

tensor.equal

tensor.equal

...

```
Copy fnequal(self:@Tensor, other:@Tensor)->Tensor;
```

...

Check if two tensors are equal element-wise. The input tensors must have either:

- Exactly the same shape
- The same number of dimensions and the length of each dimension is either a common length or 1.
-

Args

- self
- (@Tensor
-) - The first tensor to be equated
- other
- (@Tensor
-) - The second tensor to be equated
-

Panics

- Panics if the shapes are not equal or broadcastable
-

Returns

A newTensor of booleans (1 if equal, 0 otherwise) with the same shape as the broadcasted inputs.

Examples

Case 1: Compare tensors with same shape

...

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

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

```
fneq_example()->Tensor { lettensor_1=TensorTrait::new( shape:array![3,3,3].span(), data:array![0,1,2,3,4,5,6,7,8].span(), );
```

```
lettensor_2=TensorTrait::new( shape:array![3,3,3].span(), data:array![0,1,2,3,4,5,9,1,5].span(), );
```

```
// We can call equal function as follows. returntensor_1.equal(@tensor_2); }
```

```
[1,1,1,1,1,0,0,0]
```

...

Case 2: Compare tensors with different shapes

...

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

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

```
fneq_example()->Tensor { lettensor_1=TensorTrait::new( shape:array![3,3,3].span(), data:array![0,1,2,3,4,5,6,7,8].span(), );
```

```
lettensor_2=TensorTrait::new(shape:array![3].span(), data:array![0,1,2].span(),);
```

```
// We can call equal function as follows. returntensor_1.equal(@tensor_2); }
```

```
[1,1,1,0,0,0,0,0,0]
```

...

[Previous tensor.log](#) [Next tensor.greater](#)

Last updated 3 months ago