Low-Cost Lipschitz-Independent Adaptive Importance Sampling of Stochastic Gradients

Xiaolu Wang

Presentation for ICPR 2020

Joint Work with: Huikang Liu, Jiajin Li, Anthony Man-Cho So

Department of Systems Engineering and Engineering Management

The Chinese University of Hong Kong

December 17, 2020

Introduction

Consider the following empirical risk minimization (ERM) problem

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

- It finds many applications in machine learning and pattern recognition;
- $f_i(\mathbf{w}) = \ell(h(\mathbf{x}_i;\mathbf{w}), \mathbf{y}_i)$ where $(\mathbf{x}_i, \mathbf{y}_i)$ is the i-th training example;
  - h is the decision function parameterized by $\mathbf{w}$;
  - ℓ is the loss function.
- SGD plays a central role in solving optimization problem (1):

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \nabla f_{i_k}(\mathbf{w}_k), \tag{2}$$

Adaptive Importance Sampling

To control the variance of the stochastic gradient, SGD with adaptive importance sampling is introduced:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{\eta_k}{n p_{i_k}^k} \nabla f_{i_k}(\mathbf{w}_k), \tag{3}$$

where $\boldsymbol{p}^k := (p_1^k, p_2^k, \ldots, p_n^k)^\top$ is the importance sampling distribution.
A natural idea of choosing distribution $\boldsymbol{p}^k$ is to minimize the variance

$$\min_{\boldsymbol{p}^k} \text{Var}\left[\frac{1}{n p_i^k} \nabla f_i(\mathbf{w}_k)\right] \quad \text{s.t.} \sum_{j=1}^n p_j^k = 1,\ p_i^k \geq 0,\ \forall i \in \{1, \ldots, n\}. \tag{4}$$

Problem (4) has a closed-form optimal solution, which is

$$(p_i^k)^* = \frac{\|\nabla f_i(\mathbf{w}_k)\|_2}{\sum_{j=1}^n \|\nabla f_j(\mathbf{w}_k)\|_2},\ \forall i \in \{1, \ldots, n\}. \tag{5}$$

Question: How to compute the optimal sampling distribution?
Key Idea: Using the most recently evaluated gradient norms $\|\nabla f_i(\mathbf{w}_{k'})\|_2$ to approximate $\|\nabla f_i(\mathbf{w}_k)\|_2$.

SGD-AIS Algorithm

Algorithm 1 SGD-AIS

1: **Input:** step sizes $\{\eta_k\}$, weights $\alpha_k \in [\underline{\alpha}, \overline{\alpha}] \subset (0,1)$ for all $k \in \mathbb{N}$.
2: **Initialize:** $\mathbf{w}_1$, $\pi_i = 1$ for all $i \in \{1, \ldots, n\}$
3: **for** $k = 1, , 2 \ldots$ **do**
4: Update the sampling probabilities for all $i \in \{1, \ldots, n\}$

$$p_i = \alpha_k \frac{\pi_i}{\sum_{j=1}^{n} \pi_j} + (1 - \alpha_k)\frac{1}{n} \tag{6}$$

5: Randomly pick $i_k \in [n]$ based on distribution $\boldsymbol{p}$
6: Compute stochastic gradient $\mathbf{g}_k = \frac{1}{np_{i_k}} \nabla f_{i_k}(\mathbf{w}_k)$
7: Set $\pi_{i_k} = \|\nabla f_{i_k}(\mathbf{w}_k)\|_2$
8: Set $\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \mathbf{g}_k$
9: **end for**

Complexity: By resorting to a binary tree data structure, only additional $O(\log n)$ per-iteration cost is needed to implement the adaptive sampling.

SGDm-AIS and ADAM-AIS

Applying AIS strategy to SGD with momentum, we just need $\mathbf{g}_k$ to be

$$\mathbf{g}_k = \theta \mathbf{g}_{k-1} + (1-\theta)\frac{1}{np_{i_k}^k}\nabla f_{i_k}(\mathbf{w}_k). \tag{7}$$

Applying AIS strategy to ADAM, we just need $\mathbf{g}_k$ to be

$$\mathbf{g}_k = \frac{\hat{\mathbf{m}}_k}{\sqrt{\hat{\mathbf{h}}_k} + \varepsilon}, \tag{8}$$

where

$$\hat{\mathbf{m}}_k = \left(\theta_1 \mathbf{m}_{k-1} + (1-\theta_1)\frac{1}{np_{i_k}^k}\mathbf{g}_k\right) \Big/ (1-\theta_1^k), \tag{9}$$

and

$$\hat{\mathbf{h}}_k = \left(\theta_2 \mathbf{h}_{k-1} + (1-\theta_2)\frac{1}{np_{i_k}^k}\mathbf{g}_k^2\right) \Big/ (1-\theta_2^k). \tag{10}$$

