

机器学习与深度学习 ——AI/ML/DL 概述



Personal Website: <https://www.miaopeng.info/>



Email: miaopeng@stu.scu.edu.cn



Github: <https://github.com/MMeowhite>



Youtube: <https://www.youtube.com/@pengmiao-bmm>

目录章节

CONTENTS

01 导言：AI 是什么？

02 机器学习（ML）

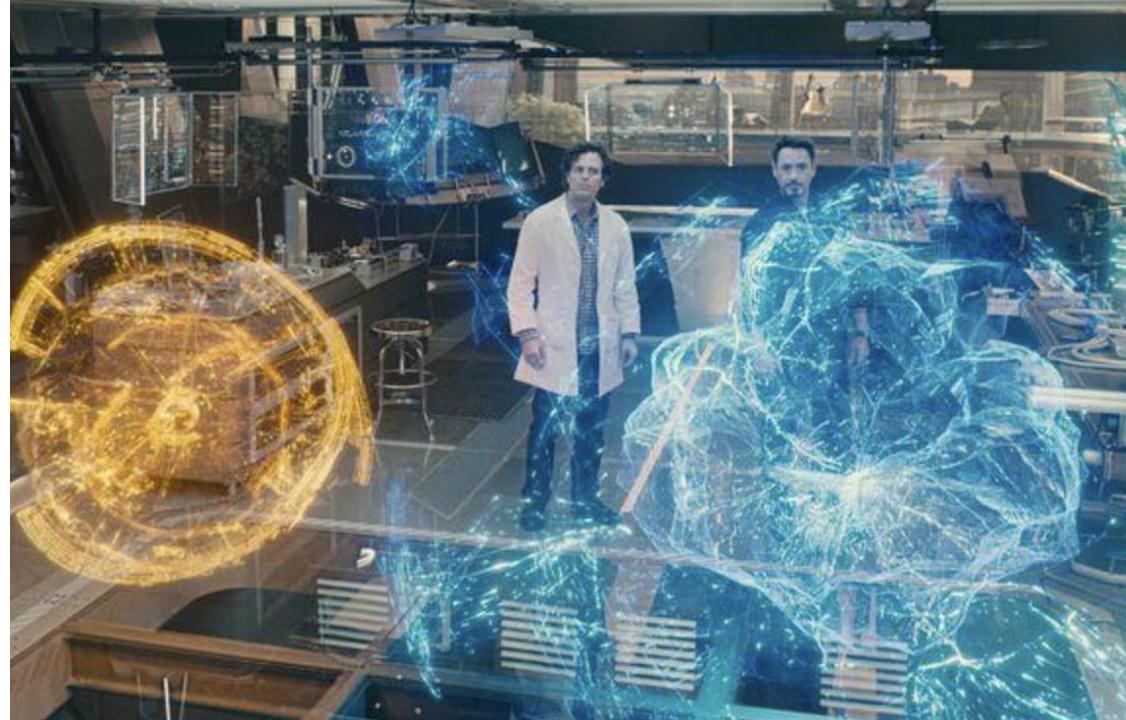
03 深度学习（DL）

04 生成式 AI 与大模型

05 总结

► 什么是人工智能（AI）？

- 人工智能（Artificial Intelligence, AI）是让计算机具备模拟人类智能的能力，如感知、推理、学习、规划和自然语言处理等。



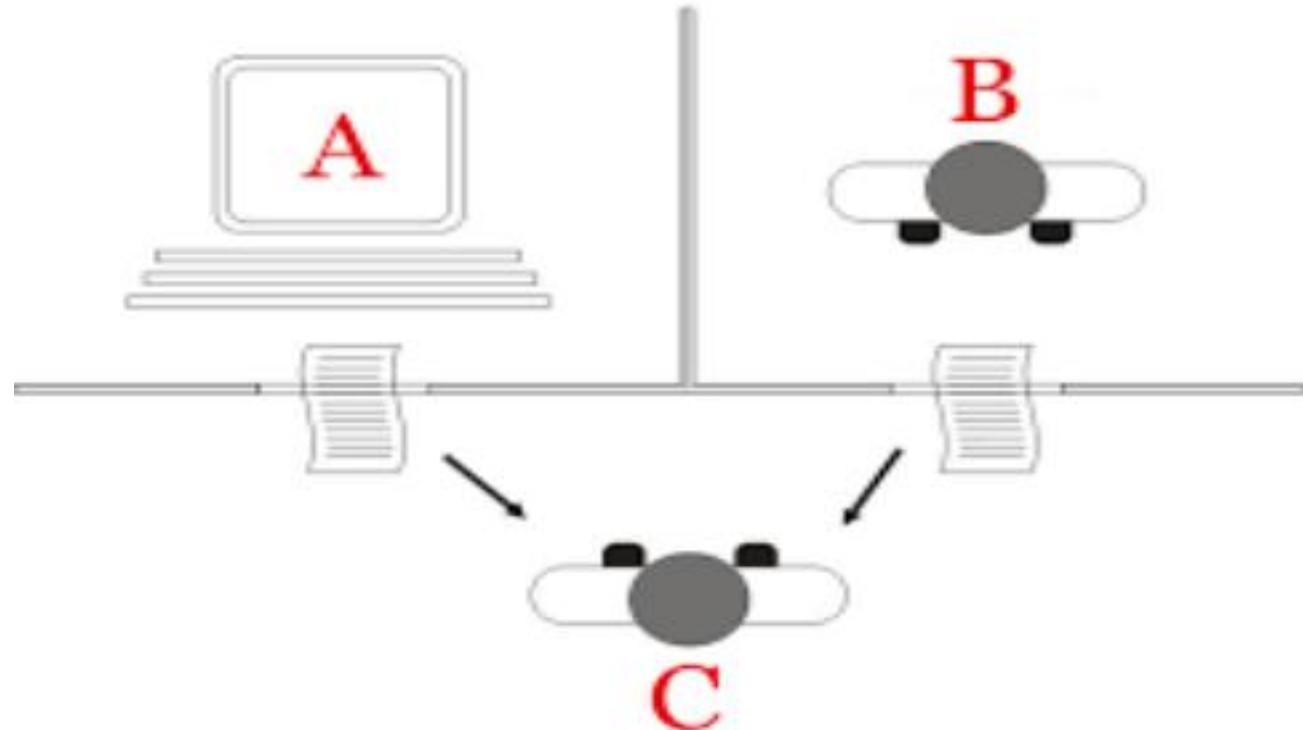
人工智能是继工业、电力、核革命之后，人类文明迎来的又一次技术变革

► 从图灵到OpenAI：AI 大师眼中的智能世界

- 如果一个人在与机器对话时无法判断对方是人还是机器，那么这台机器就具有智能（**图灵测试**）。



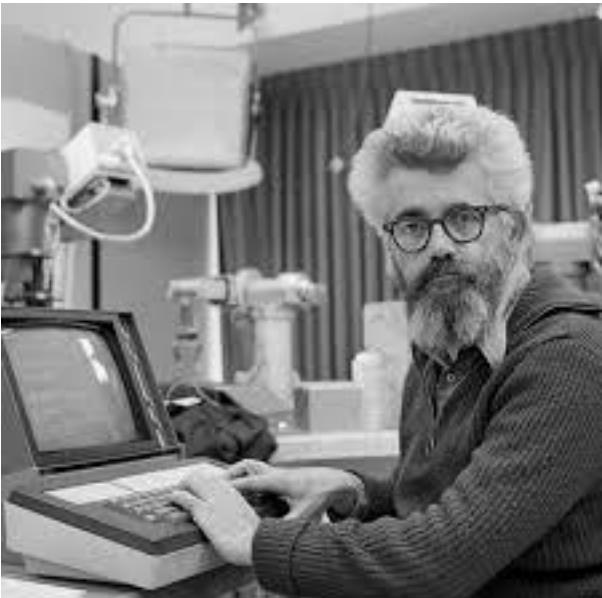
(Alan Turing, 计算机之父, 1950 年)



背景：提出“图灵测试”，早期用于衡量机器智能

► 从图灵到OpenAI：AI 大师眼中的智能世界

- 人工智能是使机器表现出智能行为的科学与工程（**达特茅斯会议，Dartmouth Summer Research Project on Artificial Intelligence**）。



(John McCarthy, AI 之父, 1956 年)



背景：提出“AI”一词，达特茅斯会议正式奠基人工智能领域。

► 从图灵到OpenAI：AI 大师眼中的智能世界

- 机器学习是让计算机具有无需显式编程就能学习的能力。



(Arthur Samuel, 1959 年)

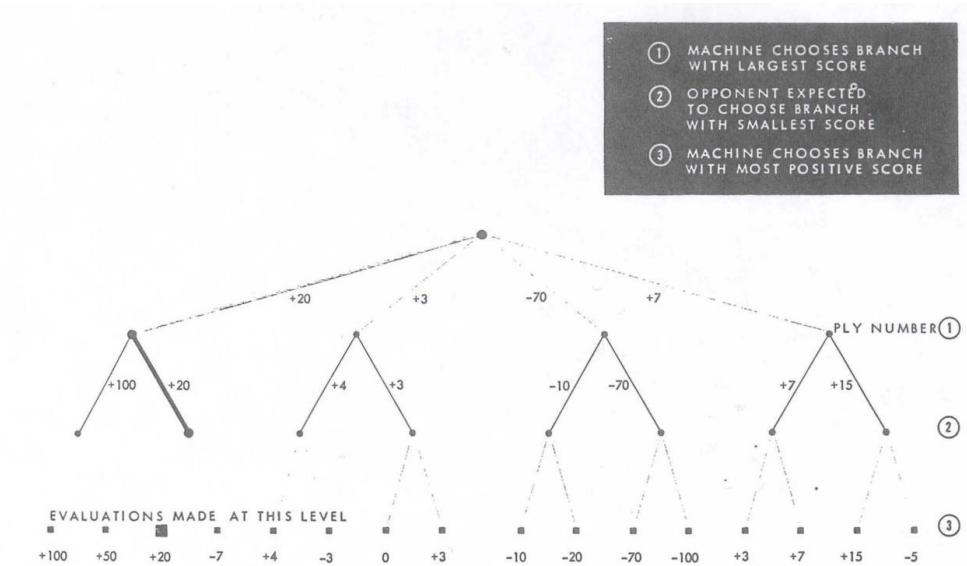


Figure 2 Simplified diagram showing how the evaluations are backed-up through the "tree" of possible moves to arrive at the best next move. The evaluation process starts at ③.

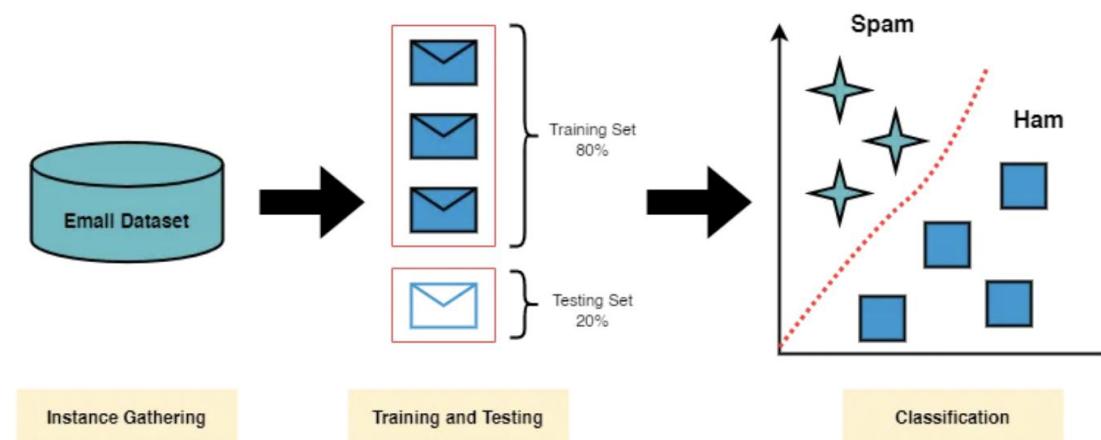
背景：编写了跳棋程序（启发式搜索 (Minimax + α - β 剪枝 + Evaluation 函数），首次提出“Machine Learning”概念与雏型。

► 从图灵到OpenAI：AI 大师眼中的智能世界

- 如果一个程序在某类任务 T 上的表现度量 P 随着经验 E 的增加而提升，那么它就能从经验中学习。



(Tom M. Mitchell, 1997 年)



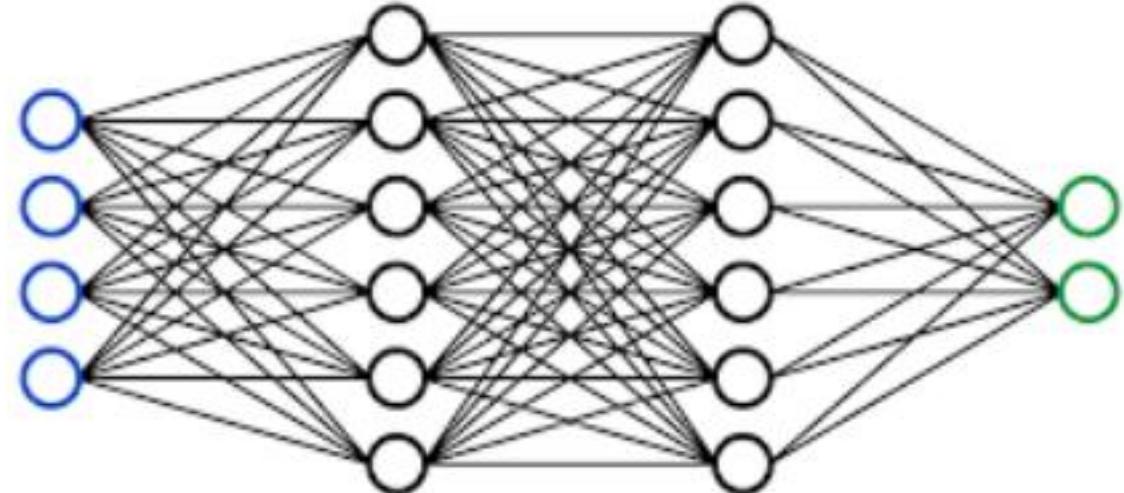
示例：垃圾邮件分类 (T)、经验为训练数据 (E)、准确率为性能指标 (P)

