Optimization for Data Science
ETH Zürich, FS 2022 261-5110-00L

Lecture 6: Stochastic Optimization

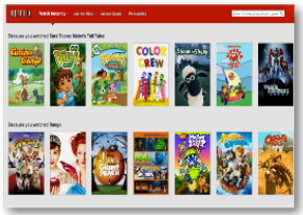
Bernd Gärtner
Niao He

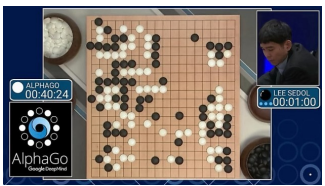
https://www.ti.inf.ethz.ch/ew/courses/ODS22/index.html

March 28, 2022

Introduction

- Stochastic optimization involves decision-making in the presence of randomness and lies at the heart of Data Science.





Stochastic Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x},\boldsymbol{\xi})] \tag{SO}$$

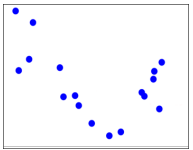
- $\boldsymbol{\xi}$ is a **random vector** with support $\Xi \subset \mathbb{R}^m$ and distribution P.
- For simplicity, assume $f(\mathbf{x},\boldsymbol{\xi})$ is continuously differentiable for any $\boldsymbol{\xi} \in \Xi$.
- In general, $F(x)$ and $\nabla F(x)$ are **hard to compute** even if P is given.
- In practice, P is often unknown and can be accessed through data.

An Important Special Case: Finite Sum Problems

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

- This can be viewed as a special case of stochastic optimization.
- $F(\mathbf{x}) = \mathbb{E}_\xi[f_\xi(\mathbf{x})]$, where ξ is uniformly distributed over $\{1, 2, \ldots, n\}$.

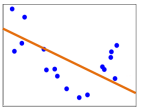
Application: Supervised Learning



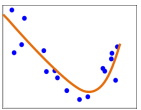
$(\mathbf{x}_i, y_i), y_i \sim h(\mathbf{x}_i)$
$i = 1, \ldots, n$

Data

- Linear model: $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$



- Nonlinear model: $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$



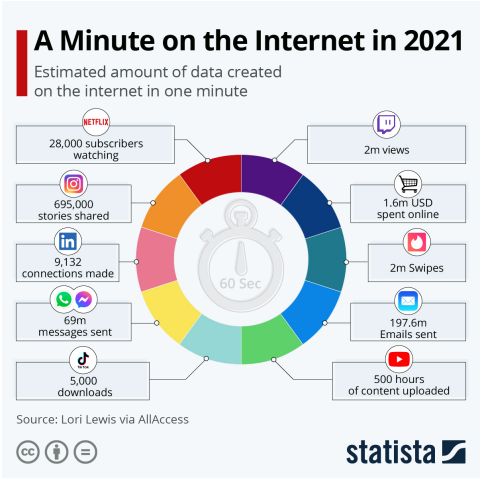
- Multi-layer network model:
  $h_{\mathbf{w}}(\mathbf{x}) = W_3^T g_2(W_2^T g_1(W_1^T \mathbf{x}))$

Model

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(h_{\mathbf{w}}(\mathbf{x}_i), y_i)$$

Optimization

Modern Big Data Challenge

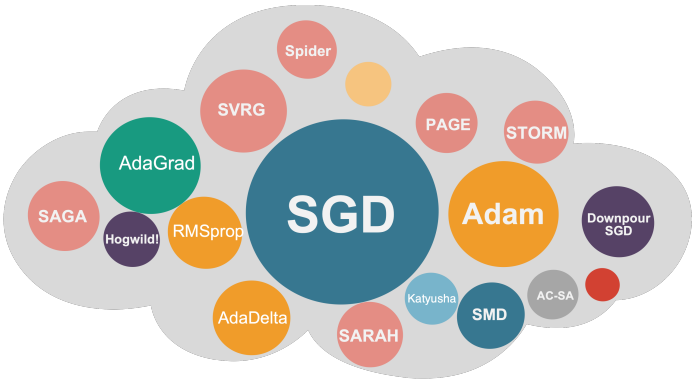


Big n !

