

# 第 0 章：AI 底层逻辑——听懂这些，你就懂了 AI 的核心

本章目标：学完后你将能够——

- 用通俗语言解释 AI 的底层原理
- 说出为什么跑 AI 需要显卡
- 理解 Token 是什么、怎么产生的
- 掌握 AI、大模型、Token 之间的关系

## 第一部分：AI 到底是什么？

### 1.1 一句话定义

AI（人工智能）= 让机器模仿人类智能的技术。

比喻：AI 就像一个“超级模仿者”——你给它看很多例子，它学着做。

### 1.2 传统软件 vs AI

对比	传统软件	AI
工作方式	人告诉机器“规则”	机器从数据中“学会”规则
典型例子	计算器：2+3=5 操作系统	人脸识别：可能是张三（95%）
能不能处理没见过的情況	不能（规则没写就不会）	能（基于相似性推测）
答案是否确定	确定	不确定（概率）
适应场景	规则明确、流程固定	模糊、复杂、需泛化

金句：传统软件是“告诉机器怎么做”，AI 是“让机器自己学会怎么做”。

### 1.3 人类、传统软件、AI 对比

AI 在学习方式和错误表现上更像人类，而非传统软件——它会学习、会犯错、难解释，这是智能进化的必然代价。

维度	人类	传统软件	AI
学习方式	经验+反思	预设规则	数据驱动
错误类型	认知偏差、疏忽	仅由代码缺陷导致	数据偏见、过拟合、对抗攻击
可解释性	强（可自述）	极强（可追踪）	弱（黑盒）
适应性	极强（举一反三）	无（需重编程）	强（基于相似性泛化）

#### 1. 如何学会一件事？

**人类：**通过观察、实践与反思积累经验。比如学骑车，摔几次后身体就“记住”了平衡感。

**传统软件：**完全依赖程序员预先写好的规则。没有“学习”过程，只是精确执行指令。

**AI：**从海量数据中自动提取规律，像学生做题总结方法。例如，AI 看十万张猫图后，自己归纳出“尖耳朵+胡须=猫”。

□ **关键差异：**人类和 AI 都具备经验驱动的学习能力，而传统软件是“规则驱动”的执行机器。

#### 2. 会不会犯错？

**人类：**会犯错，但能从错误中反思改进。比如医生误诊后会复盘调整诊断思路。

**传统软件：**只要代码无漏洞，不会出错。2+2 永远等于 4，结果可重复验证。

**AI:** 会犯错，且错误常带有“合理性”——比如把考拉识别成熊，因为它学到了“毛茸茸+圆耳朵”的模式，但泛化过度。

□ 风险提示：**AI** 的错误不像程序崩溃那样明显，而是“看似合理实则错误”，更难察觉。

### 3. 能不能解释“为什么”？

**人类:** 能清晰讲述决策逻辑。“我判断是肺炎，因为患者有高烧、咳嗽，CT 显示肺部阴影。”

**传统软件:** 可追溯每一步执行路径。日志能精确显示“第 5 行代码触发了报警”。

**AI:** 很难解释，尤其是深度神经网络，被称为“黑盒”模型。它可能说“这张图 95% 是猫”，但说不出具体依据哪个特征做判断。

🔍 这正是当前 **AI** 研究的核心挑战之一：可解释性。医疗、金融等高风险领域尤其需要“透明决策”。

## 第二部分：AI 的三大要素——数据、算力、算法

一、数据是“生产原料”（如对数据进行标注：青菜、番茄、萝卜）  
AI 的学习过程依赖大量真实、高质量的数据。就像人通过阅读和经历积累经验，AI 通过数据“见多识广”。数据的多样性与准确性，直接决定了模型的认知边界和判断能力。例如，医疗 AI 若仅用欧美人群数据训练，在亚洲人群中可能表现不佳，这正是“数据偏态”带来的局限。

二、算法是“生产工艺”（就产生了如：文字模型 图片模型 视频模型 多模态）

算法决定了如何从数据中提取规律，是 AI 的“大脑”。从早期的决策树到如今的深度神经网络、Transformer 架构，算法的演进让 AI 能处理更复杂的任务，如自然语言理解、图像生成等。2022 年 ChatGPT 的爆发，正是算法突破的集中体现。

三、算力是“生产设备”（锅碗瓢盆 燃气灶）

没有强大的算力支撑，再好的算法也无法运行。GPU、TPU 等芯片和云计算平台，为千亿参数模型的训练与推理提供“超级引擎”。近年来，国产算力加速发展，智算中心建设提速，为大模型落地提供了坚实底座。

