Introduction to Foundation Models

Momentum SGD

Kun Yuan

Peking University

Main contents in this lecture

- Momentum SGD

- Convergence analysis

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]

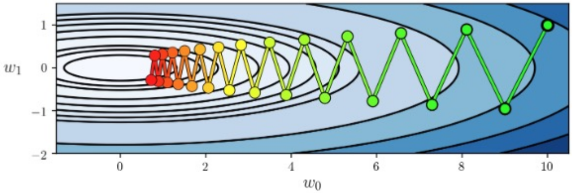

Figure: GD converges slow for ill-conditioned problem

[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

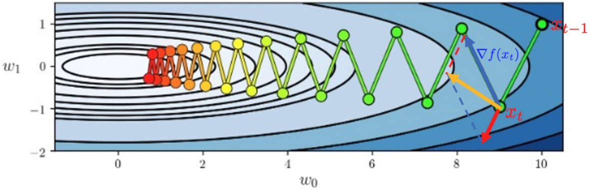

Figure: Momentum can alleviate the "Zig-Zag"

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

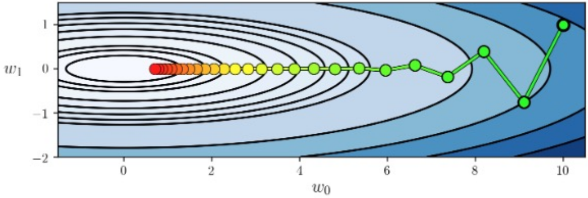

Figure: Momentum can alleviate the "Zig-Zag"

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

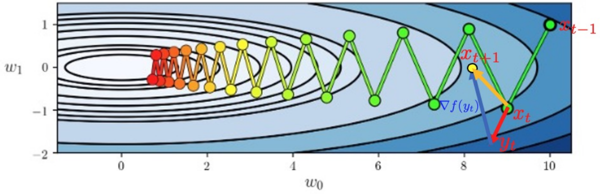

Figure: Nesterov method can alleviate the "Zig-Zag"

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 implements 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

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

$$
\begin{aligned}
\mathbb{E}[\nabla F(y_{k-1}; \xi_k) \mid \mathcal{F}_k] =& \nabla f(y_{k-1}), && (2) \\
\mathbb{E}[\|\nabla F(y_{k-1}; \xi_k) - \nabla f(y_{k-1})\|_2^2 \mid \mathcal{F}_k] \leq& \sigma^2. && (3)
\end{aligned}
$$

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

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

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

- 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