Rethinking Attention:

$$Att(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

Q K V have shape $(L \times d)$ (L sequence length)

(d dimension of queries/keys/values)

General time complexity for matrix $\underset{(n\times m)}{A} \times \underset{(m\times p)}{B} = \underset{(n\times p)}{C}$

complexity $O(nmp)$

If we ignore softmax and say $Att' = (QK^T)V$ we get different complexities depending on the matrix multiplication order.

$Att' = (QK^T)V$

$(L\times d)\ (d\times L)\ (L\times d)$

$(L\times L) \times (L\times d)$

output shape $(L\times d)$

complexity $(LLd) = O(L^2 d)$

$Att' = Q(K^T V)$

$(L\times d)\ (d\times L)\ (L\times d)$

$(L\times d)\ (d\times d)$

output shape $(L\times d)$

complexity $O(Ld^2)$

the latter $O(d^2 L)$ is preferred as we can choose d l

assume $d < L$. $O(L^2 d)$ is quadratic complexity wrt.

sequence length. But this is only possible by removing the softmax.

Can we find an approximation such that $Q' K'^T \approx sm\left(\frac{Q K^T}{\sqrt{d}}\right)$

Softmax is a function given vector z, of length n, normalizes the elements

as: $\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$

Kernels: Kernels are functions that are equivalent to the dot

product of some feature map ϕ

$\gamma(x, y) = \phi(x)^T \phi(y)$ we would like a function γ that

avoids computing in ϕ's high dimensional space.

If we assume A is a kernel matrix with $A_{ij} = \gamma(q_i, k_j) = e^{(q_i k_j^T)}$

q_i & K_j are raw vectors in Q & K

$$A_{ij} = \gamma(q_i, k_j) = e^{q_i k_j^T} = \phi(q_i)^T \phi(k_j)$$

Most kernels can be approximated via:

$$\phi(x) = \frac{h(x)}{\sqrt{m}} \left(f_1(w_1^T x), \dots f_1(w_m^T), \dots f_l(w_1^T x), \dots f_l(w_m^T x) \right)$$

$h(x)$ & $f_1 \dots f_l$ $(w_j^T x)$ are deterministic functions,

$w_1 \dots w_m$ are random values taken from distribution D, $\phi(x)$

is a vector with shape $(L \times m)$

Gaussian Kernel if $h=1$ $l=2$, $f_1 = \sin x$, $f_2 = \cos x$, $D = N(0, I_d)$

so if we draw w from a normal distribution with mean 0 & unit

variance, we obtain the gaussian kernel by using the feature map.

$$\phi_{gauss}(x) = \frac{1}{\sqrt{m}} \left(\sin(w_1^T x), \dots, \sin(w_m^T x) \dots \cos(w_1^T x) \dots \cos(w_m^T x) \right)$$

$$\gamma_{guss} = e^{\left(-\frac{\|x-y\|^2}{2}\right)} *$$

No we want $\gamma_{sm}(x,y) = e^{(x^T y)}$

$$* \quad \gamma_{guss} = e^{\frac{-\|x-y\|\|x^T-y\|}{2}} = e^{\frac{-|x|^2 - |x||y| - |y||x| + |y|^2}{2}}$$

for product to work $|x||y| = x^T y$

$$\gamma_{guss} = e^{\frac{(2x^T y - |x|^2 - |y|^2)}{2}}$$

$$\gamma_{sm}(x,y) = e^{x^T y} = \gamma_{guss}\, e^{\frac{\|x\|^2}{2}}\, e^{\frac{\|y\|^2}{2}}$$

we can reuse the feature maps that leads to the gaussian kernel changing the h function from $h(x) = 1$ to $h(x) = e^{\frac{\|x\|^2}{2}}$

A few issues with this, SM always produces +ve values, the current approximation will give some -ve values, since we are drawing w from a gaussian with mean 0, this will lead to some abnormal behaviors

Attention as fast weight linear dl.

