On the momentum term in gradient descent learning algorithms

Daniel Richards Ravi Arputharaj , Adhithyan Kalaivanan

KTH

Formulation

A cornerstone of deep learning is non-convex optimization, and gradient descent is often the preferred method. Conventionally, a "momentum" term is added in the parameter update, to improve convergence rate and prevent being trapped in local minima.

$$\Delta w_t = -\epsilon \nabla_w \mathbf{L}(w) + p \Delta w_{t-1} \tag{1}$$

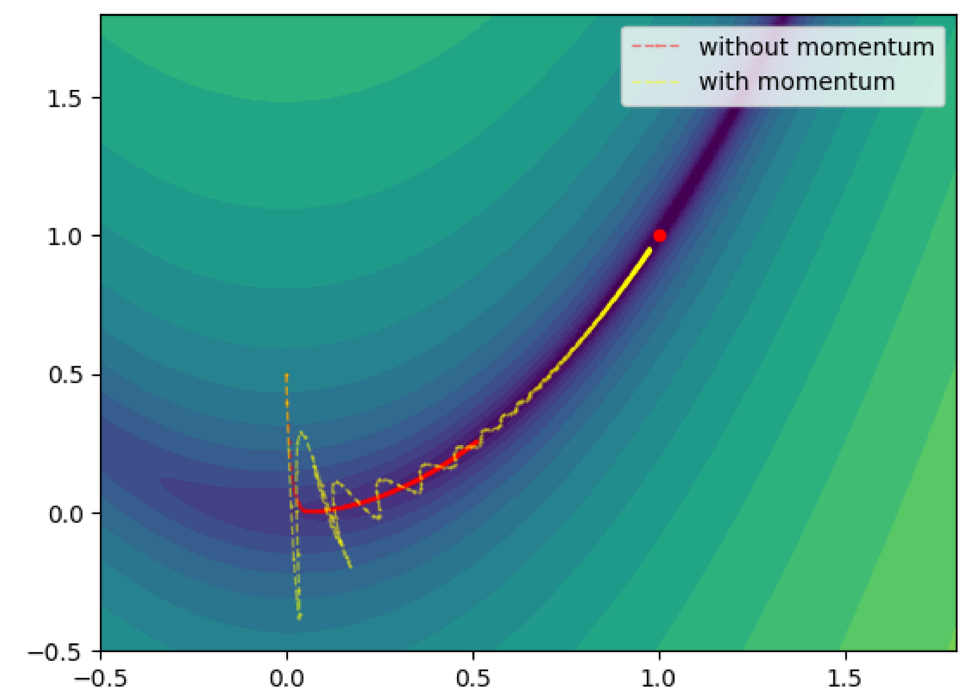

Figure 1. Effect of using momentum

An intuitive visual analogy is a ball rolling down the landscape of loss. Its inertia serves as an accelerator, helping speed of convergence.

Physical Analogy

Consider a point mass with mass m in a conservative force field with potential $\mathcal{E}(\mathbf{w})$ and friction coefficient μ [1]. The system is described by,

$$m\frac{d^2\mathbf{w}}{dt^2} + \mu\frac{d\mathbf{w}}{dt} = -\nabla_w \mathcal{E}(\mathbf{w}) \tag{2}$$

where $\mathbf{w}$ is the coordinate vector of the particle. Discretizing and re-arranging,

$$\mathbf{w}_{t+\Delta t} - \mathbf{w}_t = -\frac{(\Delta t)^2}{m+\mu\Delta t}\nabla_w \mathcal{E}(\mathbf{w}) + \frac{m}{m+\mu\Delta t}(w_t - w_{t-\Delta t}) \tag{3}$$

Comparing with equation 1, we see momentum term does act as inertia

$$\epsilon = \frac{(\Delta t)^2}{m+\mu\Delta t} \tag{4}$$

$$p = \frac{m}{m+\mu\Delta t} \tag{5}$$

Close to minima $_0$, equation 2 is approximated by,

$$m\frac{d^2 w}{dt^2} + \mu\frac{dw}{dt} = -\mathbf{H}(\mathbf{w} - \mathbf{w}_0) \tag{6}$$

where $\mathbf{H}$ is the Hessian matrix. As it is p.s.d. $\mathbf{H} = QKQ^T$, with $K = diag(k_i|i = 1,2,3....n)$, where n is number of weights. Setting $\mathbf{w}' = Q^T\mathbf{w}$ then,

$$m\frac{d^2 w_i'}{dt^2} + \mu\frac{dw_i'}{dt} = -k_i w_i' \tag{7}$$

which resembles a set of uncoupled damped harmonic oscillators.

