Attention and Transformers: Background

John Chiasson[1] *and Ruthvik Vaila*[2]
Boise State University[1] *and LambWeston Holdings Inc.*[2]

Figure 1 shows the architecture of the Attention and Transformer Network.

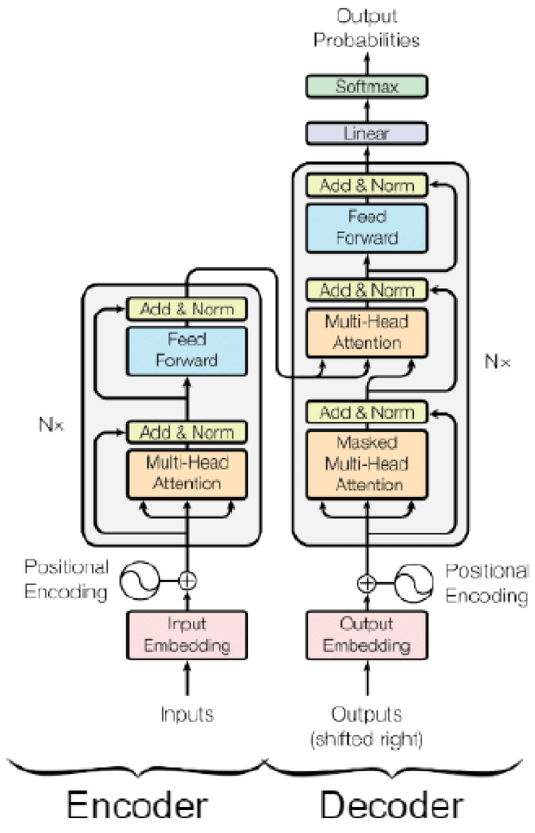


Figure 1: From *Attention Is All You Need* at *https://arxiv.org/abs/1706.03762*

Suppose we want to translate French into English. We limit each sentence to no more than n words. Each word in the English dictionary is embedded into a vector in

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_{512} \end{bmatrix} \in \mathbb{R}^{1\times 512}.$$

Note that x is a *row* vector. A sentence is then represented by

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times 512}$$

where each row represents a word. Suppose the input sentence is only 3 words long. Then $x^{(4)}$ is an end of sentence token and $x^{(5)}, x^{(6)}, ..., x^{(n)}$ are empty tokens, That is,

$$\begin{aligned} x^{(4)} &= \begin{bmatrix} \mathrm{eos}_1 & \mathrm{eos}_2 & \cdots & \mathrm{eos}_{512} \end{bmatrix} \in \mathbb{R}^{1\times 512} \\ x^{(5)} &= \cdots = x^{(n)} = \begin{bmatrix} \mathrm{empty}_1 & \mathrm{empty}_2 & \cdots & \mathrm{empty}_{512} \end{bmatrix} \in \mathbb{R}^{1\times 512}. \end{aligned}$$

Single Head Attention

To start our explanation of attention we fist define the *query, key,* and *value* associated with an input word.

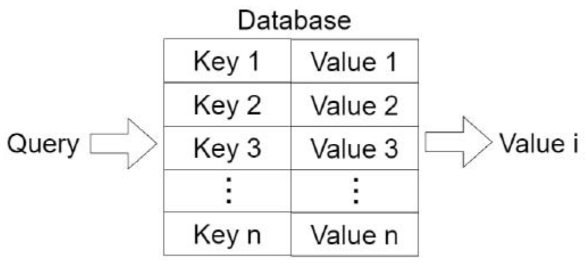


Figure 5. Query-retrieval in a database.

Figure 2: Database Query.

Let

$$W_Q \in \mathbb{R}^{512\times 64}, W_K \in \mathbb{R}^{512\times 64}, W_V \in \mathbb{R}^{512\times 64}$$

and for each word $x^{(i)}$ define the i^{th} query, key, and value vectors by

$$\begin{aligned} q^{(i)} &\triangleq x^{(i)} W_Q \in \mathbb{R}^{1\times 64} \\ k^{(i)} &\triangleq x^{(i)} W_K \in \mathbb{R}^{1\times 64} \\ v^{(i)} &\triangleq x^{(i)} W_V \in \mathbb{R}^{1\times 64}. \end{aligned}$$

Then set

$$\begin{aligned} Q &\triangleq \begin{bmatrix} q^{(1)} \\ q^{(2)} \\ \vdots \\ q^{(n)} \end{bmatrix} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W_Q = XW_Q \in \mathbb{R}^{n\times 64} \\ K &\triangleq \begin{bmatrix} k^{(1)} \\ k^{(2)} \\ \vdots \\ k^{(n)} \end{bmatrix} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W_K = XW_K \in \mathbb{R}^{n\times 64} \\ V &\triangleq \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \vdots \\ v^{(n)} \end{bmatrix} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W_V = XW_V \in \mathbb{R}^{n\times 64} \end{aligned}$$

The matrices $W_Q \in \mathbb{R}^{512\times 64}, W_K \in \mathbb{R}^{512\times 64}, W_V \in \mathbb{R}^{512\times 64}$ will be learned during training by updating their values in order to minimize the cost.

