Optimization for Deep Learning

Lecture 8-1: Momentum SGD

Kun Yuan

Peking University

Main contents in this lecture

- Momentum SGD
- Convergence analysis
- Lower bound

Gradient descent can be slow

- Gradient descent can be very slow for ill-conditioned problems
- For example, GD converges very slow when μ/L is sufficiently small[1]

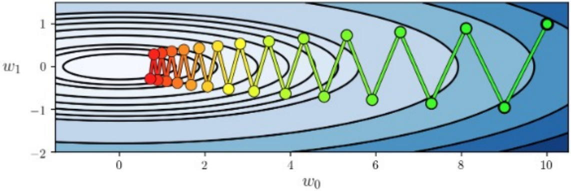

Figure: GD converges slow for ill-conditioned problem

[1] Image is from `https://github.com/jermwatt/machine_learning_refined`

Gradient descent with Polyak's momentum

- We have to alleviate the "Zig-Zag" to accelerate the algorithm
- **Polyak's momentum** method, a.k.a, **heavy-ball** gradient method

$$x_k = x_{k-1} - \gamma \nabla f(x_{k-1}) + \beta(x_{k-1} - x_{k-2})$$

where $\beta \in (0, 1)$ is the momentum parameter

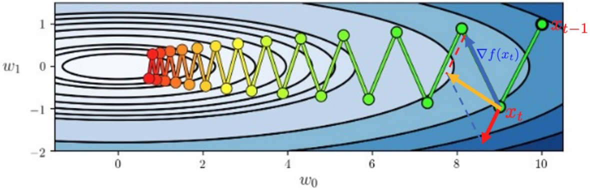

Figure: Momentum can alleviate the "Zig-Zag"

Gradient descent with Polyak's momentum

- We have to alleviate the "Zig-Zag" to accelerate the algorithm

- **Polyak's momentum** method, a.k.a, **heavy-ball** gradient method

$$x_k = x_{k-1} - \gamma \nabla f(x_{k-1}) + \beta(x_{k-1} - x_{k-2})$$

where $\beta \in (0, 1)$ is the momentum parameter

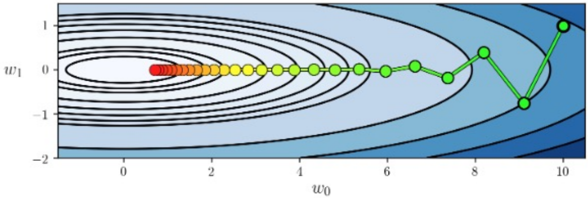

Figure: Momentum can alleviate the "Zig-Zag"

Gradient descent with Nesterov's momentum

- Gradient descent with **Nesterov's momentum**, a.k.a, **Nesterov accelerated gradient (NAG)** method

$$y_{k-1} = x_{k-1} + \beta(x_{k-1} - x_{k-2})$$
$$x_k = y_{k-1} - \gamma \nabla f(y_{k-1})$$

where $\beta \in (0, 1)$ is the momentum parameter

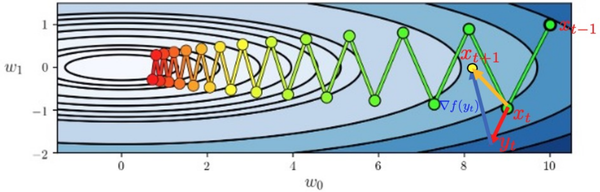

Figure: Nesterov method can alleviate the "Zig-Zag"

The convergence rate of accelerated GD

Method	Convexity	Rate	Complexity
GD	Non-convex	$O(L/k)$	$O(L/\epsilon)$
	Convex	$O(L/k)$	$O(L/\epsilon)$
	Strongly convex	$O((1-\frac{\mu}{L})^k)$	$O(\frac{L}{\mu}\log(1/\epsilon))$
NAG	Non-convex	$O(L/k)$	$O(L/\epsilon)$
	Convex	$O(L/k^2)$	$O(L/\sqrt{\epsilon})$
	Strongly convex	$O((1-\sqrt{\frac{\mu}{L}})^k)$	$O(\sqrt{\frac{L}{\mu}}\log(1/\epsilon))$
Lower bound	Non-convex	$\Omega(L/k)$	$\Omega(L/\epsilon)$
	Convex	$\Omega(L/k^2)$	$\Omega(L/\sqrt{\epsilon})$
	Strongly convex	$\Omega((1-\sqrt{\frac{\mu}{L}})^k)$	$\Omega(\sqrt{\frac{L}{\mu}}\log(1/\epsilon))$

NAG and GD has the **same** rate and complexity in non-convex scenarios (Carmon et al., 2020); **GD is optimal and cannot be improved!**

Momentum SGD

- Recall the stochastic optimization

$$\min_{x \in \mathbb{R}^d} \quad \mathbb{E}_{\xi \in \mathcal{D}}[F(x; \xi)]$$

- The standard SGD algorithm is

$$x_{k+1} = x_k - \gamma \nabla F(x_k; \xi_k)$$

- Momentum SGD

$$x_{k+1} = x_k - \gamma \nabla F(x_k; \xi_k) + \beta(x_k - x_{k-1}) \tag{1}$$

where $\beta \in [0, 1)$ is the momentum coefficient

Momentum SGD

- Momentum SGD can be rewritten into

$$\begin{aligned} m_k &= \beta m_{k-1} + \nabla F(x_k; \xi_k) \\ x_{k+1} &= x_k - \gamma m_k \end{aligned}$$

where $m_0 = 0$. This is how PyTorch implement it[2].