Theoretical Results

Theorem

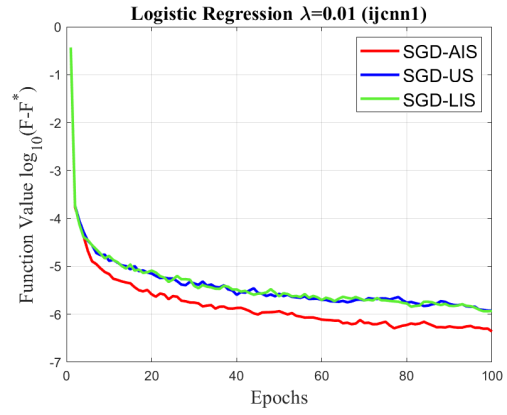
Under some mild assumptions, the sequence $\{\mathbf{w}_k\}$ *generated by SGD-AIS with a fixed stepsize* $\eta_k = \eta$ *for all* $k \in \mathbb{N}$ *satisfying*

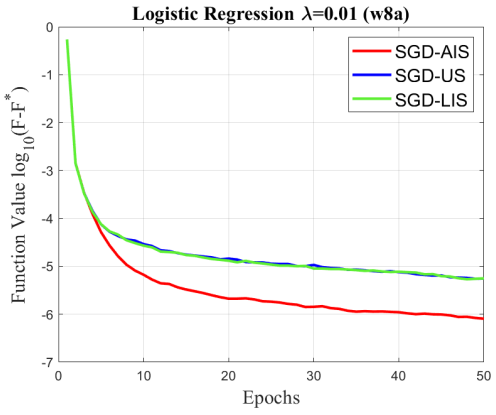
$$\begin{aligned}\mathbb{E}[F(\mathbf{w}_k) - F^*] \leq & \frac{\eta L(1-\gamma)G^2}{4\sigma} + (1-2\eta\sigma)^{k-1}\left(F(\mathbf{w}_1) - F^*\right) \\ & \xrightarrow{k\to\infty} \frac{\eta L(1-\gamma)G^2}{4\sigma}.\end{aligned} \tag{11}$$

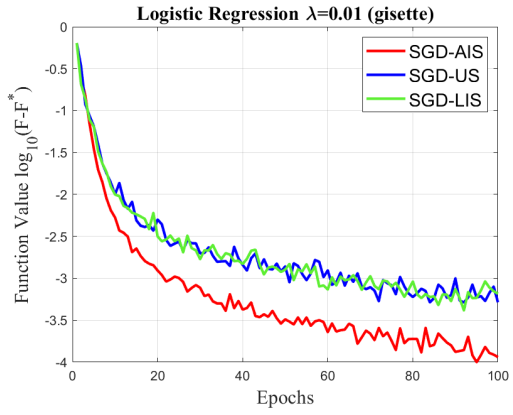
- Compared with vanilla SGD, the convergence bounds of SGD-AIS are improved by a factor of $1-\gamma < 1$;
- Similar improvement still holds if we choose diminishing stepsize;
- We also provide more convergence analysis under the nonconvex settings.

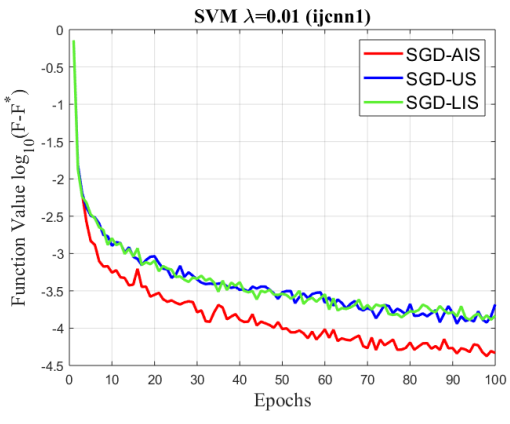
SGD for Logistic Regression and SVM

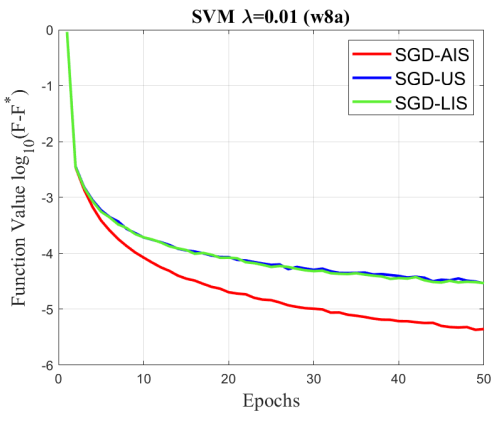
We implement three algorithms, which are SGD-AIS, SGD with uniform sampling (SGD-US), SGD with Lipschitz-based importance sampling (SGD-LIS) for performance comparison.

