Accelerating Gradient Descent

Zachary Ross

February 15, 2023

In the following writeup, we explore accelerated gradient descent (GD) methods and their convergence. These are adaptations to the general method

$$x_{t+1} = x_t - \eta \nabla f(x_t), \tag{1}$$

where η is the learning rate and $f : \mathbb{R}^n \to \mathbb{R}$ is a *convex* objective function with a unique minimum, which use previous descent directions as part of current iterates computation.

1 Accelerated Methods

The intuition behind using accelerated methods is that it introduces a physics-inspired component to GD which mimics a dampened oscillator ordinary differential equation (ODE). This effect minimizes the oscillations that tend to occur in gradient descent by using previous gradient computations to penalize rapid changes in direction and reward movement towards an optimal point.

We reconstruct this via a method similar to as was done by Gadat et al.: the ODE limit for gradient descent is given by the equation

$$\frac{dx}{dt} = -\nabla f(x). \tag{2}$$

This can be modified to mimic a dampened oscillator ODE by including acceleration and dampening term $\gamma \geq 0$

$$\frac{d^2x}{dt^2} + \gamma \frac{dx}{dt} = -\nabla f(x) \tag{3}$$

and then rearranged to interpret in terms of acceleration, along with a simplification to velocity v of x, i.e. $v = \frac{dx}{dt}$,

$$\frac{dv}{dt} = -\gamma v - \nabla f(x). \tag{4}$$

Conversion to a discrete time step $\sqrt{\eta}$ with a right-derivative yields the equations

$$\frac{dv_t}{dt} \approx \frac{v_{t+1} - v_t}{\sqrt{\eta}}, \quad \text{and} \quad v_t = \frac{dx_t}{dt} \approx \frac{x_{t+1} - x_t}{\sqrt{\eta}} \tag{5}$$

which can be used to derive a formula for the dampened oscillator GD. Let $\beta = 1 - \gamma\sqrt{\eta}$. Then at iteration $t-1$

$$\begin{aligned}
\frac{v_t - v_{t-1}}{\sqrt{\eta}} &= -\gamma v_{t-1} - \nabla f(x_{t-1}) \\
v_t - v_{t-1} &= -\gamma\sqrt{\eta} v_{t-1} - \sqrt{\eta}\nabla f(x_{t-1}) \\
v_t &= \beta v_{t-1} - \sqrt{\eta}\nabla f(x_{t-1}) \\
\frac{x_{t+1} - x_t}{\sqrt{\eta}} &= \beta \frac{x_t - x_{t-1}}{\sqrt{\eta}} - \sqrt{\eta}\nabla f(x_{t-1}) \\
x_{t+1} &= x_t + \beta(x_t - x_{t-1}) - \eta \nabla f(x_{t-1})
\end{aligned} \tag{6}$$

By modifying this equation to evaluate the gradient at x_t rather than x_{t-1}, this yields the *heavyball* method as was discovered by Polyak

