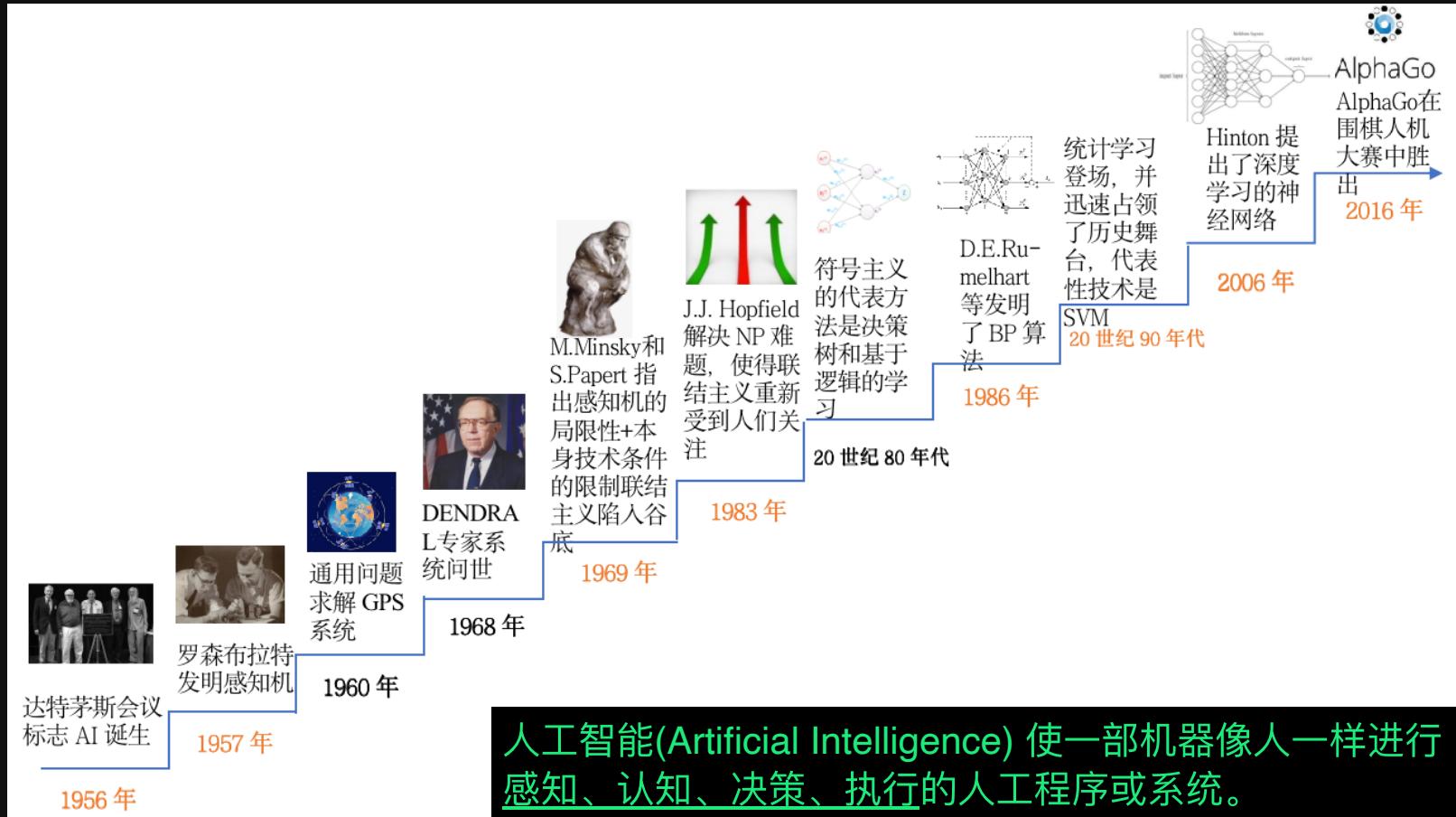
- 机器学习: 人工智能的一个重要学科分支, 多领域交叉学科
- 数据驱动: 在数据上通过算法总结规律模式, 应用在新数据上





深度学习技术的起源

- 人工智能发展标志事件 (Before 2017~)





深度学习技术的起源

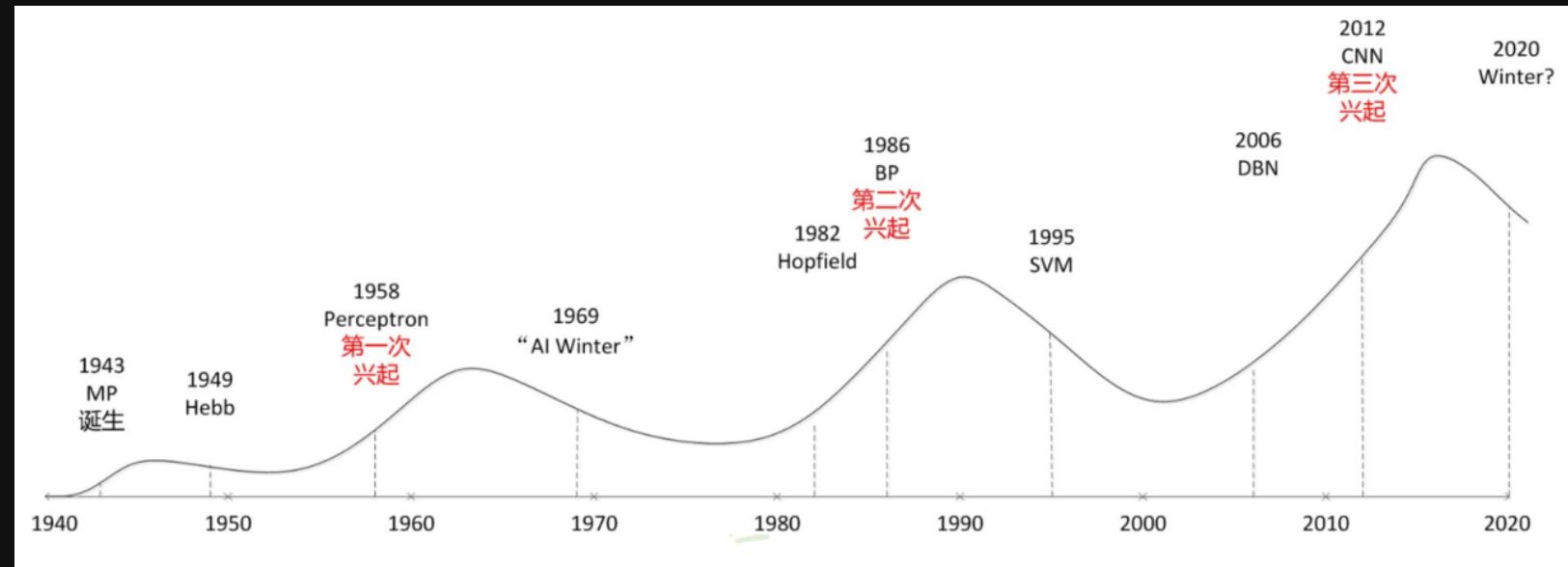
- 人工智能发展标志事件 (Before 2017~)

萌芽期	1943年，人工神经网络和数学模型建立， 人工神经网络研究时代开启 ； 1950年，计算机与人工智能之父图灵发表《机器能思考吗？》，提出“图灵测试”；	
启动期	1956年，达特茅斯会议召开，标志着 人工智能的诞生 ；期间，国际学术界人工智能研究潮流兴起，罗素《数学原理》被算法全部证明，学术交流频繁；	
消沉期	1969年，作为主要流派的 连接主义与符号主义进入消沉 ，四大预言遥遥无期，在计算能力的限制下，国际及公众信心持续减弱；	
突破期	1975年， BP 算法 开始研究，第五代计算机开始研制，专家系统的研究和应用艰难前行，半导体技术发展，计算机成本和计算能力逐步提高， 人工智能逐渐开始突破 ；	
发展期	1986年， BP 网络 实现，神经网络得到广泛认知，基于人工神经网络算法研究突飞猛进； 计算机硬件 能力快速提升；互联网构建， 分布式网络 降低了人工智能的计算成本；	
高速发展期	2006年，深度学习被提出，人工智能算法产生突破性发展； 2010年，移动互联网发展，人工智能应用场景开始增多； 2012年，深度学习算法在语言和视觉识别上实现突破，同年融资规模开始快速增长，人工智能 商业化高速发展 。	



深度学习技术的起源

- 神经网络的三起两落
 - 深度学习的起源可以追溯到神经网络的发展历程，这个过程经历了三次兴起和两次衰落。

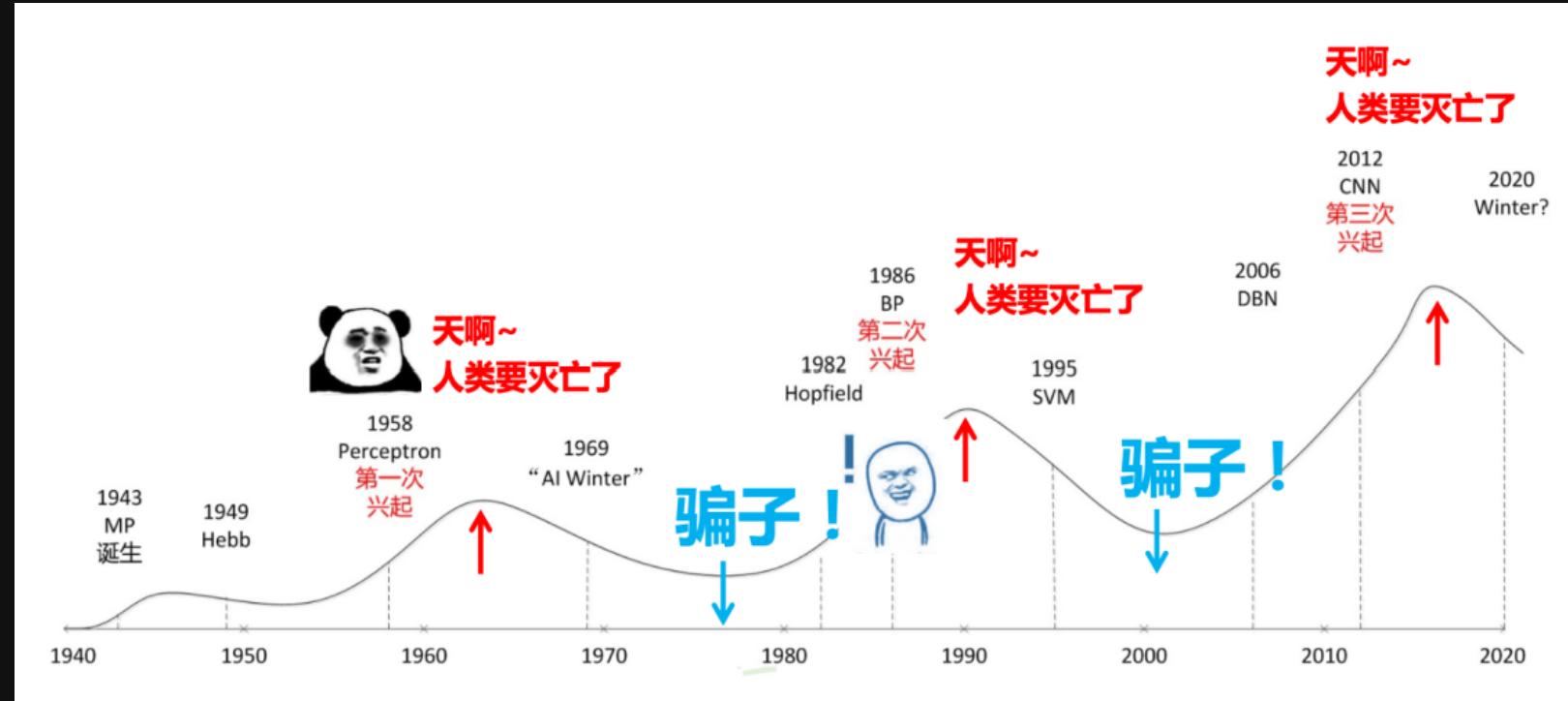


- 第一次兴起发生在20世纪50-60年代，当时研究者们提出了感知机模型，这是最早的神经网络模型之一。然而，由于感知机模型无法解决异或（XOR）问题，神经网络进入了第一次衰落期。
- 第二次兴起发生在80年代，当时研究者们提出了反向传播算法，使得神经网络能够学习到更复杂的模式。然而，由于计算能力的限制和过拟合问题，神经网络在90年代再次进入衰落期。
- 第三次兴起发生在21世纪初，这次兴起主要得益于计算能力的大幅提升和大数据的出现。这使得研究者们能够训练出更深层次的神经网络，也就是我们现在所说的深度学习模型。此外，新的优化算法和正则化技术的出现也帮助解决了过拟合问题，使得深度学习在各种任务上都取得了显著的效果。至今，深度学习仍在持续发展中，正在推动着人工智能领域的进步。

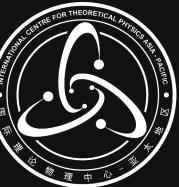


深度学习技术的起源

- 神经网络的三起两落



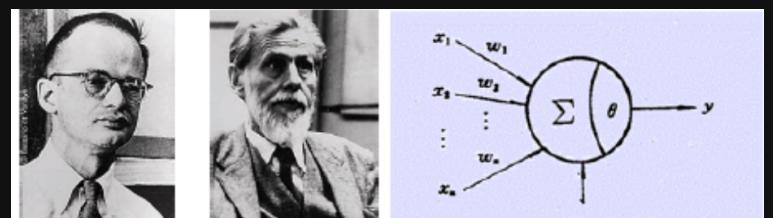
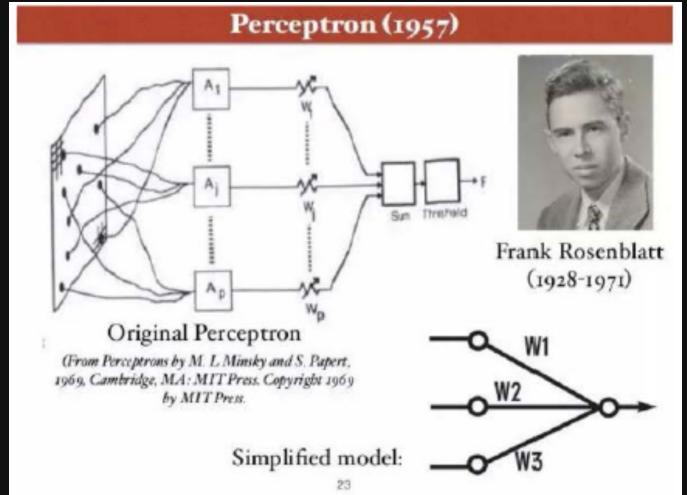
- 第一次兴起发生在20世纪50-60年代，当时研究者们提出了感知机模型，这是最早的神经网络模型之一。然而，由于感知机模型无法解决异或（XOR）问题，神经网络进入了第一次衰落期。
- 第二次兴起发生在80年代，当时研究者们提出了反向传播算法，使得神经网络能够学习到更复杂的模式。然而，由于计算能力的限制和过拟合问题，神经网络在90年代再次进入衰落期。
- 第三次兴起发生在21世纪初，这次兴起主要得益于计算能力的大幅提升和大数据的出现。这使得研究者们能够训练出更深层次的神经网络，也就是我们现在所说的深度学习模型。此外，新的优化算法和正则化技术的出现也帮助解决了过拟合问题，使得深度学习在各种任务上都取得了显著的效果。至今，深度学习仍在持续发展中，正在推动着人工智能领域的进步。



一切的开始：感知器

- Rosenblatt & Perceptron (感知器)

- **计算模型**: 1943 年最初由 Warren McCulloch 和 Walter Pitts 提出, 称为MP模型。他们通过MP模型提出了神经元的形式化数学描述和网络结构方法, 证明了单个神经元能执行逻辑功能, 从而开创了人工神经网络研究的时代。
- 1949年, 心理学家提出了突触联系强度可变的设想。
- 康奈尔大学 Frank Rosenblatt 1957年提出, 解决二分类问题, 利用梯度下降法, 自动更新权值。1962年, 改方法被证明收敛。
- Perceptron 是第一个具有自组织自学习能力的**数学模型**
- **Rosenblatt 乐观预测**: 感知器最终可以“学习, 做决定, 翻译语言”
- 感知器技术六十年代一度走红, 美国海军曾出自支持, 期望它“以后可以自己走, 说活看, 读, 自我复制, 甚至拥有自我意识”



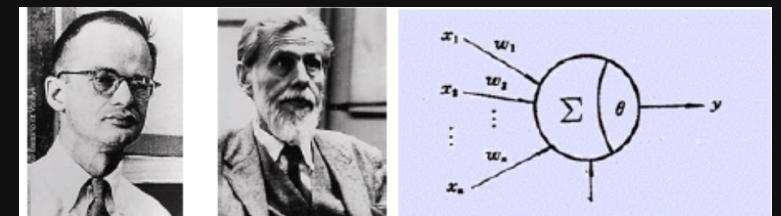
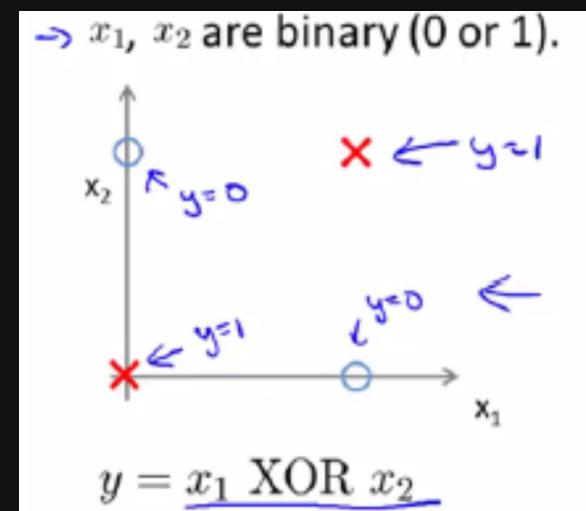
$$\begin{aligned}
 y(x) &= \operatorname{Sgn}(\sum_{i=1}^n w_i x_i - \theta) \\
 &= \begin{cases} 1, & \sum_{i=1}^n w_i x_i - \theta \geq 0 \\ 0, & \text{otherwise} \end{cases}
 \end{aligned}$$



一切的开始：感知器

- Rosenblatt & Minsky

- Rosenblatt 和 Minsky 是间隔一级的高中校友。但是六十年代，两个人在感知器的问题上展开了长时间的激辩：**R** 认为感知器将无所不能，**M** 则认为它应用有限
- 1969 年，Marvin Minsky 和 Seymour Papert 出版了新书：《感知器：计算几何简介》。书中论证了感知器模型的两个关键问题：
 - 第一，单层的神经网络无法解决不可线性划分的问题，典型例子如**异或门**
 - 第二，更致命的问题是：当时的电脑完全沒有能力完成神经网络模型所需的**超大计算量**
- 此后的十几年，以神经网络为基础的人工智能研究进入**低潮**（业界的核冬天：约20年停滞发展期）



$$y(x) = \text{Sgn}(\sum_{i=1}^n w_i x_i - \theta)$$

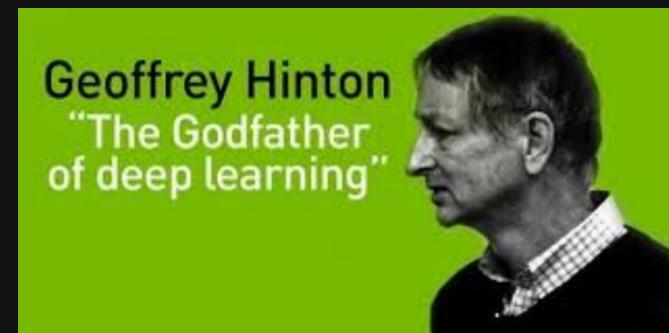
$$= \begin{cases} 1, & \sum_{i=1}^n w_i x_i - \theta \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



