Advanced stochastic methods. Variance reduction

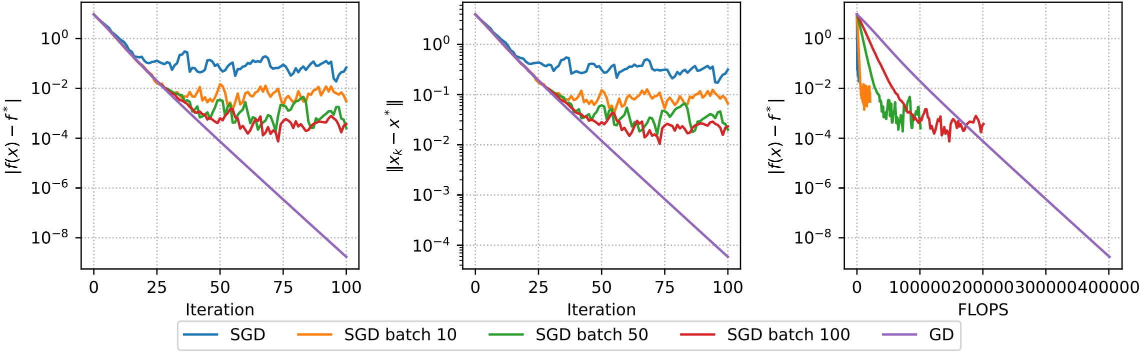
Seminar

Optimization for ML. Faculty of Computer Science. HSE University

Main problem of SGD

$$
f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}
$$

Strongly convex binary logistic regression. m=200, n=10, mu=1.



Key idea of variance reduction

Principle: reducing variance of a sample of X by using a sample from another random variable Y with known expectation:

$$
Z_\alpha = \alpha(X - Y) + \mathbb{E}[Y]
$$

- $\mathbb{E}[Z_\alpha] = \alpha\mathbb{E}[X] + (1-\alpha)\mathbb{E}[Y]$
- $\text{var}(Z_\alpha) = \alpha^2\left(\text{var}(X) + \text{var}(Y) - 2\text{cov}(X, Y)\right)$
    - If $\alpha = 1$: no bias
    - If $\alpha < 1$: potential bias (but reduced variance).
- Useful if Y is positively correlated with X.

Application to gradient estimation ?

- SVRG: Let $X = \nabla f_{i_k}(x^{(k-1)})$ and $Y = \nabla f_{i_k}(\tilde{x})$, with $\alpha = 1$ and $\tilde{x}$ stored.
- $\mathbb{E}[Y] = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{x})$ full gradient at $\tilde{x}$;
- $X - Y = \nabla f_{i_k}(x^{(k-1)}) - \nabla f_{i_k}(\tilde{x})$

SVRG (Stochastic Variance Reduced gradient) [1]

- **Initialize:** $\tilde{x} \in \mathbb{R}^d$
- **For** $i_{epoch} = 1$ **to** `# of epochs`
    - Compute all gradients $\nabla f_i(\tilde{x})$; store $\nabla f(\tilde{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{x})$
    - Initialize $x_0 = \tilde{x}$
    - **For** `t = 1` **to** `length of epochs (m)`
        - $x_t = x_{t-1} - \alpha \left[\nabla f(\tilde{x}) + \left(\nabla f_{i_t}(x_{t-1}) - \nabla f_{i_t}(\tilde{x})\right)\right]$
        - Update $\tilde{x} = x_t$

Notes:

- Two gradient evaluations per inner step.
- Two parameters: length of epochs + step-size γ.
- Linear convergence rate, simple proof.

[1] Johnson, R., Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. Advances in neural information processing systems, 26.

SAG (Stochastic average gradient) [2]

- Maintain table, containing gradient g_i of f_i, $i = 1, \dots, n$
- Initialize $x^{(0)}$, and $g_i^{(0)} = \nabla f_i(x^{(0)})$, $i = 1, \dots, n$
- At steps $k = 1, 2, 3, \dots$, pick random $i_k \in \{1, \dots, n\}$, then let

$$g_{i_k}^{(k)} = \nabla f_{i_k}(x^{(k-1)}) \quad \text{(most recent gradient of } f_{i_k}\text{)}$$

  Set all other $g_i^{(k)} = g_i^{(k-1)}$, $i \neq i_k$, i.e., these stay the same
- Update

$$x^{(k)} = x^{(k-1)} - \alpha_k \frac{1}{n} \sum_{i=1}^{n} g_i^{(k)}$$

- SAG gradient estimates are no longer unbiased, but they have greatly reduced variance
- Isn't it expensive to average all these gradients? Basically just as efficient as SGD, as long we're clever:

$$x^{(k)} = x^{(k-1)} - \alpha_k \underbrace{\left(\frac{1}{n} g_i^{(k)} - \frac{1}{n} g_i^{(k-1)} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} g_i^{(k-1)}}_{\text{old table average}} \right)}_{\text{new table average}}$$

[2] Schmidt, M., Le Roux, N., Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. Mathematical Programming, 162, 83-112.

