ECE826 Lecture 7:

A Primer on SGD

Contents

- Complexity of GD
- Intro to SGD, and convergence guarantees
- Comparisons between SGD to GD
- Towards rates for nonconvex functions

Minimizing the Empirical Risk

- The empirical cost function that we have access to

$$\min_{h \in \mathscr{H}} \left(R_S[h] = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i); y_i) \right)$$

- Question: Can we approximate the solution to this minimization? If so how fast?

- The answer must depend on:

  1) n, the sample size
  2) $\mathscr{H}$, the hypothesis class and loss function
  3) $\mathscr{D}$, the data distribution
  4) the optimization algorithm that outputs our classifier

Last time: GD's Convergence Rates

Function Class	Convergence Rate
Lipschitz	$\frac{RL}{\sqrt{T}}$
smooth	$\frac{R^2\beta}{T}$
Lipschitz + str. cvx	$\frac{L^2}{\lambda T}$
smooth + str. cvx	$R^2 e^{-\frac{T}{\kappa}}$

- The structure of a function can help in improving computational complexity. However, we should be cautious that the bounds of complexity are not always tight.

How expensive is GD in practice?

Computational complexity of GD

Gradient Descent Method:

Run the following for T_ϵ steps

$$w_{k+1} = w_k - \gamma \nabla f(w_k)$$

Computational complexity of GD

Gradient Descent Method:

Run the following for T_ϵ steps

$$w_{k+1} = w_k - \gamma \nabla f(w_k)$$

- unit if cost = number of $\nabla f(w)$ computations
- total cost = $O(T_\epsilon \cdot \mathrm{cost}(\nabla f))$

Computational complexity of GD

Gradient Descent Method:

Run the following for T_ϵ steps

$$w_{k+1} = w_k - \gamma \nabla f(w_k)$$

- unit if cost = number of $\nabla f(w)$ computations
- total cost = $O(T_\epsilon \cdot \text{cost}(\nabla f))$
- Let's see an example: logistic regression

Computational complexity of GD

Example:

A $f(w)$ is the logistic loss across that is both $\{x_1, \ldots, x_n\}$ plus a regularizer

$$f(w) = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i \langle w, x_i \rangle}\right) + \frac{\lambda}{2} \|w\|^2$$

Computational complexity of GD

Example:

A $f(w)$ is the logistic loss across that is both $\{x_1, \ldots, x_n\}$ plus a regularizer

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$$

A few facts:

- $\log(1+e^x)$ is 1-Lipschitz and $1/4$-smooth
- $\langle x, w\rangle$ is $\|x\|$-Lipschitz and $\|x\|^2$-smooth

Computational complexity of GD

Example:

A $f(w)$ is the logistic loss across that is both $\{x_1, \ldots, x_n\}$ plus a regularizer

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$$

A few facts:

- $\log(1+e^x)$ is 1-Lipschitz and $1/4$-smooth
- $\langle x, w\rangle$ is $\|x\|$-Lipschitz and $\|x\|^2$-smooth
- $g_1(g_2(x))$ is an $L_1 \cdot L_2$-Lipschitz function
- $g_1(x) + g_2(x)$ is an $(L_1 + L_2)$-Lipschitz function and a $(\beta_1 + \beta_2)$-smooth function

Computational complexity of GD

Example:

A $f(w)$ is the logistic loss across that is both $\{x_1, \ldots, x_n\}$ plus a regularizer

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$$

A few facts:

- $\log(1 + e^x)$ is 1-Lipschitz and $1/4$-smooth
- $\langle x, w\rangle$ is $\|x\|$-Lipschitz and $\|x\|^2$-smooth
- $g_1(g_2(x))$ is an $L_1 \cdot L_2$-Lipschitz function
- $g_1(x) + g_2(x)$ is an $(L_1 + L_2)$-Lipschitz function and a $(\beta_1 + \beta_2)$-smooth function
- $g(\langle x, w\rangle + b)$ is a $(\beta \cdot \|x\|^2)$-smooth function

