Minimization by Incremental Stochastic Surrogate with Application to Bayesian Deep Learning

B. Karimi1,2, E. Moulines2
INRIA1, CMAP École Polytechnique2
belhal.karimi@polytechnique.edu

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

$$\min_{\mathbf{\theta}} \left\{ f_{\mathbf{\theta}} \right\} \triangleq \sum_{i=1}^{N} f_i(\mathbf{\theta}) $$

(1)

Theorem: MISSO Convergence Guarantees
Assume $M \geq M_1$. Let $(\mathbf{\theta}^t)_{t \geq 1}$ be a sequence generated from $\mathbf{\theta}^0 \in \Theta$ by the iterative algorithm described by Algorithm 1. Then:

(a) $(\mathbf{\theta}^t)_{t \geq 1}$ converges almost surely.

(b) $(\mathbf{\theta}^t)_{t \geq 1}$ satisfies the asymptotic stationary point condition.

Application to Variational Bayesian Inference

• Let $x = (x_i \in \mathbb{N})$ and $y = (y_i \in \mathbb{N})$ be i.i.d., input-output pairs and $z$ be a global latent variable taking values in W as subset of $\mathbb{R}^2$. A natural decomposition of the joint distribution is:

$$p(x, z, y) = p(z) \prod_i p(x_i | z_i, y_i)$$

(9)

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

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

$$\mathcal{D}(\mathfrak{p}\|\mathfrak{q}) = \log \frac{\mathfrak{p}(\mathbf{z})}{\mathfrak{q}(\mathbf{z})} = \int_{\mathbf{z}} \log \frac{\mathfrak{p}(\mathbf{z})}{\mathfrak{q}(\mathbf{z})} p(x_i, y_i) z_i \mathbf{d}z_i$$

(10)

where for all $\mathbf{\theta} \in \Theta$,

$$f_{\mathbf{\theta}}(\mathfrak{q}) \triangleq \int \log q(z_i | x_i, y_i) p(x_i, y_i) \mathbf{d}z_i + \frac{1}{2} \sum_{i=1}^{N} \mathbb{E}_{q(z_i | x_i, y_i)} \left[ \mathbb{V} q(z_i | x_i, y_i) \right] $$

(11)

• Define following quadradic surrogate at $\mathfrak{q}$ if:

$$f_{\mathbf{\theta}}(\mathfrak{q}) = f_{\mathbf{\theta}}(\mathfrak{q}) + \frac{1}{2} \left( \mathfrak{q} - \mathfrak{q}_{\mathbf{\theta}} \right)^{\top} \mathfrak{K} \left( \mathfrak{q} - \mathfrak{q}_{\mathbf{\theta}} \right)$$

(12)

where $\mathfrak{K}$ is the $\ell_1$-norm and $\mathbb{K}$ is an upper bound of the spectral norm of the Hessian of $f_{\mathbf{\theta}}$.

Convergence Guarantees Assumptions

- We use pMatovtChae Monte Carlo or direct simulation, we need to control the norm summaries of the fluctuations of the Monte Carlo approximation. Let $f_i(\mathbf{\theta})$ be a family of measurable functions, $\lambda_i$ a probability measure on $\mathbb{R} \times Z_i$, we define:

$$C_i(\mathbf{\theta}) \triangleq \sup_{\mathbf{\theta}} \mathbb{E}_{\lambda_i} \left[ \sum_{M=1}^{\infty} \beta_{M,0} f_i(\mathbf{\theta}) \right]$$

(7)

$$\lim_{M \to \infty} C_i(\mathbf{\theta}) < \infty \quad \text{and} \quad \lim_{M \to \infty} C_i(\mathbb{V}) < \infty$$

(8)

References


[Roux et al. (2021)] S. Roux, S. Reid, and B. Bach. The Jacobian of the function $h : \mathbb{R} \rightarrow \mathbb{R}$ with respect to $x$.

The pair $(r_{\mathbf{\theta}}(x, e), \mathbf{\theta}(x, e))$ defining $f_{\mathbf{\theta}}(\mathfrak{q})$ is given by:

$$r_{\mathbf{\theta}}(x, e) \triangleq \left( -\log q(x_i | z_i, y_i) + d \mathbf{q}_{\mathbf{\theta}} \right)$$

(13)

The MISSO algorithm consists in:

1. Picking a set $J_0$ uniformly on $(A \subset \mathbb{N})$ such that $|J_0| = \eta p$.

2. Sampling a Monte Carlo batch $(z_{i,J_0}, x_{i,J_0}, y_{i,J_0})$ from the standard Gaussian distribution.

3. Setting $\mathfrak{q}^{(0)} = \arg \min_{\mathfrak{q} \in \Theta} \sum_{i \notin J_0} f_{\mathbf{\theta}}(\mathfrak{q})$ where $\mathfrak{q}$ are defined recursively as follows:

$$\hat{q}_{\mathbf{\theta}}(z_i | x_i, y_i) \triangleq \frac{1}{\mathbb{E}_{q(z_i | x_i, y_i)} \left[ \mathbb{V} q(z_i | x_i, y_i) \right]} q(z_i | x_i, y_i)$$

(14)

Training a Bayesian Neural Network on MNIST

• 2-layer Bayesian neural network

• Tanh activation function

• Standard Gaussian prior on the weights

• Gaussian variational posterior independent of $i$ and $l$ (layers)

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.

Table 1: Comparison of ADAM, Momentum, SGD, RMS, and MISSO on MNIST.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>ADAM 1%</th>
<th>Momentum 1%</th>
<th>SGD 1%</th>
<th>RMS 1%</th>
<th>MISSO 1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.85</td>
<td>0.84</td>
<td>0.83</td>
<td>0.82</td>
<td>0.85</td>
</tr>
<tr>
<td>Loss</td>
<td>0.15</td>
<td>0.16</td>
<td>0.17</td>
<td>0.19</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Figure 1: ELBO convergence.