SMG: A Shuffling Gradient-Based Method with Momentum

The 38th International Conference on Machine Learning (ICML 2021)

July 2021

Trang H. Tran (Cornell),
Lam M. Nguyen (IBM Research), Quoc Tran-Dinh (UNC)

INTRODUCTION

Problem Description

We consider the following finite-sum minimization:

$$\min_{w\in\mathbb{R}^d} \left\{ F(w) := \frac{1}{n}\sum_{i=1}^{n} f(w;i) \right\}, \tag{1}$$

where $f(\cdot;i) : \mathbb{R}^d \to \mathbb{R}$ is a given smooth and possibly nonconvex function for $i \in [n] := \{1, \ldots, n\}$.

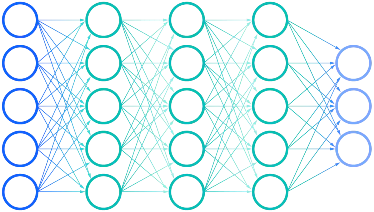

Figure: Neural network.[1]

[1] Image source: `https://www.ibm.com/`

Regular (Standard) Scheme vs. Shuffling Scheme

1. Regular (Standard) Scheme

   - **Uniformly at random:** at each iteration of epoch t, sample an index i uniformly at random from $[n] := \{1, \dots, n\}$.

Regular (Standard) Scheme vs. Shuffling Scheme

1. Regular (Standard) Scheme

   - **Uniformly at random:** at each iteration of epoch t, sample an index i uniformly at random from $[n] := \{1, \ldots, n\}$.

1	2	3	...	$n-1$	n

1 epoch $= n$ gradient evaluations

Regular (Standard) Scheme vs. Shuffling Scheme

1. Regular (Standard) Scheme

   - **Uniformly at random:** at each iteration of epoch t, sample an index i uniformly at random from $[n] := \{1, \ldots, n\}$.

1	2	3	...	$n-1$	n

1 epoch $= n$ gradient evaluations

2. Shuffling Scheme

   - **Incremental Gradient:** for all epoch t, use a fixed permutation $\pi^{(t)} := \{1, \ldots, n\}$.
   - **Shuffle Once:** at the first epoch $t = 1$, random shuffle a permutation $\pi^{(t)}$ from $[n] := \{1, \ldots, n\}$ and use it for all epochs.
   - **Random Reshuffling:** at epoch t, random shuffle a permutation $\pi^{(t)}$ from $[n] := \{1, \ldots, n\}$.

ALGORITHMS

Algorithm 1 Shuffling Momentum Gradient (SMG)

Initialization: Choose $\tilde{w}_0 \in \mathbb{R}^d$ and set $\tilde{m}_0 := \mathbf{0}$.
for $t := 1, 2, \cdots, T$ **do**
  Set $w_0^{(t)} := \tilde{w}_{t-1}$; $m_0^{(t)} := \tilde{m}_{t-1}$; and $v_0^{(t)} := \mathbf{0}$;
  Generate a deterministic or random permutation $\pi^{(t)}$ of $[n]$;
  for $i := 0, \cdots, n-1$ **do**
    Query $g_i^{(t)} := \nabla f(w_i^{(t)}; \pi^{(t)}(i+1))$;
    Choose $\eta_i^{(t)} := \frac{\eta_t}{n}$ and update

$$\begin{cases} m_{i+1}^{(t)} & := & \beta m_0^{(t)} + (1-\beta) g_i^{(t)} \\ v_{i+1}^{(t)} & := & v_i^{(t)} + \frac{1}{n} g_i^{(t)} \\ w_{i+1}^{(t)} & := & w_i^{(t)} - \eta_i^{(t)} m_{i+1}^{(t)}; \end{cases}$$

New update

  end for
  Set $\tilde{w}_t := w_n^{(t)}$ and $\tilde{m}_t := v_n^{(t)}$;
end for
Output: Choose $\hat{w}_T \in \{\tilde{w}_0, \cdots, \tilde{w}_{T-1}\}$ at random with probability $\mathbb{P}[\hat{w}_T = \tilde{w}_{t-1}] = \frac{\eta_t}{\sum_{t=1}^{T} \eta_t}$.

