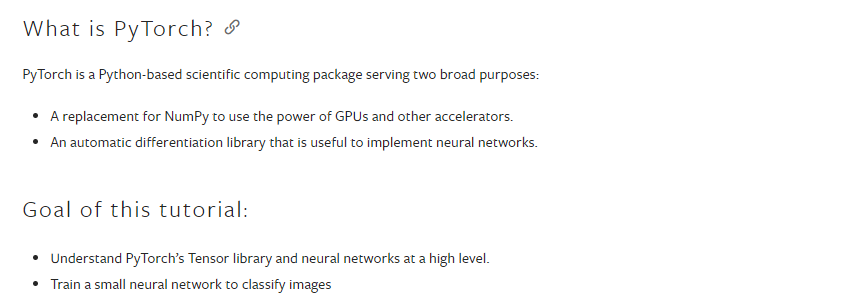
Topic:

Deep Learning with PyTorch: A 60 Minute Blitz ----

Link: <https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html#deep-learning-with-pytorch-a-60-minute-blitz>



In this tutorial we have 4 modules:

1 Basics of PyTorch Tensors

2 Learn about Autograd

3 Train neural networks in pytorch

4 train an image classifier in py torch using CIFAR10 dataset

\*\*BASICS OF PYTORCH TENSORS

===== provided pytorch tutorial..

https://colab.research.google.com/drive/1XmH94BRXUVfpMmLoCl6xKG6nE7BgBMqd?usp=sharing

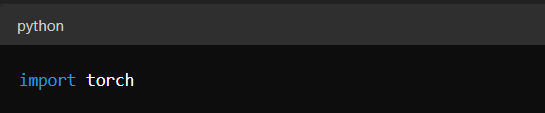
-------------------------------------------------------------------------------------------------------------------------------------------------

Further understanding on pytorch tensors

Step 1: Installing PyTorch

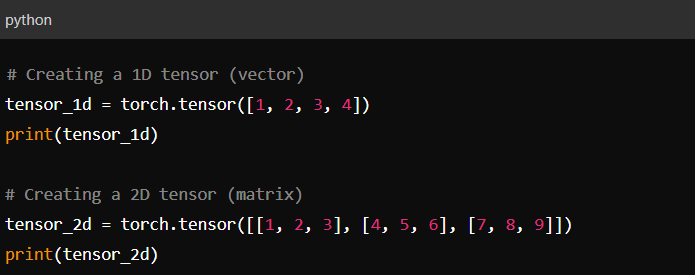


Step 2: Importing PyTorch

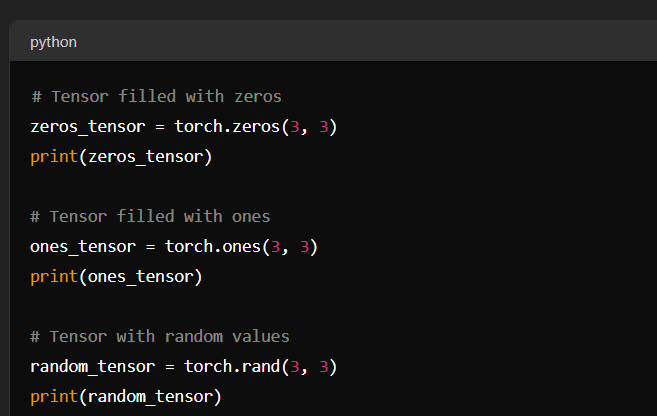


Step 3: Creating Tensors

🡪3.1 Creating a Tensor from a List



🡪3.2 Creating Tensors with Specific Values



O/p: tensor([[0., 0., 0.],

[0., 0., 0.],

[0., 0., 0.]])

O/p: tensor([[1., 1., 1.],

[1., 1., 1.],

[1., 1., 1.]])

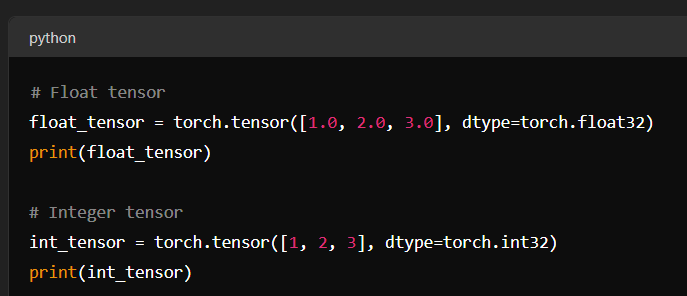
O/p: tensor([[0.5751, 0.4605, 0.8536],

[0.8367, 0.6621, 0.4264],

[0.1332, 0.6057, 0.7026]])

----------------------------------------------------------------------------------------------------------------------------------------------------------

🡪3.3 Creating Tensors with Specific Data Types



o/p: tensor([1., 2., 3.])

tensor([1, 2, 3], dtype=torch.int32)

--------------------------------------------------------------------------------------------------------------------------------------------------------

Step 4: Tensor Operations

PyTorch tensors support various operations similar to NumPy arrays.

