

tensor.flatten

...

```
Copy fnflatten(self:@Tensor, axis:usize)->Tensor;
```

...

Flattens the input tensor into a 2D tensor. If input tensor has shape (1, 2, 3,...n) then the output will have shape (1 * 2 * 3 * ... (axis-1), axis * (axis+1) * ... n).

Args

- self
- (@Tensor
-) - The input tensor.
- axis
- (usize
-) - Indicate up to which input dimensions (exclusive) should be flattened.
-

Panics

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

Returns

A newTensor instance containing the flattened version of the input tensor.

Examples

Case 1: flatten with axis 0

...

```
Copy fnflatten_example()->Tensor { lettensor=TensorTrait:::new( shape:array![2,2,2].span(), data:array!  
[0,1,2,3,4,5,6,7].span(), extra:Option::None(()) );
```

```
returntensor.flatten(0);// equivalent to tensor.reshape(1,8) }
```

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

...

Case 2: flatten with axis 1

...

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

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

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

```
returntensor.flatten(1);// equivalent to tensor.reshape(2,4) }
```

```
[[0,1,2,3],[4,5,6,7]]
```

...

Case 3: flatten with axis 2

...

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

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

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

```
return tensor.flatten(2); // equivalent to tensor.reshape(4,2) }
```

```
[[0,1],[2,3],[4,5],[6,7]]
```

```
...
```

[Previous tensor.asin](#) [Next tensor.sinh](#)

Last updated 3 months ago