- To see it, we have the following recursion from the second line

$$\beta x_k = \beta x_{k-1} - \gamma \beta m_{k-1}.$$

Subtract it from the second line, we have

$$\begin{aligned} x_{k+1} - \beta x_k &= x_k - \beta x_{k-1} - \gamma(m_k - \beta m_{k-1}) \\ &= x_k - \beta x_{k-1} - \gamma \nabla F(x_k; \xi_k). \end{aligned}$$

Regrouping the terms, we achieve Momentum SGD recursion (1).

[2] https://pytorch.org/docs/stable/generated/torch.optim.SGD.html

Can momentum accelerate SGD?

Consider the linear regression problem:

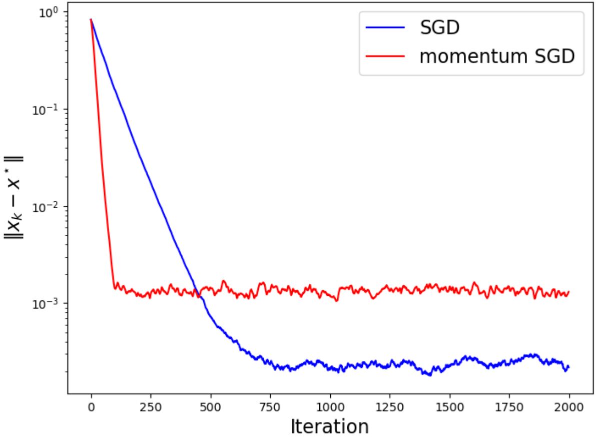

Figure: Momentum SGD with $\gamma = 0.01$ and $\beta = 0.8$, and SGD with $\gamma = 0.01$

Momentum SGD **accelerates** the rate but **deteriorates** the performance

Can momentum accelerate SGD?

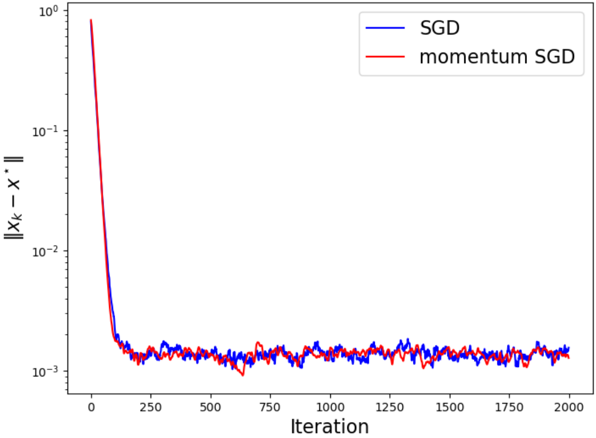

Figure: Momentum SGD with $\gamma = 0.01$ and $\beta = 0.8$, and SGD with $\gamma = 0.05$

Momentum SGD is observed **equivalent** to SGD with enlarged learning rate

Momentum SGD is equivalent to standard SGD

- Consider the standard SGD algorithm

$$x_{k+1} = x_k - \gamma_{\mathrm{s}} \nabla F(x_k; \xi_k)$$

- Consider the momentum SGD algorithm

$$x_{k+1} = x_k - \gamma_{\mathrm{m}} \nabla F(x_k; \xi_k) + \beta(x_k - x_{k-1})$$

Theorem (Yuan et al. (2016), Informal)

Assume $f(x)$ is L-smooth and strongly convex. If γ_{m} is sufficiently small, β is bounded away from 1, and $\gamma_{\mathrm{s}} = \gamma_{\mathrm{m}}/(1-\beta)$, then

*momentum SGD is **equivalent** to standard SGD*

Momentum SGD is equivalent to standard SGD

Theorem (Yuan et al. (2016), Informal)

Assume $f(x)$ is L-smooth and strongly convex. If γ_{m} is sufficiently small, β is bounded away from 1, and $\gamma_{\mathrm{s}} = \gamma_{\mathrm{m}}/(1-\beta)$, then

$$\textit{momentum SGD is } \textbf{equivalent} \textit{ to standard SGD}$$

- This result implies that momentum SGD is equivalent to SGD with enlarged learning rate; perfectly explains the numerical observation
- This result implies that momentum SGD does not bring any benefits when learning rate is sufficiently small; a negative result on momentum
- This motivates us to seek advanced algorithms with momentum