Analysis

The general solution to equation 7 is given by

$$w_i'(t) = c_1 e^{\lambda_{i,1}t} + c_2 e^{\lambda_{i,2}t}$$
$$\lambda_{i,1,2} = -\frac{\mu}{2m} \pm \sqrt{\frac{\mu}{m}\left(\frac{\mu}{4m} - \frac{k_i}{\mu}\right)} \tag{8}$$

If there were no momentum, the solution is simply

$$w_i'(t) = c_0 e^{\lambda_{i,0}t}, \lambda_{i,0} = -k_i/\mu \tag{9}$$

Observations:

- Convergence guaranteed: $Re(\lambda_{i,2}) \leq Re(\lambda_{i,1}) < 0$
- Momentum speed-up: $|Re(\lambda_{i,1})| > Re(|\lambda_{i,0}|)$, if and only if $k_i < \mu^2/2m$
- Setting $|Re(\lambda_{i,1})| := \alpha|Re(\lambda_{i,0})|$,

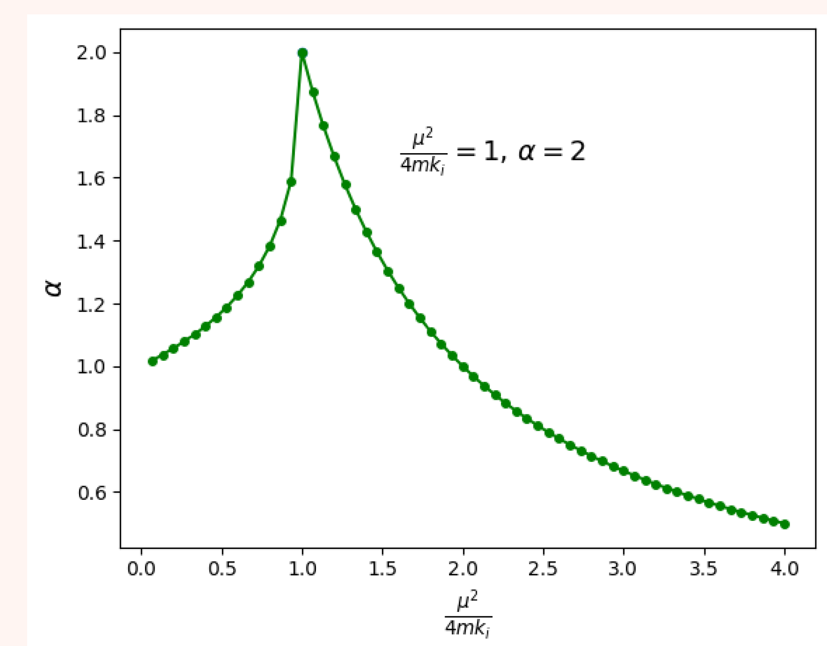

Figure 2. Speed-up provided by momentum

so momentum is most effective when $k_i = \mu^2/4m$, where speed up is doubled

Discrete Case

On discretization and approximating the gradient using the Hessian matrix,

$$w_{i,t+1}' = [(1+p)\mathcal{I} - \epsilon k_i]w_{i,t}' - pw_{i,t-1}' \tag{10}$$

Adding a dummy equation $w_{i,t}' = w_{i,t}'$, we get the following

$$\begin{pmatrix} w_{i,t}' \\ w_{i,t+1}' \end{pmatrix} = A^t \begin{pmatrix} w_{i,0}' \\ w_{i,1}' \end{pmatrix} \tag{11}$$

where $A = \begin{pmatrix} 0 & 1 \\ -p & 1+p-\epsilon k_i \end{pmatrix}$, whose eigen values are given by

$$\lambda_{i,1,2} = \frac{1+p-\epsilon k_i \pm \sqrt{(1+p-\epsilon k_i)^2 - 4p}}{2} \tag{12}$$