► 从图灵到OpenAI：AI 大师眼中的智能世界

- 深度学习的核心是使用多层神经网络从原始数据中自动提取特征。



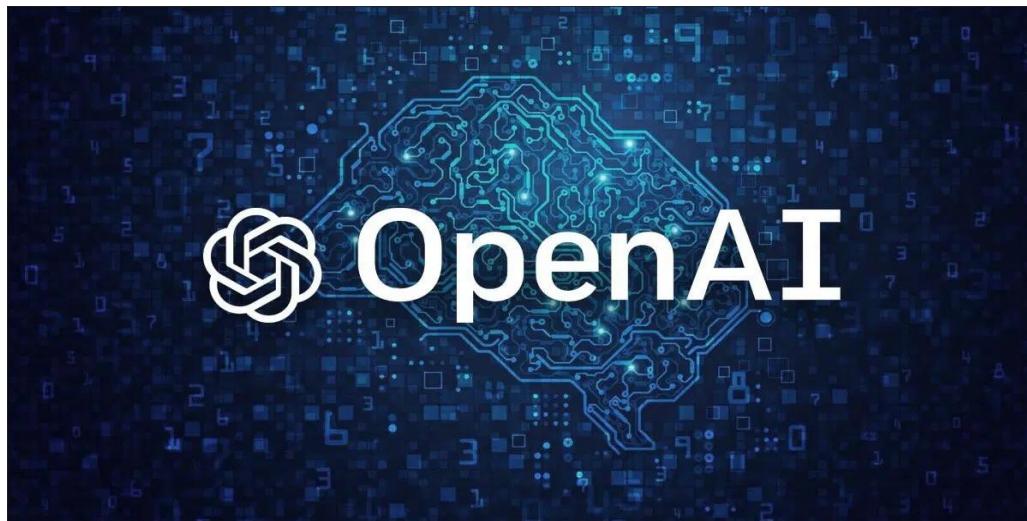
(Geoffrey Hinton, 深度学习之父,
2014 年)



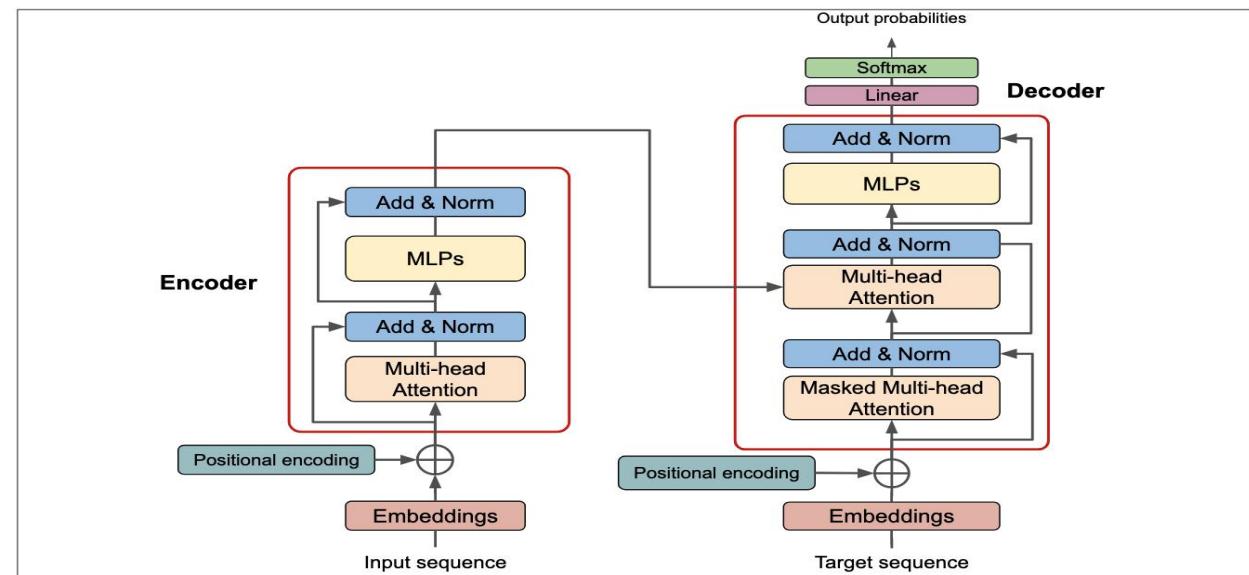
背景：创建的深度学习模型，引发 2012 年后 AI
爆发（AlphaGo, AlphaFold, ChatGPT…）。

► 从图灵到OpenAI：AI 大师眼中的智能世界

- GPT 是一种大规模自回归语言模型，它通过预测下一个词来生成连贯文本。



(OpenAI, 2023 年)



背景：开发 ChatGPT、Prompt 工程、大语言模型架构（Transformer）等突破性工作

► AI 的发展阶段（简史）

- 从规则驱动，到统计学习，再到深度学习和大模型，AI 已历经数十年演化，进入智能革命新时代。

阶段	时间	特点
萌芽期	1950s – 1970s	图灵测试、逻辑推理、专家系统雏形（规则驱动）
冬天期	1970s – 1980s	由于缺乏算力和数据，AI 停滞，称为“AI 寒冬”
复苏期	1980s – 2010	统计学习方法发展、SVM、机器学习崛起
井喷期	2012 – 至今	深度学习（DL）兴起，大数据 + GPU 算力推动突破，ChatGPT、AlphaGo 等爆红

AI：从逻辑推理 → 机器学习 → 深度学习 → 生成式智能。

目录章节

CONTENTS

01

导言：AI 是什么？

02

机器学习（ML）

03

深度学习（DL）

04

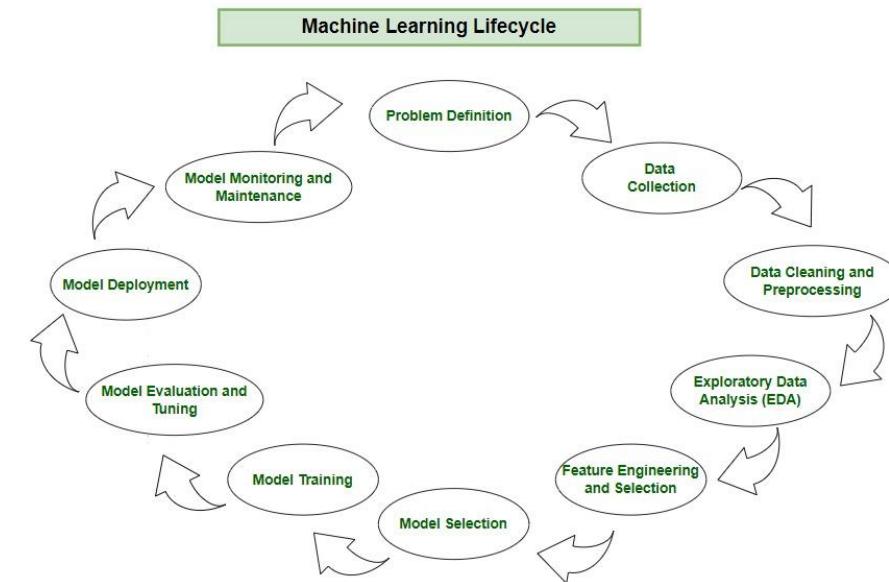
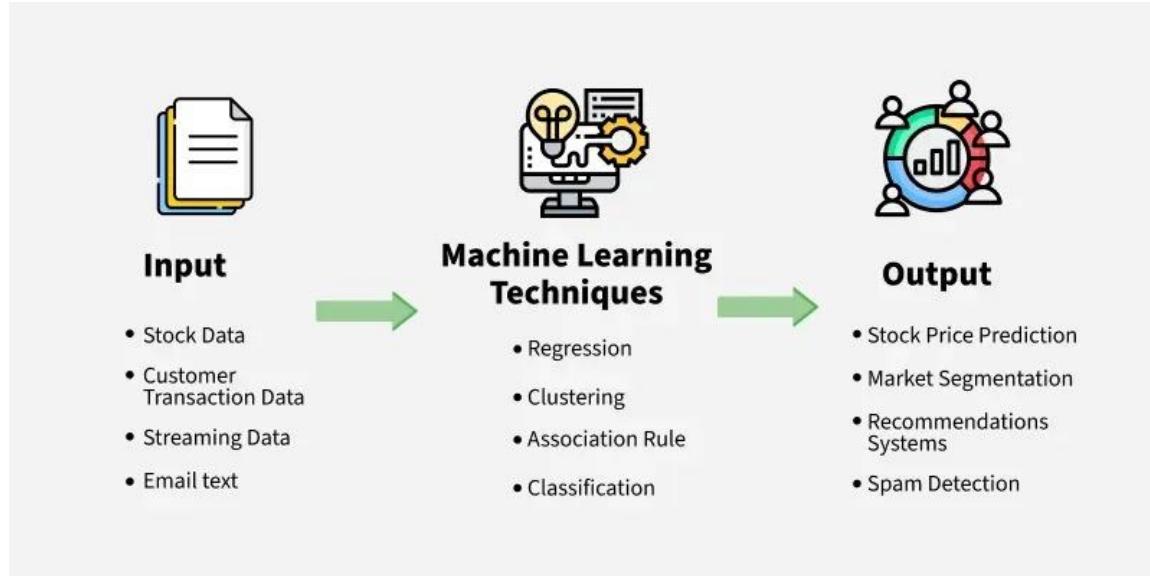
生成式 AI 与大模型

05

总结

▶ 定义与核心思想

- 机器学习是让计算机从数据中**自动学习规律**，无需明确编程。
- 核心思想是“用**数据训练模型**”，使系统能做出预测或决策。



本质：本质是优化模型，使其在新数据上表现良好。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/ml-machine-learning/>

► 机器学习数据：特征与标签

- 特征是模型用来“理解”每个数据样本的输入信息，它们是数据中具有代表性的**属性**，选取什么特征、如何处理特征，直接影响模型的学习效果和泛化能力。
- 标签是机器学习中用来告诉模型“正确答案”的部分，它代表了训练数据中每个样本所属的**目标或类别**，监督学习依赖标签进行训练，而标签质量直接决定模型表现。

	gene_1	gene_2	gene_n	cell_type
cell_1	0.20	0.30	0.00	0(T cell)
cell_2	0.21	0.90	0.80	1(B cell)
.....
cell_n-1	0.33	0.40	0.20	2(NK cell)
cell_n	0.18	0.94	0.78	1(B cell)

特征 标签

机器学习通过特征来理解输入，通过标签来学习目标，两者共同驱动模型训练。

► 机器学习数据：训练集/测试集/验证集

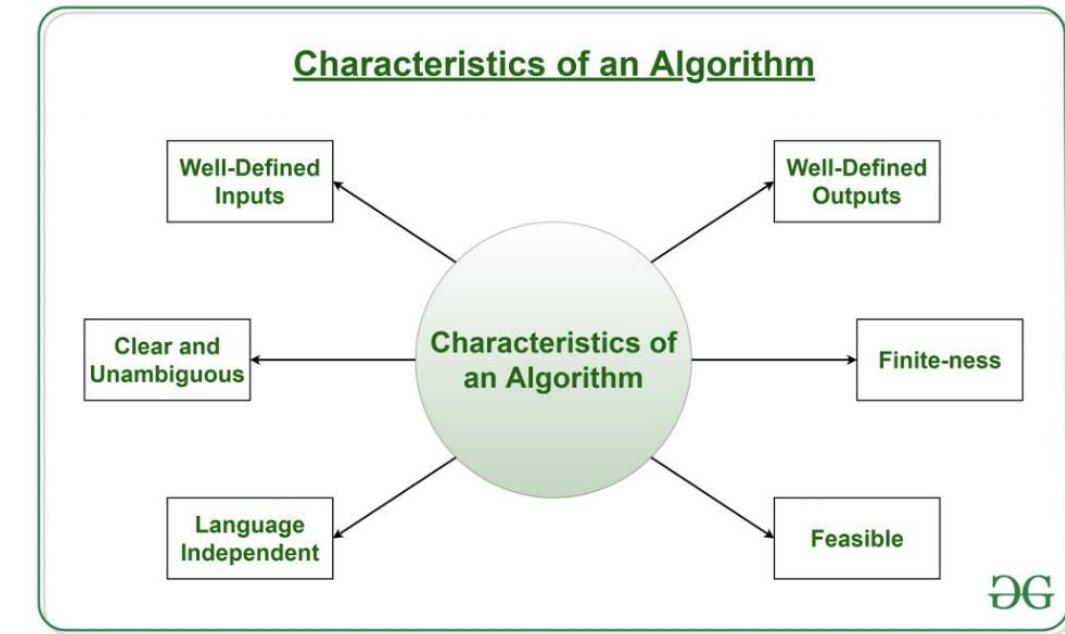
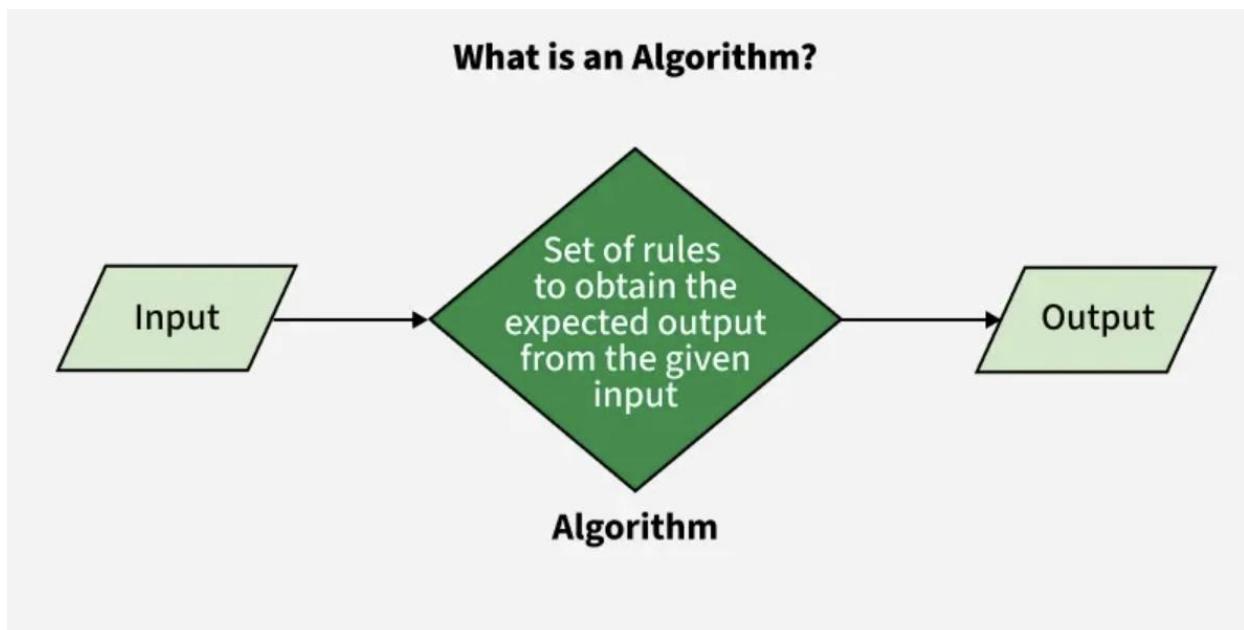
- 训练集：用于模型**学习**参数，是模型“看到”的数据，模型在这部分数据上拟合规律。
- 验证集：用于**调整**模型超参数，评估不同模型选择，不参与训练，但用于“调优”和“提前停止”。
- 测试集：用于最终**评估**模型性能，完全独立于训练过程，检验模型的泛化能力。

	gene_1	gene_2	gene_n	cell_type
训练集 (70%)	cell_1	0.20	0.30	0.00
训练集 (20%)
测试集(10%)

