Training Deep Models Faster with Robust, Approximate Importance Sampling

Tyler B. Johnson; and Carlos Guestrin

Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)

Deep learning, Importance sampling
https://dl.acm.org/doi/10.5555/3327757.3327829

Abstract

In theory, importance sampling speeds up stochastic gradient algorithms for supervised learning by prioritizing training examples. In practice, the cost of computing importances greatly limits the impact of importance sampling. We propose a robust, approximate importance sampling procedure (RAIS) for stochastic gradient descent. By approximating the ideal sampling distribution using robust optimization, RAIS provides much of the benefit of exact importance sampling with drastically reduced overhead. Empirically, we find RAIS-SGD and standard SGD follow similar learning curves, but RAIS moves faster through these paths, achieving speed-ups of at least 20% and sometimes much more.

Problem Statement and Research Objectives

- **Importance sampling** prioritizes training examples for SGD in a principled way. The technique suggests sampling example i with probability proportional to the norm of loss term i's gradient. This distribution both prioritizes challenging examples and minimizes the stochastic gradient's variance.
  - Some previous studies analyze importance sampling for SGD and convex problems.
    - But practical versions of these algorithms sample **proportional to fixed constants**.
  - For deep models, other algorithms **attempt closer approximations of gradient norms**.
    - But these algorithms are **not inherently robust**. Without carefully chosen hyperparameters or additional forward passes, these algorithms do not converge, let alone speed up training.

Proposed Method

We propose RAIS, an importance sampling procedure for SGD with several appealing qualities.

- First, RAIS determines each sampling distribution by solving a robust optimization problem. As a result, each sampling distribution is minimax optimal with respect to an uncertainty set. Since RAIS trains this uncertainty set in an adaptive manner, RAIS is not sensitive to hyperparameters.
- In addition, RAIS maximizes the benefit of importance sampling by adaptively increasing SGD's learning rate. Interestingly, when plotted in terms of "epochs equivalent," the learning curves of the algorithms align closely.
- RAIS applies to any model that is trainable with SGD. RAIS also combines nicely with standard "tricks," including data augmentation, dropout, and batch normalization.

O-SGD (SGD with "*oracle*" importance sampling) vs. **U-SGD** (SGD with uniform sampling)

- samples training examples non-uniformly in a way that minimizes the variance of the stochastic gradient. (This is not new.)
  - Training examples with largest gradient norm are most important for further decreasing F, and these examples receive priority.
- to compensate for the first improvement, O-SGD adaptively increases the learning rate.
  - Because importance sampling reduces the stochastic gradient's variance—possibly by a large amount— we find it important to adaptively increase O-SGD's learning rate compared to U-SGD.
  - we propose a learning rate that depends on the "gain ratio" $r_O^{(t)} \in \mathbb{R}_{\geq 1}$:
    $r_O^{(t)} = \mathbb{E}\left[\|g_U^{(t)}\|^2\right] \ / \ \mathbb{E}\left[\|g_O^{(t)}\|^2\right]$.
    - U-SGD with learning rate in iteration t : $\eta_U^{(t)} = \text{lr\_sched}(t)$
    - O-SGD with learning rate in iteration t : $\eta_O^{(t)} = r_O^{(t)} \eta_U^{(t)}$

Robust approximate importance sampling (RAIS)

$\mathcal{M}^{(t)}$: minibatch
$\mathbf{p}^{(t)}$: discrete distribution of samples
$\mathbf{g}_R^{(t)}$: stochastic gradient
$v_i^* = \|\nabla f_i(w^{(t')})\|, \mathbf{v}^* = \left[v_1^*, v_2^*, \cdots, v_n^*\right]^T$

1. Determining a robust sampling distribution

- RAIS defines $\mathbf{p}^{(t)}$ by approximating $\mathbf{v}^*$
  - Naive algorithms approximate $\mathbf{v}^*$ using a **point estimate $\hat{\mathbf{v}}$**
    ➔ **drawback**: extreme sensitivity to differences between $\hat{\mathbf{v}}$ and $\mathbf{v}^*$.
  - Instead of point estimate, RAIS approximates $\mathbf{v}^*$ with uncertainty set $\mathcal{U}^{(t)} \subset \mathbb{R}^n_{\geq 0}$, which we expect contains (or nearly contains) $\mathbf{v}^*$.
  - Given $\mathcal{U}^{(t)}$, RAIS defines $\mathbf{p}^{(t)}$ **by minimizing the worst-case value** of $\mathbb{E}\left[\|g_R^{(t)}\|^2\right]$ over **all gradient norm possibilities in $\mathcal{U}^{(t)}$**