The interpretation here is that the key vector $k^{(i)} \in \mathbb{R}^{1\times 64}$ corresponds to the value vector $v^{(i)} \in \mathbb{R}^{1\times 64}$. For a query vector $q^{(i)} \in \mathbb{R}^{1\times 64}$ we want to know how closely it is related to each of the value vectors $v^{(j)}, j = 1, ..., n$. To do this we first compute the *similarity* s_{ij} of query $q^{(i)}$ to key $k^{(j)}$ as defined by

$$s_{ij} \triangleq \frac{q^{(i)} k^{T(j)}}{\sqrt{64}} = \frac{\begin{bmatrix} \quad q^{(i)} \quad \end{bmatrix} \begin{bmatrix} k^{T(j)} \\ \\ \end{bmatrix}}{\sqrt{64}}.$$

Then softmax of these similarities s_{ij} with respect to the second index is

$$\text{softmax}(s_{ij}) = \frac{e^{s_{ij}}}{\sum_{j=1}^{n} e^{s_{ij}}}.$$

More generally we compute

$$\begin{aligned}
S = \{s_{ij}\}_{i,j=1,\ldots,n} \triangleq \frac{QK^T}{\sqrt{64}} &= \frac{1}{\sqrt{64}} \begin{bmatrix} q^{(1)} \\ q^{(2)} \\ \vdots \\ q^{(n)} \end{bmatrix} \begin{bmatrix} k^{T(1)} & k^{T(2)} & \cdots & k^{T(n)} \end{bmatrix} \\
&= \frac{1}{\sqrt{64}} \begin{bmatrix} q^{(1)}k^{T(1)} & q^{(1)}k^{T(2)} & \cdots & q^{(1)}k^{T(n)} \\ q^{(2)}k^{T(1)} & q^{(2)}k^{T(2)} & \cdots & q^{(2)}k^{T(n)} \\ \vdots & \vdots & \cdots & \vdots \\ q^{(n)}k^{T(1)} & q^{(n)}k^{T(n)} & \cdots & q^{(n)}k^{T(n)} \end{bmatrix}
\end{aligned}$$

Take the softmax of S by the *row* to obtain

$$\text{softmax}(S) = \text{softmax}\left(\frac{QK^T}{\sqrt{64}}\right) = \begin{bmatrix} \frac{e^{s_{11}}}{\sum_{j=1}^{n} e^{s_{1j}}} & \frac{e^{s_{12}}}{\sum_{j=1}^{n} e^{s_{1j}}} & \cdots & \frac{e^{s_{1n}}}{\sum_{j=1}^{n} e^{s_{1j}}} \\ \frac{e^{s_{21}}}{\sum_{j=1}^{n} e^{s_{2j}}} & \frac{e^{s_{22}}}{\sum_{j=1}^{n} e^{s_{2j}}} & \cdots & \frac{e^{s_{2n}}}{\sum_{j=1}^{n} e^{s_{2j}}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{e^{s_{n1}}}{\sum_{j=1}^{n} e^{s_{nj}}} & \frac{e^{s_{n2}}}{\sum_{j=1}^{n} e^{s_{nj}}} & \cdots & \frac{e^{s_{nn}}}{\sum_{j=1}^{n} e^{s_{nj}}} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}.$$

Note that the elements of softmax(S) are all positive and each row sums to one. That is, $\sum_{j=1}^{n} p_{ij} = 1$ for $i = 1, 2, ..., n$. Finally the *attention* (row) vectors $z^{(i)}$ for the words in the input sentence are

$$\begin{aligned}
Z = \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(n)} \end{bmatrix} = \text{softmax}\left(\frac{QK^T}{\sqrt{64}}\right) V &= \begin{bmatrix} \frac{e^{s_{11}}}{\sum_{j=1}^{n} e^{s_{1j}}} & \frac{e^{s_{12}}}{\sum_{j=1}^{n} e^{s_{1j}}} & \cdots & \frac{e^{s_{1n}}}{\sum_{j=1}^{n} e^{s_{1j}}} \\ \frac{e^{s_{21}}}{\sum_{j=1}^{n} e^{s_{2j}}} & \frac{e^{s_{22}}}{\sum_{j=1}^{n} e^{s_{2j}}} & \cdots & \frac{e^{s_{2n}}}{\sum_{j=1}^{n} e^{s_{2j}}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{e^{s_{n1}}}{\sum_{j=1}^{n} e^{s_{nj}}} & \frac{e^{s_{n2}}}{\sum_{j=1}^{n} e^{s_{nj}}} & \cdots & \frac{e^{s_{nn}}}{\sum_{j=1}^{n} e^{s_{nj}}} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \vdots \\ v^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times 64} \\
&= \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \vdots \\ v^{(n)} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^{n} p_{1j} \begin{bmatrix} v^{(j)} \end{bmatrix} \\ \sum_{j=1}^{n} p_{2j} \begin{bmatrix} v^{(j)} \end{bmatrix} \\ \vdots \\ \sum_{j=1}^{n} p_{nj} \begin{bmatrix} v^{(j)} \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{n\times 64}
\end{aligned}$$

p_{11} is the "probability" of how much $v^{(1)}$ is "connected/related/attentive/similar" to $v^{(1)}$.
p_{12} is the "probability" of how much $v^{(2)}$ is "connected/related/attentive/similar" to $v^{(1)}$.
$\vdots$
p_{1n} is the "probability" of how much $v^{(n)}$ is "connected/related/attentive/similar" to $v^{(1)}$.
Then

$$z^{(1)} = \sum_{j=1}^{n} p_{1j} v^{(j)} \in \mathbb{R}^{1\times 64}$$

is the attention vector for the word $x^{(1)}$ and is a weighted sum of the value vectors of all the other words in the sentence.

