ChatGPT

Johannes and Johannes

January 19, 2023

Outline

Attention

Attention

- (Scaled dot-product) *self-attention* relates different positions of a single sequence in order to represent the entire sequence.
- Takes a series of vectors as input and will result in adjusted vectors of the same dimension.
- In practice, calculation happens in matrix form:

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

with Q being a matrix of query vectors, K being a matrix of key vectors and V being a matrix of value vectors.

- We'll first demonstrate (scaled dot-product) *self-attention* for a single vector

Source: Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

Consider these arbitrary inputs and weight matrices:

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad \vec{b} = \begin{pmatrix} 2 \\ 4 \\ 1 \\ 3 \end{pmatrix} \quad \vec{c} = \begin{pmatrix} 1 \\ 4 \\ 3 \\ 2 \end{pmatrix} \quad \vec{d} = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 4 \end{pmatrix}$$

$$W^Q = \begin{bmatrix} 0.5 & 2 & 0.5 & 2 \\ 2 & 0.5 & 2 & 0.5 \\ 0.5 & 2 & 0.5 & 2 \\ 2 & 0.5 & 2 & 0.5 \end{bmatrix} \quad W^K = \begin{bmatrix} 0.5 & 1 & 1.5 & 2 \\ 1 & 1.5 & 2 & 0.5 \\ 1.5 & 2 & 0.5 & 1 \\ 2 & 0.5 & 1 & 1.5 \end{bmatrix} \quad W^V = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

In practice, the weight matrices are subject to training.

Dissecting Self-Attention

Multiply inpuct vectors with weight matrices

$$\vec{q_a} = \begin{pmatrix} 14 \\ 11 \\ 14 \\ 11 \end{pmatrix} \quad \vec{q_b} = \begin{pmatrix} 15.5 \\ 9.5 \\ 15.5 \\ 9.5 \end{pmatrix} \quad \vec{q_c} = \begin{pmatrix} 14 \\ 11 \\ 14 \\ 11 \end{pmatrix} \quad \vec{q_d} = \begin{pmatrix} 12.5 \\ 12.5 \\ 12.5 \\ 12.5 \end{pmatrix}$$

$$\vec{k_a} = \begin{pmatrix} 15 \\ 12 \\ 11 \\ 12 \end{pmatrix} \quad \vec{k_b} = \begin{pmatrix} 12.5 \\ 11.5 \\ 14.5 \\ 11.5 \end{pmatrix} \quad \vec{k_c} = \begin{pmatrix} 13 \\ 14 \\ 13 \\ 10 \end{pmatrix} \quad \vec{k_d} = \begin{pmatrix} 13.5 \\ 10.5 \\ 11.5 \\ 14.5 \end{pmatrix}$$

$$\vec{v_a} = \begin{pmatrix} 9 \\ 8 \\ 7 \\ 6 \end{pmatrix} \quad \vec{v_b} = \begin{pmatrix} 8 \\ 6 \\ 9 \\ 7 \end{pmatrix} \quad \vec{v_c} = \begin{pmatrix} 9 \\ 6 \\ 7 \\ 8 \end{pmatrix} \quad \vec{v_d} = \begin{pmatrix} 7 \\ 9 \\ 8 \\ 6 \end{pmatrix}$$

Dissecting Self-Attention, exemplary for $\vec{a}$

Calculate dot product of $\vec{q_a}$ with all $\vec{k_i}$

$$
\begin{aligned}
\vec{q_a} \circ \vec{k_a} &= 14 \cdot 15 + 11 \cdot 12 + 14 \cdot 11 + 11 \cdot 12 &&= 628 \\
\vec{q_a} \circ \vec{k_b} &= \dots &&= 631 \\
\vec{q_a} \circ \vec{k_c} &= \dots &&= 628 \\
\vec{q_a} \circ \vec{k_d} &= \dots &&= 625
\end{aligned}
$$

Divide by $\sqrt{d_k}$

$$
\vec{x} = \begin{pmatrix} 314 \\ 315.5 \\ 314 \\ 312.5 \end{pmatrix}
$$

In this step, relation between the input vector $\vec{a}$ and the other vectors happens.

Dissecting Self-Attention, exemplary for $\vec{a}$

$$\text{softmax}\left(\vec{x}\right) = \frac{e^{\vec{x}}}{\sum_{i=1}^{d_x} e^{\vec{x_i}}} = \frac{e^{\vec{x}}}{e^{314} + e^{315.5} + e^{314} + e^{312.5}} \approx \begin{pmatrix} 0.149 \\ 0.668 \\ 0.149 \\ 0.033 \end{pmatrix}$$

Dissecting Self-Attention, exemplary for $\vec{a}$

Multiply $\vec{v_i}$ with the respective softmax **result**

$$\vec{v_a'} = \vec{v_a} \cdot 0.149 \approx \begin{pmatrix} 1.342 \\ 1.193 \\ 1.044 \\ 0.895 \end{pmatrix} \quad \vec{v_b'} = \vec{v_b} \cdot 0.668 \approx \begin{pmatrix} 5.347 \\ 4.011 \\ 6.016 \\ 4.679 \end{pmatrix}$$

$$\vec{v_c'} = \vec{v_c} \cdot 0.149 \approx \begin{pmatrix} 1.342 \\ 0.895 \\ 1.044 \\ 1.193 \end{pmatrix} \quad \vec{v_d'} = \vec{v_d} \cdot 0.033 \approx \begin{pmatrix} 0.233 \\ 0.300 \\ 0.266 \\ 0.200 \end{pmatrix}$$

This way, less important relations receive lower attention than those with higher $\vec{q} \circ \vec{k}$.