2. Modeling the uncertainty set

- For each example i, we define a feature vector $s_i^{(t)} \in \mathbb{R}^{d_R}_{\geq 0}$. (A useful feature for $s_i^{(t)}$ is the gradient norm, $\|\nabla f_i(w^{(t')})\|$)
- Given $s_i^{(t)}$ for all examples, RAIS defines the uncertainty set as an **axis-aligned ellipsoid**.
  - RAIS **parameterizes this uncertainty set** with two vectors, $\mathbf{c} \in \mathbb{R}^{d_R}_{\geq 0}$ and $\mathbf{d} \in \mathbb{R}^{d_R}_{\geq 0}$.
    ➔ **parameters of the ellipsoid**

$$v_i^{(t)} = \langle \mathbf{c}, \mathbf{s}_i(t) \rangle + k \langle \mathbf{d}, \mathbf{s}_i(t) \rangle$$

3. Learning the uncertainty set

In order to make $\mathbb{E}\left[\|g_R^{(t)}\|^2\right]$ small but still ensure $\mathbf{v}^*$ likely lies in $\mathcal{U}_{\mathbf{cd}}^{(t)}$, RAIS adaptively define $\mathbf{c}$ and $\mathbf{d}$. To do so, RAIS minimizes the size of $\mathcal{U}_{\mathbf{cd}}^{(t)}$ subject to a constraint that encourages $\mathbf{v}^* \in \mathcal{U}_{\mathbf{cd}}^{(t)}$:

$$\mathbf{c}, \mathbf{d} = \arg\inf\{\sum_{i=1}^{n} \left\langle \mathbf{d}, \mathbf{s}_i^{(t)} \right\rangle \mid \mathbf{c}, \mathbf{d} \in \mathbb{R}^{d_R}_{\geq 0}, \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \tilde{w}_i Q_{\mathbf{cd}}(\tilde{s}_i, \tilde{v}_i) \leq 1\} \qquad \mathbf{(PT)}$$

4. Approximating the gain ratio

- In addition to the sampling distribution, RAIS must approximate the gain ratio in O-SGD. Define $g_{R1}^{(t)}$ as a stochastic gradient of the form (4) using minibatch size 1 and RAIS sampling. Define $g_{U1}^{(t)}$ in the same way but with uniform sampling. From (5), we can work out that the gain ratio satisfies

$$\mathbb{E}\left[\|g_U^{(t)}\|^2\right] \ / \ \mathbb{E}\left[\|g_R^{(t)}\|^2\right] = 1 + \frac{1}{|\mathcal{M}|}\left(\mathbb{E}\left[\|g_{U1}^{(t)}\|^2\right] \ / \ \mathbb{E}\left[\|g_{R1}^{(t)}\|^2\right]\right) / \mathbb{E}\left[\|g_R^{(t)}\|^2\right]$$

- To approximate the gain ratio, RAIS estimates the three moments on the right side of this equation. RAIS estimates $\mathbb{E}\left[\|g_R^{(t)}\|^2\right]$ using an exponential moving average of $\|g_R^{(t)}\|^2$ from recent iterations:

$$\mathbb{E}\left[\|g_R^{(t)}\|^2\right] \approx \alpha \left[\|g_R^{(t)}\|^2 + (1-\alpha)\|g_R^{(t-1)}\|^2 + (1-\alpha)^2\|g_R^{(t-2)}\|^2 + \cdots\right]$$