合理划分训练、验证和测试集，是确保模型性能可靠的关键。

▶ 常见算法：算法概述

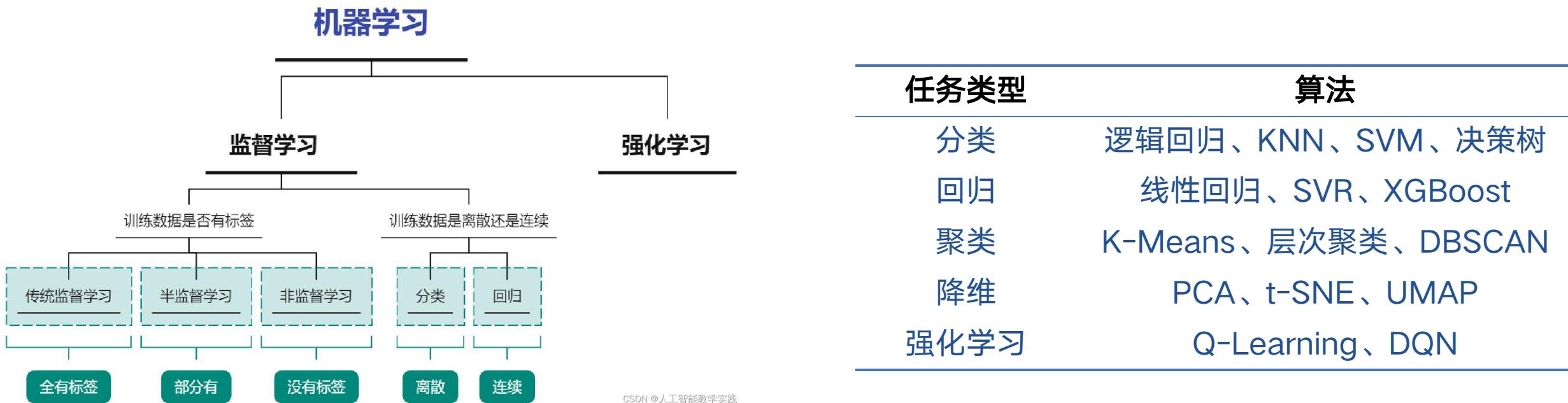
- 算法是解决问题的**有限步骤集合**，是计算机程序的核心，涵盖排序、查找、图论、动态规划、分治等多种经典方法。
- 算法特性：正确性、可读性、高效性、可扩展性和健壮性



扩展阅读资料：<https://www.geeksforgeeks.org/dsa/introduction-to-algorithms/>

▶ 常见算法：机器学习算法概述

- 机器学习算法通过从数据中自动学习模式，实现**预测、分类或决策**。
- 常见算法包括监督学习（如线性回归、决策树、SVM）、半监督学习（Self-training、Graph-based）、无监督学习（如K-Means、PCA）**、强化学习（如Q-learning）**。

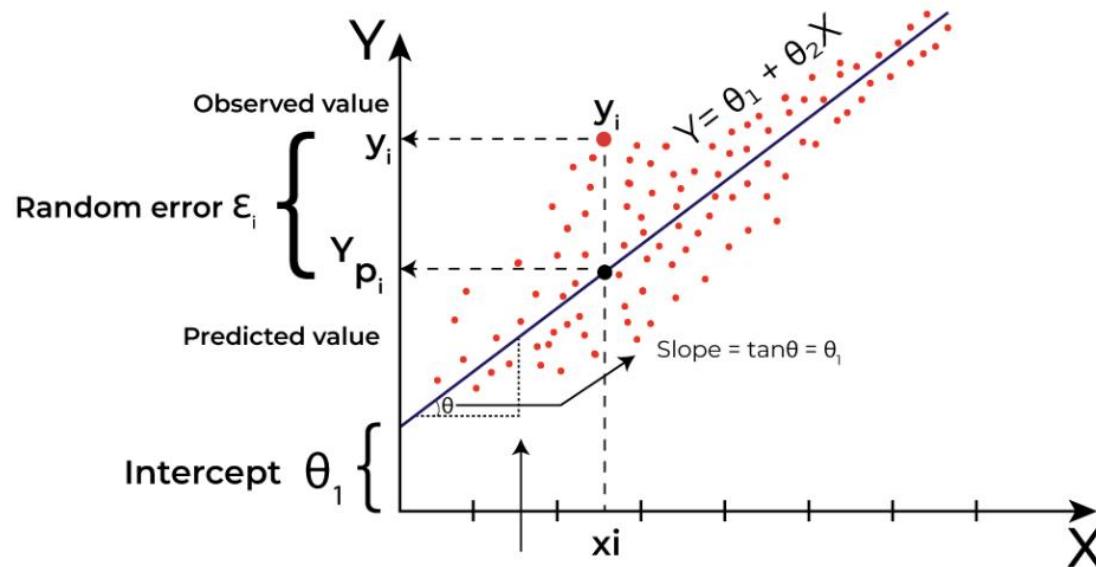


不同算法适用于不同任务，取决于数据特性、学习目标和问题复杂度。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/machine-learning/>

▶ 常见算法：线性回归（Linear Regression）

- 线性回归用于拟合一条直线预测连续数值。
- 通过最小化预测值与真实值之间的误差（**最小二乘法**）来确定最优参数。



$$m = \frac{N \sum(xy) - \sum x \sum y}{N \sum(x^2) - (\sum x)^2}$$

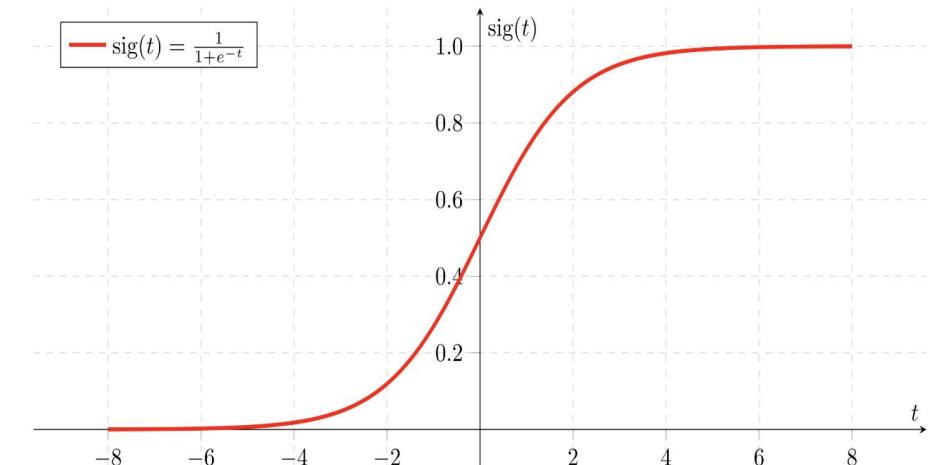
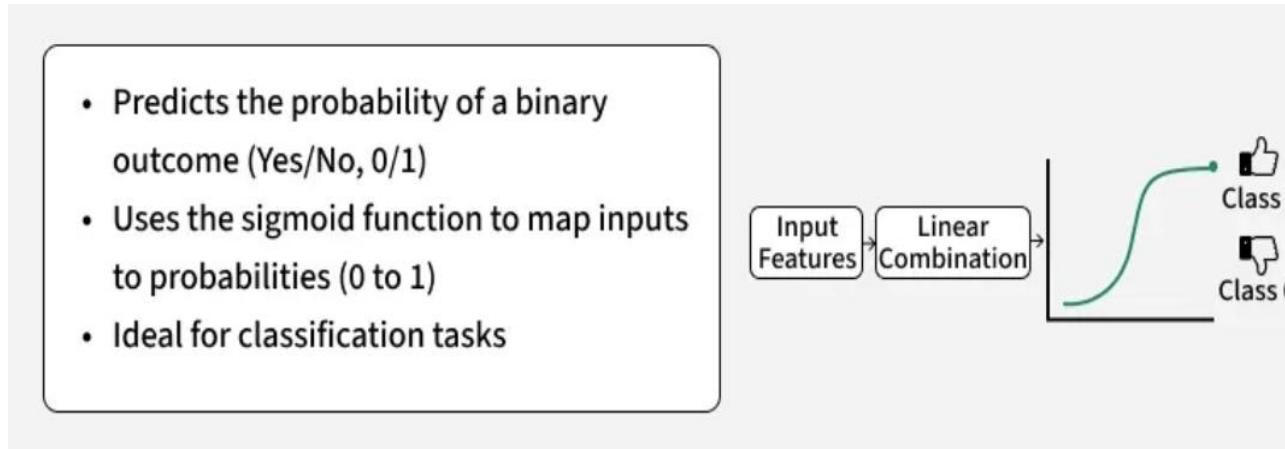
$$b = \frac{\sum y - m \sum x}{N}$$

应用：适用于房价预测、销售趋势分析等**连续变量**场景

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/>

▶ 常见算法：逻辑回归 (Logistic Regression)

- 逻辑回归是分类算法，不是回归，用于预测事件发生的概率。
- 它将线性组合输入通过 Sigmoid 函数映射到 0 – 1 之间。

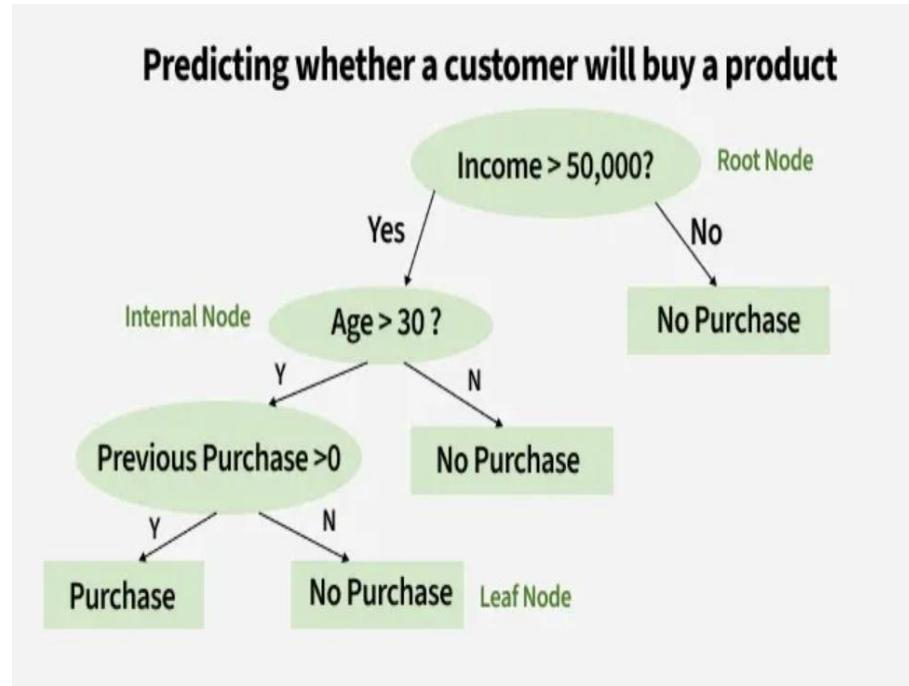


应用：常用于**二分类任务**，如垃圾邮件识别、用户是否流失等

扩展阅读资料：<https://www.geeksforgeeks.org/dsa/introduction-to-algorithms/>

▶ 常见算法：决策树（Decision Trees）

- 决策树通过一系列“是/否”问题逐步划分样本空间。
- 模型结构直观，便于解释（**可解释性强**），适合分类和回归问题。



算法步骤：

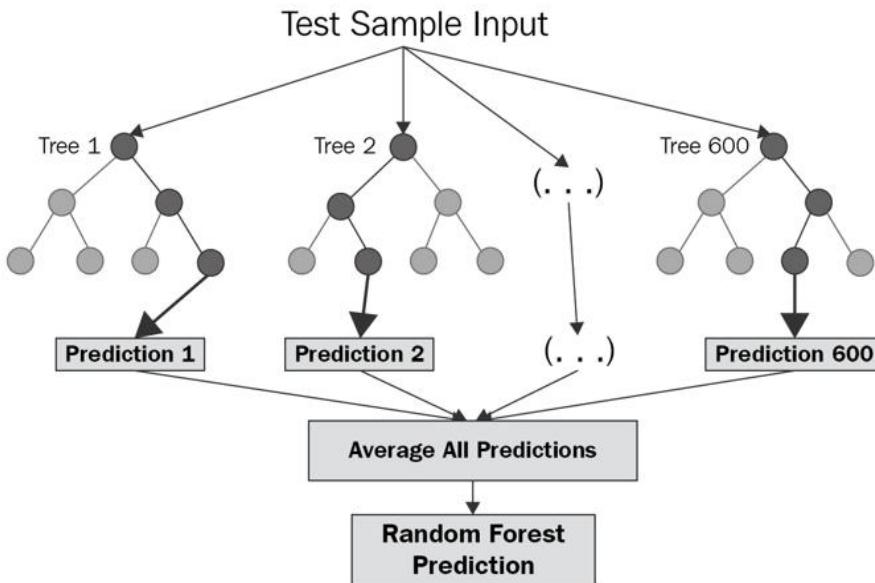
- (1) 如果当前数据集中样本全属于同一类别，则该节点为叶节点，类别为该类；
- (2) 如果当前特征集为空或达到停止条件（如最大深度），则该节点为叶节点，类别为样本中出现频率最高的类；
- (3) 否则，选择一个“最优划分特征”作为当前节点的划分
- (4) 根据该特征的不同取值将数据集划分为若干子集，分别构建子节点；
- (5) 对每个子集递归调用步骤 (1) – (4)，继续划分，直到满足停止条件；

