UNetWithAttention Class Documentation

Niccolò Ferrari

July 31, 2024

1 Class UNetWithAttention

The `UNetWithAttention` class implements a U-Net architecture with attention mechanisms and configurable skip connections.

Attributes

- **input_shape** (tuple): The shape of the input image, e.g., (128, 128, 1) for grayscale images.
- **timestamp_dim** (int): The dimension of the timestamp input.
- **filter_list** (list): A list of integers representing the number of filters for each encoder/decoder block.
- **num_skip_connections** (int): The number of skip connections from the deepest layer.
- **num_heads** (int): The number of attention heads in the multi-head attention layers.
- **key_dim** (int): The dimensionality of the key space for the multi-head attention.
- **use_bias** (bool): Whether to include biases in the layers.
- **activation** (str): The activation function to use in the convolutional layers.
- **model** (Model): The Keras model instance.

Methods

Constructor: __init__

```
__init__(input_shape, timestamp_dim, filter_list, num_skip_connections, num_heads=4, key_dim
```

Initializes the UNetWithAttention class.

input_shape (tuple): The shape of the input images.

timestamp_dim (int): The dimensionality of the timestamp input.

filter_list (list): A list of filters for each encoder and decoder block.

num_skip_connections (int): The number of skip connections to include, starting from the deepest layer.

num_heads (int): The number of heads for the multi-head attention layers.

key_dim (int): The dimensionality of the key vectors for multi-head attention.

use_bias (bool): Whether to use biases in convolutional layers and attention mechanisms.

activation (str): The activation function to use in the convolutional layers. Default is 'swish'.

Private Methods

These methods are intended for internal use within the class.

`_conv_block(x, filters)` Creates a convolutional block consisting of Conv2D, BatchNormalization, and the specified activation.

`_residual_block(x, filters)` Creates a residual block with a skip connection.

`_multihead_attention_block(x)` Applies a multi-head attention mechanism followed by a residual connection and layer normalization.

`_positional_embedding(x)` Adds positional embeddings to the input tensor.

`_encoder_block(x, filters)` Creates an encoder block consisting of a residual block followed by a downsampling operation.

`_decoder_block(x, skip_features, filters)` Creates a decoder block consisting of upsampling, concatenation with skip connections, and a residual block.

Public Methods

These methods are available for external use.

`build_model()` Builds the U-Net model with attention mechanisms and stores it in the `model` attribute.

`print_model()` Prints the summary of the built model.

`save_model_plot(filename='unet_model.png')` Saves a plot of the model architecture to a file.