- Cannot afford computing the gradient
- Cannot afford going through data many times

The Zoo of Stochastic Gradient Based Methods

- Stochastic Gradient Descent (SGD), first introduced by Robbins & Monro in 1951, has become one of the most popular algorithms for learning from big data.
- A plethora of SGD variants are developed in the past decade.
- Adam [Kingma & Ba, 2015] has over 100000 citations (despite a wrong proof)!



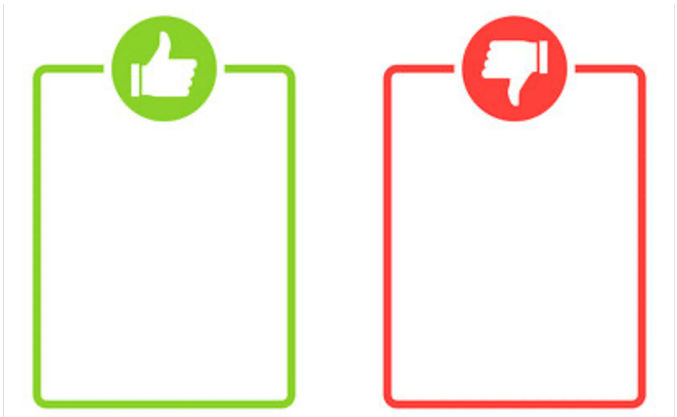
Survey

Q: Form small groups (3-5) with your neighbors and discuss (**3-5 mins**)

- If you have heard of or used SGD or any of its variants before, what's your observation on their performance, pros and cons? For which applications do they work well? Which variant of SGD is your favorite?
- If you have never learned about SGD before, what's your expectation of how SGD should work?

After your discussion, we will select groups to share your input.

Survey



Lecture Outline

Stochastic Gradient Descent

Adaptive Stochastic Gradient Methods

Convergence Analysis of SGD

Convergence of Adaptive SGD Methods

SGD for Finite Sum Problem

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

SGD:

Sample $i_t \in [n]$ uniformly at random

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f_{i_t}(\mathbf{x}_t)$$

- **Unbiasedness**: $\mathbb{E}_{i_t}[\nabla f_{i_t}(\mathbf{x})] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x}) = \nabla F(\mathbf{x})$.
- Each iteration is $O(n)$ cheaper than full gradient descent.

SGD for General Stochastic Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x}, \boldsymbol{\xi})]$$

SGD

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t), \text{ where } \boldsymbol{\xi}_t \overset{iid}{\sim} P(\boldsymbol{\xi})$$

- **Unbiasedness:** $\mathbb{E}[\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)|\mathbf{x}_t] = \nabla F(\mathbf{x}_t)$ under mild regularity conditions
- W.l.o.g., we always assume the stochastic gradient is unbiased.
- Note SGD is not a monotonic descent method.

Stepsize (or learning rate)

Q: Can we use fixed stepsize for SGD as in GD? Does it converge to the optimal solution (almost surely)?

A. Yes

B. No

- Stepsize should decrease to 0, $\gamma_t \to 0$
- For example, use polynomial rate $\gamma_t = O(t^{-a})$ with some $a > 0$
- In practice, use the form $\gamma_t = \frac{\gamma_0}{1+\beta t}$ and tune hyperparameters γ_0, β
- In deep learning, often adopt step decay - drop the learning rate by a factor every few epochs

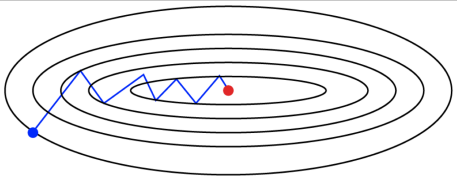
Example: Least Square Regression I

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \frac{1}{2n}\sum_{i=1}^{n}(\mathbf{a}_i^T\mathbf{x} - b_i)^2$$

