Painless Stochastic Gradient
Interpolation, Line-Search, and Convergence Rates

NeurIPS 2019

Sharan Vaswani (**Mila**)

Aaron Mishkin (**UBC**)

Issam Laradji (**UBC, EAI**)

Mark Schmidt (**UBC**)

Gauthier Gidel **(Mila, EAI)**

Simon Lacoste-Julien (**Mila**)

SGD is awesome, but...

Objective

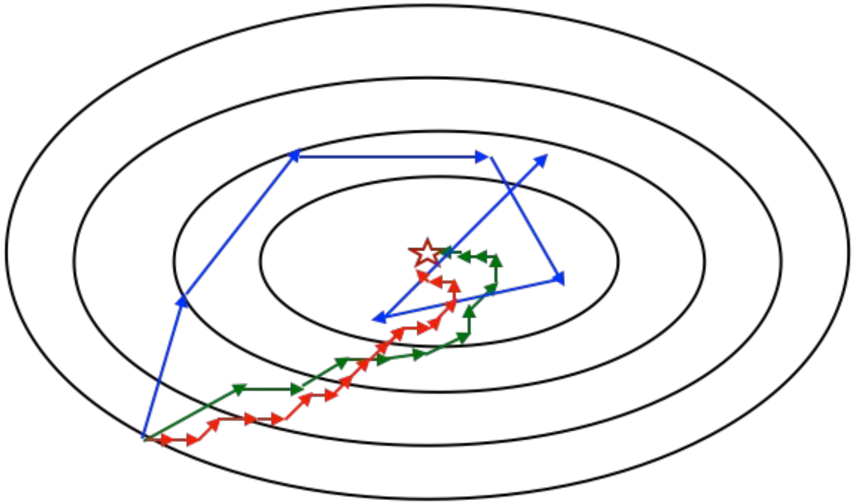
$$\min f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

Stochastic Gradient Descent (SGD)

$$w_{k+1} = w_k - \eta_k \nabla f_{ik}(w_k)$$

- + Simple to implement and use.
- + Good generalization properties.

- - Slow convergence.
- - Need to carefully tune the step-size.



Painful ! :(

Making SGD painless

Ingredient 1: Interpolation

- Satisfied by large *over-parametrized* models including typical neural networks and expressive kernel mappings.

$$||\nabla f(w^*)|| = 0 \implies \forall i, ||\nabla f_i(w^*)|| = 0$$

- Results in fast convergence of constant step-size SGD.

Ingredient 2: Stochastic Line Search (SLS)

- Don't manually tune the step-size, automatically *search* for it.

Strong theoretical results and good empirical performance!

SLS is simply SGD + Line search

In iteration k,

1. Compute the gradients $\nabla f_{ik}(w_k)$ for a given training batch

2. Search for a step-size η_k that satisfies the following *stochastic Armijo condition,*

$$f_{ik}\left(w_k - \eta_k \nabla f_{ik}(w_k)\right) \leq f_{ik}(w_k) - c\, \eta_k \,||\nabla f_{ik}(w_k)||^2$$

3. Use the step-size and update the model parameters with SGD,

$$w_{k+1} = w_k - \eta_k \nabla f_{ik}(w_k)$$

SLS is theoretically sound

Interpolation enables SLS

- Match *full-batch* convergence rates for smooth convex functions and strongly-convex functions.
- Achieve fast convergence rates for non-convex functions*.