深度学习技术的发展

- Geoffrey Hinton & Neural Networks

- 1970 年，当神经网络研究的第一个寒冬降临时，在英国的爱丁堡大学，一位 23 岁的年轻人 **Geoffrey Hinton**，刚刚获得心理学的学士学位。
- Hinton 六十年代还是中学生就对**脑科学着迷**。当时一个同学给他介绍关于大脑记忆的理论是：大脑对于事物和概念的记忆，不是存储在某个单一的地点，而是像全息照片一样，分布式地存在于一个巨大的神经元的网络里。
- 分布式表征（**Distributed Rep.**）和传统的**局部表征（Localized Rep.）**相比：
 - 存储效率高：线性增加的神经元数目，可以表达指数级增加的大量不同概念。
 - 鲁棒性好：即使局部出现硬件故障，信息的表达不会受到根本性的破坏。
- 这个理念让 Hinton 顿悟，使他 40 多年来一致在**神经网络研究的领域内坚持**。
 - 本科毕业后，Hinton 选择继续在爱丁堡大学读研，把人工智能作为自己的博士研究方向。
 - 1978 年，Hinton 在爱丁堡获得博士学位后，来到美国继续他的研究工作。





深度学习技术的发展



All the knowledge is in the connections

— David Rumelhart —

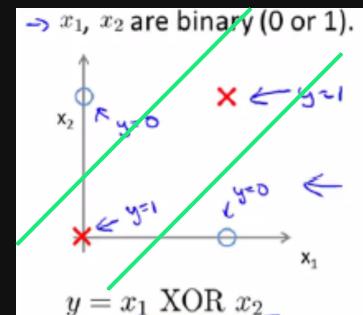
AZ QUOTES

- D. Rumelhart & BP Algorithm

- 神经网络被 Minsky 贬病的问题：巨大的计算量；XOR 问题；
- 传统的感知器用所谓“梯度下降”的算法纠错时，耗费的计算量和神经元数目的平方成正比，当神经元数目增多，庞大的计算量是当时的硬件无法胜任的。
- 1982年，美国加州理工物理学家J.J.Hopfield提出了 Hopfield 神经网格模型，引入了“计算能量”概念，给出了网络稳定性判断。
- 1986 年 7 月，Hinton 和 David Rumelhart 合作在 Nature 杂志上发表论文：*Learning Representations by Back-propagating Errors.* 第一次系统简洁地阐述 BP 算法及其应用：
 - 反向传播算法把纠错的运算量下降到只和神经元数目本身成正比；
 - BP 算法通过在神经网络里增加一个所谓隐层 (hidden layer) ，解决了 XOR 难题
 - 使用了 BP 算法的神经网络在做如形状识别之类的简单工作时，效率比感知器大大提高，八十年代末计算机的运行速度，也比二十年前高了几个数量级；
- 神经网络及其应用的研究开始复苏！

$$h_j = \text{Sgn}(\sum_{i=1}^n w_{ji} x_i - \theta_j)$$

$$y = \text{Sgn}(\sum_{j=1}^m w_j h_j - \theta)$$

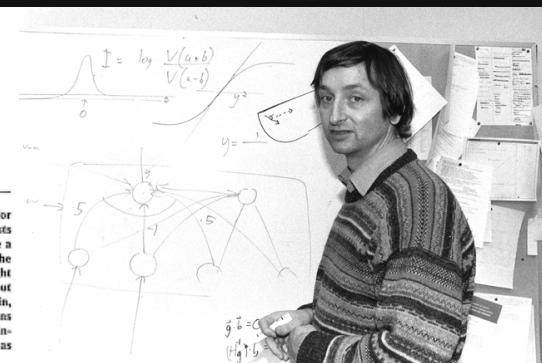


Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Sciences, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure*.



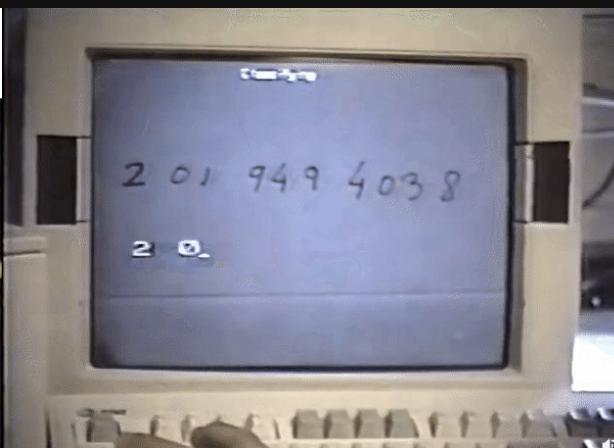


深度学习技术的发展



- **Yann LeCun (杨立昆) & CNN**

- Yann LeCun 于 1960 年出生于巴黎。
- 1987 年在法国获得博士学位后，他曾追随 Hinton 教授到多伦多大学做了一年博士后的工作，随后搬到新泽西州的 Bell Lab 继续研究工作。
- 在 Bell Lab, Lecun 1989 年发表了论文，“**反向传播算法在手写邮政编码上的作用**”。他用美国邮政系统提供的近万个手写数字的样本来训练神经网络系统，训练好的系统在独立的测试样本中，错误率只有 5%。
- Lecun 进一步运用一种叫做“卷积神经网络”（Convolutional Neural Networks, CNN）的技术，开发出商业软件，用于读取银行支票上的手写数字，这个支票识别系统在九十年代末占据了美国接近 20% 的市场。



2003 年，Yann LeCun 等人在 NEC 实验室的使用 CNN 进行人脸检测。

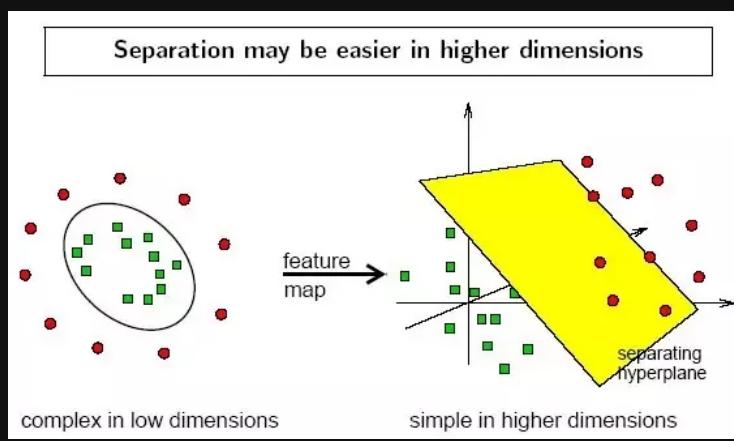


深度学习技术的发展



- **Yann LeCun (杨立昆) & CNN**

- Yann LeCun 于 1960 年出生于巴黎。
- 1987 年在法国获得博士学位后，他曾追随 Hinton 教授到多伦多大学做了一年博士后的工作，随后搬到新泽西州的 Bell Lab 继续研究工作。
- 在 Bell Lab, LeCun 1989 年发表了论文，“**反向传播算法在手写邮政编码上的作用**”。他用美国邮政系统提供的近万个手写数字的样本来训练神经网络系统，训练好的系统在独立的测试样本中，错误率只有 5%。
- LeCun 进一步运用一种叫做“卷积神经网络”（Convolutional Neural Networks, CNN）的技术，开发出商业软件，用于读取银行支票上的手写数字，这个支票识别系统在九十年代末占据了美国接近 20% 的市场。
- 此时就在 Bell Lab, Yann LeCun 临近办公室的一个同事 Vladimir Vapnik 的工作，又把神经网络研究带入第二个寒冬！



SVM (support vector machines)



在90年代，人工神经网络缺少严格的数学理论支撑，统计学习大发展。Vapnik提出支持向量机(SVM)，改进了感知器的一些缺陷(例如创建灵活的特征而不是手编的非适应的特征)。它同样解决了线性不可分问题，但是对比神经网络有全方位优势：

1. 高效，可以快速训练；
2. 无需调参，没有梯度消失问题；
3. 高效泛化，全局最优解，不存在过拟合问题。



深度学习技术的发展



- **Hinton & Deep Learning**

- 2003年, Geoffrey Hinton 还在多伦多大学, 在神经网络的领域苦苦坚守。
- 2003 年在温哥华大都会酒店, 以 Hinton 为首的十五名来自各地的不同专业的科学家, 和加拿大先进研究员 (Canadian Institute of Advanced Research, CIFAR) 的基金管理负责人 Melvin Silverman 交谈。
 - Silverman 问大家, 为什么 CIFAR 要支持他们的研究项目。
 - 计算神经科学的研究者, Sebastian Sung (现为普林斯顿大学教授) 回答道: “喔, 因为我们有点古怪。如果 CIFAR 要跳出自己的舒适区, 寻找一个高风险, 极具探索性的团体, 就应当资助我们了! ”
 - 最终 CIFAR 同意从 2004 年开始资助这个团体十年, 总额一千万加元。CIFAR 成为当时世界上唯一支持神经网络研究的机构。

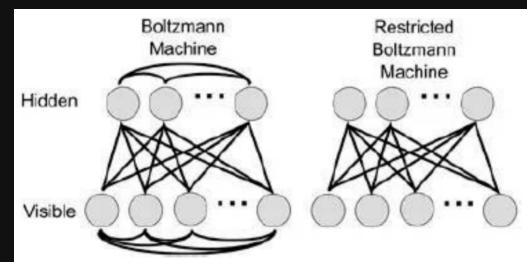
- Hinton 拿到资金支持不久做的第一件事, 就是把“神经网络”改名换姓为“深度学习”。
- 此后, Hinton 的同时不时会听到他突然在办公室大叫: “我知道人脑是如何工作的了! ”
- 2006 年 Hinton 和合作者发表革命性的论文: *A Fast Learning Algorithm for Deep Belief Nets* .

[Reducing the dimensionality of data with neural networks](#)
GE Hinton, RR Salakhutdinov
science, 2006 · science.org

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such “autoencoder” networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool

展开 ▾

☆ 保存 99 引用 被引用次数: 21794 相关文章 所有 26 个版本



- 逐层初始化 (layer-wise pre-training)
 - 预训练 (pre-training)
 - 微调 (fine-tuning)
- 被 Hinton 首次定义为深度学习过程

CIFAR
40
YEARS · ANS



深度学习技术的发展

- Andrew Y. Ng & GPU
 - 2007 年之前，用 GPU 编程缺乏一个简单的软件接口，编程繁琐，Debug 困难。2007 年 NVIDIA 推出 CUDA 的 GPU 软件接口后才真正改善。
 - 2009 年 6 月，斯坦福大学的 Rajat Raina 和吴恩达合作发表论文：*Large-scale Deep Unsupervised Learning using Graphic Processors* (ICML09)；论文采用 DBNs 模型和稀疏编码 (Sparse Coding)，模型参数达到一亿（与 Hinton 模型参数的对比见下表）。
 - 结论结果显示：使用 GPU 运行速度和用传统双核 CPU 相比，最快时要快近 70 倍。在一个四层，一亿个参数的 DBN 网络上使用 GPU 把程序运行时间从几周降到一天。

Published source	Application	Params
Hinton et al., 2006	Digit images	1.6mn
Hinton & Salakhutdinov	Face images	3.8mn
Salakhutdinov & Hinton	Sem. hashing	2.6mn
Ranzato & Szummer	Text	3mn
Our model		100mn

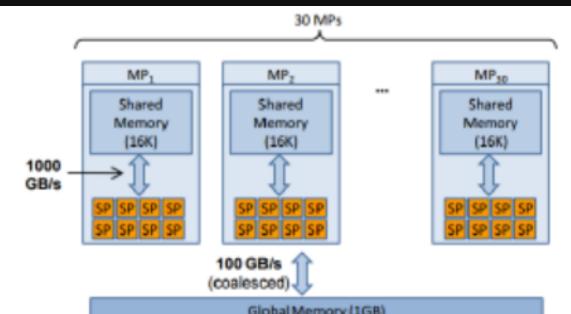


Figure 1. Simplified schematic for the Nvidia GeForce GTX 280 graphics card, with 240 total cores (30 multi-processors with 8 stream processors each).

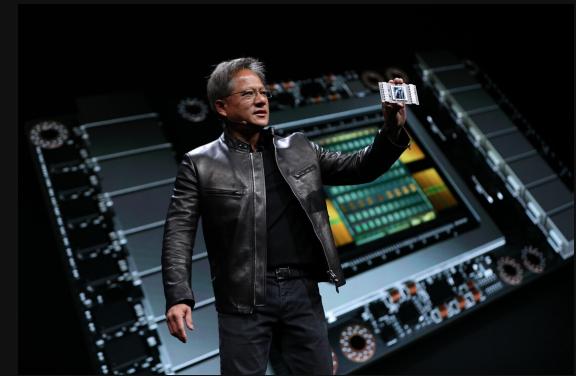




深度学习技术的发展

- Jen-Hsun Huang & GPU

- 黄仁轩，1963 年出生于台湾。1993 年斯坦福大学硕士毕业后不久创立了 NVIDIA。
- NVIDIA 起家时做的是图像处理的芯片，主要面对电脑游戏市场。1999 年 NVIDIA 推销自己的 Geforce 256 芯片时，发明了 GPU (Graphics Processing Unit) 这个名词。
- GPU 的主要任务，是要在最短时间内显示上百万、千万甚至更多的像素。这在电脑游戏中是最核心的需求。这个计算工作的核心特点，是要同时并行处理海量的数据。
- 传统的 CPU 芯片架构，关注点不在并行处理，一次只能同时做一两个加减法运算。而 GPU 在最底层的算术逻辑单元 (ALU, Arithmetic Logic Unit)，是基于所谓的 Single Instruction Multiple Data (单指令多数据流) 的架构，擅长对于大批量数据并行处理。
- 一个 GPU，往往包含几百个 ALU，并行计算能力极高。所以尽管 GPU 内核的时钟速度往往比 CPU 的还要慢，但对大规模并行处理的计算工作，速度比 CPU 快许多。
- 神经网络的计算工作，本质上就是大量的矩阵计算的操作，因此特别适合于使用 GPU。

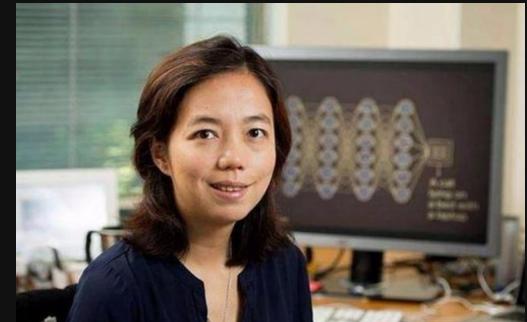
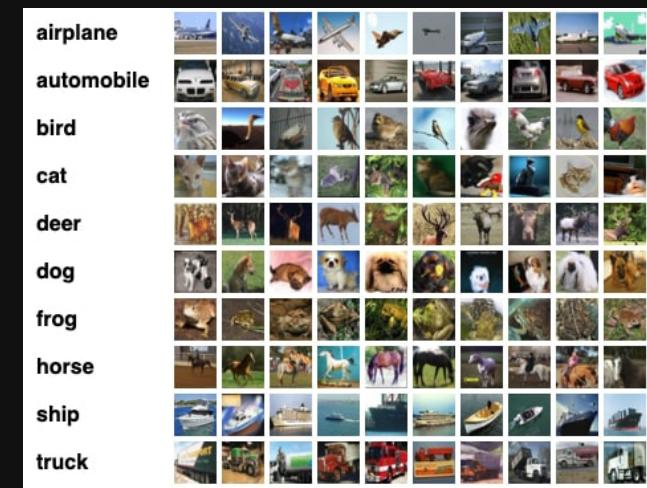




深度学习技术的发展

- Big Data: ImageNet

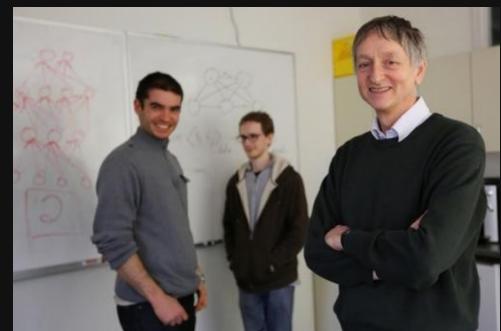
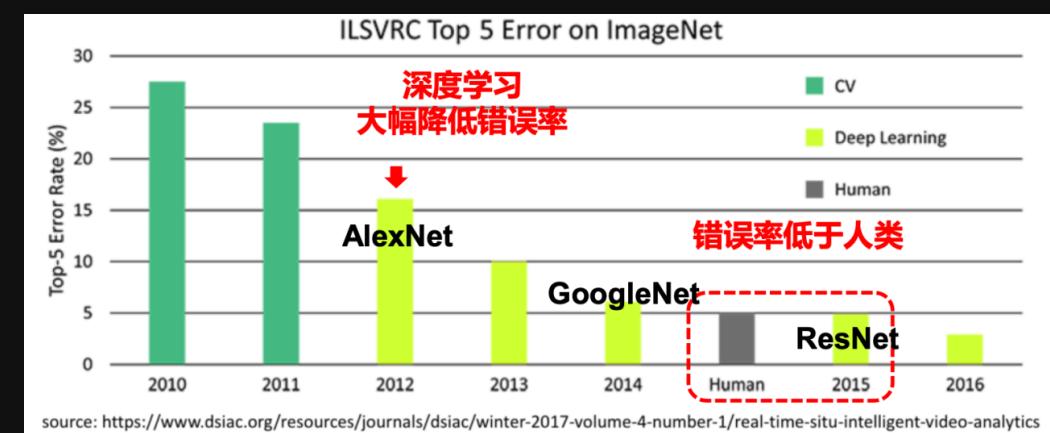
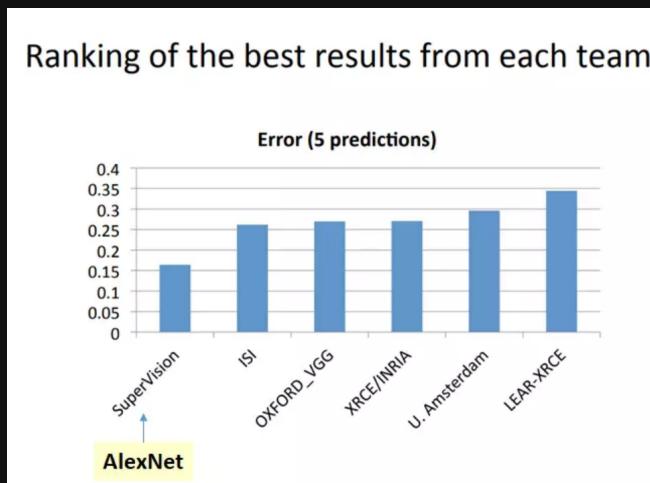
- 2009 年，一群普林斯顿大学计算机系的华人学者（李飞飞教授领衔）发表了论文：
ImageNet: A large scale hierarchical image database, 宣布建立了第一个超大型图像数据库供计算机视觉研究者使用。
- 数据库建立之初，包含了 320 万个图像。它的目的，是要把英文里的 8 万个名词，每个词收集到五百到一千个高清图片，存放到数据库里，最终达到五千万以上的图像。
- 2010 年，以 ImageNet 为基础的大型图像识别竞赛，*ImageNet Large Scale Visual Recognition Challenge 2010* (ILSVRC2010) 第一次举办。[<http://www.image-net.org/>]
- 竞赛最初的规则：以数据库内 120 万个图像为训练样本，这些图像从属于一千多个不同的类别，都被手工标记。经过训练的程序，再用于 5 万个测试图像评估分类准确率。





