A Mathematical View of Attention Models in Deep Learning

Shuiwang Ji
Department of Computer Science & Engineering
Texas A&M University

Attention Model

1. Given a set of n query vectors $q_1, q_2, \cdots, q_n \in \mathbb{R}^d$, m key vectors $k_1, k_2, \cdots, k_m \in \mathbb{R}^d$, and m value vectors $v_1, v_2, \cdots, v_m \in \mathbb{R}^p$, the attention mechanism computes a set of output vectors $o_1, o_2, \cdots, o_n \in \mathbb{R}^q$ by linearly combining the g-transformed value vectors $g(v_i) \in \mathbb{R}^q$ using the relations between the corresponding query vector and each key vector as coefficients.
2. Formally,

$$o_j = \frac{1}{C} \sum_{i=1}^{m} f(q_j, k_i) g(v_i), \tag{1}$$

where $f(q_j, k_i)$ characterizes the relation (*e.g.*, similarity) between q_j and k_i, $g(\cdot)$ is commonly a linear transformation as $g(v_i) = \boldsymbol{W}_v v_i \in \mathbb{R}^q$, where $\boldsymbol{W}_v \in \mathbb{R}^{q \times p}$, and $C = \sum_{i=1}^{m} f(q_j, k_i)$ is a normalization factor.

Attention Model

1. A commonly used similarity function is the embedded Gaussian, defined as $f(q_j, k_i) = \exp\left(\theta(q_j)^T \phi(k_i)\right)$, where $\theta(\cdot)$ and $\phi(\cdot)$ are commonly linear transformations as $\theta(q_j) = \boldsymbol{W}_q q_j$ and $\phi(k_i) = \boldsymbol{W}_k k_i$.
2. Note that if we treat the value vectors as inputs, each output vector o_j is dependent on all input vectors. When the embedded Gaussian similarity and linear transformation are used, these computations can be expressed succinctly in matrix form as

$$\boldsymbol{O} = \boldsymbol{W}_v \boldsymbol{V} \times \text{softmax}\left((\boldsymbol{W}_k \boldsymbol{K})^T \boldsymbol{W}_q \boldsymbol{Q}\right), \tag{2}$$

   where $\boldsymbol{Q} = [q_1, q_2, \cdots, q_n] \in \mathbb{R}^{d \times n}$, $\boldsymbol{K} = [k_1, k_2, \cdots, k_m] \in \mathbb{R}^{d \times m}$, $\boldsymbol{V} = [v_1, v_2, \cdots, v_m] \in \mathbb{R}^{p \times m}$, $\boldsymbol{O} = [o_1, o_2, \cdots, o_n] \in \mathbb{R}^{q \times n}$, and softmax$(\cdot)$ computes a normalized version of the input matrix, where each column is normalized using the softmax function to sum to one.
3. Note that the number of output vectors is equal to the number of query vectors. In self-attention, we have $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V}$.

