



Positional Encodings

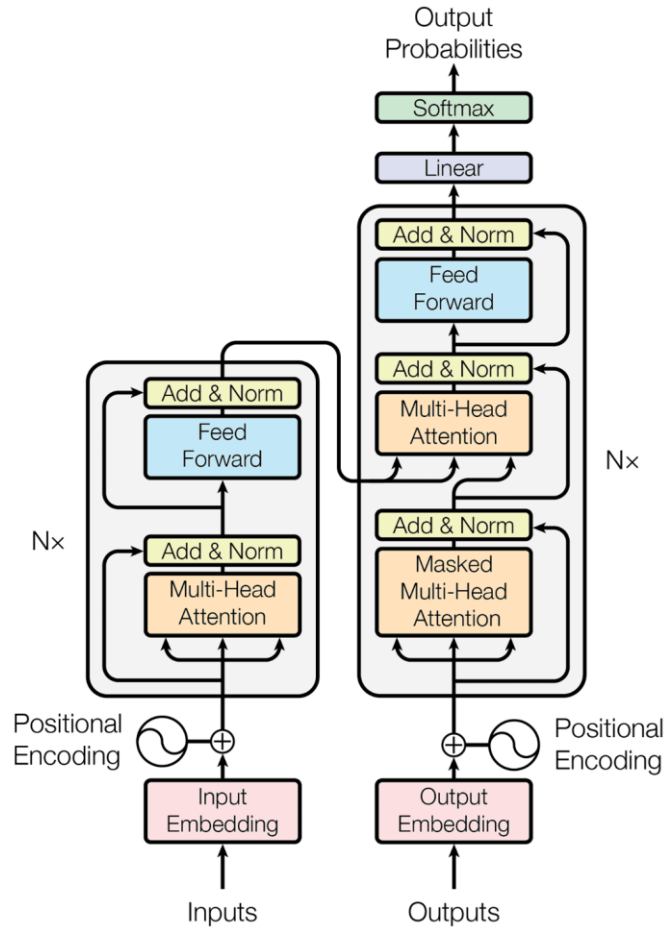
Sudipto Ghosh

(Based on Slides of Tanmoy Chakraborty, Blogs by Amirhossein Kazemnejad, Abhinav Kumar et al, and Videos by Batool Arhamna Haider, Bai Li)

Agenda

- Motivation
- Absolute PE
- Relative PE
- RoPE

Recap: Transformer Architecture



Motivation

Order Matters or Not?

Self-attention is permutation invariant!

- Consider permutations of **the sun rises in the east**:

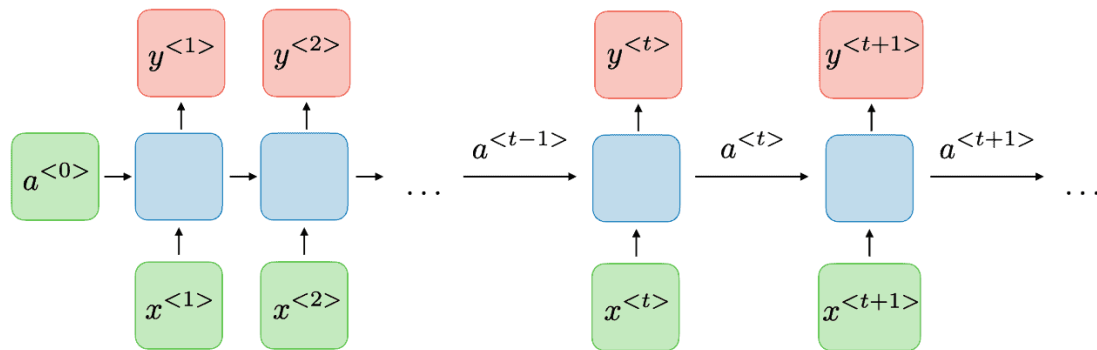
- rises in the sun the east
- the east rises in the sun

In natural language, it is important to take into account the order of words in a sentence.

- Seq2Seq Models like RNN are inherently sequential.
- Transformers process all tokens in parallel (no order).
Similar to Bag-of-Words. Sequential information is lost.

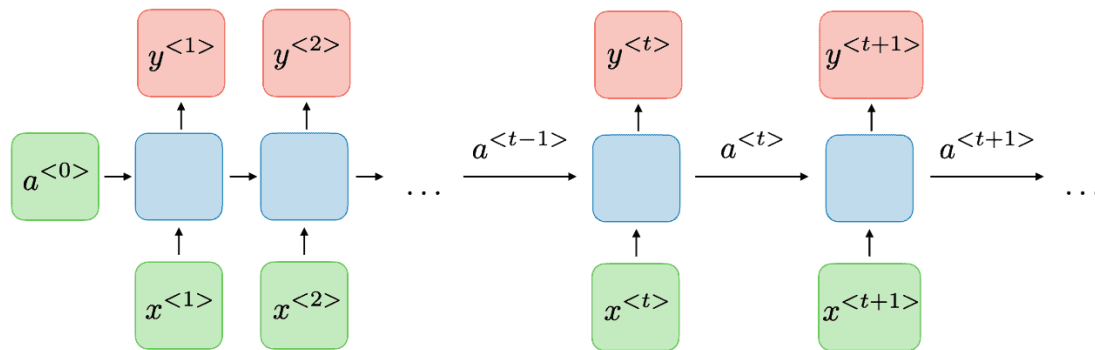


Order Matters or Not?



- Seq2Seq Models like RNN are inherently sequential.
 - Transformers process all tokens in parallel (no order). Similar to Bag-of-Words. Sequential information is lost.
-

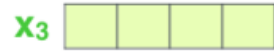
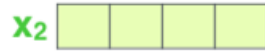
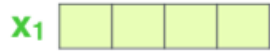
Order Matters or Not?



SOLUTION?

Explicitly add positional information to indicate where a word appears in a sequence.

