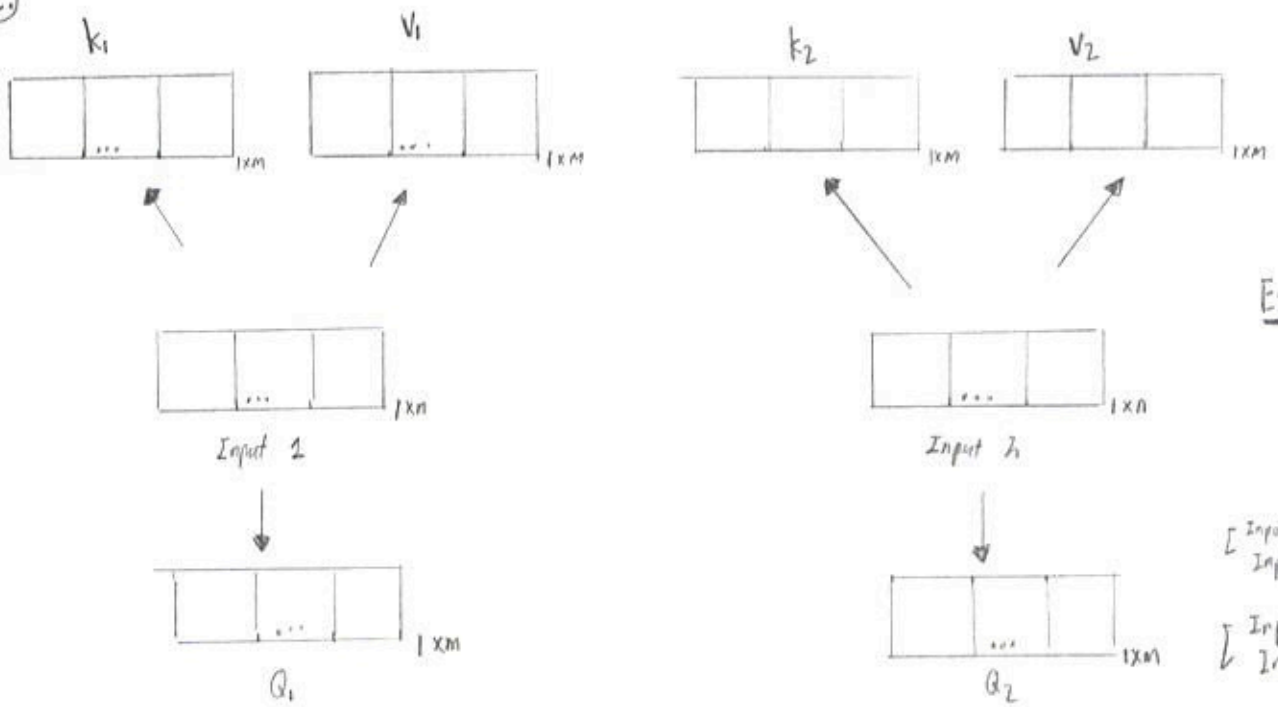
Self-Attention encodes a sequence into a form that accounts for the other respective data in the sequence

* each element in new form is aware of surrounding data

(1.)



$W = [K_{n \times m}, V_{n \times m}, Q_{n \times m}]$

- n ⇒ input dim
- m ⇒ output dim

Ex: $\begin{bmatrix} \text{Input 1} \\ \text{Input 2} \end{bmatrix}_{2 \times n} K_{n \times m} = \begin{bmatrix} k_{1,(1 \times m)} \\ k_{2,(1 \times m)} \end{bmatrix}_{2 \times m}$

$\begin{bmatrix} \text{Input 2} \\ \text{Input 2} \end{bmatrix}_{2 \times n} V_{n \times m} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_{2 \times m}$

$\begin{bmatrix} \text{Input 2} \\ \text{Input 2} \end{bmatrix}_{2 \times n} Q_{n \times m} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}_{2 \times m}$

(2.) Repeat for each input

For Input 1:

(A.) $S_{1,(1 \times 2)} = Q_1 k_1^T$

$S_{2,(1 \times 2)} = Q_1 k_2^T$

$\hat{S}_1 = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$

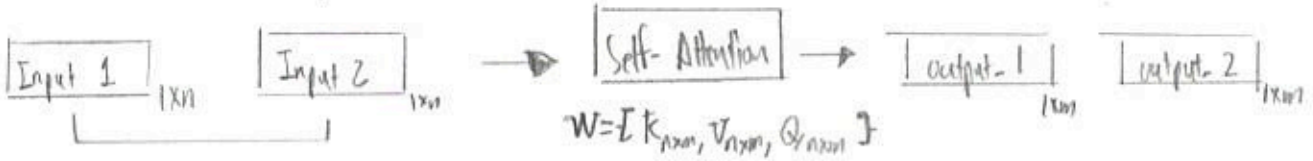
(B.) $\text{softmax}(\hat{S}_1) \rightarrow \hat{\hat{S}}_1$