GD:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \tfrac{\gamma}{n}\textstyle\sum_{i=1}^{n}(\mathbf{a}_i^T\mathbf{x}_t - b_i)\mathbf{a}_i$$

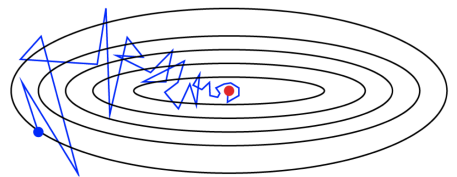
- Expensive iteration cost
- Fast convergence



SGD:

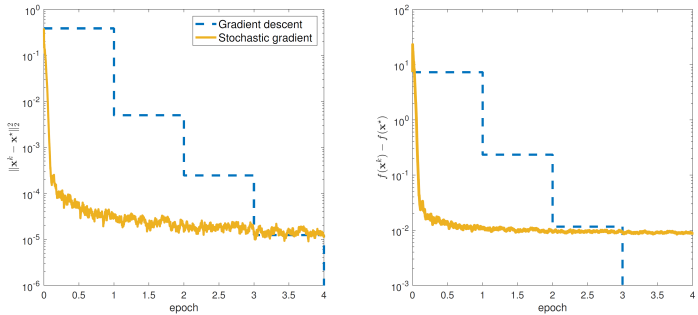
$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t(\mathbf{a}_{i_t}^T\mathbf{x}_t - b_{i_t})\mathbf{a}_{i_t}$$

- Cheap iteration cost
- Less stable, slow convergence



Example: Least Square Regression II

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \frac{1}{2n}\sum_{i=1}^{n}(\mathbf{a}_i^T\mathbf{x} - b_i)^2 \qquad (n = 10^4, d = 10^2)$$



(1 epoch = 1 full gradient)

From Volkan Cevher's EE 556 lecture note

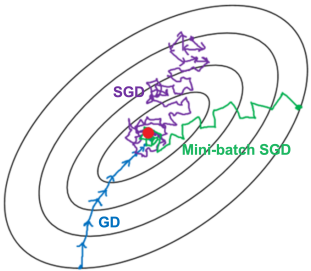
Simple Improvements

- **Mini-batch SGD**: use b random samples to construct gradient estimator

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \cdot \frac{1}{b} \sum_{j \in J, |J|=b} \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_j)$$

- **SGD with iterate averaging**:

$$\bar{\mathbf{x}}_t = \frac{1}{t} \sum_{\tau=1}^{t} \mathbf{x}_\tau$$



- Averaging and mini-batch sampling can help reduce the variance.
- Still, SGD can be very sensitive to the choice of stepsize.

Lecture Outline

Stochastic Gradient Descent

Adaptive Stochastic Gradient Methods

Convergence Analysis of SGD

Convergence of Adaptive SGD Methods

Adaptive Stochastic Gradient Methods

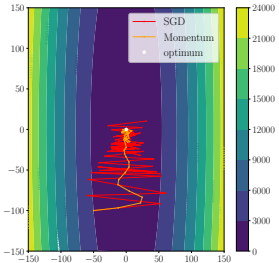
- Some limitations of SGD:
  - learning rate tuning
  - uniform learning rate for all coordinates

- Adaptive stepsizes are widely used in practice to improve the performance of SGD:
  - AdaGrad [Duchi, Hazan, & Singer, 2011]
  - RMSProp [Tieleman & Hinton, 2012]
  - ADAM [Kingma & Ba, 2015]
  - AMSGrad [Reddi, Kale, & Kumar,2018]
  - ....

