attention的计算

给定 q, k_i, v_i $\{i=1 \cdots N\}$

op1 = BMM

$$S_i = \text{dot}(q, k_i) \quad \{i=1 \cdots N\}$$

$$m(x) = \max_i S_i$$

$$f(x) = [\exp(S_1 - m(x)) \cdots \exp(S_N - m(x))]$$

为了数值计算的稳定性，要把未归一化的分值全部计算完毕后，求max value，并减去，这会需要 N × M 的中间结果

$$l(x) = \sum_i f(x)_i$$

改善思路 ① 归一化因子 $l(x)$ 可以移动到整个attention计算的最后.

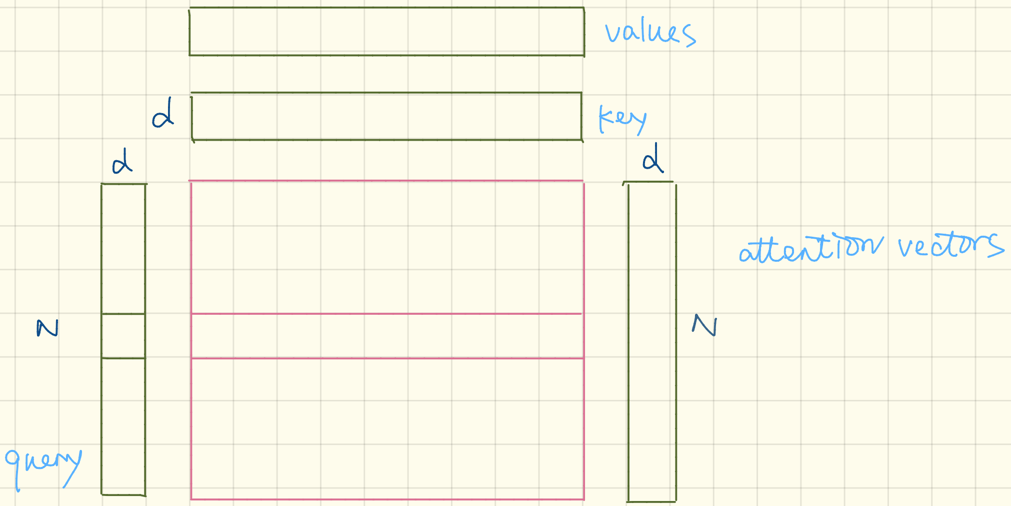
op2 = softmax

$$\text{softmax} = \frac{f(x)}{l(x)}$$

② 每次计算一小块，只考虑当前块的数值稳定性和归一化，当计算新一块时重新归一化

$$\text{attention}(q, k, v) = \frac{\sum_i f(x)_i v_i}{l(x)}$$

op3 = BMM



我们来考虑一种最简单的情况，把 q. k. v 分成两块来计算 attention

$x = [a \ . \ b]_{2N}$ a 和 b 是一个长为 N 的向量.

$q = [q_a \ \ q_b]$ $k = [k_a \ \ k_b]$ $v = [v_a \ v_b]$

$a_i = dot(q_a, k_{ai})$ 计算第一块

$m(a) = \max_i a_i$

$f(a) = [\exp(a_1 - m(a)) \ \cdots \ \exp(a_N - m(a))]$

$l(x) = \sum_i f(a)_i$

$o_a = attention(q_a, k_a, v_a) = \dfrac{\sum_i f(a)_i v_i}{\sum_i f(a)_i}$

$b_i = dot(q_b \ k_{bi})$ 计算第二块

$m(b) = \max_i b_i$

$f(b) = [\exp(b_1 - m(b)) \ \cdots \ \exp(b_N - m(b))]$ 基于当前块进行查询模块的 exp 计算

$l(b) = \sum_{i=1}^{N} f(b)_i$ 相当于对当前块 local 块计算 soft max

这部分和原 soft max 公式没有任何变化

$m(x) = \max(m(a), m(b))$ ← 需要对 a 进行重新归一化

$\begin{bmatrix} e^{m(a)-m(x)} \\ e^{m(b)-m(x)} \end{bmatrix}$ 是两块的重新归一化因子. 很容易验证.

$e^{m(a)-m(x)} \cdot e^{a_i - m(a)} = e^{m(a)-m(x)+a_i-m(a)} = e^{a_i - m(x)}$

相对每一个新块都累加式地重新归一化

于是有如果考虑到更新归一化. 第二块的计算变为:

$$f(x) = \left[e^{m(a)-m(x)} f(a) \quad e^{m(b)-m(x)} f(b) \right]$$

$$l(x) = e^{m(a)-m(x)} \sum_i^N f(a_i) + e^{m(b)-m(x)} \sum_i^N f(b_i)$$

$$= e^{m(a)-m(x)} l(a) + e^{m(b)-m(x)} l(b)$$

$$\text{attention}(q,k,v) = \frac{e^{m(a)-m(x)} f(a) V_a + e^{m(b)-m(x)} f(b) V_b}{l(x)}$$

$$= \frac{O_a \cdot l(a) \cdot e^{m(a)-m(x)} + e^{m(b)-m(x)} f(b) V_b}{l(x)}$$

从第二块的计算过程我们可以得到以下观测结果:

存储上一块的 max value: $m(a)$ 和归一化因子 $l(a)$. 在新一块计算时持续得更新 $m(x)$ 和 $l(x)$. 就能够以每次只计算一小块的方式计算 softmax.

相当于可以将 softmax 与后面的矩阵乘法进一步进行融合.

上面只考虑了两块的情况. NLV 的论文中同样形式证明了分 N 块时, 同样成立. 只用维护 $m(x)$, $l(x)$ 持续累加结果到输出 O.