NAG

NAG, which stands for Nesterov Accelerated Gradient (or Nesterov Momentum), is an optimization algorithm that builds upon momentum-based gradient descent. It aims to further improve convergence speed and potentially alleviate issues like getting stuck in shallow valleys.

Steps:

1. **Initialize Parameters**: Start with initial values for the parameters you want to optimize.

2. **Set Hyperparameters**

3. **Initialize Velocity**: Set the initial velocity (v_0) to zero or a small random value.

4. **Calculate Gradient:**
   - **Formula:** ∇J(θ) (nabla f of theta)

5. **Look-ahead Update:**
   - **Formula:**

$$v_t = \beta * v_(t-1) + \alpha * \nabla J(\theta_t)$$

6. **Parameter Update with Look-ahead:**
   - **Formula:**

$$\theta_t + 1 = \theta_t - \alpha * \nabla f(\theta_t + \beta * v_(t-1))$$

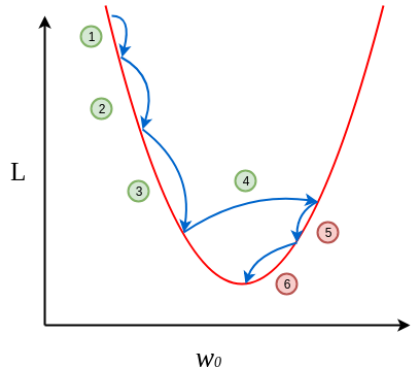
Benefits of NAG:

- Potentially faster convergence compared to standard momentum.
- May be more effective in overcoming shallow valleys in the loss function.

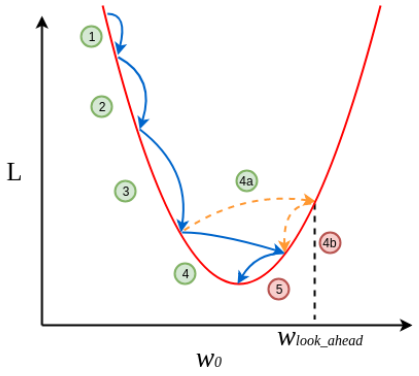
Drawbacks of NAG:

- Requires tuning the same hyperparameters (lr and beta) as momentum.
- Convergence guarantees are similar to momentum (not guaranteed to reach global minimum in all cases).
- Can be slightly more computationally expensive than momentum due to the additional "look-ahead" step.

NAG optimizer aims to reduce oscillations in momentum during training.



(a) Momentum-Based Gradient Descent



(b) Nesterov Accelerated Gradient Descent

$\frac{\partial L}{\partial w_0} = \frac{Negative(-)}{Positive(+)}$

$\frac{\partial L}{\partial w_0} = \frac{Negative(-)}{Negative(-)}$