Computational complexity of GD

Example:

A $f(w)$ is the logistic loss across that is both $\{x_1, \ldots, x_n\}$ plus a regularizer

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$$

A few facts:

- $\log(1 + e^x)$ is 1-Lipschitz and $1/4$-smooth
- $\langle x, w\rangle$ is $\|x\|$-Lipschitz and $\|x\|^2$-smooth
- $g_1(g_2(x))$ is an $L_1 \cdot L_2$-Lipschitz function
- $g_1(x) + g_2(x)$ is an $(L_1 + L_2)$-Lipschitz function and a $(\beta_1 + \beta_2)$-smooth function
- $g(\langle x, w\rangle + b)$ is a $(\beta \cdot \|x\|^2)$-smooth function

What properties does the regularized log. loss ERM have?

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_{i} \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_{i} \|x_i\| + \lambda \cdot \max_{w \in \mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_{i} \|x_i\|^2 + \lambda\right)$-smooth and

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Iterations for GD to reach error ϵ

$$T_\epsilon = O\left(\frac{\beta}{\lambda}\log(\|w_0 - w^*\|/\epsilon)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathcal{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Iterations for GD to reach error ϵ

$$T_\epsilon = O\left(\frac{\beta}{\lambda}\log(\|w_0 - w^*\|/\epsilon)\right)$$

$$= O\left(d \log\left(\frac{d}{\epsilon}\right)\right)$$

Gradient Cost?

Proposition:

For loss functions function written as $f(w) = \sum_{i=1}^{n} \ell(\langle w, x_i \rangle)$ computing $\nabla f(w)$ takes time $O(\text{nnz}(X)) = O(nd)$

Gradient Cost?

Proposition:

For loss functions function written as $f(w) = \sum_{i=1}^{n} \ell(\langle w, x_i \rangle)$ computing $\nabla f(w)$ takes time $O(\text{nnz}(X)) = O(nd)$

- Proof sketch: the gradient with respect to the model for each loss is equal to $\nabla_w \ell(\langle w, x_i \rangle) = \ell'(\langle w, x_i \rangle) \cdot x_i$.

Gradient Cost?

Proposition:

For loss functions function written as $f(w) = \sum_{i=1}^{n} \ell(\langle w, x_i \rangle)$ computing $\nabla f(w)$ takes time $O(\text{nnz}(X)) = O(nd)$

- Proof sketch: the gradient with respect to the model for each loss is equal to $\nabla_w \ell(\langle w, x_i \rangle) = \ell'(\langle w, x_i \rangle) \cdot x_i$.

- The cost of $\ell'(\langle w, x_i \rangle)$ is proportional to the cost of $\langle w, x_i \rangle$

Gradient Cost?

Proposition:

For loss functions function written as $f(w) = \sum_{i=1}^{n} \ell(\langle w, x_i \rangle)$ computing $\nabla f(w)$ takes time $O(\text{nnz}(X)) = O(nd)$

- Proof sketch: the gradient with respect to the model for each loss is equal to $\nabla_w \ell(\langle w, x_i \rangle) = \ell'(\langle w, x_i \rangle) \cdot x_i$.
- The cost of $\ell'(\langle w, x_i \rangle)$ is proportional to the cost of $\langle w, x_i \rangle$

One "full-batch" gradient requires a full pass over the data, and costs linear in the size of the data set