Assumptions

Assumption 1

Problem (1) satisfies:

(a) **(Boundedness)** $F_* := \inf_{w \in \mathbb{R}^d} F(w) > -\infty$.

(b) **(L-smoothness)** $f(\cdot; i)$ is L-smooth for all $i \in [n]$, i.e., there exists a constant $L > 0$ such that for all $w, w' \in \mathrm{dom}\,(F)$:

$$\|\nabla f(w; i) - \nabla f(w'; i)\| \leq L\|w - w'\|. \tag{2}$$

(c) **(Generalized bounded variance)** There exist two finite constants $\Theta, \sigma \geq 0$ such that for any $w \in \mathrm{dom}\,(F)$:

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f(w; i) - \nabla F(w)\|^2 \leq \Theta\|\nabla F(w)\|^2 + \sigma^2. \tag{3}$$

Non-convex results

Theorem 1

Suppose that Assumption 1 holds for (1). Let $\{w_i^{(t)}\}_{t=1}^T$ be generated by Algorithm 1 with a fixed momentum weight $0 \leq \beta < 1$ and an epoch learning rate $\eta_i^{(t)} := \frac{\eta_t}{n}$ for every $t \geq 1$. Assume that $\eta_0 = \eta_1$, $\eta_t \geq \eta_{t+1}$, and $0 < \eta_t \leq \frac{1}{L\sqrt{K}}$ for $t \geq 1$, where $K := \max\left\{\frac{5}{2}, \frac{9(5-3\beta)(\Theta+1)}{1-\beta}\right\}$. Then

$$\mathbb{E}\left[\|\nabla F(\hat{w}_T)\|^2\right] \leq \frac{4[F(\tilde{w}_0) - F_*]}{(1-\beta)\sum_{t=1}^T \eta_t} + \frac{9\sigma^2 L^2(5-3\beta)}{(1-\beta)} \left(\frac{\sum_{t=1}^T \eta_{t-1}^3}{\sum_{t=1}^T \eta_t}\right).$$

This result is flexible enough to cover multiple different learning rates.

Corollary

Corollary (Constant learning rate)

Let us fix the number of epochs $T \geq 1$, and choose a constant learning rate $\eta_t := \frac{\gamma}{T^{1/3}}$ for some $\gamma > 0$ such that $\frac{\gamma}{T^{1/3}} \leq \frac{1}{L\sqrt{K}}$ for $t \geq 1$ in Algorithm 1. Then, under the conditions of Theorem 1:

$$\mathbb{E}\left[\|\nabla F(\hat{w}_T)\|^2\right] \leq \frac{1}{T^{2/3}} \left(\frac{4[F(\tilde{w}_0) - F_*]}{(1-\beta)\gamma} + \frac{9\sigma^2(5-3\beta)L^2\gamma^2}{(1-\beta)} \right).$$

With a constant LR, the convergence rate of SMG is exactly expressed as

$$\mathcal{O}\left(\frac{[F(\tilde{w}_0) - F_*] + \sigma^2}{T^{2/3}} \right),$$

which matches the best known rate in the literature in term of T for general shuffling-type strategies [Nguyen et al., 2020].

Other learning rate schemes

- Diminishing learning rate [Nguyen et al., 2020]:

$$\eta_t := \frac{\gamma}{(t+\lambda)^\alpha} \quad \text{where } \alpha, \gamma > 0, \text{ and } \lambda \geq 0.$$

  Choosing $\alpha = 1/3$, the convergence rate of SMG is $\mathcal{O}(T^{-2/3}\log(T))$ in epoch.