Popular Variants

- Momentum SGD

$$\begin{cases} \mathbf{m}_t & = \alpha \mathbf{m}_{t-1} + (1-\alpha)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \gamma_t \mathbf{m}_t \end{cases}$$

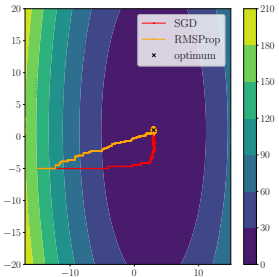


- AdaGrad

$$\begin{cases} \mathbf{v}_t & = \mathbf{v}_{t-1} + \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\epsilon + \sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \end{cases}$$

- RMSProp

$$\begin{cases} \mathbf{v}_t & = \beta \mathbf{v}_{t-1} + (1-\beta)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\varepsilon + \sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \end{cases}$$



ADAM

ADAM $\approx$ RMSProp + Momentum (>100K citations)

$$\begin{cases} \mathbf{v}_t & = \beta \mathbf{v}_{t-1} + (1-\beta) \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{m}_t & = \alpha \mathbf{m}_{t-1} + (1-\alpha) \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\varepsilon + \sqrt{\tilde{\mathbf{v}}_t}} \odot \tilde{\mathbf{m}}_t \end{cases}$$

- Exponential decay of previous information $\mathbf{m}_t, \mathbf{v}_t$.
- Note $\tilde{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta^t}$ and $\tilde{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\alpha^t}$ are bias-corrected estimates.
- In practice, α and β are chosen to be close to 1.

Numerical Illustration

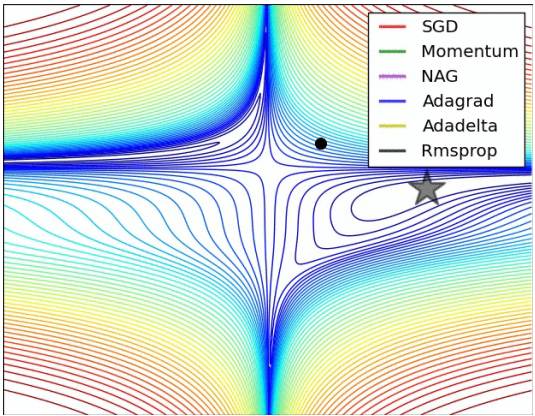


Image Credit: CS231n (`https://cs231n.github.io/neural-networks-3/`)

Half-way Summary

Learning objective: master the design principles and convergence analysis of SGD and its adaptive variants

- What is SGD?
- How to select stepsize and make it adaptive?

This Lecture.

- When and Why does SGD converge?
- Where does SGD converge to?
- Which SGD variant should we adopt in practice?

Next Lecture!

Lecture Outline

Stochastic Gradient Descent

Adaptive Stochastic Gradient Methods

Convergence Analysis of SGD

Convergence of Adaptive SGD Methods

Problem Settings

We first focus on the analysis of general stochastic optimization:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x}, \boldsymbol{\xi})]$$

$$\textbf{SGD}: \qquad \mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t), \text{ where } \boldsymbol{\xi}_t \overset{iid}{\sim} P(\boldsymbol{\xi})$$

- Three settings: (i) convex functions, (ii) strongly convex functions, (iii) smooth and strongly convex setting;
- Results also apply to finite sum problems.

Convergence for Stochastic Convex Problems

Theorem 6.1 (Convex, weighted averaging)

Suppose $F(\mathbf{x})$ *is convex and* $\mathbb{E}[\|\nabla f(\mathbf{x}, \boldsymbol{\xi})\|_2^2] \leq B^2, \forall \mathbf{x}$. *Then SGD satisfies that*

$$\mathbb{E}[F(\hat{\mathbf{x}}_T) - F(\mathbf{x}^*)] \leq \frac{R^2 + B^2 \sum_{t=1}^T \gamma_t^2}{2 \sum_{t=1}^T \gamma_t},$$

where $\hat{\mathbf{x}}_T := \sum_{t=1}^T \gamma_t \mathbf{x}_t / \sum_{t=1}^T \gamma_t$ *and* $\|\mathbf{x}_1 - \mathbf{x}^*\|_2 \leq R$.

- If $\gamma_t \equiv \frac{R}{B\sqrt{T}}$, $\mathbb{E}[F(\hat{\mathbf{x}}_T) - F(\mathbf{x}^*)] = O\left(\frac{BR}{\sqrt{T}}\right)$.
- This further implies the $O(1/\epsilon^2)$ sample complexity required by SGD.

