NOTES ON ATTENTION AND POSITIONAL ENCODINGS

IORDAN GANEV

Version 1.0

Contents

1. Introduction

In this expository note, we collect basic constructions related to the attention mechanism used in transformer models, and to sequential positional encodings. The main references are [VSP+17, ZLLS23].

The starting point for our discussion of attention is: given a database of key-value pairs, and a query in the same space as the keys, what is a meaningful value to assign to the key? The approach we investigate involves assigning a weighted average of the values to a query, where the weights, also known as "attention scores", measure what proportion of the query's total "attention" each key captures. We mainly focus on "dot-product" attention where the attention scores are determined by the dot product of the query and each key (with extra normalization and standardization steps). Building on the core definition of attention, which appears in Section 2.2, we also define self-attention in Section 2.4, multi-head attention in Section 2.5, and masked attention in Section 2.6. Furthermore, in Section 2.7, we discuss attention kernels in general that give rise to different attention scores.

As a function of the queries, keys, and values, the attention map has various equivariances, most notably invariance with respect to permutations of the key-value pairs. This fact, which we formulate precisely in Section 2.3, necessitates positional encodings when the key-value pairs emerge from sequential data. In Section 3, we give an overview of such encodings as derived from a square matrix M and an evaluation vector $\mathbf{x}$. The j-th power M^j of the matrix corresponds to the j-th position in the sequence, and we obtain the additive encoding vector by evaluating these powers at the evaluation vector: $M^j\mathbf{x}$. This encoding vector is added to the j-th vector in our input data sequence. In Example 3.6, we derive the positional encodings corresponding to a particular orthogonal matrix M appearing in the literature.

Before proceeding, we remark on the notation used in this note.

Vectors and dot products. Unless specified otherwise, all vectors are column vectors, which we write as:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} = (x_1, \dots, x_d) \in \mathbb{R}^d$$

so that $\mathbf{x}^T = \begin{bmatrix} x_1 & \cdots & x_d \end{bmatrix} \in \mathbb{R}^{1\times d}$ is a row vector. The dot product of two vectors in $\mathbb{R}^d$ is $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y} = \sum_{i=1} x_i y_i \in \mathbb{R}$.

Batch size first. When grouping vectors into a matrix, we use the "batch size first" convention. For example, if $\mathbf{x}_1, \dots, \mathbf{x}_n$ are vectors in $\mathbb{R}^d$, then the corresponding matrix X is of size n by d, and has the row vector $\mathbf{x}_i^T$ as its i-th row.

Softmax. We define the softmax function as:

$$\mathsf{SoftMax} : (\mathbb{R} \cup \{-\infty\})^d \setminus \{-\infty\}^d \to \mathbb{R}^d, \qquad \mathsf{SoftMax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^d e^{x_j}}$$

where $e^{-\infty} := \lim_{x\to-\infty} e^x = 0$. To be clear, the domain of SoftMax is the set of all d-tuples with entries either in $\mathbb{R}$ or equal to $-\infty$, where at least one entry is in $\mathbb{R}$. The output of SoftMax lies in the set of $\mathbf{y} \in \mathbb{R}^d$ such that $0 \le y_i \le 1$ and $\sum_{i=1}^d y_i = 1$.

2. Attention

2.1. **Single query attention.** The basic premise of the attention mechanism is the following. Suppose we a database of n key-value pairs

$$\mathcal{D} = \{(\mathbf{k}_1, \mathbf{v}_1), \dots, (\mathbf{k}_n, \mathbf{v}_n)\} \subset \mathbb{R}^\ell \times \mathbb{R}^d$$

where each key is ℓ-dimensional and each value is d-dimensional. Given a query $\mathbf{q} \in \mathbb{R}^\ell$, how can we get an approximating matching value?

