tensor.reduce_min

tensor.reduce_min

```

Copy fnreduce_min(self:@Tensor, axes:Option>, keepdims:Option, noop_with_empty_axes:Option)->Tensor;

```

Computes the min of the input tensor's elements along the provided axes.

Args

- self
- (@Tensor
-) - The input tensor.
- axes
- (Option>
-) - Optional input list of integers, along which to reduce. The default is to reduce over all the dimensions of the input tensor if 'noop_with_empty_axes' is false, else act as an Identity op when 'noop_with_empty_axes' is true.
- keepdims
- (Option
-) - Keep the reduced dimension or not, default true means keep reduced dimension.
- noop_with_empty_axes
- (Option
-) - Defines behavior if 'axes' is empty. Default behavior with 'false' is to reduce all axes. When axes is empty and this attribute is set to true, input tensor will not be reduced,and the output tensor would be equivalent to input tensor.
-

Panics

- Panics if axis is not in the range of the input tensor's dimensions.
-

Returns

A newTensor instance with the specified axes reduced by minimum of its elements.

Examples

```

Copy usecore::array::{ArrayTrait,SpanTrait};

useorion::operators::tensor::{TensorTrait,Tensor,U32Tensor};

fnreduce_min_example()->Tensor { lettensor=TensorTrait::::new( shape:array![2,2,2].span(), data:array![0,1,2,3,4,5,6,7].span(), );

// We can call reduce_min function as follows. returntensor.reduce_min(axes:array![1].span(), keepdims:Option::None(()), noop_with_empty_axes:Option::None(())); }

                    [[0,1],[4,5]]

```

Previous tensor.gather_nd Next tensor.shrink

Last updated3 months ago