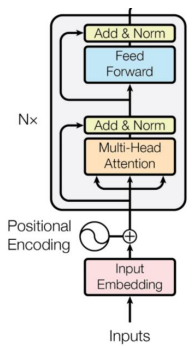
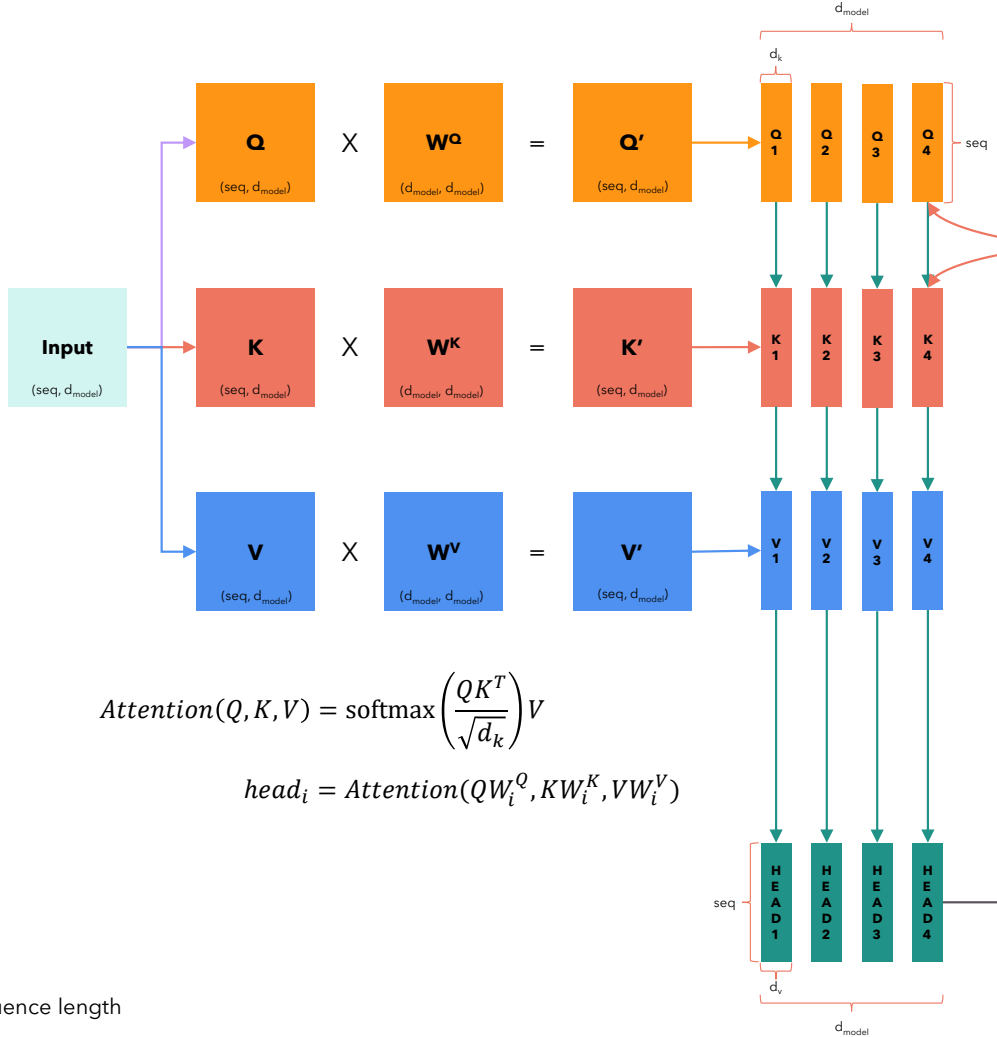
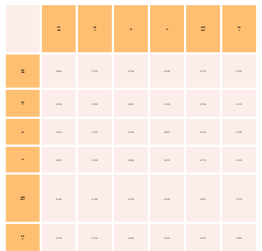
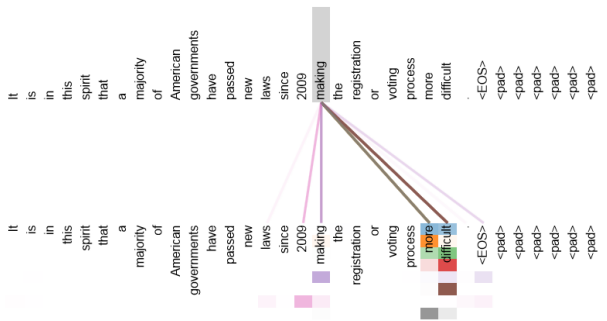