EMBEDDING
WITH TIME
SIGNAL

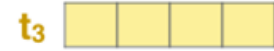


=

=

=

POSITIONAL
ENCODING

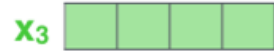
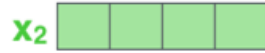
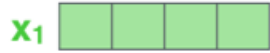


+

+

+

EMBEDDINGS



INPUT

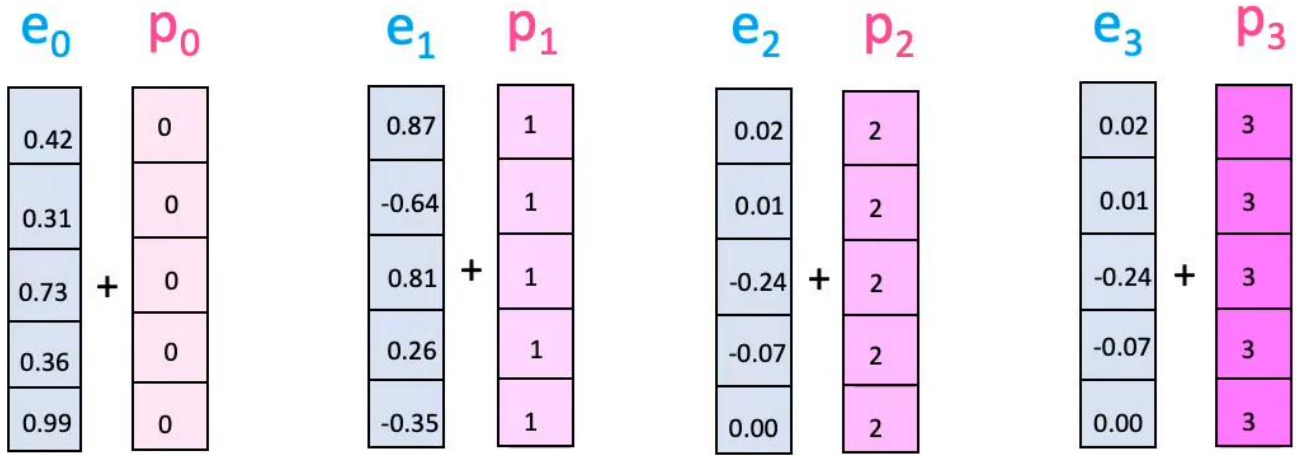
Je

suis

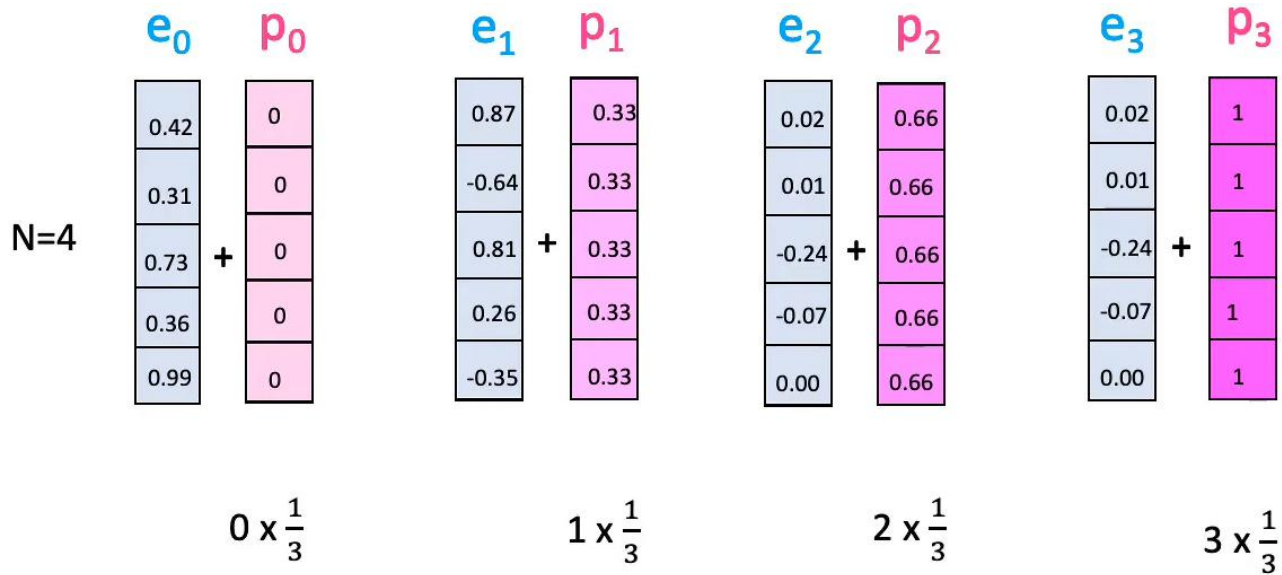
étudiant

Positional Encoding

e_0	p_0	e_1	p_2	e_2	p_1	e_3	p_3
0.42	?	0.87	?	0.02	?	0.02	?
0.31	?	-0.64	?	0.01	?	0.01	?
0.73	?	0.81	?	-0.24	?	-0.24	?
0.36	?	0.26	?	-0.07	?	-0.07	?
0.99	?	-0.35	?	0.00	?	0.00	?



Option 1



$$\frac{1}{N-1} = \frac{1}{3}$$

Option 2

Think About It

- We could just concatenate a fixed value to each time step that corresponds to its position, but then what happens if we get a sequence with 5000 words at test time?
 - We want something that can generalize to arbitrary sequence lengths. We also may want to make attending to relative positions easier.
 - Distance between two positions should be consistent with variable-length inputs.
-

Positional Encodings

General Properties of PEs

Define a function $\phi(\cdot, \cdot)$ to measure the proximity between PEs.

Monotonicity $m > n \iff \phi(\vec{x}, \overrightarrow{x+m}) < \phi(\vec{x}, \overrightarrow{x+n})$

The proximity of position embeddings decreases when positions are further apart.

Translation Invariance $\phi(\vec{x}_1, \overrightarrow{x_1+m}) = \phi(\vec{x}_2, \overrightarrow{x_2+m})$
 $= \dots = \phi(\vec{x}_n, \overrightarrow{x_n+m})$

The proximity of embedded positions are translation invariant.

Symmetry $\phi(\vec{x}, \vec{y}) = \phi(\vec{y}, \vec{x})$

The proximity of embedded positions is symmetric.

