MoMo: Momentum Models for Adaptive Learning Rates

Fabian Schaipp [1] Ruben Ohana [2] Michael Eickenberg [2] Aaron Defazio [3] Robert M. Gower [2]

[1]Technical University of Munich [2]Flatiron Institute, CCM [3]Meta AI, Fundamental AI Research

TLDR

How to use knowledge on lower bounds of the loss to reduce tuning effort for the learning rate of momentum methods like SGD-M and Adam.

Background

We consider training problems of the form

$$\min_{x\in\mathbb{R}^d} f(x), \quad f(x)=\mathbb{E}_{s\sim\mathcal{D}}[f(x,s)], \tag{1}$$

with *parameters* $x\in\mathbb{R}^d$, *batch of data* s from train set $\mathcal{D}$, and *loss* $f(x,s)$.

Model-based stochastic optimization ***(cf. [1, 2])***

Many stochastic optimization methods can be summarized as follows: in each iteration **sample** s_k, then **build model** $m_k(x)$ of the loss, then **update**

$$x^{k+1}=\arg\min_{x\in\mathbb{R}^d} \; m_k(x)+\frac{1}{2\alpha_k}\|x-x^k\|^2. \tag{2}$$

1) **Linear model:** if we choose $m_k(x)=f(x^k,s_k)+\langle g_k, x-x^k\rangle$ where $g_k=\nabla f(x^k,s_k)$, then

$$x^{k+1}=x^k-\alpha_k g_k. \tag{SGD}$$

2) **Truncated model:** if we know a **lower bound** $\inf f(\cdot,s)$ of the loss (for example, zero is often a lower bound), then a better model is

$$m_k(x)=\max\{f(x^k,s_k)+\langle g_k,x-x^k\rangle,\ \inf f(\cdot,s)\}.$$

This leads to the stochastic Polyak step size [6, 4]

$$x^{k+1}=x^k-\min\left\{\alpha_k, \frac{f(x^k,s_k)-\inf f(\cdot,s_k)}{\|g_k\|^2}\right\} g_k. \tag{SPS}$$

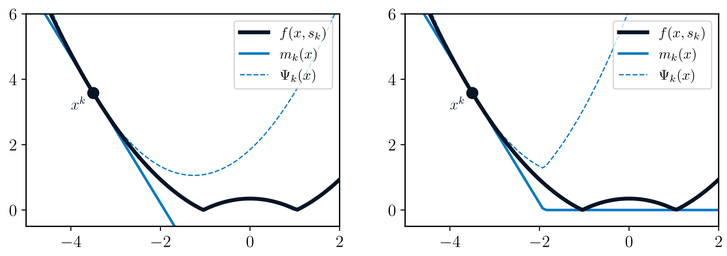

Figure 1. Denote $\Psi_k(x):=m_k(x)+\frac{1}{2\alpha_k}\|x-x^k\|^2$. Left: Linear model $m_k(x)=f(x^k,s_k)+\langle g_k,x-x^k\rangle$. Right: Truncated model $m_k(x)=\max\{f(x^k,s_k)+\langle g_k,x-x^k\rangle,\ \inf f(\cdot,s)\}$.

Observation 1

Because SPS uses a better model, it needs less tuning for the user-specified learning rate α_k than SGD.

Research Questions

Question 1: In practice, momentum typically improves training. How to combine momentum and SPS?

Question 2: Can we improve upon Adam by using a better model?

MoMo: model-based momentum

Main insight: Build a model for $f(x)$ and not for $f(x,s)$.

We can build a model of $f(x)$ by taking a weighted average over past data points. With weights $\rho_{j,k}>0$ and $\sum_{j=1}^k\rho_{j,k}=1$, we have that

$$f(x)=\mathbb{E}_{s\sim\mathcal{D}}[f(x,s)]\approx\sum_{j=1}^k\rho_{j,k}f(x,s_j).$$

Linearizing each loss around the point it was last sampled gives the model

$$m_k^{\text{avg}}(x):=\sum_{j=1}^k\rho_{j,k}\big[f(x^j,s_j)+\langle\nabla f(x^j,s_j),x-x^j\rangle\big].$$

Using **exponential moving averages**, that is $\rho_{j,k}=(1-\beta)\beta^{k-j}$, update (2) with $m_k^{\text{avg}}(x)$ turns out to be SGD with momentum [5], given by

$$\begin{aligned} d_k&=\beta d_{k-1}+(1-\beta)\nabla f(x^k,s_k),\\ x^{k+1}&=x^k-\alpha_k d_k.\end{aligned} \tag{SGD-M}$$

Our method: truncate the *momentum model* at a lower bound estimate f_*^k

$$m_k(x):=\max\{m_k^{\text{avg}}(x),f_*^k\}. \tag{3}$$

E.g. $f_*^k=0$ for positive losses. Plugging (3) into update formula (2) gives

$$x^{k+1}=x^k-\min\left\{\alpha_k,\frac{(\bar f_k+\langle d_k,x^k\rangle-\gamma_k-f_*^k)_+}{\|d_k\|^2}\right\}d_k, \tag{MoMo}$$

where
$$\begin{aligned}\bar f_k&:=\beta\bar f_{k-1}+(1-\beta)f(x^k,s_k),\\ \gamma_k&:=\beta\gamma_{k-1}+(1-\beta)\langle\nabla f(x^k,s_k),x^k\rangle.\end{aligned}$$

MoMo can also handle weight decay by adding a term $\frac{\lambda}{2}\|x\|^2$ in (2).