$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) =$$



Attention Visualizations



$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

$$MultiHead(Q,K,V) = Concat(head_1 \ \ldots head_h)W^O$$

seq = sequence length

d_{model} = size of the embedding vector

h = number of heads

$d_k = d_v$ = d_{model} / h

Umar Jamil – https://github.com/hkproj/transformer-from-scratch-notes

Multi-Head Attention

Given a sequence of embedding vectors $X \in R^{N \times D}$, we obtain 3 different projections of it as:

$$Q = X W_Q \in R^{N \times D}$$

$$K = X W_K \in R^{N \times D}$$

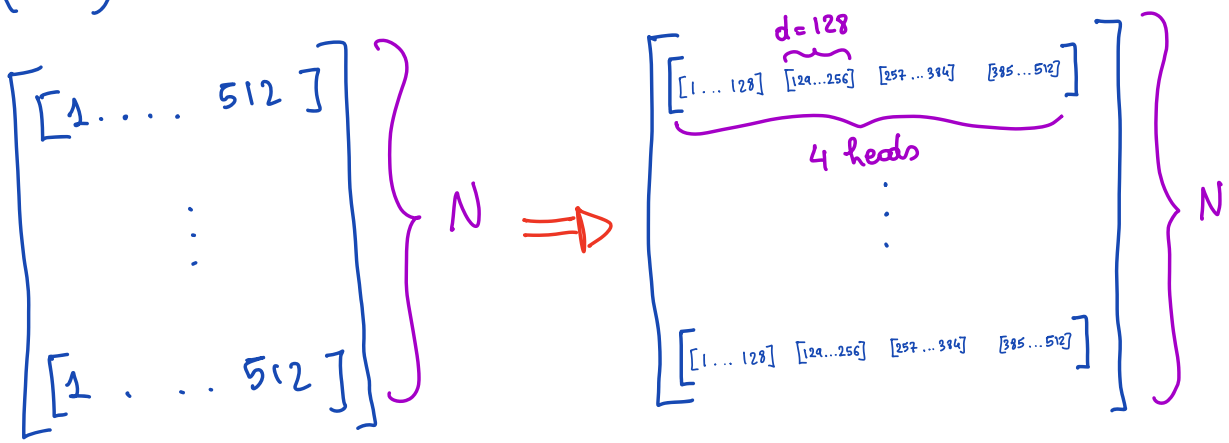
$$V = X W_V \in R^{N \times D}$$

where W_Q, W_K and W_V are learnable parameter matrices.

Note: in case of cross-attention, we use another sequence $Y \in R^{N \times D}$ to compute K and V

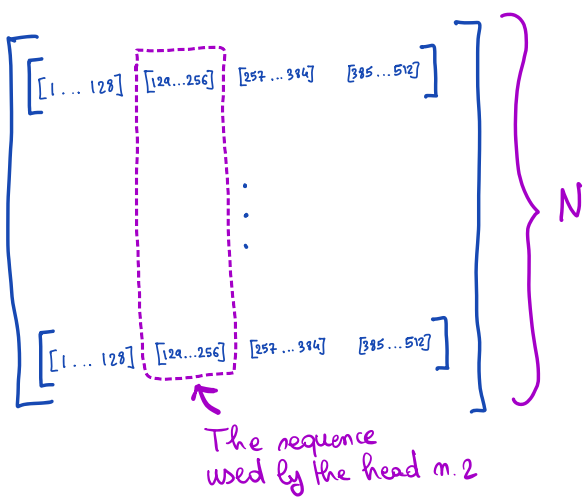
Why is it called Multi-Head?