Absolute Positional Encodings

- Direct addition of positional encoding into the embedding vector.

$$\mathbb{E}_N = \{\mathbf{x}_i\}_{i=1}^N$$

$$f_{t:t \in \{q,k,v\}}(\mathbf{x}_i, i) := \mathbf{W}_{t:t \in \{q,k,v\}}(\mathbf{x}_i + \mathbf{p}_i)$$

$$\mathbf{q}_m = f_q(\mathbf{x}_m, m)$$

$$\mathbf{k}_n = f_k(\mathbf{x}_n, n)$$

$$\mathbf{v}_n = f_v(\mathbf{x}_n, n),$$

$$a_{m,n} = \frac{\exp\left(\frac{\mathbf{q}_m^\top \mathbf{k}_n}{\sqrt{d}}\right)}{\sum_{j=1}^N \exp\left(\frac{\mathbf{q}_m^\top \mathbf{k}_j}{\sqrt{d}}\right)}$$

$$\mathbf{o}_m = \sum_{n=1}^N a_{m,n} \mathbf{v}_n$$

0 :	0	0	0	0	8 :	1	0	0	0
1 :	0	0	0	1	9 :	1	0	0	1
2 :	0	0	1	0	10 :	1	0	1	0
3 :	0	0	1	1	11 :	1	0	1	1
4 :	0	1	0	0	12 :	1	1	0	0
5 :	0	1	0	1	13 :	1	1	0	1
6 :	0	1	1	0	14 :	1	1	1	0
7 :	0	1	1	1	15 :	1	1	1	1

Intuition with Bit Flipping

Absolute Positional Encodings

- Why sin, cos? - Strategic
$$\begin{cases} p_{t,2i} = \sin\left(t/10000^{2i/d}\right) \\ p_{t,2i+1} = \cos\left(t/10000^{2i/d}\right) \end{cases}$$

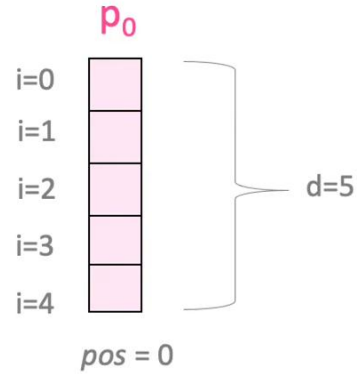
$$p_t \cdot p_{t+\phi} = \begin{bmatrix} \sin(tw_k) \\ \cos(tw_k) \end{bmatrix} \cdot \begin{bmatrix} \sin((t+\phi)w_k) \\ \cos((t+\phi)w_k) \end{bmatrix}$$

$$= \cos((t+\phi)w_k) \cos(tw_k) + \sin((t+\phi)w_k) \sin(tw_k) = \boxed{\cos(\phi w_k)}$$

CHALLENGED!

- Sinusoidal PE would enable the model to learn relative positions.
 - Might also allow it to extrapolate to sequence lengths longer than those seen during training
-

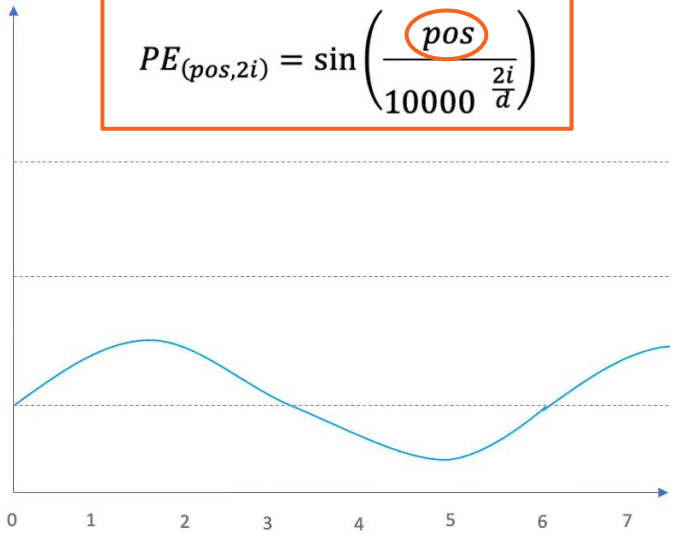
$t = pos$



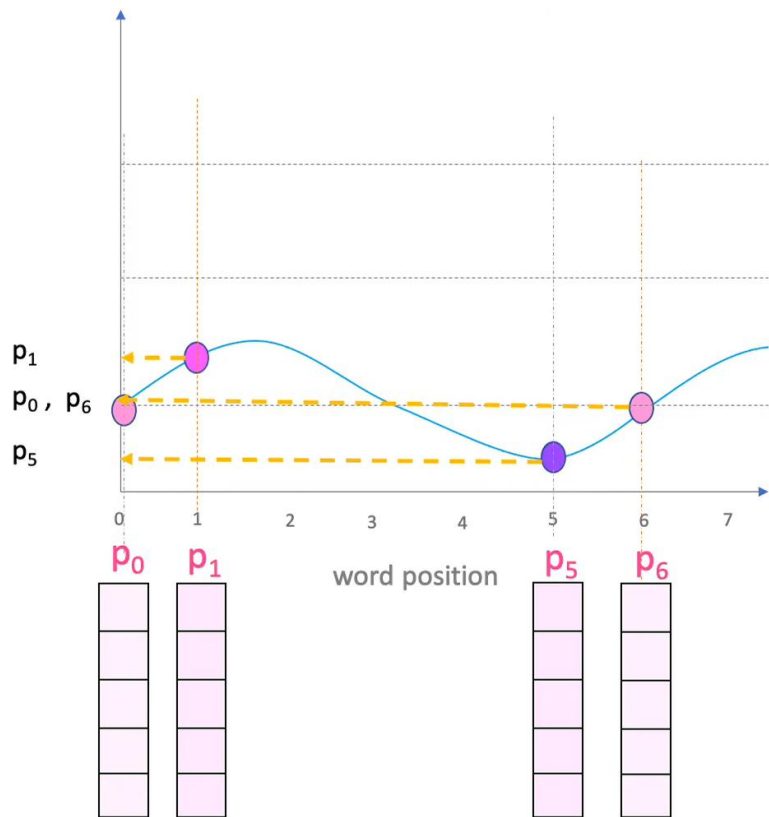
$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

