tensor.matmul

```

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

```

Performs matrix product of two tensors. The behavior depends on the dimensionality of the tensors as follows:

- If both tensors are 1-dimensional, the dot product is returned.
- If both arguments are 2-dimensional, the matrix-matrix product is returned.
- If the first argument is 1-dimensional and the second argument is 2-dimensional, a 1 is prepended to its dimension for the purpose of the matrix multiply. After the matrix multiply, the prepended dimension is removed.
- If the first argument is 2-dimensional and the second argument is 1-dimensional, the matrix-vector product is returned.
-

Args

- self
- (@Tensor
-) - the first tensor to be multiplied
- other
- (@Tensor
-) - the second tensor to be multiplied
-

Panics

- Panics if the dimension of the tensors is higher than two.
-

Returns

A newTensor resulting from the matrix multiplication.

Examples

Case 1: Dot product of two vectors (1D * 1D)

```

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

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

fndot_product_example()->Tensor { lettensor_1=TensorTrait:::new(shape:array![3].span(), data:array![0,1,2].span(),);

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

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

                    [5]

```

Case 2: Matrix multiplication (2D * 2D)

```

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

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

fnmatrix_mul_example()->Tensor { lettensor_1=TensorTrait:::new( shape:array![2,2].span(), data:array!
[244,99,109,162].span() );

lettensor_2=TensorTrait::::new( shape:array![2,2].span(), data:array![151,68,121,170].span() );

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

[[48823,33422],[36061,34952]]

```

Case 3: Matrix-Vector multiplication (2D x 1D)

```

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

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

fnmatrix_vec_mul_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].span(), data:array![0,1,2].span(),);

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

[5,14,23]

```

Previous tensor.argmin Next tensor.exp

Last updated1 month ago
```