

Contiguous Tensors in PyTorch

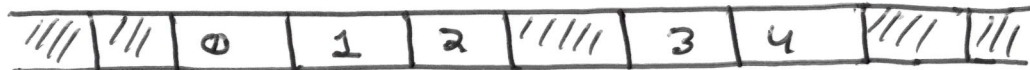
#1

① A one-dimensional array (tensor) in PyTorch is contiguous if its components are laid out in memory in consecutive positions such that the following representation is true:

$t = [0, 1, 2, 3, 4]$



② It is not contiguous if the region in memory where it is stored looks like this:



③ For two-dimensional arrays (tensors) or even higher-dimensional tensors to be contiguous, elements must also be next to each other but the order follows different conventions.

④ Let's consider the following two-dimensional tensor:

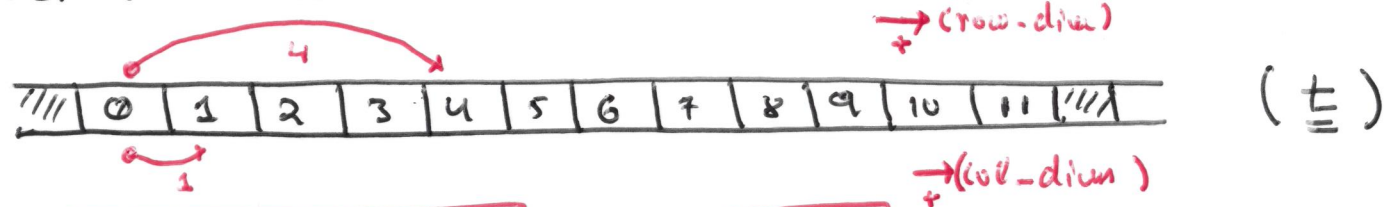
```
t = torch.tensor([0, 1, 2, 3], [4, 5, 6, 7], [8, 9, 10, 11])  
t.shape = torch.Size([3, 4])
```

5) Tensor t is of the following form:

t =

0	1	2	3
4	5	6	7
8	9	10	11

6) The memory allocation for t is C contiguous if row elements (rows) are stored next to each other like this:



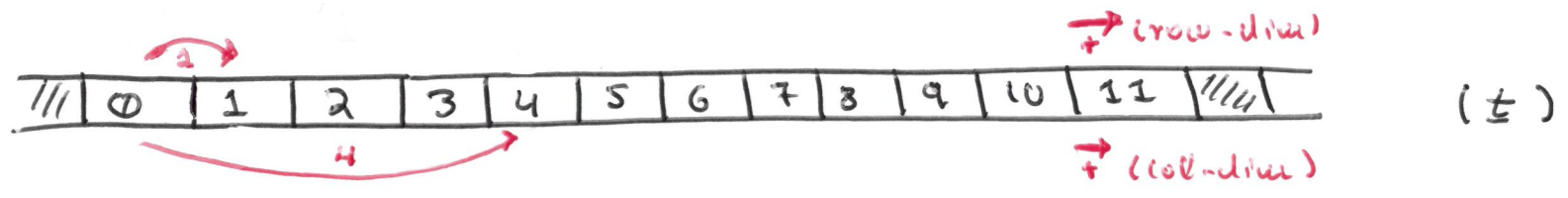
7) Executing `t.is_contiguous()` returns `True`.

8) Executing `t.stride()` returns `(4, 1)`. PyTorch's `stride()` method gives the number of bytes to skip in order to get the next element in each dimension. Thus, we need 4 bytes to go to the next line and 1 byte to go to the next element in the same line, i.e. the next column. (row-dim = 0, column-dim = 1)

9) Transposing tensor t can be performed by executing:

```
t = t.transpose(0, 1)
t.shape = torch.Size([4, 3])
```

(10) The memory allocation for the two-dimensional tensor did not change.



t =

0	4	8
1	5	9
2	6	10
3	7	11

(11) `t.stride()` returns `(1, 4)`

Thus, we need 1 byte to go to the next line and 4 bytes to go to the next element in the same line, i.e. the next column.

(12) Executing `t.is_contiguous()` returns `False`.

(13) Converting the transposed vector to a contiguous one yields:

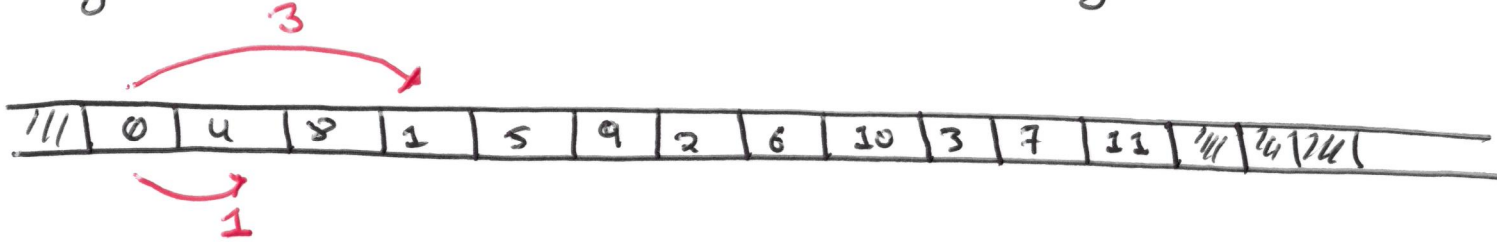
`t = t.contiguous()`

`t.shape = torch.size([4, 3])` (Tensor's shape does not change)

(14) Executing the `t.stride()` return `(3, 1)`.

has been

⑮ This means that the internal representation of the array
changed so that the tensor is C contiguous.



⑯ Therefore, 3 bytes need to be skipped in order to get to the next line
and 1 byte needs to be skipped in order to go to the next element
within the same line (i.e. next column).