assignment-03-a

September 19, 2020

1 Manipulate PyTorch Tensors

1.1 Matrix manipulation

[1]:

```
import torch
```

1.1.1 Make the matrices A and B below. Add them together to obtain a matrix C. Print these three matrices.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} \qquad C = A + B = ?$$

[4]:

```
# write your code here

A = torch.tensor([[1, 2], [3, 4]])
B = torch.tensor([[10, 20], [30, 40]])
C = torch.add(A, B)



# print
print(A)
print('')
print(B)
print('')
print(C)
```

```
tensor([[1, 2],
        [3, 4]])

tensor([[10, 20],
        [30, 40]])

tensor([[11, 22],
        [33, 44]])
```

1.1.2 Print the dimension, size and type of the matrix A. Remember, the commands are dim(), size() and type()

[20]:

```
# write your code here

print(A.dim())    # print the dimension of the matrix A
print('')
print(A.size())    # print the size of the matrix A
print('')
print(A.type())    # print the type of the matrix A
```

```
2

torch.Size([2, 2])

torch.LongTensor
```

1.1.3 Convert the matrix A to be an integer matrix (type LongTensor). Remember, the command is long(). Then print the type to check it was indeed converted.

[22]:

```
# write your code here

A_long = A.long()

print(A_long)    # print the type of A_long
print('')
print(A_long.type())    # print the type of A
```

```
tensor([[1, 2],
        [3, 4]])

torch.LongTensor
```

1.1.4 Make a random 5 x 2 x 3 Tensor. The command is torch.rand. Then do the following: 1) Print the tensor, 2) Print its type, 3) Print its dimension, 4) Print its size, 5) Print the size of its middle dimension.

[33]:

```
# write your code here

A = torch.rand(5, 2, 3)

print(A)
print(A.type())    # print the type of A
print(A.dim())    # print the dimension of A
print(A.size())    # print the size of A
print(A.size()[1])    # print the size of the middle (second) dimension
```

```
tensor([[[0.3183, 0.3791, 0.8339],
         [0.3556, 0.2423, 0.5296]],

        [[0.9731, 0.2611, 0.2696],
         [0.5181, 0.1741, 0.6677]],

        [[0.7691, 0.3227, 0.5804],
         [0.4827, 0.6731, 0.7520]],

        [[0.9679, 0.0374, 0.9244],
         [0.2635, 0.6079, 0.1828]],

        [[0.5541, 0.3967, 0.6108],
         [0.2386, 0.8806, 0.0037]]])
torch.FloatTensor
3
torch.Size([5, 2, 3])
2
```

1.1.5 Make 2 x 3 x 4 x 5 tensor filled with zeros then print it. (The command is torch.zeros). See if you can make sense of the display.

[24]:

```python
# write your code here

A = torch.zeros(2, 3, 4, 5)

print(A)
```

```
tensor([[[[0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.]],

         [[0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.]],

         [[0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.]]],


        [[[0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.]],

         [[0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.]],

         [[0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.],
          [0., 0., 0., 0., 0.]]]])
```