Convergence for Stochastic Convex Problems

Proof of Theorem 6.1

Similar analysis as GD/subgradient descent, except bounds hold in expectation.

.... to be filled in ....

Convergence for Stochastic Strongly Convex Problems

Theorem 6.2 (Strong convex, diminishing stepsize, last iterate)

Assume $F(\mathbf{x})$ *is* μ*-strongly convex and* $\mathbb{E}\left[||\nabla f(\mathbf{x}, \boldsymbol{\xi})||_2^2\right] \leq B^2, \forall \mathbf{x}$, *then SGD with* $\gamma_t = \frac{\gamma}{t}$ $(\gamma > \frac{1}{2\mu})$ *satisfies*

$$\mathbb{E}\left[||\mathbf{x}_t - \mathbf{x}^*||_2^2\right] \leq \frac{C(\gamma)}{t},$$

where $C(\gamma) = \max\left\{\frac{\gamma^2 B^2}{2\mu\gamma - 1}, ||\mathbf{x}_1 - \mathbf{x}^*||_2^2\right\}$.

- If F is also L-smooth, this further implies that $\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^*)] = O\left(\frac{L \cdot C(\gamma)}{t}\right)$.
- The sample complexity required by SGD is $O(1/\epsilon)$ in this case.

Convergence for Stochastic Strongly Convex Problems

Proof of Theorem 6.2

Use key recursion from descent lemma:

$\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2] \leq (1 - 2\mu\gamma_t)\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2] + \gamma_t^2 B^2.$

.... to be filled in ....

Clicker Question

Consider the one-dimensional problem

$$\min_x \; F(x) := \frac{1}{2}\mathbb{E}_{\xi\sim N(0,1)}[(x-\xi)^2].$$

Run stochastic gradient descent with $x_1 = 10$ and stepsize $\gamma_t = \frac{1}{t}$. Then

$$\mathbb{E}[|x_{t+1} - x^*|^2] =?$$

A. $\mathbb{E}[|x_{t+1} - x^*|^2] = \frac{1}{t}$.
B. $\mathbb{E}[|x_{t+1} - x^*|^2] > \frac{1}{t}$.
C. $\mathbb{E}[|x_{t+1} - x^*|^2] < \frac{1}{t}$.

Remarks

- The previous example suggests that the convergence analysis is tight for SGD.
- So far, we see the necessity of diminishing stepsize and convexity for SGD to converge to an optimal solution.
- Consequently, SGD exhibits slow sublinear convergence rates.
- However, in practice, people often use constant stepsize for SGD regardless.

Q: What guarantees do we have if we use constant stepsize?

SGD under Constant Stepsize

Assume that

- $F(\mathbf{x}) := \mathbb{E}[f(\mathbf{x}, \boldsymbol{\xi})]$ is μ-strongly convex and L-smooth;
- The unbiased estimator satisfies that for all $\mathbf{x}$:

$$\mathbb{E}[\|\nabla f(\mathbf{x}, \boldsymbol{\xi})\|_2^2] \leq \sigma^2 + c\|\nabla F(\mathbf{x})\|_2^2.$$

Theorem 6.3 (strongly convex and smooth, constant step-size,[Bot16])

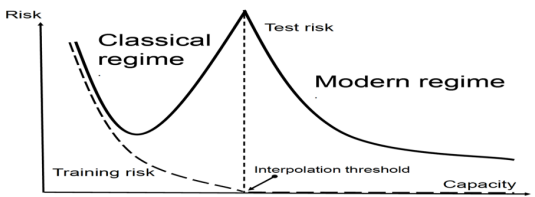
Under the above assumption, SGD with $\gamma_t = \gamma \leq \frac{1}{Lc}$ achieves:

$$\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^*)] \leq \frac{\gamma L \sigma^2}{2\mu} + (1 - \mu\gamma)^{t-1}(F(\mathbf{x}_1) - F(\mathbf{x}^*))$$