深度学习技术的发展

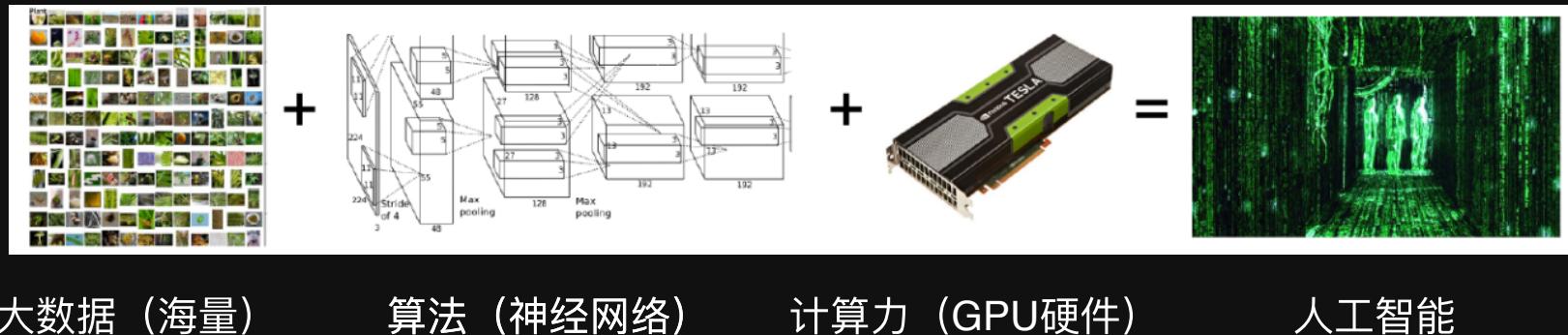
- Image Classification: ILSVRC 竞赛
 - 2010 年冠军：NEC 和伊利诺伊大学香槟分校的联合团队，用支持向量机（SVM）的技术识别分类的错误率 28%。
 - 2011 年冠军：用 Fisher Vector 的计算方法（类似 SVM），将错误率降到了 25.7%。
 - 2012 年冠军：Hinton 和两个 Alex Krizhevsky, Illya Sutskever，利用 CNN+Dropout 算法 +RELU 激励函数，用了两个 NVIDIA 的 GTX580 GPU（内存 3GB，计算速度 1.6 TFLOPS），花了接近 6 天时间，错误率只有 15.3%。
 - 2012 年 10 月 13 日，当竞赛结果公布后，学术界沸腾了。这是神经网络二十多年来，第一次在图像识别领域，毫无疑义的，大幅度挫败了别的技术。
 - 这是人工智能技术突破的一个重要转折点！





深度学习技术的应用

- 深度学习的三个助推剂



- 深度学习三巨头+粉丝

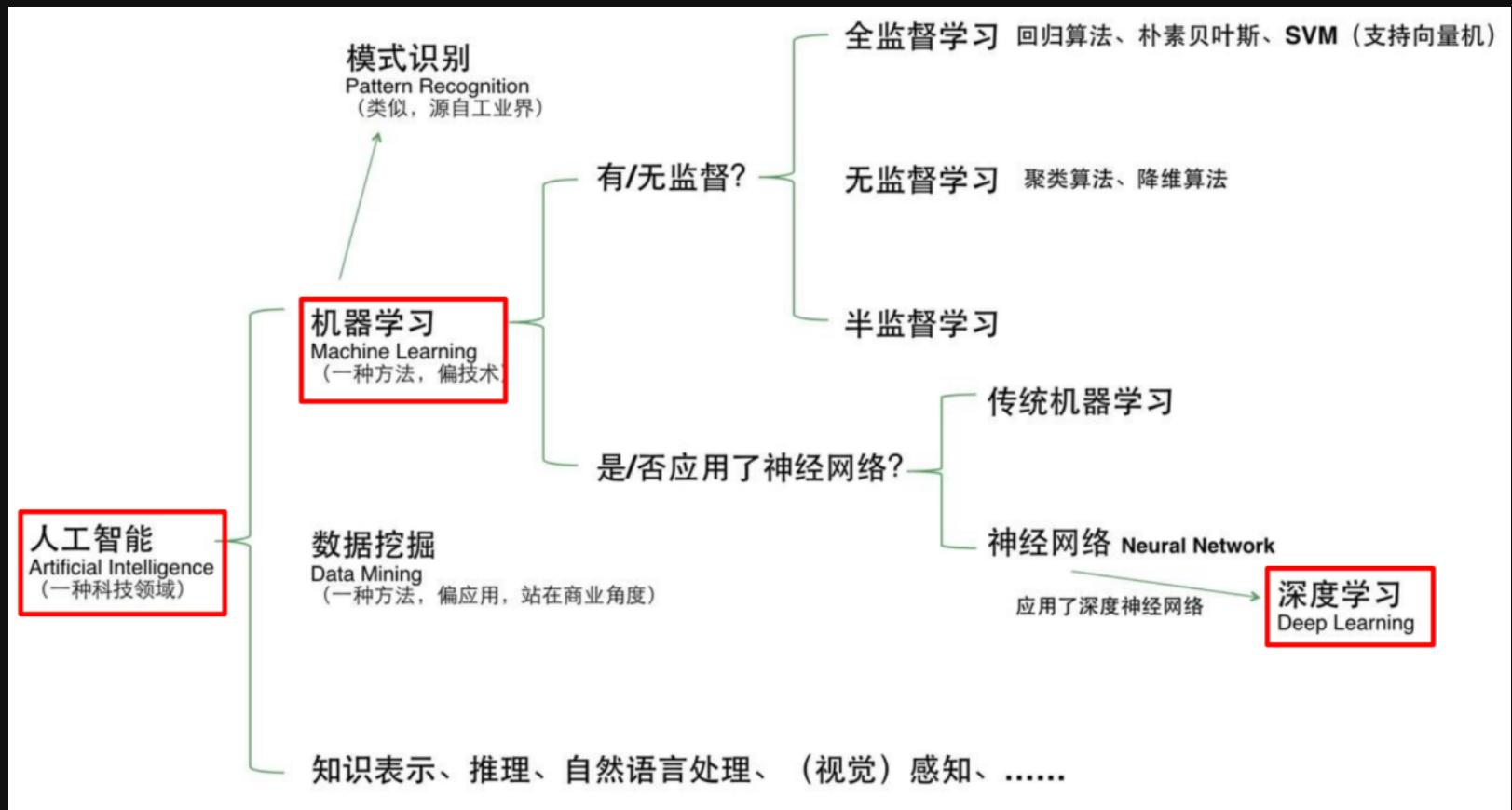


LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning." *Nature* 521, no. 7553 (May 1, 2015): 436–44.
<https://doi.org/10.1038/nature14539>.



深度学习技术的特点

- 人工智能 > 机器学习 > 深度学习





深度学习技术的特点

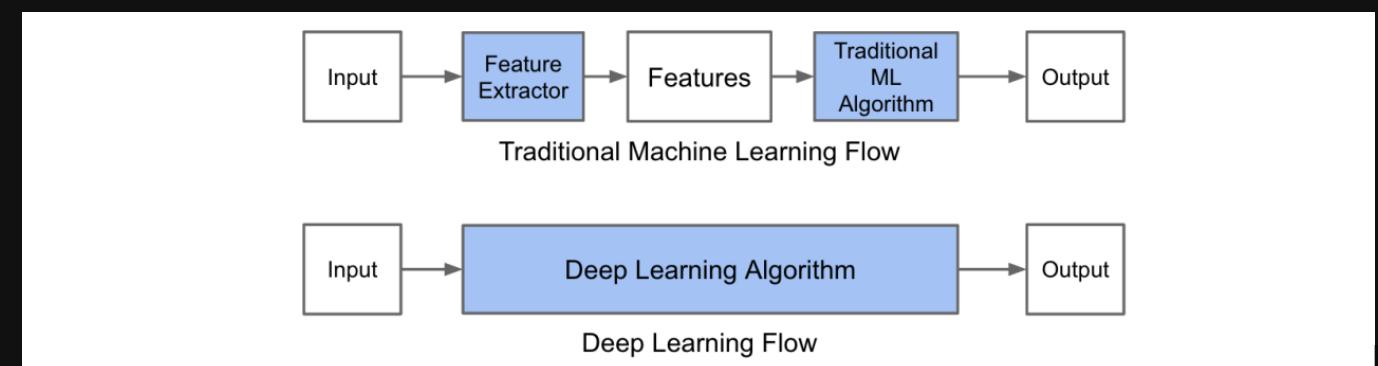
- 传统机器学习 vs 深度学习
 - 传统机器学习：人工设计特征
 - 在实际应用中，设计**特征**往往比分类器更重要
 - 预处理：经过数据的预处理，如去除噪声等。比如在文本分类中，去除停用词等。
 - 特征提取：从原始数据中提取一些有效的特征。比如在图像分类中，提取边缘、尺度不变特征变换特征等。
 - 特征转换：对特征进行一定的加工，比如降维和升维。降维包括
 - 特征抽取（Feature Extraction）：PCA、SVD、LDA
 - 特征选择（Feature Selection）：互信息、TF-IDF
 - 深度学习：一种**端到端** [end-to-end] 的学习范式
 - 推动了一大类**非线性映射函数学习问题** 的解决
 - 从人工编码知识 → 从数据中学习知识
 - 分而治之 → 全盘考虑
 - 重算法 → 重数据





深度学习技术的特点

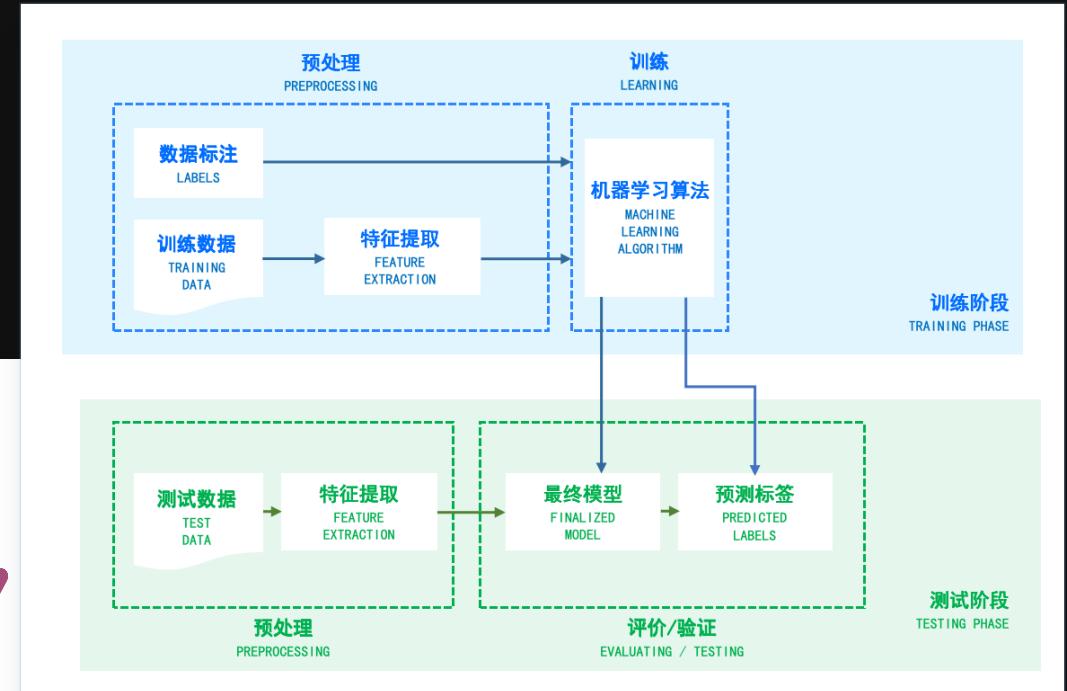
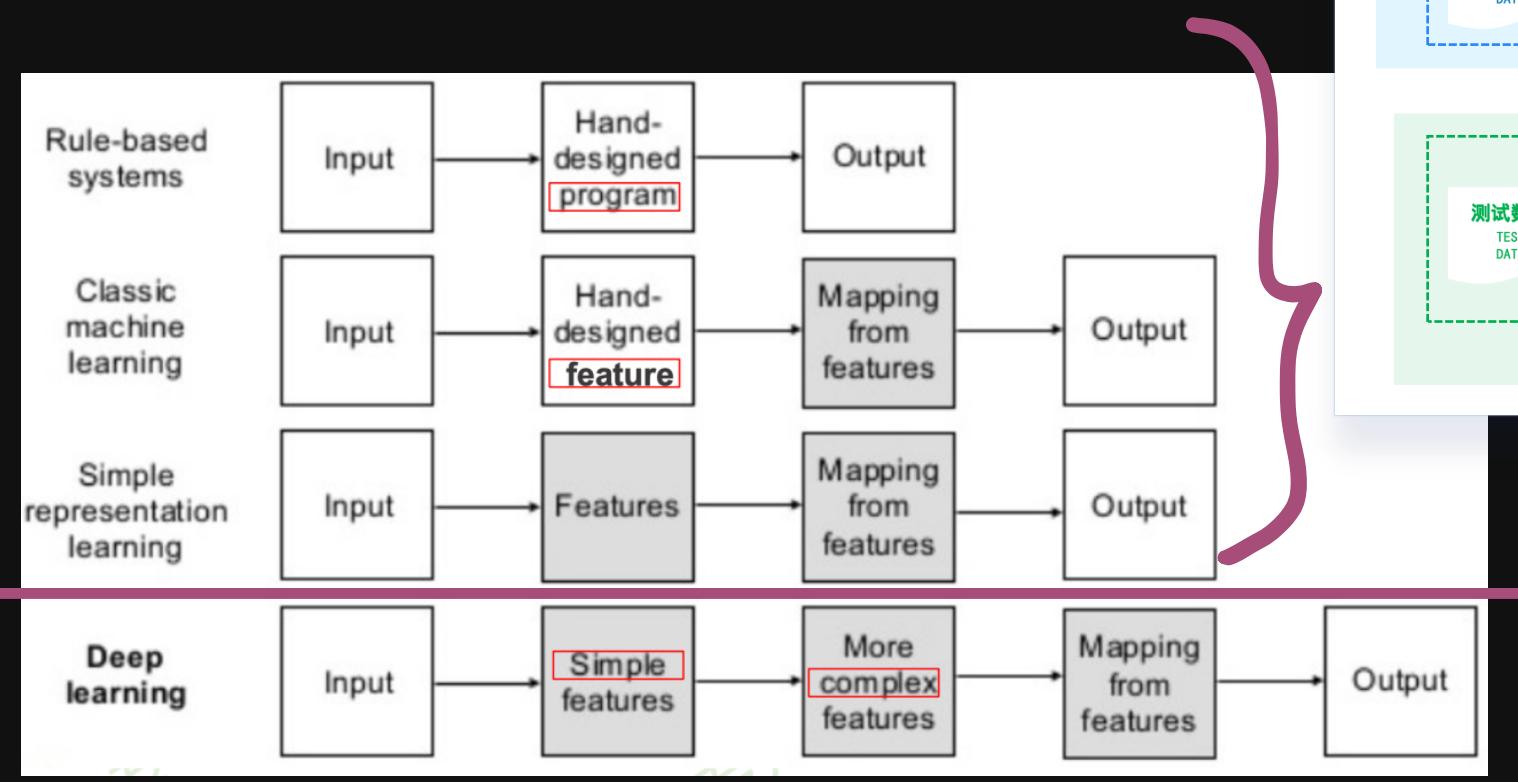
- 传统机器学习 vs 深度学习
 - 传统机器学习：人工设计特征
 - 在实际应用中，设计**特征**往往比分类器更重要
 - 预处理：经过数据的预处理，如去除噪声等。比如在文本分类中，去除停用词等。
 - 特征提取：从原始数据中提取一些有效的特征。比如在图像分类中，提取边缘、尺度不变特征变换特征等。
 - 特征转换：对特征进行一定的加工，比如降维和升维。降维包括
 - 特征抽取（Feature Extraction）：PCA、SVD、LDA
 - 特征选择（Feature Selection）：互信息、TF-IDF
 - 深度学习：一种**端到端** [end-to-end] 的学习范式
 - 推动了一大类**非线性映射函数学习问题** 的解决。
 - 从人工编码知识 → 从数据中学习知识
 - 分而治之 → 全盘考虑
 - 重算法 → 重数据





深度学习技术的特点

- 深度学习：一种端到端 [end-to-end] 的学习范式
 - 推动了一大类 非线性映射函数学习问题 的解决。
 - 从人工编码知识 → 从数据中学习知识
 - 分而治之 → 全盘考虑
 - 重算法 → 重数据



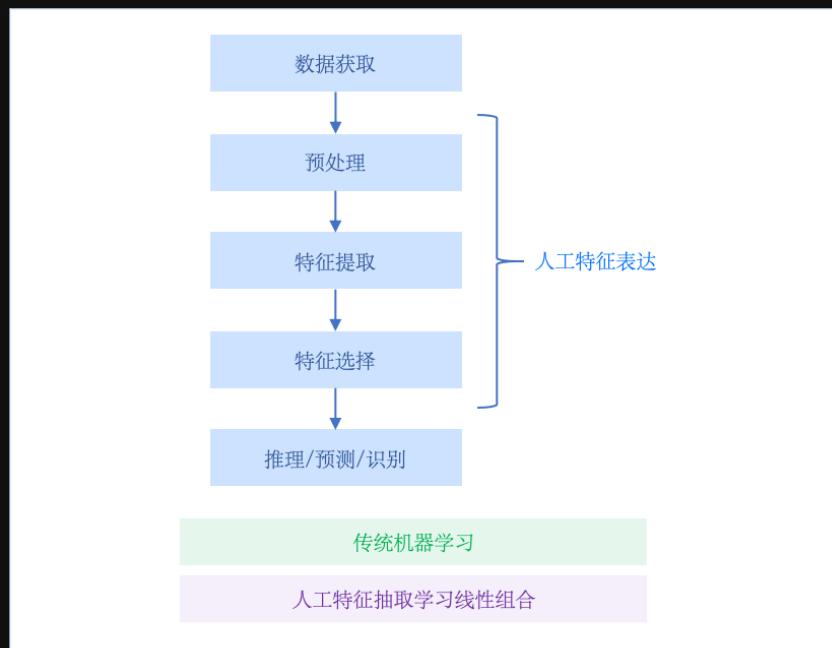
- End-to-end principle



深度学习技术的特点

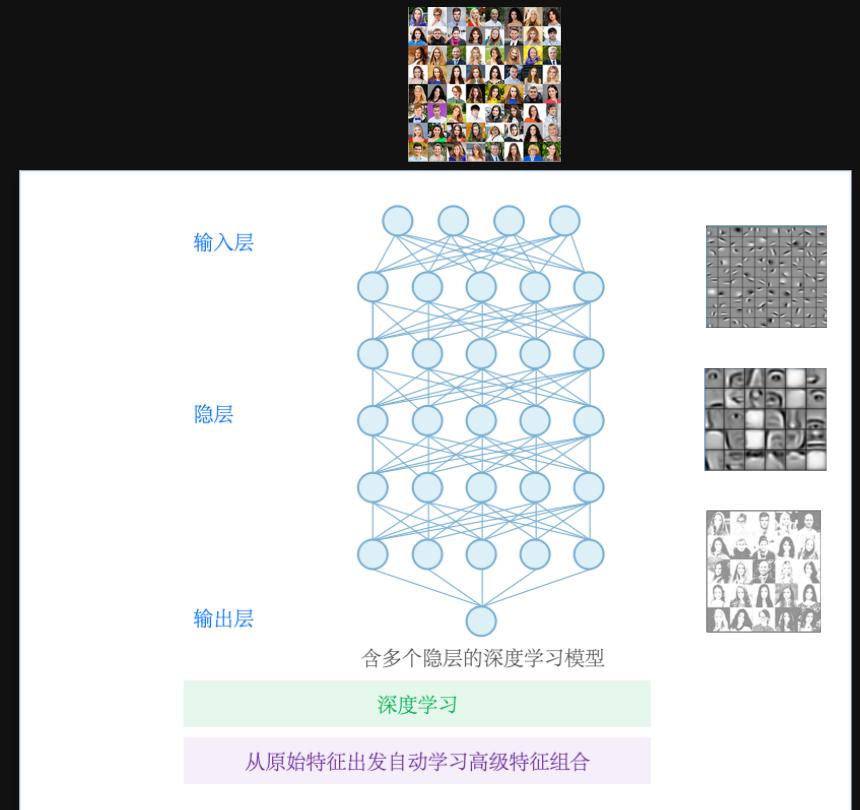
- 机器学习

- 依赖于**人工设计的特征**: 在机器学习的过程中, 我们通常需要依赖于专家知识或者经验规则来设计和选择特征。这些特征能够帮助模型从数据中学习到有用的信息, 从而进行有效的预测或决策。



