

# nn.sigmoid

...

```
Copy fnsigmoid(tensor:@Tensor)->Tensor;
```

...

Applies the Sigmoid function to an n-dimensional input tensor rescaling them so that the elements of the n-dimensional output Tensor lie in the range [0,1].

$$\text{sigmoid}(x_i) = \frac{1}{1 + e^{-x_i}}$$

## Args

- tensor
- (@Tensor
- ) - The input tensor.
- 

## Returns

A Tensor of fixed point numbers with the same shape than the input Tensor.

## Type Constraints

Constrain input and output types to fixed point tensors.

## Examples

...

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

```
useorion::operators::tensor::{TensorTrait,Tensor,FP8x23}; useorion::operators::nn::{NNTrait,FP8x23NN};
useorion::numbers::{FP8x23,FixedTrait};
```

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

```
returnNNTrait::sigmoid(@tensor); }
```

```
[[4194304,6132564],[7388661,7990771]] // The fixed point representation of // [[0.5,
0.7310586],[0.88079703, 0.95257413]]
```

...

[Previous nn.leaky\\_relu](#) [Next nn.softmax](#)

Last updated 3 months ago