Residual Addition and Layer Normalization

The output of the multihead attention block is $Z_X \in \mathbb{R}^{n\times 512}$. The residual addition consists of adding the input $X \in \mathbb{R}^{n\times 512}$ to Z_X to obtain

$$Z_X + X \in \mathbb{R}^{n\times 512}.$$

Let $m = \begin{bmatrix} m_1 & m_2 & \cdots & m_n \end{bmatrix} \in \mathbb{R}^{1\times 512}$ be the mean of each *row* of $Z_X + X$ and $\sigma = \begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{bmatrix} \in \mathbb{R}^{1\times 512}$ be the variance of each *row* of $Z_X + X$. With

$$M_n \triangleq \begin{bmatrix} m_1 & m_1 & \cdots & m_1 \\ m_2 & m_2 & \cdots & m_2 \\ \vdots & \vdots & \cdots & \vdots \\ m_n & m_n & \cdots & m_n \end{bmatrix} \in \mathbb{R}^{n\times 512}, \quad \sigma_n = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_n \end{bmatrix}$$

the normalization of $Z_X + X$ is given by

$$Z_X^{(\text{normal})} \triangleq \sigma_n^{-1}\Big((Z_X + X) \; - M_n\Big) \in \mathbb{R}^{n\times 512}.$$

Finally with

$$\gamma = \begin{bmatrix} \gamma_1 & 0 & 0 & 0 \\ 0 & \gamma_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \gamma_n \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad \beta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_n \\ \beta_1 & \beta_2 & \cdots & \beta_n \\ \beta_1 & \beta_2 & \cdots & \beta_n \\ \beta_1 & \beta_2 & \cdots & \beta_n \end{bmatrix} \in \mathbb{R}^n$$

this normalization is rescaled to obtain the layer normalization given by

$$Z'_X = \text{LayerNorm}(Z_X + X) \triangleq Z_X^{(\text{normal})}\gamma + \beta = \sigma_n^{-1}\Big((Z_X + X) \; - M_n\Big)\gamma + \beta \in \mathbb{R}^{n\times 512}$$

where γ, β are learned parameters (updated by backpropagation).

Feedforward with Residual Addition & Layer Normalization

Let $W_1 \in \mathbb{R}^{512\times 1024}, b_1 \in \mathbb{R}^{1\times 1024}, W_2 \in \mathbb{R}^{1024\times 512}, b_2 \in \mathbb{R}^{1\times 512}$ and

$$B_1 \triangleq \begin{bmatrix} b_1 \\ b_1 \\ \vdots \\ b_1 \end{bmatrix} \in \mathbb{R}^{n\times 1024}, \quad B_2 \triangleq \begin{bmatrix} b_2 \\ b_2 \\ \vdots \\ b_2 \end{bmatrix} \in \mathbb{R}^{n\times 512}.$$

Note the each row of B_1 is the same as is each row of B_2. the quantaties W_1, b_1, W_2, b_2 are all learnable parameters (updated by backpropagation). The feedforward operation transforms Z'_X to

$$Z_X^{\text{ff}} \triangleq \Big(\max(0_{n\times 1024}, Z'_X W_1 \; + B_1)\Big) W_2 + B_2 \in \mathbb{R}^{n\times 512}.$$

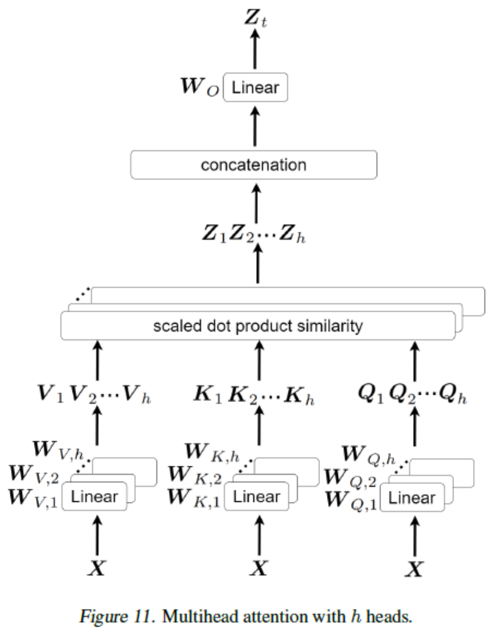
The residual addition followed by layer normalization is done in the same manner as above to get

$$Z_X^{\text{encoder}} \triangleq \text{LayerNorm}(Z_X^{\text{ff}} + Z'_X) \in \mathbb{R}^{n\times 512}.$$

Z_X^{encoder} is fed into key input of the decoder's multihead attention block of the decoder (the one without attention) and also into the value input of the multihead attention block. The query is from the output of the masked multihead attention block which is explained below.

Multihead Attention in the Encoder

Use 8 parallel attention layers or heads. Each head will process $512/8 = 64$ components of each embedded word.



Z_t
W_O Linear
concatenation
$Z_1 Z_2 \cdots Z_h$
scaled dot product similarity
$V_1 V_2 \cdots V_h$
$K_1 K_2 \cdots K_h$
$Q_1 Q_2 \cdots Q_h$
$W_{V,h}$ $W_{V,2}$ $W_{V,1}$ Linear
$W_{K,h}$ $W_{K,2}$ $W_{K,1}$ Linear
$W_{Q,h}$ $W_{Q,2}$ $W_{Q,1}$ Linear
X
X
X
Figure 11. Multihead attention with h heads.

