Generalized Attention Mechanism and Relative Position for Transformer

R. V. R. Pandya
rajavikram.pandya@outlook.com
July 22, 2022

Abstract

In this paper, we propose generalized attention mechanism (GAM) by first suggesting a new interpretation for self-attention mechanism of Vaswani et al. [5]. Following the interpretation, we provide description for different variants of attention mechanism which together form GAM. Further, we propose a new relative position representation within the framework of GAM. This representation can be easily utilized for cases in which elements next to each other in input sequence can be at random locations in actual dataset/corpus.

1 Introduction

Vaswani et al. [5] proposed self-attention mechanism based neural network for sequence transduction, namely Transformer, as computationally efficient alternative to recurrent and convolutional neural networks. Self-attention mechanism has been interpreted in terms of query, key and value of different elements of the sequence. Their work includes absolute position representation of these elements through sine and cosine functions. Later Shaw et al. [4] improved it by including relative position representation for different elements of sequence. Since then, within the framework of self-attention mechanism, different models have been suggested for relative position representation (see [1, 2] and references cited therein).

In this paper, we first describe self-attention mechanism as suggested by Vaswani et al. [5] using tensor notations. We use Einstein summation convention (i.e. summation over repeated indices) while writing various equations in tensor notations. An alternate interpretation for attention mechanism is suggested which does not require query and key terminology. Then we provide details of our generalized attention mechanism and inclusion of relative position.

2 Self-Attention Mechanism of Vaswani et al. [5, 4]

Consider input sequence $\mathbf{Y}$ of n elements $\mathbf{y}^\alpha$ where superscript $\alpha = 1, 2, 3, \ldots, n$ represents different elements of the sequence. For language processing each $\mathbf{y}^\alpha$ is a vector corresponding to a word/token ν^α. $\mathbf{Y}$ can be written as

$$\begin{aligned} \mathbf{Y} &= (\nu^1, \nu^2, \ldots, \nu^\alpha, \ldots, \nu^n) \\ &= (\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^\alpha, \ldots, \mathbf{y}^n). \end{aligned} \tag{1}$$

We use Einstein summation convention (i.e. summation over repeated indices) while writing various equations in tensor notations in this paper. Each element $\mathbf{y}^\alpha$ is tensor of rank $R \geq 0$ and in present case of language processing we limit our discussion to element being M-dimensional vector (tensor of $R = 1$), written as

$$\mathbf{y}^\alpha = (y_1^\alpha, y_2^\alpha, \ldots, y_M^\alpha) \equiv y_i^\alpha \text{ where } \{i = 1, 2, \ldots, M\}. \tag{2}$$

Similarly, the output sequence $\mathbf{Z}$ of n elements $\mathbf{z}^\alpha$ $(\alpha = 1, 2, 3, \ldots, n)$ can be written as

$$\mathbf{z}^\alpha = (z_1^\alpha, z_2^\alpha, \ldots, z_{M_v}^\alpha) \equiv z_i^\alpha \text{ where } \{i = 1, 2, \ldots, M_v\}. \tag{3}$$

For multi-head case, later we use ${}^a z_i^\alpha$ where superscript $a = 1, 2, \ldots, h$ on z is used to represent output $\mathbf{z}^\alpha$ in particular head a from self-attention sub-layer.

The three parameter matrices ${}^a\mathbf{W}^Q$, ${}^a\mathbf{W}^K$, ${}^a\mathbf{W}^V$ for query (Q), key (K) and value (V), respectively, for a particular head a are written using tensor notation as

$$\begin{aligned} {}^a\mathbf{W}^Q &= {}^aW_{ij}^Q, && (4) \\ {}^a\mathbf{W}^K &= {}^aW_{ij}^K, && (5) \\ {}^a\mathbf{W}^V &= {}^aW_{ij}^V, && (6) \\ & && (7) \end{aligned}$$

where superscript $a = 1, 2, \ldots, h$ on W is used to represent any particular head a in h multi-head. Also, subscripts i and j are used to represent different elements of matrices.

The jth component of output vector from self-attention sub-layer of head a, corresponding to element α of input sequence, is given by

$${}^a z_j^\alpha = \sum_{\beta=1}^{n} {}^a\Phi^{\alpha\beta}\, y_i^\beta\, ({}^aW_{ij}^V). \tag{8}$$

The weight coefficient ${}^a\Phi^{\alpha\beta}$ is given by

$${}^a\Phi^{\alpha\beta} = \frac{\exp({}^a e^{\alpha\beta})}{\sum_{\gamma=1}^{n} \exp({}^a e^{\alpha\gamma})}. \tag{9}$$

Also

$${}^a e^{\alpha\beta} = \frac{1}{\sqrt{d_k}} \left\{ y_r^\alpha\, ({}^aW_{rs}^Q) \right\} \left\{ y_t^\beta\, ({}^aW_{tu}^K) \right\} \delta_{su}, \tag{10}$$

where δ_{su} is Kronecker delta ($\delta_{su} = 1$ when $s = u$ and $\delta_{su} = 0$ when $s \neq u$). It should be noted that summation is implied on repeated indices in Eq. (10).

