Optimization for Machine Learning
机器学习中的优化方法

陈 程

华东师范大学 软件工程学院

chchen@sei.ecnu.edu.cn

Outline

Empirical Risk Minimization

Let $\{\mathbf{a}_i, b_i\}_{i=1}^n$ be n random samples. In machine learning, we usually learn model parameters $\mathbf{x}$ by optimizing

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}; \{\mathbf{a}_i, b_i\}).$$

- hinge loss (support vector machine):

$$f(\mathbf{x}; \{\mathbf{a}_i, b_i\}) = \max\{1 - b_i \mathbf{a}_i^\top \mathbf{x}, 0\}$$

- logistic loss (logistic regression):

$$f(\mathbf{x}; \{\mathbf{a}_i, b_i\}) = \log(1 + \exp(-b_i \mathbf{a}_i^\top \mathbf{x}))$$

- neural network

Stochastic Optimization

More generally, we consider the stochastic optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq \underbrace{\mathbb{E}_\xi[f(\mathbf{x};\xi)]}_{\text{expectation setting}},$$

where the random variable $\xi \sim \mathcal{D}$.

- ξ is the randomness in problem.
- In this lecture, we suppose $f(\cdot,\xi)$ is convex for all ξ, and thus $F(\mathbf{x})$ is convex.

Finite-sum Setting

The finite-sum setting is a special case of the expectation setting:

$$F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}).$$

If one draws index i from $\{1, 2, \ldots, n\}$ uniformly at random, then

$$F(\mathbf{x}) = \mathbb{E}_i[f_i(\mathbf{x})].$$

A Natural Idea

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{x}_t - \eta_t \nabla F(\mathbf{x}_t) \\ &= \mathbf{x}_t - \eta_t \nabla \mathbb{E}[f(\mathbf{x}_t, \xi)] \\ &= \mathbf{x}_t - \eta_t \mathbb{E}[\nabla_{\mathbf{x}} f(\mathbf{x}_t, \xi)]\end{aligned}$$

issues:

- For the expectation setting, distribution of ξ may be unknown.
- For the finite-sum setting, computing full gradient is very expensive when n is very large.

Outline

Stochastic Gradient Descent (SGD)

Stochastic gradient descent:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t g(\mathbf{x}_t, \xi),$$

where $g(\mathbf{x}_t, \xi)$ is an unbiased estimator of $\nabla F(\mathbf{x}_t)$, i.e.,

$$\mathbb{E}[g(\mathbf{x}_t, \xi)] = \nabla F(\mathbf{x}_t).$$

For the finite-sum setting, we can choose index i_t from $\{1, 2, \ldots, n\}$ uniformly at random. Then $\nabla f_{i_t}(\mathbf{x}_t)$ is an unbiased estimator of $\nabla F(\mathbf{x}_t)$.

Outline

Strongly convex and smooth problems

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \triangleq \mathbb{E}_\xi[f(\mathbf{x};\xi)]$$

Assumptions:

- F is L-smooth and μ-strongly convex;
- Given $\xi_0, \ldots, \xi_{t-1}$, $g(\mathbf{x}_t, \xi_t)$ is an unbiased estimator of $\nabla F(\mathbf{x}_t)$;
- For all $\mathbf{x}$, we have $\underbrace{\mathbb{E}[\|g(\mathbf{x},\xi)\|_2^2] \leq \sigma^2}_{\text{bounded variance}}$.

Convergence with fixed stepsizes

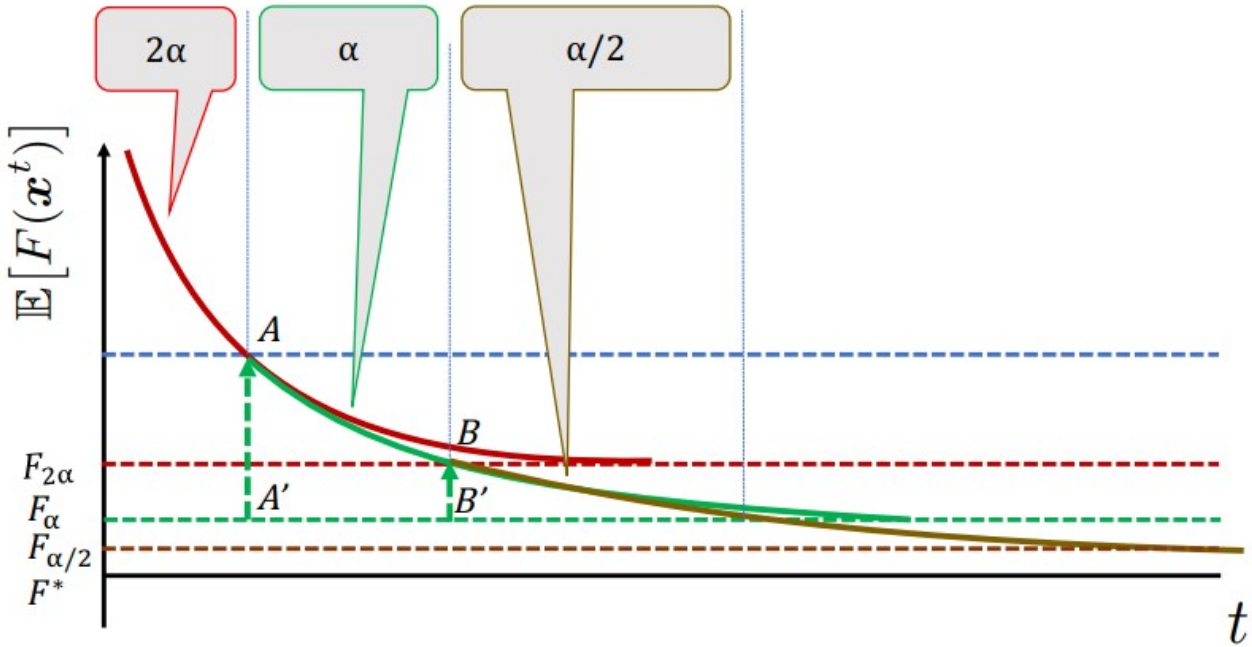
Under the assumptions in page 7, if $\eta_t = \eta \leq 1/(2L)$, then SGD achieves

$$\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2] \leq (1 - 2\eta\mu)^t \|\mathbf{x}_0 - \mathbf{x}^*\|_2^2 + \frac{\eta\sigma^2}{2\mu}$$