Figure 3: Multihead attention

Let

$$\begin{aligned} W_{Q,\ell} &\in \mathbb{R}^{512\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \\ W_{K,\ell} &\in \mathbb{R}^{512\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \\ W_{V,\ell} &\in \mathbb{R}^{512\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \end{aligned}$$

and

$$\begin{aligned} Q_\ell &\triangleq XW_{Q,\ell} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W_{Q,\ell} \in \mathbb{R}^{n\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \\ K_\ell &\triangleq XW_{K,\ell} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W_K, \ell \in \mathbb{R}^{n\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \\ V_\ell &\triangleq XW_{V,\ell} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} W_{V,\ell} \in \mathbb{R}^{n\times 64} \quad \text{for} \quad \ell = 1, ..., 8. \end{aligned}$$

With

$$\begin{aligned} \text{concat}(Q_1, Q_2, ..., Q_8) &\triangleq \begin{bmatrix} Q_1 & Q_2 & \cdots & Q_8 \end{bmatrix} = X \begin{bmatrix} W_{Q,1} & W_{Q,2} & \cdots & W_{Q,8} \end{bmatrix} \in \mathbb{R}^{n\times 512} \\ \text{concat}(K_1, K_2, ..., K_8) &\triangleq \begin{bmatrix} K_1 & K_2 & \cdots & K_8 \end{bmatrix} = X \begin{bmatrix} W_{K,1} & W_{K,2} & \cdots & W_{K,8} \end{bmatrix} \in \mathbb{R}^{n\times 512} \\ \text{concat}(V_1, V_2, ..., V_8) &\triangleq \begin{bmatrix} V_1 & V_2 & \cdots & V_8 \end{bmatrix} = X \begin{bmatrix} W_{V,1} & W_{V,2} & \cdots & W_{V,8} \end{bmatrix} \in \mathbb{R}^{n\times 512} \end{aligned}$$

take the softmax of each head to obtain

$$\left[\begin{array}{cccc} \text{softmax}\left(\frac{Q_1K_1^T}{\sqrt{64}}\right)V_1 & \text{softmax}\left(\frac{Q_2K_2^T}{\sqrt{64}}\right)V_2 & \cdots & \text{softmax}\left(\frac{Q_8K_8^T}{\sqrt{64}}\right)V_8 \end{array}\right] \in \mathbb{R}^{n\times 512}.$$

With $W_0 \in \mathbb{R}^{n\times n}$ the attention from all the heads is defined as

$$\begin{aligned} Z_X &\triangleq \left[\begin{array}{cccc} Z_1 & Z_2 & \cdots & Z_8 \end{array}\right] \\ &= W_0\left[\begin{array}{cccc} \text{softmax}\left(\frac{Q_1K_1^T}{\sqrt{64}}\right)V_1 & \text{softmax}\left(\frac{Q_2K_2^T}{\sqrt{64}}\right)V_2 & \cdots & \text{softmax}\left(\frac{Q_8K_8^T}{\sqrt{64}}\right)V_8 \end{array}\right] \in \mathbb{R}^{n\times 512} \end{aligned}$$

where W_0 is learned during training.

Masked Attention in the Decoder

Let

$$Y \triangleq \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times 512}$$

be a target sentence where each *row* of Y represents a target word. The first word $y^{(1)}$ of the target sentence is a start sentence token given by

$$y^{(1)} = \left[\begin{array}{cccc} \text{start}_1 & \text{start}_2 & \cdots & \text{start}_{512} \end{array}\right] \in \mathbb{R}^{1\times 512}.$$

and the same for all target sentences. Suppose the target sentence is only 5 words long. Then $y^{(2)}, ..., y^{(6)}$ make up the sentence, $y^{(7)}$ is an end of sentence token and $y^{(8)}, ..., y^{(n)}$ are empty tokens, i.e.,

$$\begin{aligned} y^{(7)} &= \left[\begin{array}{cccc} \text{eos}_1 & \text{eos}_2 & \cdots & \text{eos}_{512} \end{array}\right] \in \mathbb{R}^{1\times 512} \\ y^{(8)}, ..., y^{(n)} &= \left[\begin{array}{cccc} \text{empty}_1 & \text{empty}_2 & \cdots & \text{empty}_{512} \end{array}\right] \in \mathbb{R}^{1\times 512}. \end{aligned}$$

We next define the query, key, and value (row) vectors for the target sentences.

$$\begin{aligned} Q_\ell &\triangleq YW_{Q,\ell} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} W_{Q,\ell} \in \mathbb{R}^{n\times 64}, \quad W_{Q,\ell} \in \mathbb{R}^{512\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \\ K_\ell &\triangleq YW_{K,\ell} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} W_K, \ell \in \mathbb{R}^{n\times 64}, \quad W_{K,\ell} \in \mathbb{R}^{512\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \\ V_\ell &\triangleq YW_{V,\ell} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix} W_{V,\ell} \in \mathbb{R}^{n\times 64}, \quad W_{V,\ell} \in \mathbb{R}^{512\times 64} \quad \text{for} \quad \ell = 1, ..., 8 \end{aligned}$$

For each head $\ell = 1, 2, ..., 8$ the similarity matrix S_ℓ for the target sentence is defined as

$$S_\ell \triangleq \frac{Q_\ell K_\ell^T}{\sqrt{64}} = \frac{1}{\sqrt{64}}\begin{bmatrix} q^{(1)}k^{(1)T} & q^{(1)}k^{(2)T} & \cdots & q^{(1)}k^{(n)T} \\ q^{(2)}k^{(1)T} & q^{(2)}k^{(2)T} & \cdots & q^{(2)}k^{(n)T} \\ \vdots & \vdots & \cdots & \vdots \\ q^{(n)}k^{(1)T} & q^{(n)}k^{(2)T} & \cdots & q^{(n)}k^{(n)T} \end{bmatrix} = \begin{bmatrix} s_{11}^{(\ell)} & s_{12}^{(\ell)} & \cdots & s_{1n}^{(\ell)} \\ s_{21}^{(\ell)} & s_{22}^{(\ell)} & \cdots & s_{2n}^{(\ell)} \\ \vdots & \vdots & \cdots & \vdots \\ s_{n1}^{(\ell)} & s_{n2}^{(\ell)} & \cdots & s_{nn}^{(\ell)} \end{bmatrix}.$$

