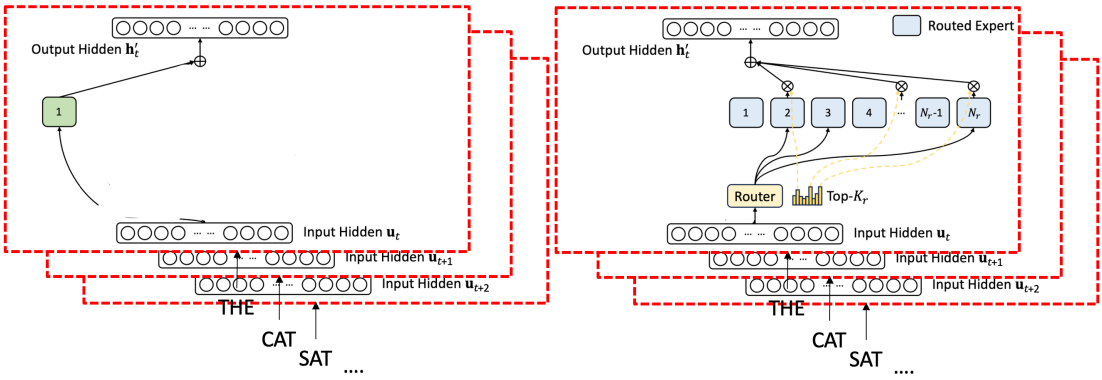
MoE: Mixture of Experts

对于传统的 Embedding 前向使用 FFN：

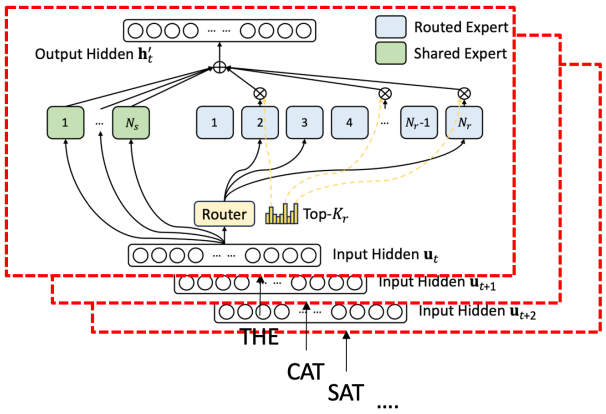
$$h_t = u_t + FFN(u_t)$$



如果 Expert 是 FFN，那么我们可以认为是一个概率加权

$$h'_t = u_t + \sum_{i=1}^{N_r} g_{i,t} FFN_i^{(r)}(u_t) \qquad g_{i,t} = \frac{g'_{i,t}}{\sum_{j=1}^{N_r} g'_{j,t}} \qquad s_{i,t} = \sigma(u_t^\top e_i)$$

$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in TopK(K_r, \{s_{j,t} \mid j \in [1, N_r]\}) \\ 0, & otherwise \end{cases}$$



Router Collapse: 向专家模型分配令牌(tokens)不均衡的情况。缓解方法包括：

1. Token Dropping - 如果专家 e 超载，则不计算$FFN_e(h_t)$
2. Auxiliary Loss: 通过损失函数惩罚不平衡的专家利用率
3. Bias Term: 手动向亲和度分数(affinity scores)添加/减去偏置项以平衡专家

$$g'_{i,t} = \begin{cases} s_{i,t} + b_i, & s_{i,t} \in TopK(K_r, \{s_{j,t} \mid j \in [1, N_r]\}) \\ 0, & otherwise \end{cases}$$

LoRA

$$y = XW = X(W_0 + \Delta W)$$

$$
\begin{aligned}
&= XW_0 + X\Delta W \\
&\approx XW_0 + XL_1L_2 \\
W_0 &: D \times D. \, L_1 : D \times r. \, L_2 : r \times D
\end{aligned}
$$

Quantisation

Absolute Maximum Quantization

$$
\begin{aligned}
X^{Int8} &= round\left(\frac{127}{absmax(X^{FP32})} X^{FP32}\right) = round(c \cdot X^{FP32}) \\
X^{FP32} &= dequantise(c, X^{Int8}) = \frac{X^{int8}}{c^{FP32}}
\end{aligned}
$$

对于特别大的数据数据会立群，导致小数据偏移更大，因此将数据分块，每一块使用单独的 c。

4bit NF Q 是把 FP32 压缩到了 NF4

我们假设每个块有 64 个数，每数 4bit。而每一个块需要一个 FP32 的c，因此 $\frac{32}{64\times4} = 12.5\%$。这占比有点高，我们把 256 个量化常数作为一组，进行一个 8bit 量化，也就是会多一个 32bit。

因此我们可以看作第一部分的量化常数从原来的 FP32 变成 NP8。

$$
\frac{32}{256 \times 64 \times 4} + \frac{8}{64 \times 4} = 3.17\%
$$

也就是先把所有的数据压缩到 64 个 NF4 组成的 Chunk, 每个 chunk 有一个 FP8 的常数($NF4 \to FP32$)。每 256 个 chunk 有一个 FP 32 的常数（$FP8 \to FP32$）。

QLoRA

QLoRA = 4bit Normal Float Quantisation + Double Quantisation + Page Optimisation

$$
y = X \; dequantise(W_0) + XL_1L_2
$$