Dissecting Self-Attention, exemplary for $\vec{a}$

Sum all $\vec{v_i'}$ vectors to get $\vec{z_a}$

$$\begin{pmatrix}1.342\\1.193\\1.044\\0.895\end{pmatrix} + \begin{pmatrix}5.347\\4.011\\6.016\\4.679\end{pmatrix} + \begin{pmatrix}1.342\\0.895\\1.044\\1.193\end{pmatrix} + \begin{pmatrix}0.233\\0.300\\0.266\\0.200\end{pmatrix} \approx \begin{pmatrix}8.265\\6.398\\8.370\\6.967\end{pmatrix}$$

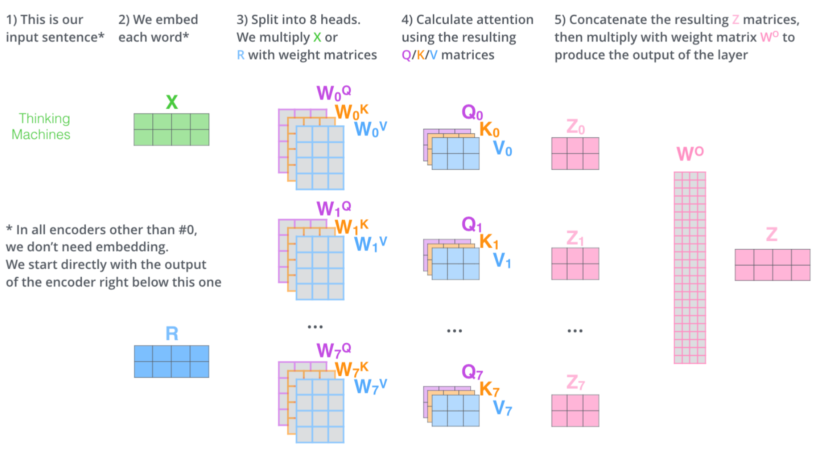
! Note that the order of magnitude of the cells of $\vec{z_a}$ equals the one from the most related value vector $\vec{v_b}$ (reminder: $\vec{v_b} = \langle 8, 6, 9, 7 \rangle$).

Self-Attention in matrix form

$$Z = \begin{bmatrix} 8.265 & 6.398 & 8.370 & 6.967 \\ 8.088 & 6.097 & 8.817 & 7.000 \\ 8.265 & 6.398 & 8.370 & 6.967 \\ 9.250 & 7.250 & 7.750 & 6.75 \end{bmatrix}$$

Multi-head attention

In practice, multiple Z matrices are computed, each resulting from different weight-matrices.



Source: `https://jalammar.github.io/illustrated-transformer/`

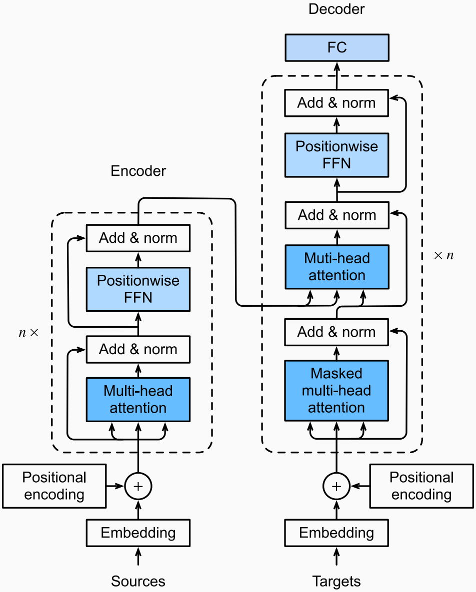
Transformers

Advantages over RNNs

- No problem with long-range dependencies
  - Input is read all at once
- Slim architecture
  - No need for rolling out backprop over many layers
  - No vanishing/exploding gradients
  - Fewer parameters
- Heavy parallelization is possible
  - No need for sequential computations

Source: `https://towardsdatascience.com/`

`all-you-need-to-know-about-attention-and-transformers-in-depth-understanding-part-1-552f0b41d021`



Source: `2dl.ai`

ChatGPT/InstructGPT

GPT-N, InstructGPT and ChatGPT

- The *GPT-N* (generative pre-trained transformer) series of transformers generally follow the aforementioned transformer architecture.
- There is no paper describing the process for training ChatGPT, but there is InstructGPT
- InstructGPT is based on GPT-3, followed by *reinforcement learning with human feedback*.

Source: Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022).

Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*. 2022.

Conflict between LM objective and user expectation

[...] the language modeling objective used for many recent large LMs — predicting the next token on a webpage from the internet — is different from the objective 'follow the user's instructions helpfully and safely'

Source: Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022).

Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*. 2022.