语言模型训练目标：自回归、掩码建模与优化采样实践

2025 年 10 月 25 日

1 自回归语言建模（Causal LM Loss）

1.1 目标函数与信息建模

自回归语言模型（AR LM）假设序列生成服从链式法则，将句子 $x_{1:T}$ 的联合概率分解为条件概率乘积：

$$p_\theta(x_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}). \tag{1}$$

训练目标是最小化负对数似然（等价于最大化对数似然）：

$$\mathcal{L}_{\mathrm{AR}}(\theta) = -\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}) = \sum_{t=1}^{T} \mathrm{CE}\big(\delta_{x_t}, \hat{p}_\theta(\cdot \mid x_{<t})\big), \tag{2}$$

其中 δ_{x_t} 为目标词的 one-hot 分布，$\hat{p}_\theta$ 是模型在 softmax 输出空间的预测。该损失直接对模型的预测分布施加监督，使其拟合语料的条件熵结构。

AR 框架天然适配解码器式 Transformer：通过上三角掩码维持自回归约束，并借助 KV Cache 实现高效推理。具体实现中常见的两种策略是：

- **Teacher Forcing**：在训练时全部使用真实历史 token 作为条件输入，保持梯度估计无偏且便于大规模并行化。
- **串行采样**：在推理时递归采样，前一步输出即为下一步的条件，模型面对暴露偏差（Exposure Bias）挑战。

1.2 序列长度与上下文建模

长上下文训练需要处理梯度截断、显存瓶颈和注意力复杂度。常用的配套技术包括：

- **梯度累计（Gradient Accumulation）**：在较短序列上累计多次反向传播再更新参数，以近似长上下文梯度。

- **记忆复用（Memory Replay）**：Transformer-XL、GPT-NeoX 等模型使用缓冲区拼接前一批次的隐状态，延伸有效上下文。
- **位置外推（RoPE、ALiBi）**：专用位置编码使模型能够在训练长度之外稳定推理。

此外，跨语种建模或多任务训练需要共享词表与归一化策略，以减小条件分布漂移。

1.3 对比学习与正则化增强

纯粹的语言建模损失往往侧重流畅度，难以掌握事实性与约束性。常见增强策略：

- **对比约束（Contrastive Loss）**：在语言建模之外，增加噪声样本的对比项，压缩错误预测空间。
- **正则化项**：标签平滑（Label Smoothing）、变长截断、随机 DropToken 等方法提高泛化，限制模型过拟合高频模式。
- **课程学习**：先在短文本、低温采样上训练，再逐步扩展复杂度，帮助模型稳定收敛。

2 掩码语言建模（Masked LM Loss）

2.1 双向信息建模

掩码语言模型（MLM）通过掩蔽局部片段，让模型在双向上下文条件下预测缺失 token。对输入序列 $x_{1:T}$ 引入掩码集合 $\mathcal{M}$，训练目标为：

$$\mathcal{L}_{\text{MLM}}(\theta) = -\sum_{t\in\mathcal{M}} \log p_\theta(x_t \mid x_{\setminus\mathcal{M}}). \tag{3}$$

这种方式保留了自编码器（encoder-only）结构的并行优势，适合理解类任务（分类、问答、序列标注等）。然而，MLM 在推理时缺乏直接的生成机制，通常需要结合附加头或解码器。

2.2 掩码策略设计

掩码位置的选择直接影响语义覆盖率与训练效率。主流策略：

- **随机掩码**：BERT 经典方案，以 80% 替换为 [MASK]、10% 替换为随机词、10% 保留原词，提升多样性。
- **Whole Word Masking**：针对中文或词粒度应用，将一个词的所有子词同时掩码，保持语义完整性。

- **Span Masking**：如 SpanBERT、T5，按片段掩码以建模长距离依赖，对生成和抽取任务均有帮助。
- **动态掩码**：每个 epoch 重新采样掩码位置，使模型见到的上下文组合更加多样。

2.3 扩展任务与联合训练

为了缓解预训练任务与下游任务的差距（pretrain-finetune mismatch），MLM 常与其他预训练目标联合：

- **下一句预测（NSP）/句子顺序预测（SOP）**：强化句间关系建模，适用于段落级推理。
- **替换词检测（RTD）**：ELECTRA 将判别任务融入预训练，以更低计算量学习高质量表示。
- **多任务混合**：结合监督信号（QA、翻译、摘要），形成统一的指令或 span 级框架（如 T5 的 text-to-text）。

在多语言或跨模态场景下，MLM 可扩展为掩码语音/图像建模（HuBERT、MAE），共享统一的掩码重建范式。

3 Tokenization（BPE, SentencePiece, tiktoken）

3.1 分词器设计原则

Tokenization 决定序列长度、稀疏度和词表规模。理想的分词方案需要在以下维度取得平衡：

- **覆盖度**：词表应能重构语料，避免过多未登录词（UNK）。
- **压缩比**：过大的词表增加 embedding 和 softmax 参数；过小则导致序列冗长。
- **跨语种适配**：兼容多语言字符集，兼顾偏旁部首、音标等粒度。

