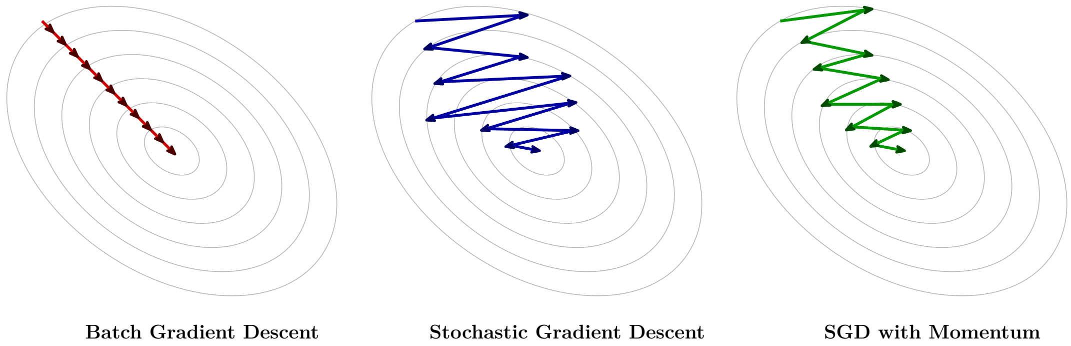
The parameter α controls the **contribution of the previous velocity** in the current parameter update. If α is greater than η, the current update is more influenced by the previous gradients, thus increasing **convergence stability** and helping the network to **avoid local minima**.

Stochastic Gradient Descent with Momentum, while accelerating the process of reaching the minimum, can sometimes "overshoot" the desired target due to the inertia effect introduced by speed. However, it manages to reach the minimum much **faster than simple SGD**, making it an effective choice for optimising model parameters when training neural networks.



Batch Gradient Descent **Stochastic Gradient Descent** **SGD with Momentum**

BGD vs SGD vs SGD with Momentum

3.1.5. Stochastic Gradient Descent with Nesterov Momentum

Stochastic Gradient Descent with Nesterov Momentum is a variant of the previous algorithm that adds a correction to the previous momentum before calculating the gradient. It works by firstly taking a step in the direction of the accumulated gradient (point 2 of the algorithm), and secondly calculating the gradient (point 4) and making the correction accordingly (point 5). (in simple momentum, we first compute the gradient and then make the correction)

Algorithm **Stochastic Gradient Descent with Nesterov Momentum**

```
Require: Learning Rate $\eta$
Require: Momentum Parameter $\alpha$
Require: Initial Parameters $\mathbf{w}$
Require: Initial Velocity $\mathbf{v}$
1: while Stopping Criteria not met do
2:     Update parameters temporarily: $\widetilde{\mathbf{w}} \leftarrow \mathbf{w} + \alpha\mathbf{v}$
3:     Compute gradient estimate on a random training example $(\mathbf{x}^{(i)}, y^{(i)})$
4:     $\widehat{\mathbf{g}} \leftarrow \nabla_{\widetilde{\mathbf{w}}} L(f(\mathbf{x}^{(i)}, \widetilde{\mathbf{w}}), y^{(i)})$
5:     Update velocity: $\mathbf{v} \leftarrow \alpha\mathbf{v} - \eta\widehat{\mathbf{g}}$
6:     Update parameters: $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{v}$
7: end while
```

The main difference between the Nesterov momentum and the standard momentum lies in **where the gradient is evaluated**. In the Nesterov momentum the gradient term is not calculated from the current position in parameter space, but rather from an intermediate position. This approach offers a significant advantage: while the gradient term always points in the optimal direction, **the momentum term may not align consistently**. Consequently, if the momentum goes in the wrong direction or overshoots the target, the gradient term can still "go back" and correct it **in the same update step**.