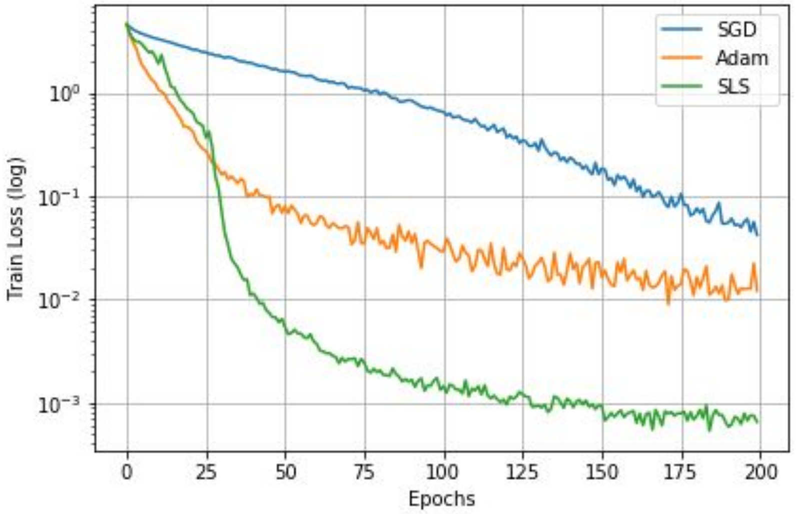
Other results:

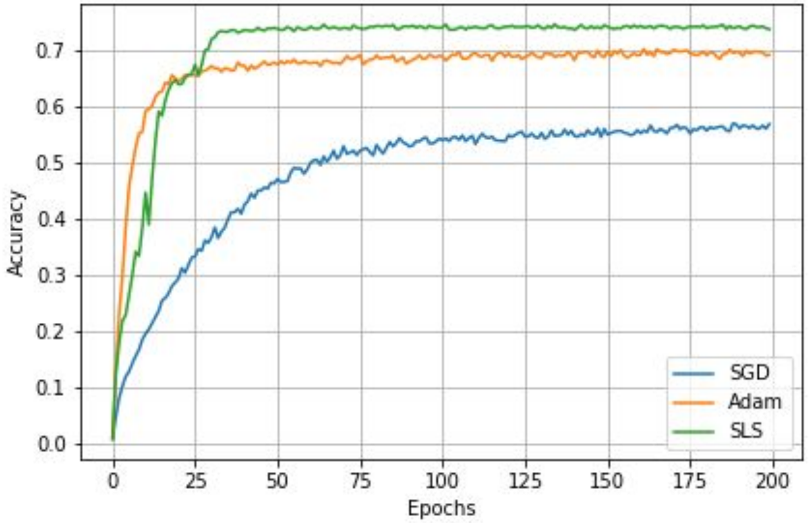
- *Lipschitz line search* for Stochastic Extra-gradient (SEG) and its fast convergence for a subset of non-convex functions.

- Fast convergence of SEG + Lipschitz line search for a class of saddle point problems.

* Additional assumptions apply

SLS optimizes faster and better





ResNet-34 on CIFAR100

Other results: Deep matrix factorization, Kernels, Min-max games

Try SLS in your project

1. Install our optimizer

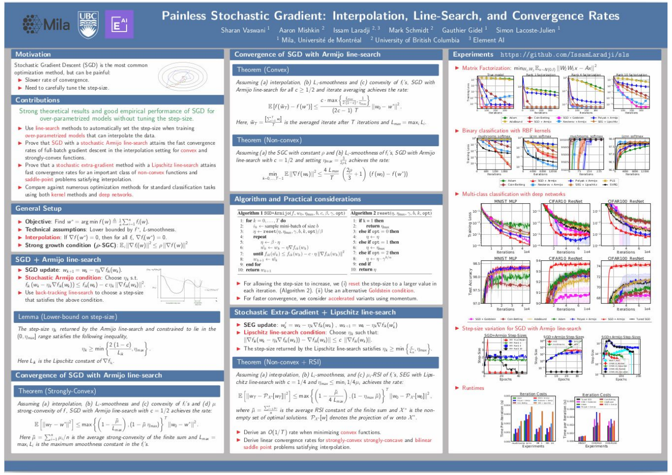
```
pip install --upgrade git+https://github.com/IssamLaradji/sls.git
```

2. Check out our Github code:

https://github.com/IssamLaradji/sls

3. Read the paper:

https://arxiv.org/abs/1905.09997



Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates

Sharan Vaswani [1] Aaron Mishkin [2] Issam Laradji [2, 3] Mark Schmidt [2] Gauthier Gidel [1] Simon Lacoste-Julien [1]
[1] Mila, Université de Montréal [2] University of British Columbia [3] Element AI

Motivation

Stochastic Gradient Descent (SGD) is the most common optimization method, but can be painful:
- Slower rate of convergence.
- Need to carefully tune the step-size.

