

test

April 10, 2025

```
[1]: import numpy as np
import torch
```

```
[14]: # NumPy

a = np.array([3, 4, 5])
b = np.array([6, 0, -1])
c = a + b # [5, 7, 9]
print(c)
```

[9 4 4]

```
[7]: # PyTorch

a = torch.tensor([3, 4, 5])
b = torch.tensor([6, 0, -1])
c = a + b # tensor([5, 7, 9])
print(c)
```

tensor([9, 4, 4])

```
[8]: # matrix = torch.randint(0, 10, (3, 4))
# print(matrix)

matrix = torch.tensor([[5, 0, 3, 2],
                        [7, 7, 6, 3],
                        [1, 9, 5, 6]])
```

```
[10]: result = a @ matrix
print(result)
```

tensor([48, 73, 58, 48])

```
[1]: import torch

# Check if CUDA (GPU support) is available
print("CUDA available:", torch.cuda.is_available())

# Print the current device
```

```

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print("Using device:", device)

# Optional: Get the name of the GPU (if available)
if torch.cuda.is_available():
    print("GPU name:", torch.cuda.get_device_name(0))

```

CUDA available: True

Using device: cuda

GPU name: NVIDIA GeForce RTX 3070

```

[ ]: #Detta är en funktionell version av a * b. Den gör samma sak som elementvis
      ↳multiplikation:
torch.mul(a, b)
# elementvis = a * b

```

```

[ ]: tensor([18,  0, -5])

```

```

[ ]: #Detta är matris-multiplikation men funkar bara för 2D-tensorer. Går ej att
      ↳använda om a eller b är mer än 2D.
torch.mm(a, b)

```

```

[ ]: #Batch matrix multiplication - används när du har flera matriser i en batch.
      ↳Tensorerna måste ha shape (batch_size, n, m).
torch.bmm(a, b)

```

```

[ ]: torch.matmul(a, b)
# a @ b
# Detta är matrisprodukt eller dot product beroende på dimensionerna.
#     Vektor · vektor → skalar
#     Matris · vektor → vektor
#     Matris · matris → matris

```