**PYTORCH**

**1.Importing pytorch:**

import torch

**2.Pytorch Functions:**

* torch.rand(2,2)

Displays two dimensional tensor with random values.

tensor([[0.8999, 0.3852],

[0.8046, 0.8834]])

* x=torch.empty(4)

print(x)

Displays uninitialized data.

tensor([-1.2717e+07, 4.5052e-41, -2.0080e-09, 3.3476e-41])

* torch.empty(2,3)

tensor([[-1.2717e+07, 4.5052e-41, -2.0909e-09],

[ 3.3476e-41, 4.4842e-44, 0.0000e+00]])

* torch.empty(2,2,3)

tensor([[[ 1.1625e+27, 1.4580e-19, 7.1856e+22],

[ 4.3605e+27, 1.5766e-19, 7.1856e+22]],

[[ 1.5835e-43, 0.0000e+00, -1.9545e-09],

[ 3.3476e-41, -1.9608e-09, 3.3476e-41]]])

* torch.zeros(2,2)

Displays 2D tensor with zero values.

tensor([[0., 0.],

[0., 0.]])

* x=torch.ones(2,2)

print(x)

Displays 2D tensor with ones values.

tensor([[1., 1.],

[1., 1.]])

* x.size()

Displays size of the function.

torch.Size([2, 2])

* x.ndim

2

* torch.manual\_seed(45)

x=torch.rand(2,2)

print(x)

y=torch.rand(2,2)

print(y)

Fixes the random values that never change whenever execution takes place.

tensor([[0.1869, 0.9613],

[0.6834, 0.8988]])

tensor([[0.0505, 0.5555],

[0.7861, 0.0566]])

* z=x+y

print(z)

tensor([[0.2375, 1.5168],

[1.4696, 0.9554]])

Adds x and y value.

* torch.add(x,y)

Adds values using torch .

tensor([[0.2375, 1.5168],

[1.4696, 0.9554]])

* x[:,0]

Displays 1st column from the data.

* x[:,1]

Displays 2nd column from the data.

* x=torch.rand(1)

print(x)

x.item()

Displays actual value of only one tensor.

* x=torch.rand(4,4)

print(x)

x.view(16)

Makes 2D array to 1D array.

tensor([[0.1032, 0.0170, 0.5811, 0.3412],

[0.2945, 0.3636, 0.3370, 0.1347],

[0.9134, 0.2588, 0.7716, 0.4666],

[0.0070, 0.5136, 0.4281, 0.8688]])

tensor([0.1032, 0.0170, 0.5811, 0.3412, 0.2945, 0.3636, 0.3370, 0.1347, 0.9134,

0.2588, 0.7716, 0.4666, 0.0070, 0.5136, 0.4281, 0.8688])

* a=torch.ones(6)

print(a)

print(type(a))

b=a.numpy()

print(b)

print(type(b))

* Converts tensor to numpy array.

tensor([1., 1., 1., 1., 1., 1.])

<class 'torch.Tensor'>

[1. 1. 1. 1. 1. 1.]

<class 'numpy.ndarray'>

* import numpy as np

a=np.ones(5)

print(a)

print(type(a))

b=torch.from\_numpy(a)

print(b)

* Converts numpy array to tensor.

[1. 1. 1. 1. 1.]

<class 'numpy.ndarray'>

tensor([1., 1., 1., 1., 1.], dtype=torch.float64)