2. Transformer - Self-Attention

a list of N embedding vectors, N is a hyperparameter ①

Multiply	X_1 a word's embedding 1×512	X_2 1×512	
W^Q 512×64	q_1 1×64	q_2 1×64	Query
W^K 512×64	K_1 1×64	k_2 1×64	Key
W^V 512×64	V_1 1×64	V_2 1×64	Value

$X_1 \times W^Q = q_1$

Step 1: Calculate q, k, v vectors (1×64) by multiplying Input vector with the weight vectors W^Q, W^K, W^V (512×64) (1×512)

Step 2: for each word (input), Score it by calculating the dot product of its q and each k

② score vector:

$s_1 = [\, q_1 \cdot k_1 \quad q_1 \cdot k_2 \,]$

$s_2 = [\, q_2 \cdot k_1 \quad q_2 \cdot k_2 \,]$

score $= [s_1 \; s_2]$

Step 3: divide each score by the square root of the dimension of the key vector (original paper: $\sqrt{64} = 8$) ⇒ for stabler gradients

③

$$s_1 = \left[\frac{q_1 \cdot k_1}{\sqrt{d_k}} \quad \frac{q_1 \cdot k_2}{\sqrt{d_k}}\right]$$

$$s_2 = \left[\frac{q_2 \cdot k_1}{\sqrt{d_k}} \quad \frac{q_2 \cdot k_2}{\sqrt{d_k}}\right]$$

Step 4: Softmax the score vector normalize the scores and make them add up to 1

④

$s_1 = \text{Softmax}(s_1)$

$s_2 = \text{Softmax}(s_2)$

scores $= [s_1 \; s_2]$

Step 5: hadamard product (element-wise multiplication) of the value vectors and the Softmax-ed score vector to get the weighted values of each word for the word in the current position

⑤

weighted_V_1 $= \text{softmax}(s_1) \odot [V_1 \; V_2]$

weighed_V_2 $= \text{Softmax}(s_2) \odot [V_1 \; V_2]$

Step 6: Sum the weighted values - that's the output of the Self-Attention layer

⑥

$z_1 = \text{Sum}(\text{weighted\_}V_1)$

$z_2 = \text{Sum}(\text{weighted\_}V_2)$

Self-attention output vector : $[z_1 \; z_2]$

→ This is to be fed into the Feed-forward Neural Network

Matrix Calculation for faster processing

① $X \times W^Q = Q$ $(N\times512) \times (512\times64) = (N\times64)$

K V

N: num of words in the longest sentence a hyperparameter

②~⑥

$$Z = \text{Softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) \times V$$

↑ ⊙ then Sum