3 Present Generalized Attention Mechanism (GAM)

Consider Eq. (10) which can be also written as

$$ {}^{a}e^{\alpha\beta} = \Big\{ y_r^{\alpha}\, y_t^{\beta} \Big\}\, {}^{a}B_{rt}, \tag{11} $$

where

$$ {}^{a}B_{rt} = \Big\{ \frac{1}{\sqrt{d_k}} ({}^{a}W_{rs}^{Q})\, ({}^{a}W_{tu}^{K})\, \delta_{su} \Big\}. \tag{12} $$

It should be noted in view of Eq. (11) that self-attention mechanism of Vaswani et al. can be interpreted as combination of higher order features of input (i.e. $y_r^{\alpha}\, y_t^{\beta}$) and square matrix ${}^{a}\mathbf{B}$ whose components are denoted by ${}^{a}B_{rt}$. These higher order features along with ${}^{a}\mathbf{B}$ are responsible for evolving value vectors of elements of input sequence. The dot-product attention of Vaswani et al. can be completely described (for head a) using two parameter matrices ${}^{a}\mathbf{B}$, ${}^{a}\mathbf{W}^{V}$ instead of three parameter matrices ${}^{a}\mathbf{W}^{Q}$, ${}^{a}\mathbf{W}^{K}$, ${}^{a}\mathbf{W}^{V}$. And interpretation of their attention mechanism in terms of query and key may be abandoned.

We base generalized attention model (GAM) on Eqs. (8,9,11) requiring learnable parameter matrices ${}^{a}\mathbf{B}$ and ${}^{a}\mathbf{W}^{V}$. The GAM equations can be written as

$$ {}^{a}z_j^{\alpha} = \sum_{\beta=1}^{n} {}^{a}\Phi^{\alpha\beta}\, y_i^{\beta}\, ({}^{a}W_{ij}^{V}), \tag{13} $$

$$ {}^{a}\Phi^{\alpha\beta} = \sum_{i=1}^{N_B} W_i^{P}\, ({}^{a}\Psi_i^{\alpha\beta}), \tag{14} $$

where

$$ {}^{a}\Psi_i^{\alpha\beta} = \frac{\exp({}^{a}\epsilon_i^{\alpha\beta})}{\sum_{\gamma=1}^{n} \exp({}^{a}\epsilon_i^{\alpha\gamma})}, \tag{15} $$

$$ {}^{a}\epsilon_i^{\alpha\beta} = f(y_r^{\alpha}\, y_t^{\beta})\, \Big[{}^{(a,i)}B_{rt} \Big], \tag{16} $$

and N_B are number of different parameter matrix ${}^{(a,i)}\mathbf{B}$ in the same attention head. Each ${}^{(a,i)}\mathbf{B}$ of head a can be think of as different portion of 'brain' in the head. Also $f(\ldots)$ represents function of $y_r^{\alpha}\, y_t^{\beta}$ and superscript (a, i) on the left of B represents ith parameter matrix ${}^{(a,i)}\mathbf{B}$ in attention head a. Two functional form for $f(y_r^{\alpha}\, y_t^{\beta})$ of power law and polynomial types can be considered for GAM and which are written as

Power law type:

$$ f(y_r^{\alpha}\, y_t^{\beta}) = (y_r^{\alpha} y_t^{\beta})^{n_1},\ n_1 > 0, \tag{17} $$

Polynomial type:

$$ f(y_r^{\alpha}\, y_t^{\beta}) = \sum_{l=1}^{L} A_l (y_r^{\alpha} y_t^{\beta})^{l}. \tag{18} $$

where A_l's are learnable scalar parameters.

Also, W_i^P can be either considered equal to $1/N_B$ or following constraint can be utilized during learning:

$$\sum_{i=1}^{N_B} W_i^P = 1. \tag{19}$$

4 Relative Position in GAM

Different models exist for inclusion of relative position representation within the framework of self-attention mechanism, for example see references [4, 3, 1, 2]. Here we suggest yet another model to include effect of relative position of input elements within the framework of GAM.