SGD with Nesterov momentum

- Consider SGD with Nesterov momentum

$$\begin{aligned} y_k =& (1-\beta_k)x_{k-1} + \beta_k v_{k-1}, \\ x_k =& y_{k-1} - \gamma \nabla F(y_{k-1};\xi_k), \\ v_k =& \beta_k^{-1} x_k + (1-\beta_k^{-1}) x_{k-1}, \end{aligned}$$

  where β_k is a time-varying momentum parameter

- As usual, we assume unbiased stochastic gradient and bounded variance

$$\mathbb{E}[\nabla F(y_{k-1};\xi_k) \mid \mathcal{F}_k] = \nabla f(y_{k-1}), \tag{2}$$

$$\mathbb{E}[\|\nabla F(y_{k-1};\xi_k) - \nabla f(y_{k-1})\|_2^2 \mid \mathcal{F}_k] \le \sigma^2. \tag{3}$$

SGD with Nesterov momentum

Theorem

Suppose $f(x)$ is L-smooth and convex, and conditions (2) and (3) hold. If we choose proper γ (see our notes), SGD with Nesterov momentum converges at the following rate:

$$\mathbb{E}[f(x_K) - f^\star] = \mathcal{O}\left(\sqrt{\frac{\sigma^2}{K}} + \frac{1}{K^2}\right)$$

- Reduce to accelerated GD when $\sigma^2 = 0$.
- Recall standard SGD converges as follows

$$\mathbb{E}[f(x_K) - f^\star] = \mathcal{O}\left(\sqrt{\frac{\sigma^2}{K}} + \frac{1}{K}\right)$$

SGD with Nesterov momentum accelerates SGD when σ^2 is small or when $1/K$ dominates; but **cannot** accelerate SGD when K is large

Lower bound of stochastic optimization

- The algorithms discussed above cannot accelerate SGD when K is large
- Is it possible to develop algorithms that can accelerate SGD when K is large?
- **No we cannot** (when only convexity and smoothness assumptions are used)

Lower bound of stochastic optimization

Theorem

For any first-order algorithm satisfying

$$x_t \in x_0 + \textit{span}\{\nabla F(x_0;\xi_0), \nabla F(x_1;\xi_1), \cdots, \nabla F(x_{k-1};\xi_{k-1})\},$$

there always exists some convex and L-smooth $f(x)$ such that

$$f(x_K) - f^\star = \Omega\left(\sqrt{\frac{\sigma^2}{K}} + \frac{1}{K^2}\right)$$

- This implies that SGD with Nesterov momentum achieves the optimal rate
- It is impossible to achieve better rate than Nesterov momentum
- It is impossible to accelerate the dominant term $\sqrt{\sigma^2/K}$; all accelerated stochastic algorithms perform **no better than SGD** when K is large

Lower bound of stochastic optimization

Table: Lower and upper complexity bounds in stochastic optimization

Algorithm	non-convex	generally-convex	strongly-convex
SGD	$\mathcal{O}\left(\frac{L\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{L\sigma^2}{\mu^2\epsilon}\ln\left(\frac{1}{\epsilon}\right) + \frac{L}{\mu}\ln\left(\frac{1}{\epsilon}\right)\right)$
Nesterov SGD	$\mathcal{O}\left(\frac{L\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)$	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2} + \sqrt{\frac{L}{\epsilon}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu\epsilon} + \sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$
Lower Bound	$\Omega\left(\frac{L\sigma^2}{\epsilon^2} + \frac{L}{\epsilon}\right)$	$\Omega\left(\frac{\sigma^2}{\epsilon^2} + \sqrt{\frac{L}{\epsilon}}\right)$	$\tilde{\Omega}\left(\frac{\sigma^2}{\mu\epsilon} + \sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$

- $\tilde{\mathcal{O}}(\cdot)$ and $\tilde{\Omega}(\cdot)$ hide all logarithm polynomials
- SGD with Nesterov momentum has nearly achieved the optimal rate
- When ϵ is sufficiently small, SGD has achieved the lower bound; no chance to improve it

However, momentum is very useful in real implementations

- In theory, no algorithm can outperform SGD when solving non-convex and smooth problems; SGD has achieved the optimal convergence rate
- In practice, momentum SGD is believed to significantly accelerate standard SGD, and is widely used in almost all deep learning tasks
- There is **an obvious gap** between theory and implementations
- We conjecture that deep learning models have more structures that we did not utilize in theoretical analysis. How to show the benefits of momentum in theory is sitll an open question

Summary

- Momentum SGD is equivalent to vanilla SGD when learning rate is small and momentum coefficient is not close to 1

- SGD with Nesterov momentum can accelerate SGD when σ^2 is small or when $1/K$ dominates the rate due to

$$\mathbb{E}[f(x_K) - f^\star] = \mathcal{O}\left(\sqrt{\frac{\sigma^2}{K}} + \frac{1}{K}\right)$$

- When σ^2 or K is large, vanilla SGD has achieved the optimal rate; no algorithm can improve vanilla SGD on the order of convergence rate

- There is a gap between the theoretical understanding and real implementations in momentum

References I

Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points i," *Mathematical Programming*, vol. 184, no. 1-2, pp. 71–120, 2020.

K. Yuan, B. Ying, and A. H. Sayed, "On the influence of momentum acceleration on online learning," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 6602–6667, 2016.