(C.) $\hat{V}_1 = V_1 \odot \hat{\hat{S}}_1$

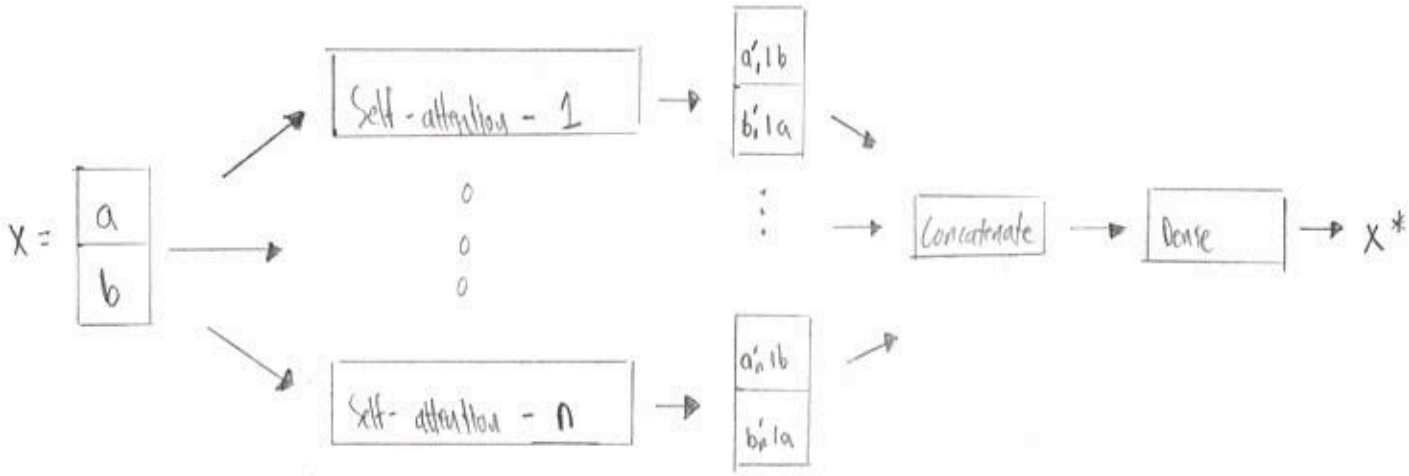
$\hat{V}_2 = V_2 \odot \hat{\hat{S}}_2$

(D.) $\text{Output\_1}_{(1 \times m)} = \sum_{i=1}^{2} \hat{V}_i$

(3.)

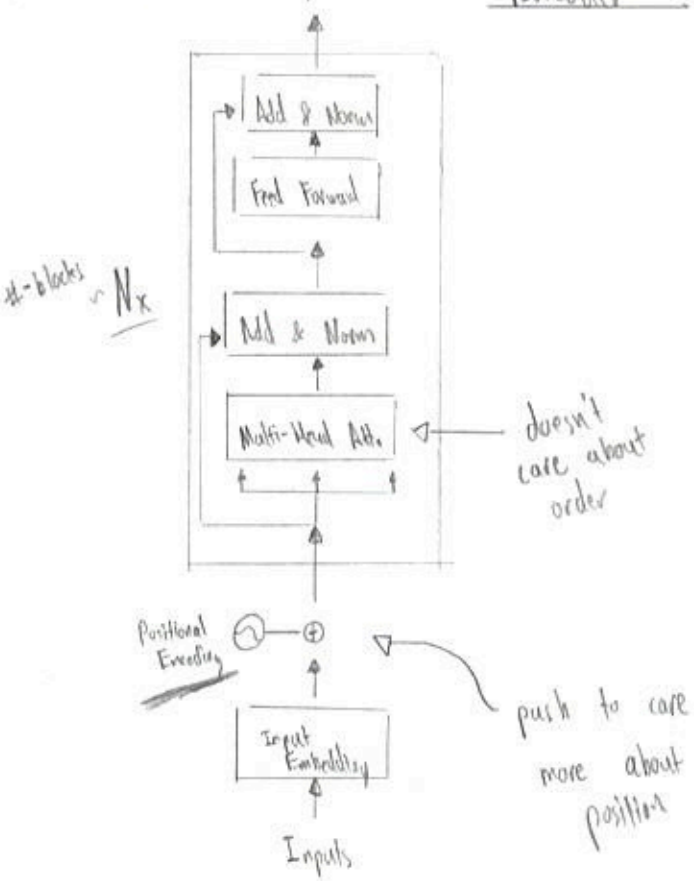


Multi-head attention



Transformers

Encoder



Recurrent Network vs MH-attention

- RN can't be parallized, MH-att can
- RN not numerically stable