Our approach starts with the proposal to assign $\mathbf{q}$ to the linear combination of the values $\mathbf{v}_i$ where the i-coefficient is given by the dot product of $\mathbf{q}$ and key $\mathbf{k}_i$:

$$\mathbf{q} \mapsto \sum_{i=1}^n \langle \mathbf{q}, \mathbf{k}_i \rangle \mathbf{v}_i \qquad \in \mathbb{R}^d \tag{2.1}$$

This is motivated by the case where the $\mathbf{k}_i$ form an orthonormal basis, in which case $\mathbf{q}$ itself is uniquely a linear combination of the $\mathbf{k}_i$ with coefficients given by dot products. It is desirable to normalize coefficients to be positive and sum to one; the standard way to do so is to apply softmax to the coefficients:

$$\mathbf{q} \mapsto \sum_{i=1}^n \left(\frac{e^{\langle \mathbf{q}, \mathbf{k}_i \rangle}}{\sum_j e^{\langle \mathbf{q}, \mathbf{k}_j \rangle}} \right) \mathbf{v}_i \qquad \in \mathbb{R}^d \tag{2.2}$$

Hence, the query $\mathbf{q}$ defines a probability distribution on the discrete set $\{\mathbf{v}_i\}_{i=1}^n$ and we assign $\mathbf{q}$ to the expected value of this distribution. Finally, we add a standardization term to account for the increase in the dot product with the increase in dimension:

$$\mathbf{q} \mapsto \sum_{i=1}^{n} \left(\frac{\exp\left(\frac{\langle \mathbf{q}, \mathbf{k}_i \rangle}{\sqrt{\ell}}\right)}{\sum_j \exp\left(\frac{\langle \mathbf{q}, \mathbf{k}_j \rangle}{\sqrt{\ell}}\right)} \right) \mathbf{v}_i \qquad \in \mathbb{R}^d \tag{2.3}$$

The motivation for the standardization term is that, if the entries of $\mathbf{q}$ and the $\mathbf{k}_i$ are independent and identically distributed with mean 0 and variance 1, then the dot product $\langle \mathbf{q}, \mathbf{k}_i \rangle$ has mean 0 and variance ℓ[1].

2.2. **Batches.** Matrix notation allows one to write the equations above without indices. Let $K \in \mathbb{R}^{n \times \ell}$ and $V \in \mathbb{R}^{n \times d}$ be the matrices whose i-th rows are $\mathbf{k}_i^T$ and $\mathbf{v}_i^T$ (using the "batch size first" convention). Then Equations 2.1, 2.2, and 2.3 become, respectively:

$$\mathbf{q} \mapsto V^T K \mathbf{q} \qquad \mathbf{q} \mapsto V^T \mathsf{SoftMax}(K\mathbf{q}) \qquad \mathbf{q} \mapsto V^T \mathsf{SoftMax}\left(\frac{K\mathbf{q}}{\sqrt{\ell}}\right)$$

Furthermore, if we have multiple queries $\mathbf{q}_1, \ldots, \mathbf{q}_r$ we group them in a matrix $Q \in \mathbb{R}^{r \times \ell}$ whose i-th row is $\mathbf{q}_i^T$. Equations 2.1 and 2.2 now become:

$$Q \mapsto QK^T V \in \mathbb{R}^{r \times d} \qquad Q \mapsto \mathsf{SoftMax}_{\text{row}}(QK^T)V \in \mathbb{R}^{r \times d}$$

where the "batch size first" convention requires taking transposes, and softmax is applied row-wise. We isolate the batch version of Equation 2.3 as the definition of attention:

Definition 2.1. The *attention function* defined by the database $\mathcal{D} = (K, V)$ is:

$$\mathsf{Attention}_{\mathcal{D}} : \mathbb{R}^{r \times \ell} \to \mathbb{R}^{r \times d} \qquad Q \mapsto \mathsf{SoftMax}_{\text{row}}\left(\frac{QK^T}{\sqrt{\ell}}\right) V$$

More precisely, this is the standardized, normalized dot product attention function with r queries with respect to the database $\mathcal{D} = (K, V)$. As a function of both the database and the query, we set:

$$\mathsf{Att} : \mathbb{R}^{r \times \ell} \times \mathbb{R}^{n \times \ell} \times \mathbb{R}^{n \times d} \longrightarrow \mathbb{R}^{r \times d}$$
$$(Q\ ,\ K\ ,\ V) \mapsto \mathsf{Attention}_{\mathcal{D}}(Q) = \mathsf{SoftMax}_{\text{row}}\left(\frac{QK^T}{\sqrt{\ell}}\right) V$$

Remark 2.2. The linear regression approach to the query problem models the values as a linear function of the keys. The least squares solution assigns a batch of queries $Q \in \mathbb{R}^{r \times \ell}$ to the batch of outputs $Q(K^TK)^{-1}K^TV \in \mathbb{R}^{r \times d}$ as long as the matrix K^TK is invertible. So the dot product approach described above (without normalization or standardization) reduces to linear regression when the columns of K are orthonormal.