🡪Basic Arithmetic Operations

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

b = torch.tensor([4, 5, 6])

# Addition

c = a + b

print(c) ///tensor([5, 7, 9])

# Subtraction

d = a - b

print(d) ///tensor([-3, -3, -3])

# Multiplication

e = a \* b

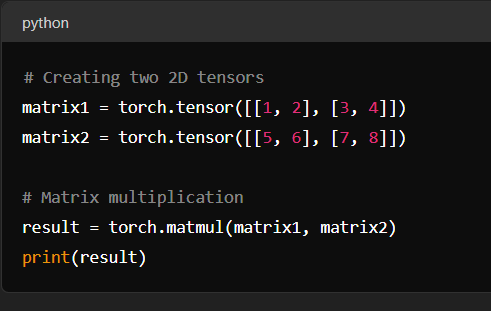
print(e) //tensor([ 4, 10, 18])

# Division

f = a / b

print(f) //tensor([0.2500, 0.4000, 0.5000])

🡪 Matrix Multiplication



o/p: tensor([[19, 22],

[43, 50]])

-------------------------------------------------------------------------------------------------------------------------------------------------------

🡪 In-place Operations

In-place operations modify the tensor in place and are denoted with an underscore suffix.

x = torch.tensor([1, 2, 3])

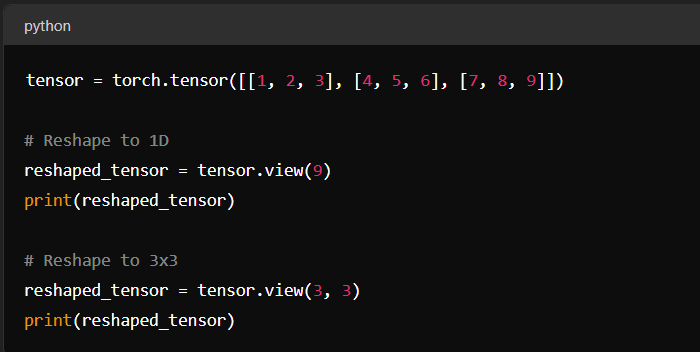
x.add\_(5)

print(x) //tensor([6, 7, 8])

-------------------------------------------------------------------------------------------------------------------------------------------------------

Step 5: Reshaping and Slicing Tensors

🡪Reshaping Tensors



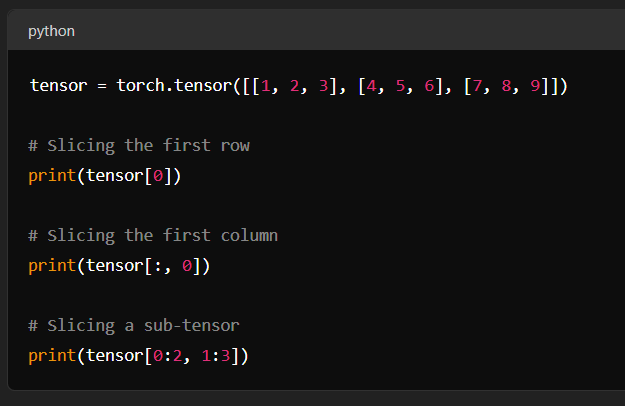
O\p: tensor([1, 2, 3, 4, 5, 6, 7, 8, 9])

tensor([[1, 2, 3],

[4, 5, 6],

[7, 8, 9]])

🡪 Slicing Tensors



o/p: tensor([1, 2, 3])

tensor([1, 4, 7])

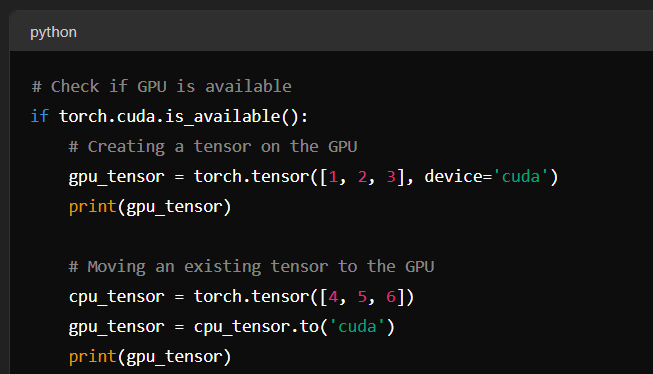
tensor([[2, 3],

[5, 6]])

-------------------------------------------------------------------------------------------------------------------------------------

Step 6: GPU Tensors

PyTorch allows you to move tensors to and perform computations on a GPU.