Sinusoidal Positional Encoding

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000} \frac{2i}{d}\right)$$



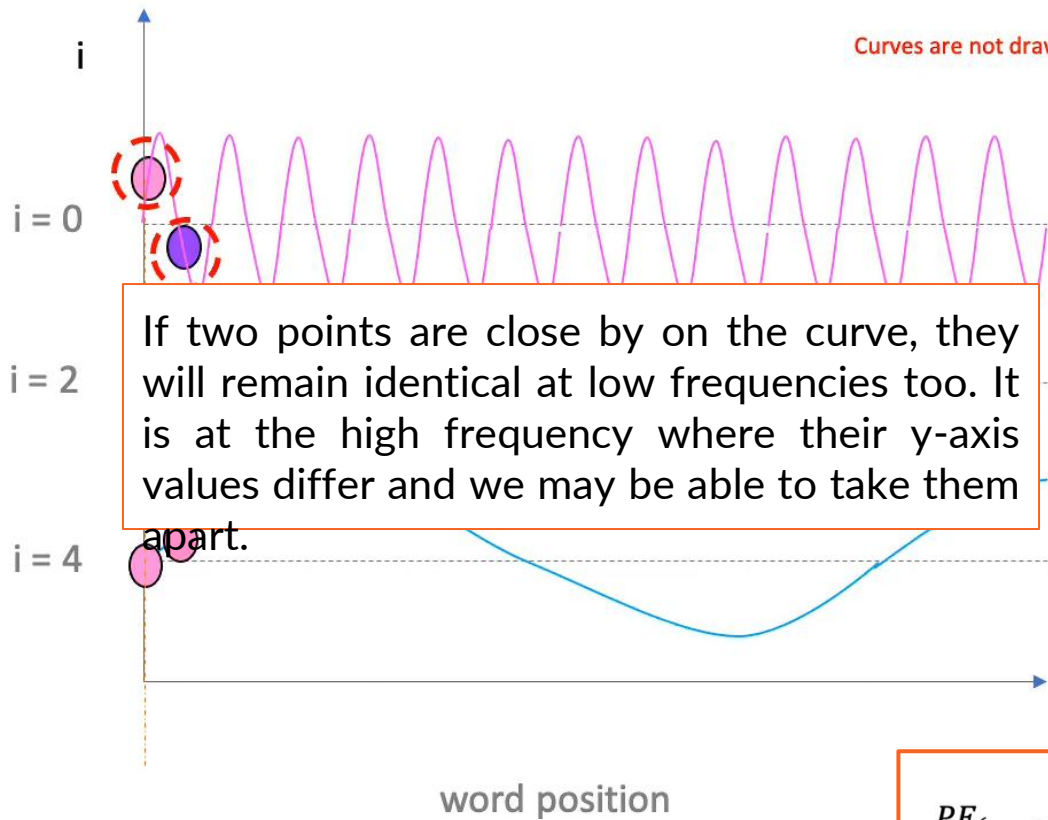
Only Varying The Position



Pros It is independent of the length of the sequence.

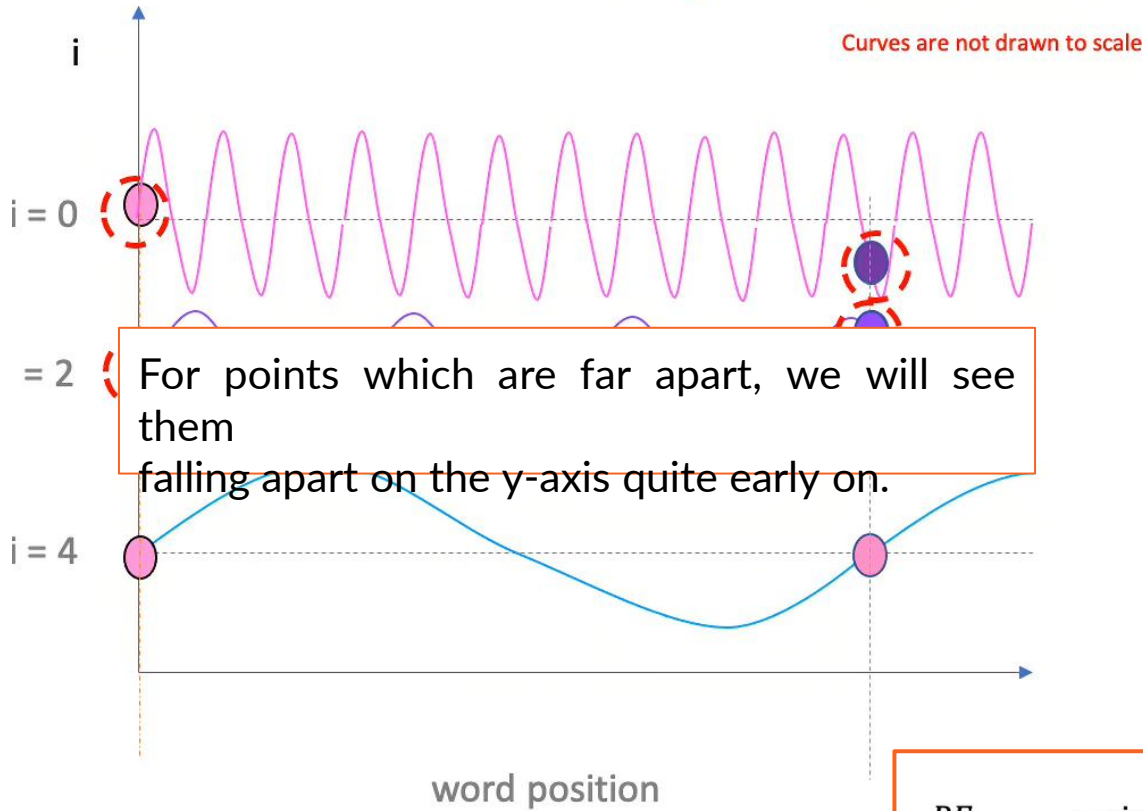
Cons p_0 and p_6 have same positional embeddings

Only Varying The Position



$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000} \frac{2i}{d}\right)$$

Varying Both pos and i



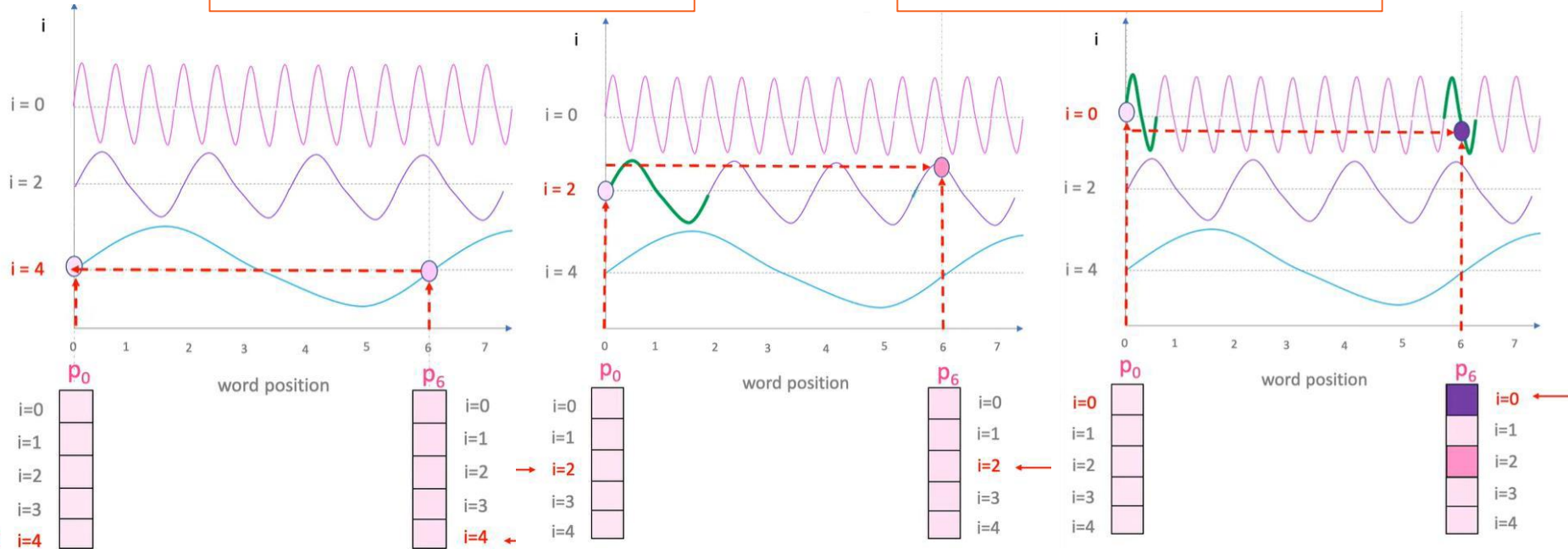
For points which are far apart, we will see them falling apart on the y-axis quite early on.