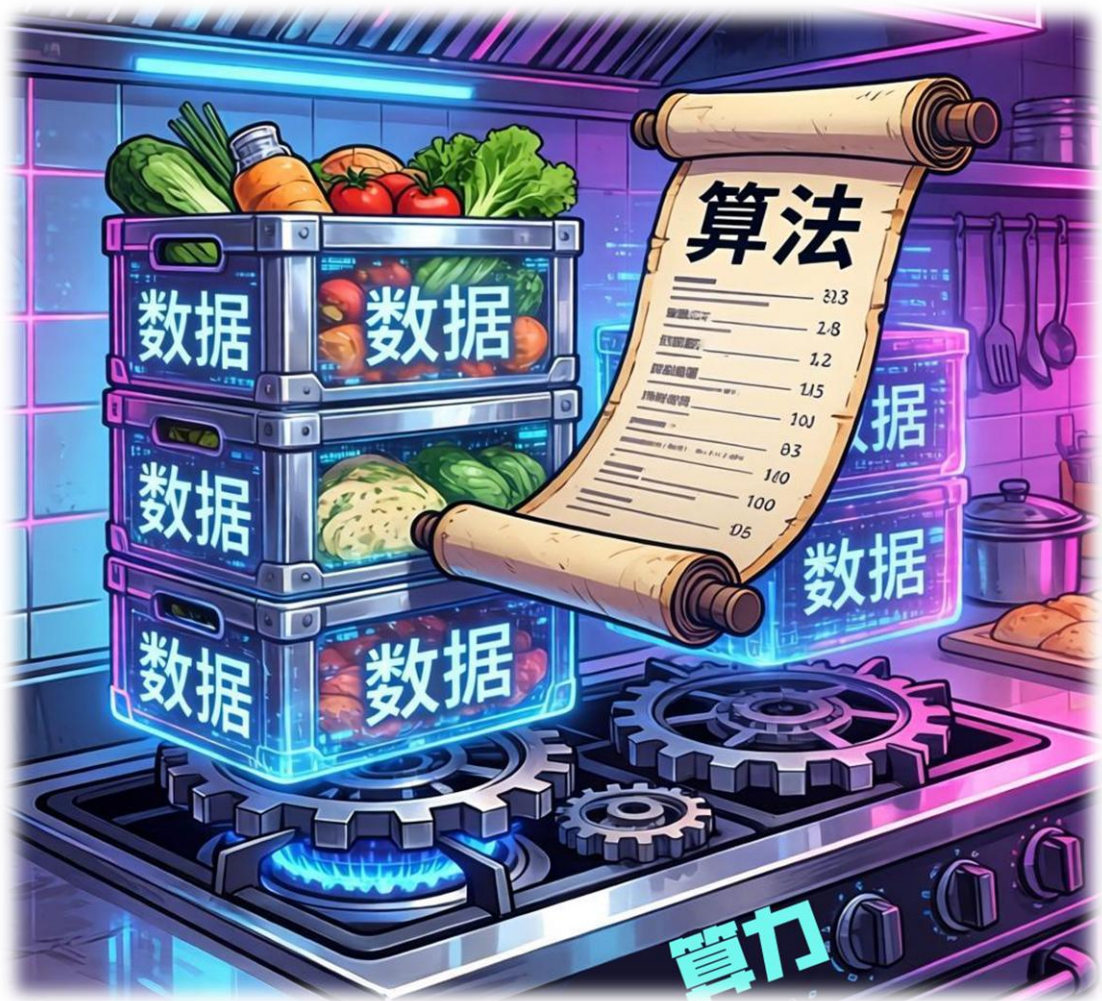
★ 一个形象比喻：

数据 是食材，

算法 是菜谱，

算力 是灶火与厨具。

三者齐备，才能“烹饪”出真正智能的 AI。



1. **数据**：如同烹饪的食材。没有足够多样、高质量的数据，AI 无法学会规律。
2. **算法**：相当于“菜谱”，指导 AI 如何从数据中提取知识，比如深度学习模型。
3. **算力**：好比灶台和锅具，GPU 等硬件提供强大计算能力，支撑模型训练。

这三者结合，就像用优质食材、秘制配方和高性能灶具，才能做出一道“AI 大餐”。

要素	比喻	说明
数据	原材料 食材	AI 学习的“教材”，没有数据 AI 什么都学不会
算力	引擎	计算能力，决定 AI 学习的速度
算法	菜谱 厨艺 方向	学习方法，决定 AI 往哪个方向学