Algorithm 4.1 RAIS-SGD

input objective function F, minibatch size $|\mathcal{M}|$, learning rate schedule `lr_sched`$(\cdot)$
input RAIS training set size $|\mathcal{D}|$, exponential smoothing parameter α for gain estimate
initialize $\mathbf{w}^{(1)} \in \mathbb{R}^d$, $\mathbf{c}, \mathbf{d} \in \mathbb{R}^{d_R}_{\geq 0}$; $\hat{t}^{(1)} \leftarrow 1$; `r_estimator` $\leftarrow$ `GainEstimator`(α)
for $t = 1, 2, \ldots, T$ **do**
  $\mathbf{v}^{(t)} \leftarrow \operatorname{argmax}\{ (\sum_{i=1}^n v_i)^2 \mid \mathbf{v} \in \mathcal{U}_{\mathbf{cd}}^{(t)}\}$ *# see Proposition 4.1 for closed-form solution*
  $\mathbf{p}^{(t)} \leftarrow \mathbf{v}^{(t)} / \|\mathbf{v}^{(t)}\|_1$
  $\mathcal{M}^{(t)} \leftarrow$ `sample_indices_from_distribution`$(\mathbf{p}^{(t)}$, `size` $= |\mathcal{M}|)$
  $\mathbf{g}_{\mathrm{R}}^{(t)} \leftarrow \frac{1}{|\mathcal{M}|}\sum_{i \in \mathcal{M}^{(t)}} \frac{1}{np_i^{(t)}} \nabla f_i(\mathbf{w}^{(t)}) + \lambda \mathbf{w}^{(t)}$
  `r_estimator.record_gradient_norms`$(\|\mathbf{g}_{\mathrm{R}}^{(t)}\|, (\|\nabla f_i(\mathbf{w}^{(t)})\|, p_i^{(t)})_{i \in \mathcal{M}^{(t)}})$
  $\hat{r}^{(t)} \leftarrow$ `r_estimator.estimate_gain_ratio()`   *# see §4.4*
  $\eta^{(t)} \leftarrow \hat{r}^{(t)} \cdot$ `lr_sched`$(\hat{t}^{(t)})$
  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta^{(t)} \mathbf{g}_{\mathrm{R}}^{(t)}$
  $\hat{t}^{(t+1)} \leftarrow \hat{t}^{(t)} + \hat{r}^{(t)}$
  if $\operatorname{mod}(t, \lceil |\mathcal{D}|/|\mathcal{M}| \rceil) = 0$ **and** $t \geq (n + |\mathcal{D}|)/|\mathcal{M}|$ **then**
    $\mathbf{c}, \mathbf{d} \leftarrow$ `train_uncertainty_model()`   *# see §4.2*
return $\mathbf{w}^{(T+1)}$

5. Practical considerations

- Solving $\mathbf{(PT)}$
  - RAIS should not solve $\mathbf{(PT)}$ during every iteration. Our implementation solves $\mathbf{(PT)}$ asynchronously after every $\lceil |\mathcal{D}|/|\mathcal{M}| \rceil$ minibatches, with updates to $w^{(t)}$ continuing during the process.
- Approximating per-example gradient norms
  - Unfortunately, existing software tools do not provide efficient access to per-example gradient norms. Instead, libraries are optimized for aggregating gradients over minibatches.
  - We do so by replacing $\|\nabla f_i(w^{(t)})\|$ with the norm of only the loss layer's gradient (with respect to this layer's inputs).

Evaluation and Results

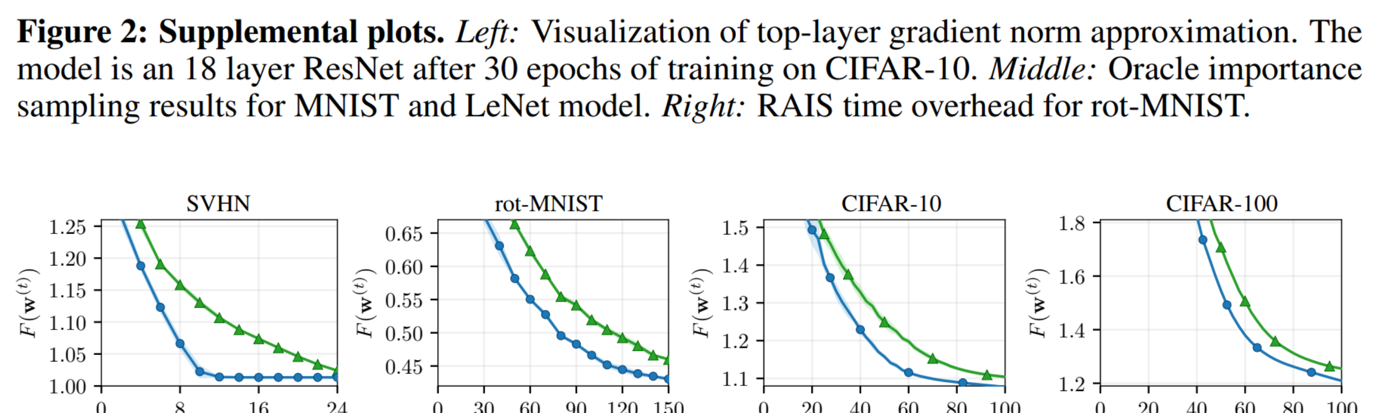

Figure 2: Supplemental plots. *Left:* Visualization of top-layer gradient norm approximation. The model is an 18 layer ResNet after 30 epochs of training on CIFAR-10. *Middle:* Oracle importance sampling results for MNIST and LeNet model. *Right:* RAIS time overhead for rot-MNIST.

