

# nn.softplus

...

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

...

Applies the Softplus function to an n-dimensional input Tensor such that the elements of the n-dimensional output Tensor lie in the range [-1,1].

$\text{softplus}(x_i) = \log(1 + e^{x_i})$   $\text{softplus}(x_i) = \log(1 + e^{x_i})$

## 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};
```

```
fnsoftplus_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::softplus(@tensor); }
```

```
[[5814540,11016447],[17841964,25573406]] // The fixed point representation of //  
[[0.6931452, 1.31326096],[2.12692796, 3.04858728]]
```

...

[Previous nn.softsign](#) [Next nn.linear](#)

Last updated 3 months ago