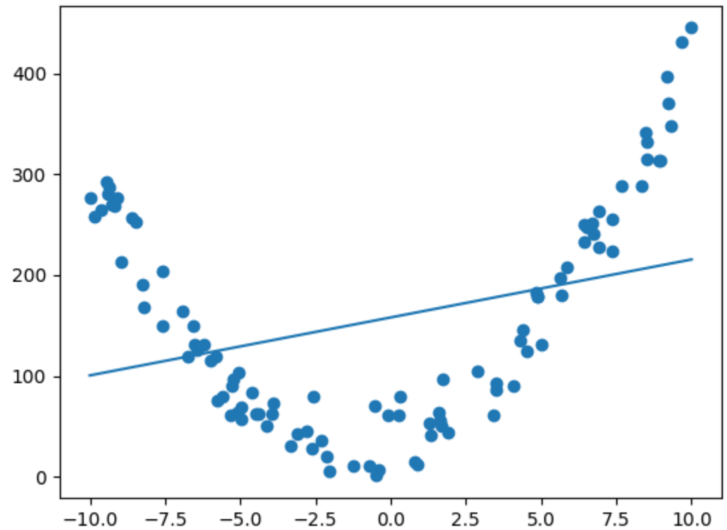
MILESTONE 1: PROTOTYPE USING PYTHON

I Intro: Basic Curve Fitting with Gradient Descent

i Linear Fit

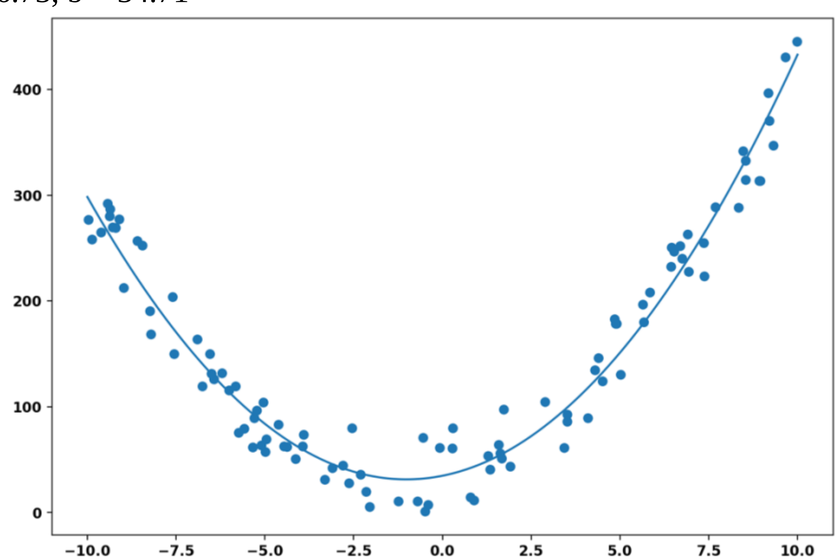
When learning rate = 0.02, epsilon = 0.001, it converged in 313 iterations. The results are shown below.



