normalize.predict

```

Copy fnpredict(X:Tensor, norm:NORM)->Tensor;

```

Returns the normalized input. Tree different types of normalization can be performed and are defined as follow : MAX: Y = \frac{X}{max(X)} L1: Y = \frac{X}{sum(X)} L2: Y = \frac{X}\sqrt{{sum(X²)}} For batches, that is, [N,C] tensors, normalization is done along the C axis. In other words, each row of the batch is normalized independently.

Args

- X
- (@Tensor
-) - Input 2D tensor.
- norm
- (NORM
-) - NORM::MAX, NORM::L1 or NORM::L2
-

Returns

- Tensor - output tensor
-

Examples

```

Copy useorion::numbers::FP16x16; useorion::operators::tensor::
{Tensor,TensorTrait,FP16x16Tensor,FP16x16TensorDiv,FP16x16TensorPartialEq};

useorion::operators::ml::normalizer::normalizer::{ NormalizerTrait,NORM };

fnnormalizer_max()->Tensor { letmutshape=ArrayTrait:::new(); shape.append(3); shape.append(3);

letmutdata=ArrayTrait::new(); data.append(FP16x16{ mag:65536, sign:true}); data.append(FP16x16{ mag:52428,
sign:true}); data.append(FP16x16{ mag:39321, sign:true}); data.append(FP16x16{ mag:26214, sign:true});
data.append(FP16x16{ mag:13107, sign:true}); data.append(FP16x16{ mag:0, sign:false}); data.append(FP16x16{
mag:13107, sign:false}); data.append(FP16x16{ mag:26214, sign:false}); data.append(FP16x16{ mag:39321, sign:false});

letX=TensorTrait::new(shape.span(), data.span());

returnNormalizerTrait::predict(X, NORM::MAX); }

                    [[-1.-0.8-0.6] [-1.-0.50.] [0.33333330.66666661.]]

```

Previous Normalizer Next Models

Last updated15 days ago