An Adam version

We can see Adam [3] as *preconditioned* SGD-M, that is

$$\begin{aligned} v_k&=\beta_2 v_{k-1}+(1-\beta_2)(g_k\odot g_k),\quad \mathbf{D}_k=\varepsilon+\sqrt{v_k},\\ x^{k+1}&=x^k-\alpha_k\mathbf{D}_k{}^{-1}d_k.\end{aligned}$$

This is a model-based update with adaptive norm:

$$x^{k+1}=\arg\min_{x\in\mathbb{R}^d} m_k^{\text{avg}}(x)+\frac{1}{2\alpha_k}\|x-x^k\|^2_{\mathbf{D}_k}.$$

Plugging in the MoMo model (3) instead, we obtain

$$x^{k+1}=x^k-\min\left\{\alpha_k,\frac{(\bar f_k+\langle d_k,x^k\rangle-\gamma_k-f_*^k)_+}{\|d_k\|^2_{\mathbf{D}_k}}\right\}\mathbf{D}_k^{-1}d_k. \tag{MoMo-Adam}$$

(compatible with weight decay, omitted bias correction here for simplicity)

Note: The same technique can be applied to any preconditioner $\mathbf{D}_k$!

Theory

If $f(\cdot,s)$ is convex, interpolation $\inf f(\cdot,s)=f(x^*,s)=:f^*$ holds for all s, and has *locally bouned gradients* with $\max_{x:\,\|x-x^*\|\le\|x^1-x^*\|}\mathbb{E}_{s\sim\mathcal{D}}\|\nabla f(x,s)\|^2=:G^2<\infty$ then MoMo with $f_*^k=f^*,\ \alpha_k=+\infty$ converges

$$\min_{k=1,\dots,K}\mathbb{E}[f(x^k)-f(x^*)]\le\frac{G\|x^1-x^*\|}{\sqrt{K}(1-\beta)}.$$

+ online lower bound estimation (see paper for details).

Experimental setup

The step size of MoMo is the minimum of a *(user-specified) learning rate* α_k and an *adaptive term* (computed on the fly).

Main question: can this reduce the tuning effort for α_k?

Results

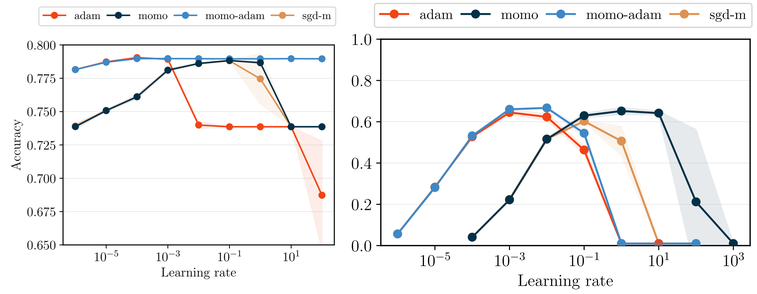

Figure 2. On the x-axis, we vary the (constant) learning rate α_k. Left: DLRM on Criteo. Right: ResNet110 on CIFAR100.

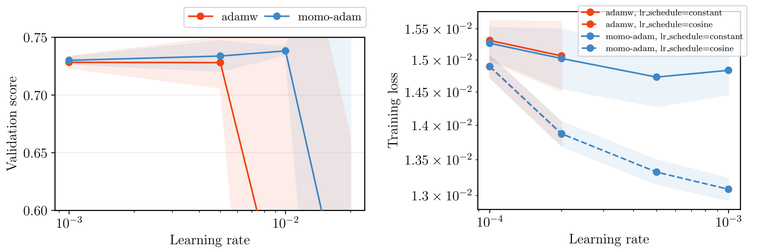

Figure 3. Left: ViT on Imagenet. Right: Diffusion model; Adam diverges for large α_k.

References

[1] Hilal Asi and John C. Duchi. Stochastic (approximate) proximal point methods: convergence, optimality, and adaptivity. *SIAM J. Optim.*, 2019.

[2] Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM J. Optim.*, 29(1):207–239, 2019.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[4] Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence. In *AISTATS*, 2021.

[5] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Phys.*, 1964.

[6] Boris T. Polyak. *Introduction to optimization.* 1987.

Available in Pytorch and Optax:

```
pip install momo-opt
```