Stochastic Gradient descent with momentum

Stochastic gradient descent creates some kind of oscillation motion(noise) which increases the time for the descent.

To reduce the noise, we add a correction term which is the average of all previous noises. This is called exponentially weighed averages.

We take v at $t(0) = 0$ and calculate at any iteration t with

$$v(t) := \beta v(t) + (1\text{-}\beta)\partial C/\partial w$$

$\beta \epsilon (o, 1)$ Generally, it is taken to be 0.9

$$w := w - \alpha v(t)$$

Here α is learning rate.

Here we can consider the cost function as a bowl and we want to reach the bottom. The $\partial C/\partial w$ term is like the acceleration and $\beta v(t)$ is like friction term which reduces the motion in the direction of acceleration.

Why does adding momentum work?

With Stochastic Gradient Descent we don't compute the exact derivate of our loss function. Instead, we're estimating it on a small batch. Which means we're not always going in the optimal direction, because our derivatives are 'noisy'. So, exponentially weighed averages can provide us a better estimate which is closer to the actual derivate than our noisy calculations. This is one reason why momentum might work better than classic SGD.

https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d

https://www.youtube.com/watch?v=k8fTYJPd3_I

https://towardsdatascience.com/stochastic-gradient-descent-momentum-explanation-8548a1cd264e