Other learning rate schemes

- Diminishing learning rate [Nguyen et al., 2020]:

$$\eta_t := \frac{\gamma}{(t+\lambda)^\alpha} \quad \text{where } \alpha, \gamma > 0, \text{ and } \lambda \geq 0.$$

Choosing $\alpha = 1/3$, the convergence rate of SMG is $\mathcal{O}(T^{-2/3}\log(T))$ in epoch.

- Exponential learning rate [Li et al., 2020]:

$$\eta_t := \frac{\gamma\alpha^t}{T^{1/3}}, \quad \text{where } \gamma > 0, \rho > 0, \text{ and } \alpha := \rho^{1/T} \in (0,1).$$

- Cosine learning rate [Loshchilov and Hutter, 2017, Smith, 2017]:

$$\eta_t := \frac{\gamma}{T^{1/3}}\left(1 + \cos\frac{t\boldsymbol{\pi}}{T}\right), \quad \text{where } \gamma > 0.$$

The scheduled exponential and cosine learning rates still preserve our best known convergence rate $\mathcal{O}(T^{-2/3})$.

Theorem RR

Theorem 2

Suppose that Assumption 1 holds for (1). Let $\{w_i^{(t)}\}_{t=1}^T$ be generated by Algorithm 1 under a **randomized reshuffling strategy**, a fixed momentum weight $0 \le \beta < 1$, and an epoch learning rate $\eta_i^{(t)} := \frac{\eta_t}{n}$ for every $t \ge 1$. Assume that $\eta_t \ge \eta_{t+1}$ and $0 < \eta_t \le \frac{1}{L\sqrt{D}}$ for $t \ge 1$, where $D = \max\left(\frac{5}{3}, \frac{6(5-3\beta)(\Theta+n)}{n(1-\beta)}\right)$ and $\eta_0 = \eta_1$. Then

$$\mathbb{E}\left[\|\nabla F(\hat{w}_T)\|^2\right] \le \frac{4\left[F(\tilde{w}_0) - F_*\right]}{(1-\beta)\sum_{t=1}^T \eta_t} + \frac{6\sigma^2(5-3\beta)L^2}{n(1-\beta)}\left(\frac{\sum_{t=1}^T \eta_{t-1}^3}{\sum_{t=1}^T \eta_t}\right).$$

With a randomized reshuffling strategy and constant learning rates, the convergence rate of SMG is improved to

$$\mathcal{O}\left(\frac{\left[F(\tilde{w}_0) - F_*\right] + \sigma^2}{n^{1/3}\,T^{2/3}}\right),$$

which matches the best known rate in the literature in term of T for general shuffling-type strategies [Mishchenko et al., 2020].

Single Shuffling Momentum Gradient

Algorithm 2 Single Shuffling Momentum Gradient

Initialization: Choose $\tilde{w}_0 \in \mathbb{R}^d$ and set $\tilde{m}_0 := \mathbf{0}$;
Generate a permutation π of $[n]$;
for $t := 1, 2, \cdots, T$ **do**
  Set $w_0^{(t)} := \tilde{w}_{t-1}$ and $m_0^{(t)} := \tilde{m}_{t-1}$;
  for $i =: 0, \cdots, n-1$ **do**
    Query $g_i^{(t)} := \nabla f(w_i^{(t)}; \pi(i+1))$;
    Choose $\eta_i^{(t)} := \frac{\eta_t}{n}$ and update

$$\begin{cases} m_{i+1}^{(t)} & := & \beta m_i^{(t)} + (1-\beta) g_i^{(t)} \\ w_{i+1}^{(t)} & := & w_i^{(t)} - \eta_i^{(t)} m_{i+1}^{(t)}; \end{cases}$$

  end for
  Set $\tilde{w}_t := w_n^{(t)}$ and $\tilde{m}_t := m_n^{(t)}$;
end for
Output: Choose $\hat{w}_T \in \{\tilde{w}_0, \cdots, \tilde{w}_{T-1}\}$ at random with probability $\mathbb{P}[\hat{w}_T = \tilde{w}_{t-1}] = \frac{\eta_t}{\sum_{t=1}^T \eta_t}$.

