Minimization by Incremental Stochastic Surrogate with Application to Bayesian Deep Learning

B. Karimi[1,2], **E. Moulines**[2]

INRIA[1], CMAP École Polytechnique[2]

belhal.karimi@polytechnique.edu

Problem Statement

We are interested in the constrained minimization of a large sum of nonconvex functions defined as:

$$\min_{\theta\in\Theta}\left[f(\theta)\triangleq\sum_{i=1}^{N}f_i(\theta)\right] \tag{1}$$

Beforehand, let $\mathcal{T}(\Theta)$ be a neighborhood of Θ and assume that:

M 1. *For all $i\in[\![N]\!]$, f_i is continuously differentiable on $\mathcal{T}(\Theta)$.*

M 2. *For all $i\in[\![N]\!]$, f_i is bounded from below, i.e. there exist a constant $M_i\in\mathbb{R}$ such as for all $\theta\in\Theta$, $f_i(\theta)\geq M_i$.*

For any $\theta\in\Theta$ and $i\in[\![N]\!]$, we say, following (Mairal, 2015) that a function $f_{i,\theta}:\mathbb{R}^p\to\mathbb{R}$ is a surrogate of f_i at θ if the following properties are satisfied:

- the function $\vartheta\to f_{i,\theta}(\vartheta)$ is continuously differentiable on $\mathcal{T}(\Theta)$
- for all $\vartheta\in\Theta$, $f_{i,\theta}(\vartheta)\geq f_i(\vartheta)$, $f_{i,\theta}(\theta)=f_i(\theta)$ and $\nabla f_{i,\theta}(\vartheta)\big|_{\vartheta=\theta}=\nabla f_i(\vartheta)\big|_{\vartheta=\theta}$.

The gap $f_{i,\theta}-f_i$ plays a key role in the convergence analysis and we require this error to be L-smooth for some constant $L>0$ Denote by $\langle\cdot,\cdot\rangle$ the scalar product, we also introduce the following stationary point condition:

Definition 1. (Asymptotic Stationary Point Condition)
A sequence $(\theta^k)_{k\geq0}$ satisfies the asymptotic stationary point condition if

$$\liminf_{k\to\infty}\inf_{\theta\in\Theta}\frac{\langle\nabla f(\theta^k),\theta-\theta^k\rangle}{\|\theta-\theta^k\|_2}\geq0. \tag{2}$$

MISO Scheme

The incremental scheme of (Mairal, 2015) computes surrogate functions, at each iteration of the algorithm, for a mini-batch of components:

Algorithm 1 MISO algorithm

Initialization: given an initial parameter estimate θ^0, for all $i\in[\![N]\!]$ compute a surrogate function $\vartheta\to f_{i,\theta^0}(\vartheta)$.

Iteration k: given the current estimate θ^{k-1}:

1. Pick a set I_k uniformly on $\{A\subset[\![N]\!],\mathrm{card}(A)=p\}$
2. For all $i\in I_k$ and compute $\vartheta\to f_{i,\theta^{k-1}}(\vartheta)$, a surrogate of f_i at θ^{k-1}.
3. Set $\theta^k\in\arg\min_{\vartheta\in\Theta}\sum_{i=1}^N a_i^k(\vartheta)$ where $a_i^k(\vartheta)$ are defined recursively as follows:

$$a_i^k(\vartheta)\triangleq\begin{cases}f_{i,\theta^{k-1}}(\vartheta) & \text{if } i\in I_k\\ a_i^{k-1}(\vartheta) & \text{otherwise}\end{cases} \tag{3}$$

MISSO Scheme

- Case when the surrogate functions computed in Algorithm 1 **are not tractable**.
- Assume that the surrogate can be expressed as an integral over a set of latent variables $z=(z_i\in\mathsf{Z}_i,i\in[\![N]\!])\in\mathsf{Z}$ where $\mathsf{Z}=\bigtimes_{i=1}^N\mathsf{Z}_i$ where Z_i is a subset of $\mathbb{R}^{m_i}$.

$$f_{i,\theta}(\vartheta)\triangleq\int_{\mathsf{Z}_i}r_{i,\theta}(z_i,\vartheta)p_i(z_i,\theta)\mu_i(\mathrm{d}z_i)\quad\text{for all }(\theta,\vartheta)\in\Theta^2. \tag{4}$$

- Our scheme is based on the computation, at each iteration, of stochastic auxiliary functions for a mini-batch of components. For $i\in[\![N]\!]$, the auxiliary function, noted $\hat{f}_{i,\theta}(\vartheta)$ is a Monte Carlo approximation of the surrogate function $f_{i,\theta}(\vartheta)$ defined by (4) such that:

$$\hat{f}_{i,\theta}(\vartheta)\triangleq\frac{1}{M}\sum_{m=0}^{M-1}r_{i,\theta}(z_i^m,\vartheta)\quad\text{for all }(\theta,\vartheta)\in\Theta^2 \tag{5}$$

where $\{z_i^m\}_{m=0}^{M-1}$ is a Monte Carlo batch.