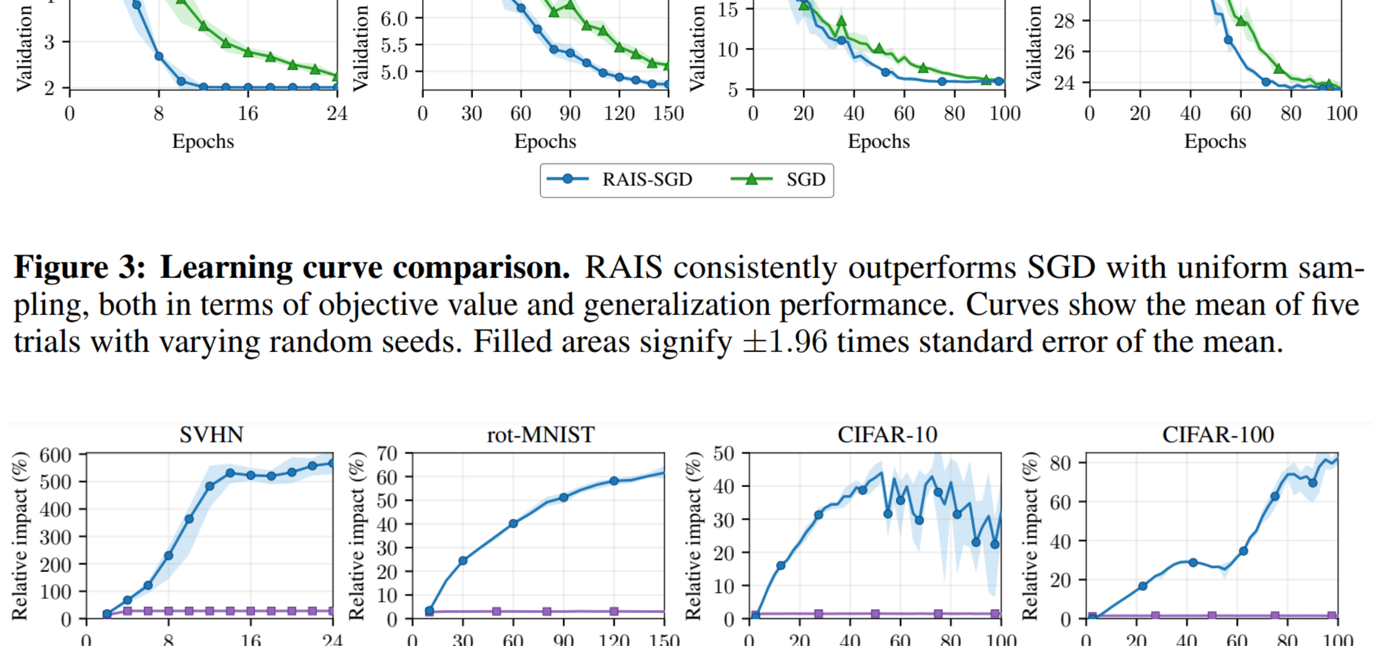

Figure 3: Learning curve comparison. RAIS consistently outperforms SGD with uniform sampling, both in terms of objective value and generalization performance. Curves show the mean of five trials with varying random seeds. Filled areas signify ± 1.96 times standard error of the mean.