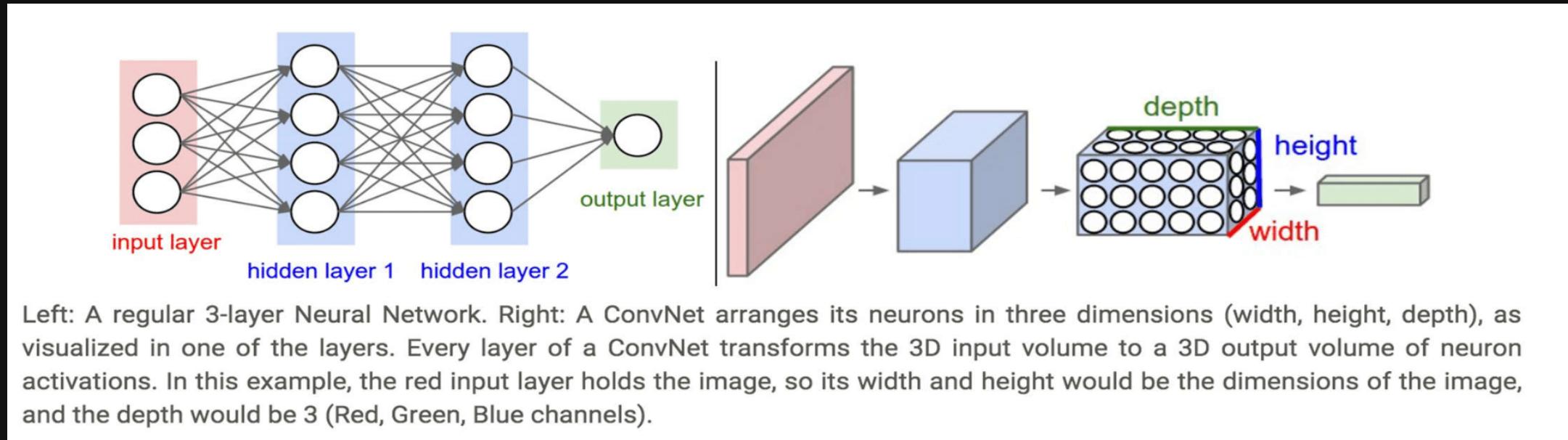
- 深度学习

- 端到端的学习范式: 深度学习是一种**自动化的特征学习方法**, 它能够自动地从原始数据中学习到有用的特征。这种端到端的学习范式避免了人工设计特征的需要, 使得模型能够直接从数据中学习到解决问题所需的所有知识, 大大提高了模型的学习能力和效率。





深度学习技术的本质

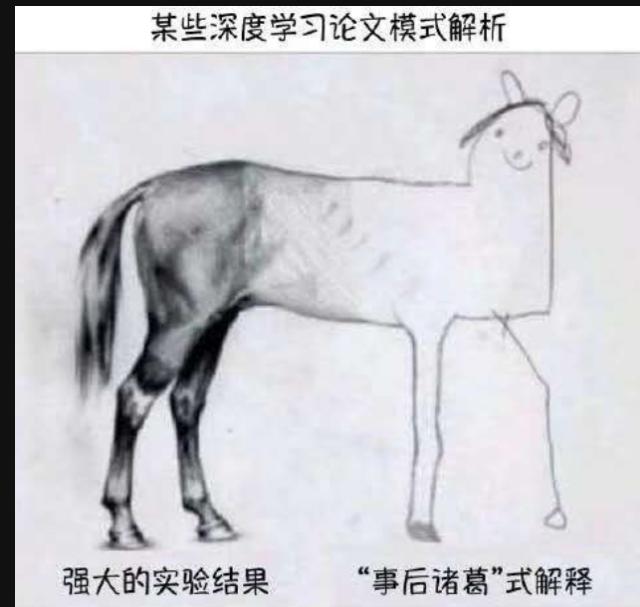
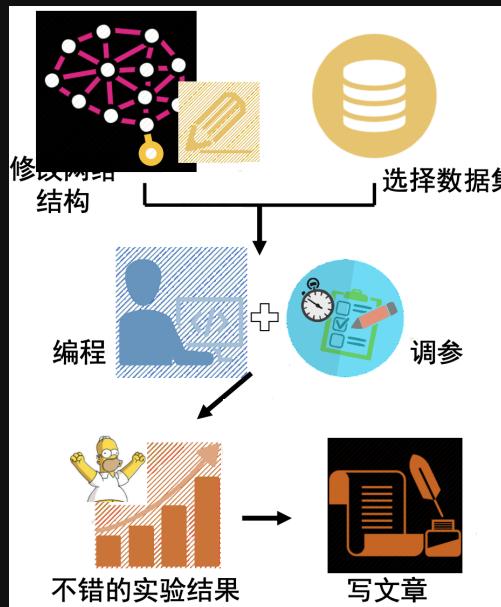


- **本质**: 通过构建多隐层的模型和海量训练数据(可为无标签数据), 来学习更有用的特征, 从而最终提升分类或预测的准确性。“深度模型”是手段, “特征学习”是目的。
- **与浅层学习区别**:
 1. 强调了模型结构的深度, 通常有5-10多层的隐层节点;
 2. 明确突出了特征学习的重要性, **通过逐层特征变换, 将样本在原空间的特征表示变换到一个新特征空间**, 从而使分类或预测更加容易。与人工规则构造特征的方法相比, 利用大数据来学习特征, 更能够刻画数据的丰富内在信息。



深度学习技术：工程技术 or 科学研究？

- 工程低门槛 & 科研高门槛（**门槛非常低 + 天花板非常高**）
- "科学为技术的发展提供基础和支撑，而技术进步则不断地向科学研究提出新的课题，反过来激励科学发展。"
- "不应该简单地把发展技术的思路和措施直接搬过来为发展科学铺路，也不应该简单地套用管理技术发展的政策和方式来经营科学发展。"



EDITORIAL

National Science Review
4: 665, 2017
doi: 10.1093/nsr/nwx071
Advance access publication 24 June 2017

Science and technology, not SciTech

Guanrong Chen

The recurrent phrase 'science and technology' ranks one ahead of the other, but the two words have been treated the same by many and, in practice, their priorities have often been swapped.

Science, more precisely natural science, refers to a system or notion of acquiring knowledge through experimentation, simulation and analysis to understand and explain natural phenomena. Within the context of this discussion, it could also include mathematical science. Technology, on the other hand, refers to the collection of techniques, methods, skills and processes that are applicable to the generation of products or services beneficial to human society.

Between the two, science provides a foundation for technology to develop. Conversely, advancement in technology continuously generates new motivation and poses new questions to science. The late great scientist Qian Xuesen (Hsue-Shen Tsien, 1911–2009) believed that there is an important component, which he named engineering science, connecting the two together.

Scientific advances have mostly been driven by human curiosity to understand the basic principles governing the natural world, rather than the desire to meet human needs. Many incidences of discovery emerged unexpectedly, beyond human prediction or planning, and they might not be recognized as such within a short time. To name a couple of examples, mathematical number theory has a 3000-year-old history but it was considered particularly useful only when it was successfully applied to modern cryptography. The esoteric theory of general relativity of Albert Einstein had been placed in Heaven but recently stepped down to Earth with the GPS application. The structure of the DNA double helix was discovered due to the curiosity of James Watson and Francis Crick about genetic inheritance, which has lately revolutionized both life sciences and biotechnology.

It thus has become clear that, in promoting science and technology, one should not take the same approach and, in particular, one should not simply borrow the ideas from technology development to pave the way for science to evolve. Methodologies and policies from technology management should not be simply applied to managing science. However, it is not uncommon today that many administrative decision makers in academia rely on their 'technological thinking' to target everything including science, believing that centralized planning, big money and fast-track promotions alike would be able to spur science to develop

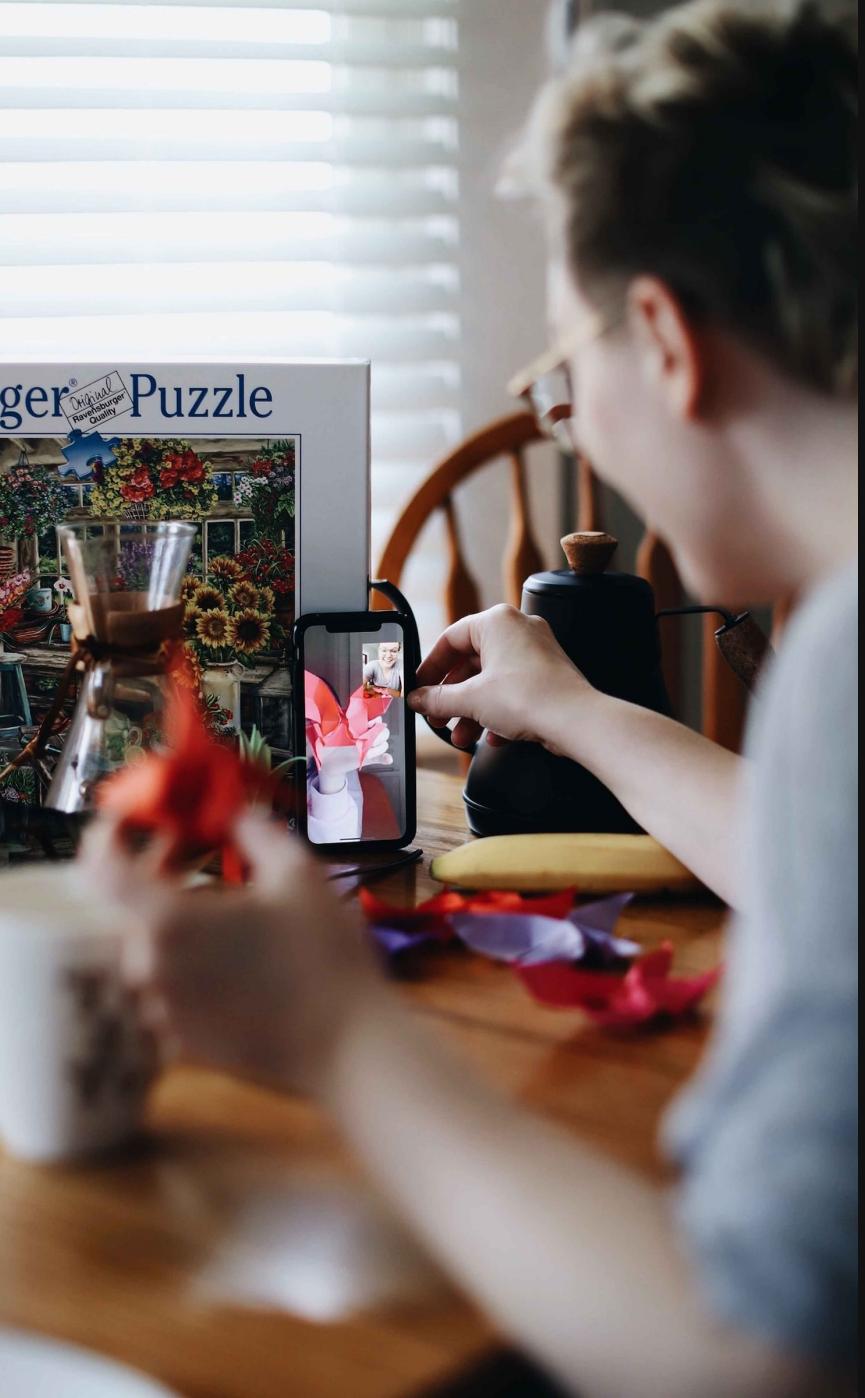
and excel. Furthermore, prevailing views and policies measure the values of scientific research based solely on whether it is useful in providing services to the society or whether it is able to deliver marketable products in the foreseeable future. In so doing, some long-term fundamental scientific research would be ruled out because it could be labeled 'useless' from a technological point of view, especially at its initial stage.

In responding to such science and technology governing, Helmut Schwarz, President of the Alexander von Humboldt Foundation, recently points out that 'most breakthroughs in research are not and could not be planned. Rather, they appear, like Puck, in entirely unexpected corners. Because it is the passion of individuals that sparks major discoveries or inventions, choosing outstanding people and providing intellectual freedom and generous funding are key to the success of academic institutions' (On the usefulness of useless knowledge. *Nature Reviews Chemistry* 2017; doi: 10.1038/S41570-016-0001).

Notably, in the common Chinese wording of SciTech (科技), this compound abbreviation of 'science and technology' is usually understood and presented as one single subject, leading to the widespread misconception of science and technology as synonym, to be viewed and managed in the same way. This is a problem throughout the long history of China. Cumulated observations and evidence suggest that this view of SciTech may be one of the reasons that modern science did not emerge in China. In fact, most Chinese ancient advances were developed towards technology for their practical values but did not evolve into building fundamental scientific knowledge and theories. For example, the discovery of gunpowder did not lead to modern theoretical chemistry, the creation of the compass did not lead to modern electromagnetics theory or theoretical physics and the ancient Chinese remainder theorem did not lead to modern number theory in mathematics.

That technological innovations were not accompanied by the establishment of modern science has long been a big puzzle that remains for Chinese scientists and technologists to be fully unraveled which, if well resolved, might quickly lead Chinese modern science to the forefront.

Guanrong Chen
Chair Professor & Director, Centre for Chaos and Complex Networks,
City University of Hong Kong
Editorial Board Member of NSR
E-mail: eegchen@cityu.edu.hk



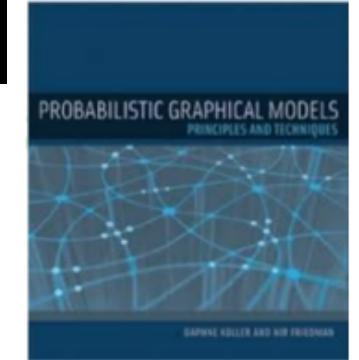
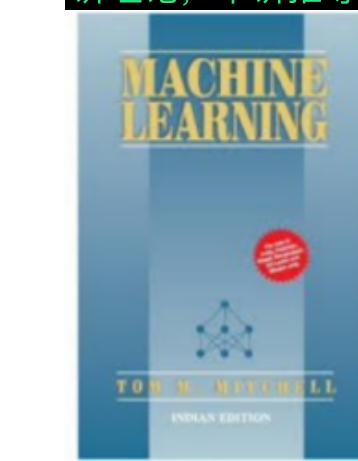
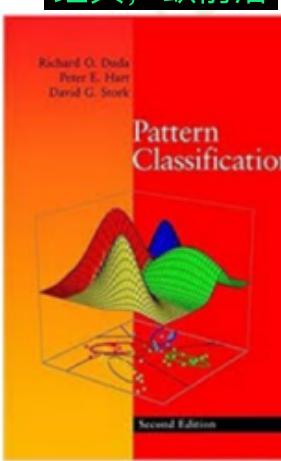
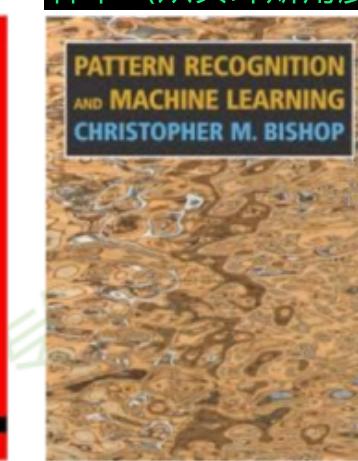
人工智能技术自学材料推荐

- 经典书籍和专著
- 优秀课程资源
- 值得关注的公众号



机器学习与深度学习技术的学习材料

2k 多页，难啃，概率
模型的角度出发

<p>书中例子多而形象， 适合当做工具书</p>  <p>讲理论，不讲推导</p>	<p>模型+策略+算法 (从概率角度)</p>  <p>经典，缺前沿</p>	<p>机器学习 (公理化角度)</p>  <p>神书（从贝叶斯角度）</p>	<p>PROBABILISTIC GRAPHICAL MODELS PRINCIPLES AND TECHNIQUES DAPHNE KELLER AND KIRI FRIEDMAN</p>  <p>花书：DL 圣经</p>
			 <p>科普，培养直觉</p>
			



机器学习与深度学习技术的学习材料

工程角度，
无需高等
数学背景



参数非参数
+频率贝叶
斯角度

讲理论，
不会讲推导

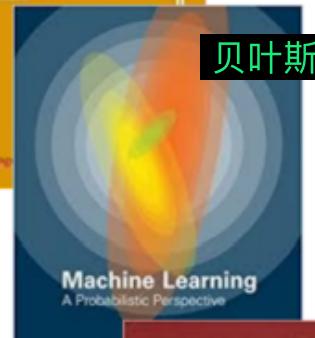
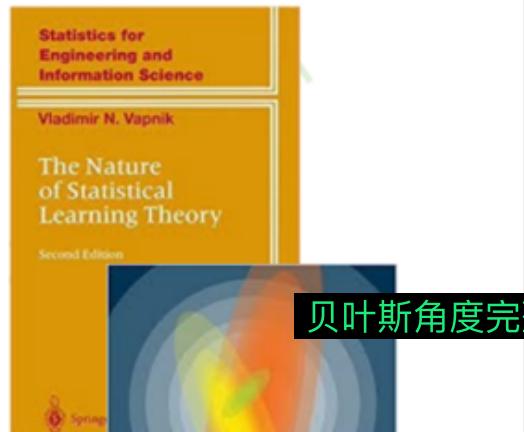


统计角度

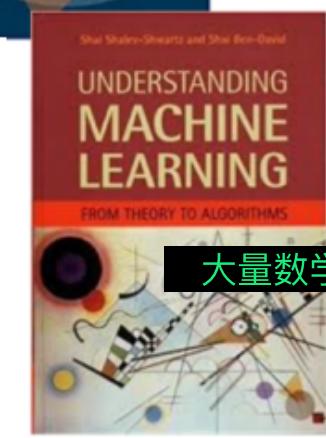
贝叶斯角度

DL 应用角度

统计方法集大成的书



贝叶斯角度完整介绍



大量数学推导



机器学习与深度学习技术的学习材料

优秀课程资源：

- CS231n (Stanford 李飞飞) / CS229 / CS230
- 吴恩达 (ML / DL ...)
- 李宏毅 (最佳中文课程, 没有之一)
- 邱锡鹏
- 李沐-动手学深度学习 (MXNet / PyTorch / TensorFlow)
- ... (多翻翻 Bilibili 就对了)

值得关注的公众号：

- 机器之心 (顶流)
- 量子位 (顶流)
- 新智元 (顶流)
- 专知 (偏学术)
- 微软亚洲研究院
- 将门创投
- 旷视研究院
- DeepTech 深科技 (麻省理工科技评论)
- 极市平台 (技术分享)
- ...



