Variance Reduction Methods (Part I)

SIE 449/549: Optimization for Machine Learning

Afrooz Jalilzadeh

Department of Systems and Industrial Engineering
University of Arizona

THE UNIVERSITY OF ARIZONA

Deterministic vs Stochastic

- Consider the following convex optimization problem:

$$\min \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

- **Deterministic Method:** Uses all n gradients:

$$x_{k+1} = x_k - \alpha \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) \right)$$

- **Stochastic Method:** Approximates gradient with 1 sample:

$$x_{k+1} = x_k - \alpha \nabla f_{i_k}(x)$$

- **Mini-batch Approach**: Uses average of b sample gradinet:

$$x_{k+1} = x_k - \alpha \left(\frac{1}{b}\sum_{i \in I_k} \nabla f_i(x) \right), \quad I_k \subset \{1,\dots,n\},\ |I_k| = b$$

- **Convergence Rate:** GD: $\mathcal{O}(1/k)$, SGD: $\mathcal{O}(1/\sqrt{k})$

Increasing Batch Size Method

- Increase the batch size at each iteration:

$$x_{k+1} = x_k - \alpha \left(\frac{1}{|\beta_k|} \sum_{i \in \beta_k} \nabla f_i(x) \right), \quad \{\beta_k\} \text{ is an increasing sequance}$$

- Increasing the batch size is a form of variance-reduction
- At some point switch from stochastic to deterministic, i.e., $|\beta_k| = n$
- If we use constant stepsize $\alpha = 1/L$, we get:

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|^2 + \frac{1}{2L} \|w_k\|^2,$$

where w_k is the error of estimating the gradient

Effect of Batch Size on Error

- If we sample with replacement we get:

$$\mathbb{E}[\|w_k\|^2] =$$

where σ^2 is the variance of the gradient norms

- If we sample without replacement we get:

$$\mathbb{E}[\|w_k\|^2] =$$

which drives error to zero as batch size approaches n

- Disadvantages of increasing batch size:
  - Variance should be bounded
  - Per iteration complexity increases

Stochastic Average Gradient (SAG)*

- Growing $|\beta_k|$ eventually requires $\mathcal{O}(n)$ iteration cost
- Can we have 1 gradient per iteration and get convergence rate of $\mathcal{O}(1/k)$?
- To motivate SAG, let's view gradient descent as performing the iteration

$$x_{k+1} = x_k - \frac{\alpha_k}{n}\sum_{i=1}^{n} g_k^{(i)},$$

where on each step we set $g_k^{(i)} = \nabla f_i(x_k)$

- **SAG method**:
  - Only set $g_k^{(i_k)} = \nabla f_{i_k}(x_k)$ for a randomly-chosen i_k
  - All other $g_k^{(i)}$ are kept at their previous value

*Schmidt, Mark, Nicolas Le Roux, and Francis Bach. "Minimizing finite sums with the stochastic average gradient." Mathematical Programming 162.1-2 (2017): 83-112.

Stochastic Average Gradient (SAG)

- Maintain table, containing $g^{(i)}(x) = \nabla f_i(x)$, $i = 1, \ldots, n$

- Initialize x_0 and set $g = 0$ and $g^{(i)} = 0$ for all $i = 1, \ldots, n$
- At step $k = 1, 2, 3, \ldots$ pick random $i_k \in \{1, \ldots, n\}$ and then let
$$g_k^{(i_k)} = \nabla f_i(x_{k-1})$$
set all other $g_k^{(i_k)} = g_{k-1}^{(i_k)}$, $i \neq i_k$, i.e., they stay the same
- Update
$$x_k = x_{k-1} - \alpha_k \left(\frac{1}{n} \sum_{i=1}^{n} g_k^{(i)} \right)$$

- Isn't it expensive to average all these gradients? (Especially when n is large?) Basically just as efficient as SGD, as long we're clever:

SAG Variance Reduction

- SAG gradient estimates, $\theta = \frac{1}{n}\left(g_k^{(i_k)} - g_{k-1}^{(i_k)}\right) + \frac{1}{n}\sum_{i=1}^{n} g_{k-1}^{(i)}$,
  - are no longer unbiased
  - have greatly reduced variance
- Let $X = g_k^{(i_k)}$ and $Y = g_{k-1}^{(i_k)}$, then $\theta = \frac{1}{n}(X - Y) + \mathbb{E}[Y]$:

$$\mathbb{E}[\theta] =$$

SAG Convergence Analysis

Theorem 1

Assume $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ is convex with Lipchitz gradinet. SAG with constant step size $\alpha = 1/(16L)$ and the initialization

$$g_0^{(i)} = \nabla f_i(x_0) - \nabla f(x_0), \quad i = 1, \dots, n$$

satisfies

$$\mathbb{E}[f(\bar{x}_T)] - f(x^*) \leq \frac{48n}{T}(f(x_0) - f(x^*)) + \frac{128L}{T}\|x_0 - x^*\|^2,$$

where $\bar{x}_T = \frac{1}{T}\sum_{k=0}^{T-1} x_k$.

SAGA

- Maintain table, containing $g^{(i)}(x) = \nabla f_i(x)$, $i = 1, \dots, n$
- Initialize x_0 and set $g = 0$ and $g^{(i)} = 0$ for all $i = 1, \dots, n$
- At step $k = 1, 2, 3, \dots$ pick random $i_k \in \{1, \dots, n\}$ and then let
$$g_k^{(i_k)} = \nabla f_i(x_{k-1})$$
set all other $g_k^{(i_k)} = g_{k-1}^{(i_k)}$, $i \neq i_k$, i.e., they stay the same
- Update
$$x_k = x_{k-1} - \alpha_k \left(g_k^{(i_k)} - g_{k-1}^{(i_k)} + \frac{1}{n} \sum_{i=1}^{n} g_{k-1}^{(i)} \right)$$

Notice that the only difference between SAG and SAGA is the heavier weight on the updated gradient at step k. We have:

$$g_k^{(i_k)} - g_{k-1}^{(i_k)} + \frac{1}{n} \sum_{i=1}^{n} g_{k-1}^{(i)}$$

instead of

$$\frac{g_k^{(i_k)}}{n} - \frac{g_{k-1}^{(i_k)}}{n} + \frac{1}{n} \sum_{i=1}^{n} g_{k-1}^{(i)}$$

SAGA vs SAG

- Interestingly, the SAGA gradient is unbiased
- SAGA gradient estimates, $\theta = g_k^{(i_k)} - g_{k-1}^{(i_k)} + \frac{1}{n}\sum_{i=1}^{n} g_{k-1}^{(i)}$
- Let $X = g_k^{(i_k)}$ and $Y = g_{k-1}^{(i_k)}$, then $\theta = X - Y + \mathbb{E}[Y]$:

$$\mathbb{E}[\theta] =$$

SAGA vs SAG

- SAGA has a higher variance than SAG but is unbiased
- SAG gradient estimates, $\frac{1}{n}\left(g_k^{(i_k)} - g_{k-1}^{(i_k)}\right) + \frac{1}{n}\sum_{i=1}^{n} g_{k-1}^{(i)}$
- SAGA gradient estimates, $g_k^{(i_k)} - g_{k-1}^{(i_k)} + \frac{1}{n}\sum_{i=1}^{n} g_{k-1}^{(i)}$
- Let $X = g_k^{(i_k)}$ and $Y = g_{k-1}^{(i_k)}$, then we have

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

where X, Y are corrolated and $\alpha = 1$ for SAGA and $\alpha = \frac{1}{n}$ for SAG