

LLM 成本建模、Token 量预测与业务价值估算

Contents

LLM 成本建模、Token 量预测与业务价值估算1

1. 每百万 Token 的成本建模1

2. Token 量预测2

3. 业务价值估算3

4. 落地实践清单4

5. 常见陷阱4

6. 一页纸速查公式4

LLM 成本建模、Token 量预测与业务价值估算

围绕 LLM 应用落地中最核心的三个问题：1. 每百万 Token 的成本如何建模？ 2. 如何预测未来 Token 消耗量？ 3. 如何估算对应的业务价值？

输出语言：中文 | 适用读者：AI 产品经理、解决方案架构师、FinOps、ToB 销售

1. 每百万 Token 的成本建模

1.1 单价的拆解

LLM 厂商普遍以 “每百万 Token (per 1M tokens)” 为计费单位。完整单价应拆为四层：

层级	组成	示例（以 Claude Opus 5 为参考）
输入 Token 单价	提示词（Prompt）+ 上下文缓存命中/未命中	缓存写入 vs 缓存读取价差可达 5-10 倍
输出 Token 单价	模型生成内容	通常是输入的 3-5 倍
系统开销	工具调用、Function Calling、Tool Use	按调用次数或隐含在输出 Token 中
附加费用	微调托管、批处理折扣、长上下文溢价>200K 上下文通常 +50%~100%	

基础公式：

单次请求成本 C_req

= (N_in × P_in + N_cached_in × P_cached_in + N_out × P_out) × (1 + α_overhead)

其中：- N_in = 输入 Token 数（首次/未缓存） - N_cached_in = 缓存命中的输入 Token 数 - N_out = 输出 Token 数 - α_overhead = 系统/工具调用附加比例（典型 5%-15%）

1.2 平均成本（Blended Cost）

业务上更常用 加权平均单价（Blended Price per 1M）：

Blended Price = (Σ 请求级别成本) / (Σ Token 数) × 1,000,000

可通过 A/B 抽样 1% 真实流量得到稳定的 blended 值，用于月度复盘。

1.3 自托管 / 私有化场景

若使用自托管模型：

$$C_{total} = C_{gpu} \times N_{gpu} + C_{storage} + C_{network} + C_{ops} + C_{amortized}$$

- C_{gpu} ：按 H100/H200 时租价估算（云上 ~\$2-4/hr）
- 单 Token 成本 ≈ GPU 总时价 ÷ (GPU 数 × 每秒 Token 吞吐 × 3600 × 30)

1.4 成本敏感度的边际分析

杠杆	影响幅度
切换到缓存读取	降本 80%–90%
压缩 Prompt（去冗余 / Few-shot 优化）	降本 30%–60%
模型分层（小模型先过滤 → 大模型兜底）	降本 50%–70%
输出长度控制（max_tokens + Stop Sequence）	降本 20%–40%
批处理 / 异步任务	降本 10%–30%

2. Token 量预测

2.1 自顶向下 (Top-Down)

适合规划期、缺历史数据时：

$$T_{total} = U_{users} \times S_{sessions} \times Q_{requests} \times (L_{in} + L_{out})$$

- U_{users} ：活跃用户数（DAU/MAU）
- $S_{sessions}$ ：人均日会话数
- $Q_{requests}$ ：每次会话请求数
- L_{in}, L_{out} ：单次请求输入/输出 Token 数（按业务类型取中位数）

行业基准参考：

场景	单次请求输入	单次请求输出
客服问答	800–2,000	200–500
文档摘要（10 页）	4,000–8,000	300–800
代码助手	1,500–4,000	300–1,500
RAG 检索增强	3,000–10,000	400–1,200
长文档分析（100 页+）	30,000–80,000	1,000–3,000

2.2 自底向上 (Bottom-Up)

适合已有 PoC 数据：

1. **样本采集**：跑 1,000–10,000 条真实请求，统计 N_{in} 、 N_{out} 的 P50 / P90 / P99
2. **建立分布**：用对数正态分布拟合，避免被极长尾拉高均值
3. **业务增长系数**：乘以用户增长率 × 渗透率 × 功能使用频率
4. **敏感性分析**：对 L_{out} 单独建上限场景（如创意生成场景输出可能 10×）

2.3 时间序列预测

有 ≥ 3 个月数据时，可叠加：

- **季节性分解**：工作日 vs 周末、白天 vs 夜间
- **ARIMA / Prophet**：适合月级别容量规划
- **LLM 自身预测**：用历史调用日志作为 Prompt，让模型估算未来 30 天用量（适合早期）

2.4 关键校准因子

因子	说明
Prompt 模板膨胀 长上下文功能开放 缓存命中率 工具调用膨胀	业务迭代会让 Prompt 自然增长 10%–30% / 季度 一旦上线 >200K 上下文，平均输入可能瞬时翻倍 命中后输入成本骤降，但 Token 总量不变，需分开记账 Agent 类应用一次”任务”可能触发 5–20 次模型调用

