# nn.gemm

```

Copy fngemm( A:Tensor, B:Tensor, C:Option>, alpha:Option, beta:Option, transA:bool, transB:bool )->Tensor;

```

Performs General Matrix multiplication:https://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms#Level_3

- A' = transpose(A) if transA else A
- B' = transpose(B) if transB else B
-

ComputeY = alpha * A' * B' + beta * C , where input tensor A has shape (M, K) or (K, M), input tensor B has shape (K, N) or (N, K), input tensor C is broadcastable to shape (M, N), and output tensor Y has shape (M, N).A will be transposed before doing the computation if attributetransA istrue , same forB andtransB .

Args

- A
- (Tensor
- ) - Input tensor A. The shape ofA
- should be (M, K) iftransA
- isfalse
- , or (K, M) iftransA
- istrue
- .
- B
- (Tensor
- ) - Input tensor B. The shape ofB
- should be (K, N) iftransB
- isfalse
- , or (N, K) iftransB
- istrue
- .
- C
- (Option>
- ) - Optional input tensor C. The shape of C should be unidirectional broadcastable to (M, N).
- alpha
- (Option
- ) - Optional scalar multiplier for the product of input tensorsA * B
- .
- beta
- (Option
- ) - Optional scalar multiplier for input tensorC
- .
- transA
- (bool
- ) - WhetherA
- should be transposed.
- transB
- (bool
- ) - WhetherB
- should be transposed.
-

Returns

ATensor of shape (M, N).

Examples

```

Copy modinput_0; modinput_1; modinput_2;

useorion::operators::nn::NNTrait; useorion::numbers::FixedTrait; useorion::operators::nn::FP16x16NN;
useorion::operators::tensor::FP16x16TensorPartialEq;
```

```
fngemm_all_attributes_example()->Tensor { letinput_0=input_0::input_0();// shape [4;3] letinput_1=input_1::input_1();// shape [5;4] letinput_2=input_2::input_2();// shape [1;5]

lety=NNTrait::gemm( input_0, input_1, Option::Some(input_2), Option::Some(FixedTrait::new(16384,false)),// 0.25 Option::Some(FixedTrait::new(22938,false)),// 0.35 true, true );

returny; }

                    tensor of shape [3;5]
```

Previous nn.thresholded_relu Next Machine Learning

Last updated1 month ago