- 爱可可-爱生活 (微博、公众号、知乎、b站...)
- 陈光老师, 北京邮电大学PRIS模式识别实验室





深度学习技术的“不能”

- 算法输出不稳定，容易被“攻击”
- 模型复杂度高，难以纠错和调试
- 模型层级复合程度高，参数不透明
- 端到端训练方式对数据依赖性很强，模型增量性差
- 专注直观感知类问题，对开放性推理问题无能为力



Deep Learning's Bottlenecks

- Deep learning thus far
 - is data hungry
 - is shallow & has limited capacity for transfer
 - has no natural way to deal with hierarchical structure
 - has struggled with open-ended inference
 - is not sufficiently transparent
 - has not been well integrated with prior knowledge
 - cannot inherently distinguish causation from correlation
 - presumes a largely stable world
 - its answer often cannot be fully trusted
 - is difficult to engineer with

Deep Learning: A Critical Appraisal

Gary Marcus

Although deep learning has historical roots going back decades, neither the term "deep learning" nor the approach was popular just over five years ago, when the field was reignited by papers such as Krizhevsky, Sutskever and Hinton's now classic (2012) deep network model of Imagenet. What has the field discovered in the five subsequent years? Against a background of considerable progress in areas such as speech recognition, image recognition, and game playing, and considerable enthusiasm in the popular press, I present ten concerns for deep learning, and suggest that deep learning must be supplemented by other techniques if we are to reach artificial general intelligence.

Comments: 1 figure

Subjects: Artificial Intelligence (cs.AI); Machine Learning (cs.LG); Machine Learning (stat.ML)

MSC classes: 97R40

ACM classes: I.2.0; I.2.6

Cite as: [arXiv:1801.00631 \[cs.AI\]](https://arxiv.org/abs/1801.00631)

(or [arXiv:1801.00631v1 \[cs.AI\]](https://arxiv.org/abs/1801.00631v1) for this version)

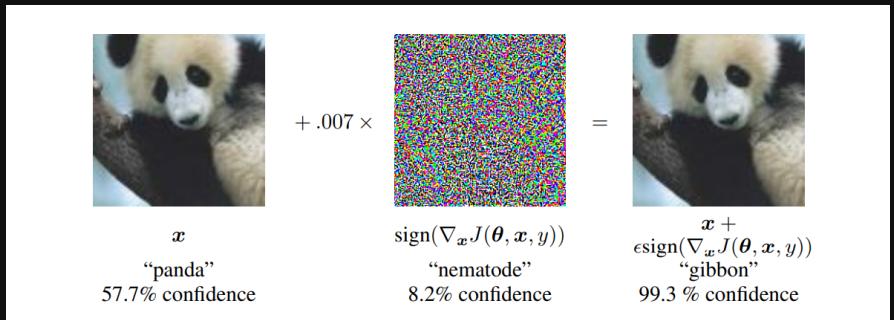
<https://doi.org/10.48550/arXiv.1801.00631> ⓘ

“Despite all of the problems I have sketched, I don't think that we need to abandon deep learning... Rather, we need to reconceptualize it: not as a universal solvent, but simply as one tool among many



深度学习技术的“不能” (1/n)

- 算法输出不稳定，容易被“攻击”



arXiv:1312.6199



Su J, Vargas D V, Sakurai K.
One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. (2019)

Jadhav et al. 2306.11797

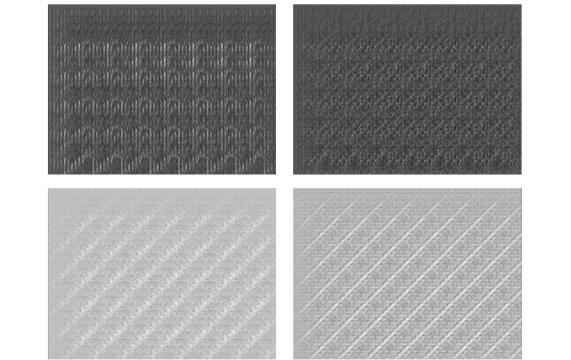
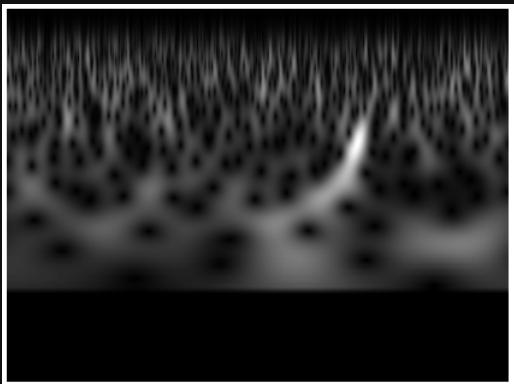


FIG. 10. Examples generated by performing adversarial attacks on the fine-tuned classifier model. Despite having no discernible CBC features, the classifier identified all of these examples as CBC signals with $P_{\text{cbc}} > 99\%$.

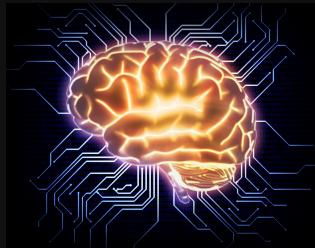
VS





深度学习技术的“不能” (2/n)

- 模型复杂度高，难以纠错和调试



大众眼中的我们



工程师眼中的我们



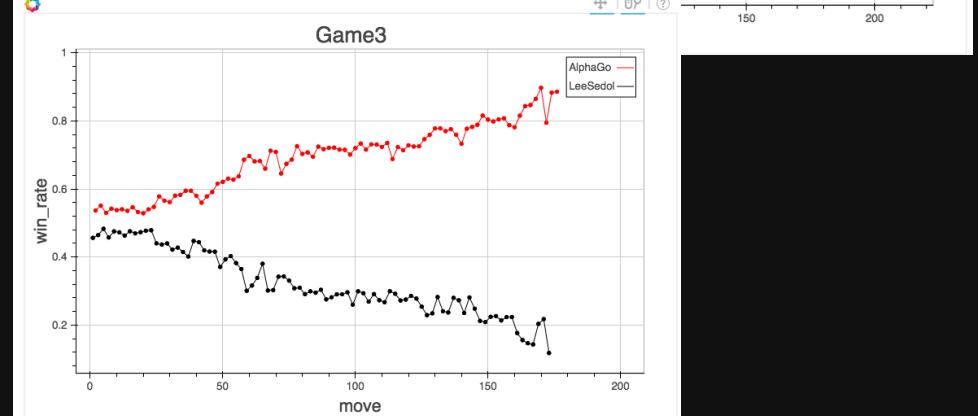
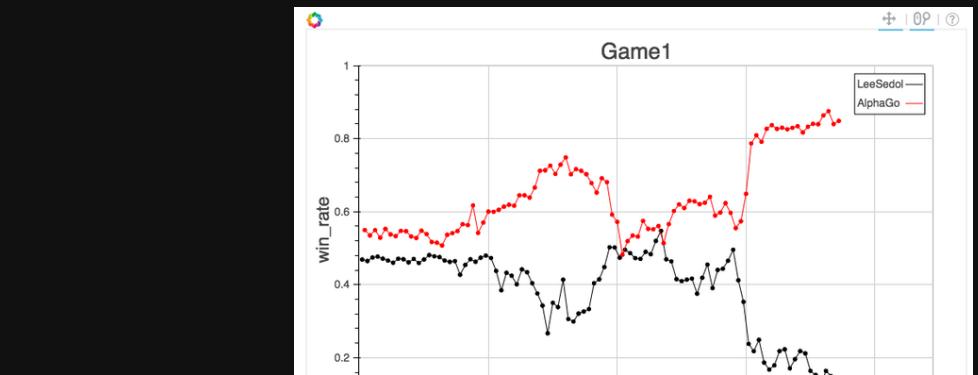
数学家眼中的我们



我们眼中的自己



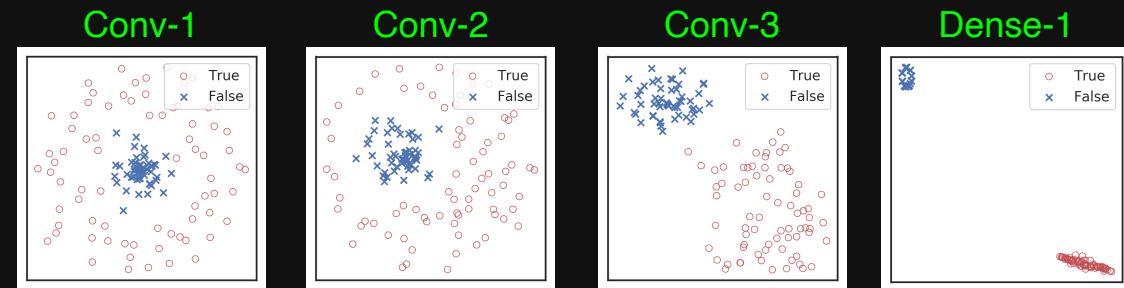
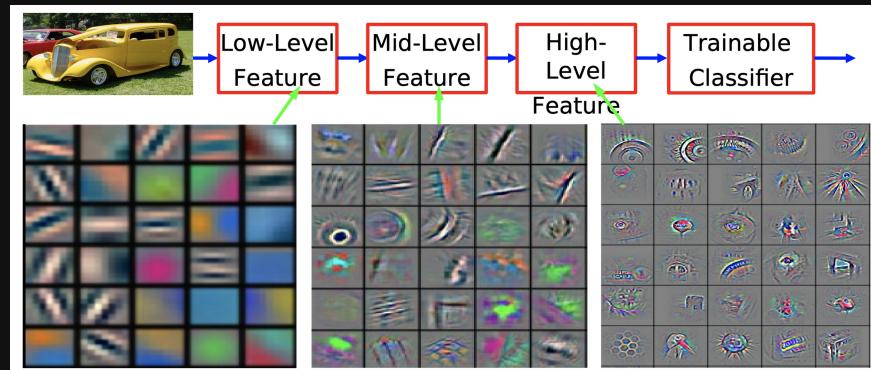
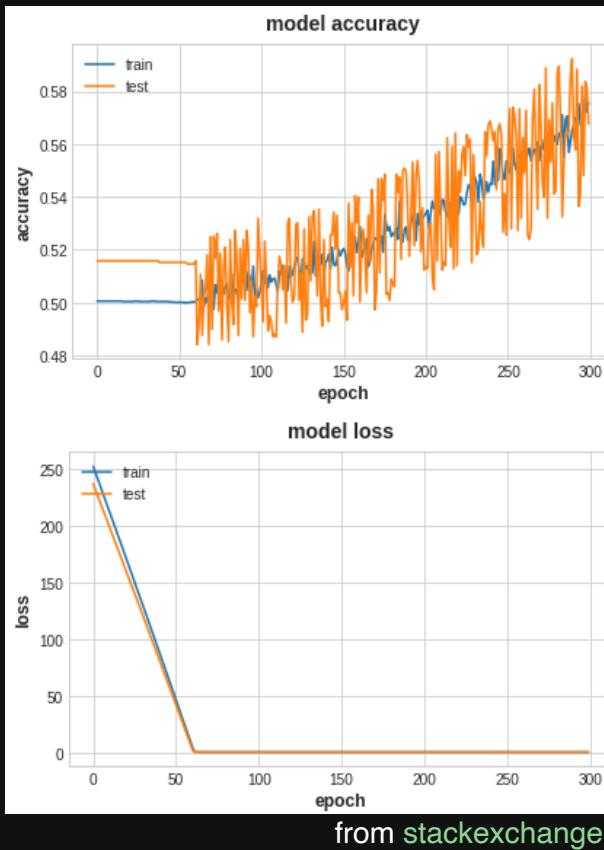
实际的我们



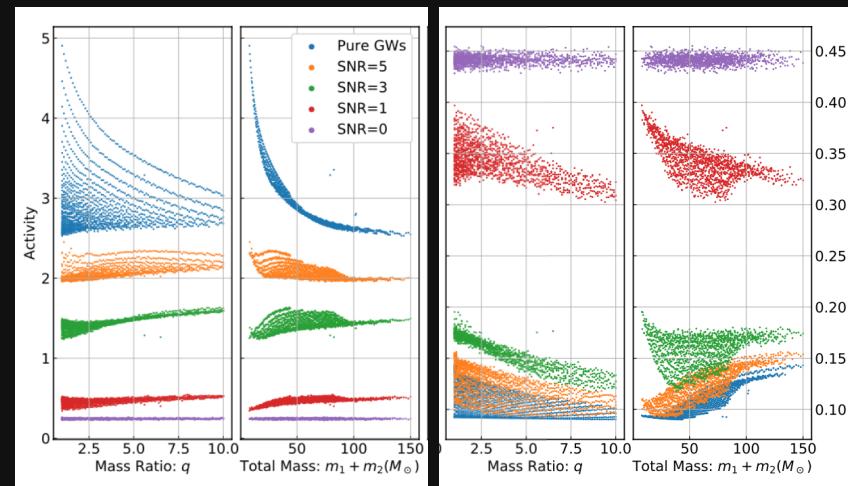


深度学习技术的“不能” (3/n)

- 模型层级复合程度高，参数不透明



on the top
activated
neurons





深度学习技术的“不能” (4/n)

- 端到端训练方式对数据依赖性很强，模型增量性差
 - 当样本**数据量**小的时候，深度学习无法体现强大拟合能力
 - 模型**有效容量 (effective capacity)** 的上界：参数 / VC 维/ Rademacher 复杂度

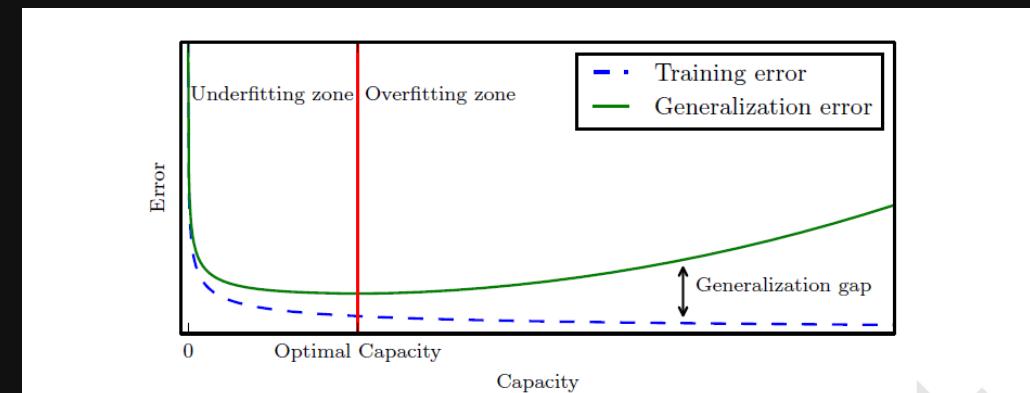
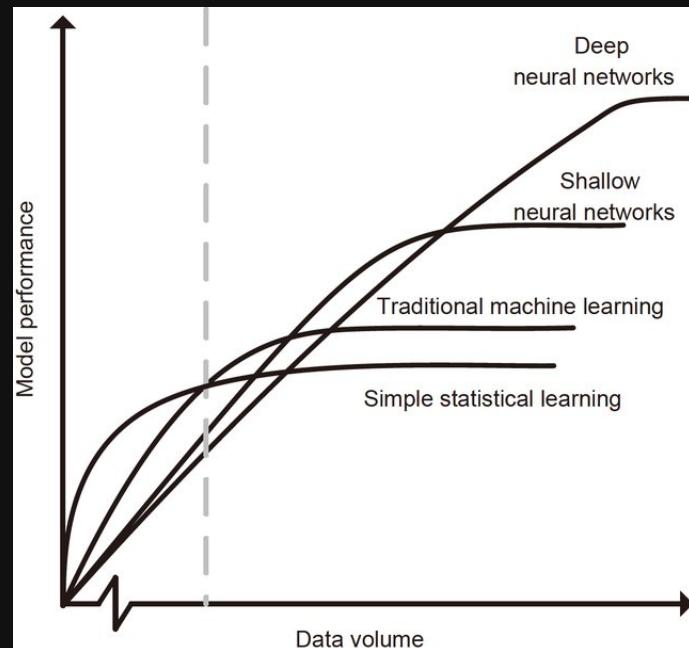


图 5.3: 容量和误差之间的典型关系。训练误差和测试误差表现得非常不同。在图的左端，训练误差和泛化误差都非常高。这是欠拟合期 (underfitting regime)。当我们增加容量时，训练误差减小，但是训练误差和泛化误差之间的间距却不断扩大。最终，这个间距的大小超过了训练误差的下降，我们进入到了过拟合期 (overfitting regime)，其中容量过大，超过了最佳容量 (optimal capacity)。



深度学习技术的“不能” (4/n)

- 专注直观感知类问题，对开放性推理问题无能为力
 - 乌鸦给我们的启示：
 - 完全自主的智能
 - 大数据非必须
 - 海量计算非必须

请点击下图中所有的“好”男人

请找出图中所有的纯天然女星

微信号: KABCclub



Article Not peer-reviewed version

GPT-4 Can't Reason

Konstantine Arkoudas *

Posted Date: 2 August 2023

doi: 10.20944/preprints202308.0148.v1

Keywords: GPT-4; LLM; AI; reasoning; inference

KA: Select two random numbers between 1381 and 1453 and multiply them together, reporting the result.

GPT-4: Sure, let's select two random numbers in the range of 1381 to 1453. Let's say these numbers are 1405 and 1421.