应用：根据年龄、收入判断是否批准贷款。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/decision-tree-introduction-example/>

► 常见算法：随机森林（Random Forest）

- 随机森林是多个决策树的集成模型，通过投票或平均来输出结果。
- 它能有效**提升模型的稳定性和准确性，降低过拟合风险。**



算法步骤：

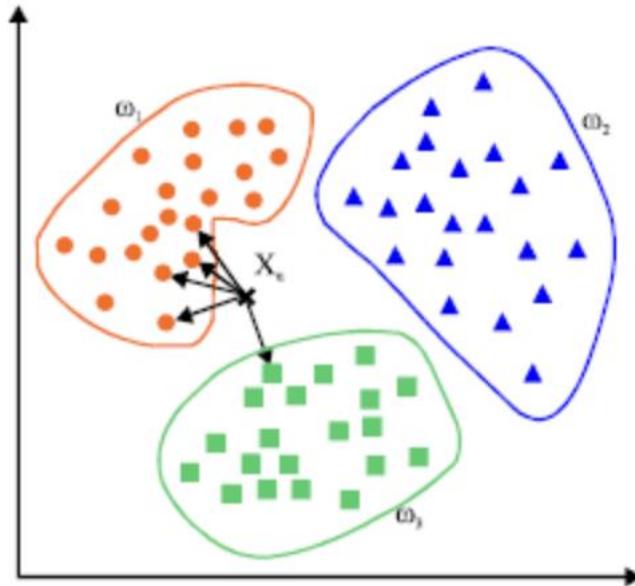
- (1) 使用 Bootstrap 方法从训练集中随机有放回采样，生成 n 组子数据集，每组训练一个决策树；
- (2) 在构建每棵决策树时的每个分裂节点，随机选取部分特征，在这些特征中选择**最优划分**；
- (3) 每棵树都单独对当前点进行**分类预测**，输出一个类别标签；
- (4) 收集所有树的预测结果，统计各类别的出现频率；
- (5) 返回出现频率最高的类别作为当前点的**最终预测分类**。

应用：金融风控、信用评分、图像识别等任务。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/>

▶ 常见算法：K 近邻算法（KNN）

- KNN 预测新样本时，查找其**最近的 K 个邻居**并“投票”决定分类或回归值。
- 无需训练过程，但**预测阶段计算量大**。



算法步骤：

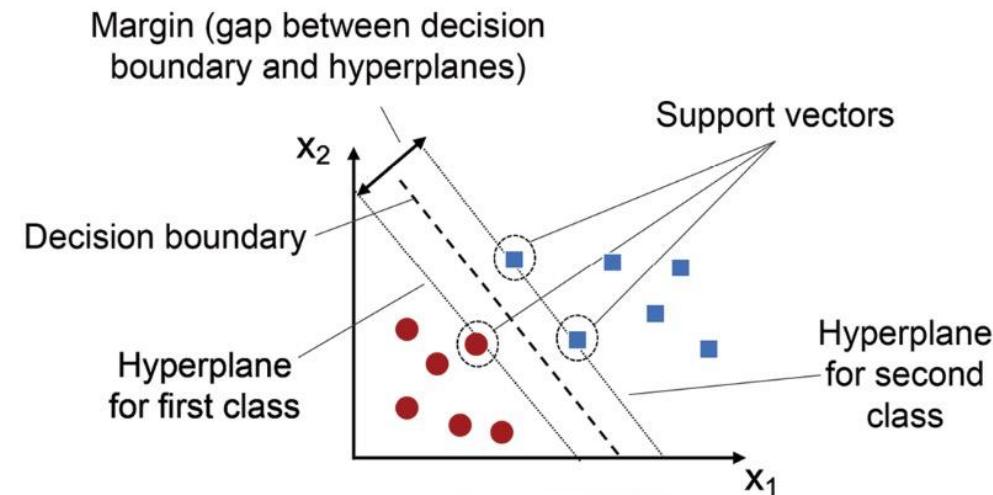
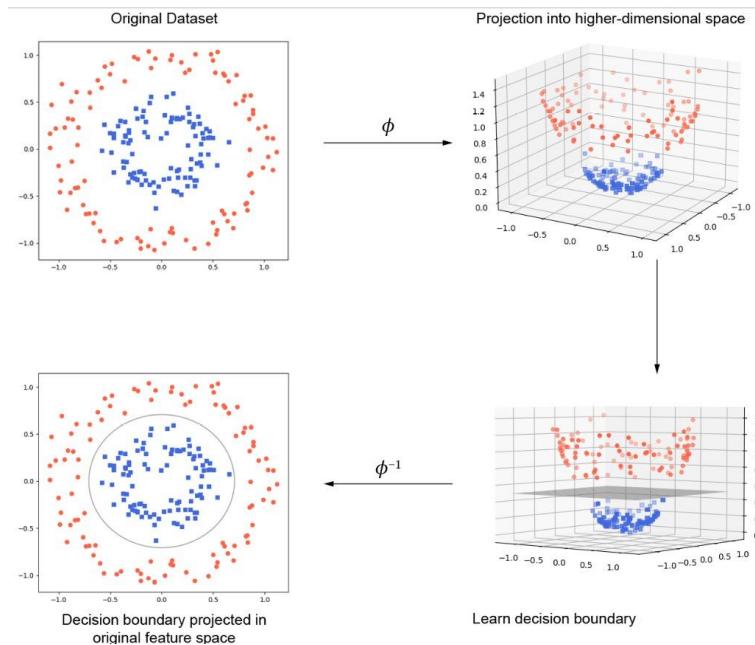
- (1) 计算测试集数据点与训练集每个点与当前点之间的距离；
- (2) 按照距离递增排序；
- (3) 选取与当前点距离最小的**k**个点；
- (4) 统计前k个点的类别频率；
- (5) 返回前k个点出现频率最高的类别作为当前点的预测分类

应用：适用于**小样本、维度较低**的问题，如图像识别、推荐系统。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbours/>

▶ 常见算法：向量支持机（SVM）

- SVM 通过寻找**最大间隔的超平面**来区分不同类别样本。
- 可通过核函数扩展到**非线性**分类任务【不能用一条直线（或一个平面）把不同类别分开】。

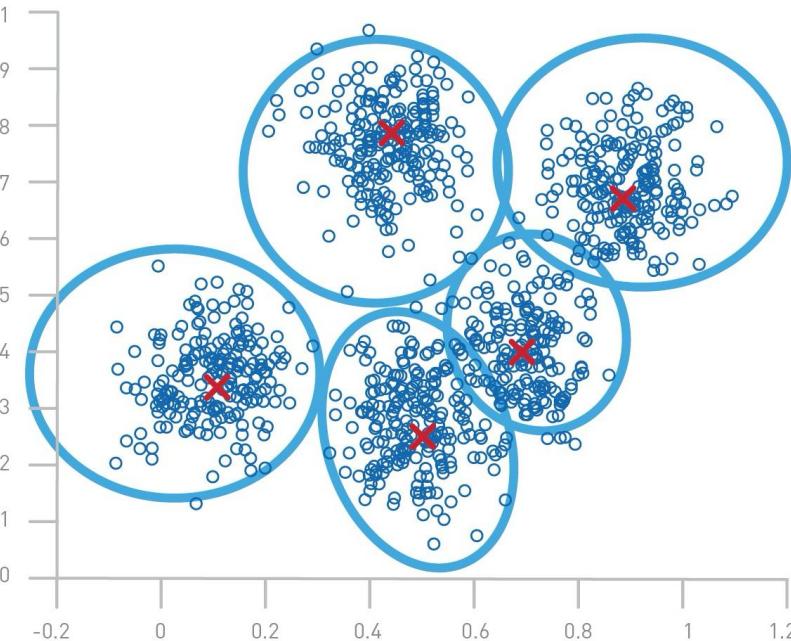


应用：文本分类、人脸识别。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbours/>

▶ 常见算法：K-Means 聚类（KMeans cluster）

- K-Means 是无监督学习算法，将数据**自动划分为 K 个相似群组。**
- 通过**反复更新簇中心，最小化组内距离。**



算法步骤：

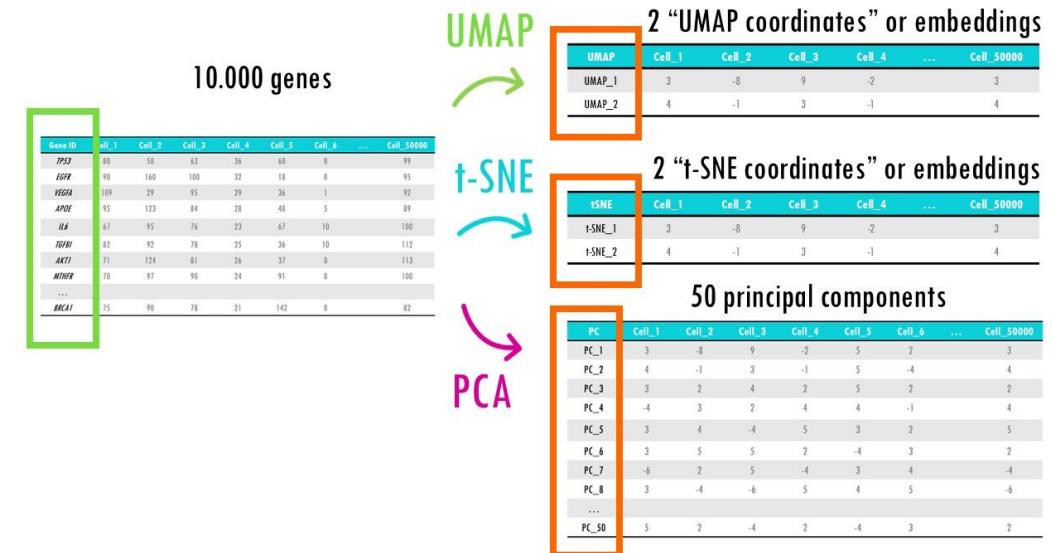
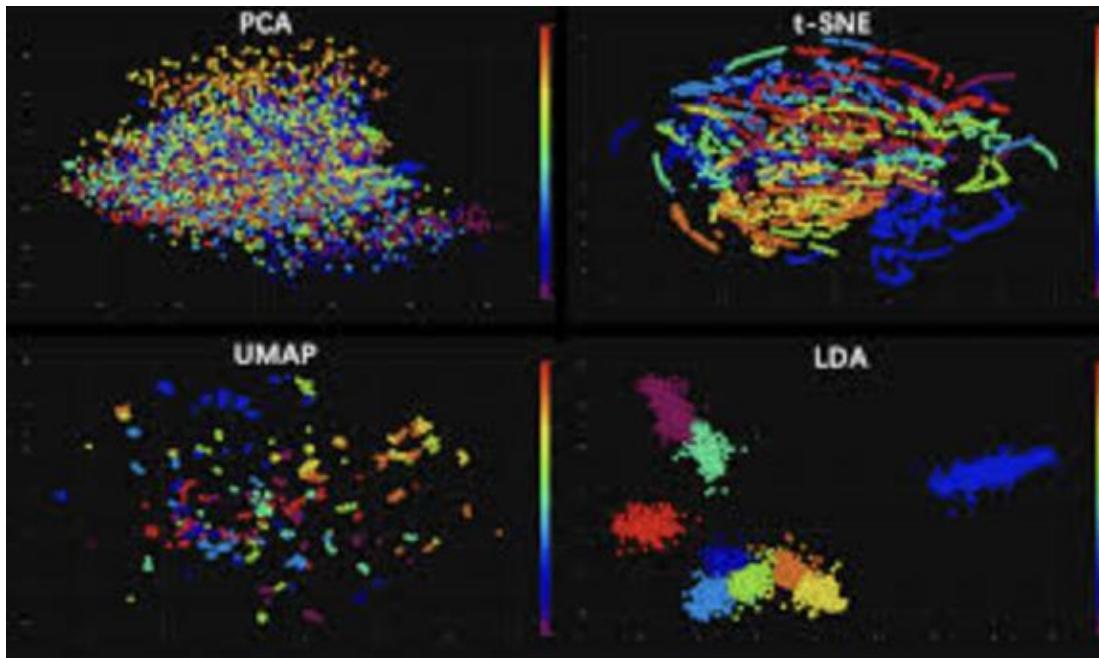
- (1) 随机选择 K 个样本作为**初始聚类中心（质心）**；
- (2) 对数据集中每个点，计算其与所有聚类中心的**距离**；
- (3) 将每个点分配给距离它最近的聚类中心，形成 K 个簇；
- (4) 对每个簇，计算其所有样本的均值作为新的聚类中心；
- (5) 如果所有聚类中心的位置不再变化，或达到最大迭代次数，则停止；否则返回步骤 (2)；

