Matrix Operations with PyTorch

Bindeshwar Singh Kushwaha

PostNetwork Academy

Matrix Addition and Scalar Multiplication

- **Matrix Addition:** We add corresponding elements of the same-sized matrices:

$$A + B = [a_{ij} + b_{ij}]$$

Matrix Addition and Scalar Multiplication

- **Matrix Addition:** We add corresponding elements of the same-sized matrices:

$$A + B = [a_{ij} + b_{ij}]$$

- **Scalar Multiplication:** Multiply each element of the matrix by the scalar value:

$$kA = [k \cdot a_{ij}]$$

Matrix Addition and Scalar Multiplication

- **Matrix Addition:** We add corresponding elements of the same-sized matrices:

$$A + B = [a_{ij} + b_{ij}]$$

- **Scalar Multiplication:** Multiply each element of the matrix by the scalar value:

$$kA = [k \cdot a_{ij}]$$

- **Example:**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 4 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 6 & 8 \\ 1 & -3 & -7 \end{bmatrix}$$

Matrix Addition and Scalar Multiplication

- **Matrix Addition:** We add corresponding elements of the same-sized matrices:

$$A + B = [a_{ij} + b_{ij}]$$

- **Scalar Multiplication:** Multiply each element of the matrix by the scalar value:

$$kA = [k \cdot a_{ij}]$$

- **Example:**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 4 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 6 & 8 \\ 1 & -3 & -7 \end{bmatrix}$$

- **Results:**

$$A + B = \begin{bmatrix} 5 & 4 & 11 \\ 1 & 1 & -2 \end{bmatrix}, \quad 3A = \begin{bmatrix} 3 & -6 & 9 \\ 0 & 12 & 15 \end{bmatrix}$$

Matrix Addition and Scalar Multiplication

- **Matrix Addition:** We add corresponding elements of the same-sized matrices:

$$A + B = [a_{ij} + b_{ij}]$$

- **Scalar Multiplication:** Multiply each element of the matrix by the scalar value:

$$kA = [k \cdot a_{ij}]$$

- **Example:**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 4 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 6 & 8 \\ 1 & -3 & -7 \end{bmatrix}$$

- **Results:**

$$A + B = \begin{bmatrix} 5 & 4 & 11 \\ 1 & 1 & -2 \end{bmatrix}, \quad 3A = \begin{bmatrix} 3 & -6 & 9 \\ 0 & 12 & 15 \end{bmatrix}$$

- **Note:** Matrix addition is only defined when both matrices have the same dimensions.

Matrix Multiplication

- **Matrix Multiplication:** Multiply row of A with column of B:

$$(AB)_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

Matrix Multiplication

- **Matrix Multiplication:** Multiply row of A with column of B:

$$(AB)_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

- This operation is only valid if the number of columns in A equals the number of rows in B.

Matrix Multiplication

- **Matrix Multiplication:** Multiply row of A with column of B:

$$(AB)_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

- This operation is only valid if the number of columns in A equals the number of rows in B.
- **Example:**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 4 & 5 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 \\ 1 & -1 \\ 3 & 4 \end{bmatrix}$$

Matrix Multiplication

- **Matrix Multiplication:** Multiply row of A with column of B:

$$(AB)_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

- This operation is only valid if the number of columns in A equals the number of rows in B.
- **Example:**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 4 & 5 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 \\ 1 & -1 \\ 3 & 4 \end{bmatrix}$$

- **Result:**

$$AC = \begin{bmatrix} 9 & 14 \\ 19 & 16 \end{bmatrix}$$

Matrix Multiplication

- **Matrix Multiplication:** Multiply row of A with column of B:

$$(AB)_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

- This operation is only valid if the number of columns in A equals the number of rows in B.
- **Example:**

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 4 & 5 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 \\ 1 & -1 \\ 3 & 4 \end{bmatrix}$$

- **Result:**

$$AC = \begin{bmatrix} 9 & 14 \\ 19 & 16 \end{bmatrix}$$

- **Note:** Matrix multiplication is not commutative, i.e., $AB \neq BA$.

PyTorch Implementation

This Python code performs the matrix operations above using PyTorch tensors.

Python Code

```
import torch

A = torch.tensor([[1, -2, 3], [0, 4, 5]], dtype=torch.float)
B = torch.tensor([[4, 6, 8], [1, -3, -7]], dtype=torch.float)
C = torch.tensor([[2, 0], [1, -1], [3, 4]], dtype=torch.float)

print("A + B =", A + B)
print("3A =", 3 * A)
print("2A - 3B =", 2 * A - 3 * B)
print("A x C =", torch.matmul(A, C))
```

Note: ‘torch.matmul‘ or ‘@‘ operator is used for matrix multiplication.

PyTorch Implementation of Example 2.4

We compute the same dot products using PyTorch. Note:

- 'torch.dot(t1, t2)' performs the dot product when 't1' and 't2' are 1D tensors.
- Dot product is equivalent to:

$$\text{row vector} \times \text{column vector} \Rightarrow 1 \times n \cdot n \times 1 = \text{scalar}$$

- 'torch.matmul' requires 2D input for matrix multiplication. For example:

$$\texttt{torch.matmul(u.view(1,3), v.view(3,1))} \Rightarrow [[18]]$$

- This simulates full matrix multiplication:

$$\begin{bmatrix} 7 & -4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = [18]$$

PyTorch Implementation of Example 2.4

Python Code

```
import torch
u = torch.tensor([7, -4, 5], dtype=torch.float)
v = torch.tensor([3, 2, 1], dtype=torch.float)
dot_a = torch.dot(u, v)  # Equivalent to matrix 1x3 * 3x1
x = torch.tensor([6, 1, 8, 3], dtype=torch.float)
y = torch.tensor([4, 9, 2, 5], dtype=torch.float)
dot_b = torch.dot(x, y)

print("Dot Product (a):", dot_a.item())  # 18
print("Dot Product (b):", dot_b.item())  # 64
```

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!