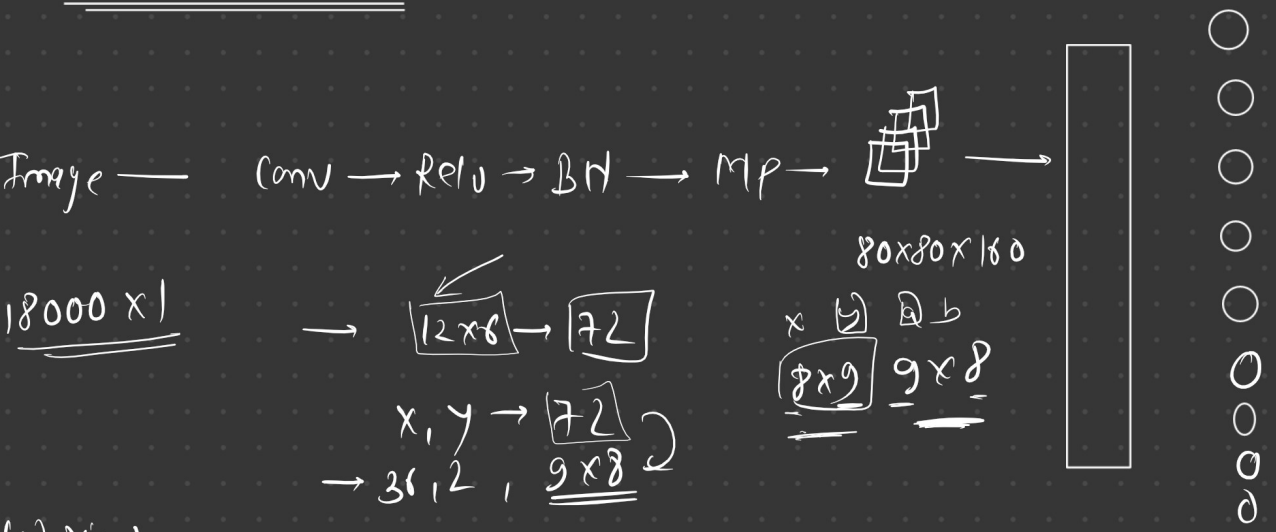
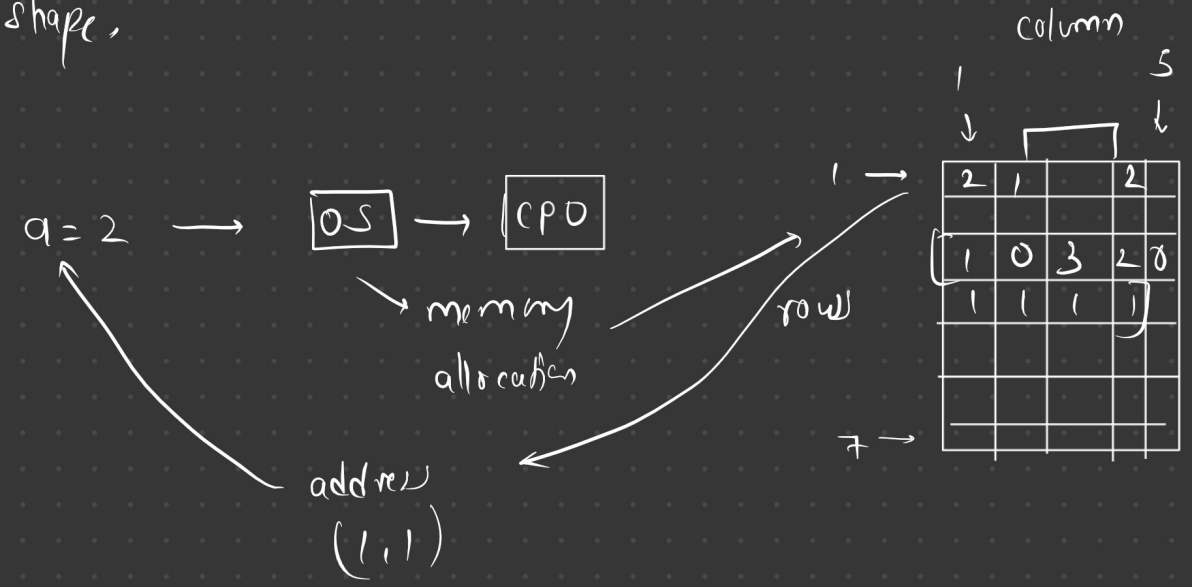
21-12-2024

Agenda - Pytorch I - II



(1) View

Return a new tensor with the same data but with a different shape.



a = [1, 2]

sequential memory → [1 0 3 / 2 0 1 / 1 1 1] → address back storage done is sequential.