[1]More generally, if the common mean is μ and the common variance is σ^2, then the dot product has mean $n\mu^2$ and variance $n\sigma^2 + 2n\mu^2\sigma^2 + n\mu^4$.

2.3. **Equivariances.** In order to describe the symmetries of the attention function, we first collect observations about various group actions. The query space $\mathbb{R}^{r\times\ell}$ has a left action of the symmetric group S_r by permuting the rows, and a commuting right action of the general linear group $\mathrm{GL}_\ell(\mathbb{R})$ by matrix multiplication. Similar actions hold for the key space $\mathbb{R}^{n\times\ell}$, the value space $\mathbb{R}^{n\times d}$, and the output space $\mathbb{R}^{r\times d}$.

Lemma 2.3. *Let $\sigma \in S_n$ and $\tau \in S_r$ be permutations, let $g \in O(d) \subseteq \mathrm{GL}_\ell(\mathbb{R})$ be an orthogonal matrix, and let $h \in \mathrm{GL}_d(R)$ be an invertible matrix. We have:*

$$\mathsf{Att}\left(\tau Q g\ ,\ \sigma K g\ ,\ \sigma V h\right) = \tau \mathsf{Att}\left(Q\ ,\ K\ ,\ V\right) h$$

where juxtaposition indicates the action of a group element on a matrix, noting that the left and right actions commute in all cases.

Proof. The proof is a straightforward computation:

$$\begin{aligned}\mathsf{Att}\left(\tau Q g\ ,\ \sigma K g\ ,\ \sigma V h\right) &= \mathsf{SoftMax}_{\mathrm{row}}\left(\frac{\tau Q g g^T K^T \sigma^T}{\sqrt{\ell}}\right)\sigma V h \\ &= \tau \mathsf{SoftMax}_{\mathrm{row}}\left(\frac{\tau Q K^T}{\sqrt{\ell}}\right)\sigma^T \sigma V h \\ &= \tau \mathsf{SoftMax}_{\mathrm{row}}\left(\frac{\tau Q K^T}{\sqrt{\ell}}\right) V h = \tau \mathsf{Att}\left(Q\ ,\ K\ ,\ V\right) h\end{aligned}$$

where we use the facts that: (1) g is orthogonal, (2) the permutation σ, regarded as a matrix in $\mathrm{GL}_n(\mathbb{R})$, is also orthogonal, and (3) permutations commute with softmax. □

2.4. **Self-attention.** In *self-attention*, we have a single data matrix $X \in \mathbb{R}^{n\times p}$ which is transformed into queries, keys, and values via matrices:

$$S_Q \in \mathbb{R}^{p\times\ell}, \qquad S_K \in \mathbb{R}^{p\times\ell}, \qquad S_V \in \mathbb{R}^{p\times d}$$

The self-attention function is defined as:

$$\begin{aligned}&\mathsf{SelfAttention}_{\mathcal{S}} : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times d} \\ &\qquad X \mapsto \mathsf{Att}\left(XS_Q\ ,\ XS_K\ ,\ XS_V\right) = \mathsf{Softmax}_{\mathrm{row}}\left(\frac{XS_Q S_K^T X^T}{\sqrt{\ell}}\right) XS_V\end{aligned}$$

where $\mathcal{S} = (S_Q, S_K, S_V) \in \mathbb{R}^{p\times\ell} \times \mathbb{R}^{p\times\ell} \times \mathbb{R}^{p\times d}$. In other words, self-attention is obtained from attention with queries $Q = XS_Q$, keys $K = XS_K$, and values $V = XS_V$. In the case of self-attention, the number of samples is the same as the number of queries, i.e., $n = r$. A special case of self-attention is where $\ell = d = p$ and all the S matrices are the identity:

$$\mathsf{SelfAttention}_{(\mathrm{id},\mathrm{id},\mathrm{id})}(X) = \mathsf{Att}\left(X\ ,\ X\ ,\ X\right) = \mathsf{Softmax}_{\mathrm{row}}\left(\frac{XX^T}{\sqrt{\ell}}\right) X$$