Define the mask M of the matrix S_ℓ as

$$M(S_\ell) = \begin{bmatrix} s_{11}^{(\ell)} & -\infty & -\infty & \cdots & -\infty & -\infty \\ s_{21}^{(\ell)} & s_{22}^{(\ell)} & -\infty & \cdots & -\infty & -\infty \\ s_{31}^{(\ell)} & s_{32}^{(\ell)} & s_{33}^{(\ell)} & -\infty & -\infty & -\infty \\ \cdots & \cdots & \cdots & \ddots & -\infty & -\infty \\ s_{n-1,1}^{(\ell)} & s_{n-1,2}^{(\ell)} & s_{n-1,3}^{(\ell)} & \cdots & s_{n-1,n-1}^{(\ell)} & -\infty \\ s_{n1}^{(\ell)} & s_{n2}^{(\ell)} & s_{n3}^{(\ell)} & \cdots & s_{n,n-1}^{(\ell)} & s_{nn}^{(\ell)} \end{bmatrix}.$$

That is

$$M(S_\ell(i,j)) = \begin{cases} s_{ij}^{(\ell)}, & j \le i \\ -\infty, & j > i \end{cases} \quad \text{for } \ i,j = 1,2,...,n.$$

Proceeding, for each head $\ell = 1,2,...,8$ we take the softmax of each row of $M(S_\ell)$ to obtain

$$\text{softmax}\left(M(S_\ell)\right) = \begin{bmatrix} p_{11}^{(\ell)} & 0 & 0 & 0 & \cdots & 0 \\ p_{21}^{(\ell)} & p_{22}^{(\ell)} & 0 & 0 & \cdots & 0 \\ p_{31}^{(\ell)} & p_{32}^{(\ell)} & p_{33}^{(\ell)} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{n-1,1}^{(\ell)} & p_{n-1,2}^{(\ell)} & p_{n-1,3}^{(\ell)} & \cdots & p_{n-1,n-1}^{(\ell)} & 0 \\ p_{n1}^{(\ell)} & p_{n2}^{(\ell)} & p_{n3}^{(\ell)} & \cdots & p_{n,n-1}^{(\ell)} & p_{nn}^{(\ell)} \end{bmatrix}.$$

Then the attention (row) vectors of the target words are

$$Z_Y^{(\ell)} = \begin{bmatrix} z^{\ell(1)} \\ z^{\ell(2)} \\ \vdots \\ z^{\ell(n)} \end{bmatrix} = \text{softmax}\left(M(S_\ell)\right) V_\ell \ = \begin{bmatrix} p_{11}^{(\ell)} & 0 & 0 & 0 & \cdots & 0 \\ p_{21}^{(\ell)} & p_{22}^{(\ell)} & 0 & 0 & \cdots & 0 \\ p_{31}^{(\ell)} & p_{32}^{(\ell)} & p_{33}^{(\ell)} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & 0 \\ p_{n-1,1}^{(\ell)} & p_{n-1,2}^{(\ell)} & p_{n-1,3}^{(\ell)} & \cdots & p_{n-1,n-1}^{(\ell)} & 0 \\ p_{n1}^{(\ell)} & p_{n2}^{(\ell)} & p_{n3}^{(\ell)} & \cdots & p_{n,n-1}^{(\ell)} & p_{nn}^{(\ell)} \end{bmatrix} \begin{bmatrix} v^{\ell(1)} \\ v^{\ell(2)} \\ v^{\ell(3)} \\ \vdots \\ v^{\ell(n-1)} \\ v^{\ell(n)} \end{bmatrix} \in \mathbb{R}^{n\times 64}$$

$$= \begin{bmatrix} \sum_{j=1}^{n} p_{1j} \left[\quad v^{\ell(j)} \quad \right] \\ \sum_{j=1}^{n} p_{2j} \left[\quad v^{\ell(j)} \quad \right] \\ \sum_{j=1}^{n} p_{3j} \left[\quad v^{\ell(j)} \quad \right] \\ \vdots \\ \sum_{j=1}^{n} p_{n-1,j} \left[\quad v^{\ell(j)} \quad \right] \\ \sum_{j=1}^{n} p_{nj} \left[\quad v^{\ell(j)} \quad \right] \end{bmatrix} \in \mathbb{R}^{n\times 64}$$

This masking forces *causality* on the training process. That is, when the i^{th} word $x^{(i)}$ of the input sentence is presented to the encoder, the decoder will only have information on the words $y^{(j)}$ for $j \le i$ of the target sentence.

Word Embedding (Tokens)

Suppose we are translating from French to English. Let's restrict vocabulary of both languages are restricted to the 10,000 most commonly used words. The end of sentence word, the start of sentence word and the empty word (no word) are always assumed to be in both vocabularies.

