PyTorch Tensors

Lesson Downloads

Show TranscriptSummarize Video

PyTorch tensors are crucial tools in the world of programming and data science, which work somewhat like building blocks helping to shape and manage data effortlessly. These tensors allow us to deal with data in multiple dimensions, which is especially handy when working with things like images or more complex structures. Getting to know tensors is a step forward in understanding how PyTorch simplifies the processes of deep learning, enabling us to perform intricate numerical computations efficiently.

Technical Terms Explained:

Tensors: Generalized versions of vectors and matrices that can have any number of dimensions (i.e. multi-dimensional arrays). They hold data for processing with operations like addition or multiplication.

Matrix operations: Calculations involving matrices, which are two-dimensional arrays, like adding two matrices together or multiplying them.

Scalar values: Single numbers or quantities that only have magnitude, not direction (for example, the number 7 or 3.14).

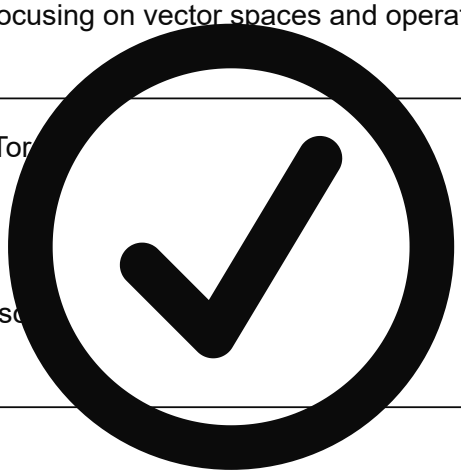
Linear algebra: An area of mathematics focusing on vector spaces and operations that can be performed on vectors and matrices.

Quiz Question

How are matrices related to tensors in PyTor

- Tensors are a type of matrix
- Matrices are a two-dimensional form of tensors
- Matrices are completely unrelated to tens

Submit



Code Examples

Images as PyTorch Tensors

```
import torch

# Create a 3-dimensional tensor
images = torch.rand((4, 28, 28))

# Get the second image
second_image = images[1]
```

Displaying Images

```
import matplotlib.pyplot as plt

plt.imshow(second_image, cmap='gray')
plt.axis('off') # disable axes
plt.show()
```

Matrix Multiplication

```
a = torch.tensor([[1, 1], [1, 0]])

print(a)
# tensor([[1, 1],
```

```
#         [1, 0]])

print(torch.matrix_power(a, 2))
# tensor([[2, 1],
#         [1, 1]])

print(torch.matrix_power(a, 3))
# tensor([[3, 2],
#         [2, 1]])

print(torch.matrix_power(a, 4))
# tensor([[5, 3],
#         [3, 2]])
```

Resources

PyTorch Tensors tutorial

PyTorch Tensors documentation