LLM 三种架构速览手册

从《大致了解LLM三种架构》笔记提炼 · 5 页速览

一句话主线：Transformer 引出「自注意力 + 编码器 + 解码器」，三种架构的差别，本质是「看全文 / 遮后面 / 两者拼接」。

> 本 PDF 用于验证博客内嵌 PDF 阅读功能：可翻页、缩放、缩略图跳页、全屏。

一、三种架构对照

架构	代表	擅长 / 原理 / 训练方式
Encoder-only 编码器	BERT	**擅长**：文本分析、分类、总结、判断 **原理**：全读，把一大段文字整体理解 **训练**：挖空，据上下文推出空缺句
Decoder-only 解码器	主流模型	**擅长**：推理、因果推断 **原理**：一个一个往后生成，每个 token 都含其「左边」内容 **训练**：用掩码遮住后面，再对比结果 **为何主流**：训练方式与使用方式一致，省掉额外训练成本
Encoder-Decoder 混合	—	**擅长**：强输入型输出工具 —— 需要很完备的输入材料，再产出结果

> 解编码器最核心的差别就是：**有无掩码、有无自回归。**

二、前置知识点

嵌入

文字先切成小块（**token**），经模型内部字典换成编号，最后变成一长串数（**向量**），这个过程叫嵌入。

作用：意思相近的词，点挨得近；计算机靠它区分字词意思。字典需要训练。

注意力

Q（Query 查询）、**K**（Key 键）、**V**（Value 值）。

拿 Q 与每个字的 K 比对算出相关度，归一化成加和为 100 的百分比（softmax），再把所有字的 V 混合起来得到新表示。

自注意力

序列内部自己与自己算注意力（Encoder / Decoder 都有）。

交叉注意力

拿一个序列的 Q 去查另一个序列的 K、V（编码器向解码器传递信息时用）。

自回归

把自己刚才的输出，当成下一步的输入。

三、Decoder 的细化：Dense 与 MoE

Dense

稠密型模型

让每个参数都被激活，计算复杂度与参数数量呈线性关系。

- **优点**：稳定、可预测，训练方法成熟，泛化能力强
- **缺点**：过于全面，训练成本高

MoE

混合专家模型

只激活部分参数；有一个总开关（**路由器**）决定分配给哪些专家处理，因此可拥有远多于 Dense 的参数量。

- **优点**：训练效率高，推理速度快
- **缺点**：内存需求大（所有专家都要加载）；路由造成稳定性较差、时长延误

> 趋势判断：未来 Dense 可能会与稀疏模型混合，在成本与效果之间取平衡。

四、其他提到的架构（待深入）

架构	要点
SSM 状态空间模型	有选择性机制，动态决定保留或忽略输入信息；用状态递推替代全局注意力
RNN 及其变体	递归结构，非全局注意力
RetNet 保留网络	可看作 RNN 的现代版；与 SSM 类似，同属非全局注意力路线
多模态架构	让模型同时吃各种格式的文件并相互理解（这块较为复杂，留待后续）

> **下一步**：把「掩码 / 自回归」与「路由器 / 专家」两条线索分别打通， 再回看 2.6 元婴期的 Transformer 章节，就能互相印证。