H:\Uni\WiSe 2024\ML\Exercises\2\Task2.py

```python
import numpy as np
import matplotlib.pyplot as plt

epochs = 3
lr = 0.01 # learning rate
x = np.array([[1,1] , [2,1], [3,1]])
y = np.array([3,4,6]).reshape((3,1))
theta = np.zeros(2).reshape((2,1))

losses = []
thetas = []
for i in range(epochs):
    y_predict = x@theta
    loss = np.mean(
        (y_predict - y)**2
    )
    losses.append(loss)
    thetas.append(theta.copy())
    if i == 2:
        break
    grad_theta_0 = (
        2 * np.sum(x.T @ (y_predict - y))
        ) / len(x)
    grad_theta_1 = (
        2 * np.sum((y_predict - y))
        ) / len(x)
    theta[0] = theta[0] - lr*grad_theta_0
    theta[1] = theta[1] - lr*grad_theta_1

thetas = np.array(thetas).reshape(3, 2)
epoch_range = list(range(epochs))

fig, ax = plt.subplots(nrows=2, ncols=1, layout='constrained')
# plot Losses
ax[0].set_xlabel('Epoch')
ax[0].set_ylabel('Loss')
ax[0].plot(epoch_range, losses, ".-", label='plot')
ax[0].legend()
ax[0].set_title("Losses")

# plot Thetas
ax[1].set_xlabel('Epoch')
ax[1].set_ylabel('Thetas')
ax[1].plot(epoch_range, thetas[:,0], ".-r", label='Theta0')
ax[1].plot(epoch_range, thetas[:,1], ".-b", label='Theta1')
ax[1].legend()
ax[1].set_ylim(0, 1)
ax[1].set_title("Thetas")

plt.show()
```