SSM vs Transformer

杨子凡

Jul 08, 2025

1 从 Mamba 到 Attention，如何选择下一代序列建模引擎

当前大模型时代对长序列处理的需求呈指数级增长，尤其在基因组分析、语音识别和视频理解等领域。然而传统 Transformer 架构面临严峻挑战：其自注意力机制的计算复杂度随序列长度呈二次方增长，导致处理超长序列时出现显存墙问题。核心矛盾在于全局建模能力与计算效率的权衡，以及结构化先验假设与数据驱动归纳偏置的冲突。本文旨在破除「Transformer 是唯一解」的认知定式，提供可落地的技术选型框架。

2 技术深潜：SSM 与 Transformer 原理解析

2.1 Transformer 架构核心机制

Transformer 依赖自注意力机制实现全局依赖建模，其计算复杂度为 $O(N^2d)$（N 为序列长度，d 为特征维度）。位置编码技术从最初的绝对位置编码演进至旋转位置编码（RoPE），显著提升了长程依赖捕获能力。但推理过程中的 KV Cache 机制导致显存占用与序列长度线性相关，成为部署瓶颈。主流改进如稀疏注意力（Sparse Attention）通过限制注意力范围将复杂度降至 $O(N\sqrt{N})$，线性注意力（Linear Transformer）则利用核函数近似实现 $O(N)$ 复杂度，但往往牺牲建模精度。

2.2 状态空间模型 (SSM) 的革命性突破

状态空间模型将连续系统微分方程离散化处理。其数学本质可表述为：

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

其中 A, B, C, D 为可学习参数，通过零阶保持器离散化得到递归形式。结构化状态空间序列模型（S4）引入 HiPPO 理论，该理论通过勒让德多项式投影实现历史信息的最优逼近，数学表达为：

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t) \quad \text{其中} \quad A_{nk} = -\begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ n+1 & \text{if } n = k \end{cases}$$

Mamba 架构的突破在于三方面创新：首先引入输入依赖的状态转移机制，使 B, C 矩阵动态变化；其次设计硬件感知的并行扫描算法，将递归计算转化为并行操作；最后通过选择性信息传递门控实现情境感知建模。

3 全方位对比：5 大维度 PK

计算复杂度方面，Transformer 的 $O(N^2)$ 与 SSM 的 $O(N)$ 形成鲜明对比，万 token 序列下 SSM 可提速 10 倍以上。内存占用维度，Transformer 的 KV Cache 机制导致显存需求与序列长度成正比，而 SSM 仅需固定大小的状态向量。并行能力上，Transformer 训练并行但推理串行，SSM 支持训练推理全流程并行，这对实时语音处理至关重要。归纳偏置差异体现在：Transformer 依赖海量数据学习结构，SSM 内置时间连续性先验，在小样本时序预测中表现更鲁棒。当前扩展性仍是 Transformer 的优势领域，其千亿参数规模已验证，而 SSM 尚在百亿级验证阶段。

4 选型决策树：何时选择哪种架构?

选型决策需分步判断：若输入序列超过 1K token，进入因果建模需求判断。严格因果场景（如实时语音）优先选择 SSM；非因果场景则考察硬件内存限制，内存敏感场景（边缘设备）选择 SSM，否则进一步分析全局上下文需求。需全局建模的任务（如多模态理解）适用 Transformer，局部依赖任务（基因序列分析）则 SSM 性价比更高。典型场景中，SSM 在超长 1D 信号处理、低延迟语音流、内存敏感边缘计算具显著优势；Transformer 则在多模态语义对齐、复杂符号推理、小样本学习场景不可替代。

5 融合创新：混合架构前沿探索

融合架构正成为研究热点。Transformer 与 SSM 分支的混合设计（如 JetMoE）在保留全局建模能力的同时降低 40% 计算开销。Attention 矩阵的 SSM 近似方案（如 H3, Hyena）通过卷积核替代注意力实现：

```
# Hyena 算子伪代码
def hyena_operator(x, filters):
    k = generate_conv_kernel(filters) # 生成动态卷积核
    return fft_conv(x, k) # 频域卷积计算
```

系统优化层面，FlashAttention 通过 SRAM 分级存储优化注意力计算，FlashMamba 则利用并行扫描算法实现 8 倍吞吐提升。产业实践中，Mistral 的 SSM-MoE 实验显示每 token 计算量降低 60%，特斯拉车载系统采用 SSM 实现毫秒级时序预测。

6 实战建议：架构迁移指南

从 Transformer 转向 SSM 需警惕位置敏感任务（如机器翻译）的性能衰减，建议采用残差路径融合位置编码。归一化方案需重构，LayerNorm 在 SSM 中可替换为 StateNorm：

```
class StateNorm(nn.Module):
    def __init__(self, dim):
        super().__init__()
        self.gamma = nn.Parameter(torch.ones(dim))
```

```
    def forward(self, x):
        # 对状态向量进行缩放
        return x * self.gamma[None, None, :]
```

超参调优重点差异显著：Transformer 需优化注意力头数和 FFN 维度，SSM 则需调整状态维度 d_state（推荐值 16-64）和离散化步长 Δ（影响时序粒度）。部署优化时，Transformer 可采用 KV 量化和动态批处理，SSM 则可复用状态缓存并利用 CUDA 的 warp 级并行指令。

7 未来展望

理论边界亟待突破：SSM 的表示能力等价性证明近期在 LTI 系统领域取得进展，但非线性扩展仍开放。Attention 与 SSM 的泛化等价猜想（如 $\exists f:$ Attention $\cong$ SSM $\circ f$）引发热议。硬件协同创新存机遇：存内计算架构天然适配 SSM 的向量外积计算，光计算芯片的微分方程求解优势可达成纳秒级延迟。杀手级应用可能在生物计算领域爆发，AlphaFold3 已尝试 SSM 处理蛋白质折叠。万亿 token 级通用模型的架构抉择，将取决于 SSM 在 10K+ 上下文窗口的泛化能力验证。

核心洞见可总结为：「Transformer 是通用计算的 CPU，SSM 是信号处理的 DSP」。技术决策者应建立包含序列长度、延迟要求、内存预算、数据规模的四维评估矩阵，定期重验架构假设。当处理 DNA 测序等超长序列时，Mamba 的 $O(N)$ 复杂度是破局关键；但构建多模态语义系统时，Transformer 的跨模态注意力仍不可替代。最终，架构选型本质是在计算效率、建模能力、部署成本间的动态平衡。

8 附录（可选）

关键论文索引：S4（ICLR 2022）、Mamba（arXiv:2312.00752）、RWKV（NeurIPS 2023）、Griffin（arXiv:2402.19427）。代码实践推荐 causal-conv1d 库的 SSM 层实现，mamba-minimal 的 300 行参考代码值得研读。基准测试建议采用 Long Range Arena 的 Path-X 任务（序列长度 16K）。