**Masking:**

In TensorFlow, masking refers to the process of selectively ignoring certain values in a tensor by setting them to zero. This is often done when working with sequences, such as in natural language processing or time series analysis, where not all elements of the sequence may be relevant to the task at hand. By masking out these irrelevant values, the algorithm can focus on the important ones and improve its accuracy and efficiency. TensorFlow provides several functions for creating and applying masks, such as tf.where and tf.boolean\_mask.

**A screenshot of a computer

Description automatically generated with low confidence**

**Positional encoding:**

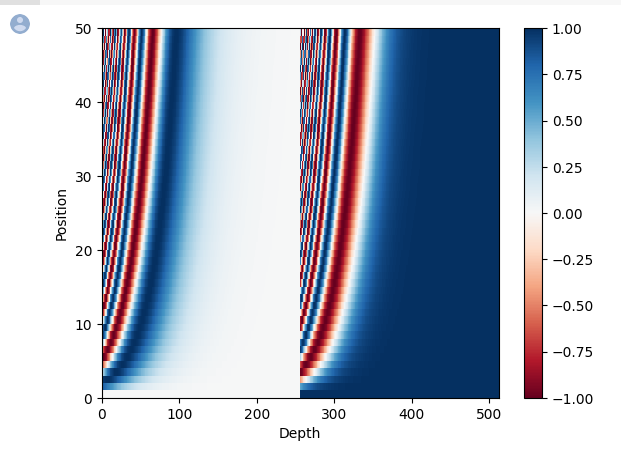
The importance of positional encoding is to provide additional information about the position of each word in a sentence. In other words, it helps the model to distinguish between the meaning of the same word used in different positions within a sentence. This is done by adding a fixed vector to each word embedding based on its position in the sequence. This helps to overcome the shortcomings of the bag-of-words model, where the order of words is not considered. Overall, positional encoding helps the model to capture both the meaning and the order of words in a sentence, resulting in more accurate predictions.

The `get\_angles` method calculates the angles for the positional encoding. It takes `position, i,` and `d\_model` as input. Here, `position` is the position in the sequence, `i` is the dimension of the embedding vector, and `d\_model` is the size of the embedding vector. It calculates the angles based on the formula `angles = 1 / tf.pow(10000, (2 \* (i // 2)) / tf.cast(d\_model, tf.float32))`.The `positional\_encoding` method calculates the positional encodings for the input. It takes `position` and `d\_model` as inputs. It calls the `get\_angles` method to get the angles and then calculates the sin and cos values for the positional encoding.

A screenshot of a computer

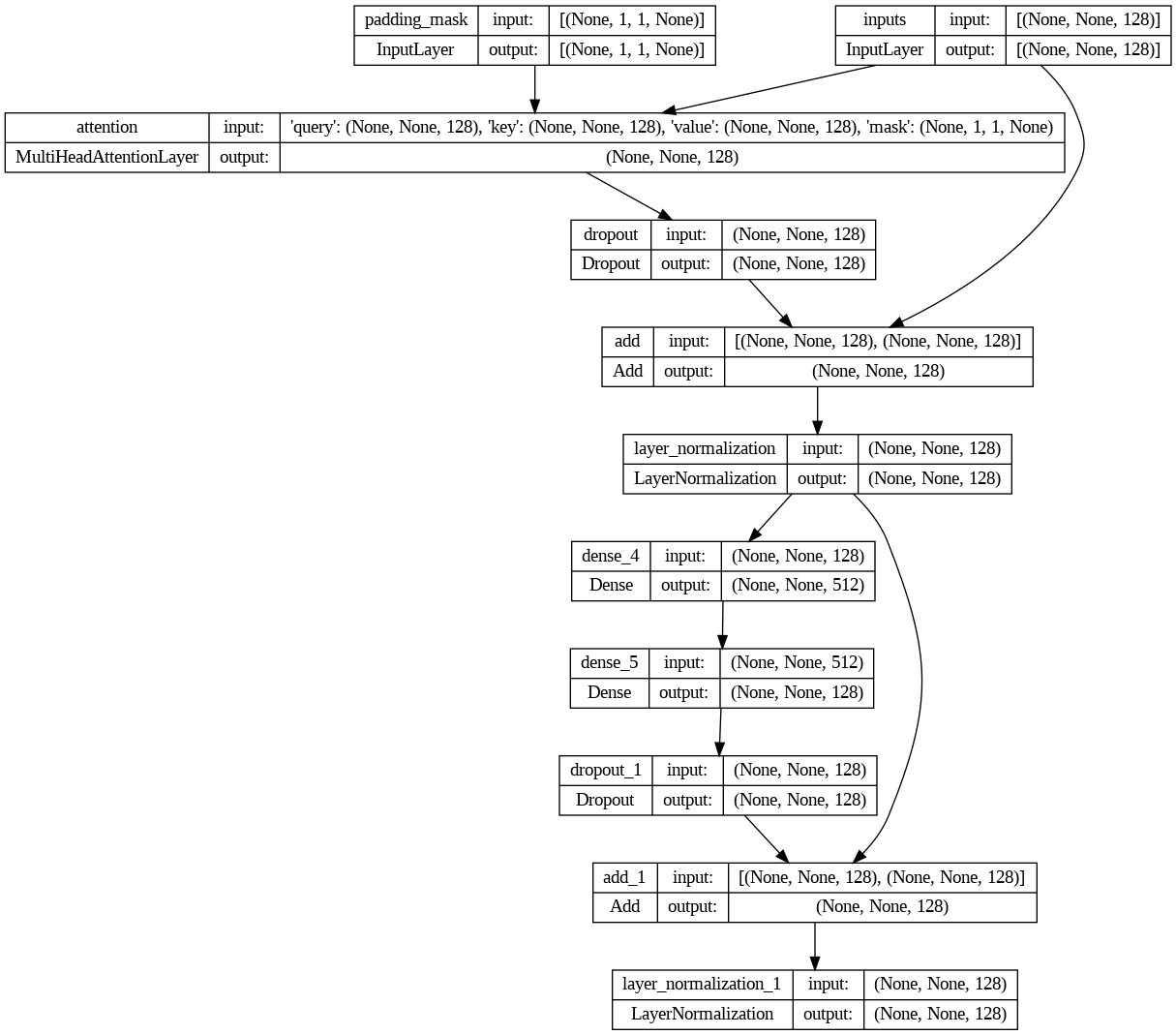
Description automatically generated with medium confidence

