

Labwork 1: Gradient descent

1. Introduction

Gradient descent: Gradient descent is an optimization algorithm that is commonly used to train machine learning models and neural networks. It trains machine learning models by minimizing errors between predicted and actual results. usage: minimize the value of error between predicted and actual.

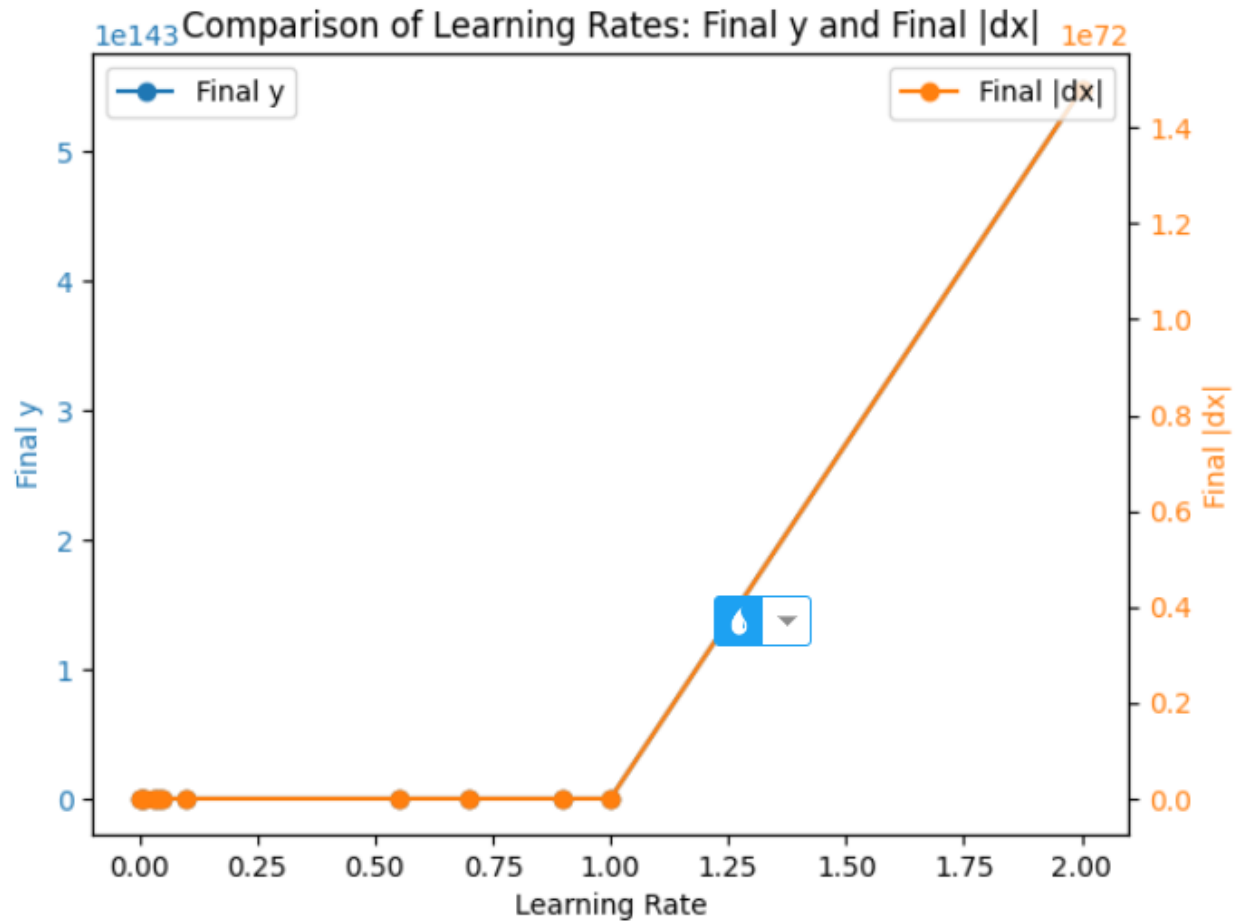
2. Implementation

- First, I create two functions for calculating the assigned function $y = x*x$ and its derivative $y' = 2x$.
- Secondly, Then I calculate the gradient descent based on the functions. gradient descent contains 4 input parameters
 - + init-input: the first input of the model
 - + lr: learning rate, set to try many of learning rates later.
 - + threshold: the threshold for the value of dx , that allows us to stop iteration when we reach expected value.
 - + n-iterations: set limitation for iteration, that doesn't allow too much iteration without convergence.
- Evaluation test with `learning_rate = learning_rates = [0.001, 0.005, 0.01, 0.03, 0.04, 0.05, 0.1, 0.55, 0.7, 0.9, 1, 2]`.