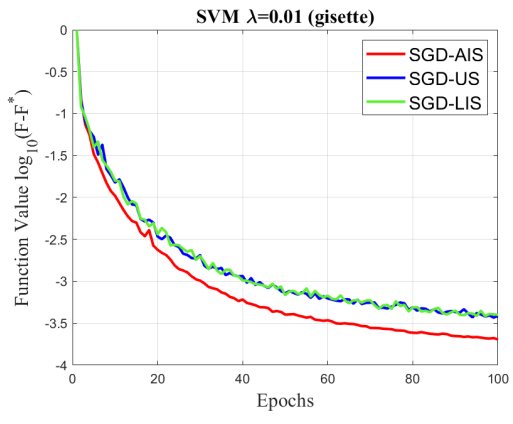






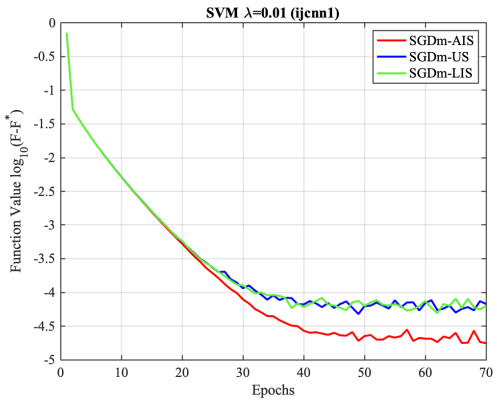


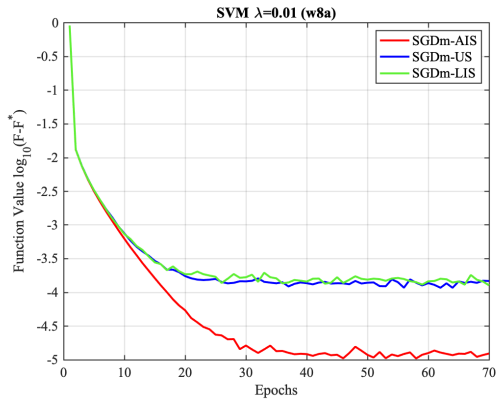


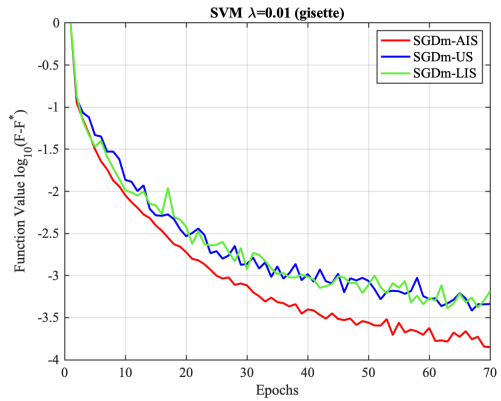


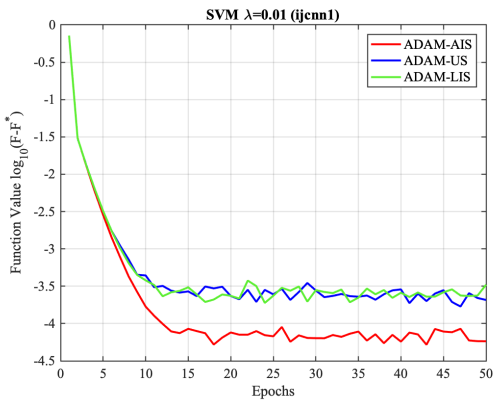
SGDm and ADAM for SVM

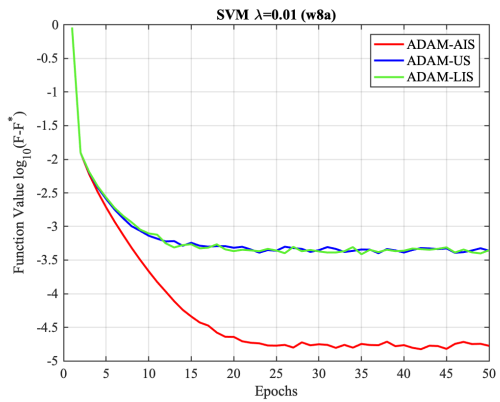
We conduct experiments on SVM with squared hinge loss to evaluate the performance of SGDm-AIS and ADAM-AIS.