应用：用于客户分群、图像压缩等应用场景。

扩展阅读资料：<https://www.nvidia.com/en-us/glossary/k-means/>

▶ 常见算法：PCA / t-SNE / UMAP（降维算法）

- PCA 是线性降维方法，保留数据中最有代表性的方向。
- t-SNE 和 UMAP 是非线性降维方法，适合可视化高维数据的结构。



应用：数据预处理或可视化，如单细胞测序细胞表达降维。

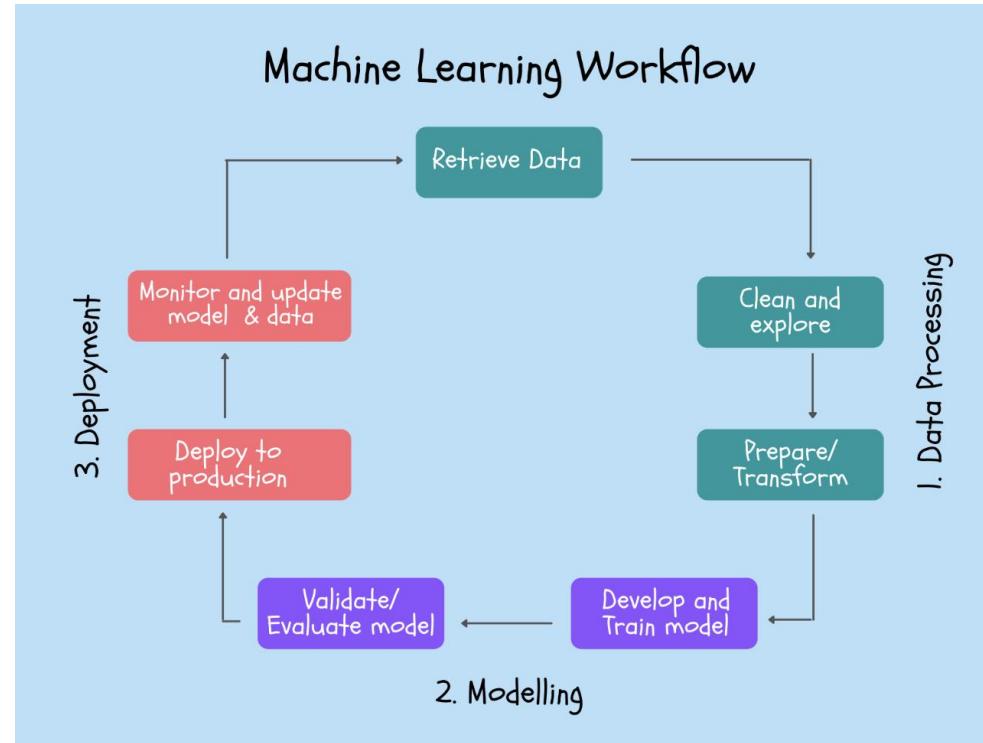
扩展阅读资料：<https://biostatsquid.com/pca-umap-tsne-comparison/>

▶ 常见算法：总结

模型	类型	是否监督	应用	总结
线性回归	回归	是	连续预测	拟合一条直线预测连续值，用最小误差寻找最优解
逻辑回归	分类	是	二分类问题	用 S 型函数将线性输出映射为分类概率
决策树	分类/回归	是	可解释模型	通过一系列特征判断构建“if-else” 规则树进行分类或回归。
随机森林	分类/回归	是	集成、鲁棒性强	多棵决策树投票组成的集成模型，准确率高且抗过拟合
KNN	分类/回归	是	相似度计算	通过计算距离最近的K个样本来预测当前的类型或数值
SVM	分类	是	二/多分类	寻找能最大分割不同类别的边界线或超平面
KMeans	聚类	否	分群	讲样本自动划分为K个组，使组内相似、组间差异最大
PCA、t-SNE、UMAP	降维	否	信息压缩	将高维数据压缩到低维空间，提取最有信息量的方向

► 工作流程回顾

- 核心流程：用预处理数据训练模型，从中学习规律，再用它对新数据做出预测或决策，最后返回进一步修正模型，形成训练环路。



ML: 数据预处理 → 特征工程 → 模型选择 → 训练与评估 → 部署。

扩展阅读资料: <https://towardsdatascience.com/the-machine-learning-workflow-explained-557abf882079/>

目录章节

CONTENTS

01

导言：AI 是什么？

02

机器学习（ML）

03

深度学习（DL）

04

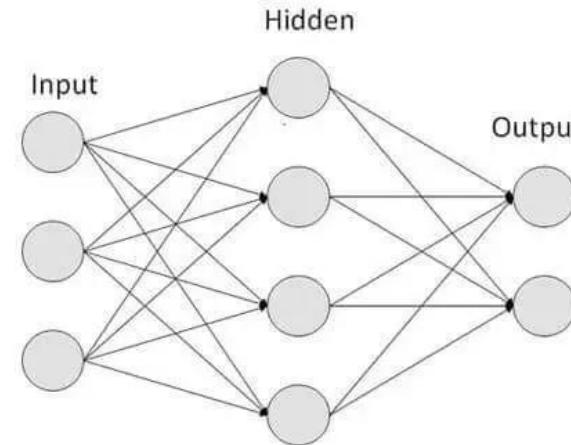
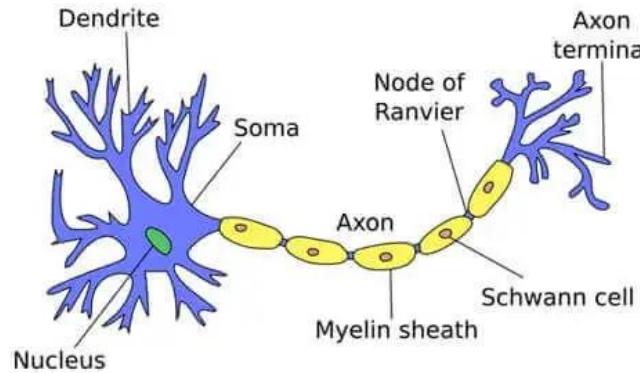
生成式 AI 与大模型

05

总结

▶ 定义与核心思想

- 深度学习是机器学习的一个分支，基于**多层神经网络自动提取特征并进行学习**。
- 核心思想是通过构建“深层结构”**模拟人脑神经元**，实现从数据中端到端的学习。



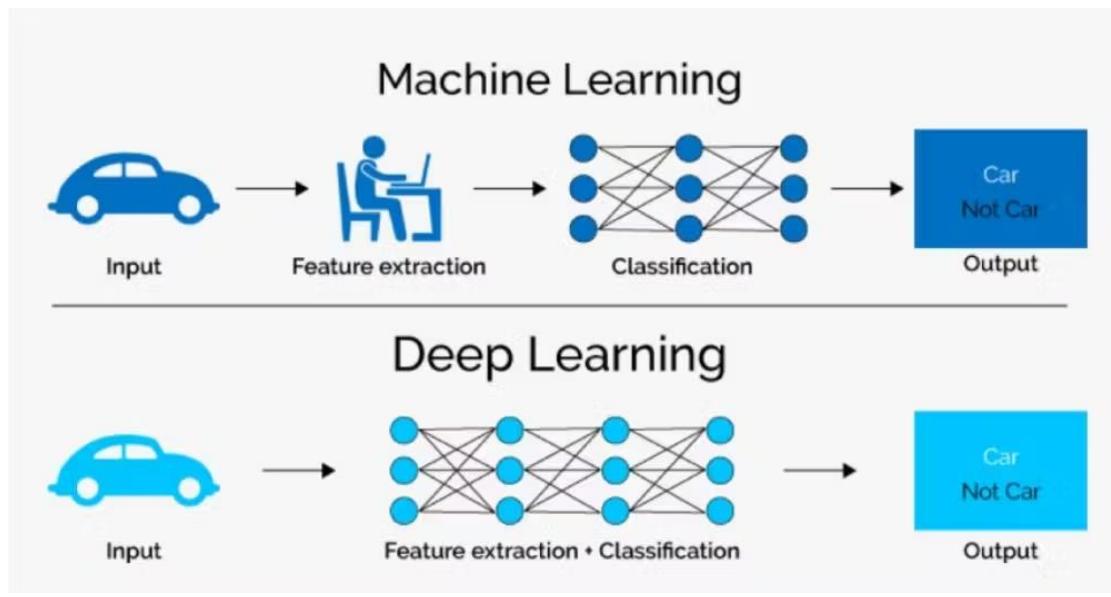
Notes: 端对端学习是一种训练方式，模型直接从原始输入学习到最终输出，中间过程不依赖人工规则或手工特征工程。

擅长处理**非结构化数据**（如图像、语音、文本），通过大量数据不断优化性能。

扩展阅读资料：<https://towardsdatascience.com/the-machine-learning-workflow-explained-557abf882079/>

► DL与ML 的区别

- 传统机器学习依赖人工设计特征，适合**结构化**数据，如表格或数值型数据。
- 深度学习通过神经网络自动提取特征，能处理图像、音频、文本等**非结构化**数据。

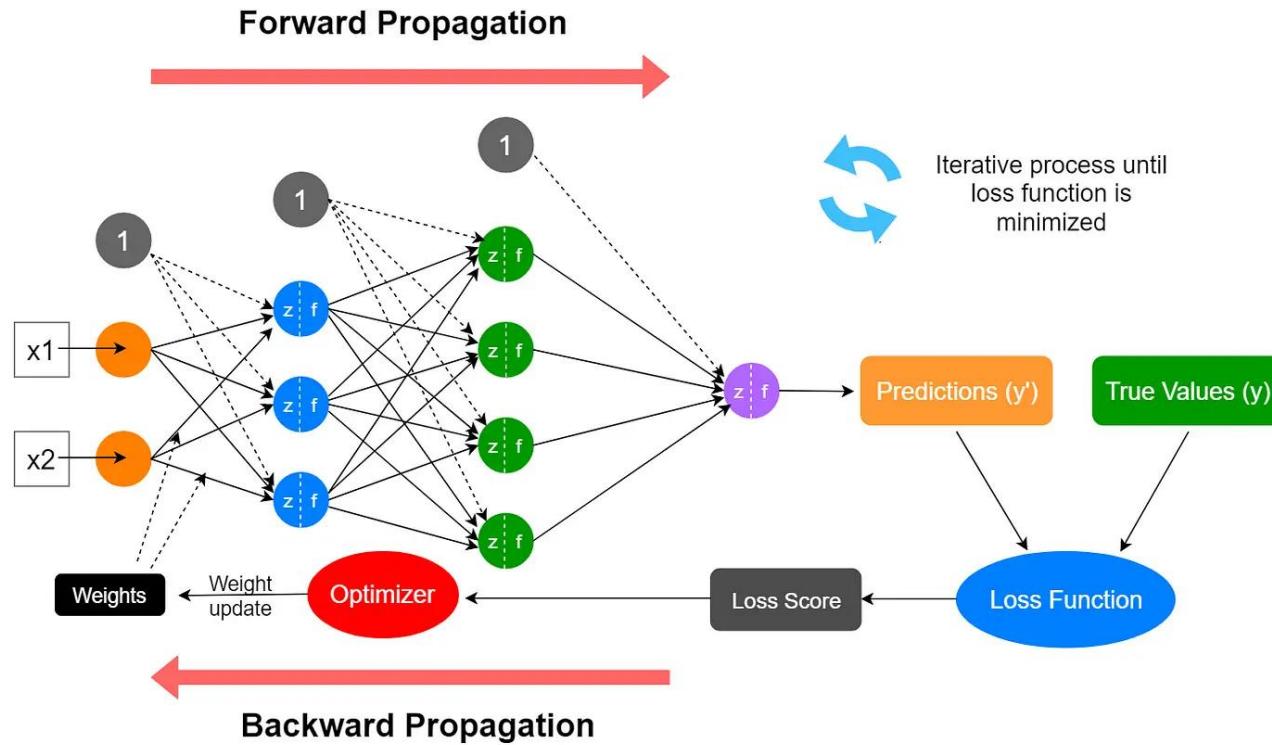


对比维度	ML	DL
特征工程	人工设计特征	自动学习特征
数据类型	结构化数据	非结构化数据
模型复杂度	模型相对简单、可解释	模型复杂、可解释差
性能依赖	对特征质量敏感	对数据量和计算资源敏感
常用模型	SVM、决策树、KNN、线性回归等	CNN、RNN、Transformer 等

DL在大数据和计算资源充足的情况下通常表现更好，但模型更**复杂、不易解释**

▶ 算法基础：神经网络算法

- 神经网络是一种由多层神经元连接构成的模型，能够从数据中**自动学习复杂的非线性映射关系。**



算法步骤

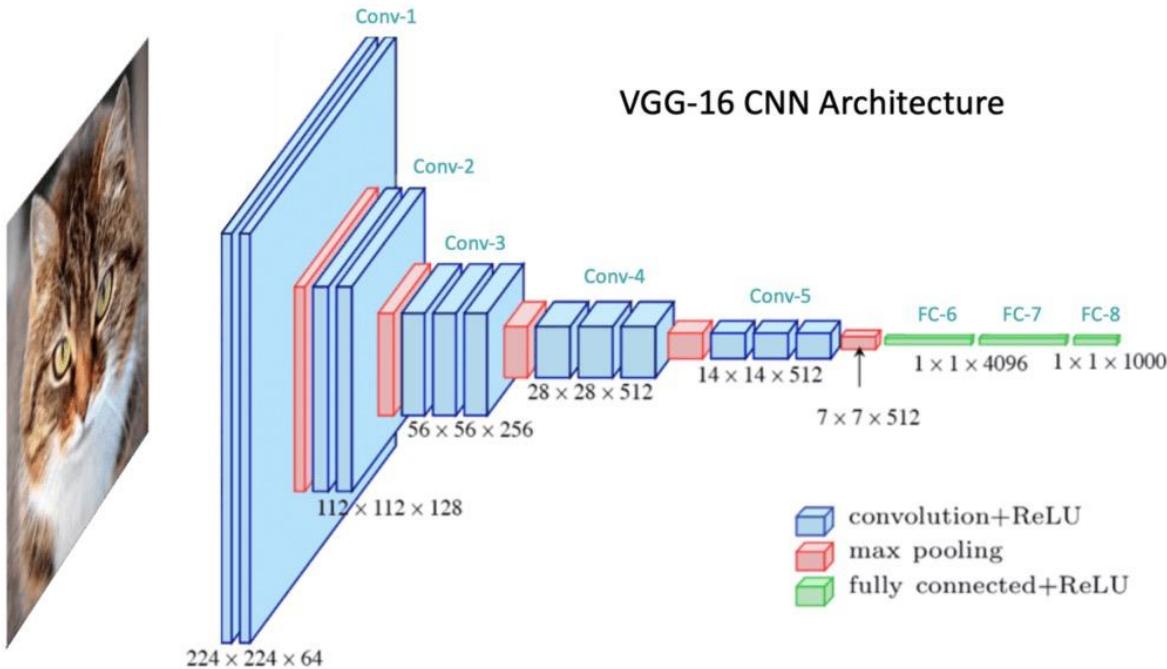
- 初始化网络结构与参数，包括输入层、隐藏层、输出层的权重与偏置；
- 将输入数据通过网络进行前向传播，逐层计算输出；
- 使用损失函数计算预测结果与真实值之间的误差；
- 通过反向传播算法计算每层的梯度，利用梯度下降更新参数；
- 重复训练多个轮次，直到损失收敛或满足终止条件。