3. 业务价值估算

3.1 价值的两条主线

维度	含义	度量
收入侧（Revenue Lift）	AI 直接带来的新增营收	客单价提升、转化率提升、新 SKU
效率侧（Cost Avoidance）	AI 替代/加速人力产生的节省	节省人时 $\times$ 时薪

3.2 收入侧建模

$$\Delta \text{Revenue} = (\text{Conversion_old} \times \text{ARPU_old}) \rightarrow (\text{Conversion_new} \times \text{ARPU_new}) \times \text{Volume}$$

或者更精细：

$$\text{AI 收入贡献} = \text{触达用户数} \times \text{AI 触达率} \times \text{转化率提升} \times \text{客单价}$$

示例：电商导购助手 - 日活 100 万，AI 触达 30% - 转化率从 2.0% $\rightarrow$ 2.4% (+0.4pp) - 客单价 \$50 - 日增收
 $= 1,000,000 \times 30\% \times 0.4\% \times \$50 = \$60,000/\text{天}$

3.3 效率侧建模

$$\text{节省人时} = \Sigma (\text{任务量} \times \text{自动化率} \times \text{单任务耗时})$$

$$\text{节省金额} = \text{节省人时} \times \text{全负载小时薪资} \times \text{业务负载率}$$

示例：客服工单处理 - 月工单 50,000 - AI 自动化率 40% - 单工单处理 15 分钟 - 客服月薪 ¥10,000，月有效工时 160h - 月节省
 $= 50,000 \times 40\% \times 15\text{min} \div 60 \div 160 \times ¥10,000 \approx ¥31,250$

3.4 综合 ROI 模型

$$\text{ROI} = (\text{Value_Revenue} + \text{Value_Efficiency} - \text{Cost_LLM} - \text{Cost_Infra} - \text{Cost_Ops}) / (\text{Cost_LLM} + \text{Cost_Infra} + \text{Cost_Ops})$$

更稳健的版本用 NPV（净现值），考虑：-12–24 个月的生命周期 - 12%–15% 的折现率 - 模型迭代带来的边际成本下降（每年 -20%–30%）

3.5 价值可信度分级

级别	证据强度	何时使用
L1 假设值	仅基于行业 benchmark	立项 / 内部讨论
L2 试点值	PoC / 灰度数据	立项答辩
L3 上线值	生产 30 天以上数据	预算评审
L4 验证值	A/B 实验 + 因果推断	大规模投入决策

3.6 反向校验：价值上限

业务价值的天花板 = 用户愿意为该场景支付的最高单价 × 体量。一旦 ROI 估算出的“价值”超过这个上限，几乎一定是估算口径出错，而非 AI 真的如此万能。

4. 落地实践清单

4.1 第 0 周：基线测量

- ☐ 抽取 1,000 条真实请求，统计 P50/P90/P99 Token 数
- ☐ 锁定 2-3 个主流使用场景的 Blended Price
- ☐ 与业务方对齐“成功”的具体定义（哪条 KPI？提升多少？）

4.2 第 1-4 周：PoC 与试点

- ☐ 用 Top-Down + Bottom-Up 双轨估算 Token 量级
- ☐ 上线用量看板（按场景 / 模型 / 用户分桶）
- ☐ 设置月度成本告警（>预算 110% 触发）

4.3 第 2-3 月：价值闭环

- ☐ 灰度 A/B 实验，输出 L3 上线值
- ☐ 建立“每 \$1 Token 成本 → 多少业务价值”的核心指标
- ☐ 季度复盘：模型分层、Prompt 优化、缓存策略

4.4 长期：组织能力

- ☐ **FinOps 角色**：每月出成本-价值对账
- ☐ **Prompt Owner**：每个场景一个负责人，背负 P90 Token
- ☐ **模型选型委员会**：季度评估，平衡能力 / 成本 / 延迟

5. 常见陷阱

1. 只算 API 成本，忽略工程成本：向量库、检索、日志、Guardrail 可能占 30%+。
2. 用输入 Token 算成本：输出通常更贵，且 Agent 场景输出膨胀极快。
3. 用 P50 估算总体成本：P99 / 长尾才是成本失控的来源。
4. 业务价值与 Token 成本未对齐口径：前者按月，后者按天，难以决策。
5. 忽略缓存命中率”假降本”：缓存是技术优化，不是业务增长。

6. 一页纸速查公式

单价 (per 1M tokens)

$\text{Blended_Price} = \Sigma(C_{\text{req}}) / \Sigma(N_{\text{in}} + N_{\text{out}}) \times 1e6$

月度成本

$$\text{Monthly_Cost} = U \times S \times Q \times (L_{\text{in}} \times P_{\text{in}} + L_{\text{out}} \times P_{\text{out}})$$

Token 量

$$T_{\text{month}} = \text{DAU} \times \text{Sessions_per_DAU} \times \text{Req_per_Session} \times \text{Avg_Tokens_per_Req} \times 30$$

业务价值

$$\text{Value} = \Delta \text{Revenue} + \text{Saved_Hours} \times \text{Loaded_Hourly_Cost}$$

ROI

$$\text{ROI} = (\text{Value} - \text{Total_Cost}) / \text{Total_Cost}$$

文档版本: v1.0 | 2026-09-14