Contributions

Strong theoretical results and good empirical performance of SGD for over-parametrized models without tuning the step-size.
- Use line-search methods to automatically set the step-size when training over-parametrized models that can interpolate the data.
- Prove that SGD with a stochastic Armijo line-search attains the fast convergence rates of full-batch gradient descent in the interpolation setting for convex and strongly-convex functions.
- Prove that a stochastic extra-gradient method with a Lipschitz line-search attains fast convergence rates for an important class of non-convex functions and saddle-point problems satisfying interpolation.
- Compare against numerous optimization methods for standard classification tasks using both kernel methods and deep networks.

General Setup

- **Objective**: Find $w^* = \arg\min f(w) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(w)$.
- **Technical assumptions**: Lower bounded by f^*, L-smoothness.
- **Interpolation**: If $\nabla f(w^*) = 0$, then for all f_i, $\nabla f_i(w^*) = 0$.
- **Strong growth condition (ρ-SGC)**: $\mathbb{E}_i \|\nabla f_i(w)\|^2 \le \rho \|\nabla f(w)\|^2$

SGD + Armijo line-search

- **SGD update**: $w_{k+1} = w_k - \eta_k \nabla f_{ik}(w_k)$.
- **Stochastic Armijo condition**: Choose η_k s.t. $f_{ik}(w_k - \eta_k \nabla f_{ik}(w_k)) \le f_{ik}(w_k) - c\,\eta_k \|\nabla f_{ik}(w_k)\|^2$.
- Use back-tracking line-search to choose a step-size that satisfies the above condition.

Lemma (Lower-bound on step-size)

The step-size η_k returned by the Armijo line-search and constrained to lie in the $(0, \eta_{max}]$ range satisfies the following inequality,

$$\eta_k \ge \min\left\{\frac{2(1-c)}{L_{ik}}, \eta_{max}\right\}.$$

Here L_{ik} is the Lipschitz constant of ∇f_{ik}.

Convergence of SGD with Armijo line-search

Theorem (Strongly-Convex)

Assuming (a) interpolation, (b) L_i-smoothness and (c) convexity of f_i's and (d) μ strong-convexity of f, SGD with Armijo line-search with $c = 1/2$ achieves the rate:

$$\mathbb{E}\left[\|w_T - w^*\|^2\right] \le \max\left\{\left(1 - \frac{\bar{\mu}}{L_{max}}\right), (1 - \bar{\mu}\,\eta_{max})\right\}^T \|w_0 - w^*\|^2.$$

Here $\bar{\mu} = \sum_{i=1}^{n} \mu_i / n$ is the average strong-convexity of the finite sum and $L_{max} = \max_i L_i$ is the maximum smoothness constant in the f_i's.

Convergence of SGD with Armijo line-search

Theorem (Convex)

Assuming (a) interpolation, (b) L_i-smoothness and (c) convexity of f_i's, SGD with Armijo line-search for all $c \ge 1/2$ and iterate averaging achieves the rate:

$$\mathbb{E}[f(\bar{w}_T) - f(w^*)] \le \frac{c \cdot \max\left\{\frac{L_{max}}{2(1-c)}, \frac{1}{\eta_{max}}\right\}}{(2c-1)\,T} \|w_0 - w^*\|^2.$$

Here, $\bar{w}_T = \frac{\left[\sum_{i=1}^{T} w_i\right]}{T}$ is the averaged iterate after T iterations and $L_{max} = \max_i L_i$.

