tensor.unique

```

Copy fnunique(self:@Tensor, axis:Option, sorted:Option)->(Tensor,Tensor,Tensor,Tensor);

```

Identifies the unique elements or subtensors of a tensor, with an optional axis parameter for subtensor slicing. This function returns a tuple containing the tensor of unique elements or subtensors, and optionally, tensors for indices, inverse indices, and counts of unique elements.

- axis
- (Option
-) - Specifies the dimension along which to find unique subtensors. A None value means the unique elements of the tensor will be returned in a flattened form. A negative value indicates dimension counting from the end.
- sorted
- (Option
-) - Determines if the unique elements should be returned in ascending order. Defaults to true.
-

Returns

A tuple containing:

- A Tensor with unique values or subtensors from self.
- A Tensor with the first occurrence indices of unique elements in self. If axis is given, it returns indices along that axis; otherwise, it refers to the flattened tensor.
- A Tensor mapping each element of self to its index in the unique tensor. If axis is specified, it maps to the subtensor index; otherwise, it maps to the unique flattened tensor.
- A Tensor for the counts of each unique element or subtensor in self.
-

Example

```

Copy usearray::{ArrayTrait,SpanTrait};

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

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

returntensor.unique( axis:Option::None(()) sorted:Option::Some(false) ); }

                    ( [2,1,3,4], [0,1,3,4], [0,1,1,2,3,2], [1,2,2,1] )

```

or

```

Copy usearray::{ArrayTrait,SpanTrait};

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

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

returntensor.unique( axis:Option::Some(0) sorted:Option::Some(true) ); }

                    ( [[1,0,0], [2,3,4]], [0,2], [0,0,1], [2,1] )

```

Previous tensor.reduce_log_sum Next tensor.compress

Last updated3 months ago