大模型架构与系统优化：主流设计、归一化、激活与推理加速

2025 年 10 月 25 日

1 主流架构对比（GPT, LLaMA, Qwen, Mistral, Mixtral）

1.1 GPT 系列：解码器标准范式

OpenAI 的 GPT-3/4 延续解码器式 Transformer，采用交替的自注意力与前馈层，使用 LayerNorm + MLP 结构，并引入多头注意力与定制的字节对编码词表。关键特征：

- **高容量参数：** GPT-3（175B）使用 96 层、12288 维隐藏层；GPT-4 进一步扩展层数与多专家模块，强调对齐与多模态能力。
- **数据混合：**兼顾网页、代码、对话、专业文献，配合 RLHF 与工具集成，实现泛化能力。
- **推理策略：**强化上下文学习能力，使用较宽的上下文窗口和高效的 KV Cache 管理。

GPT 架构侧重稳定性与通用性，是后续模型的基准 baseline。

1.2 LLaMA 家族：轻量化与长上下文

Meta 的 LLaMA、LLaMA2、LLaMA3 强调数据与工程效率：

- **RMSNorm 替代 LayerNorm：**提升数值稳定，同时减少归一化开销。
- **SwiGLU 激活：**相较 GELU 提升表达能力，降低训练损失。
- **Grouped-Query Attention (GQA)：**在 LLaMA2-70B 引入组查询注意力，减少 KV Cache 存储，提升长上下文效率。

LLaMA 数据集强调开源语料 + 安全对齐，支持开源社区微调与部署。

1.3 Qwen 系列：多语言与多模态扩展

阿里巴巴的 Qwen-1.5/2.5 聚焦中文、英文双语兼容及工具使用：

- **多语言词表**：采用 1512M 词表，覆盖多语种并引入函数调用 token，便于工具链集成。
- **增强位置编码**：使用 RoPE 外推 + 动态压缩策略支持 32K 以上上下文。
- **多模态模块**：在 Qwen-VL 提供图文多模态能力，通过外部视觉编码器 + 文本解码器融合实现。

Qwen 在信息检索、代码生成等领域提供任务专项微调权重。

1.4 Mistral 与 Mixtral：高效小模型与稀疏专家

Mistral AI 推出的 Mistral 7B、Mixtral-8x7B 致力于推理效率：

- **Sliding Window Attention**： Mistral 7B 使用滑动窗口 + short/long attention 组合，在保持性能的同时降低计算成本。
- **Multi-Query + Multi-Head 混合**：减少 KV Cache，支持高吞吐生成。
- **Mixtral MoE**：采用 8 专家，每次激活 top-2 专家；共享稀疏门控提升参数利用率，实现 46.7B 总参数但 12.9B 激活参数的效率优势。

Mistral 系列在开源社区广受欢迎，易于部署和推理优化。

2 层归一化方式（LayerNorm, RMSNorm）

2.1 LayerNorm：经典方案

LayerNorm 针对每个 token 的隐藏向量执行归一化：

$$\text{LayerNorm}(h) = \frac{h - \mu}{\sqrt{\sigma^2 + \epsilon}} \odot \gamma + \beta, \tag{1}$$

其中 μ 和 σ 为维度均值和标准差，γ、β 为可训练缩放和偏移。优点在于稳定梯度，但存在：

- 计算中需开方操作，影响性能；
- 对零均值假设敏感，需配合残差技巧；
- 在低精度或大批量训练中容易出现数值抖动。

2.2 RMSNorm：简化与稳定

RMSNorm 摒弃均值项，仅使用均方根归一化：

$$\mathrm{RMSNorm}(h) = \frac{h}{\mathrm{rms}(h)} \odot \gamma, \qquad \mathrm{rms}(h) = \sqrt{\frac{1}{d}\sum_{i=1}^{d} h_i^2 + \epsilon}. \tag{2}$$

要点：

- **无偏移**：没有 β 项，减少参数与计算；
- **数值稳定**：更适合半精度与大模型；LLaMA、Mistral 等均采用；
- **预归一化**：通常在残差前应用（Pre-Norm），改善梯度流动。