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w \in \mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Iterations for GD to reach error ϵ

$$T_\epsilon = O\left(\frac{\beta}{\lambda}\log(\|w_0 - w^*\|/\epsilon)\right)$$

$$= O\left(d \log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Iterations for GD to reach error ϵ

$$T_\epsilon = O\left(\frac{\beta}{\lambda}\log(\|w_0 - w^*\|/\epsilon)\right)$$

$$= O\left(d\log\left(\frac{d}{\epsilon}\right)\right)$$

Total computational cost

$$O(T_\epsilon \cdot \text{cost}(\nabla f)) = O\left(\text{nnz}(X)\cdot d\log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Iterations for GD to reach error ϵ

$$T_\epsilon = O\left(\frac{\beta}{\lambda}\log(\|w_0 - w^*\|/\epsilon)\right)$$

$$= O\left(d\log\left(\frac{d}{\epsilon}\right)\right)$$

Total computational cost

$$O(T_\epsilon \cdot \text{cost}(\nabla f)) = O\left(\text{nnz}(X)\cdot d\log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2\log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

In this case GD has a cost that is linear in the number of data points, but quadratic with regards to input dimension = too large!

Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

- Note: we haven't used the fact that $f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$

Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

- Note: we haven't used the fact that $f(w) = \dfrac{1}{n}\sum_{i=1}^{n} f_i(w)$
- Idea ('50s, '60s [Robbins, Monro], [Widrow, Hoff]):
  instead of computing $\nabla f(w)$ we can sample one f_i at random and compute its gradient

Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

- Note: we haven't used the fact that $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$
- Idea ('50s, '60s [Robbins, Monro], [Widrow, Hoff]):
  instead of computing $\nabla f(w)$ we can sample one f_i at random and compute its gradient
- Why does that make sense? In "expectation" it's the same algorithm, i.e.,

$$E_{i\sim\text{uniform}} \nabla f_i = \sum_i \frac{1}{n} \nabla f_i = \nabla f(w)$$

Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

- Note: we haven't used the fact that $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$
- Idea ('50s, '60s [Robbins, Monro], [Widrow, Hoff]):
  instead of computing $\nabla f(w)$ we can sample one f_i at random and compute its gradient
- Why does that make sense? In "expectation" it's the same algorithm, i.e.,

$$E_{i \sim \text{uniform}} \nabla f_i = \sum_i \frac{1}{n} \nabla f_i = \nabla f(w)$$

SGD: $w_{k+1} = w_k - \gamma \nabla f_{i_k}(w_k)$

Can we make GD faster?

Gradient Descent Method: $w_{k+1} = w_k - \gamma \nabla f(w_k)$

- Note: we haven't used the fact that $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$
- Idea ('50s, '60s [Robbins, Monro], [Widrow, Hoff]):
  instead of computing $\nabla f(w)$ we can sample one f_i at random and compute its gradient

- Why does that make sense? In "expectation" it's the same algorithm, i.e.,

$$E_{i\sim\text{uniform}} \nabla f_i = \sum_i \frac{1}{n} \nabla f_i = \nabla f(w)$$

SGD: $w_{k+1} = w_k - \gamma \nabla f_{i_k}(w_k)$

The Uber-Algorithm

Different names and flavors

ML / Optimization / Statistics / EE

Perceptron

Incremental Gradient

Back Propagation (NNs)

Oja's iteration (PCA)

LMS Filter

. . .

Has been around for a while, for good reasons:

Robust to noise

Simple to implement

Near-optimal learning performance *

Small computational foot-print

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1 - \gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$$

$$\begin{aligned}
\mathbb{E}\|w_{k+1} - w^*\|^2 &= \mathbb{E}\|w_k - \gamma \nabla f_{s_k} - w^*\|^2 \\
&= \mathbb{E}\|w_k - x^*\|^2 - 2\gamma \mathbb{E} \left\langle \nabla f_{s_k}(w_k), w_k - x^* \right\rangle + \gamma^2 \mathbb{E}\|\nabla f_{s_k}(w_k)\|^2
\end{aligned}$$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1 - \gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$$

