从语言模型到大语言模型的演进

2025 年 10 月 23 日

1 语言模型的定义与目标函数

语言模型通过估计文本序列 $\mathbf{x} = (x_1, \ldots, x_T)$ 的概率来度量语句的合理性，常用的因式分解为

$$p_\theta(\mathbf{x}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}), \tag{1}$$

其中 $x_{<t}$ 表示历史上下文。最大似然训练最小化负对数似然：

$$\mathcal{L}(\theta) = -\sum_{\mathbf{x}\in\mathcal{D}} \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}). \tag{2}$$

困惑度（perplexity）是常见评估指标，

$$\mathrm{PPL}(\mathcal{D}) = \exp\left(-\frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} \frac{1}{T}\log p_\theta(\mathbf{x})\right), \tag{3}$$

值越小说明模型越善于预测。为了防止过拟合，实践中常结合 dropout、标签平滑、梯度裁剪等正则化手段。语言建模本质上是自监督学习，仅依赖原始文本即可构建训练信号；额外的对比目标（如下一句预测、句子排序）有助于丰富语义约束。

2 N-gram、RNN 到 Transformer 的演化

2.1 n 元统计模型

n 元模型遵循低阶马尔可夫假设：

$$p(x_t \mid x_{1:t-1}) \approx p(x_t \mid x_{t-n+1:t-1}), \tag{4}$$

基于计数构造条件概率。Kneser–Ney、Good–Turing 等平滑策略可缓解稀疏，但由于上下文长度受限且参数量随 n 指数增长，难以捕获长距离依赖。

2.2 神经语言模型与循环网络

神经语言模型引入分布式词向量和非线性组合。循环神经网络（RNN）通过隐藏状态 $\mathbf{h}_t = f_\theta(x_t, \mathbf{h}_{t-1})$ 汇总历史，LSTM/GRU 进一步通过门控结构（输入门、遗忘门、输出门）控制信息流：

$$\mathbf{i}_t = \sigma(\mathbf{W}_i x_t + \mathbf{U}_i \mathbf{h}_{t-1}), \tag{5}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f x_t + \mathbf{U}_f \mathbf{h}_{t-1}), \tag{6}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c x_t + \mathbf{U}_c \mathbf{h}_{t-1}). \tag{7}$$

与 n 元模型相比，RNN 能捕获更长依赖，但其串行结构导致训练与推理难以并行，并且面对特别长的文本仍会退化。

2.3 注意力机制与 Transformer

Transformer 使用自注意力替代递归，任意两个位置之间都可以直接交互。对查询、键、值矩阵 $\mathbf{Q}$、$\mathbf{K}$、$\mathbf{V}$，缩放点积注意力为

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}. \tag{8}$$

多头注意力、残差连接、层归一化等模块让模型既能捕获全局依赖又易于并行。相对位置编码、稀疏注意力、FlashAttention 等改进进一步拓展序列长度，为大规模预训练奠定基础。

3 自回归（AR）与自编码（AE）的区别

3.1 自回归建模

自回归模型（GPT 系列）沿时间方向逐 token 预测，用因果掩码保证仅依赖历史信息。其优点是训练稳定、推理自然，尤其适合对话、续写、代码等文本生成任务；但训练与推理之间存在暴露偏差，且无法在建模时使用右侧上下文。

3.2 自编码建模

自编码模型（BERT 系列）通过掩码、删除等方式破坏输入，再恢复原始 token。典型的掩码语言模型损失为

$$\mathcal{L}_{\mathrm{MLM}} = -\mathbb{E}_{\mathbf{x},\mathbf{m}} \sum_{t \in \mathbf{m}} \log p_\theta(x_t \mid \mathbf{x}_{\setminus \mathbf{m}}), \tag{9}$$

使模型获得双向语义表征，适合分类、阅读理解、序列标注等理解类任务。然而，直接用于流畅生成较困难，通常需引入 encoder-decoder 架构或迭代修正策略。

3.3 混合范式

序列到序列 Transformer、MASS、BART、T5 等方法结合 AR 与 AE 优势：编码端吸收双向上下文，解码端按自回归方式生成。Prefix LM、UniLM 通过灵活掩码在同一模型内支持理解与生成。

4 GPT 与 BERT 的基本思想比较

4.1 模型结构与训练目标

- **GPT**：采用解码器堆叠和因果掩码，目标是下一个 token 预测；训练语料覆盖海量网页、书籍和代码，强调长文本生成。
- **BERT**：采用编码器堆叠和双向注意力，主要任务是掩码语言模型与句子级对比任务（NSP、SOP）；聚焦于高质量语义特征。

4.2 下游应用方式

GPT 常通过 prompt、few-shot/in-context learning、指令微调以及 RLHF 等方式完成摘要、对话、创作等生成任务；借助工具调用、检索增强等技术，其功能不断扩展。BERT 系列通常在特定任务上添加轻量分类头或进行全量微调，用于分类、问答、实体识别、句子匹配等理解任务。

4.3 规模化策略与演化

GPT 路线沿着参数、数据、计算同步扩展，衍生出 GPT-3、GPT-4、PaLM、LLaMA、Mixtral 等模型，并引入 MoE、检索增强、插件生态。BERT 路线产生了 RoBERTa、DeBERTa、ELECTRA、SpanBERT 等改进版本，从目标设计、预训练语料、多模态扩展等方面持续演化。

5 工程实践提示

- **数据治理**：进行去重、质量过滤、多语言平衡，可提升收敛稳定性并降低隐私风险。
- **训练优化**：使用混合精度、梯度检查点、ZeRO、流水线并行来控制显存与吞吐；学习率热身、余弦退火、动量校正常与之配合。
- **评估与安全**：除 GLUE、SuperGLUE、MMLU、BIG-Bench 等传统指标外，还需关注幻觉、偏见、毒性等安全属性，确保可部署性。

延伸阅读

- Jurafsky & Martin：《Speech and Language Processing》。
- Bengio 等：《A Neural Probabilistic Language Model》，JMLR 2003。
- Vaswani 等：《Attention is All You Need》，NeurIPS 2017。
- Devlin 等：《BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding》，NAACL 2019。
- Kaplan 等：《Scaling Laws for Neural Language Models》，2020。