

## Tensor operations part 2

**Tip 1:** You can find descriptions of the operations needed to complete tasks by typing the names operation in the "search" field in the upper right corner:

<https://www.tensorflow.org/>.

For example for tf.where type "tf where", etc.

**Tip 2:** Instead of tf.expand\_dims operation, you can use indexing from "tf.newaxis"

Example:

```
foo = tf.constant([[1,2,3], [4,5,6], [7,8,9]])
print(foo[tf.newaxis, :, :]) # => [[[1,2,3], [4,5,6], [7,8,9]]]
print(foo[:, tf.newaxis, :]) # => [[[1,2,3]], [[4,5,6]], [[7,8,9]]]
print(foo[:, :, tf.newaxis]) # => [[[1],[2],[3]], [[4],[5],[6]], [[7],[8],[9]]]
```

If we do not want to write many ":" then we can use "...".

The expression: `foo[tf.newaxis, :, :]` is equivalent to `foo[tf.newaxis, ...]`.

More about tensor slicing here: [https://www.tensorflow.org/guide/tensor\\_slicing](https://www.tensorflow.org/guide/tensor_slicing)

### Task 1

Write the function that:

- Will accept a tensor of any shape of tf.float32 type.
- Creates a mask according to the condition  $x > 0$
- Based on the mask, it will create a vector of elements from the input tensor
- Will carry out an algebraic multiplication of the resulting vector by itself, but with transposition of the second factor.
- Adds an extra dimension in the 0 axis and repeats the tensor 5 times along that axis
- Return the result

Call function for three tensors created by using the `tf.random.uniform()` operation with values in the range `<-10,10>` with "shape" arguments:

- [3]
- [33, 7]
- [2, 32, 6]

Use operations: `tf.boolean_mask`, `tf.transpose`, `tf.matmul`, `tf.expand_dims`, `tf.repeat`

### Task 2

Write the function that:

- Will accept a tensor of any shape of tf.float32 type.
- Checks if the minimum value in the tensor is greater than zero. If it is, then It will separate the first column of the tensor, and if not, the second column (using slicing).
- Divides the resulting tensor by 8
- Round the values in the tensor to the nearest integers
- Change the tensor type to tf.int32 and return it

Call function for tensors:

```
[[5, 6], [7, 3], [4, 5]] and [[-4, 5, 6], [7, 8, 9]]
```

Use operations: `tf.constant`, `tf.where`, `tf.math.round`, `tf.cast`

### Task 3

Write the function that:

- Accepts tensors of arbitrary but identical shapes of `tf.float32` type.
- Creates a tensor of element indices greater than 0 and less than 4 for the tensor  
First
- Based on this index tensor, it will replace the corresponding values in the tensor  
the second at 7
- Returns the modified tensor

Call function for tensors:

- `x1 = [[8, 9, 0], [2, 3, 4], [6, 1, 8]]` and `x2 = [[4, 5, 8], [23, 32, 14], [6, 1, 15]]`
- `x1 = [[8, 9], [3, 4], [6, 2]]` and `x2 = [[5, 8], [23, 14], [6, 1]]`

Use operations: `tf.constant`, `tf.where`, `tf.scatter_nd_update`