$$x_{t+1} = x_t + \beta(x_t - x_{t-1}) - \eta \nabla f(x_t). \tag{7}$$

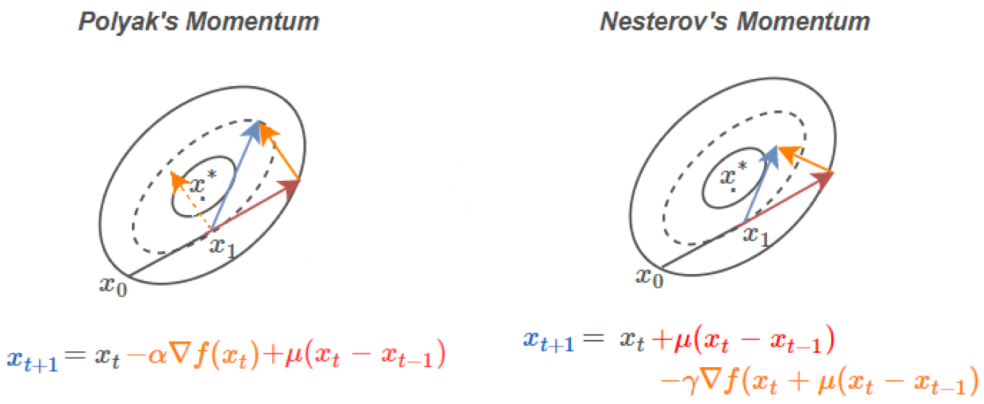

Figure 1: Comparison between Polyaks and Nesterovs momentum. The gradient descent step (orange arrow) is perpendicular to the level set before applying momentum to x_1 (red arrow) in Polyaks algorithm, whereas it is perpendicular to the level set after applying momentum to x_1 in Nesterovs algorithm. Graphic provided by **?** .

Despite the effectiveness of this method, Polyak's still has the potential to oscillate infinitely under specific conditions, as observed by Lessard et al.. This is due to the gradient being evaluated *before* momentum is applied and x_t sometimes serving as a poor approximate for x_{t-1}. Nesterov altered this method slightly by evaluating the gradient *after* momentum is applied. This derivation can be made by reinterpreting Equation 8 as a left derivative with time step $\sqrt{\delta}$ for some $\delta \geq 0$

$$\frac{dv_t}{dt} \to \frac{v_t - v_{t-1}}{\sqrt{\delta}}, \quad \text{and} \quad v_t = \frac{dx_t}{dt} \to \frac{x_t - x_{t-1}}{\sqrt{\delta}} \tag{8}$$

and re-deriving Equation 9 under this condition. Let $\beta = \left(1 + \gamma\sqrt{\delta}\right)^{-1}$ and $\eta = \beta\delta$. Then at iteration $t+1$

$$\begin{aligned}
\frac{v_{t+1} - v_t}{\sqrt{\delta}} &= -\gamma v_{t+1} - \nabla f(x_{t+1}) \\
v_{t+1} - v_t &= -\gamma\sqrt{\delta} v_{t+1} - \sqrt{\delta}\nabla f(x_{t+1}) \\
\beta^{-1} v_{t+1} - v_t &= -\sqrt{\delta}\nabla f(x_{t+1}) \\
\frac{x_{t+1} - x_t}{\beta\sqrt{\delta}} - \frac{x_t - x_{t-1}}{\sqrt{\delta}} &= -\sqrt{\delta}\nabla f(x_{t+1}) \\
\frac{x_{t+1} - x_t}{\beta\sqrt{\delta}} &= \frac{x_t - x_{t-1}}{\sqrt{\delta}} - \sqrt{\delta}\nabla f(x_{t+1}) \\
x_{t+1} - x_t &= \beta(x_t - x_{t-1}) - \eta\nabla f(x_{t+1}) \\
x_{t+1} &= x_t + \beta(x_t - x_{t-1}) - \eta\nabla f(x_{t+1}).
\end{aligned} \tag{9}$$

By evaluating the gradient at $x_t + \beta(x_t - x_{t-1})$ rather than x_{t+1}, this then yields Nesterov's accelerated method.

$$x_{t+1} = x_t + \beta(x_t - x_{t-1}) - \eta\nabla f(x_t + \beta(x_t - x_{t-1})). \tag{10}$$

The difference between these methods is shown in Figure 1.

2 First Order Methods

This section discusses methods of the form

$$x_{t+1} = x_t - \eta H_t^{-1}\nabla f(x_t) \tag{11}$$

where H_t is a first-order hessian approximation. Hessian approximation methods make use of the geometry of the data to approximate the Fisher information matrix, which is itself a hessian approximation, given by

$$I(x) = \mathbb{E}_x \left[\nabla f(x) \nabla f(x)^\intercal \right] \tag{12}$$

to find a solution to the root-finding problem

$$f(x^*) \approx f(x_t) + \nabla f(x_t)^\intercal (x^* - x_t) + \frac{1}{2}(x^* - x_t)^\intercal H_t^{-1} (x^* - x_t). \tag{13}$$

First order methods utilize this by approximating the diagonal of this matrix via

$$I(x) \approx \operatorname{diag} \left\{ \mathbb{E}_x \left[\nabla f(x) \odot \nabla f(x) \right] \right\}. \tag{14}$$

rather than using expensive computations in Equation 12.