$$\begin{aligned}
\mathbb{E}\|w_{k+1} - w^*\|^2 &= \mathbb{E}\|w_k - \gamma \nabla f_{s_k} - w^*\|^2 \\
&= \mathbb{E}\|w_k - x^*\|^2 - 2\gamma \mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - x^* \right\rangle + \gamma^2 \mathbb{E}\|\nabla f_{s_k}(w_k)\|^2 \\
&\leq \mathbb{E}\|w_k - w^*\|^2 - 2\gamma \mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - w^* \right\rangle + \gamma^2 L^2 \\
&\leq (1 - \gamma \cdot m)\mathbb{E}\|w_k - x^*\|^2 + \gamma^2 M^2
\end{aligned}$$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$$

$$\begin{aligned}
\mathbb{E}\|w_{k+1} - w^*\|^2 &= \mathbb{E}\|w_k - \gamma \nabla f_{s_k} - w^*\|^2 \\
&= \mathbb{E}\|w_k - x^*\|^2 - 2\gamma \mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - x^* \right\rangle + \gamma^2 \mathbb{E}\|\nabla f_{s_k}(w_k)\|^2 \\
&\leq \mathbb{E}\|w_k - w^*\|^2 - 2\gamma \mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - w^* \right\rangle + \gamma^2 L^2 \\
&\leq (1 - \gamma \cdot m)\mathbb{E}\|w_k - x^*\|^2 + \gamma^2 M^2 \\
&\vdots \\
&\leq (1 - \gamma \cdot m)^k \mathbb{E}\|w_0 - x^*\|^2 + \sum_{i=1}^{k} (1-\gamma\lambda)^i \gamma^2 M^2
\end{aligned}$$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k\|w_0 - w^*\|^2 + \gamma\frac{L^2}{\lambda}$$

$$\begin{aligned}
\mathbb{E}\|w_{k+1} - w^*\|^2 &= \mathbb{E}\|w_k - \gamma\nabla f_{s_k} - w^*\|^2 \\
&= \mathbb{E}\|w_k - x^*\|^2 - 2\gamma\mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - x^* \right\rangle + \gamma^2\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2 \\
&\leq \mathbb{E}\|w_k - w^*\|^2 - 2\gamma\mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - w^* \right\rangle + \gamma^2 L^2 \\
&\leq (1-\gamma\cdot m)\mathbb{E}\|w_k - x^*\|^2 + \gamma^2 M^2 \\
&\vdots \\
&\leq (1-\gamma\cdot m)^k\mathbb{E}\|w_0 - x^*\|^2 + \sum_{i=1}^{k}(1-\gamma\lambda)^i\gamma^2 M^2 \\
&\leq (1-\gamma\cdot m)^k\mathbb{E}\|w_0 - x^*\|^2 + \frac{1}{\gamma\lambda}\gamma^2 M^2
\end{aligned}$$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k\|w_0 - w^*\|^2 + \gamma\frac{L^2}{\lambda}$$

$$\begin{aligned}
\mathbb{E}\|w_{k+1} - w^*\|^2 &= \mathbb{E}\|w_k - \gamma\nabla f_{s_k} - w^*\|^2 \\
&= \mathbb{E}\|w_k - x^*\|^2 - 2\gamma\mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - x^* \right\rangle + \gamma^2\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2 \\
&\leq \mathbb{E}\|w_k - w^*\|^2 - 2\gamma\mathbb{E}\left\langle \nabla f_{s_k}(w_k), w_k - w^* \right\rangle + \gamma^2 L^2 \\
&\leq (1-\gamma\cdot m)\mathbb{E}\|w_k - x^*\|^2 + \gamma^2 M^2 \\
&\vdots \\
&\leq (1-\gamma\cdot m)^k\mathbb{E}\|w_0 - x^*\|^2 + \sum_{i=1}^{k}(1-\gamma\lambda)^i\gamma^2 M^2 \\
&\leq (1-\gamma\cdot m)^k\mathbb{E}\|w_0 - x^*\|^2 + \frac{1}{\gamma\lambda}\gamma^2 M^2
\end{aligned}$$

Let's interpret these rates

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k\|w_0 - w^*\|^2 + \gamma\frac{L^2}{\lambda}$$