In both of vocabularies each word is first encoded as a one hot vector $x_{\text{one\_hot}} \in \mathbb{R}^{1\times 10000}$. With $W_{\text{embedded}} \in \mathbb{R}^{10000\times 512}$ this one hot vector is then embedded into $\mathbb{R}^{1\times 512}$ according to

$$x = x_{\text{one\_hot}} W_{\text{embedded}} \in \mathbb{R}^{1\times 512}.$$

The embedded sentence is given by

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} = \begin{bmatrix} x^{(1)}_{\text{one\_hot}} \\ x^{(2)}_{\text{one\_hot}} \\ \vdots \\ x^{(n)}_{\text{one\_hot}} \end{bmatrix} W_{\text{embedded}} \in \mathbb{R}^{n\times 512}.$$

The embedding for the target sentence is similar. W_{embedded} is a learned parameter.

Position Encoding

Order matters in the translation of one language to another. To accommodate this position vectors are added to the embedded words as follows. Let k denote the position of the word in the sentence for $k = 1, 2, ..., n$. For the sake of simplicity let the maximum sentence size n be an even number. Then define the n frequencies ω_i by

$$\omega_i \triangleq \left(\frac{1}{10000^{2/512}}\right)^i \quad \text{for} \quad i = 0, 1, ..., n.$$

Define the position vector $p^{(k)} \in \mathbb{R}^{1\times 512}$ for the k^{th} input word as

$$p^{(k)} = \begin{bmatrix} p_1^{(k)} & p_2^{(k)} & \cdots & p_{512}^{(k)} \end{bmatrix} \in \mathbb{R}^{1\times 512}$$

where

$$p_i^{(k)} \triangleq \begin{cases} \cos(k\omega_i) & i = 1, 3, 5, ..., n-1 \\ \sin(k\omega_i) & i = 2, 4, 6, ..., n. \end{cases}$$

For example, the position encoder vector for the 20^{th} word in the input sentence is

$$p^{(20)} = \begin{bmatrix} \cos(20\omega_1) & \sin(20\omega_2) & \cos(20\omega_3) & \sin(20\omega_4) & \cos(20\omega_5) & \sin(20\omega_6) & \cos(20\omega_7) & \cdots & \sin(20\omega_{512}) \end{bmatrix} \in \mathbb{R}^{1\times 512}.$$

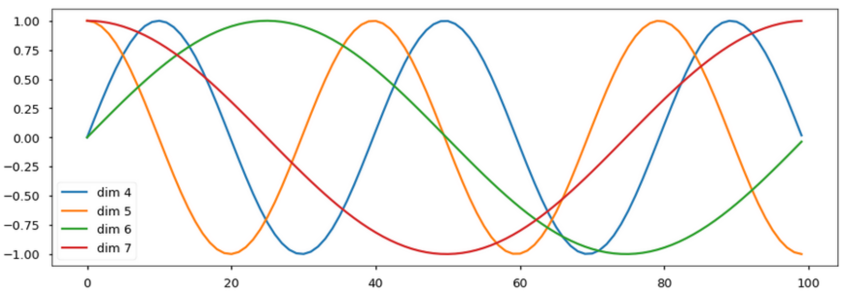


Figure 4: Position Encoding

As Figure 1 indicates, these position encoding vectors are added to the embedded words. That is,

$$X \longleftarrow \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} + \begin{bmatrix} p^{(1)} \\ p^{(2)} \\ \vdots \\ p^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times 512}$$

It might seem more intuitive to concatenate $x^{(1)}, p^{(i)}$ rather than add them together so as to not mix in the position with the word in the sentence. That is,

$$\begin{bmatrix} x^{(i)} & p^{(i)} \end{bmatrix} = \begin{bmatrix} x_1^{(k)} & x_2^{(k)} & \cdots & x_{512}^{(k)} & p_1^{(k)} & p_2^{(k)} & \cdots & p_{512}^{(k)} \end{bmatrix} \in \mathbb{R}^{1\times 1024}.$$

However, simply adding the two seems to work well enough and thus concatenation is not done.

Training

Suppose the input sentence 3 words and the target sentence is 5 words.

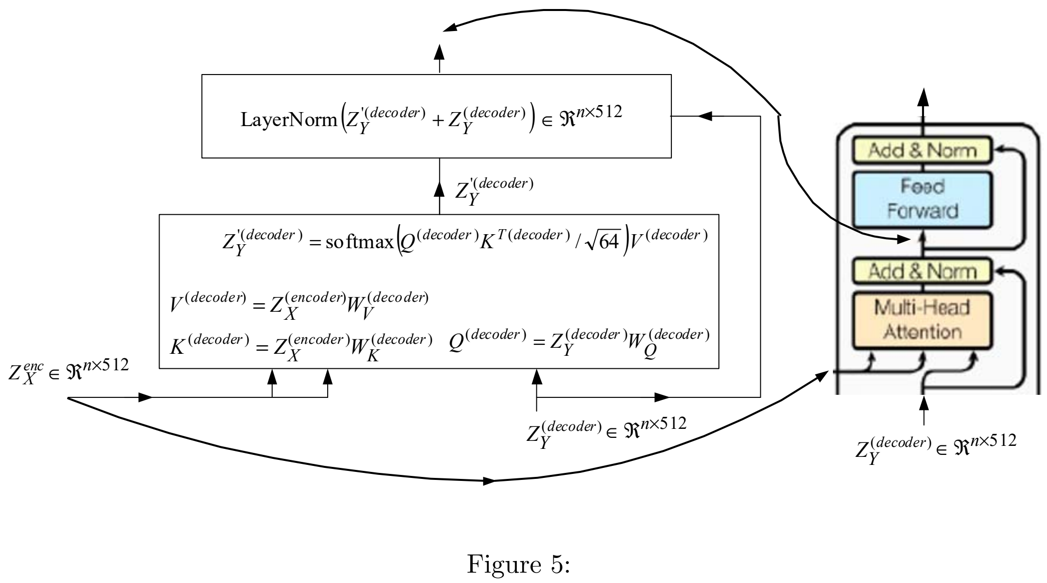
$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(\text{eos})} \\ x^{(\text{empty})} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n\times 512}, \quad Y = \begin{bmatrix} y^{(\text{start})} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ y^{(5)} \\ y^{(6)} \\ y^{(\text{eos})} \\ y^{(\text{empty})} \\ \vdots \\ y^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n\times 512}$$