Observations:

- For convergence, $\max(|\lambda_{i,1}|, |\lambda_{i,2}|) < 1$ which happens if and only if $-1 < p < 1$ and $0 < \epsilon k_i < 2 + 2p$
- Without $p = 0$, we have $0 < \epsilon k_i < 2$, thus momentum extends range of allowed epsilon, nearly doubles it.
- Compared to the continuous case, discrete system is guarenteed to converge only for some ϵ, k_i and p.
- For small ϵk_i, optimal momentum $p = (1 - \sqrt{\epsilon k_i})^2$ and corresponding $\lambda_{i,1,2} = 1 - \sqrt{\epsilon k_i} < 1 - \epsilon k_i = \lambda_i, 0$

Experiments

Consider an error function $\mathbf{w}^T A\mathbf{w}$ with $A = \begin{pmatrix} 3 & 0 \\ 0 & 30 \end{pmatrix}$. In the continuous time limit with $\mu = 4, m = 2$, and using large time step $\Delta t = 0.5$ in the discrete case

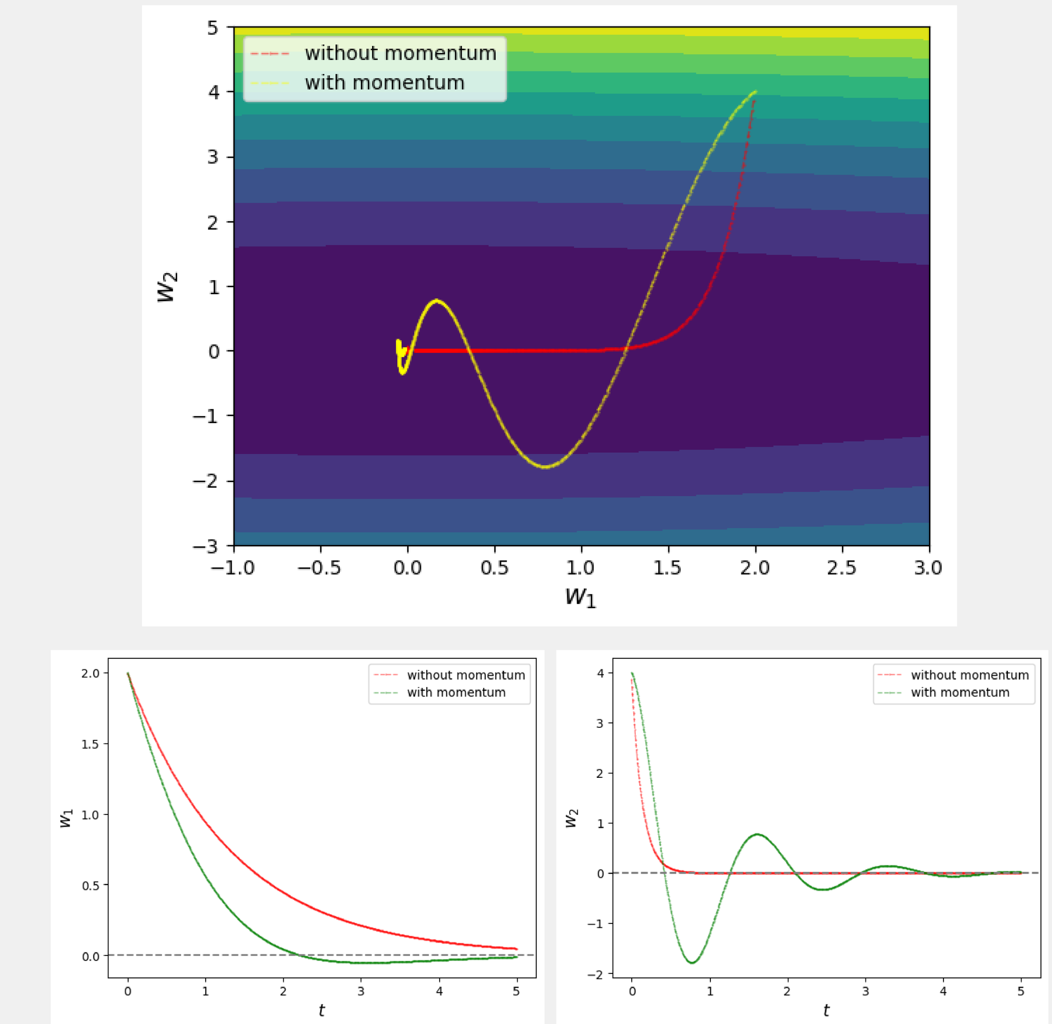

Figure 3. Continuous time

momentum improves convergence along the some directions.