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000} \frac{2i}{d}\right)$$

Varying Both pos and i

At lower frequency, the value is exactly the same.

This changes as we move to higher frequencies.



Varying Both pos and i - Example

For example, for word w at position $pos \in [0, L - 1]$ in the input sequence $\mathbf{w} = (w_0, \dots, w_{L-1})$, with 4-dimensional embedding e_w , and $d_{model} = 4$, the operation would be

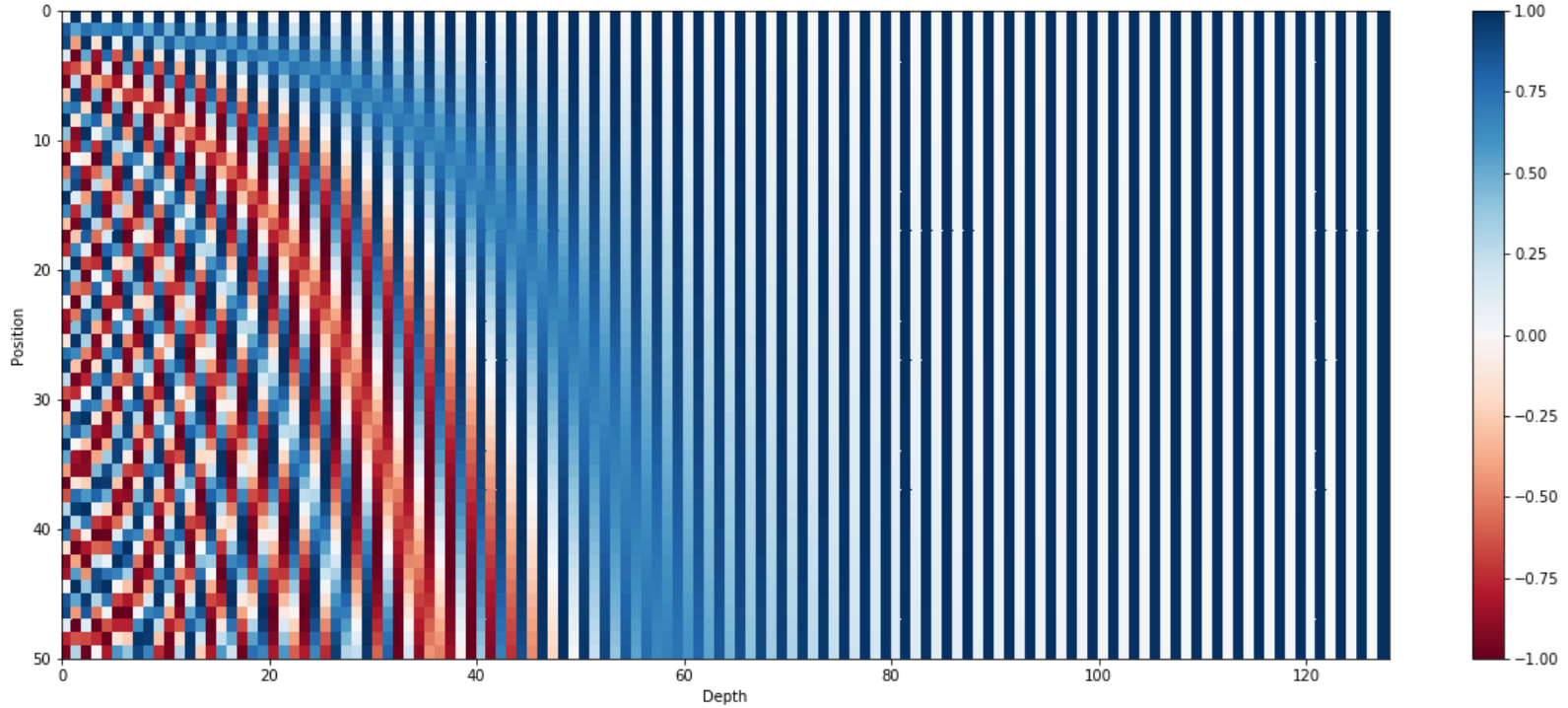
$$\begin{aligned} e'_w &= e_w + \left[\sin \left(\frac{pos}{10000^0} \right), \cos \left(\frac{pos}{10000^0} \right), \sin \left(\frac{pos}{10000^{2/4}} \right), \cos \left(\frac{pos}{10000^{2/4}} \right) \right] \\ &= e_w + \left[\sin(pos), \cos(pos), \sin \left(\frac{pos}{100} \right), \cos \left(\frac{pos}{100} \right) \right] \end{aligned}$$

where the formula for positional encoding is as follows

$$\text{PE}(pos, 2i) = \sin \left(\frac{pos}{10000^{2i/d_{model}}} \right),$$

$$\text{PE}(pos, 2i + 1) = \cos \left(\frac{pos}{10000^{2i/d_{model}}} \right).$$

with $d_{model} = 512$ (thus $i \in [0, 255]$) in the original paper.



What does this look like?