Then

$$X \to Z_X^{(\text{encoder})} \in \mathbb{R}^{n\times 512}, \quad Y \to Z_Y^{(\text{decoder})} \in \mathbb{R}^{n\times 512}$$

and

$$\begin{aligned} K^{(\text{decoder})} &\triangleq Z_X^{(\text{encoder})} W_K^{(\text{decoder})} \in \mathbb{R}^{n\times 512} \\ V^{(\text{decoder})} &\triangleq Z_X^{(\text{encoder})} W_V^{(\text{decoder})} \in \mathbb{R}^{n\times 512} \\ Q^{(\text{decoder})} &\triangleq Z_Y^{(\text{decoder})} W_Q^{(\text{decoder})} \in \mathbb{R}^{n\times 512} \end{aligned}$$



$\text{LayerNorm}\left(Z_Y^{'(decoder)} + Z_Y^{(decoder)}\right) \in \Re^{n\times 512}$
$Z_Y^{'(decoder)}$
$Z_Y^{'(decoder)} = \text{softmax}\left(Q^{(decoder)} K^{T(decoder)} / \sqrt{64}\right) V^{(decoder)}$
$V^{(decoder)} = Z_X^{(encoder)} W_V^{(decoder)}$
$K^{(decoder)} = Z_X^{(encoder)} W_K^{(decoder)}$
$Q^{(decoder)} = Z_Y^{(decoder)} W_Q^{(decoder)}$
$Z_X^{enc} \in \Re^{n\times 512}$
$Z_Y^{(decoder)} \in \Re^{n\times 512}$
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
$Z_Y^{(decoder)} \in \Re^{n\times 512}$

Figure 5:

$$\begin{aligned} Z_Y'^{(\text{decoder})} &= \text{softmax}\left(\frac{Q^{(\text{decoder})} K^{T(\text{decoder})}}{\sqrt{64}}\right) V^{(\text{decoder})} \in \mathbb{R}^{n\times 512} \\ Z_Y''^{(\text{decoder})} &= \text{LayerNorm}\left(Z_Y'^{(\text{decoder})} + Z_Y^{(\text{decoder})}\right) \end{aligned}$$

Let $W_1^{(\text{decoder})} \in \mathbb{R}^{512\times 1024}, b_1^{(\text{decoder})} \in \mathbb{R}^{1\times 1024}, W_2^{(\text{decoder})} \in \mathbb{R}^{1024\times 512}, b_2^{(\text{decoder})} \in \mathbb{R}^{1\times 512}$ and

$$B_1^{(\text{decoder})} \triangleq \begin{bmatrix} b_1 \\ b_1 \\ \vdots \\ b_1 \end{bmatrix} \in \mathbb{R}^{n\times 1024}, \quad B_2^{(\text{decoder})} \triangleq \begin{bmatrix} b_2 \\ b_2 \\ \vdots \\ b_2 \end{bmatrix} \in \mathbb{R}^{n\times 512}.$$

Note the each row of B_1 is the same as is each row of B_2. The quantities W_1, b_1, W_2, b_2 are all learnable parameters (updated by backpropagation). The feedforward operation transforms Z'_X to

$$Z_Y^{\text{ff}} \triangleq \left(\max(0_{n\times 1024}, Z''_Y W_1^{(\text{decoder})} \; + B_1^{(\text{decoder})})\right) W_2^{(\text{decoder})} + B_2^{(\text{decoder})} \in \mathbb{R}^{n\times 512}.$$

The residual addition followed by layer normalization is done in the same manner as above to get

$$Z_Y^{(\text{decoder\_out})} \triangleq \text{LayerNorm}(Z_Y^{\text{ff}} + Z''_Y) \in \mathbb{R}^{n\times 512}.$$

Now with $W^O \in \mathbb{R}^{512\times 10000}$ compute (10000 is the vocabulary size)

$$\begin{aligned} P &= \{p_{ij}\},\ i = 1, ..., n, j = 1, ..., 10000 \\ &\triangleq \text{softmax}\left(Z_Y^{(\text{decoder\_out})} W_O\right) \in \mathbb{R}^{n\times 10000} \end{aligned}$$

where the softmax is computed row by row and W_O is a learnable parameter. From the target sentence (training data) let $\ell_1, \ell_2, ..., \ell_n$ be the index in each row of P corresponding to the correct word. Then the maximum likelihood cost for the sentence is given by

$$C = -\sum_{i=1}^{n} \ln(p_{i,\ell_i}).$$

Backpropagation is done sentence by sentence (view a sentence as a batch).

