#CODE:

#LINEAR REGRESSION SCRATCH USING SOME PYTORCH----------------

import torch

#f = w\*x

X = torch.tensor([1,2,3,4], dtype = torch.float).cuda()

Y = torch.tensor([2,4,6,8], dtype = torch.float).cuda()

#initialise weight w

w=torch.tensor(0.0, dtype = torch.float, requires\_grad=True).cuda()

#print(X)

#model prediction

def forwardPass(x):

    return w\*x

#loss

def loss(y,y\_predicted):

    return ((y\_predicted-y)\*\*2).mean()

#gradient

#dJ/dw = 1/N(2x)(wx-y)

print(f'prediction before training: f(5) = {forwardPass(5):.3f}')

#training

learning\_rate = 0.01

n\_iter = 100

for epoch in range(n\_iter):

    #prediction

    y\_pred = forwardPass(X)

    #loss

    l = loss(Y,y\_pred)

    #gradients

    w.retain\_grad()

    l.backward() #backward pass

    #update weights

    with torch.no\_grad():

        w -= learning\_rate \* w.grad 🡪 IN THIS LINE

    #empty the gradients

    w.grad.zero\_()

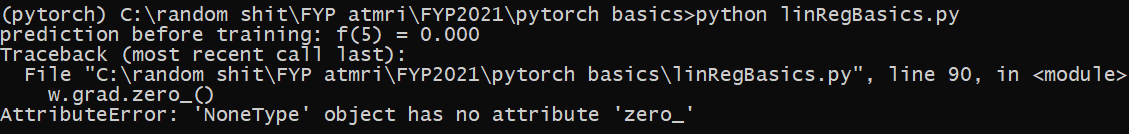
    print(f'epoch {epoch+1}: w = {w:.3f}, loss = {l:.8f}')

print(f'prediction after training: f(5) = {forwardPass(10):.3f}')

IN THE LINE HIGHLIGHTED IN BLUE:

When I write: **w -= learning\_rate \* w.grad,** it works

But when I write **w = w - learning\_rate \* w.grad,** w becomes NoneType and throws the following error:



Can you explain why this happens? As far as I knew the 2 lines have same meaning.

The code is about linear regression with hardcoded values, etc. predicting y=2x function using some pytorch tensors.