前向传播生成预测、反向传播调整权重，网络逐步优化，完成分类、回归等任务。

扩展阅读资料：<https://towardsdatascience.com/the-machine-learning-workflow-explained-557abf882079/>

▶ 主要网络结构：CNN

- 卷积神经网络是一种专门处理图像数据的神经网络，能**自动提取空间特征**。
- 通过卷积层提取局部特征、池化层降维、全连接层输出分类结果。



算法步骤：

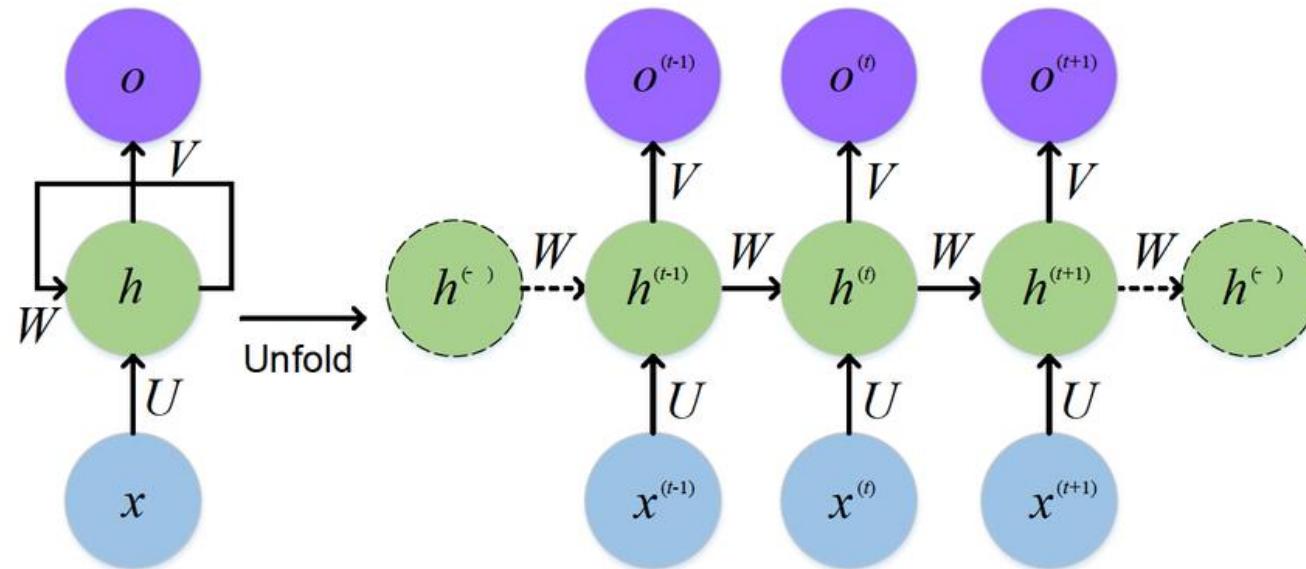
- (1) 输入图像数据，并将其转换为多维张量格式；
- (2) 通过卷积层提取局部特征，生成特征图（feature maps）；
- (3) 使用激活函数（如 ReLU）引入非线性，提高模型表达能力；
- (4) 通过池化层（如最大池化）对特征图进行降维，保留主要信息；
- (5) 展平特征图并连接全连接层，输出最终分类或预测结果。

CNN 利用权重共享和局部连接机制，大大减少参数，提高训练效率。

扩展阅读资料：<https://towardsdatascience.com/the-machine-learning-workflow-explained-557abf882079/>

► 主要网络结构：RNN

- RNN 是一种能处理序列数据的神经网络，它在每一步上使用当前输入和前一步的状态共同决定当前的输出和新状态。



$$h = \text{sigma}(U \cdot X + W \cdot h_{(t-1)} + B)$$

Notes: h represents the current hidden state; U and W are weight matrices; B is the bias.

算法步骤：

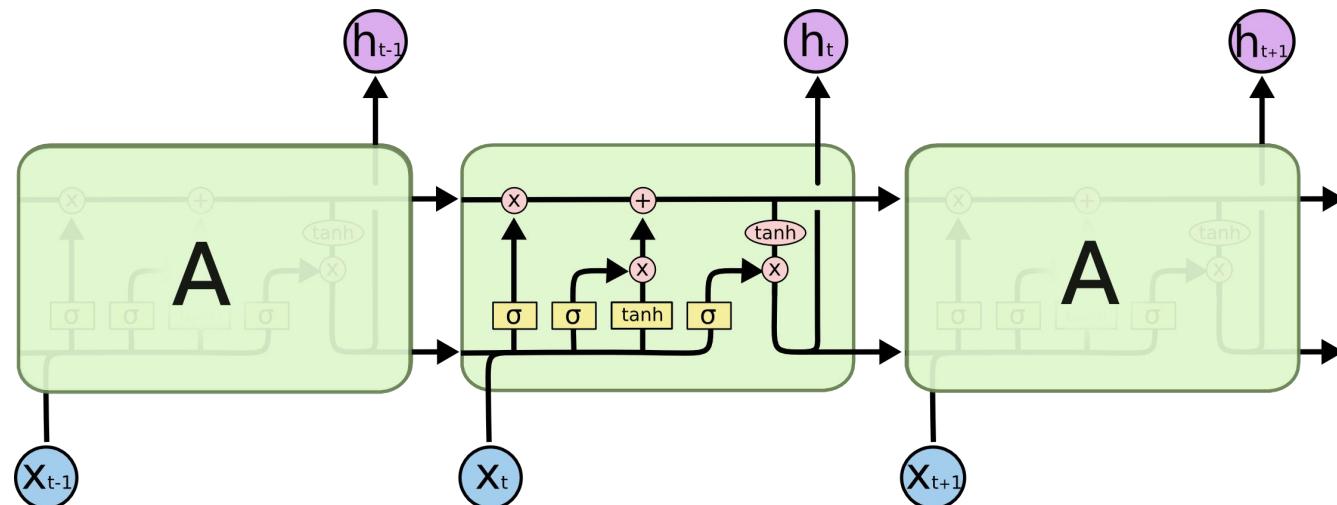
- (1) 将输入序列按时间步 (time step) 逐个送入网络；
- (2) 在每个时间步，当前输入与上一个时间步的隐藏状态一起传入，计算当前隐藏状态；
- (3) 隐藏状态在时间上传递，实现对历史信息的记忆；
- (4) 根据当前隐藏状态输出预测结果（可为每步输出，也可最后输出）；
- (5) 使用反向传播通过时间 (BPTT) 计算梯度并更新参数，训练整个网络。

ML：数据预处理 → 特征工程 → 模型选择 → 训练与评估 → 部署。

扩展阅读资料：<https://www.geeksforgeeks.org/machine-learning/introduction-to-recurrent-neural-network/>

▶ 主要网络结构：LSTM

- LSTM 是一种改进的循环神经网络（RNN），专门为解决**长期依赖**问题设计，通过引入“门机制”来控制信息的保留与遗忘，能有效处理长序列数据
- 细胞状态（cell state）像传送带一样贯穿整个序列，**有效记忆关键内容**。



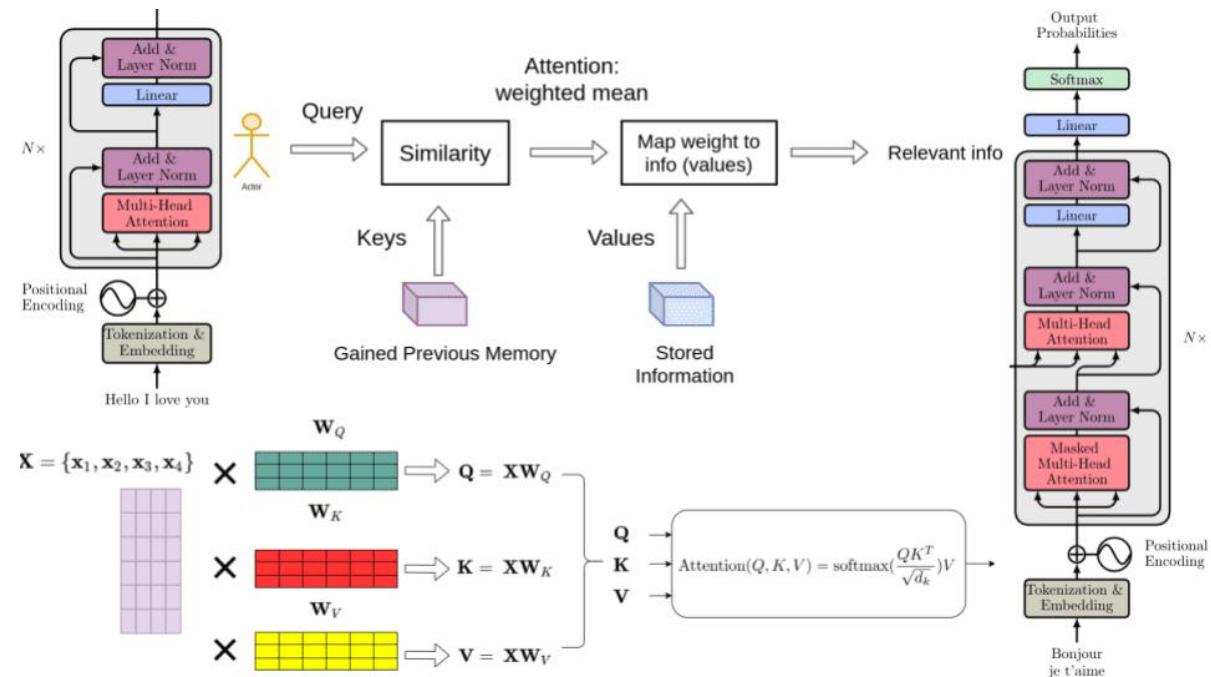
算法步骤：

- (1) 对每个时间步的输入，先通过**遗忘门**决定要丢弃多少旧的记忆；
- (2) 再通过**输入门**确定要保留多少当前输入的信息，并更新**记忆单元**；
- (3) 计算**候选记忆值**，融合到当前状态，形成新的**cell state**（记忆）；
- (4) 通过**输出门**控制将当前状态映射为**隐藏状态**（用于后续时间步或输出）；
- (5) 重复上述过程，并使用**BPTT**（时间反向传播）训练各个门的权重参数。

相比RNN，LSTM 更稳定、训练更容易，在语言模型、序列预测等任务中表现优越

▶ 主要网络结构：Transformer

- Transformer是一种基于**注意力机制**的神经网络架构，用于处理序列建模任务，如自然语言处理。
- 它完全摒弃了循环结构，依靠**多头注意力和位置编码**，高效**并行处理**长序列。



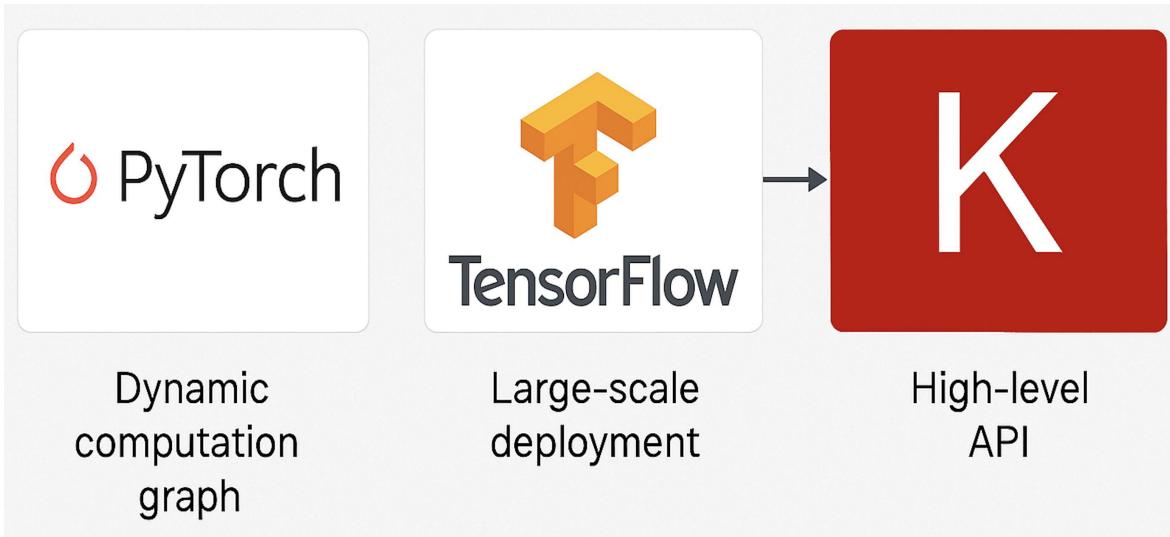
算法步骤：

- (1) 输入序列首先通过嵌入层和位置编码处理，获得带位置信息的向量；
- (2) 编码器将输入通过多个**多头自注意力 + 前馈网络**叠加处理，输出上下文表示；
- (3) 解码器每一步读取上文信息，利用掩蔽自注意力机制防止信息泄露；
- (4) 解码器还结合编码器输出，通过**交叉注意力机制**生成预测；
- (5) 最终输出通过线性层 + Softmax 得到每一步的预测结果（如翻译的下一个词）。

