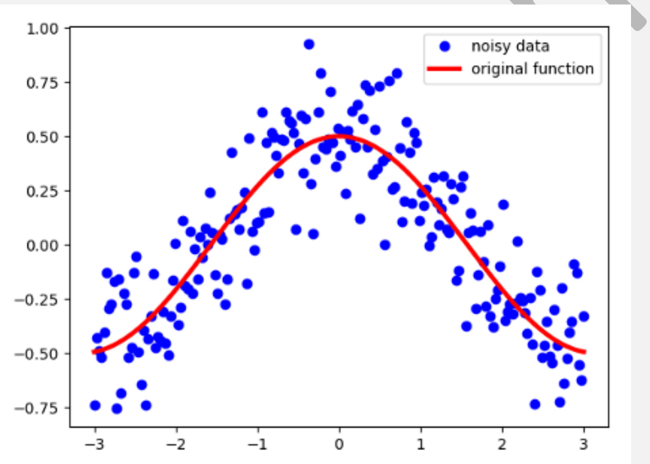
Problem 5

Gradient Descent with momentum

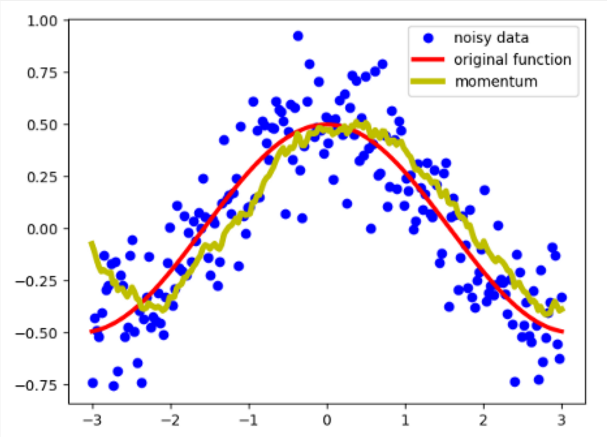
The basic idea of Gradient Descent with momentum is to calculate the exponentially weighted average of your gradients and then use that gradient instead to update your weights. Here we explain Exponentially weighed averages

Exponentially weighed averages:

Exponentially weighed averages deal with sequences of numbers. Suppose, we have some sequence S which is noisy. For this example, I plotted cosine function and added some Gaussian noise. It looks like this:



What we want to do with this data is, instead of using it, we want some kind of “moving average” which would “denoise” the data and bring it closer to the original function. Exponentially weighed averages can give us a pictures which looks like this:



Instead of having data with a lot of noise, we got much smoother line, which is closer to the original function than data we had. Exponentially weighed averages define a new sequence V with the following equation:

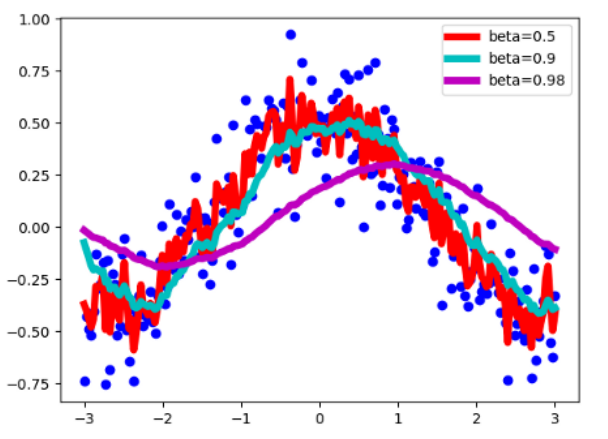
$$V_t = \beta V_{t-1} + (1-\beta)S_t \qquad \beta \, \epsilon \, [0 \; 1]$$

The advantage of this method that is, Gradient descent with momentum will always work much faster than the algorithm Standard Gradient Descent.

Beta is a hyper-parameter which takes values from 0 to one. how the choice of beta affects our new sequence V.it can cause high and low momentum problem.

high and low momentum problem

for explaining this problem which is related to hyper-parameter value, I plotted using different value for β:



As you can see, with smaller numbers of beta, the new sequence turns out to be fluctuating a lot, because we're averaging over smaller number of examples and therefore are 'closer' to the noisy data. it's called **low momentum effect.** With bigger values of beta, like beta=0.98, we get much smother curve, but it's a little bit shifted to the right, because we average over larger number of example (around 50 for beta=0.98). it's called **high momentum effect.**