→ contiguous memory allocation

[[1 2]
[3, 4]]

Hello		
	world	1
2	3	4

Dim → 2 view → X

↓

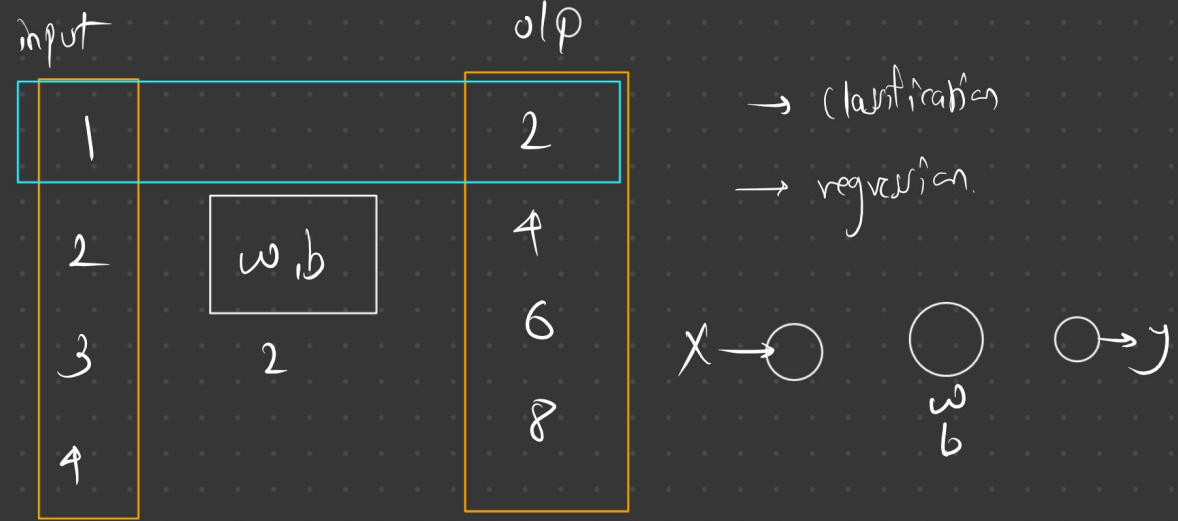
contiguous() → reallocate →

→ contiguous memory allocation

[[1.0541, -0.6352, 1.4429],
[1.5685, 1.4705, 0.1939]] → original matrix → 2x3

dim=0 <= [[[1.0541, -0.6352, 1.4429],
[1.5685, 1.4705, 0.1939]]

[[1.0541, -0.6352, 1.4429],
[1.5685, 1.4705, 0.1939]]]



Class Name (nn.module)

Name → Name of class that inherit nn.module

init → constructor

→ layers & parameter of model

super().init() → This initialize the base class nn.module()

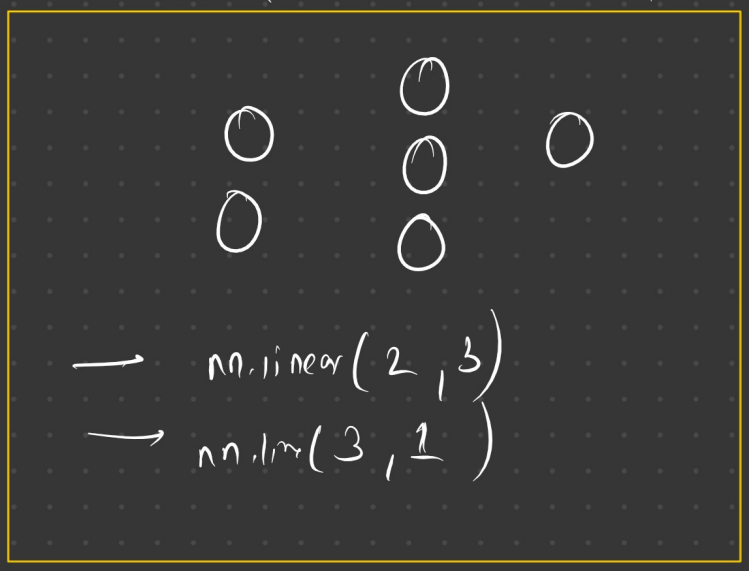
linear → nn.linear, A linear layer that apply a linear transformation.

$y = mx + c$

$y = w \cdot x + b$) same equation.

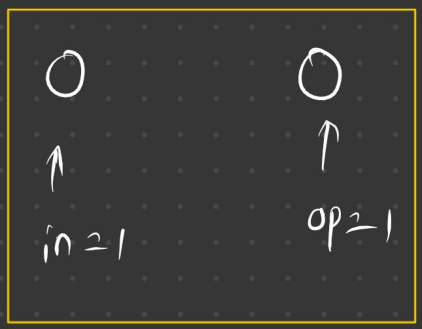
in_feature → input

out_feature → to no. of out we want



→ nn.linear(2, 3)

→ nn.lin(3, 1)



forward → The forward pass method where computation happens

out → self.linear(x) → Apply the linear transformation to the input

out → value → output of linear transformation.