**Report on Positional Encoding in Transformers**

**Introduction**

Positional encoding is crucial in transformers since, unlike RNNs and LSTMs, transformers process words in parallel, losing the sequential order information. Positional encoding introduces a mechanism to capture the position of words in a sequence, essential for understanding context.

**Initial Approach**

The initial method added a scalar to each word to store its position. However, this approach had significant drawbacks:

1. **Unbounded Values:** There was no upper limit to the scalar values, making it impractical for long sequences.
2. **Relative Position Ignored:** The method did not capture the relative positions between words effectively.

**Sinusoidal Encoding**

To address these issues, sinusoidal functions were introduced. Specifically, the sine function sin(pos) was used. This method resolved the drawbacks of the initial approach:

1. **Bounded Values:** The sine function outputs values between -1 and 1.
2. **Relative Position:** Sinusoidal encoding captures relative positions due to its periodic nature.

**Study and Findings**

A study was conducted using positional vectors with a sequence length of 100 and a dimension of 5. Key findings were:

* As the dimensions increased, the values approached zero.
* Plotting the sine of all columns revealed that the sine waves intersected at the least common multiple (LCM) of their periods(wavelengths), causing the waves to restart periodically.

**Introducing Cosine Functions**

To further refine the encoding and prevent wave intersection:

* **Sine for Even Positions:** PE(pos,2i)​=sin(pos/100002i/d)
* **Cosine for Odd Positions:** PE(pos,2i+1)​=cos(pos/100002i/d)
* This combined approach ensured that the sine and cosine waves did not intersect, providing a unique positional encoding for each word position.

**Conclusion**

Positional encoding using sinusoidal functions effectively captures word positions in transformer models. The combined use of sine and cosine functions ensures bounded, non-intersecting values, preserving both absolute and relative positions in the sequence.  
  
**References**:https://colab.research.google.com/drive/17vfKAXEQUM7cWkDJ5TWTqu8zDkBbzrrS#scrollTo=dk7DD-TS9yWs