DEMO

Relative Positional Encodings

The pig chased the dog

Once upon a time, the pig
chased the dog

Relative Positional Encodings

Vasvani et al (2017)

We chose this (sinusoidal) function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k , PE_{pos+k} can be represented as a linear function of PE_{pos}

$$r = clip(m - n, Rmin, Rmax)$$

$$\mathbf{q}_m = x_m W_Q \quad \mathbf{k}'_n = (x_n + \tilde{p}_r^k) W_K \quad \mathbf{v}'_n = (x_n + \tilde{p}_r^v) W_V$$

Relative position info is not useful beyond a certain distance.

- Unlike absolute PEs, which create embeddings for each position independently, relative PEs focus on capturing the **pairwise relationships** between tokens in a sequence.
 - Rather than directly adding the embeddings to the context vectors, the **relativeness** is added to keys and values during the attention calculation.
-

—

For every sine-cosine pair corresponding to frequency ω_k , there is a linear transformation $M \in \mathbb{R}^{2 \times 2}$ (independent of t) where the following equation holds:

$$M \cdot \begin{bmatrix} \sin(\omega_k \cdot t) \\ \cos(\omega_k \cdot t) \end{bmatrix} = \begin{bmatrix} \sin(\omega_k \cdot (t + \phi)) \\ \cos(\omega_k \cdot (t + \phi)) \end{bmatrix} \longrightarrow$$

NOTE

ϕ denotes an offset here, not the proximity fn as defined before

Proof:

Let M be a 2×2 matrix, we want to find u_1, v_1, u_2 and v_2 so that:

$$\begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix} \cdot \begin{bmatrix} \sin(\omega_k \cdot t) \\ \cos(\omega_k \cdot t) \end{bmatrix} = \begin{bmatrix} \sin(\omega_k \cdot (t + \phi)) \\ \cos(\omega_k \cdot (t + \phi)) \end{bmatrix}$$

By applying the addition theorem, we can expand the right hand side as follows:

$$\begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix} \cdot \begin{bmatrix} \sin(\omega_k \cdot t) \\ \cos(\omega_k \cdot t) \end{bmatrix} = \begin{bmatrix} \sin(\omega_k \cdot t) \cos(\omega_k \cdot \phi) + \cos(\omega_k \cdot t) \sin(\omega_k \cdot \phi) \\ \cos(\omega_k \cdot t) \cos(\omega_k \cdot \phi) - \sin(\omega_k \cdot t) \sin(\omega_k \cdot \phi) \end{bmatrix}$$

Which result in the following two equations:

$$u_1 \sin(\omega_k \cdot t) + v_1 \cos(\omega_k \cdot t) = \cos(\omega_k \cdot \phi) \sin(\omega_k \cdot t) + \sin(\omega_k \cdot \phi) \cos(\omega_k \cdot t) \quad (1)$$

$$u_2 \sin(\omega_k \cdot t) + v_2 \cos(\omega_k \cdot t) = -\sin(\omega_k \cdot \phi) \sin(\omega_k \cdot t) + \cos(\omega_k \cdot \phi) \cos(\omega_k \cdot t) \quad (2)$$

By solving above equations, we get:

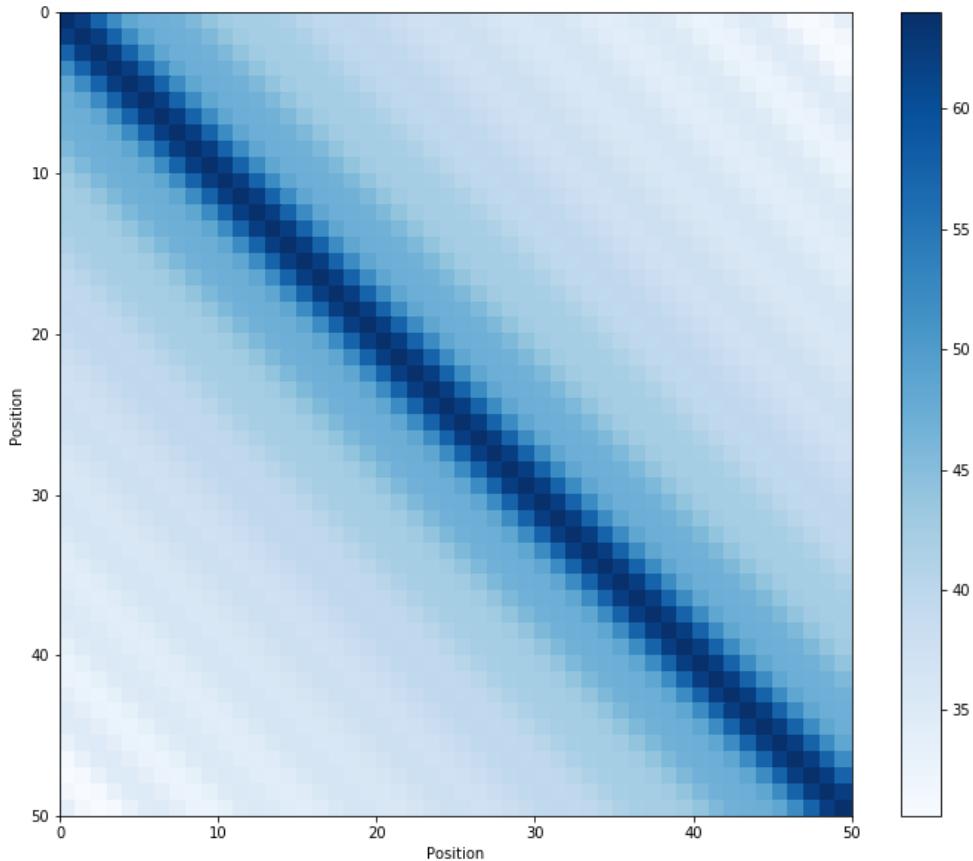
$$\begin{aligned}u_1 &= \cos(\omega_k \cdot \phi) & v_1 &= \sin(\omega_k \cdot \phi) \\u_2 &= -\sin(\omega_k \cdot \phi) & v_2 &= \cos(\omega_k \cdot \phi)\end{aligned}$$

So the final transformation matrix M is:

$$M_{\phi,k} = \begin{bmatrix} \cos(\omega_k \cdot \phi) & \sin(\omega_k \cdot \phi) \\ -\sin(\omega_k \cdot \phi) & \cos(\omega_k \cdot \phi) \end{bmatrix}$$

Looks Like
Rotation Matrix

Sinusoidal PE is Symmetric



$$\mathbf{q}_m^\top \mathbf{k}_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + b_{i,j}$$

Relative Positional Encodings

T5

$$f_{t:t \in \{q,k,v\}}(\mathbf{x}_i, i) := \mathbf{W}_{t:t \in \{q,k,v\}}(\mathbf{x}_i + \mathbf{p}_i)$$

$$\mathbf{q}_m^\top \mathbf{k}_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{p}_n + \mathbf{p}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{p}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{p}_n$$

\mathbf{W}_k is distinguished for content-based and location-based key vectors \mathbf{v}_5 and \mathbf{u}_4 denoted as \mathbf{W}_k and $\tilde{\mathbf{W}}_k$

- Replace abs pos embedding \mathbf{v}_5 with relative $\tilde{\mathbf{v}}_5$
- Replace abs pos embedding \mathbf{u}_4 with two trainable vectors = and $\tilde{\mathbf{u}}_4$ independent of query positions.

