## LLM-para

### 一 、 Moe

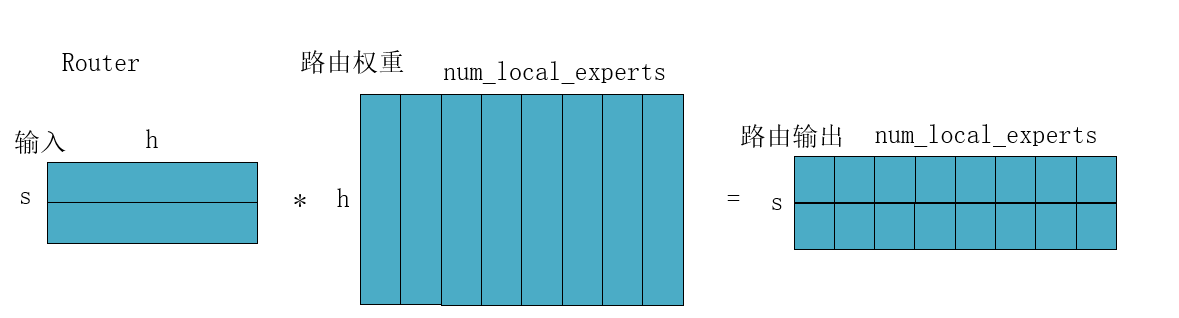
#### 1. 启用MoE

use\_moe = False  
# 如果传入 moe 参数 则启用moe计算  
if num\_experts\_per\_tok is not None:  
 use\_moe = True

#### 2. Prefill MoE 参数计算

| 分解 | Input1 | Input2 | Output | 运算量 |
| --- | --- | --- | --- | --- |
| MoE\_Router | (b,s,h) | (h,num\_local\_experts) | (b,s,num\_local\_experts) | 2bsh\*num\_local\_experts |
| MoE\_FFN1 | (b, s\*num\_experts\_per\_tok , h) | (num\_local\_experts,h, intermediate\_size\*2) | (b,s\*num\_experts\_per\_tok ,intermediate\_size \* 2) | 2bs\* num\_experts\_per\_tok \* h \* intermediate\_size \* 2 + bs \* num\_experts\_per\_tok \* intermediate\_size |
| MoE\_FFN2 | (b,s\*num\_experts\_per\_tok, intermediate\_size) | (num\_local\_experts, intermediate\_size, h) | (b, s\*num\_experts\_per\_tok , h) | 2bs \* num\_experts\_per\_tok \* intermediate\_size \* h |

##### 路由参数计算



moe\_router\_flops = b \* seq \* h \* num\_local\_experts \* 2  
param\_count = h \* num\_local\_experts  
add\_row(phase, "Router", "(b,s,h)", f"(h,{num\_local\_experts})", f"(b,s,{num\_local\_experts})",   
 moe\_router\_flops, param\_count,   
 (b,seq,h), (h,num\_local\_experts), (b,seq,num\_local\_experts), a\_bit, w\_ffn)