To get the product, we simply multiply these two numbers together:
 $1405 * 1421 = 1996025$

新智元



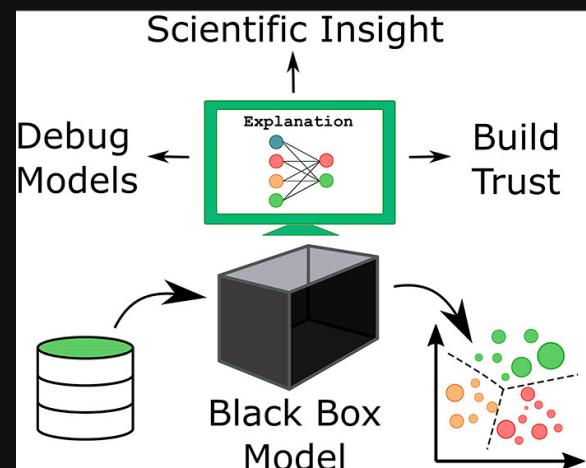
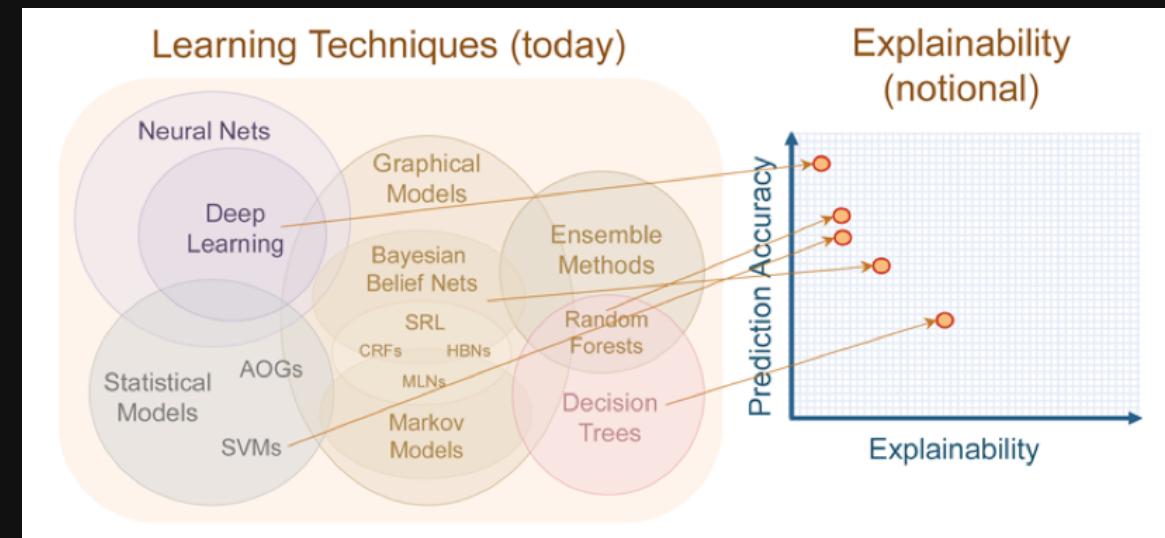
深度学习技术的“不能”与解释性

深度学习的“不能”

- 稳定性低
- 可调试性差
- 参数不透明
- 机器偏见
- 增量性差
- 推理能力差

解释性的三个层次

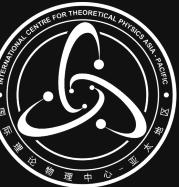
- “对症下药” (找得到)
知道那些特征输出有重要影响,
出了问题准确快速纠错
- 不再“对牛弹琴” (看得懂)
双向: 算法能被人的知识体系理解
+利用和结合人类知识
- “站在巨人的肩膀上” (留得下)
知识得到有效存储、积累和复用
→ 越学越聪明





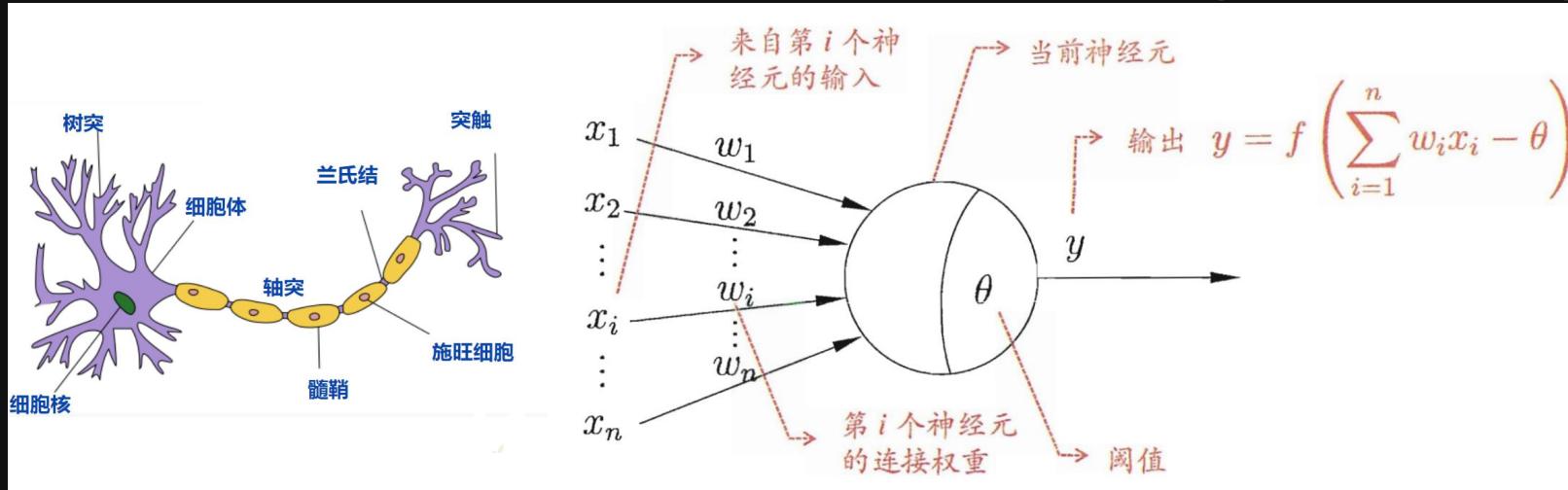
深度学习：神经网络基础

- 神经元
- 万有逼近定理
- 神经网络的参数学习



深度学习：神经网络基础

- 神经元

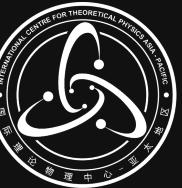


- 激活函数 f

- 没有激活函数的话，相当于一维矩阵相乘：
 - 多层和一层一样
 - 只能拟合线性函数

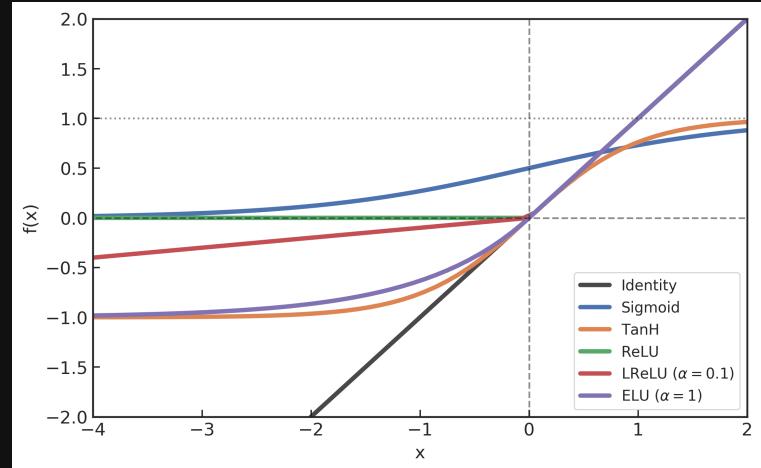
$$\sum_i w_i x_i + b = w_1 x_1 + \cdots + w_D x_D + b$$

$$\underbrace{\left[\sum_i w_i x_i + b \right]}_{1 \times 1} = \underbrace{\left[\cdots \ x_i \ \cdots \right]}_{1 \times D} \cdot \underbrace{\left[\begin{array}{c} \vdots \\ w_i \\ \vdots \end{array} \right]}_{D \times 1} + \underbrace{\left[b \right]}_{1 \times 1}$$



深度学习：神经网络基础

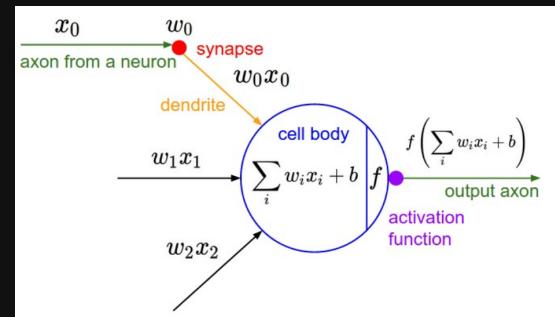
- 激活函数 f 举例
 - S 性函数 (sigmoid)
 - ReLU 修正线性单元
 - 双性 S 性函数 (tanh)
 - Leaky ReLU
 - ELU 指数线性单元
 - ...



Name	Equation	Derivative
Identity	$f(x) = x$	$f'(x) = 1$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH	$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (ReLU)	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Leaky ReLU		
Parameteric ReLU	$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)	$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$



深度学习：神经网络基础



- 一个神经元
 - Input: 一个样本
 - Input: N 个样本

$$\underbrace{\left[\sum_i w_i x_i + b \right]}_{1 \times 1} = \underbrace{\left[\cdots x_i \cdots \right]}_{1 \times D} \cdot \underbrace{\left[\begin{array}{c} \vdots \\ w_i \\ \vdots \end{array} \right]}_{D \times 1} + \underbrace{\left[b \right]}_{1 \times 1}$$

$$\underbrace{\left[\begin{array}{c} \vdots \\ \sum_j w_j x_{ij} + b_j \end{array} \right]}_{N \times 1} = \underbrace{\left[\cdots x_{ij} \cdots \right]}_{N \times D} \cdot \underbrace{\left[\begin{array}{c} \vdots \\ w_j \\ \vdots \end{array} \right]}_{D \times 1} + \underbrace{\left[\begin{array}{c} \vdots \\ b_j \\ \vdots \end{array} \right]}_{N \times 1}$$

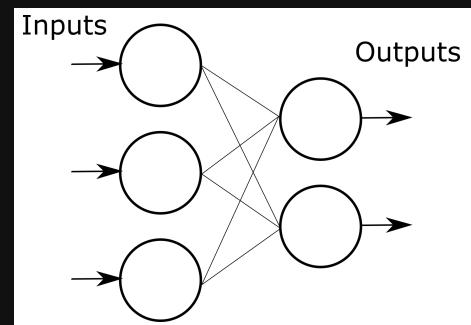
- 线性矩阵操作之后，会经过激活函数实现元素级操作，使得神经元“非线性化”。

$$\hat{x}_i = f(\sum_i w_i x_i + b) = \max(0, \sum_i w_i x_i + b)$$



深度学习：神经网络基础

- M 个神经元
 - Input: 一个样本



$$\underbrace{\left[\sum_i w_{ij} x_i + b \quad \dots \right]}_{1 \times M} = \underbrace{\left[\dots \quad x_i \quad \dots \right]}_{1 \times D} \cdot \underbrace{\left[\begin{array}{c:c:c} \vdots & w_{ij} & \dots \\ \dots & \vdots & \dots \\ \vdots & \vdots & \dots \end{array} \right]}_{D \times M} + \underbrace{\left[b \quad \dots \right]}_{\substack{1 \times M \\ \text{Broadcasting}}}$$

- Input: N 个样本 (with activation function)

$$\underbrace{\left[\begin{array}{c:c:c} \vdots & \hat{x}_{ik} & \dots \\ \dots & \vdots & \dots \\ \vdots & \vdots & \dots \end{array} \right]}_{N \times M} = f \left(\underbrace{\left[\begin{array}{c:c:c} \vdots & x_{ij} & \dots \\ \dots & \vdots & \dots \\ \vdots & \vdots & \dots \end{array} \right]}_{N \times D} \cdot \underbrace{\left[\begin{array}{c:c:c} \vdots & w_{jk} & \dots \\ \dots & \vdots & \dots \\ \vdots & \vdots & \dots \end{array} \right]}_{D \times M} + \underbrace{\left[\begin{array}{c:c:c} \vdots & b_i & \dots \\ \dots & \vdots & \dots \\ \vdots & \vdots & \dots \end{array} \right]}_{\substack{N \times M \\ \text{Broadcasting}}} \right)$$

- 留意：
 - 数据矩阵的行（样本数）、列（特征维度）
 - 一个隐层的行（对应于数据特征维度）、列（神经元的个数）
 - 每一层的非线性映射过程中，输入输出的数据矩阵行（样本数）保持不变
 - 过参数化 (Over-parameterization) 的神经网络

$f \sim$ non-linear operation



深度学习：神经网络基础

- **万有逼近定理 (Universal Approximation Theorem)**

- 只要函数 $y = \varphi(x)$ 是连续的，就存在神经网络以任意精度逼近它。
- 如果一个隐层包含足够多的神经元，三层前馈神经网络（输入-隐层-输出）能以任意精度逼近任意预定的连续函数。

定理1：(万有逼近定理) 设函数 $y = \varphi(x)$ 在区间 $[a, b]$ 上连续。则 $\forall \varepsilon > 0$ ，存在充分大的 $n \in \mathbb{N}^*$, $N \in \mathbb{N}^*$ ，及实数 a_i, b_i , $i = 1, 2, \dots, N$ ，使得函数

$$g_n(x) = \sum_{i=1}^N (b_i \cdot S_n(x; a_i))$$

满足

$$\int_a^b |g_n(x) - \varphi(x)| dx < \varepsilon$$

[Hornik et al., 1989]

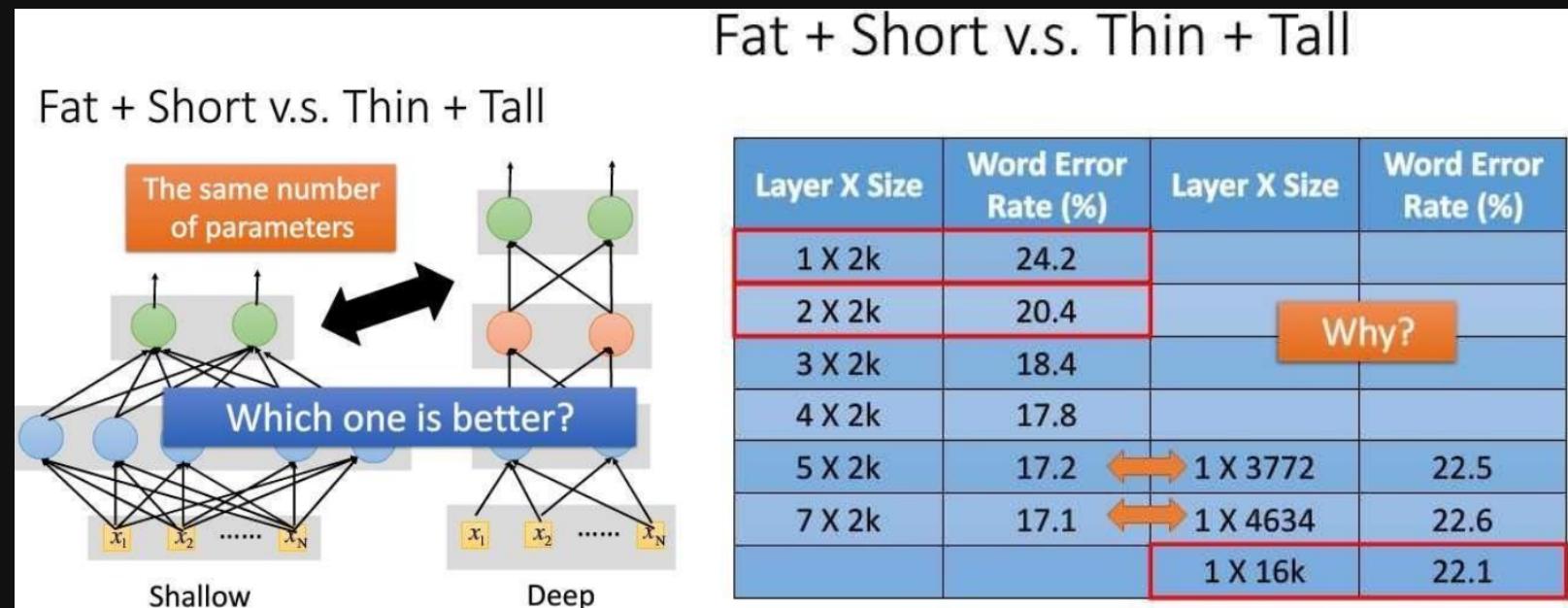
- 注意：

- 上述定理只是给出了**存在性结论**，实际应用时 n, N 可能非常大，导致运算规模异常庞大。
(70年代低谷)
- 神经网络相当于解决了最小二乘法拟合数据时“**如何选取函数型**”这一本质难点。但是因为参数过多，从神经网络中很难反映出数据背后的机理，所以**不适用于机理建模**。



深度学习：神经网络基础

- 更宽还是更深？**更深！**
- 在神经元总数相当的情况下，增加网络深度可以比增加宽度带来更强的**网络表示能力**。
- 深度和宽度对**函数复杂度**的贡献是不同的，深度的贡献是指数增长的，而宽度的贡献是线性的。



Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks[C] Interspeech. 2011.



深度学习：神经网络基础

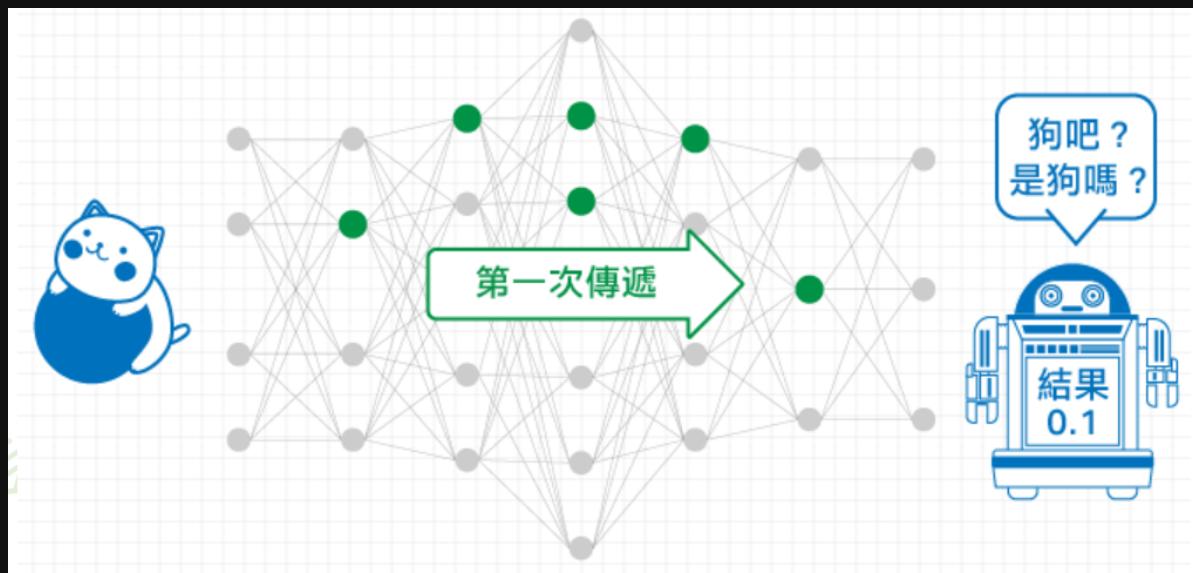
- 神经网络可视化 (<https://playground.tensorflow.org/>)

<https://playground.tensorflow.org/>

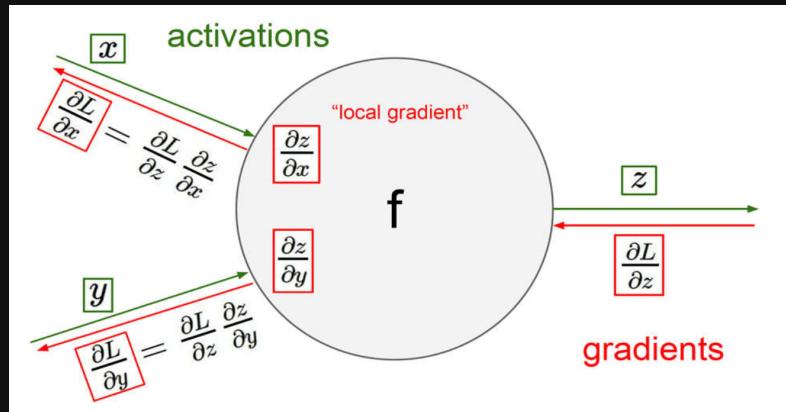


深度学习：神经网络基础

- 神经网络的参数学习：误差反向传播
 - 多层神经网络可看成是一个复合的非线性多元函数 $F(\cdot) : X \rightarrow Y$
- $$F_w(x) = f_n (\dots f_3 (f_2 (f_1(x) * \theta_1 + b) * \theta_2 + b) \dots)$$
- 给定训练数据 $\{x^i, y^i\}_{i=1:N}$, 希望损失 $\sum_i \text{loss}(F_w(x^i), y^i)$ 尽可能小.