Assumption

Assumption 2 (Bounded gradient)

There exists $G > 0$ such that $\|\nabla f(x; i)\| \leq G$, $\forall x \in \mathrm{dom}\,(F)$ and $i \in [n]$.

This assumption is slightly stronger than assumption 1(c) (Generalized bounded variance).

Theorem

Theorem 3

Let $\{w_i^{(t)}\}_{t=1}^{T}$ be generated by Algorithm 2 with a LR $\eta_i^{(t)} := \frac{\eta_t}{n}$ and $0 < \eta_t \le \frac{1}{L}$ for $t \ge 1$. Then, under Assumption 1(a)-(b) and Assumption 2, we have

$$\mathbb{E}\big[\|\nabla F(\hat{w}_T)\|^2\big] \le \frac{\Delta_1}{\big(\sum_{t=1}^{T}\eta_t\big)(1-\beta^n)} + L^2G^2\left(\frac{\sum_{t=1}^{T}\xi_t^3}{\sum_{t=1}^{T}\eta_t}\right) + \frac{4\beta^nG^2}{1-\beta^n},$$

where $\xi_t := \max(\eta_t, \eta_{t-1})$ for $t \ge 2$, $\xi_1 = \eta_1$, and

$$\Delta_1 := 2[F(\tilde{w}_0) - F_*] + \left(\frac{1}{L} + \eta_1\right)\|\nabla F(\tilde{w}_0)\|^2 + 2L\eta_1^2G^2.$$

With a constant learning rate, the convergence rate of Algorithm 2 is

$$\mathcal{O}\left(\frac{L[F(\tilde{w}_0) - F_*] + \|\nabla F(\tilde{w}_0)\|^2 + G^2}{T^{2/3}}\right).$$

EXPERIMENTS

Experiments - Neural Networks

We compare our SMG with SGD and two other methods: SGD with Momentum [Polyak, 1964] and Adam [Kingma and Ba, 2014] using the following architectures:

- Fully connected network (LeNet-300-100 [LeCun et al., 1998]) for the `Fashion-MNIST` dataset.
- Convolutional neural network (LeNet-5 [LeCun et al., 1998]) for the `CIFAR-10` dataset.

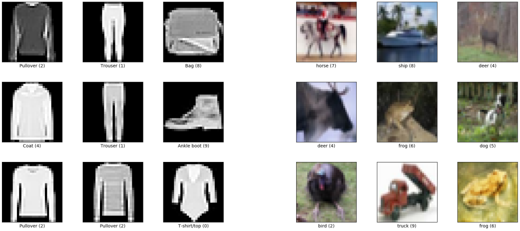

Figure: `Fashion-MNIST` dataset (left) and `CIFAR-10` dataset (right).[2]

[2]Image source: `https://www.tensorflow.org/`

Results - Neural Networks

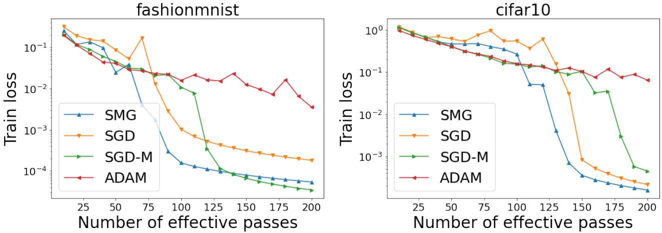

Figure: The train loss produced by SMG, SGD, SGD-M, and Adam.

