

# tensor.qlinear\_leakyrelu

...

Copy fnqlinear\_leakyrelu(self:@Tensor, a\_scale:@Tensor, a\_zero\_point:@Tensor, alpha:T)->Tensor::;

...

Applies the Leaky Relu operator to a quantized Tensor

QLinar LeakyRelu takes as input a quantized Tensor, its scale and zero point and an scalar alpha, and produces one output data (a quantized Tensor) where the function  $f(x) = \alpha * x$  for  $x < 0$ ,  $f(x) = x$  for  $x \geq 0$ , is applied to the data tensor elementwise. The quantization formula is  $y = \text{saturate}((x / y\_scale) + y\_zero\_point)$ . Scale and zero point must have same shape and the same type. They must be either scalar (per tensor) or N-D tensor (per row for 'a' and per column for 'b'). Scalar refers to per tensor quantization whereas N-D refers to per row or per column quantization.

## Args

- self
- (@Tensor
- ) - The first tensor to be multiplied (a).
- a\_scale
- (@Tensor
- ) - Scale for inputa
- .
- a\_zero\_point
- (@Tensor
- ) - Zero point for inputa
- .
- alpha
- (T
- ) - The factor multiplid to negative elements.
- 

## Returns

A newTensor , containing result of the Leaky Relu.

## Type Constraints

u32 tensor, not supported. fp8x23wide tensor, not supported. fp16x16wide tensor, not supported. bool tensor, not supported.

## Example

...

## Copy

```
usecore::array::{ArrayTrait,SpanTrait};
```

```
useorion::operators::tensor::{TensorTrait,Tensor,I8Tensor,FP16x16Tensor}; useorion::numbers::{FP16x16,FP16x16Impl,FixedTrait};
```

```
fnqlinear_leakyrelu_example()->Tensor { leta=TensorTrait::< i8
```

```
    ::new( shape:array![2,3].span(), data:array![-10, -10, -10, 10, 10, 10 ].span(), );
```

```
leta_scale=TensorTrait::< FP16x16
```

```
    ::new(shape:array![1].span(), data:array![FixedTrait:::new(327680,false)].span(),); leta_zero_point=TensorTrait::< FP16x16 ::new(shape:array![1].span(), data:array![FixedTrait:::new(131072,false)].span(),);
```

```
letalpaha=FixedTrait:::new(655360,false);
```

```
return=a .qlinear_leakyrelu( @a_scale,@a_zero_point, alpha ); }
```

```
    [[-118,-118,-118], [10,10,10]]
```

...

[Previous tensor.qlinear\\_concat](#) [Next tensor.nonzero](#)

Last updated 1 month ago