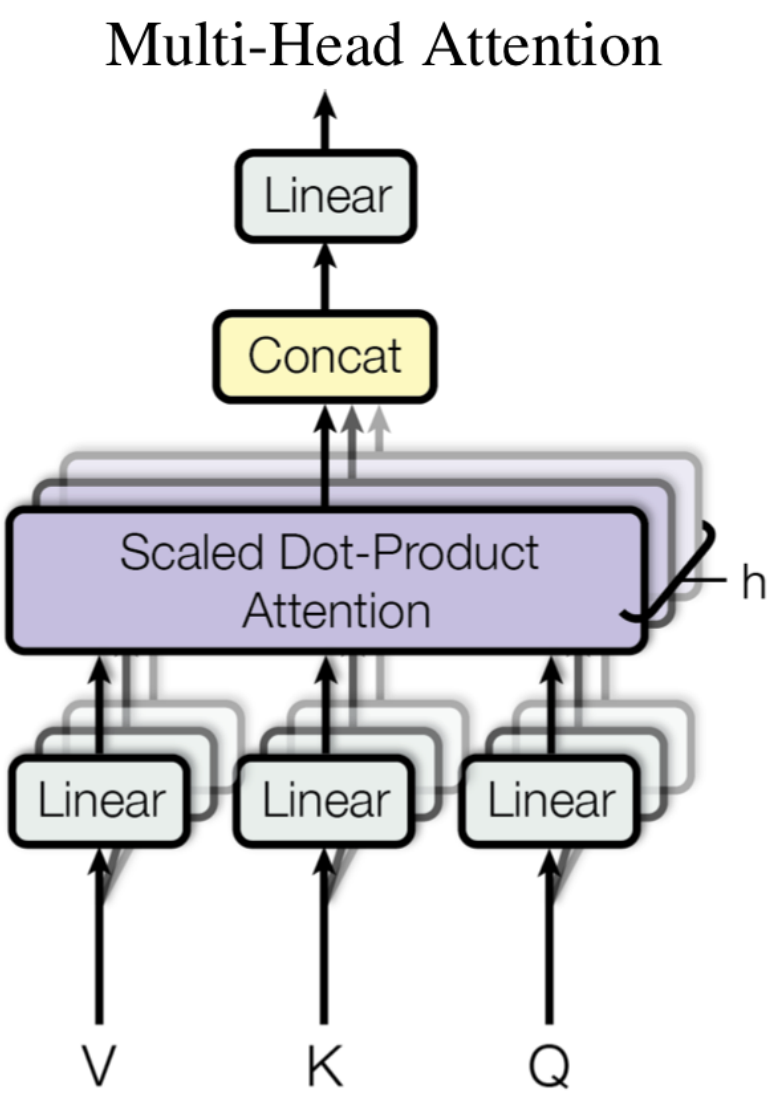
we are given a sequence of embedding vectors (N, D) and we split it into multiple heads



$(N, D) \longrightarrow (N, 4, d)$ where $d = \frac{D}{4}$

Note: D is commonly called d_{model}
while d is commonly called d_{head}




Multi-Head Attention
Linear
Concat
Scaled Dot-Product
Attention
h
Linear
Linear
Linear
V
K
Q

Q : input sequence with shape (N, d)

K : input sequence with shape (N, d)

V : input sequence with shape (N, d)

N = sequence length

d = head dimension

We compute the following:

$$S = QK^T \in R^{N \times N}$$

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$P = \text{softmax}(S) \in R^{N \times N}$$

$$O = PV \in R^{N \times d}$$

Note: we usually scale S by $\frac{1}{\sqrt{d}}$ but it can be absorbed by Q because $aQK^T = (aQ)K^T$ where a is a scalar.