Algorithm 2 MISSO algorithm

Initialization: given an initial parameter estimate θ^0, for all $i\in[\![N]\!]$ compute the function $\vartheta\to\hat{f}_{i,\theta^0}(\vartheta)$ defined by (5).

Iteration k: given the current estimate θ^{k-1}:

1. Pick a set I_k uniformly on $\{A\subset[\![N]\!],\mathrm{card}(A)=p\}$
2. For all $i\in I_k$, sample a Monte Carlo batch $\{z_i^{k,m}\}_{m=0}^{M_k-1}$ from $p_i(z_i,\theta^{k-1})$.
3. For all $i\in I_k$, compute the function $\vartheta\to\hat{f}_{i,\theta^{k-1}}(\vartheta)$ defined by (5).
4. Set $\theta^k\in\arg\min_{\vartheta\in\Theta}\sum_{i=1}^N\hat{a}_i^k(\vartheta)$ where $\hat{a}_i^k(\vartheta)$ are defined recursively as follows:

$$\hat{a}_i^k(\vartheta)\triangleq\begin{cases}\hat{f}_{i,\theta^{k-1}}(\vartheta) & \text{if } i\in I_k\\ \hat{a}_i^{k-1}(\vartheta) & \text{otherwise}\end{cases} \tag{6}$$

Convergence Guarantees Assumptions

Whether we use Markov Chain Monte Carlo or direct simulation, we need to control the supremum norm of the fluctuations of the Monte Carlo approximation. Let $i\in[\![N]\!]$, $\{j_i(z_i,\vartheta),z_i\in\mathsf{Z}_i,\vartheta\in\Theta\}$ be a family of measurable functions, λ_i a probability measure on $\mathsf{Z}_i\times\mathcal{Z}_i$. We define:

$$C_i(j_i)\triangleq\sup_{\theta\in\Theta}\sup_{M>0}M^{-1/2}\mathbb{E}_{i,\theta}\left[\sup_{\vartheta\in\Theta}\left|\sum_{m=0}^{M-1}\left\{j_i(z_i^m,\vartheta)-\int_{\mathsf{Z}_i}j_i(z_i,\vartheta)p_i(z_i,\theta)\lambda_i(\mathrm{d}z_i)\right\}\right|\right] \tag{7}$$

M 3. *For all $i\in[\![N]\!]$ and $\theta\in\Theta$:*

$$\lim_{k\to\infty}C_i(r_{i,\theta})<\infty\quad\textit{and}\quad\lim_{k\to\infty}C_i(\nabla r_{i,\theta})<\infty. \tag{8}$$

M 4. *$\{M_k\}_{k\geq0}$ is a non deacreasing sequence of integers which satisfies $\sum_{k=0}^\infty M_k^{-1/2}<\infty$.*

Theorem: MISSO Convergence Guarantees

Assume **M1-M4**. Let $\left(\theta^k\right)_{k\geq1}$ be a sequence generated from $\theta^0\in\Theta$ by the iterative application described by Algorithm 2. Then:

(i) $\left(f(\theta^k)\right)_{k\geq1}$ converges almost surely.

(ii) $\left(\theta^k\right)_{k\geq1}$ satisfies the Asymptotic Stationary Point Condition.

Application to Variational Bayesian Inference

- Let $x=(x_i,i\in[\![N]\!])$ and $y=(y_i,i\in[\![N]\!])$ be i.i.d. input-output pairs and w be a global latent variable taking values in W as subset of $\mathbb{R}^J$. A natural decomposition of the joint distribution is:

$$p(y,x,w)=p(w)\prod_{i=1}^N p_i(y_i|x_i,w) \tag{9}$$

The goal is to calculate the posterior distribution $p(w|y,x)$.

- Variational inference problem boils down to minimizing the following KL divergence:

$$\theta^*=\arg\min_{\theta\in\Theta}\mathrm{KL}(q(w;\theta)\parallel p(w|y,x))=\arg\min_{\theta\in\Theta}f(\theta) \tag{10}$$

where for all $\theta\in\Theta$, $f(\theta)=\sum_{i=1}^N f_i(\theta)$ with :

$$f_i(\theta)\triangleq-\int_{\mathsf{W}}q(w;\theta)\log p_i(y_i,x_i|w)\mathrm{d}w+\frac{1}{N}\mathrm{KL}(q(w;\theta)\parallel p(w))=r_i(\theta)+d(\theta) \tag{11}$$

- Define following quadratic surrogate at $\theta\in\Theta$:

$$f_{i,\theta}(\vartheta)\triangleq f_i(\theta)+\nabla f_i(\theta)^\top(\vartheta-\theta)+\frac{L}{2}\|\vartheta-\theta\|_2^2 \tag{12}$$

where $\|\cdot\|_2$ is the ℓ_2-norm and L is an upper bound of the spectral norm of the Hessian of f_i at θ.

