tensor.pow

tensor.pow

```

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

```

Pow takes input data (Tensor) and exponent Tensor, and produces one output data (Tensor) where the function f(x) = x^exponent, is applied to the data tensor elementwise. 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, base of the exponent.
- other
- (@Tensor
-) - The second tensor, power of the exponent.
-

Panics

- Panics if the shapes are not equal or broadcastable
-

Returns

A newTensor 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};

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

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

returntensor_1.pow(@tensor_2); }

                    [0,1,4,0,4,25,0,7,64]

```

Case 2: Compare tensors with different shapes

```

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

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

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

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

returntensor_1.pow(@tensor_2); }
```

[0,1,4,0,4,25,0,7,64]

```

Previous tensor.reduce_mean Next tensor.is_nan

Last updated3 months ago
```