金句：数据是“燃料”，算力是“引擎”，算法是“方向盘”。三者缺一不可。

## 第三部分：为什么跑 AI 需要显卡？CPU 为什么慢？

### 3.1 核心区别：CPU vs GPU

维度	CPU（中央处理器）	GPU（图形处理器）
核心数量	少（4-16 个）	多（几千个）
每个核心能力	强（能做复杂计算）	弱（只能做简单计算）
适合任务	复杂逻辑、顺序处理	大量简单计算、并行处理
比喻	一个教授	几千个小学生

比喻：CPU 像一个“教授”，什么都会，但只有一个。GPU 像“几千个小学生”，每个人只会简单的算术，但可以一起算。

### 3.2 为什么 AI 需要 GPU？

AI 训练的核心是“矩阵运算”——大量、重复、简单的乘法和加法。

任务	需要什么能力	谁更擅长
矩阵乘法（几万次）	大量简单计算	☐ GPU
判断“如果是这样，那么那样”	复杂逻辑	☐ CPU
训练神经网络	大量简单计算	☐ GPU
运行操作系统	复杂逻辑	☐ CPU

金句：CPU 是“教授”，擅长解决复杂问题；GPU 是“小学生军团”，擅长一起算简单算术。AI 训练就是大量简单算术，所以 GPU 更快。

CPU 与 GPU 的核心差异在于设计哲学：CPU 是“精兵作战”，擅长复杂逻辑；GPU 是“人海战术”，专攻海量简单任务的并行处理。

### 3.3 速度对比

任务	CPU	GPU	差距
训练一个小模型	几天	几小时	10-50 倍
训练大模型	几年	几周	100 倍+
推理（用 AI）	几秒	毫秒	10-100 倍

这就是为什么你手机上的 AI 助手响应那么快——背后是云端的 GPU 在算。

## 第四部分：什么是 Token？

### 4.1 Token 的定义

**Token** = AI 处理文字的最小单位。

语言	1 个 Token 大约对应	例子
英文	1 个单词 $\approx$ 1.3 个 Token	"Hello world" $\approx$ 2-3 个 Token
中文	1 个常用汉字 $<$ 1 个 Token	“你好” $\approx$ 1-2 个 Token

比喻：Token 就像“乐高积木”。AI 把文字拆成小积木，然后重新组合。

### 4.2 Token 是怎么产生的？

步骤：

1. 把文字拆成小块（Tokenization，分词）
2. 给每个 Token 编号（ID）
3. AI 用这些编号进行计算

例子：“我爱 AI”

步骤	结果
原始文字	我 爱 AI
分词	“我” + “爱” + “AI”
Token ID	[1245, 3872, 8921]

### 4.3 为什么需要 Token?

原因	说明
AI 不认识文字	AI 只认识数字，需要把文字转成数字
控制计算量	每个字都算会太慢，合理的分词让计算高效
统一处理	中文、英文、代码都转成 Token，AI 统一处理

## 第五部分：大模型与 Token 的关系

### 5.1 大模型是什么？

大模型 = 在海量 Token 上训练出来的超大型神经网络。

模型	训练用的 Token 量	参数量	备注
GPT-3	3000 亿 Token	1750 亿	—
GPT-4	13 万亿 Token	约 1.8 万亿	—
DeepSeek-V3	14.8 万亿 Token	激活参数 370 亿, MoE 架构)	用更少的计算跑出更强的效果

金句：Token 是“食物”，大模型是“吃了这些食物长大的巨人”。

## 5.2 大模型怎么“理解”你的话？

text

你的输入 → 拆成 Token → 转成数字 → 大模型计算 → 预测下一个 Token → 生成回答

例子：你输入“今天天气”

步骤	发生了什么
1	“今天天气”被拆成 Token
2	大模型根据学过的规律，预测下一个 Token 最可能是“怎么样”或“不错”
3	然后预测再下一个，直到生成完整句子

这就是 AI“生成”答案的原理——它不是“知道”答案，而是“猜”出最可能的下一个 Token。

## 5.3 上下文长度（Context Length）

概念	说明	比喻
上下文长度	AI 一次能“记住”的 Token 数量	短期记忆容量
128K	约一本《三体》的厚度	能记住一本书
1M（100万）	约《三体》三部曲	能记住三本书

金句：上下文越长，AI“记性”越好，聊久了也不会忘。

## 第六部分：AI 为什么现在才爆发？

条件	什么时候成熟	说明
数据	2010 年代	互联网爆炸，海量数据可用
算力	2010 年代	GPU 从游戏卡变成 AI 卡
算法	2012 年	AlexNet 图像识别夺冠（深度学习视觉突破）
	2017 年	Transformer 突破（大语言模型基石）