实践中可根据任务需求选择，RMSNorm 在许多模型中成为默认配置。

2.3 其他归一化变体

还有如 DeepNorm（调整残差缩放以稳定深层网络）、ScaleNorm（固定范数）等。对于 MoE 架构，还会结合 Experts Normalization 以平衡不同专家输出。

3 激活函数（GELU, SwiGLU）

3.1 GELU：高斯误差线性单元

GELU 定义为：

$$\mathrm{GELU}(x) = x \cdot \Phi(x), \tag{3}$$

其中 Φ 是标准正态分布的累积分布函数。常用近似：

$$\mathrm{GELU}(x) \approx 0.5x \left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right]\right). \tag{4}$$

特点：

- 平滑开关，较 ReLU 表现更优；
- 在 BERT、GPT 等模型中表现良好；
- 计算代价适中，可使用预计算或近似加速。

3.2 SwiGLU：门控激活

SwiGLU 是 GLU（Gated Linear Unit）的改进，形式：

$$\text{SwiGLU}(x) = \text{Swish}(xW_1) \odot (xW_2), \tag{5}$$

其中 $\text{Swish}(z) = z \cdot \sigma(z)$。特点：

- 引入门控机制，增强特征选择能力；
- 在 LLaMA、PaLM 等模型中显著提升困惑度；
- 常配合 MLP 宽度放大，保持参数效率。

实际部署中需关注额外的矩阵乘法开销，可通过张量并行或混合精度处理。

3.3 激活函数选择策略

选择激活需考虑：

- **性能与复杂度**： SwiGLU 性能高但计算多；GELU 兼顾性能与效率；
- **数值稳定性**：在低精度下需评估溢出风险；
- **任务特性**：生成模型通常更偏向 SwiGLU，理解模型可保留 GELU。

同时关注实验测得的困惑度、收敛速度与梯度统计。

4 FlashAttention 与 KV Cache 优化

4.1 FlashAttention：显存与带宽优化

FlashAttention 利用块状处理和寄存器重用，在不牺牲精度的前提下实现 $O(n^2)$ 注意力的高效计算。核心思想：

- **分块 softmax**：将注意力计算拆分为适配 SRAM 的小块，避免中间结果写入 HBM。
- **并行化**：结合 CUDA 高效 kernel，实现 fused attention。
- **数值稳定**：使用在线 softmax 技术，避免因分块带来的溢出问题。

FlashAttention v2 提升对多头、分组注意力的支持，并在 Triton 等 DSL 上实现自动调优。

4.2 KV Cache 管理与压缩

自回归推理中，KV Cache 存储历史 key/value，决定推理时延和显存：

- **分块存储**：按 token 维度分块写入 GPU/CPU，配合 paged attention 降低碎片化。
- **GQA 与 MQA**：使用较少的 key/value 头共享多个 query，显著降低 KV Cache 大小（GQA）。
- **压缩技术**：采用量化（INT8/FP8）或稀疏化策略，配合注意力重建减小误差。

在 Mistral、LLaMA 等模型中，KV Cache 优化可带来 2–4 倍吞吐提升。

4.3 长上下文推理与检索增强

为支持 100K+ 上下文，需要组合多项技术：

- **位置编码外推**：使用 NTK Scaling、Dynamic NTK、XPos 等方法保持高频信息。
- **滑动窗口与回流**：避免存储全部历史 token，仅保留邻近窗口或使用摘要回流。
- **检索增强**：将长文档切分并检索相关段落，结合 RAG 减少对 KV Cache 的依赖。

推理系统需动态调度显存与带宽，结合批处理与多租户策略实现稳定服务。

参考文献

- Vaswani et al. “Attention Is All You Need.” NeurIPS, 2017.
- Touvron et al. “LLaMA: Open and Efficient Foundation Language Models.” arXiv, 2023.
- Jiang et al. “Mistral 7B.” arXiv, 2023.
- Dao et al. “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness.” NeurIPS, 2022.
- Dettmers et al. “QLoRA: Efficient Finetuning of Quantized LLMs.” arXiv, 2023.