tensor.layer_normalization

```

Copy fnlayer_normalization( self:@Tensor, scale:@Tensor, B:Option<@Tensor>, axis:Option, epsilon:Option, stash_type:Option, )->(Tensor,Tensor,Tensor);

```

Layer normalization of the input, in two stages. The first stage is standardization, which makes the normalized elements have zero mean and unit variances. The second stage then scales and shifts the outcome of the first stage

Args

- self
- (@Tensor
-) - The input tensor.
- scale
- (@Tensor,
-) - Scale tensor.
- B
- (Option<@Tensor>
-) - Bias tensor.
- axis
- (Option
-) (default is -1) - The first normalization dimension. If rank(X) is r, axis' allowed range is [-r, r). Negative value means counting dimensions from the back.
- epsilon
- (Option
-) (default is 0) - The epsilon value to use to avoid division by zero.
- stash_type
- (Option
-) - Precise the computation precision - unused the precision is defined by the type of the tensor.
-

Panics

- Panics if condition rank is not equal to 1.
-

Returns

A new normalized tensorTensor . A tensor containing the meanTensor . A tensor containing the inverse standard deviationTensor .

Example

```

Copy useorion::operators::tensor::{TensorTrait,Tensor}; useorion::operators::tensor::FP16x16TensorPartialEq; usecore::array::{ArrayTrait,SpanTrait}; useorion::operators::tensor::FP16x16Tensor; useorion::numbers::{FixedTrait,FP16x16};

fnlayer_normalization_example()->(Tensor,Tensor,Tensor) { letmutshape=ArrayTrait::::new(); shape.append(3); shape.append(4);

letmutdata=ArrayTrait::new(); data.append(FP16x16{ mag:41143, sign:true}); data.append(FP16x16{ mag:51803, sign:false}); data.append(FP16x16{ mag:113556, sign:false}); data.append(FP16x16{ mag:64774, sign:false}); data.append(FP16x16{ mag:866, sign:false}); data.append(FP16x16{ mag:698, sign:true}); data.append(FP16x16{ mag:106500, sign:false}); data.append(FP16x16{ mag:98929, sign:false}); data.append(FP16x16{ mag:7551, sign:false}); data.append(FP16x16{ mag:30689, sign:true}); data.append(FP16x16{ mag:38325, sign:false}); data.append(FP16x16{ mag:48164, sign:false}); letX=TensorTrait::new(shape.span(), data.span());

letshape=ArrayTrait::::new(); shape.append(4); letmutdata=ArrayTrait::new(); data.append(FP16x16{ mag:49855, sign:false}); data.append(FP16x16{ mag:150787, sign:false}); data.append(FP16x16{ mag:83498, sign:true}); data.append(FP16x16{ mag:30346, sign:false}); letscale=TensorTrait::new(shape.span(), data.span());

letmutshape=ArrayTrait::::new(); shape.append(4); letmutdata=ArrayTrait::new(); data.append(FP16x16{ mag:54864, sign:true}); data.append(FP16x16{ mag:50952, sign:false}); data.append(FP16x16{ mag:8870, sign:true});
```

data.append(FP16x16{ mag:23216, sign:true}); letbias=TensorTrait::new(shape.span(), data.span());

returnX.layer_normalization(@scale,Option::Some(@bias),Option::None,Option::None,Option::None); }

```
                [[-0.489265531.0185822-0.02138367-0.39223218] [-0.79455490.996960460.04332176-
                0.412645] [-0.56647070.7491956-0.7896356-0.5320859]]
```

```

Previous tensor.compress Next Neural Network

Last updated2 months ago
```