



From Fundamentals to Building
Your Own Intelligent System

AI & Machine Learning Bootcamp 2025

Dr. Fazlul Hasan Siddiqui & Dr. Sabah Binte Noor

Deep
learning
library

What is PyTorch?

Automatic
differentiat
ion engine

Tensor
Library

Tensors

- A data container

Scalar (rank-0 tensor)

In pure Python

```
a = 1.  
print(a)
```

1.0

In PyTorch

```
import torch  
a = torch.tensor(1.)  
print(a)
```

tensor(1.)

Vector (rank-1 tensor)

In PyTorch

```
a = torch.tensor([1., 2., 3.])  
print(a)
```

tensor([1., 2., 3.])

Matrix (rank-2 tensor)

```
a = torch.tensor([[1., 2., 3.],  
                  [2., 3., 4.]])  
a.shape
```

torch.Size([2, 3])

1. Creating Tensors

```
import torch

m = torch.tensor([[1., 2., 3.],
                  [4., 5., 6.]])
print(v)
```

```
tensor([[1., 2., 3.],
        [4., 5., 6.]])
```

2. Checking The Shape

```
print(m)
```

```
tensor([[1., 2., 3.],  
        [4., 5., 6.]])
```

```
print(m.shape)
```

```
torch.Size([2, 3])
```

3. Checking Number of Dimensions

```
print(m)
```

```
tensor([[1., 2., 3.],  
        [4., 5., 6.]])
```

```
print(m.ndim)
```

```
2
```

4. Checking the Data Type

```
print(m)
```

```
tensor([[1., 2., 3.],  
        [4., 5., 6.]])
```

```
print(m.dtype)
```

```
torch.float32
```

```
other_m = torch.tensor([[1, 2, 3],  
                        [4, 5, 6]])
```

```
print(other_m)
```

```
tensor([[1, 2, 3],  
        [4, 5, 6]])
```

```
print(other_m.dtype)
```

```
torch.int64
```

5. Creating a Tensor from NumPy Arrays

```
import numpy as np

np_ary = np.array([1., 2., 3.])
m2 = torch.from_numpy(np_ary)

print(m2)
```

```
tensor([1., 2., 3.], dtype=torch.float64)
```


6. Changing the Tensor Shape

```
print(m)
```

```
tensor([[1., 2., 3.],  
        [4., 5., 6.]])
```

```
print(m.view(3, 2))
```

```
tensor([[1., 2.],  
        [3., 4.],  
        [5., 6.]])
```

```
print(m.view(-1, 2))
```

```
tensor([[1., 2.],  
        [3., 4.],  
        [5., 6.]])
```

```
print(m.view(3, -1))
```

```
tensor([[1., 2.],  
        [3., 4.],  
        [5., 6.]])
```

7. Transposing Tensor Shape

```
print(m)
```

```
tensor([[1., 2., 3.],  
        [4., 5., 6.]])
```

```
print(m.T)
```

```
tensor([[1., 4.],  
        [2., 5.],  
        [3., 6.]])
```

8. Multiplying Matrices

```
m.matmul(m.T)
```

[73]:

```
tensor([[14., 32.],  
        [32., 77.]])
```

Tensor Cheat Sheet

1. torch.**tensor**(data) \\ convert numpy array into tensor
2. my_tensor.**shape** \\ checking the shape
3. torch.**randn**(shape) \\ initialize a tensor with random values
4. t1 **@** t2 \\ matrix multiplication
5. t1.**view**(shape) \\ changing the shape
6. t1.**T** \\ transposing a tensor