\mathbf{q}_k

$$\mathbf{q}_m^\top \mathbf{k}_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{x}_m^\top \mathbf{W}_q^\top \tilde{\mathbf{W}}_k \tilde{\mathbf{p}}_{m-n} + \mathbf{u}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{v}^\top \mathbf{W}_q^\top \tilde{\mathbf{W}}_k \tilde{\mathbf{p}}_{m-n}$$

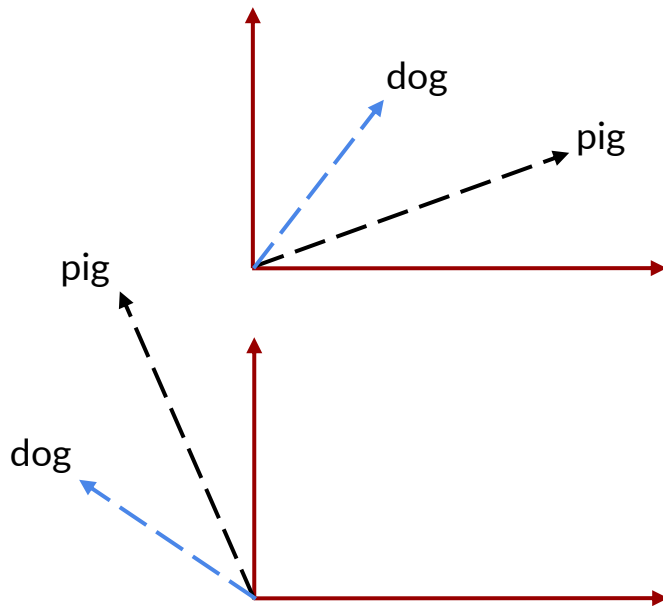
DeBERTa

$$\mathbf{q}_m^\top \mathbf{k}_n = \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n + \mathbf{x}_m^\top \mathbf{W}_q^\top \mathbf{W}_k \tilde{\mathbf{p}}_{m-n} + \tilde{\mathbf{p}}_{m-n}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{x}_n$$

Combining Abs and Rel Positions

The pig chased the dog

Once upon a time, the pig
chased the dog



Rotary Position Embedding RoPE

Rotate the affine-transformed word embedding vector by amount of angle multiples of its position

$$f_{\{q,k\}}(\mathbf{x}_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} W_{\{q,k\}}^{(11)} & W_{\{q,k\}}^{(12)} \\ W_{\{q,k\}}^{(21)} & W_{\{q,k\}}^{(22)} \end{pmatrix} \begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix}$$

$$\langle f_q(\mathbf{x}_m, m), f_k(\mathbf{x}_n, n) \rangle = g(\mathbf{x}_m, \mathbf{x}_n, m - n)$$

Euler's Formula

$$e^{im\theta} = \cos(m\theta) + i \sin(m\theta)$$

$$\begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix}$$

$$f_q(\mathbf{x}_m, m) = (\mathbf{W}_q \mathbf{x}_m) e^{im\theta}$$

$$f_k(\mathbf{x}_n, n) = (\mathbf{W}_k \mathbf{x}_n) e^{in\theta}$$

$$g(\mathbf{x}_m, \mathbf{x}_n, m - n) = \text{Re}[(\mathbf{W}_q \mathbf{x}_m)(\mathbf{W}_k \mathbf{x}_n)^* e^{i(m-n)\theta}]$$

RoPE in n-dimensions

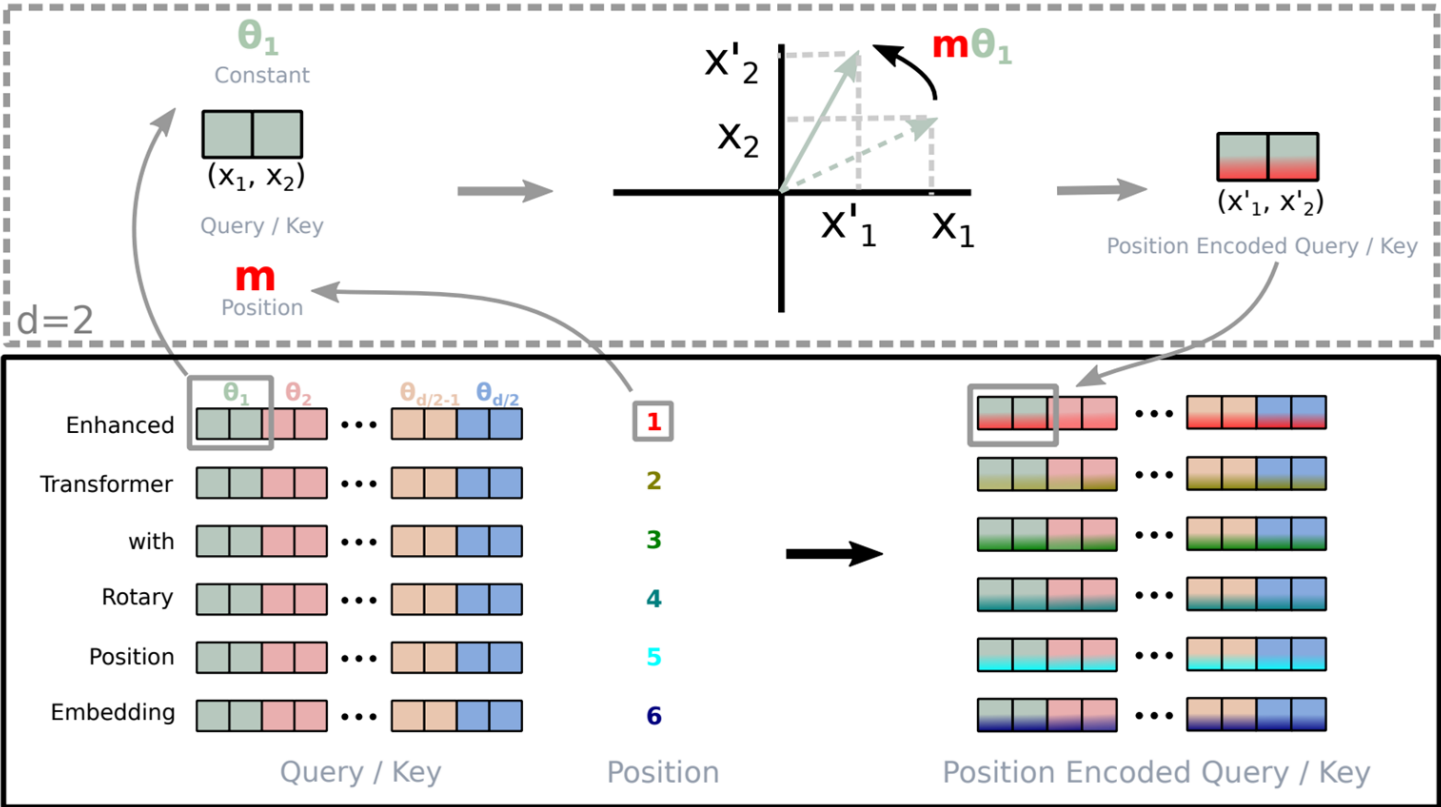
$$f_{\{q,k\}}(\mathbf{x}_m, m) = \mathbf{R}_{\Theta, m}^d \mathbf{W}_{\{q,k\}} \mathbf{x}_m$$