##### FNN-1计算

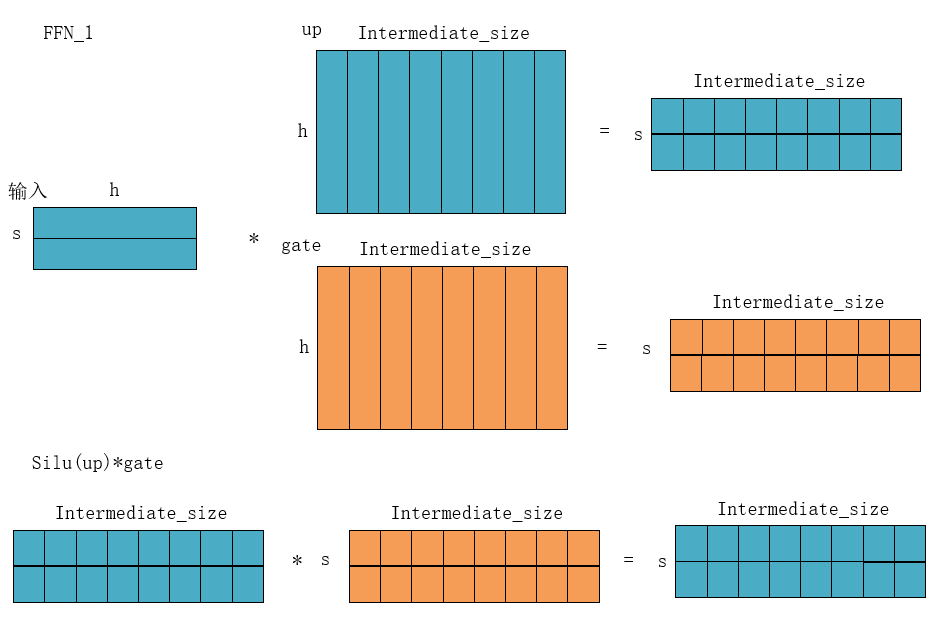
**假设prefill输入seq足够大，大到能够使用到所有expert。**

输入形状为**(b,s\*num\_experts\_per\_tok,h)**: 对于每个token，需要送入num\_experts\_per\_tok个MLP进行计算，则相当于将每个token复制了num\_experts\_per\_tok次，即有seq\*num\_experts\_per\_tok个token。

输入权重形状为**(num\_local\_experts,h,intermediate\_size\*2)**: 使用到所有expert，且加载up和gate的权重。

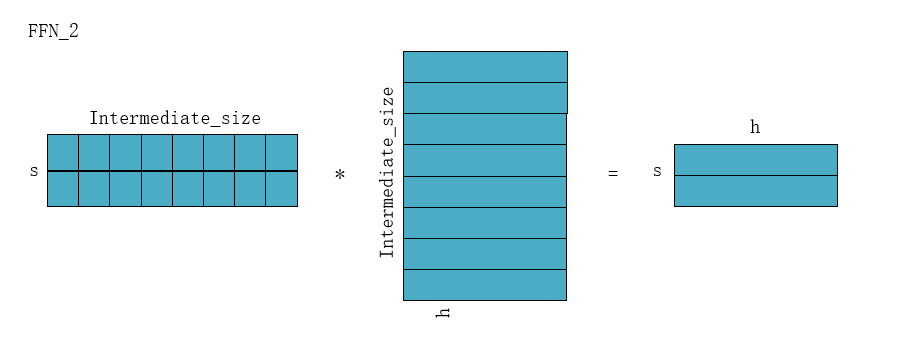
所需参数量：num\_experts\_per\_tok个 up+gate 权重矩阵

# FFN-1(up + gate)(with Moe)  
FFN\_1\_moe = b \* seq \* num\_experts\_per\_tok \* h \* intermediate\_size \* 2 \* 2 # \* 两个矩阵乘法 \* 每个token算num\_experts\_per\_tok次  
# 额外的逐元素乘法  
FFN\_1\_moe += b \* seq \* num\_experts\_per\_tok \* intermediate\_size # (W₁x) ⊙ SiLU(W\_gate\*x)  
  
param\_count = h \* intermediate\_size \* 2 \* num\_local\_experts # 两个权重矩阵 \* 假设 prefill 用到了所有expert  
add\_row(phase, "FFN-1(with Moe)", f"(b,{s}\*num\_experts\_per\_tok,h)", f"(num\_local\_experts,h,{intermediate\_size}\*2)", f"(b,{s}\*num\_experts\_per\_tok,{intermediate\_size}\*2)",   
 FFN\_1\_moe, param\_count,   
 (b, seq \* num\_experts\_per\_tok, h), (h, intermediate\_size \* 2 \* num\_local\_experts), (b, seq \* num\_experts\_per\_tok, intermediate\_size\*2 ), a\_bit, w\_ffn)



##### FFN-2 计算

# FFN-2(with Moe)  
FFN\_2\_moe = b \* seq \* intermediate\_size \* h \* 2 \* num\_experts\_per\_tok  
param\_count = intermediate\_size \* h \* num\_local\_experts  
add\_row(phase, "FFN-2(with Moe)", f"(b, {s}\*num\_experts\_per\_tok, {intermediate\_size})", f"(num\_local\_experts, {intermediate\_size}, h)", f"(b,{s}\*num\_experts\_per\_tok,h)",FFN\_2\_moe, param\_count,  
(b, seq \* num\_experts\_per\_tok, intermediate\_size), (intermediate\_size, h, num\_local\_experts), (b, seq \* num\_experts\_per\_tok, h), a\_bit, w\_ffn)



#### 3. Decode MoE参数计算

prefill 阶段的计算每次需要加载**num\_local\_experts**个MLP的参数。

decode 阶段的计算每次**只**需要加载**num\_experts\_per\_tok**个MLP的参数。