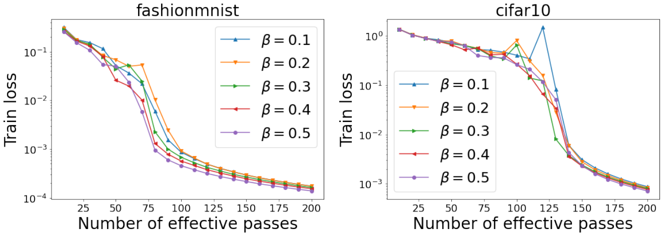

Figure: The train loss reported by SMG with different β.

Experiments - Non-convex Logistic Regression

We consider the following non-convex binary classification problem:

$$\min_{w\in\mathbb{R}^d} \Big\{ F(w) := \frac{1}{n}\sum_{i=1}^{n} \Big[\log(1+\exp(-y_i x_i^\top w)) + \lambda r(w) \Big] \Big\},$$

where $\{(x_i, y_i)\}_{i=1}^n$: a set of training samples,

$$r(w) := \frac{1}{2}\sum_{j=1}^{d} \frac{w_j^2}{1+w_j^2},$$ a nonconvex regularizer,

$\lambda := 0.01,$ a regularization parameter.

We did the similar experiments on two classification datasets `w8a` and `ijcnn1` from `LIBSVM`.

Results - Non-convex Logistic Regression

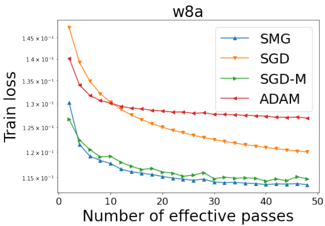

Figure: The train loss produced by SMG, SGD, SGD-M, and Adam.

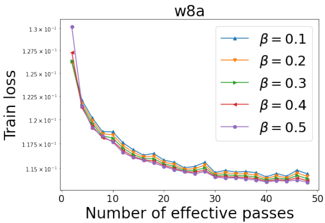

Figure: The train loss produced by SMG under different values of β.

Our Contributions

(a) We develop SMG, a novel shuffling gradient-based method with momentum for the finite-sum nonconvex minimization problem.

(b) We establish the convergence of our method and achieve the state-of-the-art $\mathcal{O}\left(1/T^{2/3}\right)$ convergence rate for all the shuffling strategies. When using a random reshuffling scheme, this rate is improved by $n^{1/3}$.

(c) We study and provide theoretical results for different learning rates, including diminishing, exponential, and cosine scheduled schemes.

(d) We analyze the convergence of a variant with traditional momentum update and achieve the same $\mathcal{O}(1/T^{2/3})$ epoch-wise rate using single shuffling strategies.

References I

Kingma, D. P. and Ba, J. (2014).
ADAM: A Method for Stochastic Optimization.
Proceedings of the 3rd International Conference on Learning Representations (ICLR), abs/1412.6980.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998).
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324.

Li, L., Zhuang, Z., and Orabona, F. (2020).
Exponential step sizes for non-convex optimization.
arXiv preprint arXiv:2002.05273.

Loshchilov, I. and Hutter, F. (2017).
Sgdr: Stochastic gradient descent with warm restarts.

Mishchenko, K., Khaled Ragab Bayoumi, A., and Richtárik, P. (2020).
Random reshuffling: Simple analysis with vast improvements.
Advances in Neural Information Processing Systems, 33.

References II

Nguyen, L. M., Tran-Dinh, Q., Phan, D. T., Nguyen, P. H., and van Dijk, M. (2020).
A unified convergence analysis for shuffling-type gradient methods.
arXiv preprint arXiv:2002.08246.

Polyak, B. T. (1964).
Some methods of speeding up the convergence of iteration methods.
USSR Computational Mathematics and Mathematical Physics, 4(5):1–17.

Smith, L. N. (2017).
Cyclical learning rates for training neural networks.
In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE.

THANK YOU!!!

Trang H. Tran - htt27@cornell.edu
https://htt-trangtran.github.io/