

Position Embeddings

Rationale

- Transformers have multiple operations that run in parallel so the order of the original sequence isn't preserved in just one operation. To overcome this limitation, we have to use another mechanism to keep track of the **positions** of the tokens in the sequence.

Position Embeddings

Method 1 - Use a Neural Network Embedding

- By leveraging information such as the length of the sequence and the dimensions associated with your token embeddings, you can build a neural network embedding that learns to assign a position embedding to each token in the sequence.

Method 2 - Use a Fixed Position Embedding

- You can use a fixed position embedding that is added to the token embedding. This is a simple and effective method that is used in many transformer models. This is done with a sinusoidal function where the frequency of the sine wave is determined by the position of the token in the sequence.

- Even dimensions

$$PE_{(pos,i)} = \sin \left(pos \cdot e^{-\frac{\log(10000)}{d_{\text{model}}}} \cdot i \right)$$

- Odd Dimensions

$$PE_{(pos,i)} = \cos \left(pos \cdot e^{-\frac{\log(10000)}{d_{\text{model}}}} \cdot (i-1) \right)$$

- $\log(10000)$ is used to make the position embeddings more stable for both short-term and long-term dependencies.