ii Second Order Fit

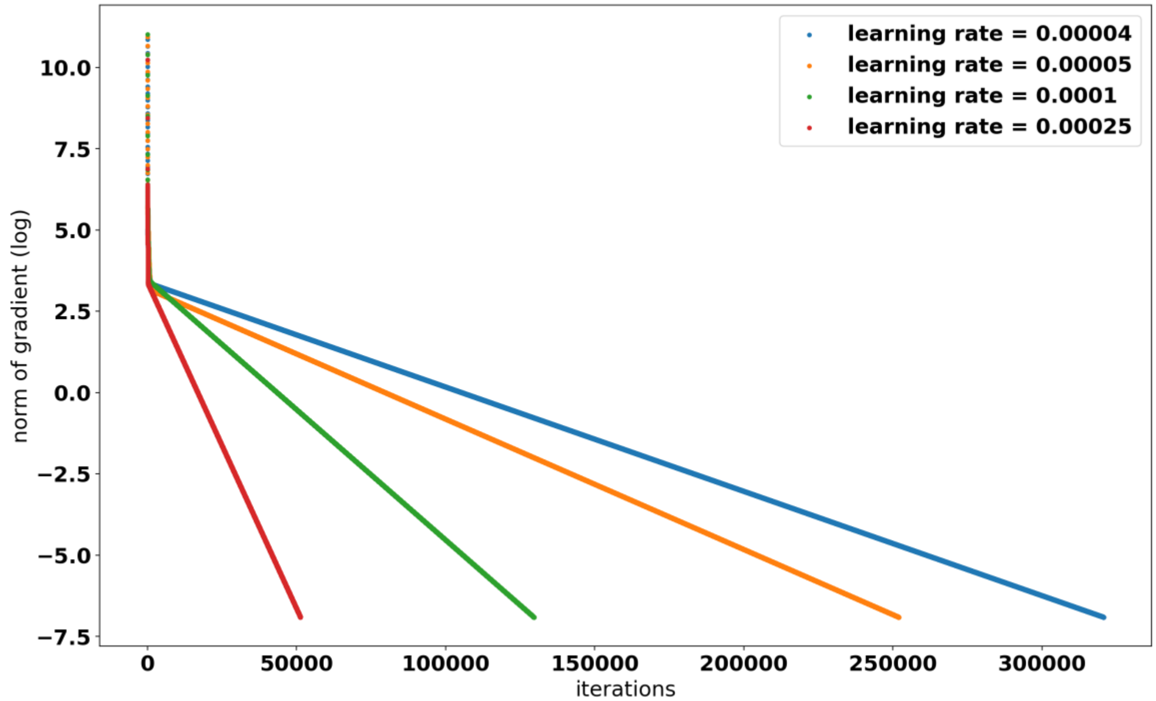
When learning rate = 0.00005, epsilon = 0.001, it converged in 253093 iterations. The results are shown below.

And h = 3.31, m = 6.73, b = 34.71



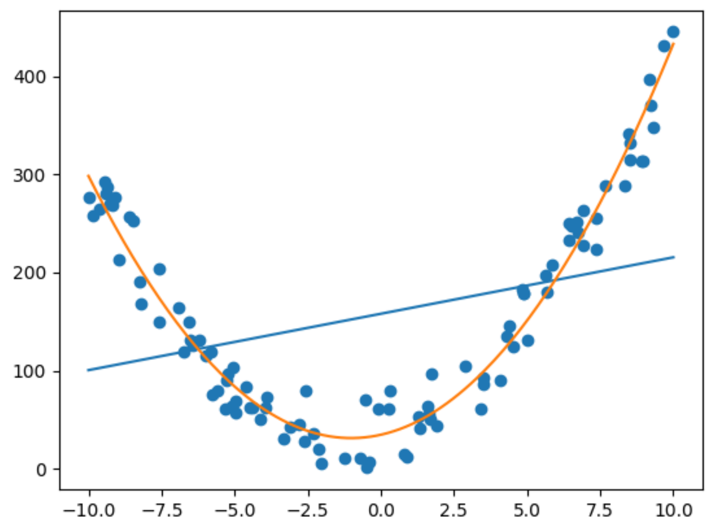
iii Analysis

The norm of the gradient vs. the iteration number for a variety of (convergent) learning rates, since the norm of gradient is too big, I take the log of it.



From the picture above we can see that as the learning rate increase, the norm of gradient drops more quickly.

If I cannot determine the gradient of the error function analytically, I can try to estimate direction of the gradient and then change the parameter by constant value.



II Neural Network

When epoch = 10, batch size = 10, I got the following results, with [784, 10, 10, 10] as the layers number and alpha = 3.0.

Epoch 0: 8246 / 10000	accuracy = 0.8246
Epoch 1: 8402 / 10000	accuracy = 0.8402
Epoch 2: 9272 / 10000	accuracy = 0.9272
Epoch 3: 9318 / 10000	accuracy = 0.9318
Epoch 4: 9414 / 10000	accuracy = 0.9414
Epoch 5: 9395 / 10000	accuracy = 0.9395
Epoch 6: 9398 / 10000	accuracy = 0.9398
Epoch 7: 9451 / 10000	accuracy = 0.9451
Epoch 8: 9425 / 10000	accuracy = 0.9425
Epoch 9: 9479 / 10000	accuracy = 0.9479