import torch

import torch.nn as nn

import torch.nn.functional as F

class CustomNet(nn.Module):

def \_\_init\_\_(self):

super(CustomNet, self).\_\_init\_\_()

self.layer1 = nn.Linear(1, 3)

self.layer2 = nn.Linear(3, 2)

self.output\_layer = nn.Linear(2, 1)

def forward(self, x):

x = F.relu(self.layer1(x))

x = torch.sigmoid(self.layer2(x))

x = torch.tanh(x)

x = self.output\_layer(x)

return x

x = torch.tensor([[1.0]], requires\_grad=True)

model = CustomNet()

output = model(x)

print("Output:", output.item())

output.backward()

print("Gradient of output wrt input:", x.grad)