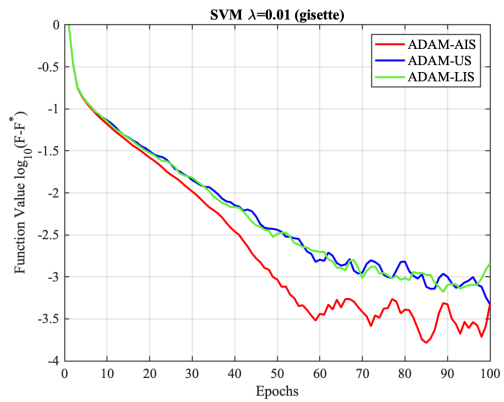












SGDm and ADAM for Neural Networks

We further conduct simulation on MLP, CNN and LeNet-5, and use two common benchmark datasets, namely MNIST and CIFAR-10.

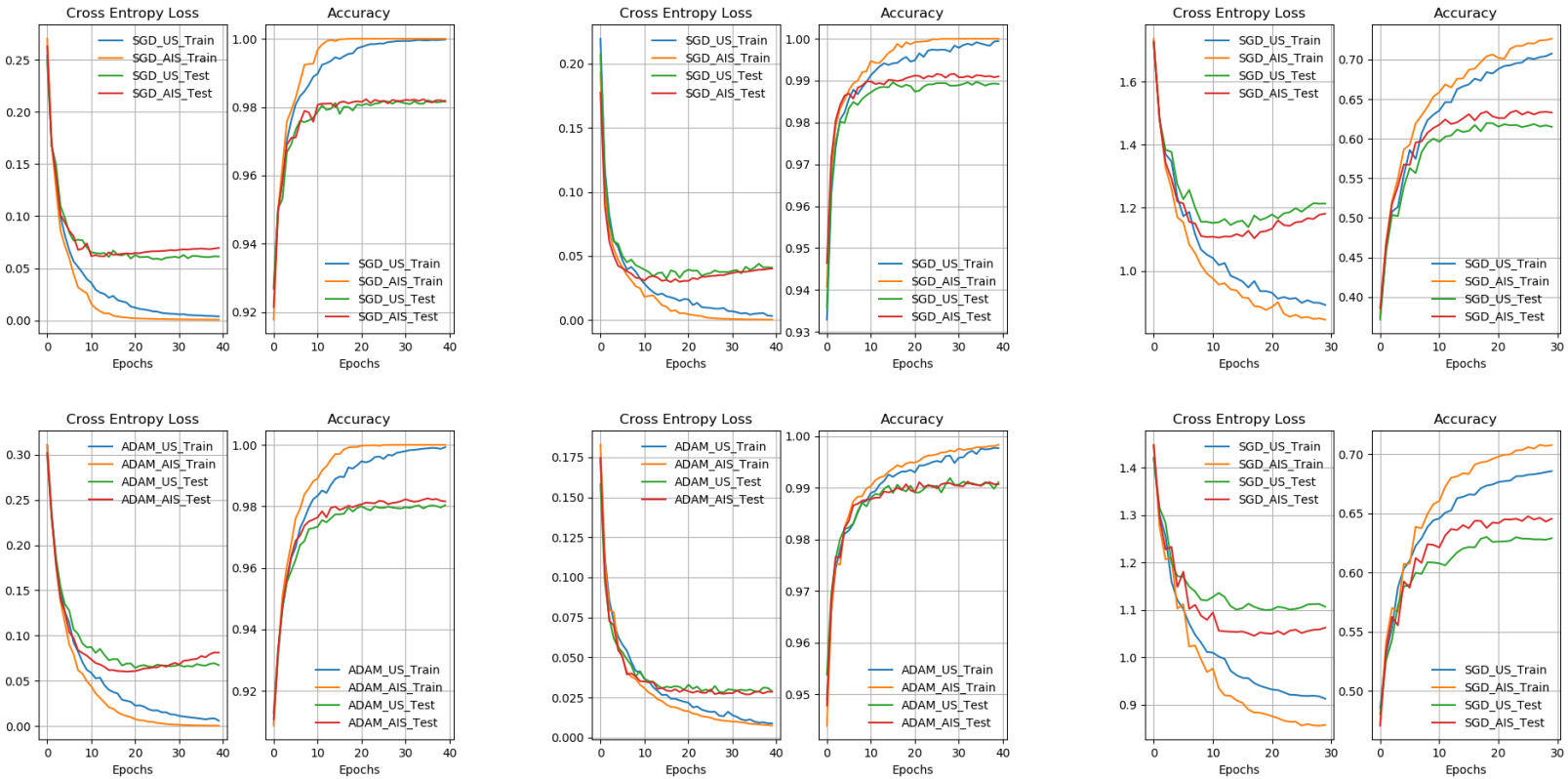


Figure: Column 1-3: MLP (MNIST), LeNet-5 (MNIST), CNN (Cifar-10); Row 1: SGD-US v.s. SGD-AIS; Row 2: ADAM-US v.s. ADAM-AIS

Thank You!