- fast (linear) convergence at the very beginning
- converges to some neighborhood of $\mathbf{x}^*$
- smaller stepsize η yield better converging points

One Practical Strategy

Run SGD with fixed stepsizes; whenever progress stalls, half the stepsize and continue SGD.



Convergence with diminishing stepsizes

Under the assumptions in page 7, if $\eta_t = \eta \leq \frac{\theta}{t+1}$ for some $\theta > \frac{1}{2\mu}$, then SGD achieves

$$\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2] \leq \frac{\alpha_\theta}{t+1}$$

where $\alpha_\theta = \max\{\|\mathbf{x}_0 - \mathbf{x}\|_2^2, \frac{2\theta^2\sigma^2}{2\mu\theta - 1}\}$

Convex and Smooth Problems

Suppose we return a weighted average

$$\tilde{\mathbf{x}} = \sum_{k=0}^{t} \frac{\eta_k}{\sum_{j=0}^{t} \eta_j} \mathbf{x}_k$$

If F is convex, we have

$$\mathbb{E}[F(\tilde{\mathbf{x}}_t) - F(\mathbf{x}^*)] \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2 + \sum_{k=0}^{t} \sigma^2 \eta_k^2}{2 \sum_{k=0}^{t} \eta_k}.$$

If we choose $\eta_t = \Theta(1/\sqrt{t})$, we can get

$$\mathbb{E}[F(\tilde{\mathbf{x}}_t) - F(\mathbf{x}^*)] \leq O(\frac{\log t}{\sqrt{t}}).$$

Outline

Stochastic Variance Reduced Gradient (SVRG)

If we have access to a history point $\tilde{\mathbf{x}}$ and $\nabla F(\tilde{\mathbf{x}})$, how to build a unbiased gradient estimator with converges to $\mathbf{0}$?

$$\underbrace{\nabla f_i(\mathbf{x}_t) - \nabla f_i(\tilde{\mathbf{x}})}_{\to \mathbf{0} \text{ if } \mathbf{x}_t \approx \tilde{\mathbf{x}}} + \underbrace{\nabla F(\tilde{\mathbf{x}})}_{\to \mathbf{0} \text{ if } \tilde{\mathbf{x}} \approx \mathbf{x}^*}$$

where i is randomly sampled from $\{1, \ldots, n\}$.

- an unbiased estimator of $F(\tilde{\mathbf{x}})$
- converges to $\mathbf{0}$ if $\mathbf{x}_t \approx \tilde{\mathbf{x}} \approx \mathbf{x}^*$

Stochastic Variance Reduced Gradient (SVRG)

- operate in epochs
- in the s-th epoch
  - **beginning:** take a snapshot $\tilde{\mathbf{x}}$ of the current iterate, and compute the batch gradient

$$\nabla f(\tilde{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\mathbf{x}}).$$

  - **inner loop:** use the snapshot point to help reduce variance

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t (\nabla f_i(\mathbf{x}_t) - \nabla f_i(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}})),$$

Stochastic Variance Reduced Gradient (SVRG)

Algorithm 1 Stochastic Variance Reduced Gradient

Input: $\mathbf{x}_0$, η, m, S
$\tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$
for $s = 0, \ldots, S-1$
  $\mathbf{x}_0 = \tilde{\mathbf{x}}^{(s)}$
  for $t = 0, \ldots, m-1$
    draw i_t from $\{1, \ldots, n\}$ uniformly at random
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\tilde{\mathbf{x}}^{(s)}) + \nabla f(\tilde{\mathbf{x}}^{(s)}))$,
  end for
  Option I: $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_m$
  Option II: $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_t$ for randomly chosen $t \in \{0, \ldots, m-1\}$
end for
Output: $\tilde{\mathbf{x}}^{(S)}$

Remark

- constant stepsize η
- each epoch contains $2m + n$ gradient computations

Stochastic Variance Reduced Gradient (SVRG)

Assume $\eta = \Theta(1/L)$ and $m = \Theta(\kappa)$ is sufficient large so that

$$\rho = \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1,$$

then SVRG holds that

$$\mathbb{E}\big[f(\tilde{\mathbf{x}}^{(s)}) - f(\mathbf{x}^*)\big] \leq \rho^s (f(\tilde{\mathbf{x}}_0) - f(\mathbf{x}^*)).$$

To achieve

$$\mathbb{E}\big[f(\tilde{\mathbf{x}}^{(s)}) - f(\mathbf{x}^*)\big] \leq \epsilon$$

we only require at most $\mathcal{O}((\kappa + n)\log(1/\epsilon))$ number of gradient computations.

Summary

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}).$$

	iteration complexity	per-iteration	total
batch GD	$\log\frac{1}{\epsilon}$	n	$n\kappa\log\frac{1}{\epsilon}$
SGD	$\frac{1}{\epsilon}$	1	$\frac{1}{\epsilon}$
SVRG	$\log\frac{1}{\epsilon}$	$n+\kappa$	$(n+\kappa)\log\frac{1}{\epsilon}$

Table: Convergence rate for the strongly convex case

Questions