Why2

1st Why: **As a result of learning, your approximate function value should move 10% of the way from its original value towards the target value. The before value of the fourth point should be nonzero**

1st Answer: The second point (4.0, 2.0 , -1) and fourth point (4.0, 2.1, -1) are really close to each other. As I mentioned in the why1 file , there are 6 common tilings for both of them. Since the program learned those 6 common tilings in the second point, it is expected to being nonzero for the fourth point.

2nd Why: **The program then continues for 10,000 more examples, printing out the MSE after each 1000, and then finally prints out the function to a file (f10000) for plotting again. You should see the MSE coming down smoothly from about 0.25 to almost 0.01 and staying there (why does it not decrease further towards zero?)**

2nd Answer: We first get 0.25 because the test2() function calculates MSE(10000) without training function, which call learn function that I wrote as well. However, for other MSE(10000) function, which is in the for loop the error gets smaller and smaller, but never become 0. Because the learn function gets close to target function, however, the targetFunction function has normal(0,0.1) in it. This normal distribution never considered in the learned function, therefore the program never be able to fix the error that is caused by this. If we change the normal(0, sth) the “sth” will be in the error as well(which I tested it)