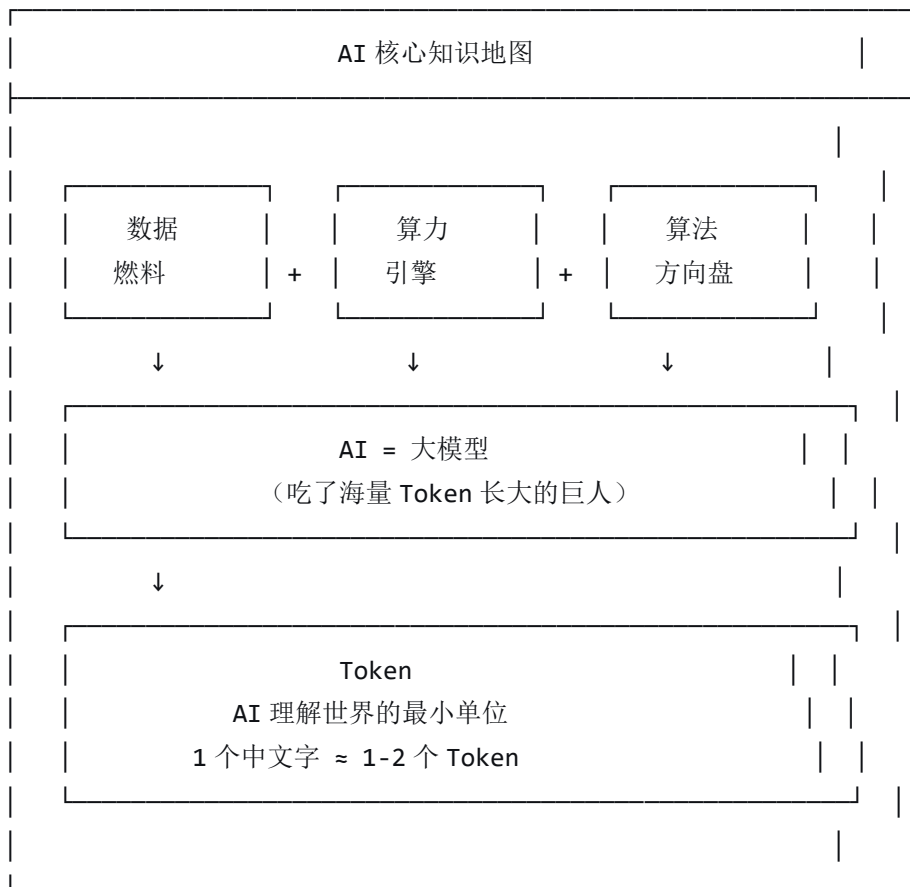
### 6.1 三大条件交汇

金句：AI 爆发，不是一夜之间，而是等了 50 年，等三个条件同时成熟。

### 6.2 时间线

时期	AI 状态	为什么
1980-2000	寒冬	数据少、算力弱、算法不行
2000-2010	萌芽	有数据了，但算力和算法还不够
2010-2015	起步	GPU 成熟，深度学习突破
2015-2020	成长	大模型开始出现
2020-今	爆发	三者全部成熟

## 第七部分：一张图总结



## 本章金句

“传统软件是‘告诉机器怎么做’，AI 是‘让机器自己学会怎么做’。”

“CPU 是教授，GPU 是小学生军团——训练 AI 需要几千个小学生一起算。”

“Token 是 AI 的‘乐高积木’，AI 把文字拆成积木，再重新组合。”

“AI 不是‘知道’答案，而是‘猜’出最可能的下一个 Token。”

“AI 爆发，等了 50 年，等数据、算力、算法三个条件同时成熟。”

## 课后作业

**作业 1：**用“教授 vs 小学生军团”的比喻，向朋友解释为什么跑 AI 需要显卡。

**作业 2：**打开 DeepSeek，问它“你知道什么是 Token 吗？用通俗语言解释给我听。”对比它的回答和本章的解释。

**作业 3：**找一段 100 字的中文，数一下大约有多少个 Token（提示：1 个中文字  $\approx$  1-2 个 Token）。

**作业 4（选做）：**查一下你常用的 AI 模型的上下文长度是多少（如 DeepSeek 是 128K，Kimi 是 1M），想想这能装下多少文字。

“现在你理解了 AI 的底层逻辑——它怎么‘想’、怎么‘学’、为什么需要显卡、Token 是什么。从下一章开始，我们要回答一个更重要的问题：这些跟你有什么关系？你会看到 2026 年的职场正在发生什么，以及为什么现在学 AI 不是‘加分项’，而是‘生存项’。”