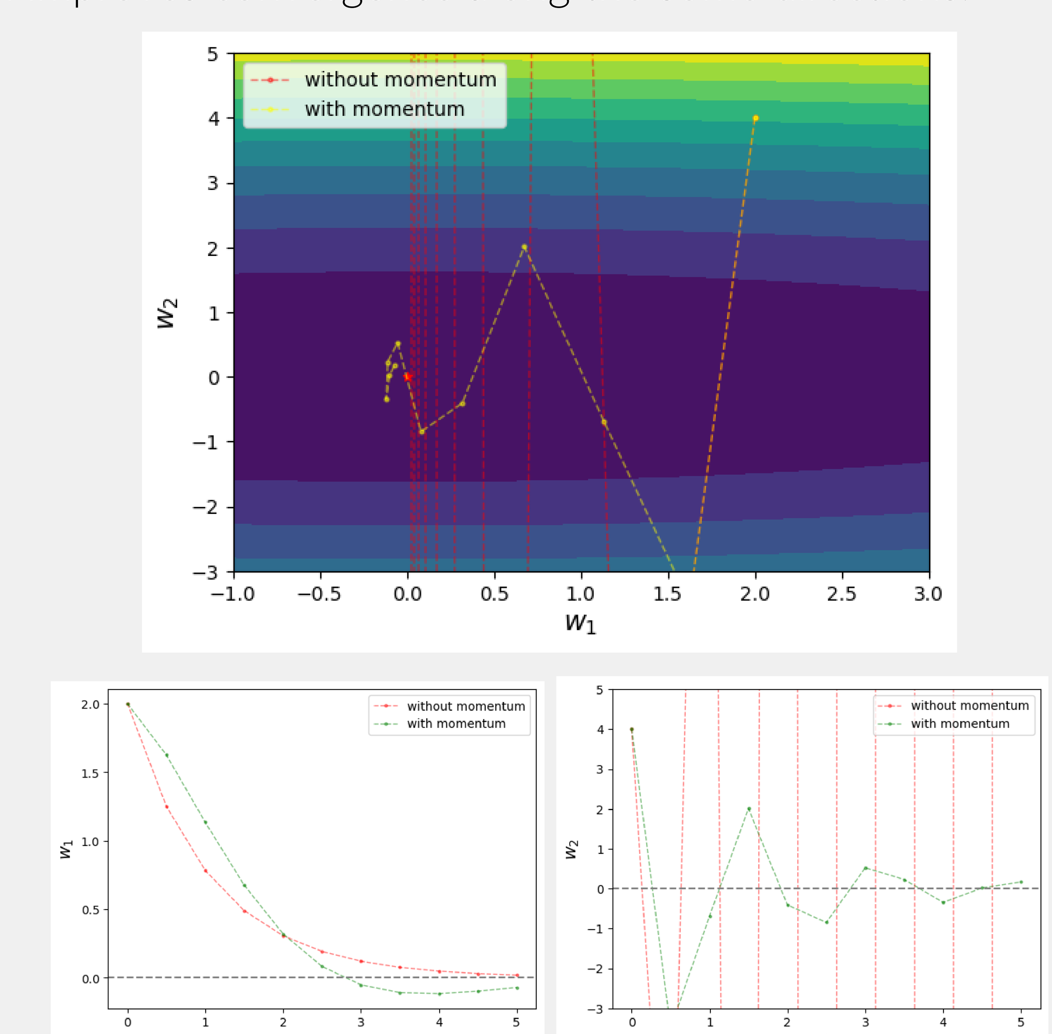

Figure 4. Discrete time

momentum allows convergence with large time steps as expected.

References

[1] Ning Qian.
On the momentum term in gradient descent learning algorithms.
Neural Networks, 12(1):145–151, 1999.

drra@kth.se, adhkal@kth.se