Transformer 模型可扩展性强，是 GPT、BERT 等大型预训练语言模型的基础。

► 现代DL框架：PyTorch、TensorFlow、Keras

- PyTorch、TensorFlow 和 Keras 是当前主流的**深度学习框架**，用于构建、训练和部署神经网络。

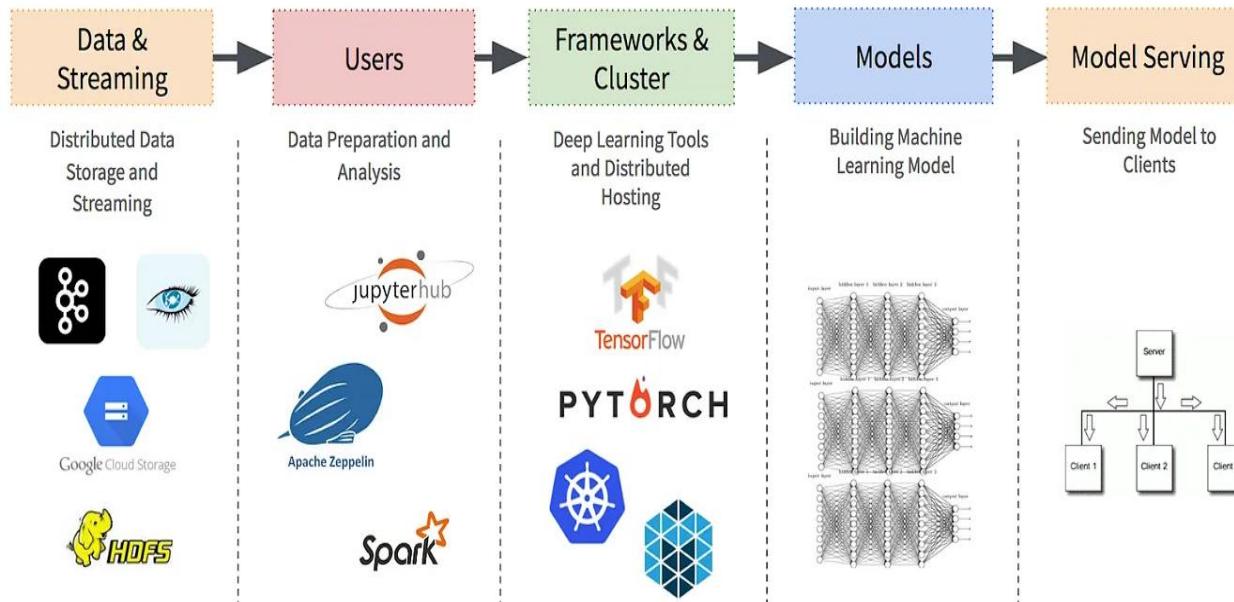


- PyTorch 以**动态图机制**著称，代码简洁、调试方便，深受研究人员欢迎。
- TensorFlow 功能强大，适合**大规模部署**，广泛用于工业界与生产环境
- Keras 是**高级封装**接口，简洁易用，初学者友好，现作为 TensorFlow 的一部分

各有特点，广泛应用于学术研究、工业实践和 AI 产品开发中。

► 现代DL框架：PyTorch、TensorFlow、Keras

- PyTorch、TensorFlow 和 Keras 是当前主流的**深度学习框架**，用于构建、训练和部署神经网络。



如何使用？

- (1) 定义模型结构（如使用类、函数式 API 或 Sequential）；
- (2) 选择损失函数和优化器（如 CrossEntropyLoss + Adam）；
- (3) 准备数据集并构建 DataLoader 或数据管道；
- (4) 在训练循环中执行前向传播、计算损失、反向传播和优化步骤；
- (5) 评估模型性能，并根据需要保存、部署或调优。

各有特点，广泛应用于学术研究、工业实践和 AI 产品开发中。

扩展阅读资料：<https://medium.com/in-pursuit-of-artificial-intelligence/choosing-a-deep-learning-framework-5669a85ebc3f>

目录章节

CONTENTS

01

导言：AI 是什么？

02

机器学习（ML）

03

深度学习（DL）

04

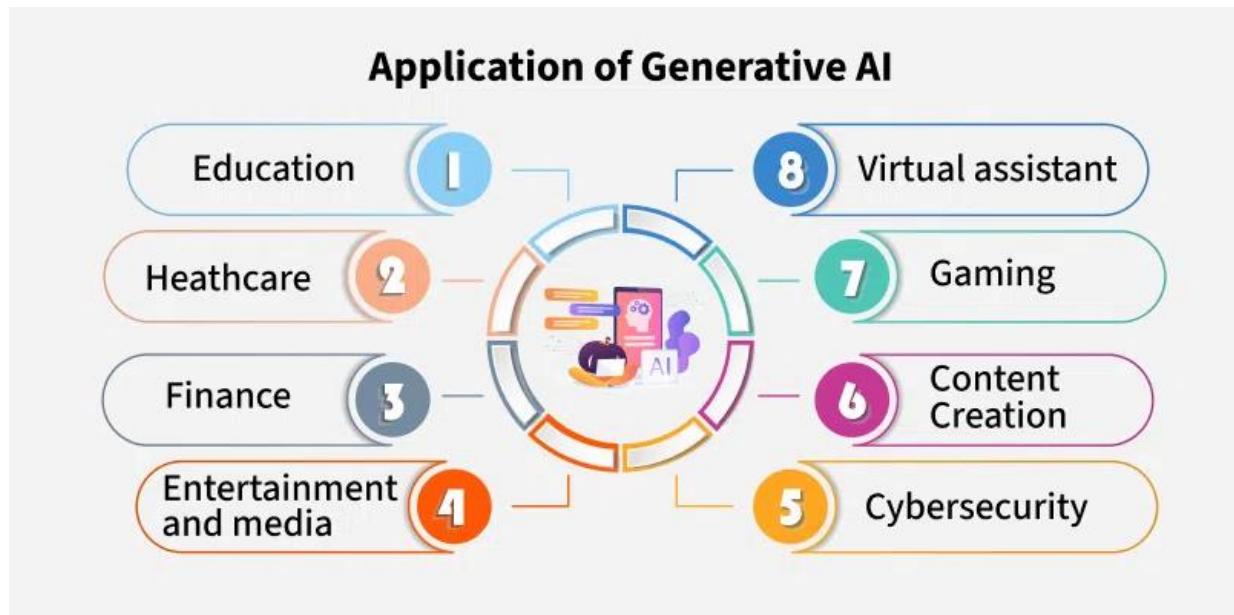
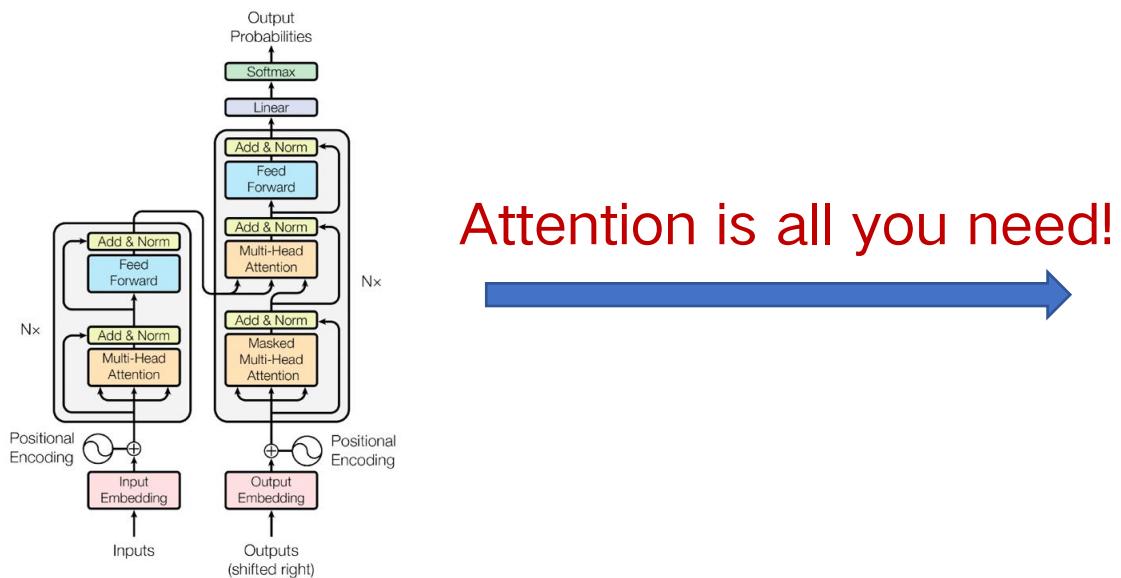
生成式 AI 与大模型

05

总结

▶ 什么是生成式 AI (Generative AI)

- 生成式 AI 是指能根据输入数据自动生成新内容的人工智能技术，核心是学习大量数据中的模式，从而能够“模仿”人类进行内容创作
- 背后的核心模型包括 Transformer、扩散模型、GAN 等。

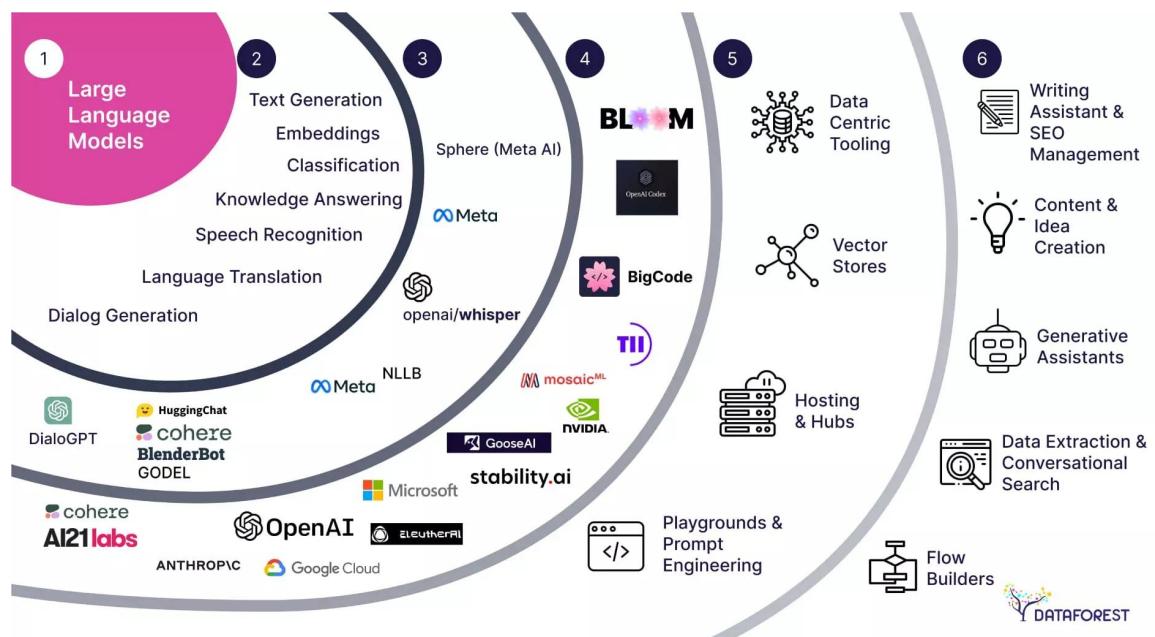


各有特点，广泛应用于学术研究、工业实践和 AI 产品开发中。

扩展阅读资料：<https://www.geeksforgeeks.org/artificial-intelligence/generative-ai-applications/>

▶ 大语言模型 (Large Language Model, LLM)

- 大语言模型是一类基于深度神经网络、训练于海量文本数据的自然语言处理模型，它能够理解、生成和推理自然语言，是生成式 AI 的核心技术之一。
- LLM 通过“自回归”或“自编码”机制预测语言中下一个词或填补空白，实现语言建模能力，其核心结构是 Transformer，通过自注意力机制捕捉语言中的复杂关系。

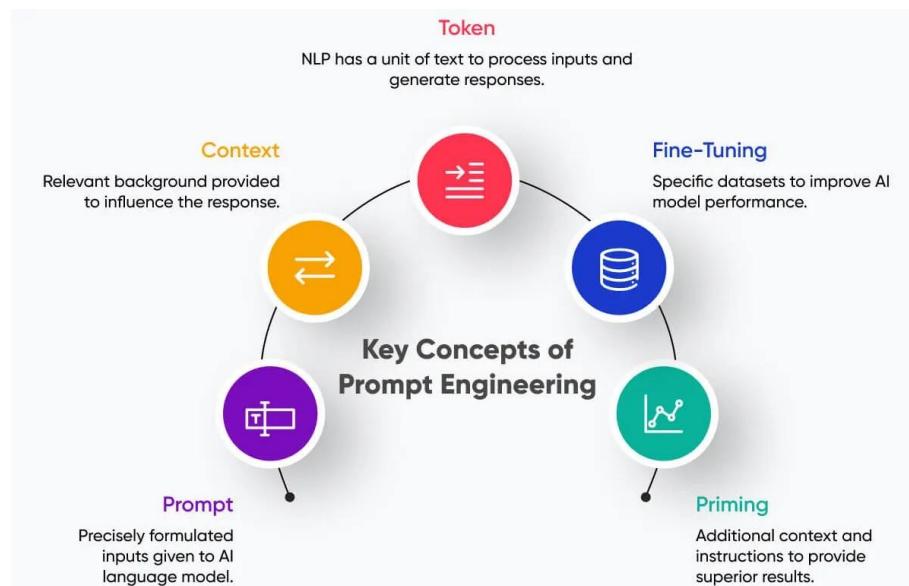


LLM 的泛化能力强，可用于以下的任务：
问答；翻译；写作；代码、图片、音频等生成；
推理

大语言模型就像一个训练过亿本书的‘超级语言引擎’，它不懂世界，但它懂语言的逻辑与表达方式。

▶ Token 与提示工程（Prompt Engineering）

- Token 是模型理解和生成文本的基本单位，是模型内部的“**语言原子**”，文本输入在进入模型前会被分词器（Tokenizer）拆分为一串 Token，并在输出时重组为可读文本。
- 提示工程 是指有意识地设计和**优化**输入提示（Prompt），以引导大语言模型生成更准确、有用、可靠的输出。