Input $x^{(i)} \quad x^{(i)} \in \mathbb{R}^{dm}$

$$k^{(i)} = W_K x^{(i)} \qquad q^{(i)} = W_q x^{(i)} \qquad v^{(i)} = W_v x^{(i)}$$

$$\mathbb{K}^{(i)} = [\mathbb{K}^{(i-1)}, k^{(i)}]$$

$$\mathbb{V}^{(i)} = [\mathbb{V}^{(i-1)}, v^{(i)}]$$

$Q K^T V$

but this breaks (commutativity ⇒ ( )

$$y^{(i)} = \mathbb{V}^{(i)} \text{softmax}\left((\mathbb{K}^{(i)})^T q^{(i)}\right)$$

$a \times b \neq b \times a$

If we ignore the normalisation & softmax ($a + (b + c) = (a + b) + c$

$$y^{(i)} = \mathbb{V}^{(i)}\left((\mathbb{K}^{(i)})^T \cdot q^{(i)}\right) = \left(\mathbb{V}^{(i)}(\mathbb{K}^{(i)})^T\right) \cdot q^{(i)}$$

$$= \left(\sum_{j=1}^{i} v^{(j)} \otimes k^{(j)}\right) \cdot q^{(i)}$$

Transpose/conjugate $\dagger$

BRA-KET $|\psi\rangle\langle\phi| = |\psi\rangle^\dagger$ Inner product

Notation $|\phi\rangle\langle\phi| = |\phi\rangle^\dagger$ $\phi^T \cdot \psi = \langle\phi|\psi\rangle$

Outer product

$\phi^\dagger \otimes \psi = |\phi \times \psi\rangle$

$$= \left(\sum_{j=1}^{i} |v^{(j)} \times k^{(j)}|\right)|q^{(i)}\rangle = W^{(i)}|q^{(i)}\rangle$$

$$W^{(i)} = W^{(i-1)} + |V^{(i)} \times K^{(i)}|$$

$$y^{(i)} = W^{(i)} |q^{(i)}\rangle$$

key / value db entries

Linear additive database: $W^{(i)} = |V^{(1)} \times K^{(1)}| + |V^{(2)} \times K^{(2)}| + \dots$

$$= \sum_{j}^{n} |V^{(j)} \otimes K^{(j)}| \quad (n \text{ entries in db})$$

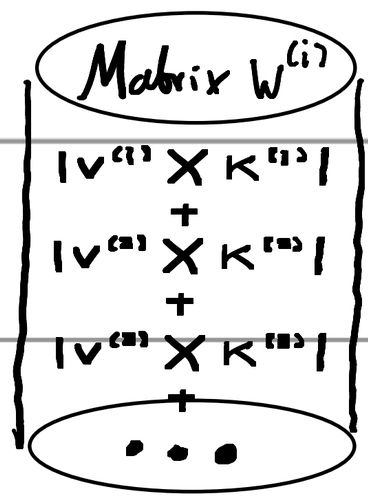
Query $|q^{(i)}\rangle$ on database $W^{(i)}$ $W^{(i)} |q^{(i)}\rangle = \left(\sum_{j}^{1} |V^{(j)} \otimes K^{(j)}| \right) |q^{(i)}\rangle$

more key / values

lets say $|q^{(i)}\rangle = |K^{(2)}\rangle$ $W^{(i)} |K^{(2)}\rangle = \left(|V^{(1)} \times K^{(1)}| + |V^{(2)} \times K^{(2)}| + \dots \right) |K^{(2)}\rangle$

$$= |V^{(1)} \times \underbrace{K^{(1)} | K^{(2)}\rangle}_{0} + |V^{(2)} \times \underbrace{K^{(2)} | K^{(2)}\rangle}_{1} + \underbrace{\dots}_{\text{all zeros}}$$

$$= |V^{(2)}\rangle$$



$|q^{(i)}\rangle = |K^{(2)}\rangle$ INPUT QUERY → OUTPUT VALUE → $W^{(i)} |q^{(i)}\rangle = |V^{(2)}\rangle$

The query $|q^{(i)}\rangle$ selects $|V^{(2)}\rangle$ from $W^{(i)}$ given $|K^{(2)}\rangle$, $|K^{(n)}\rangle$ needs to be orthogonal or you would get a mixture of multiple values.