2.5. **Multi-head attention.** In *multi-head attention* with H heads, we have a matrices

$$W_h^Q \in \mathbb{R}^{\ell \times p} \qquad W_h^K \in \mathbb{R}^{\ell \times p} \qquad W_h^V \in \mathbb{R}^{d \times d'}.$$

for $h = 1, \dots H$, and for some dimensions p and d'. We group these into a single object:

$$\mathcal{W} = \sum_{h=1}^{H} e_h \otimes (W_h^Q, W_h^K, W_h^V) \in \mathbb{R}^H \otimes \left(\mathbb{R}^{\ell \times p} \times \mathbb{R}^{\ell \times p} \times \mathbb{R}^{d \times d'} \right)$$

where $e_h{}_{h=1}^H$ is the standard basis of $\mathbb{R}^H$. These allow us to transform the original database into H different databases, and we define the multihead attention map as:

$$\begin{aligned} \mathsf{MultiHeadAtt}_{\mathcal{W}} : \mathbb{R}^{r \times \ell} \times \mathbb{R}^{n \times \ell} \times \mathbb{R}^{n \times d} &\longrightarrow \mathbb{R}^{r \times d'} \\ (Q\ ,\ K\ ,\ V) &\mapsto \sum_{h=1}^{H} \mathsf{Att}\left(QW_h^Q\ ,\ KW_h^K\ ,\ VW_h^V \right) \\ &= \sum_{h=1}^{H} \mathsf{Softmax}_{\mathrm{row}} \left(\frac{QW_h^Q \left(W_h^K\right)^T K^T}{\sqrt{\ell}} \right) VW_h^V \end{aligned}$$

Remark 2.4. In the literature, one often sees the introduction of an intermediate value dimension q, in which case the matrix W_h^V is of size $d \times q$, and there is an extra tensor $W^O \in \mathbb{R}^H \otimes \mathbb{R}^q \otimes \mathbb{R}^{d'}$, which defines a linear map $\mathbb{R}^H \otimes \mathbb{R}^q \to \mathbb{R}^{d'}$. Multihead attention is then defined as:

$$\left(\sum_{h=1}^{H} e_h \otimes \mathsf{Softmax}_{\mathrm{row}} \left(\frac{QW_h^Q \left(W_h^K\right)^T K^T}{\sqrt{\ell}} \right) VW_h^V \right) W^O \ \in \mathbb{R}^r \otimes \mathbb{R}^d \simeq \mathbb{R}^{r \times d'}$$

where the element in the outer parentheses is an element of $\mathbb{R}^H \otimes \mathbb{R}^r \otimes \mathbb{R}^q$. This set-up is equivalent to the above approach. To see this, first note that we can write $W^O = \sum_{h=1}^H e_h \otimes W_h^O$, so that multihead attention becomes:

$$\sum_{h=1}^{H} \mathsf{Softmax}_{\mathrm{row}} \left(\frac{QW_h^Q \left(W_h^K\right)^T K^T}{\sqrt{\ell}} \right) VW_h^V W_h^O \ \in \mathbb{R}^r \otimes \mathbb{R}^{d'} \simeq \mathbb{R}^{r \times d'}$$

Now we can replace $W_h^V W_h^O \in \mathbb{R}^{d \times d'}$ by a single matrix, arriving at our original W_h^V from above, with the intermediate value dimension disappearing.