Theorem (Non-convex)

Assuming (a) the SGC with constant ρ and (b) L_i-smoothness of f_i's, SGD with Armijo line-search with $c = 1/2$ and setting $\eta_{max} = \frac{3}{2\rho L}$ achieves the rate:

$$\min_{k=0,\dots,T-1} \mathbb{E}\|\nabla f(w_k)\|^2 \le \frac{4\,L_{max}}{T}\left(\frac{2\rho}{3} + 1\right)(f(w_0) - f(w^*))$$

Algorithm and Practical considerations

```
Algorithm 1 SGD+Armijo($f, w_0, \eta_{max}, b, c, \beta, \gamma$, opt)
1: for $k = 0, \dots, T$ do
2:     $i_k \leftarrow$ sample mini-batch of size $b$
3:     $\eta \leftarrow$ reset($\eta, \eta_{max}, \gamma, b, k$, opt)$/\beta$
4:     repeat
5:         $\eta \leftarrow \beta \cdot \eta$
6:         $\tilde{w}_k \leftarrow w_k - \eta \nabla f_{ik}(w_k)$
7:     until $f_{ik}(\tilde{w}_k) \le f_{ik}(w_k) - c \cdot \eta \|\nabla f_{ik}(w_k)\|^2$
8:     $w_{k+1} \leftarrow \tilde{w}_k$
9: end for
10: return $w_{k+1}$
```

```
Algorithm 2 reset($\eta, \eta_{max}, \gamma, b, k$, opt)
1: if k = 1 then
2:     return $\eta_{max}$
3: else if opt = 0 then
4:     $\eta \leftarrow \eta$
5: else if opt = 1 then
6:     $\eta \leftarrow \eta_{max}$
7: else if opt = 2 then
8:     $\eta \leftarrow \eta \cdot \gamma^{b/n}$
9: end if
10: return $\eta$
```

- For allowing the step-size to increase, we (i) reset the step-size to a larger value in each iteration. (Algorithm 2). (ii) Use an alternative Goldstein condition.
- For faster convergence, we consider accelerated variants using momentum.

Stochastic Extra-Gradient + Lipschitz line-search

- **SEG update**: $w'_k = w_k - \eta_k \nabla f_{ik}(w_k)$, $w_{k+1} = w_k - \eta_k \nabla f_{ik}(w'_k)$
- **Lipschitz line-search condition**: Choose η_k such that: $\|\nabla f_{ik}(w_k - \eta_k \nabla f_{ik}(w_k)) - \nabla f_{ik}(w_k)\| \le c\,\|\nabla f_{ik}(w_k)\|$.
- The step-size returned by the Lipschitz line-search satisfies $\eta_k \ge \min\left\{\frac{c}{L_{ik}}, \eta_{max}\right\}$.

Theorem (Non-convex + RSI)

Assuming (a) interpolation, (b) L_i-smoothness, and (c) μ_i-RSI of f_i's, SEG with Lipschitz line-search with $c = 1/4$ and $\eta_{max} \le \min_i 1/4\mu_i$ achieves the rate:

$$\mathbb{E}\left[\|w_T - \mathcal{P}_{\mathcal{X}^*}[w_T]\|^2\right] \le \max\left\{\left(1 - \frac{\bar{\mu}}{4\,L_{max}}\right), (1 - \eta_{max}\,\bar{\mu})\right\}^T \|w_0 - \mathcal{P}_{\mathcal{X}^*}[w_0]\|^2,$$

where $\bar{\mu} = \frac{\sum_{i=1}^{n} \mu_i}{n}$ is the average RSI constant of the finite sum and $\mathcal{X}^*$ is the non-empty set of optimal solutions. $\mathcal{P}_{\mathcal{X}^*}[w]$ denotes the projection of w onto $\mathcal{X}^*$.

- Derive an $O(1/T)$ rate when minimizing convex functions.
- Derive linear convergence rates for strongly-convex strongly-concave and bilinear saddle point problems satisfying interpolation.

Experiments `https://github.com/IssamLaradji/sls`

- Matrix Factorization: $\min_{W_1,W_2} \mathbb{E}_{x\sim N(0,I)} \|W_2 W_1 x - Ax\|^2$
- Binary classification with RBF kernels
- Multi-class classification with deep networks
- Step-size variation for SGD with Armijo line-search
- Runtimes

Come to our poster

Hall B + C #123

East Exhibition

Tue Dec 10th

5:30 PM