- **Reparametrization trick:** We assume that for all $\theta\in\Theta$, the distribution of the random vector $W=t(\theta,\epsilon)$ where $\epsilon\sim\mathcal{N}_d(0,\mathrm{Id})$ has a density $q(\cdot,\theta)$. Then, following (Proposition 1)blundell:

$$\nabla\int_{\mathsf{W}}\log p_i(y_i,x_i|w)q(w,\theta)\mathrm{d}w=\int_{\mathsf{W}}\mathrm{J}(\theta,e)\nabla\log p_i(y_i,x_i|t(\theta,e))\phi(e)\mathrm{d}e$$

where for each $e\in\mathbb{R}^d$, $\mathrm{J}(\theta,e)$ is the Jacobian of the function $t(\cdot,e)$ with respect to θ.

- The pair $\left(r_{i,\theta}(e,\vartheta),\phi(e)\right)$ defining $f_{i,\theta}(\vartheta)$ is given by:

$$\begin{aligned}r_{i,\theta}(e,\vartheta)\triangleq&(-\log p_i(y_i,x_i|t(\theta,e))+d(\theta))\\&+(-\mathrm{J}(\theta,e)\nabla\log p_i(y_i,x_i|t(\theta,e))+\nabla d(\theta))^\top(\vartheta-\theta)+\frac{L}{2}\|\vartheta-\theta\|_2^2\end{aligned} \tag{13}$$

The MISSO algorithm consists in:

1. Picking a set I_k uniformly on $\{A\subset[\![N]\!],\mathrm{card}(A)=p\}$.
2. Sampling a Monte Carlo batch $\{\epsilon^{k,m}\}_{m=0}^{M_k-1}$ from the standard Gaussian distribution.
3. Setting $\theta^k=\theta^{k-1}-\frac{1}{L}\sum_{i=1}^N\hat{a}_i^k$ where $\hat{a}_i^k$ are defined recursively as follows:

$$\hat{a}_i^k\triangleq\begin{cases}-\frac{1}{M_k}\sum_{m=0}^{M_k-1}\mathrm{J}(\theta,e^{k,m})\nabla_\theta\log p_i(y_i,x_i|t(\theta,e^{k,m}))+\nabla d(\theta^{k-1}) & \text{if } i\in I_k\\ \hat{a}_i^{k-1} & \text{otherwise}\end{cases} \tag{14}$$

Training a Bayesian Neural Network on MNIST

Settings

- 2-layer bayesian neural network
- Tanh activation function
- Standard Gaussian prior on the weight
- Gaussian variational posterior independent of i and l (layers)

$$p(w)=\mathcal{N}(0,\mathrm{Id})$$
$$p(y_i|x_i,w)=\mathrm{Softmax}(f(x_i,w))$$

- Input layer $d=784$
- A single hidden layer of $p=100$ hyperbolic tangent units
- Final softmax output layer with $K=10$ classes
- MNIST dataset $N=60\,000$

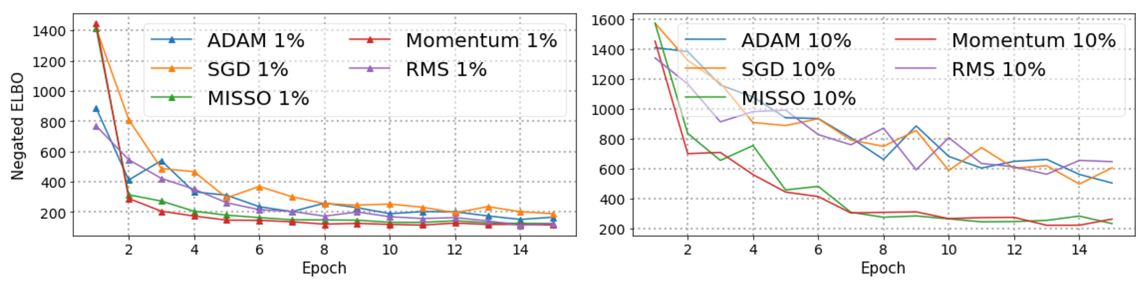

Figure 1: ELBO convergence.

References

(pai())

(Blundell et al.(2015)Blundell, Cornebise, Kavukcuoglu, and Wierstra) C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32Nd International Conference on ICML - Volume 37*, ICML'15, pages 1613–1622. JMLR.org, 2015.

(Gal(2016)) Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

(Le Roux et al.(2012)Le Roux, Schmidt, and Bach) N. Le Roux, M.W. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pages 2672–2680, 2012.

(Mairal(2015)) J. Mairal. Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning. *SIAM*, 2015.

(Ranganath et al.(2014)Ranganath, Gerrish, and Blei) R. Ranganath, S. Gerrish, and D. Blei. Black Box Variational Inference. *PMLR*, 33:814–822, 2014.