Inference

During inference one has

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(\text{eos})} \\ x^{(\text{empty})} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n\times 512},$$

but no target sentence. So one sets the target to be

$$Y^{(1)} = \begin{bmatrix} y^{(\text{start})} \\ y^{(\text{empty})} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n\times 512}.$$

X, Y get passed through the encoder and decoder respectively. The output gives a predicted Y with the first row predicted to be $y^{(\text{start})}$ and the second row $y^{(2)}$ predicting the first word in the translation. Now set

$$Y^{(2)} = \begin{bmatrix} y^{(\text{start})} \\ y^{(2)} \\ y^{(\text{empty})} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n \times 512}.$$

and input X and $Y^{(2)}$ into the encoder and decoder respectively. The output should gives a predicted Y with the first row predicted to be $y^{(\text{start})}$, the second row $y^{(2)}$ predicting the first word in the translation, and the third row $y^{(3)}$ predicting the second word in the translation. Set

$$Y^{(3)} = \begin{bmatrix} y^{(\text{start})} \\ y^{(2)} \\ y^{(3)} \\ y^{(\text{empty})} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ y^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n \times 512}.$$

This process is continued until the end of sentence word $y^{(\text{eos})}$ is predicted.

References

https://doi.org/10.48550/arXiv.1706.03762 (Attention Is All You Need, Vaswani et al.)
https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/ (The Transformer: Attention is All You Need, Rachel Draelos)
https://nlp.seas.harvard.edu/2018/04/03/attention.html (The Annotated Transformer, Alexander Rush)

===================================

Before going further, let's go back to the input sentence having 3 words and the output target sentence having 5 words. Then the relevant parts of $Q^{(\text{encoder})}K^{T(\text{decoder})}$ are then

$$\text{softmax}\left(\frac{1}{\sqrt{64}}\begin{bmatrix} q^{(1)(\text{encoder})} \\ q^{(2)(\text{encoder})} \\ q^{(3)(\text{encoder})} \\ q^{(4)(\text{encoder})} \\ q^{(5)(\text{encoder})} \\ q^{(6)(\text{encoder})} \end{bmatrix}\begin{bmatrix} k_1^{T(\text{decoder})} & k_2^{T(\text{decoder})} & k_3^{T(\text{decoder})} \end{bmatrix}\right)\begin{bmatrix} v^{(1)(\text{decoder})} \\ v^{(2)(\text{decoder})} \\ v^{(3)(\text{decoder})} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \\ p_{41} & p_{42} & p_{43} \\ p_{51} & p_{52} & p_{53} \\ p_{61} & p_{62} & p_{63} \end{bmatrix}\begin{bmatrix} v^{(1)(\text{decoder})} \\ v^{(2)(\text{decoder})} \\ v^{(3)(\text{decoder})} \end{bmatrix}$$

That is, the first six rows of $Q^{(\text{encoder})}$ will query the first three key vectors (corresponding to the three input words)

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(\text{eos})} \\ x^{(\text{empty})} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n\times 512}, \quad Y = \begin{bmatrix} y^{(\text{start})} \\ y^{(2)} \\ y^{(3)} \\ y^{(4)} \\ y^{(5)} \\ y^{(6)} \\ y^{(\text{eos})} \\ y^{(\text{empty})} \\ \vdots \\ y^{(\text{empty})} \end{bmatrix} \in \mathbb{R}^{n\times 512}$$

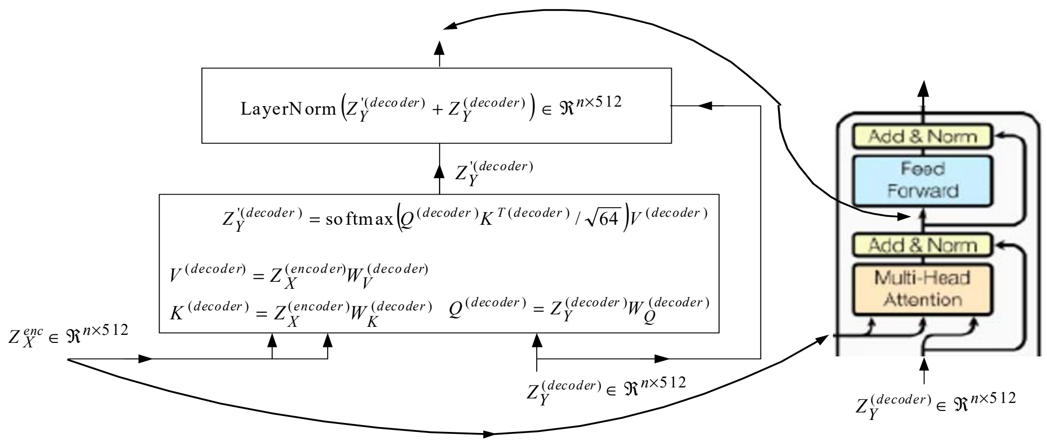

$\text{LayerNorm}\left(Z_Y^{'(decoder)} + Z_Y^{(decoder)}\right) \in \Re^{n\times 512}$
$Z_Y^{'(decoder)}$
$Z_Y^{'(decoder)} = \text{softmax}\left(Q^{(decoder)}K^{T(decoder)}/\sqrt{64}\right)V^{(decoder)}$
$V^{(decoder)} = Z_X^{(encoder)}W_V^{(decoder)}$
$K^{(decoder)} = Z_X^{(encoder)}W_K^{(decoder)}$
$Q^{(decoder)} = Z_Y^{(decoder)}W_Q^{(decoder)}$
$Z_X^{enc} \in \Re^{n\times 512}$
$Z_Y^{(decoder)} \in \Re^{n\times 512}$
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
$Z_Y^{(decoder)} \in \Re^{n\times 512}$

Figure 6:

===================================

The query is from the decoder 10x512 QK^T is 10x5
key and value from encoder and 5x512