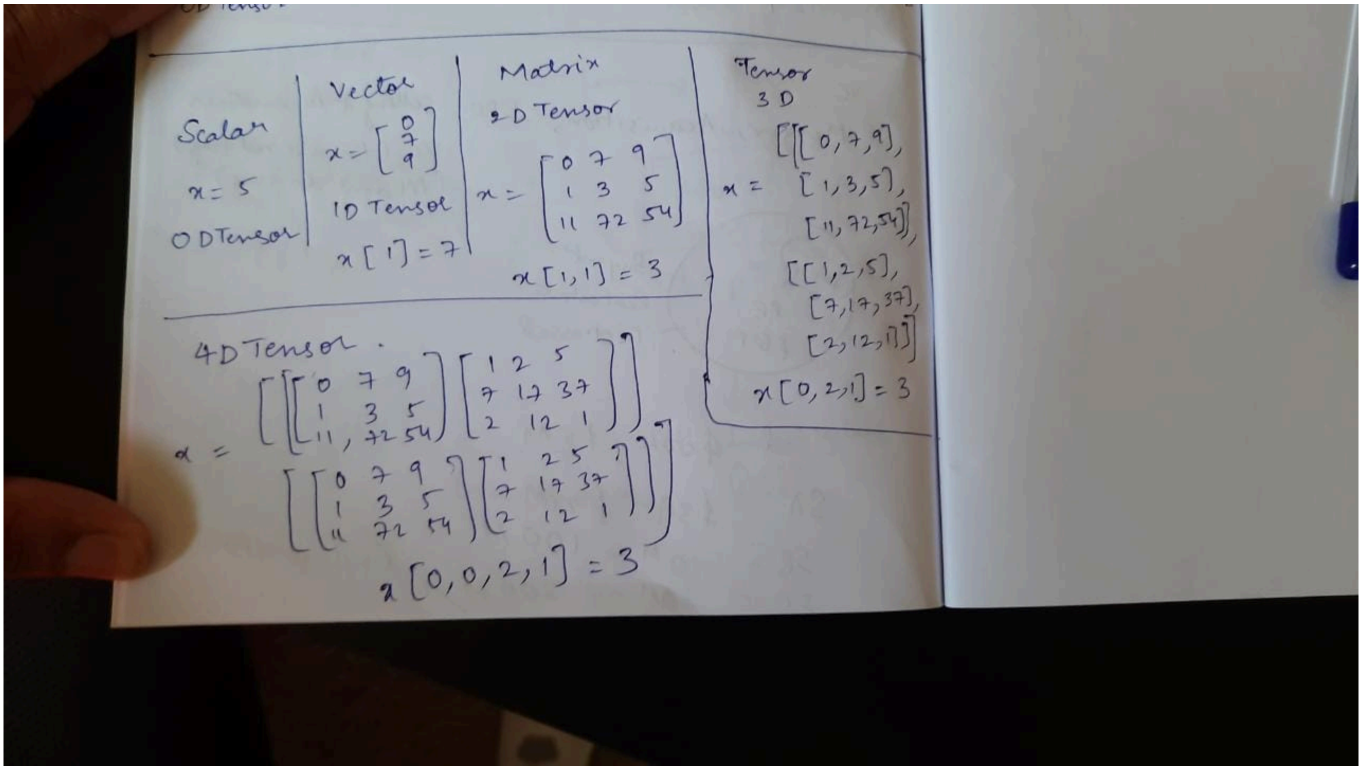
Lec 05 Submission



Scalar

$x = 5$

0 D Tensor

Vector

$x = \begin{bmatrix} 0 \\ 7 \\ 9 \end{bmatrix}$

1D Tensor

$x[1] = 7$

Matrix

2 D Tensor

$x = \begin{bmatrix} 0 & 7 & 9 \\ 1 & 3 & 5 \\ 11 & 72 & 54 \end{bmatrix}$

$x[1,1] = 3$

Tensor

3 D

$x = [[[0,7,9],$
$[1,3,5],$
$[11,72,54]],$
$[[1,2,5],$
$[7,17,37],$
$[2,12,1]]]$

$x[0,2,1] = 3$

4D Tensor.