Attention Model

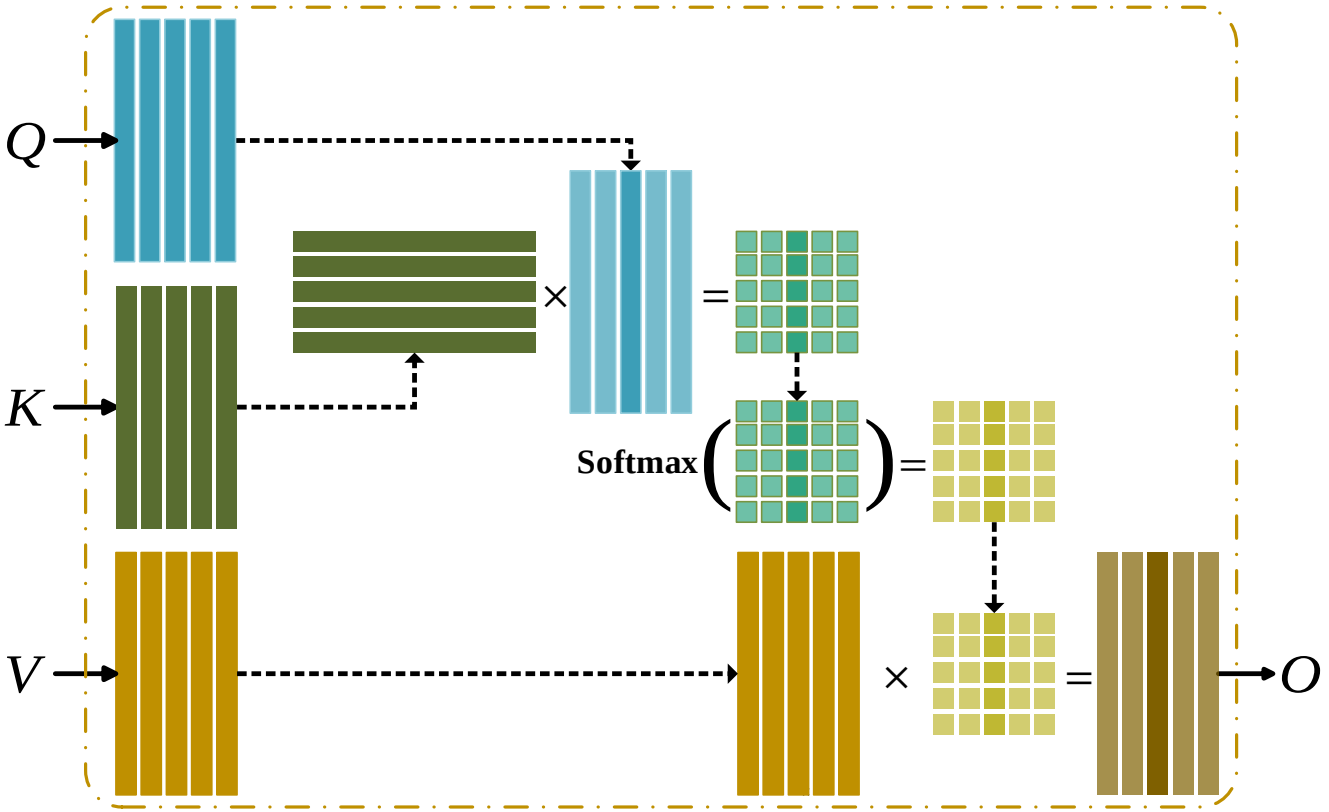

Figure: An illustration of the attention operator. Here, $\times$ denotes matrix multiplication, and Softmax$(\cdot)$ is the column-wise softmax operator. $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are input matrices. A similarity score is computed between each query vector as a column of $\boldsymbol{Q}$ and each key vector as a column in $\boldsymbol{K}$. Softmax$(\cdot)$ normalizes these scores and makes them sum to 1. Multiplication between normalized scores and the matrix $\boldsymbol{V}$ yields the corresponding output vector.

Attention for higher order data

1. The attention mechanism was originally developed in natural language processing to process 1-D data.
2. It has been extended to deal with 2-D images and 3-D video data recently.
3. When deal with 2-D data, the inputs to the attention operator can be represented as 3-D tensors $\mathcal{Q} \in \mathbb{R}^{h \times w \times c}$, $\mathcal{K} \in \mathbb{R}^{h \times w \times c}$, and $\mathcal{V} \in \mathbb{R}^{h \times w \times c}$, where h, w, and c represent the height, width, and number of channels, respectively. Note that for notational simplicity, we have assumed the three tensors having the same size.

Attention for higher order data

1. These tensors are first unfolded into matrices along mode-3, resulting in $\boldsymbol{Q}_{(3)}, \boldsymbol{K}_{(3)}, \boldsymbol{V}_{(3)} \in \mathbb{R}^{c \times hw}$.
2. Columns of these matrices are the mode-3 fibers of the corresponding tensors.
3. These matrices are used to compute output vectors as in regular attention described above. The output vectors are then folded back to a 3-D tensor $\mathcal{O} \in \mathbb{R}^{h \times w \times q}$ by treating them as mode-3 fibers of $\mathcal{O}$.
4. Note that the height and width of $\mathcal{O}$ are equal to those of $\mathcal{Q}$. That is, we can obtain an output with larger/smaller spatial size by providing an input $\mathcal{Q}$ of correspondingly larger/smaller spatial size.
5. Again, we have $\mathcal{Q} = \mathcal{K} = \mathcal{V}$ in self-attention.

Attention for higher order data

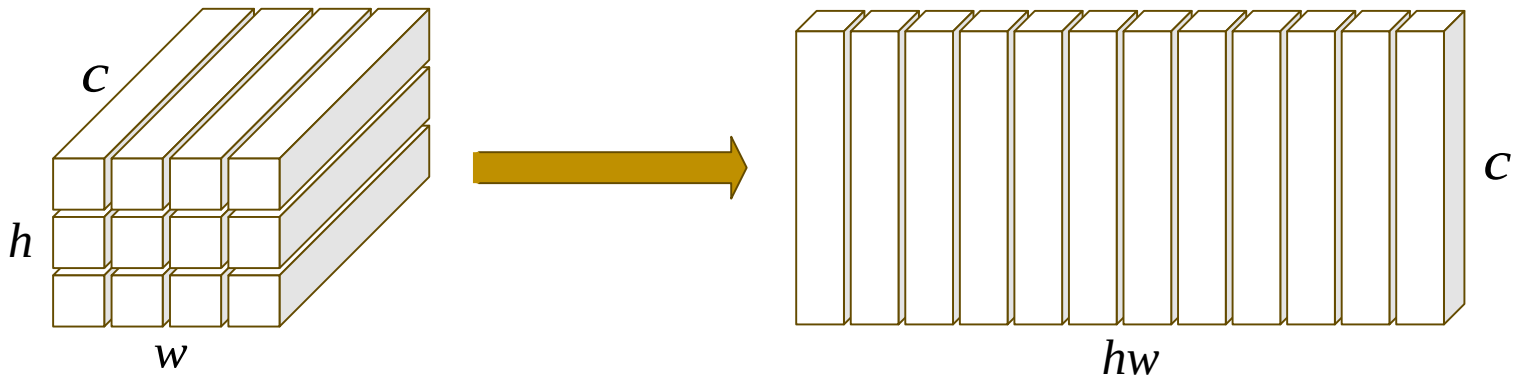

Figure: Conversion of a third-order tensor into a matrix by unfolding along mode-3. In this example, a $h \times w \times c$ tensor is unfolded into a $c \times hw$ matrix.

THANKS!