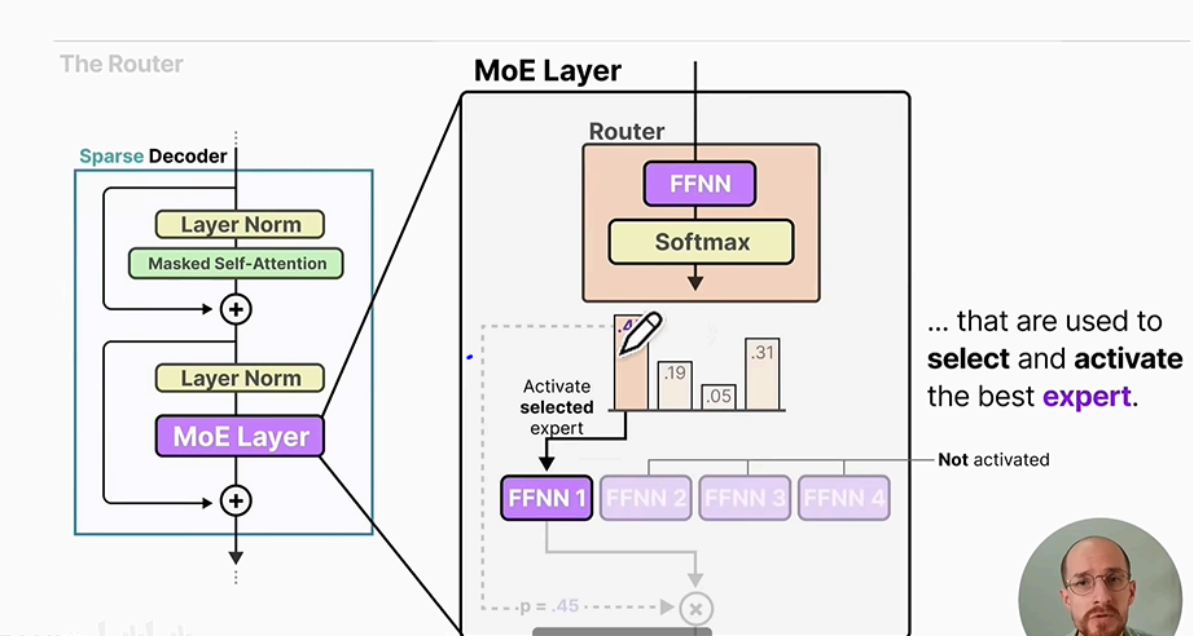
《09MOECore》解读 by【AI布道Mr.Jin】

其实在DeepSeek-R1爆火之前，DeepSeek V2在我们行业就已经妇孺皆知了，它独特的MOE结构值得研究一下。这篇文章是基于 ZOMI酱 的这个视频写的：《MOE终于迎来可视化解读！傻瓜都能看懂MoE核心原理！》。这个视频讲的很好，建议大家都学习一下！

MOE结构概述

我们可以从zomi酱视频里面的这张图开始：



MOE是mixture of experts 的缩写，简单来说，就是把传统transformer结构中decoder层里面的单个线性层替换层多个并列的线性层。在这些线性层前面还有一个Router，Router会选择并列线性层里面的一部分进行计算。这样的话，既能让模型学习更多的知识（多个“专家”），又能减少推理计算量（选择部分“专家”进行计算）。接下来我们从Router（也叫Gate）模块、MOE推理模块和损失函数模块这3个方面进行解读。

Router模块

Router模块的输入是序列特征，形状是[batch_size, seq_len, hidden_dim]，输出是select_expert_id和expert_weight，shape都是[batch_size, seq_len, topk]，topk是为每个token选择的专家数量。

Router模块实际上是由全连接层、softmax层以及topk算子组成，如果全部候选专家的数量一共是expert_num，那么全连接层的输出shape是[batch_size, seq_len, expert_num]，代表每个token被分配到每个候选专家的概率，然后使用softmax对概率值进行归一化，最后使用topk算子把概率排在前面的专家选择出来，得到的输出shape就是[batch_size, seq_len, topk]。

举个例子，假如batch_size=1, seq_len=5, expert_num=6, topk=3，那么Router模块中的topk最后输出可能是[[0, 1, 2], [2, 4, 5], [1, 2, 3], [0, 3, 5], [3, 4, 5]]和[[0.2, 0.3, 0.3], [0.25, 0.28, 0.32], [...], [...], [...]]。这个输出代表第1个token会给0号专家、1号专家和2号专家计算，然后在推理模块中会把他们的结果分别乘以0.2、0.3、0.3的权重，第2个token会给2号专家、4号专家和5号专家计算，然后在推理模块中会把他们的结果分别乘以0.25、0.28、0.32的权重，以此类推。所以，Router的功能就是把不同的token分给不同的expert，这也是它为什么叫“路由”的原因。

MOE 推理模块

完成路由之后，每个专家就要开始计算了。每个专家需要收集自己负责计算的token，还是以上面给的例子为例，0号专家负责第1个token和第4个token的计算，所以0号专家的输入shape是[2, hidden_dim]；1号专家负责第1个token和第3个token的计算，所以0号专家的输入shape是[2, hidden_dim]，以此类推。

各个专家完成计算后，我们又要把计算进行组合，得到每个token的推理结果。继续上面的例子，假如第1个token在0号专家、1号专家和2号专家的计算结果分别为result_0, result_1, result2，那么整个MOE模块对第1个token的预测结果就是result_0x0.2+result_1x0.3+result2x0.3。

损失函数模块

损失函数包含2部分：专家利用率均衡和样本分配均衡。

专家利用率均衡的计算公式是var(prob_list)，也就是所有专家被选择的概率之和的方差。如果每个专家被选择的概率相近，那么说明分配越均衡，整个系统的算力利用率就越高，否则会造成某些计算节点的闲置浪费。

然后是样本分配均衡，计算公式是sum(token_num_list*prob_list)，也就是把各专家分配到的token数量列表和概率之和列表相乘求和。样本分配越均衡，这个损失函数越小。举个例子，10个专家，10个样本，如果所有样本都分到1个专家，那么损失函数值为10x1+0+0...+0=10，如果平均分给10个专家，那么损失函数值为1x0.1+1x0.1+...+1x0.1=1。