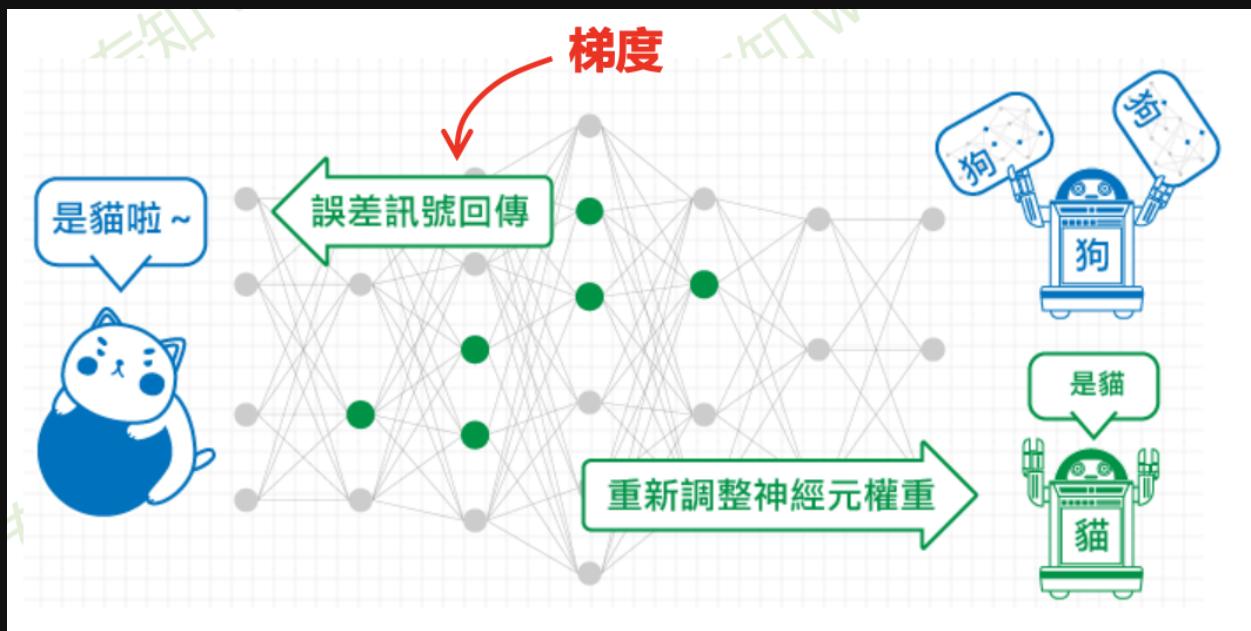
图片取自李宏毅老师《机器学习》课程



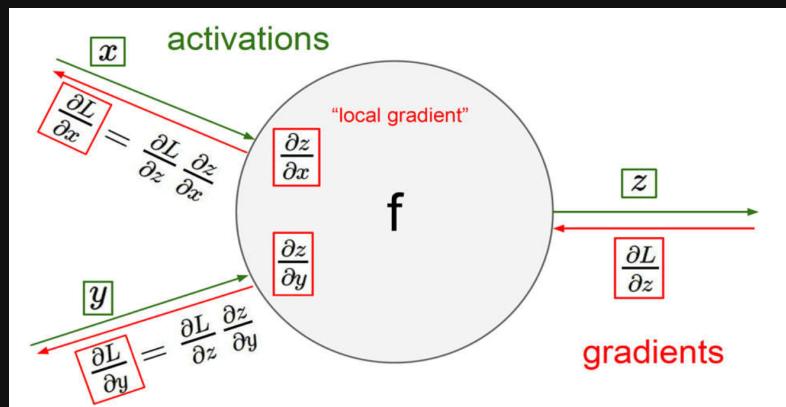


深度学习：神经网络基础

- 神经网络的参数学习：误差反向传播
 - 多层神经网络可看成是一个复合的非线性多元函数 $F(\cdot) : X \rightarrow Y$
- $$F_w(x) = f_n (\dots f_3 (f_2 (f_1(x) * \theta_1 + b) * \theta_2 + b) \dots)$$
- 给定训练数据 $\{x^i, y^i\}_{i=1:N}$, 希望损失 $\sum_i \text{loss}(F_w(x^i), y^i)$ 尽可能小.



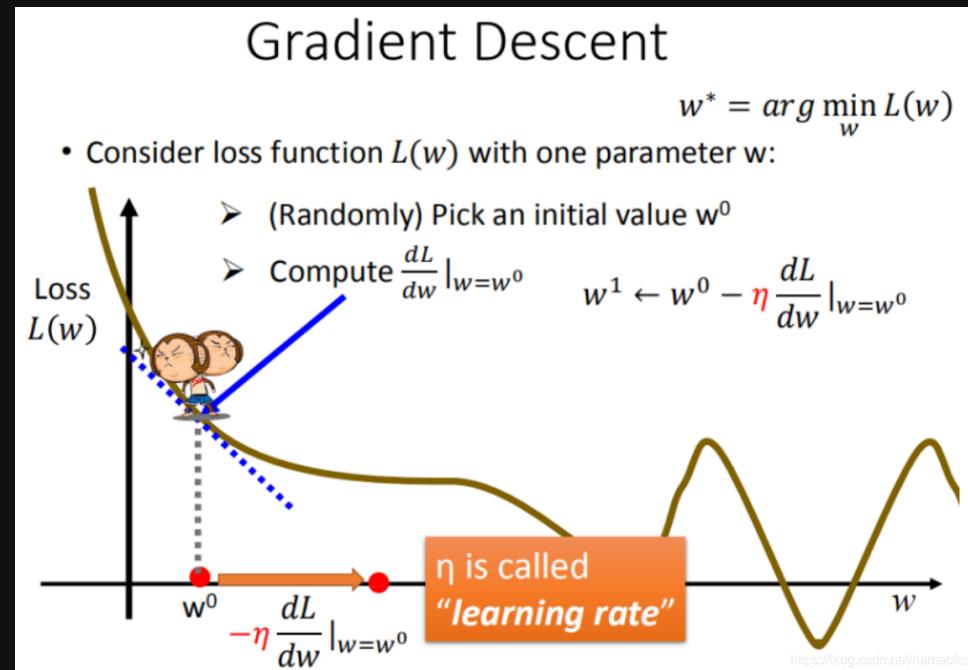
图片取自李宏毅老师《机器学习》课程



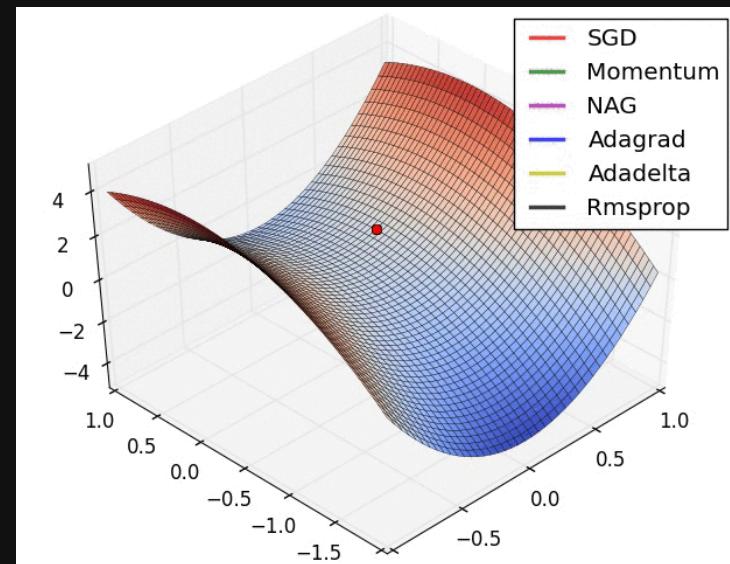


深度学习：神经网络基础

- 神经网络的参数学习：误差反向传播
 - 多层神经网络可看成是一个复合的非线性多元函数 $F(\cdot) : X \rightarrow Y$
- $$F_w(x) = f_n (\dots f_3 (f_2 (f_1(x) * \theta_1 + b) * \theta_2 + b) \dots)$$
- 给定训练数据 $\{x^i, y^i\}_{i=1:N}$, 希望损失 $\sum_i \text{loss}(F_w(x^i), y^i)$ 尽可能小
- 反向传播算法 (BP) 的目标是找损失函数关于神经网络中可学习参数 (w) 的偏导数 (证明略)



- 优化算法的选择 (略)



深度学习：神经网络基础



“Talk is cheap. Show me the code.”

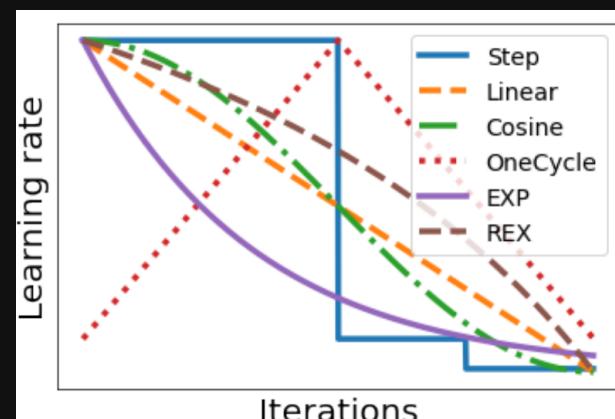
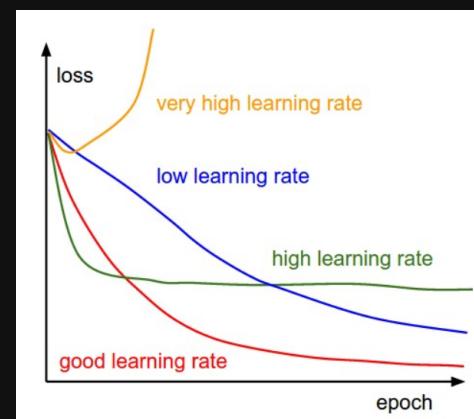
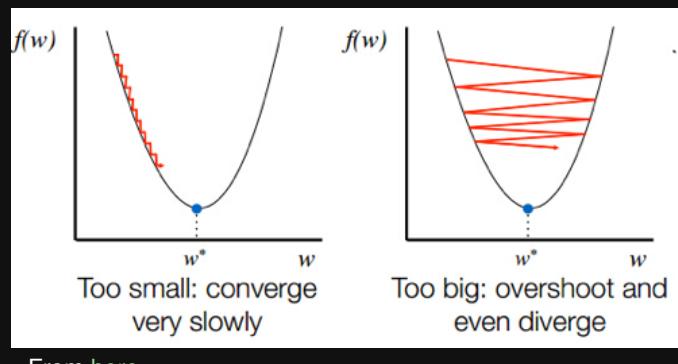
Linus Torvalds



- 神经网络的参数学习：误差反向传播
 - 多层神经网络可看成是一个复合的非线性多元函数 $F(\cdot) : X \rightarrow Y$

$$F_w(x) = f_n (\dots f_3 (f_2 (f_1(x) * \theta_1 + b) * \theta_2 + b) \dots)$$

- 给定训练数据 $\{x^i, y^i\}_{i=1:N}$, 希望损失 $\sum_i \text{loss}(F_w(x^i), y^i)$ 尽可能小
- 反向传播算法 (BP) 的目标是找损失函数关于神经网络中可学习参数 (w) 的偏导数 (证明略)
- 学习率 η 与学习率策略





模型性能评估与测试调优

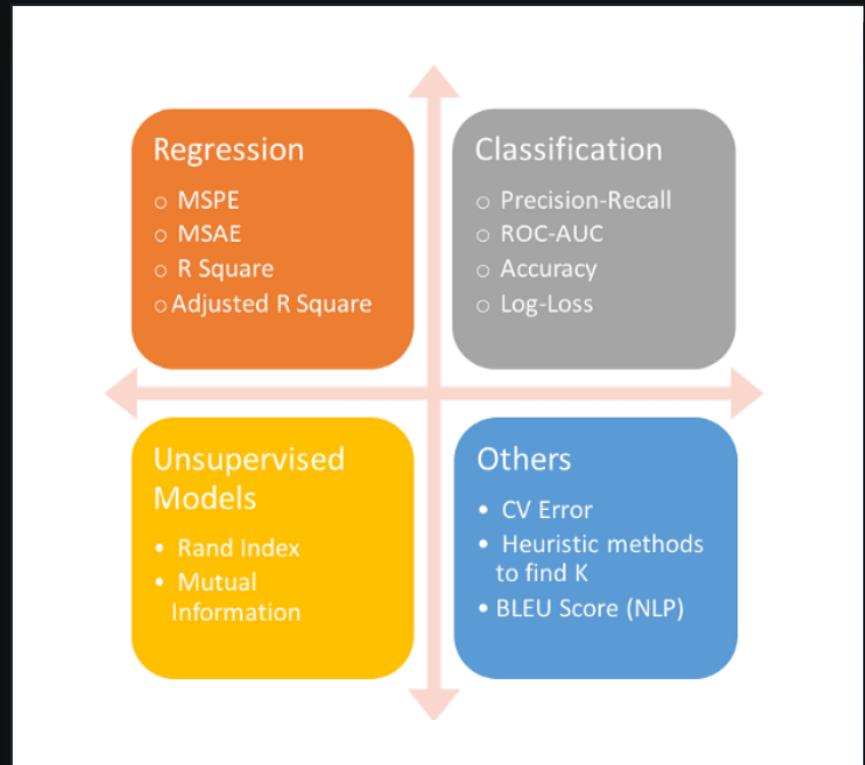
- 分类任务的评价指标
- 模型调优, 过拟合与欠拟合
- 没有免费午餐定理 (No free lunch theorem)



模型性能评估与测试调优

分类任务的评价指标

- 评价指标
 - 评价指标的选择会影响如何测量和比较机器学习算法的性能，也会影响我们在如何权衡结果中不同特征的重要性以及您选择哪种算法的最终选择。
 - 使用不同的性能度量往往会导致不同的评判结果。
- 模型的 泛化性
 - 机器学习模型的学习目标是从目标领域内的训练数据到任意其他数据上的性能良好，由此可以在未来对模型没有见过的数据进行预测。





模型性能评估与测试调优

分类任务的评价指标

- **准确率 (Accuracy)** : 也就是正确分类的样本数占总样本数的比例。
但这个指标对于不均衡数据而言，模型会有掉入“**高准确率陷阱**”。
- 举个例子：
 - 如果有一种癌症，1000个人中只有1个人会得，也就是患这个癌症的概率为0.1%。那么这个时候，我们不用机器学习，给我1000个人预测是否患癌，我只要全部猜没有，那么我就只会有1个人判错，我的准确率达到了99.9%。
 - 那么如果我们用机器学习来训练出一个预测一个人是否患有这个癌症的模型，就算这个模型最后的准确率达到了98%，那也是没有意义的。



		预测	
		是	否
实际	是	0	1
	否	0	999

混淆矩阵



模型性能评估与测试调优

分类任务的评价指标

- 混淆矩阵（Confusion matrix）：样本的真实分类值作为一个维度，把样本预测分类值作为一个维度。
 - 真正例：预测为正，实际也为正。
 - 真反例：预测为反，实际也为反。
 - 假正例：预测为反，但实际为正。
 - 假反例：预测为正，但实际为反。

		预测	
		正	反
实际	正	真正例	假正例
	反	假反例	真反例

混淆矩阵



模型性能评估与测试调优

分类任务的评价指标

- **精确率 (Precision)** : 在所有预测的正类的样本中，预测正确的样本所占有的比例。

真正例

$$\cdot \text{ 精确率} = \frac{\text{真正例}}{\text{真正例} + \text{假反例}}$$

- **召回率 (Recall)** : 在所有真实类别为正类的样本中，被正确预测为正的样本所占的比例。

真正例

$$\cdot \text{ 召回率} = \frac{\text{真正例}}{\text{真正例} + \text{假正例}}$$

- 精确率 (查准率) 评估预测的 **准不准**；
- 召回率 (查全率) 评估找的 **全不全**。

		预测	
		正	反
实际	正	真正例	假正例
	反	假反例	真反例

混淆矩阵



模型性能评估与测试调优

分类任务的评价指标

- **精确率 (Precision)** : 在所有预测的正类的样本中，预测正确的样本所占有的比例。

$$\text{精确率} = \frac{\text{真正例}}{\text{真正例} + \text{假反例}} = 80\%$$

- **召回率 (Recall)** : 在所有真实类别为正类的样本中，被正确预测为正的样本所占的比例。

$$\text{召回率} = \frac{\text{真正例}}{\text{真正例} + \text{假正例}} = 80\%$$

- 精确率 (查准率) 评估预测的 **准不准**；
- 召回率 (查全率) 评估找的 **全不全**。

		预测	
		正	反
实际	正	8	2
	反	2	988

混淆矩阵



模型性能评估与测试调优

分类任务的评价指标

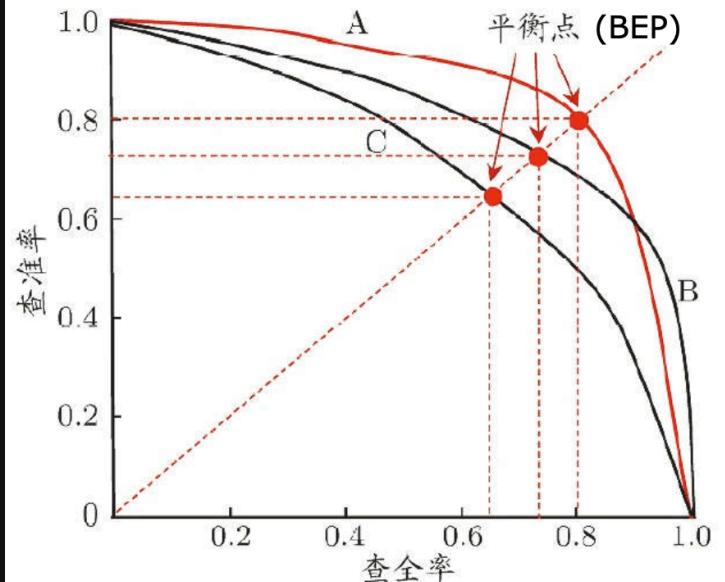
- F1 调和平均
 - 比 BEP 更常用的 F1 度量:

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$
 - F1：查准率与查全率的调和平均，调和平均更注重较小值的影响
 - 若对查准率/查全率有不同偏好：

$$F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

 $\beta > 1$ 时查全率有更大影响； $\beta < 1$ 时查准率有更大影响
 - F_β ：查准率与查全率的加权调和平均
- ROC, AUC (下一讲)

根据学习器的预测结果按正例可能性大小对样例进行排序，并逐个把样本作为正例进行预测



PR图：

- 学习器 A 优于 学习器 C
- 学习器 B 优于 学习器 C
- 学习器 A ?? 学习器 B

BEP:

- 学习器 A 优于 学习器 B
- 学习器 A 优于 学习器 C
- 学习器 B 优于 学习器 C



“Talk is cheap. Show me the code.”