3. Evaluation

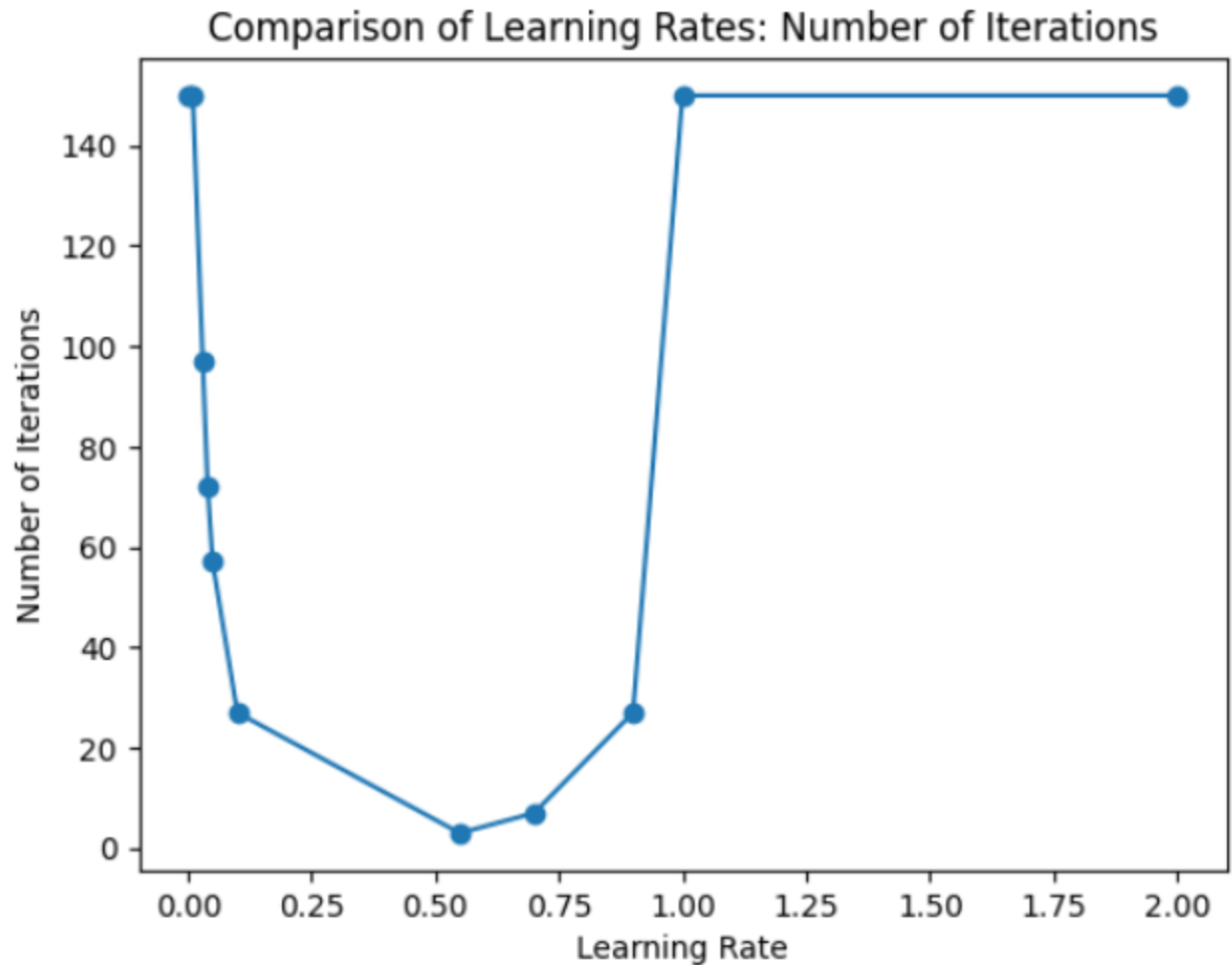
From the output of the program, we can see the effect of each learning rate on the convergence of the gradient descent:

First, we compare the value of y among the different learning rates. This value in my program should be approximately 0, it means that our error should go nearly to 0.



With the smaller value of the learning rate, we have a smaller value of y . It means that gradient descent can not converge to the expected value when learning rate is too big.

The next comparison between learning rates is the number of iterations. The number of iterations reflects the performance of gradient descent with each learning rate. It means that when we set the threshold of error, the smallest number of iterations has the best learning rate in the performance aspect.



We can see with the learning rate of approximately 0.5, we have the least number of iterations to reach an error smaller than the threshold.

Combining 2 above graphs, we can choose the appropriate learning rate for this program (the learning rate should be from 0.5 to 0.6). With this program, we can have the least number of iterations and small error of the model.

4. Conclusion

Gradient descent is a good algorithm to optimize the model, which lead to adjusting the parameters of the model for error reduction. We can choose appropriate learning rate for the model by trying many of values of learning rate, then plot the graph of error and number of iterations to make comparison.