2.6. **Masking.** Before discussing masking in the context of attention, we illustrate the underlying principle. For $1 \leq m < n$, set:

$$\mathrm{mask}_m = (0, 0, \dots, 0, -\infty, \dots, -\infty) \in (\mathbb{R} \cup \{-\infty\})^n$$

where the first m entries are zero and the rest negative infinity. For any vector $\mathbf{a} \in \mathbb{R}^n$, the sum $\mathbf{a} + \mathrm{mask}_m$ belongs to $(\mathbb{R} \cup \{-\infty\})^n$, and one can apply softmax to obtain:

$$\mathsf{SoftMax}(\mathbf{a} + \mathrm{mask}_m) = \mathsf{SoftMax}(a_1, \dots, a_m) \oplus (0, \dots, 0)$$

so that the softmax of the sum $\mathbf{a} + \mathrm{mask}$ is equal to softmax of the truncated vector $(a_1, \dots, a_m)$ extended with $n - m$ zeros. So the mask gives us a way to compute softmax

of a subvector of $\mathbf{a}$ without having to formally truncate the vector. More generally, for every nonempty subset $S \subseteq \{1, \ldots, n\}$, there is a mask defined as:

$$\text{mask}_i^S = \begin{cases} 0 & \text{if } i \in S \\ -\infty & \text{otherwise} \end{cases}$$

Then:

$$\mathsf{SoftMax}(\mathbf{a} + \text{mask}^S)_i = \begin{cases} \frac{e^{a_i}}{\sum_{s \in S} e^{a_s}} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

Or, equivalently,

$$\text{SoftMax}(\mathbf{a} + \text{mask}^S) = \frac{\exp(\mathbf{a}) \cdot_{\text{ptwise}} \exp(\text{mask}^S)}{\sum_{i=1}^n \exp(a_i) \exp(\text{mask}_i^S)}$$

where exp is applied entry-wise to vectors. Now we return to our discussion of attention. As above, let Q, K, and V be the query, key, and value matrices.

Definition 2.5. A *mask* is a r by n matrix Mask with entries in $\{0, -\infty\}$ with each row having at least one zero. Masked softmax attention is defined as:

$$\text{SoftMax}_{\text{row}}\left(\frac{QK^T + \text{Mask}}{\sqrt{\ell}}\right) V \in \mathbb{R}^{r \times d}$$

The most common mask is the lower-triangular "sequential" mask in the case $n = r$, which has (j, i) entry equal to 0 if $i \leq j$ and $-\infty$ otherwise. Masking is frequently combined with self-attention.

2.7. **Attention kernels.** The above discussion makes heavy use of the dot product, which is just one example of a similarity kernel. In this section, we investigate similarity kernels in general.

Definition 2.6. The attention function relative to a kernel $\alpha : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$ is defined as:

$$\begin{aligned} \mathsf{Att}_\alpha : \mathbb{R}^{r \times \ell} \times \mathbb{R}^{n \times \ell} \times \mathbb{R}^{n \times d} &\longrightarrow \mathbb{R}^{r \times d} \\ (Q\ ,\ K\ ,\ V) &\mapsto \mathsf{SoftMax}_{\text{row}}\left(\alpha(Q, K)\right) V \end{aligned}$$

where $\alpha(Q, K) \in \mathbb{R}^{r \times n}$ has (j, i) entry equal to $\alpha(\mathbf{q}_j, \mathbf{k}_i)$.

We elaborate on this definition before giving examples of kernels:

- Fix queries Q, keys K, and values V as above.
- Starting with the kernel $\alpha : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$, we obtain an "attention weight" for every query $\mathbf{q}_j$ and key $\mathbf{k}_i$:

  $$\alpha(\mathbf{q}_j, \mathbf{k}_i) \in \mathbb{R}$$

  These fit into a matrix $\alpha(Q, K) \in \mathbb{R}^{r \times n}$.

- Apply softmax to each row of $\alpha(Q, K)$, so that the j-th row becomes
$$\mathsf{SoftMax}\left(\alpha(\mathbf{q}_j, \mathbf{k}_1), \ldots, \alpha(\mathbf{q}_j, \mathbf{k}_n)\right) \in \mathbb{R}^n$$
This normalizes the attention weights for a given query to be positive and sum to one[2]. Denote the resulting matrix $\mathsf{SoftMax}_{\text{row}}(\alpha(Q, K)) \in \mathbb{R}^{r \times n}$. The entries of this matrix are often called "attention scores".