The graph of position encoding for the previous example in the image:

****

**Encoder layer:**

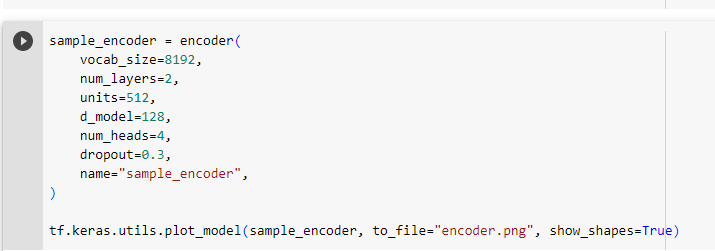
The encoder layer is a component within the encoder that performs this transformation on the input sequence. Each encoder layer typically involves multiple subcomponents, such as multi-head attention mechanisms and feedforward neural networks, that work together to compute the embeddings for the input sequence.

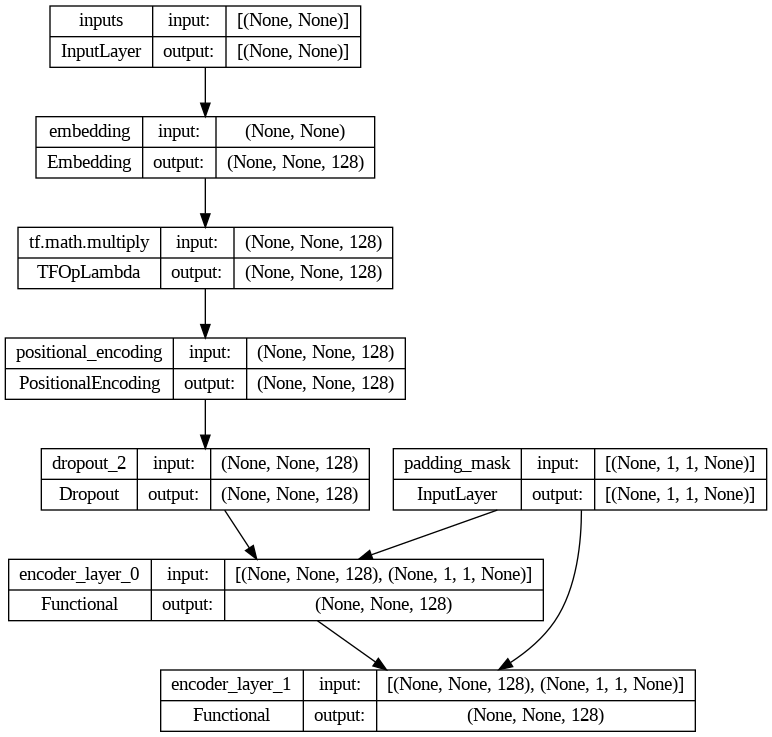


**Encoder:**

The encoder is responsible for transforming the input sequence of words into a series of dense vectors, also known as embeddings, that capture the relationships between the words in the sequence. These embeddings are then passed on to the decoder, which generates output sequences, in the case of language generation models, based on these embeddings.

And here is an example of it:





**Decoder layer:**

In short, the decoder layer is a building block that helps the decoder generate the next token in the target sequence, while the decoder is the larger component responsible for generating the entire sequence.

A picture containing text, screenshot, diagram, font

Description automatically generated

**Decoder:**

**A picture containing text, diagram, line, font

Description automatically generated**