$$\mathbf{R}_{\Theta, m}^d = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \dots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \dots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \dots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix}$$

$$\Theta = \{\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \dots, d/2]\}$$

$$\mathbf{q}_m^\top \mathbf{k}_n = (\mathbf{R}_{\Theta, m}^d \mathbf{W}_q \mathbf{x}_m)^\top (\mathbf{R}_{\Theta, n}^d \mathbf{W}_k \mathbf{x}_n) = \mathbf{x}_m^\top \mathbf{W}_q \mathbf{R}_{\Theta, n-m}^d \mathbf{W}_k \mathbf{x}_n$$

$$\mathbf{R}_{\Theta, n-m}^d = (\mathbf{R}_{\Theta, m}^d)^\top \mathbf{R}_{\Theta, n}^d$$



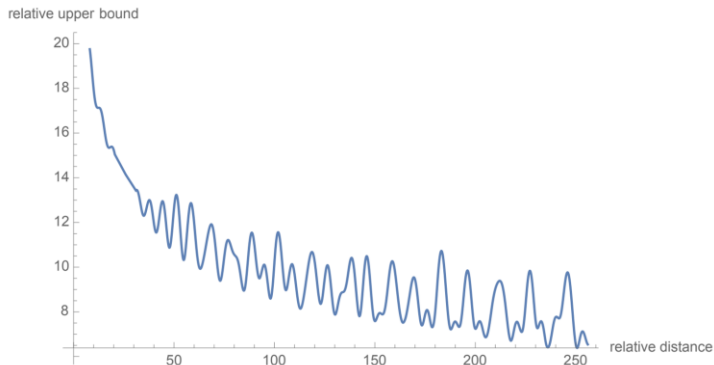
RoPE Implementation

RoPE Characteristics

- In contrast to earlier position embedding methods, RoPE is multiplicative not additive.
 - Represents token embeddings as complex numbers.
 - Represents their positions as pure rotations.
 - If we shift both the query and key by the same amount, changing absolute position but not relative position, this will lead both representations to be additionally rotated in the same manner, thus the angle between them will remain unchanged and thus the dot product will also remain unchanged.
-

RoPE Characteristics

- Long-term decay
 - Inner-product decays when the relative position increase.
 - A pair of tokens with a long relative distance should have less connection.



RoPE Characteristics

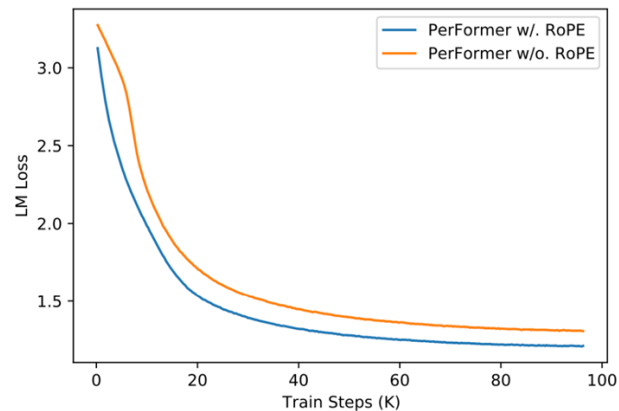
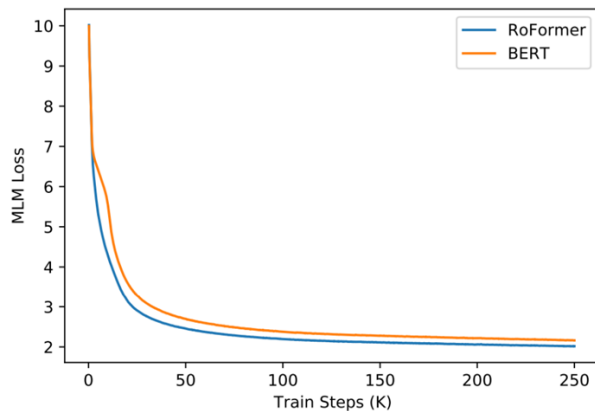
- Efficient realization of rotary matrix multiplication

$$\mathbf{R}_{\Theta, m}^d \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ x_3 \\ \vdots \\ -x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix}$$

RoPE Performance

Table 2: Comparing RoFormer and BERT by fine tuning on downstream GLEU tasks.

Model	MRPC	SST-2	QNLI	STS-B	QQP	MNLI(m/mm)
BERT Devlin et al. [2019]	88.9	93.5	90.5	85.8	71.2	84.6/83.4
RoFormer	89.5	90.7	88.0	87.0	86.4	80.2/79.8



Sources

- <https://iclr-blogposts.github.io/2025/blog/positional-embedding/>
 - https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
 - <https://www.youtube.com/watch?v=dicHlcUZfOw>
 - <https://www.youtube.com/watch?v=o29P0Kpobz0>
 - Slides from ELL881/AIL821 (Semester I, 2024-25) by Prof Tanmoy Chakraborty
 - “Attention is All you Need”
Vasvani et al (2017)
 - “RoFormer: Enhanced Transformer with Rotary Position Embedding”
Su et al (2021)
-