? provide a method which combines earlier methods AdaGrad and RMSProp to build *Adam*, which is theoretically defined for some approximate g_t s.t. $\mathbb{E}[g_t] = \nabla f(x_t)$ using

$$x_{t+1} = x_t - \eta \frac{\mathbb{E}\left[g_t\right]}{\sqrt{\mathbb{E}\left[{g_t}^2\right]}}. \tag{15}$$

This method can be seen as minimizing a *Signal-to-Noise ratio*, as it uses the gradient's first moment (mean) divided by the gradient's square root of the second moment (un-adjusted standard deviation) as a descent direction. As this value reaches an optima, the mean will tend to decline and become overcome by the noise.

Both $\mathbb{E}\left[g_t\right]$ and $\mathbb{E}\left[{g_t}^2\right]$ are approximated using exponential moving averages, where if $g_t \sim \rho(g_t)$ is the gradient distribution and g_t is selected s.t. $\mathbb{E}[g_t] = \nabla f(x_t)$ then

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \text{ and} \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \tag{16}$$

where m_0 and v_0 are zero-value initialized. Due to this initialization, both terms require bias correction, as we can see by taking the expectation of the closed form solution to v_t (the same applies to m_t)

$$\begin{aligned} \mathbb{E}\left[v_t\right] &= \mathbb{E}\left[(1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} g_i^2 \right] \\ &= \mathbb{E}\left[g_t^2\right] (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} + \xi \\ &= \mathbb{E}\left[g_t^2\right] (1 - \beta_2^t) + \xi. \end{aligned} \tag{17}$$

The bias correction is then employed by including the bias adjustment term to each equation

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \text{and} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{18}$$

which is substituted in the gradient descent equation to get

$$x_{t+1} = x_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{19}$$

where $\epsilon \ll 1$ is used to ensure the denominator never reaches 0. Note the use of ξ in Equation 17. Although this means there will never be 0 bias, the used of a β large enough will ensure that this value is sufficiently small in most cases.

3 Convergence Proofs

We begin by exploring the general convergence proof for Equation 1. This requires a few definitions prior to analysis, mainly be defining convexity from a theoretical point of view, then adding upper and lower bounds to this definition which will aid in these proofs.

Definition 1. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if for all $x, y \in \mathbb{R}^n$, for all $\lambda \in [0, 1]$,

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(\lambda x + (1 - \lambda) y). \tag{20}$$

This definition provides an upper bound for a convex function over any two points. Likewise, we have that a convex function can be lower-bounded at every point by a hyperplane which runs tangential to the function.

Lemma 1. *If a function f is convex, then for all $x, y \in \mathbb{R}^n$,*

$$f(y) \geq f(x) + \nabla f(x)^\intercal (y - x) \tag{21}$$

For these convergence proofs, convex functions need be constrained further by their upper and lower bounds. This is where the next two definitions come to play.

Definition 2. A continuously differentiable function f is β-smooth if its gradient is β-Lipschitz

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta \|x - y\|. \tag{22}$$

Definition 3. A function f is α-strongly convex if for $\alpha > 0$ and $x \in \mathbb{R}^n$

$$f(x) - \frac{\alpha}{2} \|x\|^2 \tag{23}$$

is convex.

References

[1] Sébastien Gadat, Fabien Panloup, and Sofiane Saadane. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461 – 529, 2018. doi: 10.1214/18-EJS1395. URL `https://doi.org/10.1214/18-EJS1395`.

[2] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[3] Yurii Evgen'evich Nesterov. A method of solving a convex programming problem with convergence rate o\bigl(k^2\bigr). In *Doklady Akademii Nauk*, volume 269, pages 543–547. Russian Academy of Sciences, 1983.

[4] Boris Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4:1–17, 12 1964. doi: 10.1016/0041-5553(64)90137-5.