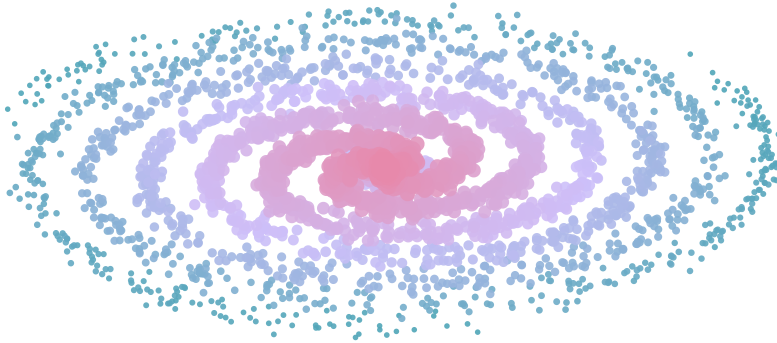


Large Model Architectures



From MoE to modern LLM systems: design choices, trade-offs, and engineering practice

AUTHOR

NJX-njx

AFFILIATION

[Hugging Face](#)

PUBLISHED

Sep. 01, 2025

Mixture of Experts : 从原理到工程落地

在大语言模型的扩展路径里，稠密模型长期遵循一条简单规律：参数更多、数据更多，效果通常更好。问题是，这条路的成本也同步上升：训练更贵、推理更慢、部署更吃硬件。

混合专家模型（Mixture of Experts, MoE）提供了另一种扩展思路：保持模型总容量很大，但在每个 token 上只激活少量参数，从而把“容量”和“计算量”部分解耦。

1) 什么是 MoE

在 Transformer 中，MoE 常见做法是把部分 FFN 层替换为“专家集合 + 路由器”：

- 专家（Experts）：多个可学习子网络（通常是 FFN）。
- 路由器（Router）：给每个 token 选择 Top-K 个专家（常见 K=1 或 K=2）。

这意味着：

- 总参数量可以很大（容量强）；
 - 每步激活参数只占一部分（计算更省）；
 - 在相同预算下，常见到更好的训练效率。
-

2) 稀疏路由与负载均衡

MoE 的核心不是“有多少专家”，而是“路由得是否均衡和稳定”。

如果大量 token 被挤到少数专家，会出现三个直接问题：

- 热门专家过载，冷门专家学不到东西；
- 通信与吞吐变差；
- 训练波动增大。

实践里通常会结合：

- 辅助负载均衡损失：鼓励 token 在专家间更均匀分布；
- 容量因子（Capacity Factor）：给每个专家设置可处理 token 上限与缓冲；
- 稳定化技巧：例如对路由 logits 的约束（如 router z-loss 思路）来抑制数值不稳定。

简化管理：MoE 的难点不是“把专家堆起来”，而是“让路由系统持续健康运行”。

3) Switch / GShard 等路线带来的启发

MoE 的工程演进里，几条经验非常关键：

- **Top-1 路由（Switch）**能显著降低路由和通信复杂度；
 - Top-2 路由在质量与稳健性上常有优势，但通信更重；
 - 容量因子不是越大越好，它在性能与通信成本之间做权衡；
 - 混合精度可提速，但路由相关计算通常需要更谨慎的数值精度控制。
-

4) 稀疏模型与稠密模型如何选择

一个实用判断框架：

- 如果你有多机多卡资源，追求高吞吐和更大规模训练，MoE 更有吸引力；
- 如果你更受限于显存、部署简单性或工具链成熟度，稠密模型更稳妥。

换句话说，MoE 往往把“算力压力”转成了“系统工程压力”。

5) transformers 里的 MoE 工程化进展

从工程视角看，MoE 进入主流框架并不只是“加一个模型类”，还涉及加载、执行和分布式机制的重构。

5.1 权重加载重构

MoE 检查点常把专家参数拆成大量独立张量；而高性能运行时通常希望专家权重按特定布局打包。新的转换式加载思路可以在加载阶段完成“源权重 -> 运行时布局”的映射，减少重复扫描与峰值内存。

5.2 专家执行后端

不同场景可选不同后端策略（如逐专家执行、批量矩阵乘、分组矩阵乘），在小 batch / 大 batch、算力受限 / 显存受限场景下有不同优势。

5.3 EXPERT PARALLELISM

当单卡放不下全部专家时，把专家沿设备切分：

- 每个设备仅持有自己负责的一部分专家；
- token 按路由分发到对应设备执行；
- 最后聚合输出。

这条并行轴让超大 MoE 在多设备上变得可落地。

6) 微调与部署：最常见的坑

微调阶段

- 稀疏模型更容易出现过拟合或训练不稳定；
- 学习率、batch size、正则化强度常需与稠密模型区别设置；
- 是否保留辅助损失，需要结合任务规模和泛化目标做实验决策。

部署阶段

- 虽然单 token 激活参数少，但通常仍需装载大规模权重；
 - 显存/带宽仍是上线门槛；
 - 蒸馏、量化、专家合并等技术常用于压缩部署成本。
-

7) 写给实践者的结论

MoE 的价值不是“参数堆得更夸张”，而是把计算预算花在更有效的位置。它非常适合追求规模化效率的团队，但要真正获得收益，必须把路由、并行、通信、加载和部署当成一个完整系统来优化。

如果把稠密模型比作“结构简单但成本线性上升”的路线，MoE 更像“系统复杂度更高、但扩展性更强”的路线。选哪条路，最终取决于你的硬件约束、团队工程能力和目标吞吐。

参考来源

- <https://huggingface.co/blog/moe-transformers#expert-parallelism>
 - <https://huggingface.co/blog/zh/moe>
-

For attribution in academic contexts, please cite this work as

NJX-njx (2025). "Large Model Architectures".

BibTeX citation

```
@misc{njx-njx2025_large_model_architectures,  
  title={Large Model Architectures},  
  author={NJX-njx},  
  year={2025},  
  doi={10.1234/abcd.efgh}  
}
```

DOI

[10.1234/abcd.efgh](https://doi.org/10.1234/abcd.efgh)

Reuse

Diagrams and text are licensed under [CC-BY 4.0](#) with the source available on [Hugging Face](#), unless noted otherwise.
Figures reused from other sources are excluded and marked in their captions ("Figure from ...").

Made with  with [research article template](#)