

```

import torch
from torch import optim
import matplotlib.pyplot as plt

def linreg_train(X, Y, max_iter=10_000, eta=0.1, print_frequency=500):
    N = len(X)
    a = torch.randn(1, requires_grad=True)
    b = torch.randn(1, requires_grad=True)
    # optimizacijski postupak: gradijentni spust
    optimizer = optim.SGD([a, b], lr=eta)

    print(max_iter, print_frequency)

    for i in range(max_iter):
        # afin regresijski model
        Y_ = a * X + b
        diff = (Y - Y_)
        # kvadratni gubitak
        loss = 1 / N * torch.sum(diff ** 2)
        # računanje gradijenata
        loss.backward()
        # korak optimizacije
        optimizer.step()
        # Postavljanje gradijenata na nulu
        optimizer.zero_grad()
        if i % print_frequency == 0:
            print(f'step: {i}, loss:{loss}, a:{a}, b {b}')

    return a, b

if __name__ == "__main__":
    X = torch.tensor([0.0, 4.0, 2.0, 6.0, 8.0])
    Y = torch.tensor([1.0, 3.1, 1.9, 4.5, 7.0])
    a, b = linreg_train(X, Y, max_iter=10_000, eta=0.001, print_frequency=500)
    X_data = X.numpy()
    Y_data = Y.numpy()
    Y_predicted = (a*X+b).detach().numpy()
    plt.scatter(X_data, Y_data)
    plt.plot(X_data, Y_predicted)
    plt.show()

```