- Finally, multiply by the matrix of values to obtain the r by d matrix whose j-th row is given by:
$$\frac{\sum_{i=1}^n e^{\alpha(\mathbf{q}_j, \mathbf{k}_i)} \mathbf{v}_i}{\sum_{s=1}^n e^{\alpha(\mathbf{q}_j, \mathbf{k}_s)}} \in \mathbb{R}^d$$
In this way, the attention of a query is a weighted average of the values, where the weights are the attention scores determined by the kernel α.

The table below lists common examples of attention kernels $\alpha : \mathbb{R}^\ell \times \mathbb{R}^\ell \to \mathbb{R}$. These kernels have the property that, roughly speaking, points closer together will have large values of α compared to points further apart.

Name	$\alpha(\mathbf{q}, \mathbf{k})$
Euclidean Norm	$-\lvert\mathbf{q} - \mathbf{k}\rvert$
Square Euclidean Norm	$-\frac{1}{2}\lvert\mathbf{q} - \mathbf{k}\rvert^2$
Epanechikov	$\max(0, 1 - \lvert\mathbf{q} - \mathbf{k}\rvert)$
Box Car	1 if $\lvert\mathbf{q} - k\rvert \leq 1$, otherwise 0

We have purposely omitted the dot product kernel from the list above, as it is closely related to the square Euclidean norm kernel. To see this, fix a query $\mathbf{q}$ and keys K. Then $\alpha(\mathbf{q}, \mathbf{k}_i)$ expands as:
$$\alpha(\mathbf{q}, \mathbf{k}_i) = -\frac{1}{2}|\mathbf{q} - \mathbf{k}_i|^2 = \mathbf{q}^T\mathbf{k}_i - \frac{1}{2}|\mathbf{k}_i|^2 - \frac{1}{2}|\mathbf{q}|^2$$
The last term disappears in softmax, so we ignore it. Batch normalization can be used to ensure that all keys have approximately the same norm, so that the second term can be treated as a constant and also disappears in softmax. We are left with the dot product $\mathbf{q}^T\mathbf{k}_i$. Hence, even though the dot product kernel does not have the property that points close together have large values compared to point far apart, it is a sufficient approximation to the square Euclidean norm. It is also computationally simple. Finally, as noted above, if the entries of the input vectors are independent and identically distributed with mean zero and unit variance, then dividing the dot product by $\sqrt{\ell}$ ensures that the output also has mean zero and unit variance.

[2]This defines a probability distribution on the set of values $\{\mathbf{v}_i\}_{i=1}^n$; in the next step we take the expected value of this distribution.

One can easily adopt the definitions of self-attention and multihead attention to accommodate for different attention kernels. In terms of equivariances, permutation symmetries are valid for any attention kernel, as are linear symmetries of the value space $\mathbb{R}^d$; more precisely:

$$\mathsf{Att}_\alpha(\tau Q\ ,\ \sigma K\ ,\ \sigma V h) = \tau \mathsf{Att}_\alpha(Q\ ,\ K\ ,\ V)h$$

for any $\tau \in S_r$, $\sigma \in S_n$, and $h \in \mathrm{GL}_d(\mathbb{R})$. If G is a group acting on $\mathbb{R}^\ell$ such that α is invariant for the diagonal action of G on $\mathbb{R}^\ell \times \mathbb{R}^\ell$, then we have an additional symmetry:

$$\mathsf{Att}_\alpha(Qg\ ,\ Kg\ ,\ V) = \mathsf{Att}_\alpha(Q\ ,\ K\ ,\ V)$$

where the appearance of g on the right indicates that we act on each row of the matrices.

Example 2.7. Recall that the Euclidean norm is invariant under orthogonal transformations, so that:

$$|g\mathbf{q} - g\mathbf{k}| = |g(\mathbf{q} - \mathbf{k})| = |\mathbf{q} - \mathbf{k}|$$

It follows that all of the examples in Table 2.7 have orthogonal symmetries: if $g \in O(\ell) \subseteq \mathrm{GL}_\ell(\mathbb{R})$, then $\alpha(g\mathbf{q}, g\mathbf{k}) = \alpha(\mathbf{q}, \mathbf{k})$ for any $\mathbf{q}, \mathbf{k} \in \mathbb{R}^\ell$.