$$x = \left[\left[\begin{bmatrix} 0 & 7 & 9 \\ 1 & 3 & 5 \\ 11, & 72 & 54 \end{bmatrix} \begin{bmatrix} 1 & 2 & 5 \\ 7 & 17 & 37 \\ 2 & 12 & 1 \end{bmatrix}\right] \left[\begin{bmatrix} 0 & 7 & 9 \\ 1 & 3 & 5 \\ 11 & 72 & 54 \end{bmatrix} \begin{bmatrix} 1 & 2 & 5 \\ 7 & 17 & 37 \\ 2 & 12 & 1 \end{bmatrix}\right]\right]$$

$x[0,0,2,1] = 3$

2. Create the above examples in PyTorch. Make sure that your examples are FloatTensors. Print the data type for each example to make sure they are of desired FloatTensor type. For every tensor, print the shape of the tensor to convince yourself that the tensor is of a desired rank.

```
import torch

# Scalar

x = torch.tensor(5, dtype=torch.float32)

print(x.dtype)
```

```
torch.float32
```

```
# Vector

v = torch.tensor([0, 7 , 9], dtype=torch.float32)

print(v.dtype)
```

```
torch.float32
```

```
# Matrix

m = torch.tensor([[0, 7 , 9],[1, 3, 5], [11,72,54]], dtype=torch.float32)
print(m.dtype)
```

```
torch.float32
```

```
# Tensor
```

```
m = torch.tensor([[[0, 7 , 9],[1, 3, 5], [11,72,54]],
                  [[1,2, 5],[7, 17, 37], [2,12,1]]], dtype=torch.float32)

print(m.dtype)
```

```
torch.float32
```

```
# 4d - Tensor

m = torch.tensor([[[[0, 7 , 9],[1, 3, 5], [11,72,54]],
                  [[1,2, 5],[7, 17, 37], [2,12,1]]],
                  [[[0, 7 , 9],[1, 3, 5], [11,72,54]],
                  [[1,2, 5],[7, 17, 37], [2,12,1]]]
                  ], dtype=torch.float32)

print(m.dtype)
```

```
torch.float32
```

3. Create a version of the above tensors that are IntTensors. Print the data type to ensure that each example is of the desired IntTensor type.

```
# Scalar

x = torch.tensor(5, dtype=torch.int)
print(x.dtype)
```

```
torch.int32
```

```
# Vector

v = torch.tensor([0, 7 , 9], dtype=torch.int)
print(v.dtype)
```

```
torch.int32
```

```
# Matrix

m = torch.tensor([[0, 7 , 9],[1, 3, 5], [11,72,54]], dtype=torch.int)

print(m.dtype)
```

```
torch.int32
```

```
# Tensor

m = torch.tensor([[[0, 7 , 9],[1, 3, 5], [11,72,54]],
                  [[1,2, 5],[7, 17, 37], [2,12,1]]], dtype=torch.int)

print(m.dtype)
```

```
torch.int32
```

```
# 4d - Tensor

m = torch.tensor([[[[0, 7 , 9],[1, 3, 5], [11,72,54]],
                  [[1,2, 5],[7, 17, 37], [2,12,1]]],
                  [[[0, 7 , 9],[1, 3, 5], [11,72,54]],
                  [[1,2, 5],[7, 17, 37], [2,12,1]]]
                  ], dtype=torch.int)

print(m.dtype)
```

```
torch.int32
```

Create two 1D Tensors of same length in PyTorch with arbitrary values. How do you determine if the angle between the two tensors is < 90 degrees or > 90 degress? Demonstrate this using code.

```
x = torch.tensor([1, 4])
y = torch.tensor([-1, -4])
```

```
sum_val = sum(x * y)
```

```
sum_val
```

```
tensor(-17)
```

```
mag_x = torch.sqrt(torch.sum(torch.square(x)))
mag_y = torch.sqrt(torch.sum(torch.square(y)))
```

```
total_mag = mag_x * mag_y
```

```
total_mag
```

```
tensor(17.)
```

```
total_inc = sum_val / total_mag
```

```
total_inc
```

```
tensor(-1.)
```

```
radians = torch.acos(total_inc)
```

```
theta = radians * 57.295
```

```
theta
```

```
tensor(179.9975)
```

As we can see the angle between these two vectors in 180 degrees

These two are exactly opposite to each other

The way to find the angle between these two vectors is the inverse cos of the dot product over the product of magnitudes of the two vectors

The value we get is in radians we need to multiply it with 57.295 to get the value in degrees by this we can find if the value is less than 90 or not

Start coding or generate with AI.