nn.softmax_zero

```

Copy fnsoftmax_zero(tensor:@Tensor, axis:usize)->Tensor;

```

Applies the Softmax zero 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] and sum to 1 while keeping the zero elements to zero.

The softmax zero on the set \mathbf{x} = (x_1, ..., x_n) is given by :

! softmax zero (x i) = { 0 x i = 0 e x i ∑ x ∈ S e x otherwise \text{softmax zero}(x_i) = \begin{cases} 0 & \qquad x_i = 0 \\ \frac{e^{x_i}}{ \sum_{x \in {S}} e^{x}} & \qquad \text{otherwise} \end{cases} where S in a subset of \mathbf{x} given by

! S = { (x 1 , … , x k) | 1 ≤ k ≤ n , x j ≠ 0 for 1 ≤ j ≤ k } \ S = { (x_1, \ldots, x_k) \mid 1 \leq k \leq n, x_j \neq 0 \text{ for } 1 \leq j \leq k }

Args

- tensor
- (@Tensor
-) - The input tensor.
- axis
- (usize
-) - The axis along which to compute the softmax zero.
-

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,FP8x23Tensor}; useorion::operators::nn::{NNTrait,FP8x23NN}; useorion::numbers::{FP8x23,FixedTrait};

usecore::debug::PrintTrait;

fnsoftmax_zero_example()->Tensor { lettensor=TensorTrait:::new( shape:array![2,2].span(), data:array![ FixedTrait::new(0,false), FixedTrait::new(8388608,false), FixedTrait::new(16777216,false), FixedTrait::new(25165824,false), ] .span(), );

returnNNTrait::softmax_zero(@tensor,1); }

[[0,0x800000],[2256043,6132564]] // The fixed point representation of // [[0, 1],[0.2689, 0.7311]]

```

Previous nn.softmax Next nn.logsoftmax

Last updated2 months ago