SGD under Constant Stepsize

- With constant stepsize, SGD converges linearly to a neighborhood around $\mathbf{x}^*$.
- Accuracy-convergence trade-off: Smaller stepsize γ implies better solution but slower rate.
- Strong Growth Condition: when $\sigma^2 = 0$, i.e., $\mathbb{E}[\|\nabla f(\mathbf{x}, \boldsymbol{\xi})\|_2^2] \leq c\|\nabla F(\mathbf{x})\|_2^2$, SGD with constant stepsize converges to the global optimum at a linear rate.
  - Consider $F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$, strong growth condition implies interpolation: at optimal solution $\mathbf{x}^*$, $\nabla f_i(\mathbf{x}^*) = 0, \forall i$.
  - Strong growth condition holds when F is smooth and satisfies Polyak- Lojasiewicz (PL) inequality. (Exercise)
  - Examples: linear regression or overparametrized neural network in the realizable case.

Modern Deep Learning Theory



	Classical (under-parameterized)	Modern (over-parameterized)
Generalization curve	U-shaped	Descending
Optimal model	Bottom of U (hard to find)	Any large model (easy to find)
Optimization landscape:	Locally convex Minimizers locally unique	Not locally convex Manifolds of minimizers Satisfies PL* condition
GD/SGD convergence	GD converges to local min SGD w. fixed learning rate does not converge	GD/SGD converge to global min SGD w. fixed learning rate converges exponentially

Figure: Modern regime of machine learning [Bel21]

Lecture Outline

Stochastic Gradient Descent

Adaptive Stochastic Gradient Methods

Convergence Analysis of SGD

Convergence of Adaptive SGD Methods

SGD Summary

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x},\boldsymbol{\xi})]$$

SGD : $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)$, where $\boldsymbol{\xi}_t \overset{iid}{\sim} P(\boldsymbol{\xi})$

	Convex	Strongly Convex
Convergence rate	$O\left(\frac{1}{\sqrt{t}}\right)$	$O\left(\frac{1}{t}\right)$
Sample complexity	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon}\right)$

Q: Is it possible to improve the sample complexity of SGD?

Lower Complexity Bound for Stochastic Optimization

- In the worst case, the sample complexity $O(1/\epsilon^2)$ and $O(1/\epsilon)$ for convex and strongly convex Lipschitz problems cannot be improved, for algorithms using only stochastic oracles. [Nemirovski & Yudin'83; Agarwal et al.'12].

  Stochastic Oracle: given input $\mathbf{x}$, stochastic oracle returns $G(\mathbf{x}, \boldsymbol{\xi})$ such that

  $$\mathbb{E}[G(\mathbf{x}, \boldsymbol{\xi})] \in \partial F(\mathbf{x}) \text{ and } \mathbb{E}[\|G(\mathbf{x}, \boldsymbol{\xi})\|_p^2] \leq M^2$$

  for some positive constant M and some $p \in [1, \infty]$.

- SGD is "optimal" for such problem classes

Lower Complexity Bound for Stochastic Optimization

Theorem 6.4 (Agarwal et al., 2012)

Let $X = B_\infty(r)$ be a ℓ_∞ ball with radius bounded by r on $\mathbb{R}^d$.

1. $\exists\ c_0 > 0$, *convex function f with $|f(x) - f(y)| < M\|x - y\|_\infty$, for any algorithm making T stochastic oracles with $1 \le p \le 2$ and generating a solution $\mathbf{x}_T$,*

$$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \ge \min\left\{c_0 M r\sqrt{\frac{d}{T}},\ \frac{Mr}{144}\right\}$$

2. $\exists\ c_1,\ c_2 > 0$, *μ-strongly convex function f, for any algorithm making T stochastic oracles with $p = 1$ and generating a solution $\mathbf{x}_T$,*

$$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \ge \min\left\{c_1\frac{M^2}{\mu^2 T},\ c_2 M r\sqrt{\frac{d}{T}},\ \frac{M^2}{1152\mu^2 d},\ \frac{Mr}{144}\right\}$$