3. Positional encodings

In this section we return to the usual dot-product attention of Definition 2.1. We have seen in Section 2.3 that the attention function is invariant with respect to permutations of the keys and queries. In the case of sequential data, it is important to break this symmetry so that the order of the inputs is retained. The most common way to do this is via sequential positional encodings, which we add a particular embedding matrix P to our data X before feeding it into a self-attention map. We begin with the following definition:

Definition 3.1. A *sequential positional encoding* of dimension d consists of a monoid homomorphism:

$$\phi : \mathbb{N} \to \mathbb{R}^{d \times d}$$

An *evaluation point* for ϕ (if it exists) is a vector $\mathbf{x} \in \mathbb{R}^d$ such that the map $j \mapsto \phi(j)\mathbf{x}$ is injective as a map $\mathbb{N} \to \mathbb{R}^d$. The corresponding *embedding matrix* for sequences of length n is the n by d matrix whose j-th row is $\phi(j-1)\mathbf{x} \in \mathbb{R}^d$, that is:

$$P = P_{\phi,\mathbf{x}} = \begin{bmatrix} (\phi(0)\mathbf{x})^T \\ \vdots \\ (\phi(n-1)\mathbf{x})^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

Finally, for $d' \leq d$, the *additive positional encoding map* relative to ϕ and $\mathbf{x}$ is:

$$\mathbb{R}^{n \times d'} \to \mathbb{R}^{n \times d}, \qquad X \mapsto X + \tau_{d'}(P_{\phi,x})$$

where $\tau_{d'}(P_{\phi,\mathbf{x}}) \in \mathbb{R}^{n \times d'}$ is the matrix formed by extracting the first d' columns of $P_{\phi,\mathbf{x}}$ and discarding the remaining $d - d'$ columns.

Remark 3.2. As a monoid homomorphism, the map ϕ is determined by $M := \phi(1)$; indeed, $\phi(j)$ is equal to the j-fold product M^j.

Remark 3.3. The matrix $P_{\phi,\mathbf{x}}$ is the image of the tensor $\sum_{j=1}^{n} e_j \otimes \phi(j-1) \in \mathbb{R}^n \otimes \mathbb{R}^{d\times d}$ under the map to $\mathbb{R}^n \otimes \mathbb{R}^d$ defined by the identity on the first tensor factor and multiplication by $\mathbf{x}$ in the second.

Remark 3.4. We regard the input $X \in \mathbb{R}^{n\times d}$ of the additive positional encoding map as a data matrix consisting of n samples, each of dimension d'.

Example 3.5. Suppose $d = 2$. Let M be the 2-dimensional rotation matrix with angle -1:

$$M = \begin{bmatrix} \cos(-1) & -\sin(-1) \\ \sin(-1) & \cos(1) \end{bmatrix} = \begin{bmatrix} \cos(1) & \sin(1) \\ -\sin(1) & \cos(1) \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

This matrix, which is orthogonal, defines a monoid homomorphism[3]:

$$\phi : \mathbb{N} \to \mathbb{R}^{2\times 2}, \qquad j \mapsto M^j = \begin{bmatrix} \cos(j) & \sin(j) \\ -\sin(j) & \cos(j) \end{bmatrix}$$

Take $\mathbf{x} = (0,1) \in \mathbb{R}^2$. The map $j \mapsto M^j\mathbf{x} = (\sin(j), \cos(j))$ is injective; the proof of comes down to the fact that the cosine and sine functions have irrational period 2π. Hence, $\mathbf{x}$ is an evaluation vector, and the corresponding embedding matrix is:

$$P = \begin{bmatrix} 0 & 1 \\ \sin(1) & \cos(1) \\ \sin(2) & \cos(2) \\ & \vdots \\ \sin(n-1) & \cos(n-1) \end{bmatrix} \in \mathbb{R}^{n\times 2}$$

