Обновление стохастического градиентного спуска (**SGD**) на каждой итерации t:

$$\theta_{t+1} = \theta_t - \eta g$$

где:

- θ – параметр, который алгоритм будет изменять для достижения приемлемых потерь;
- g – градиент, который показывает противоположное направление и насколько сильно требуется менять параметры, чтобы минимизировать потери;
- η – скорость обучения – то, насколько мы изменяем наши параметры по отношению к градиенту.

SGD может быть медленным, например, когда градиент постоянно мал. Это связано с правилом обновления алгоритма, которое на каждой итерации зависит только от градиентов. 'Шум' при подсчете градиента также может быть проблемой, поскольку стохастический градиентный спуск часто будет следовать неправильному градиенту.

При обучении нейронных сетей может быть использован градиентный спуск на основе импульса (**Momentum**) для борьбы с этими проблемами и ускорения обучения по сравнению с оригинальным **SGD**. Импульс учитывает предыдущие градиенты в правиле обновления на каждой итерации.

$$v_{t+1} = \alpha v_t - \eta g \tag{1}$$
$$\theta_{t+1} = \theta_t + v_{t+1} \tag{2}$$

где:

- v – направление и скорость, с которой параметр должен быть изменен
- α – гиперпараметр затухания, который определяет, как быстро будут затухать накопленные градиенты

Если α намного больше η, накопленные градиенты будут доминировать в правиле обновления, поэтому градиент не изменит направление слишком быстро. Это хорошо в условиях, когда градиент зашумлен, потому что градиент навсегда останется в истинном направлении. С другой стороны, если α намного меньше η, накопленные градиенты могут действовать как фактор сглаживания градиента.

Другим методом, тесно связанным с градиентным спуском на основе импульса, является ускоренный градиентный спуск Нестерова (**NAG**). Разница между **Momentum** и **NAG** заключается в фазе вычисления градиента. В методе импульса градиент был вычислен с использованием текущих параметров θt:

$$g = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \mathcal{L} \left(x^{(i)}, y^{(i)}, \theta_t \right) \tag{3}$$

в то время как в **NAG** мы применяем скорость vt к параметрам θ для вычисления промежуточных параметров θ. Затем мы вычисляем градиент, используя промежуточные параметры

$$\begin{aligned} \tilde{\theta} &= \theta_t + \alpha v_t \\ g_{NAG} &= \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \mathcal{L} \left(x^{(i)}, y^{(i)}, \tilde{\theta} \right) \\ v_{t+1} &= \alpha v_t - \eta g \\ \theta_{t+1} &= \theta_t + v_{t+1} \end{aligned} \tag{4}$$

Мы можем рассматривать **NAG** (и в принципе параметр **accelerate**) как поправочный коэффициент для **Momentum**. Рассмотрим случай, когда скорость, добавленная к параметрам, приводит к немедленным нежелательным высоким потерям, например, в случае взрывающегося градиента. В этом случае **Momentum** может быть очень медленным, поскольку выбранный путь оптимизации демонстрирует большие колебания. В случае **NAG** вы можете просмотреть его как просмотр промежуточных параметров, где добавленная скорость приведет к параметрам. Если обновление скорости приводит к серьезным потерям, то градиенты направят обновление обратно в сторону θt.

Когда скорость обучения η относительно велика, **NAG** допускают большую скорость затухания α, чем **Momentum**, предотвращая при этом колебания.