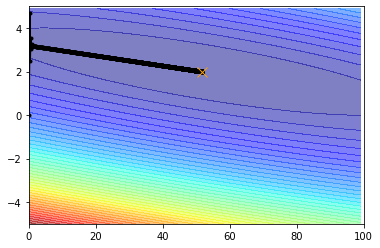
CS-3120

Dr. Feng Jiang

Xiaosong Wang

Feb 9, 2020

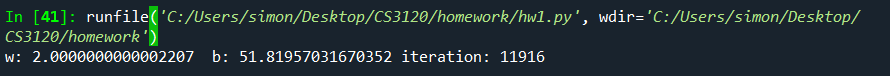
HW1 Report



I terminate the loop by checking the convergence.

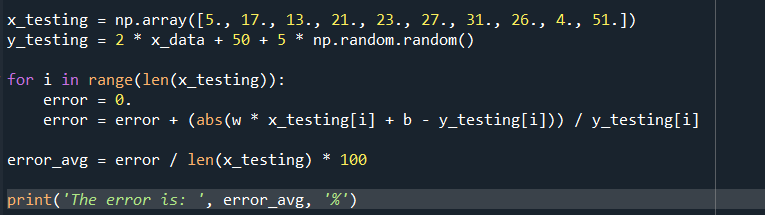


I set both initial w and b to 0, and a maximum iteration time of 50000, and a counter for checking the number of current iteration. In the loop, I first calculate the previous lost (lost\_pre) with the original w and b. Then, update the w and b by gradient descent. After that, I calculate the current lost (lost\_curr). Finally, use current lost subtract previous lost to get the difference between them. When the difference is below 0.00000…000001, there is a break commend to jump out of the loop, and print out the w, b, and the amount of iteration. If the difference does not reach the number I set up, then the loop will end in 50000 times.



Learning rate is about how big step would w and b change during each iteration, and the amount of iteration is how many time we could update w and b. When learning rate is small, the difference between each w and b would be small. If there is no enough iteration, it might not reach the valley point, and that will cause inaccuracy.

Bonus: Testing



I use the same equation to generate another ten pairs of data as testing dataset. In the loop, I first calculate the error for each pair (first calculate the predicted y by using w and b I got, then subtract the testing y data, then divide by testing y data). Finally, calculate the average error (error\_avg) to test if my model is good enough. Then average error I got is about 1%.



Reference:

Most of code I got from the homework instruction and ppt. I read the article *Linear Regression using Gradient Descent* for further understanding. I got the idea of convergence checking from a stack overflow page. The webpage addresses are list below. I did the testing part by myself.

<https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931>

<https://stackoverflow.com/questions/36805834/what-determines-whether-my-python-gradient-descent-algorithm-converges>