tensor.reverse_sequence

```
Copy fn reverse_sequence(self: @Tensor, sequence_lens: @Tensor, batch_axis: Option, time_axis: Option) -> Tensor;
```

Reverse batch of sequences having different lengths specified by sequence_lens.

- self
- (@Array>
-) - Tensor of rank r >= 2.
- sequence_lens
- (@Tensor
-) - Tensor specifying lengths of the sequences in a batch. It has shape [batch_size].
- batch_axis
- (Option
-) - (Optional) Specify which axis is batch axis. Must be one of 1 (default), or 0.
- time_axis
- (Option
-) - (Optional) Specify which axis is time axis. Must be one of 0 (default), or 1.
-

Panics

- Panics if the 'batch_axis' == 'time_axis'.
- Panics if the 'batch_axis' and 'time_axis' are not 0 and 1.
- Panics if the 'sequence_len' exceeding the sequence range.
-

Returns

Tensor with same shape of input.

Example

```
Copy usecore::array::{ArrayTrait,SpanTrait}; useorion::operators::tensor::{TensorTrait,Tensor,U32Tensor}; usecore::option::OptionTrait; fnreverse_sequence_example()->Tensor { lettensor:Tensor=TensorTrait::::new( shape:array![4,4].span(), data:array![ 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 ].span(), ); letsequence_lens=TensorTrait::::new(array![4].span(),array![1,2,3,4].span()); letbatch_axis=Option::Some(0); lettime_axis=Option::Some(1); // We can call split function as follows. returntensor.reverse_sequence(sequence_lens, batch_axis, time_axis); }

                    [ [0,1,2,3], [5,4,6,7], [10,9,8,11], [15,14,13,12] ]
```

Previous tensor.optional Next tensor.split_to_sequence

Last updated15 days ago