现代 LLM 普遍采用子词（Subword）粒度，以处理开放词汇问题。

3.2 字节对编码（BPE）

BPE 从字符级词表出发，迭代合并出现频率最高的 token 对 (u, v)，将其加入词表：

1. 初始化词表为字符（或字节）集合。
2. 统计所有相邻 token 对的出现频率。

3. 合并频率最高的对，生成新 token，并替换语料中的对应片段。

4. 重复步骤 2–3，直到达到预设词表大小。

优点包括可控的词表规模、良好的跨词缀泛化能力，以及对低频词的鲁棒性。BPE 适合拼写规则明确的语言，对中文等无空格语言通常先进行分词或直接基于字节对。

3.3 SentencePiece 与 Unigram LM

SentencePiece 提供无监督的子词建模工具，支持 BPE 与 Unigram 语言模型。Unigram LM 基于概率模型选择最优子词集合，通过 EM 算法迭代：

- 赋予候选子词集合初始概率，利用前向后向算法计算句子的分词概率。
- 修剪低概率子词，重新归一化，直至达到目标词表规模。

SentencePiece 的关键优势是无需预分词，直接在原始字符串上构建词表，天然兼容多语言和特殊符号（emoji、标点）。Google T5、mT5、ALBERT 等均采用该方案。

3.4 tiktoken 与现代实现

tiktoken 是 OpenAI 针对 GPT 系列推出的高性能分词库，特点包括：

- **字节级回退**：词表缺失的 token 会自动降级为字节序列，确保编码无信息损失。
- **稀疏矩阵优化**：使用 Patricia Trie 等结构实现快速匹配，显著提升编码速度。
- **兼容性**：预置多种模型词表（`gpt-4`, `cl100k_base` 等），便于推理与微调一致性。

在训练流水线中，分词器需与数据清洗、序列截断、特殊符号策略协同设计，如添加角色标记（<|assistant|>）、系统提示等。

4 优化与采样策略（Teacher Forcing, Top-k, Top-p）

4.1 Teacher Forcing 与暴露偏差

Teacher Forcing 在训练中使用真实标签作为下一步输入，使损失计算并行且梯度稳定。然而，推理阶段模型必须依赖自身预测，引入分布漂移。常见缓解方案：

- **Scheduled Sampling**：按概率混合真实 token 与模型预测，逐步过渡到生成式条件。
- **Professor Forcing/Adversarial Training**：引入判别器约束训练轨迹与生成轨迹的隐状态分布一致。
- **强化学习微调**：使用奖励建模（如 RLHF）在生成策略上直接优化目标。

4.2 Top-k 采样

Top-k 采样从概率分布中筛选前 k 个最有可能的 token，然后在归一化后的子分布中随机抽样：

$$\mathcal{V}_k = \{x \mid p_\theta(x \mid x_{<t}) \text{ 排名前} k\}, \quad p'(x) = \frac{p_\theta(x \mid x_{<t})}{\sum_{y \in \mathcal{V}_k} p_\theta(y \mid x_{<t})}. \tag{4}$$

较小的 k 提供更高的语言质量但降低多样性，较大的 k 有助于探索，但可能产生不连贯文本。应用时常与温度（Temperature）缩放组合：

$$p_\tau(x) = \frac{\exp(\log p_\theta(x)/\tau)}{\sum_y \exp(\log p_\theta(y)/\tau)}. \tag{5}$$

4.3 Top-p（Nucleus）采样

Top-p 采样根据累积概率选择最小集合 $\mathcal{V}_p$ 满足

$$\mathcal{V}_p = \left\{ x \mid \sum_{y \in \mathcal{V}_p} p_\theta(y \mid x_{<t}) \geq p \right\}, \tag{6}$$

在该集合内归一化后采样。Top-p 自适应调整候选集规模，能够在不同困惑度（Perplexity）段保持稳定质量。实践中常设置 $p \in [0.8, 0.95]$，并结合最小/最大候选数约束避免极端情况。

4.4 温度、惩罚与多样性控制

除了 Top-k、Top-p，还可利用以下手段控制生成：

- **重复惩罚（Repetition Penalty）**：对已经生成的 token 施加惩罚系数 γ，避免循环模式。
- **频率/存在惩罚（frequency/presence penalty）**：OpenAI API 中常用，分别基于出现次数和是否出现调整 logits。
- **对比解码（Contrastive Decoding）**：结合小模型得分过滤低质量 token，实现更加精确的语义控制。

多样性与准确性往往相互制约，需要根据任务需求（对话、写作、代码生成等）调节参数。

5 工程实践建议

- **数据工程**：训练目标与分词器设计需与语料分布匹配，确保跨域泛化和毒性控制。

- **混合精度与优化器：** AdamW、Lion 等自适应优化器配合梯度裁剪、EMA 能够提升收敛稳定性，混合精度加速同时保持损失尺度。
- **对齐与评估：**自回归/掩码模型的评估指标（困惑度、精确率、BLEU、ROUGE）应针对不同目标设计统一验证集；采样策略需在人工测评中调整，以平衡创造力和安全性。

延伸阅读

- Bengio et al. “A Neural Probabilistic Language Model.” Journal of Machine Learning Research, 2003.
- Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” NAACL 2019.
- Radford et al. “Language Models are Unsupervised Multitask Learners.” OpenAI Technical Report, 2019.
- Holtzman et al. “The Curious Case of Neural Text Degeneration.” ICLR 2020.
- Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” JMLR 2020.