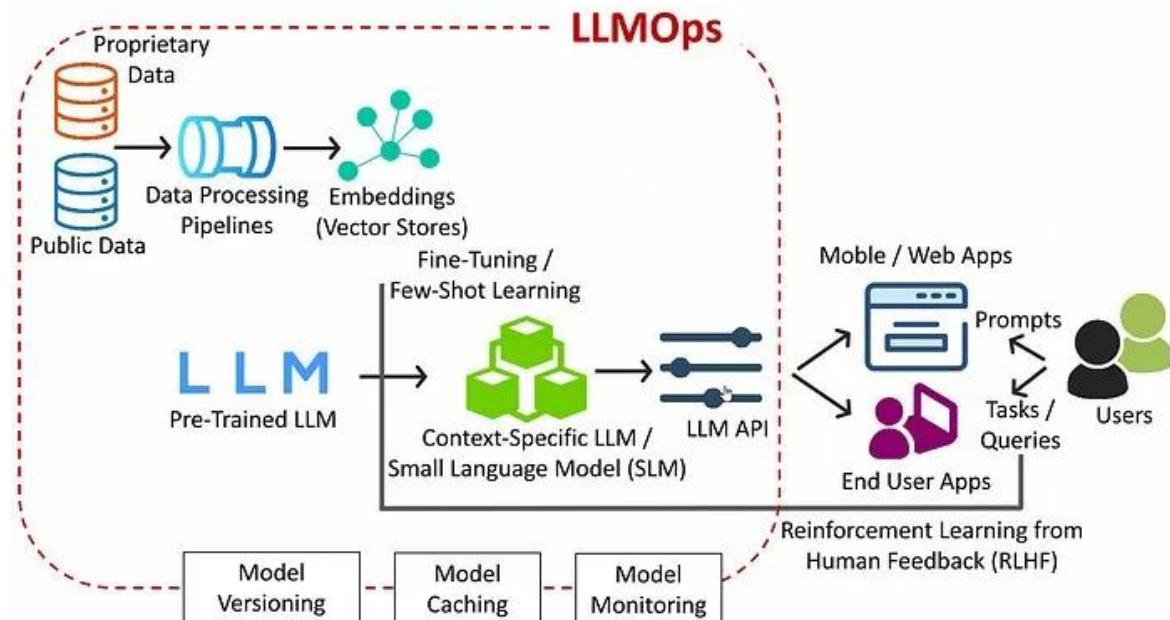
- 请帮我执行翻译任务，我目前进行的任务是英译中：
- 输入英文句子： "ChatGPT is smart."
- Token 序列： ["ChatGPT", " is", " smart", "."]
- Embedding 向量： [[0.1, ..., 0.1], [0.2, ..., 0.2], [0.3, ..., 0.3], [0.4, ..., 0.4]]
- · · · · · LLM模型（例如：Encoder + Decoder） · · · · ·
- Encoder Hidden States: [[0.5, ..., 0.5], [0.6, ..., 0.6], [0.7, ..., 0.7], [0.8, ..., 0.8], [0.9, ..., 0.9]]
- Token 序列： ["人工", "智能", "很", "聪明", "。"]
- 输出中文句子： "人工智能很聪明。"

Token 是语言模型理解文本的最小单位，提示工程是引导模型生成高质量输出的关键技术。

扩展阅读资料：<https://bhavikjikadara.medium.com/what-is-prompt-engineering-how-to-write-effective-ai-prompts-d1a253aac4ae>

► 大语言模型（LLM）训练流程

- 大语言模型的训练流程包括“**预训练 + 微调**”，从通用语言理解到特定任务适配。
- 通过大规模语料学习语言规律，再在特定任务或指令数据上精调模型行为。



LLM训练流程：

- (1) **数据准备**：收集海量高质量文本（如维基百科、书籍、网页）并进行清洗与分词；
- (2) **预训练（Self-supervised）**：通过自监督方式（如 Mask 或 Next Token Prediction）训练模型理解语言结构；
- (3) **监督微调（Fine-tuning）**：在人工标注的任务数据（如摘要、问答）上继续训练，提升特定能力；
- (4) **对齐训练（RLHF）**：通过人类反馈（如评分偏好）结合强化学习优化输出风格与安全性；
- (5) **部署与持续学习**：将模型压缩优化后部署到实际应用中，部分模型也支持在线微调或增量学习。

各有特点，广泛应用于学术研究、工业实践和 AI 产品开发中。

扩展阅读资料：<https://medium.com/@plthiyagu/comparing-llm-serving-frameworks-llmops-f02505864754>

目录章节

CONTENTS

01

导言：AI 是什么？

02

机器学习（ML）

03

深度学习（DL）

04

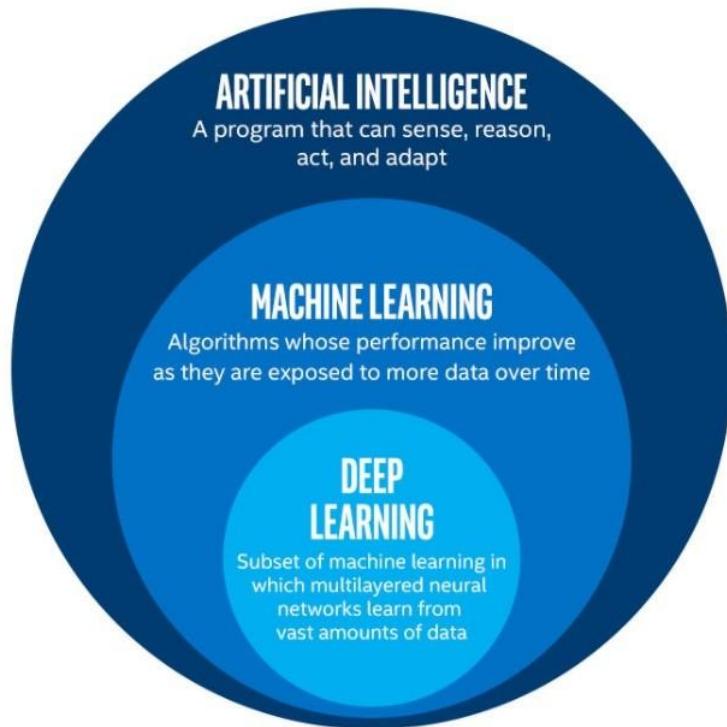
生成式 AI 与大模型

05

总结

► 总结：AI、ML、DL

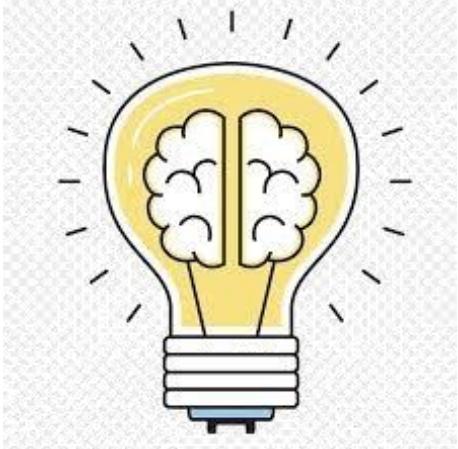
➤ 人工智能是目标，机器学习是实现路径，深度学习是当前最强的工具。



模型	定义	技术特点	应用	应用场景
人工智能 (AI)	使计算机模拟人类智能的广义科学和技术	包含规则系统、逻辑推理、专家系统、搜索算法、语音识别、推荐系统、规划、知识表示等	规则引擎	游戏AI
机器学习 (ML)	AI 的子领域，基于数据自动学习和改进	依赖数据驱动，通过训练模型实现模式识别	监督学习、无监督学分类、回归、聚类、强化学习	异常检测
深度学习 (DL)	机器学习的一个分支，使用多层神经网络进行特征抽象和学习	从数据中自动学习复杂模式，是当前 AI 的核心驱动力。	CNN、RNN、Transformer	图像识别、语音识别、自然语言处理

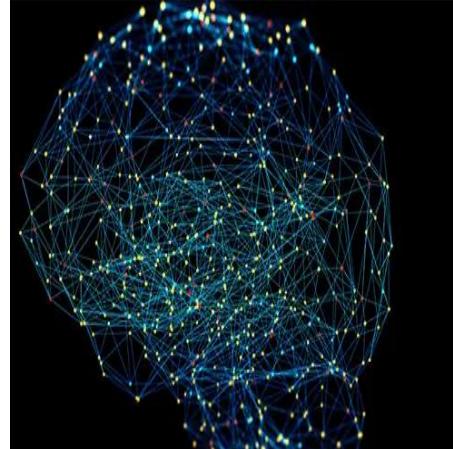
从规则驱动到数据驱动，再到模型自学习，AI 正在重塑各行各业。

▶ 总结：生成式AI、LLM、Token、Prompt



生成式AI

让机器不仅理解世界，还能创作内容，是AI从感知走向创造的关键一步。



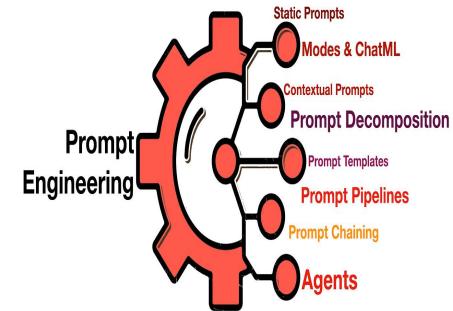
LLM

通过大规模文本训练，LLM能理解上下文并生成高质量语言，成为生成式AI的大脑。



Token

Token是模型处理语言的基本单位，决定了输入输出的长度、效率和成本。



Prompt

好提示能激发模型的最佳表现，提示工程是与大模型高效沟通的艺术与技术。

生成式AI借助大语言模型，以Token为语言单位，通过提示工程实现高效智能生成。

► 进阶学习

CS229: Machine Learning Summer 2025

Course Schedule (June – August 2025)

Note: This schedule is tentative and subject to change.

Date	Session	Topic	Details
June 24, 2025	Lecture 1	Intro; Linear Regression; Least Squares, Gradient Descent	Released: Problem Set 0 (NOT GRADED)
June 26, 2025	Lecture 2	Assessing Performance: Error Metrics, Overfitting, Bias-Variance Tradeoff	
June 27, 2025	CA Lecture 1	Linear Algebra Review	Release: Problem Set 1
June 30, 2025	CA Lecture 2	Probability Review	Moved to Monday due to July 4 Holiday
July 1, 2025	Lecture 3	Regularization; Ridge Regression; LASSO; Validation Sets, Cross Validation	Problem Set 0 Solutions Released
July 3, 2025	Lecture 4	Linear classifiers; logistic regression	
July 4, 2025	US Holiday	No CA Lecture	
July 8, 2025	Lecture 5	Generalized Linear Models; SGD	
July 10, 2025	Lecture 6	Neural Networks I	

6.S897 | Spring 2019 | Graduate

Machine Learning For Healthcare

- Syllabus
- Calendar
- Readings
- Lecture Notes
- Lecture Videos
- Projects

Course Description

This course introduces students to machine learning in healthcare, including the nature of clinical data and the use of machine learning for risk stratification, disease progression modeling, precision medicine, diagnosis, subtype discovery, and improving clinical workflows.

Course Info

INSTRUCTORS	TOPICS
Prof. Peter Szolovits	Engineering
Prof. David Sontag	Computer Science
	Artificial Intelligence
	Human-Computer Interfaces
	Health and Medicine

DEPARTMENTS

Electrical Engineering and Computer Science
Health Sciences and Technology

LEARNING RESOURCE TYPES

Lecture Videos Lecture Notes Projects



While it might seem futuristic, machine learning for healthcare is rapidly evolving.
(Courtesy of future agenda on Flickr. Used under CC BY-NC-SA.)

[Download Course](#)

CS224N Home Coursework Schedule Office Hours Final projects Lecture Videos Ed Forum

Schedule

Updated lecture [slides](#) will be posted here shortly before each lecture. Other links contain last year's slides, which are mostly similar.

Lecture [notes](#) will be uploaded a few days after most lectures. The notes (which cover approximately the first half of the course content) give supplementary detail beyond the lectures.

Disclaimer: Schedule is tentative and subject to change!

Disclaimer: Assignments change; please do not do old assignments. We will give no points for doing last year's assignments.

Date	Description	Course Materials	Events	Deadlines
Week 1	Word Vectors	[slides] [notes]	Suggested Readings: 1. Efficient Estimation of Word Representations in Vector Space [original word2vec paper]	Assignment 1 out
Tue Jun 7			2. Distributed Representations of Words and Phrases and their Compositionalities (negative sampling paper)	
Thu Jun 9	Word Vectors and Language Models	[slides] [notes] [code]	Suggested Readings: 1. Glove: Global Vectors for Word Representation (original glove paper) 2. Improving Distributional Similarity with Lessons Learned from Word Embeddings 3. Evaluation methods for unsupervised word embeddings	
Fri Jun 10	Python Review Session	[slides] [code]	Addition Readings: 1. A Latent Variable Model Approach to PMI-based Word Embeddings 2. Linear Algebraic Structure of Word Series, with Applications to Polynomials 3. On the Dimensionality of Word Embedding	Time 1:30pm–2:20pm Location Gates B01
Week 2	Backpropagation and Neural Network Basics	[slides] [notes]	Suggested Readings: 1. matrix calculus notes 2. Review of differential calculus 3. CS231n notes on network architectures 4. CS231n notes on backprop 5. Derivatives, Backpropagation, and Vectorization 6. Learning Representations by Backpropagating Errors (seminar)	Assignment 2 out [code] [handout] [latex template]
Tue Jun 14				Assignment 1 due

CS 285 at UC Berkeley

Deep Reinforcement Learning

Lectures: Mon/Wed 5-6:30 p.m., Wheeler 212

Homeworks

See [Syllabus](#) for more information (including rough schedule).

[Homework 1: Imitation Learning](#)

[Homework 2: Policy Gradients](#)

[Homework 3: Q-learning and Actor-Critic Algorithms](#)

[Homework 4: Model-Based Reinforcement Learning](#)

[Homework 5: Exploration and Offline Reinforcement Learning](#)

Lecture Slides

See [Syllabus](#) for more information.

[Lecture 1: Introduction and Course Overview](#)

[Lecture 2: Supervised Learning of Behaviors](#)

[Lecture 3: PyTorch Tutorial](#)

[Lecture 4: Introduction to Reinforcement Learning](#)

[Lecture 5: Policy Gradients](#)

[Lecture 6: Actor-Critic Algorithms](#)

[Lecture 7: Value Function Methods](#)

[Lecture 8: Deep RL with Q-Functions](#)

[Lecture 9: Advanced Policy Gradients](#)

[Lecture 10: Optimal Control and Planning](#)

[Lecture 11: Model-Based Reinforcement Learning](#)

[Lecture 12: Model-Based Policy Learning](#)

[Lecture 13: Exploration \(Part 1\)](#)

[Lecture 14: Exploration \(Part 2\)](#)

[Lecture 15: Offline Reinforcement Learning \(Part 1\)](#)

[Lecture 16: Offline Reinforcement Learning \(Part 2\)](#)

[Lecture 17: Reinforcement Learning Theory Basics](#)

[Lecture 18: Variational Inference and Generative Models](#)

[Lecture 19: Connection between Inference and Control](#)

[Lecture 20: Inverse Reinforcement Learning](#)

[Lecture 21: RL with Sequence Models](#)

[Lecture 22: Meta-Learning and Transfer Learning](#)

[Lecture 23: Challenges and Open Problems](#)

感谢聆听



Personal Website: <https://www.miaopeng.info/>



Email: miaopeng@stu.scu.edu.cn



Github: <https://github.com/MMeowhite>



Youtube: <https://www.youtube.com/@pengmiao-bmm>