nn.hard_sigmoid

```

Copy fnhard_sigmoid(tensor:@Tensor, alpha:@T, beta:@T)->Tensor;

```

Applies the HardSigmoid function to an n-dimensional input tensor.

! HardSigmoid (x i) = max (0 , min (alpha ∗ x + beta , 1)) \text{HardSigmoid}(x_i) = \text{max}(0, \text{min}(alpha * x + beta, 1))

Args

- tensor
- (@Tensor
-) - The input tensor.
- alpha
- (@T
-) - value of alpha.
- beta
- (@T
-) - value of beta.
-

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::{FP16x16,FixedTrait};

fnhard_sigmoid_example()->Tensor { lettensor=TensorTrait:::::new( shape:array![2,2].span(), data:array![ FixedTrait::new(0,false), FixedTrait::new(13107,false), FixedTrait::new(32768,false), FixedTrait::new(65536,false), ] .span(), ); letalpha=FixedTrait::new(13107,false); letbeta=FixedTrait::new(32768,false);

returnNNTrait::hard_sigmoid(@tensor,@alpha,@beta); }

                    [[32768,35389],[39321,45875]]

```

Previous nn.linear Next nn.thresholded_relu

Last updated3 months ago