Understanding PyTorch: Tensors, Vectors, and Matrices

Bindeshwar Singh Kushwaha

PostNetwork Academy

What is PyTorch?

- PyTorch is an open-source deep learning framework.

What is PyTorch?

- PyTorch is an open-source deep learning framework.
- It supports dynamic computation graphs.

What is PyTorch?

- PyTorch is an open-source deep learning framework.
- It supports dynamic computation graphs.
- Designed to be Pythonic and flexible.

What is PyTorch?

- PyTorch is an open-source deep learning framework.
- It supports dynamic computation graphs.
- Designed to be Pythonic and flexible.
- Commonly used for research and production in AI/ML.

Types of Tensors

- A **scalar** is a 0-dimensional tensor.

Types of Tensors

- A **scalar** is a 0-dimensional tensor.
- A **vector** is a 1-dimensional tensor.

Types of Tensors

- A **scalar** is a 0-dimensional tensor.
- A **vector** is a 1-dimensional tensor.
- A **matrix** is a 2-dimensional tensor.

Types of Tensors

- A **scalar** is a 0-dimensional tensor.
- A **vector** is a 1-dimensional tensor.
- A **matrix** is a 2-dimensional tensor.
- **Tensors** can be of higher dimensions (3D, 4D, ...).

Python Code

```
import torch
s = torch.tensor(5)                    # Scalar
v = torch.tensor([1.0, 2.0, 3.0])      # Vector
m = torch.tensor([[1, 2], [3, 4]])     # Matrix
print("Scalar:", s)
print("Vector:", v)
print("Matrix:", m)
```

Creating Tensors

- Use `torch.tensor()`, `torch.zeros()`, `torch.ones()`, etc.

Creating Tensors

- Use `torch.tensor()`, `torch.zeros()`, `torch.ones()`, etc.
- Specify shape, data type, and device as needed.

Python Code

```
import torch
a = torch.tensor([1, 2, 3])
b = torch.zeros((2, 3))
c = torch.ones((3, 3), dtype=torch.float32)
print(a)
print(b)
print(c)
```

Tensor Properties

- `.shape`: shape of the tensor

Tensor Properties

- `.shape`: shape of the tensor
- `.dtype`: data type

Tensor Properties

- `.shape`: shape of the tensor
- `.dtype`: data type
- `.device`: where the tensor is stored (CPU or GPU)

Python Code

```
import torch
x = torch.tensor([[1, 2, 3], [4, 5, 6]])
print(x.shape)   # torch.Size([2, 3])
print(x.dtype)   # torch.int64
print(x.device)  # cpu
```

Arithmetic Operations

- Element-wise operations: `+`, `-`, `*`, `/`

Arithmetic Operations

- Element-wise operations: `+`, `-`, `*`, `/`
- Matrix multiplication: `torch.mm()` or `@`

Python Code

```
import torch
a = torch.tensor([[1, 2], [3, 4]])
b = torch.tensor([[5, 6], [7, 8]])
print(a + b)          # Addition
print(a * b)          # Element-wise multiplication
print(torch.mm(a, b)) # Matrix multiplication
```

Tensors on GPU

- Use `.to('cuda')` or `.cuda()` to move to GPU.

Tensors on GPU

- Use `.to('cuda')` or `.cuda()` to move to GPU.
- Always check GPU availability using `torch.cuda.is_available()`
- CUDA stands for Compute Unified Device Architecture.
- It is a parallel computing platform and programming model developed by NVIDIA that allows you to use the GPU (Graphics Processing Unit) to perform general-purpose computation — not just graphics.

Python Code

```
import torch
if torch.cuda.is_available():
  x = torch.tensor([1.0, 2.0])
  x = x.to('cuda')
  print(x.device)  # Should print "cuda:0"
```

Automatic Differentiation (Autograd)

- PyTorch tracks operations on tensors with `requires_grad=True`

Automatic Differentiation (Autograd)

- PyTorch tracks operations on tensors with `requires_grad=True`
- Use `.backward()` to compute gradients.

Python Code

```
import torch
x = torch.tensor(2.0, requires_grad=True)
y = x**3 + 2*x**2 + 3*x + 1
y.backward()
print(x.grad)  # dy/dx = 3x^2 + 4x + 3 = 23 when x=2
```

Reach PostNetwork Academy

Website

www.postnetwork.co

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

Reach PostNetwork Academy

Website

www.postnetwork.co

YouTube Channel

www.youtube.com/@postnetworkacademy

Facebook Page

www.facebook.com/postnetworkacademy

LinkedIn Page

www.linkedin.com/company/postnetworkacademy

Thank You!