

tensor.cumsum

...

```
Copy fncumsum(self:@Tensor, axis:usize, exclusive:Option, reverse:Option)->Tensor;
```

...

Performs cumulative sum of the input elements along the given axis.

Args

- self
- (@Tensor
-) - The input tensor.
- axis
- (usize
-) - The axis along which to compute the cumulative sum.
- exclusive
- (Option
-) - By default, it will do the sum inclusively meaning the first element is copied as is.
- reverse
- (Option
-) - If true, the cumulative sum is performed in the opposite direction. Defaults to false.
-

Panics

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

Returns

A newTensor instance containing the cumulative sum of the input tensor's elements along the given axis.

Examples

Case 1: cumsum with default parameters

...

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

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

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

```
returntensor.cumsum(axis:2, exclusive:Option::None(), reverse:Option::None()); }
```

```
[[[0,1],[2,5]],[4,9],[6,13]]
```

...

Case 2: cumsum with exclusive = true

...

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

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

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

```
returntensor.cumsum(axis:2, exclusive:Option::Some(true), reverse:Option::None()); }
```

```
[[[0,0],[0,2]],[0,4],[0,6]]
```

...

Case 3: cumsum with exclusive = true and reverse = true

...

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

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

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

```
returntensor.cumsum(axis:2, exclusive:Option::Some(true), reverse:Option::Some(true)); }
```

```
[[[1,0],[3,0]],[[5,0],[7,0]]]
```

...

[Previous tensor.ceil](#) [Next tensor.sin](#)

Last updated3 months ago