SAG convergence

Assume that $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, where each f_i is differentiable, and ∇f_i is Lipschitz with constant L.

Denote $\bar{x}^{(k)} = \frac{1}{k}\sum_{l=0}^{k-1} x^{(l)}$, the average iterate after $k-1$ steps.

Theorem

SAG, with a fixed step size $\alpha = \frac{1}{16L}$, and the initialization

$$g_i^{(0)} = \nabla f_i(x^{(0)}) - \nabla f(x^{(0)}), \quad i = 1, \ldots, n$$

satisfies

$$\mathbb{E}[f(\bar{x}^{(k)})] - f^\star \leq \frac{48n}{k}[f(x^{(0)}) - f^\star] + \frac{128L}{k}\|x^{(0)} - x^\star\|^2$$

where the expectation is taken over random choices of indices.

SAG convergence

- Result stated in terms of the average iterate $\bar{x}^{(k)}$, but also can be shown to hold for the best iterate $x^{(k)}_{best}$ seen so far.
- This is $\mathcal{O}\left(\frac{1}{k}\right)$ convergence rate for SAG. Compare to $\mathcal{O}\left(\frac{1}{k}\right)$ rate for GD, and $\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ rate for SGD.
- But, the constants are different! Bounds after k steps:
    - GD: $\frac{L\|x^{(0)}-x^\star\|^2}{2k}$
    - SAG: $\frac{48n[f(x^{(0)})-f^\star]+128L\|x^{(0)}-x^\star\|^2}{k}$
- So the first term in SAG bound suffers from a factor of n; authors suggest smarter initialization to make $f(x^{(0)}) - f^\star$ small (e.g., they suggest using the result of n SGD steps).

SAG convergence

Assume further that each f_i is strongly convex with parameter μ.

Theorem

SAG, with a step size $\alpha = \frac{1}{16L}$ and the same initialization as before, satisfies

$$\mathbb{E}[f(x^{(k)})] - f^\star \leq \left(1 - \min\left(\frac{\mu}{16L}, \frac{1}{8n}\right)\right)^k \left(\frac{3}{2}\left(f(x^{(0)}) - f^\star\right) + \frac{4L}{n}\|x^{(0)} - x^\star\|^2\right)$$

Notes:

- This is linear convergence rate $\mathcal{O}(\gamma^k)$ for SAG. Compare this to $\mathcal{O}(\gamma^k)$ for GD, and only $\mathcal{O}\left(\frac{1}{k}\right)$ for SGD.
- Like GD, we say SAG is adaptive to strong convexity.
- Proofs of these results not easy: 15 pages, computed-aided!

The name of this anime?

~~Vinland~~ SAGA [3]

SAG:

$$x^{k+1} = x^k - \gamma \left[\frac{f_j'(x^k) - f_j'(\phi_j^k)}{n} + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right]$$

SAGA:

$$x^{k+1} = x^k - \gamma \left[f_j'(x^k) - f_j'(\phi_j^k) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right]$$

SVRG:

$$x^{k+1} = x^k - \gamma \left[f_j'(x^k) - f_j'(\tilde{x}) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\tilde{x}) \right]$$

- both SAGA and SVRG are unbiased, compared to SAG, which makes its analysis and convergence guarantees better
- has n^2 times bigger variance
- can be extened to composite optimization by use of proximal operators

[3] Defazio A. et. al. (2014). SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives.

SAGA convergence

Assume that $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$, where each f_i is convex, differentiable, and ∇f_i is Lipschitz with constant L.

Denote $\bar{x}^{(k)} = \frac{1}{k}\sum_{l=0}^{k-1} x^{(l)}$, the average iterate after $k-1$ steps.

Theorem

SAGA, with a fixed step size $\alpha = \frac{1}{3L}$, and the initialization

$$g_i^{(0)} = \nabla f_i(x^{(0)}) - \nabla f(x^{(0)}), \quad i = 1, \dots, n$$

satisfies

$$\mathbb{E}[f(\bar{x}^{(k)})] - f^\star \leq \frac{4n}{k}\left[f(x^0) - f^*\right] + \frac{8L}{k}\left[\|x^0 - x^*\|^2\right].$$

where the expectation is taken over random choices of indices.

- Due to unbiasedness of the update, the analysis is much less complex than for SAG

VR methods for NNs [4]

- training of neural networks breaks most of assumptions needed for VR method to work
- authors consider SVRG, since it does not need to have full gradient in memory
- results in poor results or even divergence
- in the end authors propose to use VR technique along with adaptivity of AdaGrad, ADAM, etc.

[4] Defazio, A., Bottou, L. (2019). On the ineffectiveness of variance reduced optimization for deep learning. Advances in Neural Information Processing Systems, 32.

Computational experiments

Let's look at computational experiments for

- SGD, SAG and SVRG in JAX.
- SVRG in JAX for VAE.
- SVRG & SARAH Pure Torch (MLP + RNN)