TokenAndPositionEmbedding layer

TokenAndPositionEmbedding class

[source]

```
keras_nlp.layers.TokenAndPositionEmbedding(
    vocabulary_size,
    sequence_length,
    embedding_dim,
    embeddings_initializer="glorot_uniform",
    mask_zero=False,
    **kwargs
)
```

A layer which sums a token and position embedding.

This layer assumes that the last dimension in the input corresponds to the sequence dimension.

Arguments

- **vocabulary_size**: The size of the vocabulary.
- **sequence_length**: The maximum length of input sequence
- **embedding_dim**: The output dimension of the embedding layer
- **embeddings_initializer**: The initializer to use for the Embedding Layers
- **mask_zero**: Boolean, whether or not the input value 0 is a special "padding" value that should be masked out. This is useful when using recurrent layers which may take variable length input. If this is True, then all subsequent layers in the model need to support masking or an exception will be raised. If mask_zero` is set to True, as a consequence, index 0 cannot be used in the vocabulary (input_dim should equal size of vocabulary + 1).

Examples

```
seq_length = 50
vocab_size = 5000
embed_dim = 128
inputs = keras.Input(shape=(seq_length,))
embedding_layer = keras_nlp.layers.TokenAndPositionEmbedding(
    vocabulary_size=vocab_size,
    sequence_length=seq_length,
    embedding_dim=embed_dim,
)
outputs = embedding_layer(inputs)
```