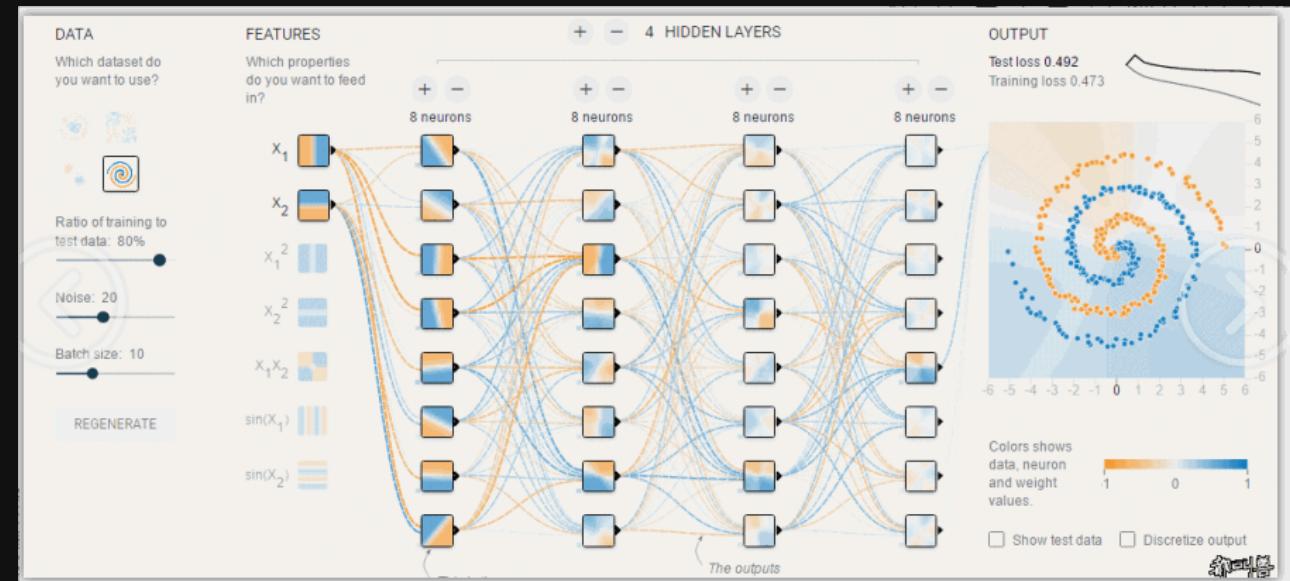
Linus Torvalds



模型性能评估与测试调优

模型调优，过拟合与欠拟合

- 调参过程相似：先产生若干模型，然后基于某种评估。
 - 算法的参数：一般由人工设定，亦称“超参数”
 - 模型的参数：一般由学习确定
- 参数调得好不好，往往对最终性能有关键影响。

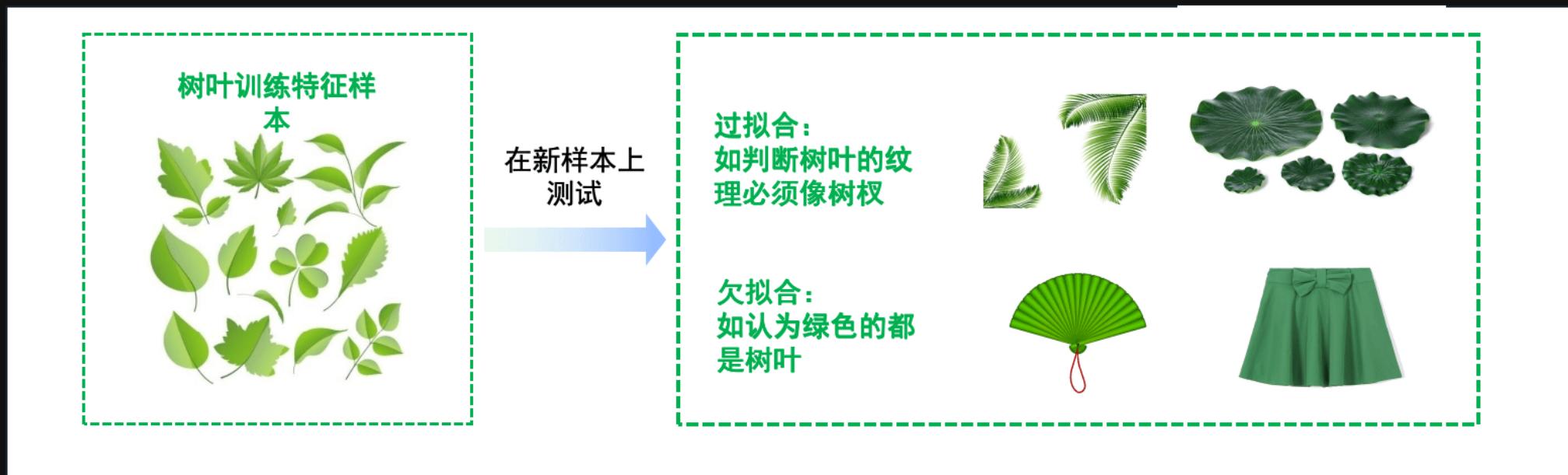




模型性能评估与测试调优

模型调优，过拟合与欠拟合

- 模型泛化性的评价：
 - 过拟合 (over-fitting)：在训练数据上表现良好，在未知数据上表现差。
 - 欠拟合 (under-fitting)：在训练数据和未知数据上表现都很差。
 - 解决办法：重新选数据，重新定模型

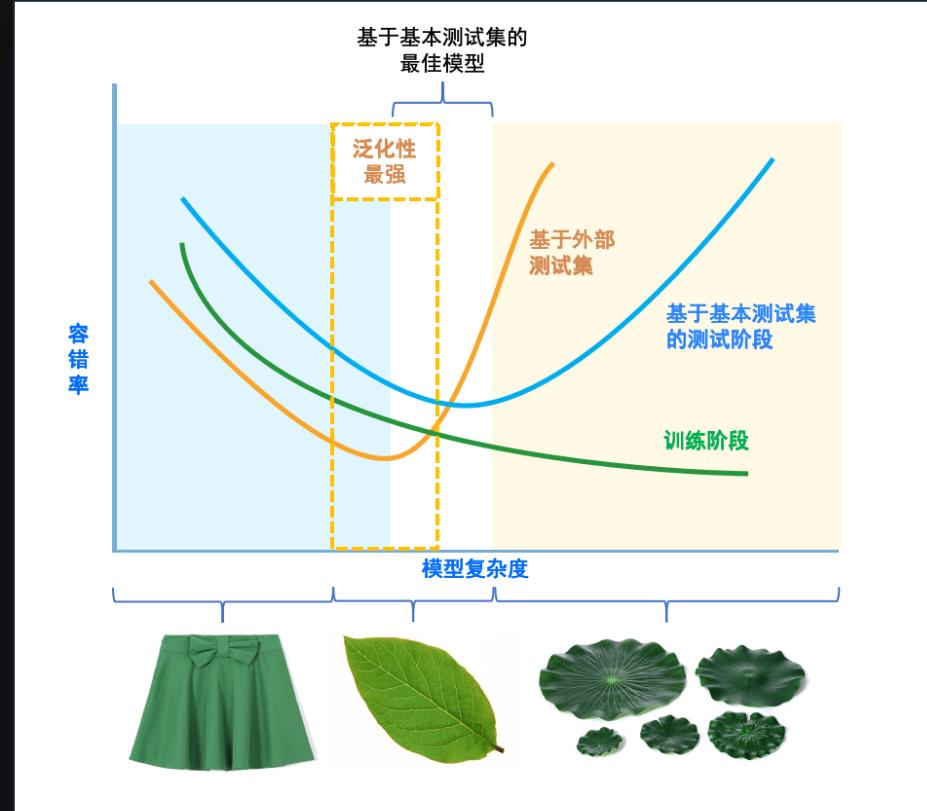
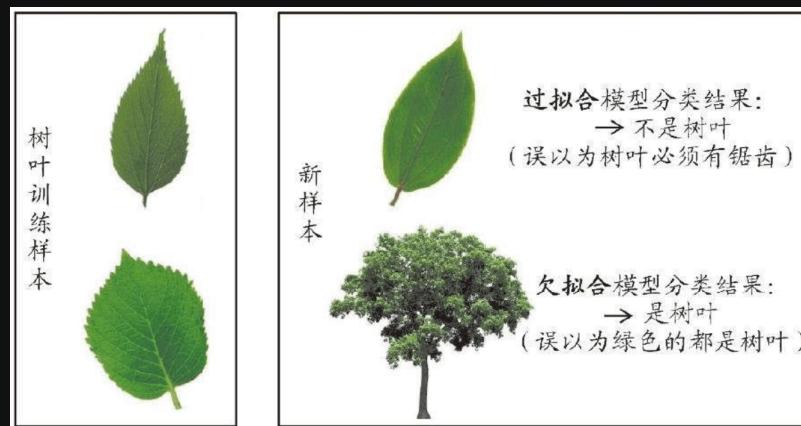




模型性能评估与测试调优

模型调优，过拟合与欠拟合

- 讨论机器学习模型学习和泛化的好坏时，通常使用术语：过拟合和欠拟合。
- 模型泛化性的评价：
 - 过拟合 (over-fitting)**：在训练数据上表现良好，在未知数据上表现差。
 - 欠拟合 (under-fitting)**：在训练数据和未知数据上表现都很差。
 - 解决办法：**重新选数据，重新定模型
- 模型怎么定？
 - 不同模型复杂度在评价指标上的表现



素材来源: DOI: 10.1177/2374289519873088

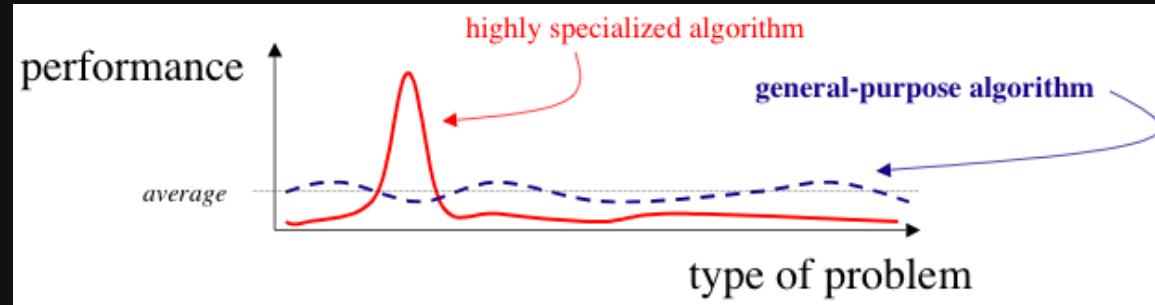


模型性能评估与测试调优

没有免费午餐定理 (No free lunch theorem)

- 对于所有可能的域 (所有可能的问题实例均来自均匀的概率分布) , 算法A和B的平均性能相同。

<https://medium.com/@LeonFedden/the-no-free-lunch-theorem-62ae2c3ed10c>



- 这是因为几乎所有非死记硬背的(non-rote)机器学习算法或统计模型都需要对预测变量和目标变量之间的关系做出了一些假设, 从而将 **偏差 (bias)** 引入了模型, 具体称为 **归纳或学习偏差 (inductive or learning bias)** 。
- 无偏差学习是徒劳的, 因为没有先验假设的学习者在提供新的, 看不见的输入数据时将没有合理的基础来创建估计。
- 这些假设使得某些算法在某些数据集上表现优秀, 而在其他数据集上表现不佳。换句话说, 一个算法的有效性取决于它的偏差 (即假设) 与数据的真实性质之间的匹配程度。这就意味着, **对于任何给定的算法, 总会存在一些它无法有效处理的数据集**。
- 算法的假设适用于某些数据集, 但不适用于其他数据集。该现象对于理解欠拟合 (underfitting) 的概念 和 偏差/方差折衷 (bias/variance tradeoff) 至关重要。

Wolpert D H. The lack of a priori distinctions between learning algorithms[J]. Neural computation, 1996, 8(7): 1341-1390.

没有免费午餐理论对于个人的指导

- 在依赖模型或搜索算法之前, 请始终检查您的假设。
- 没有“超级算法”能完美适用于所有数据集。





模型性能评估与测试调优

偏差-方差窘境 (bias-variance dilemma)

- 一般而言，偏差与方差存在冲突：
 - 训练不足时，学习器拟合学习能力不强，偏差主导
 - 随着训练程度加深，学习器拟合能力逐渐增强，方差逐渐主导
 - 训练充足后，学习器的拟合能力很强，方差主导

泛化性能 是由学习算法的能力、数据的充分性以及学习任务本身的难度共同决定。

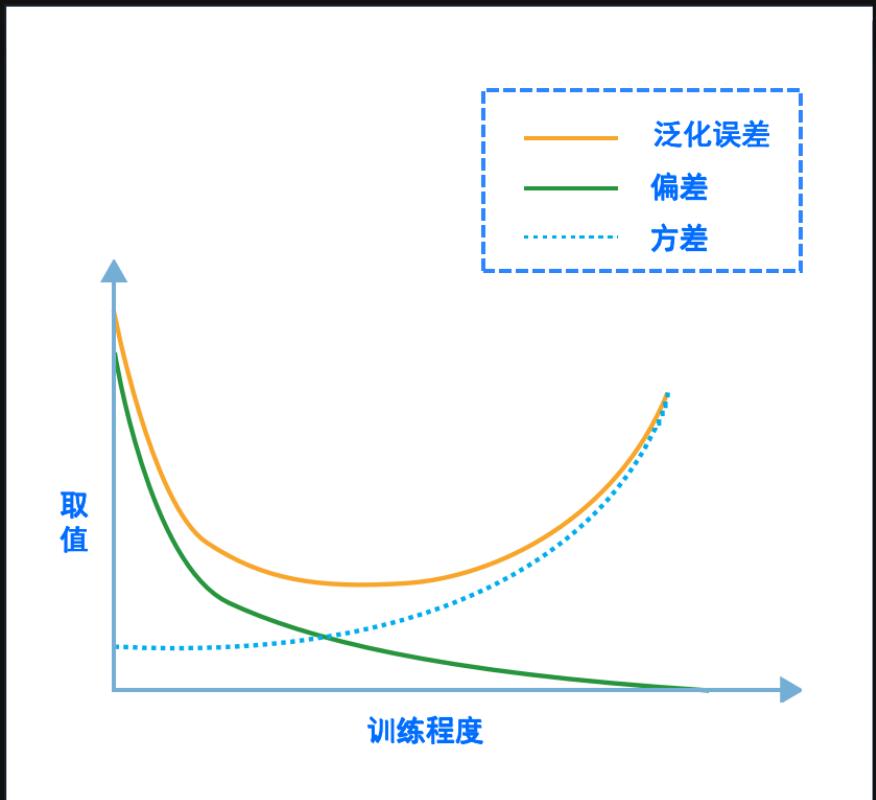
对回归任务，泛化误差可通过“偏差-方差分解”拆解为：

$$E(f; D) = \underbrace{\text{bias}^2(\mathbf{x})}_{\text{期望输出与真实输出的差别}} + \underbrace{\text{var}(\mathbf{x})}_{\substack{\text{同样大小的训练集的变动, 所导致的} \\ \text{性能变化}}} + \underbrace{\varepsilon^2}_{\substack{\text{训练样本的标记与} \\ \text{真实标记有区别}}}$$

$$\text{bias}^2(\mathbf{x}) = (\bar{f}(\mathbf{x}) - y)^2$$

$$\text{var}(\mathbf{x}) = \mathbb{E}_D \left[(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2 \right]$$

$$\varepsilon^2 = \mathbb{E}_D \left[(y_D - y)^2 \right]$$





模型性能评估与测试调优

模型调优，过拟合与欠拟合

过拟合和欠拟合是机器学习中常见的两种问题。

- 过拟合：当模型在训练数据上表现得过于优秀，但在测试数据或新数据上表现不佳时，我们称模型出现了过拟合。过拟合的模型过于复杂，以至于它甚至学习了训练数据中的噪声。在图表中，过拟合通常表现为训练误差持续降低，但验证误差开始上升。
- 解决过拟合的方法包括：
 - **增加数据量**：更多的数据可以帮助模型学习到更多的信息，减少过拟合的可能性。
 - **正则化**：正则化是一种添加惩罚项的技术，可以防止模型的权重过大，从而降低模型复杂度。
 - **早停**：在验证误差开始上升时停止训练，可以防止模型过度学习训练数据。
 - **降低模型复杂度**：简化模型，如减少神经网络的层数或神经元数量，可以降低模型的复杂度，减少过拟合的可能性。
 - ...
- 欠拟合：当模型在训练数据和测试数据上的表现都不佳时，我们称模型出现了欠拟合。欠拟合的模型过于简单，无法捕捉到数据中的模式。在图表中，欠拟合表现为训练误差和验证误差都很高。
- 解决欠拟合的方法包括：
 - **增加模型复杂度**：增加更多的特征，或者使用更复杂的模型，如增加神经网络的层数或神经元数量，可以帮助模型捕捉到更复杂的模式。
 - **减少正则化**：如果模型过于简单，可能是正则化过度，可以尝试减少正则化的程度。
 - **更换模型**：如果当前模型无法很好地拟合数据，可以尝试更换其他类型的模型。
 - ...

	欠拟合	恰到好处！	过拟合
特点	<ul style="list-style-type: none"> • 高训练误差 • 训练误差与测试误差接近 • 高偏差 	训练误差略微低于测试误差	<ul style="list-style-type: none"> • 低训练误差 • 训练误差远低于测试误差 • 高方差
案例			
深度学习建模过程			
可选的解决办法	<ul style="list-style-type: none"> • 模型复杂化 • 引入更多特征 • 延长训练时间 		<ul style="list-style-type: none"> • 模型“惩罚”（正则化） • 获取更多数据



模型性能评估与测试调优

模型评估与选择

- 比较检验：在某种度量下取得评估结果后，是否可以直接比较以评判优劣？
 - No! 因为：
 - 测试性能不等于泛化性能
 - 测试性能随着测试集的变化而变化
 - 很多机器学习算法本身有一定的随机性
- 机器学习任务 → “概率近似正确”
- 统计假设检验 (hypothesis test) 为学习器性能比较提供了重要依据 【应需要有统计显著性作为评判依据】
 - 两学习器比较
 - 交叉验证 t 检验 (基于成对 t 检验)
 - McNemar 检验 (基于列联表、卡方检验)
 - 多学习器比较
 - Kolmogorov-Smirnov Test (K-S 检验)
 - Friedman 检验 (基于序值, F 检验；判断“是否相同”)
 - Nemenyi 后续检验 (基于序值, 进一步判断两两差别)

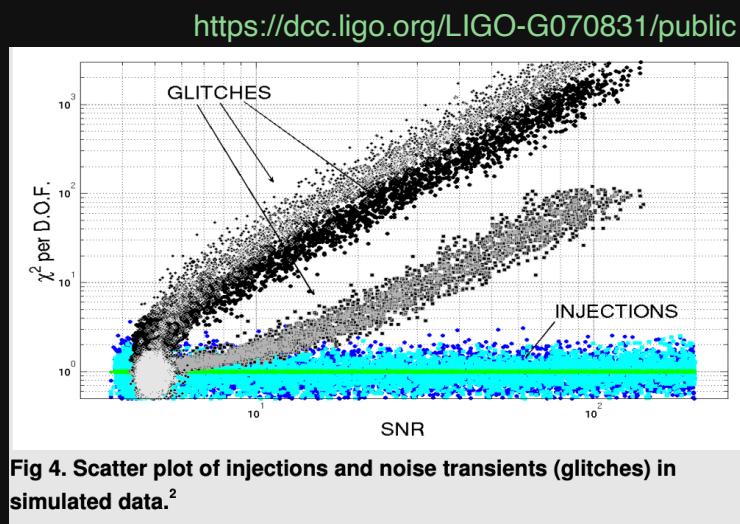


Fig 4. Scatter plot of injections and noise transients (glitches) in simulated data.²

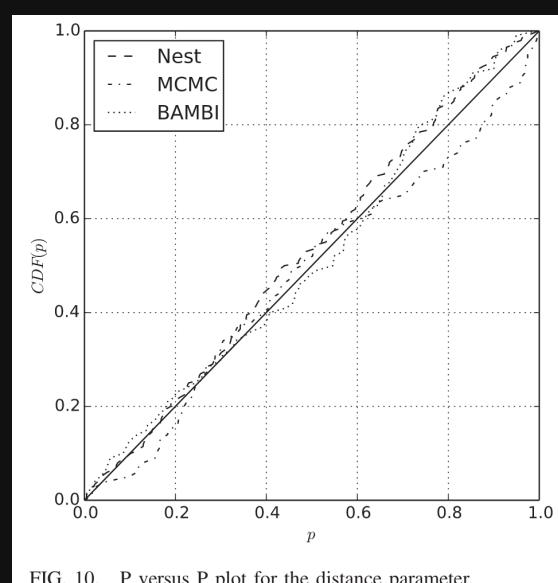


FIG. 10. P versus P plot for the distance parameter.

Veitch, J., et al. *Physical Review D* 91, no. 4 (February 2015): 042003.
<https://doi.org/10.1103/PhysRevD.91.042003>.