In this section, we assume that position related information are not included in $\mathbf{y}^\alpha$ of GAM equations which are written above. Linear combination of contributions from relative position ${}^a\pi_j^\alpha$ and ${}^a z_j^\alpha$ become output of GAM and can be written as

$$[{}^a z_j^\alpha]_{total} = (c_1)\,({}^a z_j^\alpha) + (1 - c_1)\,({}^a\pi_j^\alpha),\ 0 < c_1 < 1, \tag{20}$$

where

$${}^a\pi_j^\alpha = \sum_{\beta=1}^{n} {}^a\Theta^{\alpha\beta}\, y_i^\beta\, ({}^a W_{ij}^V), \tag{21}$$

$${}^a\Theta^{\alpha\beta} = \sum_{i=1}^{N_B} W_i^S\, ({}^a\xi_i^{\alpha\beta}), \tag{22}$$

where

$${}^a\xi_i^{\alpha\beta} = \frac{\exp({}^a\delta_i^{\alpha\beta})}{\sum_{\gamma=1}^{n} \exp({}^a\delta_i^{\alpha\gamma})} \tag{23}$$

$${}^a\delta_i^{\alpha\beta} = f(p_r^\alpha\, p_t^\beta)\,\left[{}^{(a,i)}B_{rt}^P\right]. \tag{24}$$

The function $f(\ldots)$ of $p_r^\alpha\, p_t^\beta$ can be considered as power law type (Eq. 17) or polynomial type (Eq. 18). Here constraint on Another possibility for $[{}^a z_j^\alpha]_{total}$ which can be explored is geometric average, written as

$$[{}^a z_j^\alpha]_{total} = \sqrt{({}^a z_j^\alpha)({}^a\pi_j^\alpha)}. \tag{25}$$

Now we discuss methodology to obtain relative position vector p_r^α. Consider $\mathbf{p}^\alpha$ as embedded relative position vector corresponding to input element $\mathbf{y}^\alpha$. The dimension of $\mathbf{p}^\alpha$ is identical to that of $\mathbf{y}^\alpha$ and is equal to M. The embedded vector can be written as

$$\mathbf{p}^\alpha = (p_1^\alpha, p_2^\alpha, \ldots, p_M^\alpha) \equiv p_i^\alpha \text{ where } \{i = 1, 2, \ldots, M\}. \tag{26}$$

The embedded vector for different $\alpha = 1, 2, \ldots, n$ can be learned during training from known input relative position vector $\mathbf{r}^\alpha$ whose dimension is equal to n. $\mathbf{r}^\alpha$ can be written as

$$\mathbf{r}^\alpha = (r_1^\alpha, r_2^\alpha, \ldots, r_n^\alpha) \equiv r_i^\alpha \text{ where } \{i = 1, 2, \ldots, n\}, \tag{27}$$

where

$$r_\alpha^\alpha = 1, \tag{28}$$

$$r_i^\alpha = 2 + n_e \text{ when } \alpha \neq i. \tag{29}$$

Here n_e is number of elements (in actual dataset/corpus) between elements at location α and i of input sequence $\mathbf{Y}$. For example, consider actual corpus as

$$\text{My name is Vikram} \tag{30}$$

and $n = 3$ for input sequence $\mathbf{Y}$. When

$$\begin{aligned} \mathbf{Y} &= (\nu^1, \nu^2, \nu^3), && (31) \\ &= (My, name, is), && (32) \end{aligned}$$

input relative position vectors $\mathbf{r}^\alpha$ can be written as

$$\begin{aligned} \mathbf{r}^1 &= (r_1^1, r_2^1, r_3^1) = (1, 2, 3), && (33) \\ \mathbf{r}^2 &= (r_1^2, r_2^2, r_3^2) = (2, 1, 2), && (34) \\ \mathbf{r}^3 &= (r_1^3, r_2^3, r_3^3) = (3, 2, 1). && (35) \\ & && (36) \end{aligned}$$

And when

$$\begin{aligned} \mathbf{Y} &= (\nu^1, \nu^2, \nu^3) && (37) \\ &= (My, name, Vikram), && (38) \end{aligned}$$

$\mathbf{r}^\alpha$ can be written as

$$\begin{aligned} \mathbf{r}^1 &= (r_1^1, r_2^1, r_3^1) = (1, 2, 4), && (39) \\ \mathbf{r}^2 &= (r_1^2, r_2^2, r_3^2) = (2, 1, 3), && (40) \\ \mathbf{r}^3 &= (r_1^3, r_2^3, r_3^3) = (4, 3, 1). && (41) \\ & && (42) \end{aligned}$$

5 Conclusion

We have proposed generalized attention mechanism (GAM) which also includes a new way of representing relative position of elements in actual dataset/corpus. In doing so, we have suggested different interpretation for attention mechanism which abandons requirement of query and key. In GAM, similar to self-attention mechanism of Vaswani et al. [5], initial static vector representation of various elements of input sequence are transformed into value vectors. These value vectors evolve into dynamic representations $[{}^{a}z_{j}^{\alpha}]_{total}$ under the influence of interactions among different elements of the sequence and their relative positions. These interactions are quantified in terms of higher order features of input elements, their relative positions and parameter matrices ${}^{(a,i)}\mathbf{B}$. The study on performance of GAM for various experiments of language processing and comparison with results of self-attention mechanism of Vaswani et al. [5, 4] will be performed in near future. Also, the application of GAM to time series analysis will be explored in detail.

References

[1] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. *Computational Linguistics*, pages 1–31, 2021.

[2] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*, 2020.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[4] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.