- Let us set the stepwise to $\gamma = 0.1/\lambda$, then

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq 0.9^k R^2 + 0.1\frac{L^2}{\lambda^2}$$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k\|w_0 - w^*\|^2 + \gamma\frac{L^2}{\lambda}$$

- Let us set the stepwise to $\gamma = 0.1/\lambda$, then

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq 0.9^k R^2 + 0.1\frac{L^2}{\lambda^2}$$

- For any $\epsilon \geq 2 \cdot 0.1\frac{L^2}{\lambda^2}$, we need $k \approx 42 \cdot \log\frac{R^2}{\epsilon}$ iterations.

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k\|w_0 - w^*\|^2 + \gamma\frac{L^2}{\lambda}$$

- Let us set the stepwise to $\gamma = 0.1/\lambda$, then
$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq 0.9^k R^2 + 0.1\frac{L^2}{\lambda^2}$$
- For any $\epsilon \geq 2 \cdot 0.1\frac{L^2}{\lambda^2}$, we need $k \approx 42 \cdot \log\frac{R^2}{\epsilon}$ iterations.

SGD converges exponentially fast to certain "error floor"

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1 - \gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$$

- We can go beyond the error floor: $\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1 - \gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i\|\nabla f_i(w)\| \leq L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \leq (1-\gamma\lambda)^k\|w_0 - w^*\|^2 + \gamma\frac{L^2}{\lambda}$$

- We can go beyond the error floor: $\mathbb{E}\|w_{k+1} - w^*\|^2 \leq \underbrace{(1-\gamma\lambda)^k\|w_0 - w^*\|^2}_{\varepsilon/2} + \underbrace{\gamma\frac{L^2}{\lambda}}_{\varepsilon/2}$
-

Convergence rates for SGD

Theorem

Let $f(w)$ be a λ-strongly convex and function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \le L$). Then, the iterates of SGD with step-size γ satisfy

$$\mathbb{E}\|w_{k+1} - w^*\|^2 \le (1-\gamma\lambda)^k \|w_0 - w^*\|^2 + \gamma \frac{L^2}{\lambda}$$

- We can go beyond the error floor: $\mathbb{E}\|w_{k+1} - w^*\|^2 \le \underbrace{(1-\gamma\lambda)^k \|w_0 - w^*\|^2}_{\varepsilon/2} + \underbrace{\gamma \frac{L^2}{\lambda}}_{\varepsilon/2}$

- Observe that for $\gamma = \epsilon \frac{\lambda}{2L^2}$ we get to any arbitrary error within $k = 2\left(\frac{L}{\lambda}\right)^2 \cdot \frac{1}{\epsilon} \cdot \log\left(\frac{2R}{\epsilon}\right)$ iterations

Convergence rates for SGD

Corollary:

SGD with constant stepwise achieves exponential convergence till error an error floor of

$\mathbb{E}\|w_{k+1} - w^*\|^2 \geq \epsilon \cdot O\left(\frac{L^2}{\lambda^2}\right)$ and after that achieves a rate of $O(1/T)$ for arbitrary errors.

Convergence rates for SGD

Corollary:

SGD with constant stepwise achieves exponential convergence till error an error floor of

$\mathbb{E}\|w_{k+1} - w^*\|^2 \geq \epsilon \cdot O\left(\frac{L^2}{\lambda^2}\right)$ and after that achieves a rate of $O(1/T)$ for arbitrary errors.

How does SGD compare with GD?