Example 3.6. We modify and generalize the previous example to the case where d is any even positive integer. Let $\theta \in \mathbb{R}$ be a rotation angle that is not a rational multiple of π. Set M to the d by d block diagonal matrix consisting of $d/2$ blocks, the k-th of which is the 2-dimensional rotation matrix with angle $-\theta^{k-1}$. Explicitly:

$$M = \begin{bmatrix} \cos(1) & \sin(1) & 0 & 0 & \dots & 0 & 0 \\ -\sin(1) & \cos(1) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos(\theta) & \sin(\theta) & \dots & 0 & 0 \\ 0 & 0 & -\sin(\theta) & \cos(\theta) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos\left(\theta^{d/2-1}\right) & \sin\left(\theta^{d/2-1}\right) \\ 0 & 0 & 0 & 0 & \dots & -\sin\left(\theta^{d/2-1}\right) & \cos\left(\theta^{d/2-1}\right) \end{bmatrix}$$

where we use the fact that cosine is an even function and sine is an odd function to remove the minus signs from within the functions. There is a resulting monoid homomorphism $\phi : \mathbb{N} \to \mathbb{R}^{d\times d}$ with $j \mapsto M^j$. The matrix M^j is a d by d block diagonal matrix consisting of $d/2$ blocks; now the k-th block is the 2-dimensional rotation matrix with angle $-j\theta^{k-1}$.

[3]In complex coordinates, ϕ is given by $j \mapsto e^{-ij} = \cos(j) - i\sin(j) \in \mathbb{C}^\times$, whose image lies in the unit circle S^1. One recovers the original ϕ using the inclusion $S^1 \simeq \mathrm{SO}(2) \hookrightarrow \mathbb{R}^{2\times 2}$. The real embedding matrix can be obtained from the complex one by taking the negative imaginary parts in the first column and the real parts in second column.

Set $\mathbf{x} = (0, 1, 0, 1, \ldots, 0, 1) \in \mathbb{R}^d$. Since θ is not a rational multiple of π, the map $\mathbb{N} \to \mathbb{R}^d$ defined by $j \mapsto M^j\mathbf{x}$ is injective. Hence, $\mathbf{x}$ is an evaluation vector, and the corresponding embedding matrix is:

$$P = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & \ldots \\ \sin(1) & \cos(1) & \sin(\theta) & \cos(\theta^2) & -\sin(\theta^3) & \ldots \\ \sin(2) & \cos(2) & \sin(2\theta) & \cos(2\theta^2) & -\sin(2\theta^3) & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ \sin((n-1)) & \cos((n-1)) & \sin((n-1)\theta) & \cos((n-1)\theta^2) & \sin((n-1)\theta^3) & \ldots \end{bmatrix}$$

with the last two columns given by:

$$P = \begin{bmatrix} \ldots & 0 & 1 \\ \ldots & \sin(\theta^{d/2-1}) & \cos(\theta^{d/2-1}) \\ \ldots & \sin(2\theta^{d/2-1}) & \cos(2\theta^{d/2-1}) \\ & \vdots & \vdots \\ \ldots & \sin((n-1)\theta^{d/2-1}) & \cos((n-1)\theta^{d/2-1}) \end{bmatrix} \in \mathbb{R}^{n \times d}$$

In the literature, it is common to take $\theta = \frac{1}{10000^{2/d}}$, so that $\theta^{d/2} = \frac{1}{10000}$.

Suppose we have a data matrix X of size $n \times d'$. If d' is even, we set $d = d'$ and form the additive positional encoding by adding the matrix P to X. On the other hand, if d' is odd, we set $d = d' + 1$ and form the additive positional encoding by adding to X the matrix P' formed by removing the last column of P.

Finally, we record the definition of a multiplicative positional encoding.

Definition 3.7. Given a sequential positional encoding ϕ as above, the *multiplicative positional encoding map* is:

$$\mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d} \qquad X \mapsto \begin{bmatrix} (\phi(0)\mathbf{x}_1)^T \\ \vdots \\ (\phi(n-1)\mathbf{x}_n)^T \end{bmatrix} \in \mathbb{R}^{n \times d}$$

where $\mathbf{x}_j^T$ is the j-th row of X, for $j = 1, \ldots, n$ (so that $\mathbf{x}_j \in \mathbb{R}^d$ is a column vector).

References

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, *Attention is All you Need*, Advances in Neural Information Processing Systems, 2017.

[ZLLS23] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, *Dive into Deep Learning*, Cambridge University Press, 2023.