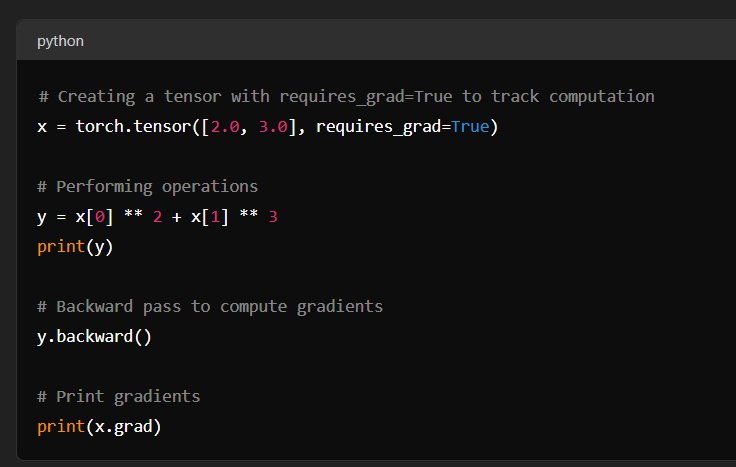
o/p: tensor([1, 2, 3], device='cuda:0')

tensor([4, 5, 6], device='cuda:0')

----------------------------------------------------------------------------------------------------------------------------------------------------

Step 7: Autograd and Automatic Differentiation

PyTorch's `autograd` package provides automatic differentiation for all operations on Tensors.



o/p: tensor(31., grad\_fn=<AddBackward0>)

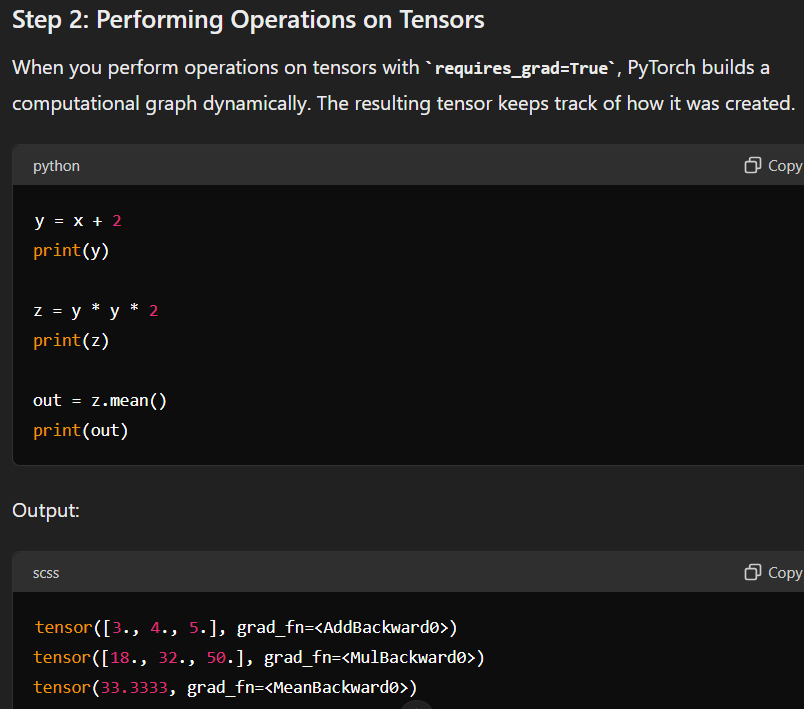
tensor([ 4., 27.])

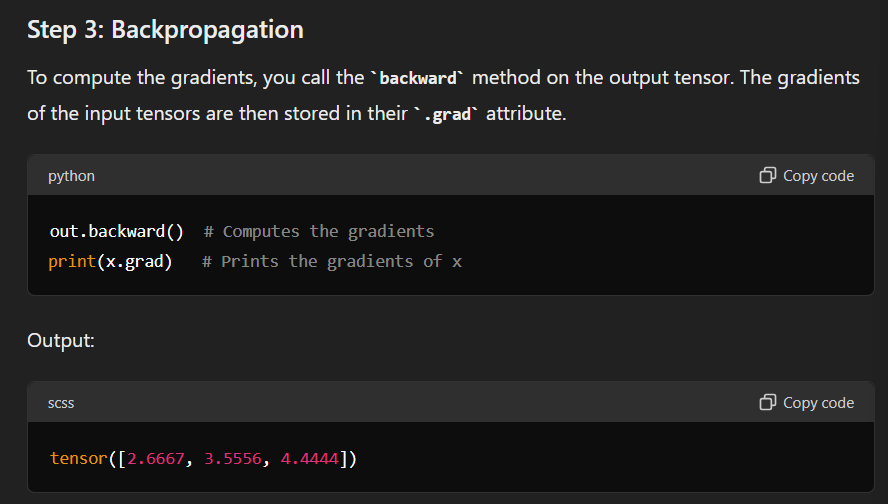
---------------------------------------------------------------------------------------------------------------------------------------------------------

====== Learn about Autograd

<https://colab.research.google.com/drive/1yHuMJiuPzSPrX8WlBkter8ON-frMydEM?usp=sharing>

further details…….





====== Train neural networks in pytorch

<https://colab.research.google.com/drive/1enw1gdcnyrL6BxrQCkXBTfVCpr7MatZ6?usp=sharing>

====== train an image classifier in py torch using CIFAR10 dataset

<https://colab.research.google.com/drive/1ihI99EIECKN7GP3wXyNaehv8TagvgTjb?usp=sharing>