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Total GD computational cost

$$O\left(T_\epsilon^{\text{GD}} \cdot \text{cost}(\nabla f)\right) = O\left(\text{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathcal{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Total GD computational cost

$$O\left(T_\epsilon^{\text{GD}} \cdot \text{cost}(\nabla f)\right) = O\left(\text{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total SGD computational cost

$$O\left(T_\epsilon^{\text{SGD}} \cdot \mathbb{E}\text{cost}(\nabla f_i)\right) = O\left(\frac{\text{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$

$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Proposition:

The function $f(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i\langle w, x_i\rangle}\right) + \frac{\lambda}{2}\|w\|^2$ is

- $\left(\frac{1}{n}\sum_i \|x_i\| + \lambda \cdot \max_{w\in\mathscr{W}} \|w\|\right)$-Lipschitz
- $\left(\frac{1}{4n}\sum_i \|x_i\|^2 + \lambda\right)$-smooth and
- λ-strongly convex

Let's make some assumptions:

$$\|x_i\|, \|w\| = O\left(\sqrt{d}\right)$$

$$\lambda = O(1)$$

Total GD computational cost

$$O\left(T_\epsilon^{\text{GD}} \cdot \text{cost}(\nabla f)\right) = O\left(\text{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total SGD computational cost

$$O\left(T_\epsilon^{\text{SGD}} \cdot \mathbb{E}\text{cost}(\nabla f_i)\right) = O\left(\frac{\text{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$

Note, cost doesn't depend on n!

$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Total GD computational cost

$$O\left(T_\epsilon^{\mathsf{GD}} \cdot \mathrm{cost}(\nabla f)\right) = O\left(\mathrm{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total GD computational cost

$$O\left(T_\epsilon^{\mathsf{SGD}} \cdot \mathbb{E}\mathrm{cost}(\nabla f_i)\right) = O\left(\frac{\mathrm{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$

$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

Computational complexity of GD

Total GD computational cost

$$O\left(T_\epsilon^{\text{GD}} \cdot \text{cost}(\nabla f)\right) = O\left(\text{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total GD computational cost

$$O\left(T_\epsilon^{\text{SGD}} \cdot \mathbb{E}\text{cost}(\nabla f_i)\right) = O\left(\frac{\text{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$

$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

- SGD is faster than GD (for regularized logistic regression and in the worst case) as long as

$$nd^2 \log\left(\frac{d}{\epsilon}\right) \geq \frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)$$

Computational complexity of GD

Total GD computational cost

$$O\left(T_\epsilon^{\mathrm{GD}} \cdot \mathrm{cost}(\nabla f)\right) = O\left(\mathrm{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$
$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total GD computational cost

$$O\left(T_\epsilon^{\mathrm{SGD}} \cdot \mathbb{E}\mathrm{cost}(\nabla f_i)\right) = O\left(\frac{\mathrm{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$
$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

- SGD is faster than GD (for regularized logistic regression and in the worst case) as long as

$$nd^2 \log\left(\frac{d}{\epsilon}\right) \geq \frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)$$
$$\implies \epsilon \geq \frac{1}{n}$$

Computational complexity of GD

Total GD computational cost

$$O\left(T_\epsilon^{\mathsf{GD}} \cdot \mathsf{cost}(\nabla f)\right) = O\left(\mathsf{nnz}(X) \cdot d \log\left(\frac{d}{\epsilon}\right)\right)$$

$$= O\left(nd^2 \log\left(\frac{d}{\epsilon}\right)\right)$$

Total GD computational cost

$$O\left(T_\epsilon^{\mathsf{SGD}} \cdot \mathbb{E}\mathsf{cost}(\nabla f_i)\right) = O\left(\frac{\mathsf{nnz}(X)}{n} \cdot \frac{1}{\epsilon} \cdot \frac{L^2}{\lambda^2} \log\left(\frac{R}{\epsilon}\right)\right)$$

$$= O\left(\frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)\right)$$

- SGD is faster than GD (for regularized logistic regression and in the worst case) as long as

$$nd^2 \log\left(\frac{d}{\epsilon}\right) \geq \frac{d^2}{\epsilon} \log\left(\frac{d}{\epsilon}\right)$$

$$\implies \epsilon \geq \frac{1}{n}$$

Sounds reasonable, especially in light of best case $\sim \frac{1}{n}$ generalization bounds

Beyond complexity, some thoughts

- The rates of SGD are in expectation.
- High probability bounds possible, e.g., by Markov's inequality on multiple runs of SGD (nobody does that in practice)
- A step of GD is trivially parallelizable, but SGD is inherently serial.
- Minibatches/Shuffling/stepsize selection??
- The generalization performance of these two algorithms is different!
- How about non-convex functions?