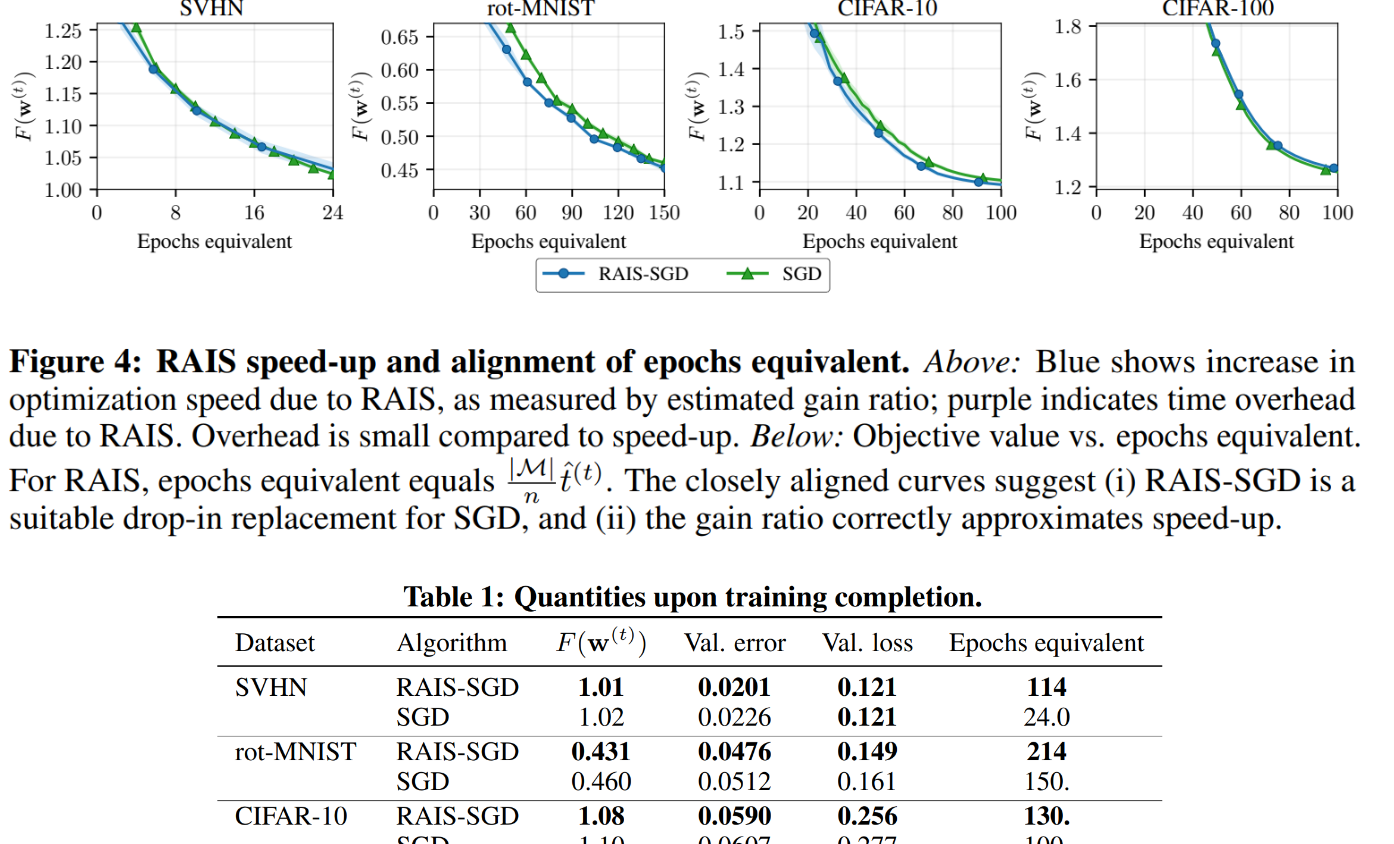

Figure 4: RAIS speed-up and alignment of epochs equivalent. *Above:* Blue shows increase in optimization speed due to RAIS, as measured by estimated gain ratio; purple indicates time overhead due to RAIS. Overhead is small compared to speed-up. *Below:* Objective value vs. epochs equivalent. For RAIS, epochs equivalent equals $\frac{|\mathcal{M}|}{n}\hat{t}^{(t)}$. The closely aligned curves suggest (i) RAIS-SGD is a suitable drop-in replacement for SGD, and (ii) the gain ratio correctly approximates speed-up.

Table 1: Quantities upon training completion.

Dataset	Algorithm	$F(\mathbf{w}^{(t)})$	Val. error	Val. loss	Epochs equivalent
SVHN	RAIS-SGD	**1.01**	**0.0201**	**0.121**	**114**
	SGD	1.02	0.0226	**0.121**	24.0
rot-MNIST	RAIS-SGD	**0.431**	**0.0476**	**0.149**	**214**
	SGD	0.460	0.0512	0.161	150.
CIFAR-10	RAIS-SGD	**1.08**	**0.0590**	**0.256**	**130.**
	SGD	1.10	0.0607	0.277	100.
CIFAR-100	RAIS-SGD	**1.21**	**0.236**	**0.962**	**138**
	SGD	1.25	**0.236**	0.989	100.

Notes

- Given $w^{(t)}$, let us define the expected training progress attributable to iteration t as
  $$\begin{aligned}\mathbb{E}\triangle^{(t)} &= \|w^{(t)} - w^*\|^2 - \mathbb{E}[\|w^{(t+1)} - w^*\|^2] \\ &= 2\eta^{(t)}\langle \nabla F(w^{(t)}), w^{(t)} - w^* \rangle - [\eta^{(t)}]^2 \mathbb{E}[\|g^{(t)}\|^2]\end{aligned}$$
  (to increase $\mathbb{E}\triangle^{(t)}$)
  - Conventionally use RMSE(Root Mean Squared Error) like L2 norm