OK what do I do in practice?

The SGD quick start guide

Newcomers to stochastic gradient descent often find all of these design choices daunting, and it's useful to have simple rules of thumb to get going. We recommend the following:

1. Pick as large a minibatch size as you can given your computer's RAM.
2. Set your momentum parameter to either 0 or 0.9. Your call!
3. Find the largest constant stepsize such that SGD doesn't diverge. This takes some trial and error, but you only need to be accurate to within a factor of 10 here.
4. Run SGD with this constant stepsize until the empirical risk plateaus.
5. Reduce the stepsize by a constant factor (say, 10)
6. Repeat steps 4 and 5 until you converge.

While this approach may not be the most optimal in all cases, it's a great starting point and is good enough for probably 90% of applications we've encountered.

[Recht, Hardt, book 2021]

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k \in [T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \dfrac{R}{\beta L^2 T}$ satisfy

$$\min_{k \in [T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k \rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma \langle \nabla f(w_k), \nabla f_{s_k}(w_k) \rangle \leq \frac{\beta}{2}\|\gamma \nabla f_{s_k}(w_k)\|^2$$

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\beta\gamma^2}{2}\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2$$

$$\implies \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k)}{\gamma} + \frac{\gamma L^2 \beta}{2}$$

$$\implies \min_k \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_1) - f(w_0)}{\gamma T} + \frac{\gamma L^2 \beta}{2T}$$

SGD/GD on general non convex functions?

Theorem

Let $f(w)$ be a β-smooth function with L-bounded stoch. gradients (i.e., $\mathbb{E}_i \|\nabla f_i(w)\| \leq L$). Then, the gradients of SGD with step-size $\gamma = \frac{R}{\beta L^2 T}$ satisfy

$$\min_{k \in [T]} \mathbb{E}\|\nabla f(w_k)\|^2 \leq 2\sqrt{\frac{R\beta L^2}{T}}$$

$$f(w_{k+1}) - f(w_k) - \langle \nabla f(w_k), w_{k+1} - w_k \rangle \leq \frac{\beta}{2}\|w_k - w_{k+1}\|^2$$

This is a very slow rate, that is very conservative

$$\implies \mathbb{E}f(w_{k+1}) - \mathbb{E}f(w_k) + \gamma \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\beta\gamma^2}{2}\mathbb{E}\|\nabla f_{s_k}(w_k)\|^2$$

It also doesn't tell us anything about the quality of the solution that SGD finds

$$\implies \min_k \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}\|\nabla f(w_k)\|^2 \leq \frac{\mathbb{E}f(w_1) - f(w_0)}{\gamma T} + \frac{\gamma L^2 \beta}{2T}$$

Next Time:
More interesting Nonconvexity

reading list

Bubeck, S., 2015. Convex Optimization: Algorithms and Complexity. Foundations and Trends® in Machine Learning, 8(3-4), pp.231-357.
https://arxiv.org/pdf/1405.4980.pdf

Understanding Machine Learning: From Theory to Algorithms, https://www.cs.huji.ac.il/w~shais/UnderstandingMachineLearning/copy.html

Bottou, L., Curtis, F.E. and Nocedal, J., 2018. Optimization methods for large-scale machine learning. Siam Review, 60(2), pp.223-311.
Vancouver
https://arxiv.org/pdf/1606.04838v1.pdf

Hardt, M. and Recht, B., 2021. Patterns, predictions, and actions: A story about machine learning. arXiv preprint arXiv:2102.05242.
https://arxiv.org/pdf/2102.05242