Connections to Statistical Learning

Statistical learning often seeks to approximate SO by empirical risk minimization:

$$\min_{\mathbf{x}\in X} \underbrace{\mathbb{E}_{\boldsymbol{\xi}\sim P}[f(\mathbf{x},\boldsymbol{\xi})]}_{\text{expected risk}} \approx \underbrace{\frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x},\boldsymbol{\xi}_i)}_{\text{empirical risk}}.$$

- In statistical learning, assumes stochastic global oracle (instead of local gradient)
- Sample complexity of statistical learning is of order $O(\frac{d}{\epsilon^2})$ to ensure uniform convergence (without requiring convexity).

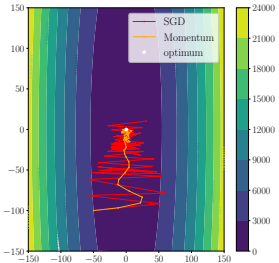
Theory-Practice Gap

- In theory, without imposing additional assumption or structure, it is impossible to achieve better rate than SGD.
- In practice, acceleration techniques such as momentum, adaptive pre-conditioning are heavily used.

Popular Variants

- Momentum SGD

$$\begin{cases} \mathbf{m}_t & = \alpha \mathbf{m}_{t-1} + (1-\alpha)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \gamma_t \mathbf{m}_t \end{cases}$$

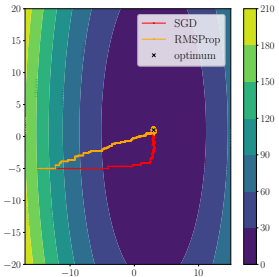


- AdaGrad

$$\begin{cases} \mathbf{v}_t & = \mathbf{v}_{t-1} + \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\epsilon + \sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \end{cases}$$

- RMSProp

$$\begin{cases} \mathbf{v}_t & = \beta \mathbf{v}_{t-1} + (1-\beta)\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\varepsilon + \sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \end{cases}$$



ADAM

ADAM $\approx$ RMSProp + Momentum (>100K citations)

$$\begin{cases} \mathbf{v}_t & = \beta \mathbf{v}_{t-1} + (1-\beta) \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^{\odot 2} \\ \mathbf{m}_t & = \alpha \mathbf{m}_{t-1} + (1-\alpha) \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\ \mathbf{x}_{t+1} & = \mathbf{x}_t - \frac{\gamma_0}{\varepsilon + \sqrt{\tilde{\mathbf{v}}_t}} \odot \tilde{\mathbf{m}}_t \end{cases}$$

- Exponential decay of previous information $\mathbf{m}_t, \mathbf{v}_t$.
- Note $\tilde{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta^t}$ and $\tilde{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1-\alpha^t}$ are bias-corrected estimates.
- In practice, α and β are chosen to be close to 1.

Generic Adaptive Scheme

The following scheme encapsulates these popular adaptive methods in a unified framework. [Reddi, Kale, & Kumar (2018)]

$$
\begin{aligned}
\mathbf{g}_t &= \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t) \\
\mathbf{m}_t &= \phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) \\
V_t &= \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) \\
\hat{\mathbf{x}}_t &= \mathbf{x}_t - \alpha_t V_t^{-1/2} \mathbf{m}_t \\
\mathbf{x}_{t+1} &= \operatorname*{argmin}_{\mathbf{x} \in X} \{(\mathbf{x} - \hat{\mathbf{x}}_t)^T V_t^{1/2} (\mathbf{x} - \hat{\mathbf{x}}_t)\}
\end{aligned}
$$

Popular Examples

- **SGD**

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \mathbf{g}_t, \quad \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \mathbb{I}$$

- **AdaGrad**

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \mathbf{g}_t, \quad \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = \frac{\mathrm{diag}(\sum_{\tau=1}^t \mathbf{g}_\tau^2)}{t}$$

- **Adam**

$$\phi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = (1-\beta_1)\sum_{\tau=1}^t \beta_1^{t-\tau}\mathbf{g}_\tau, \quad \psi_t(\mathbf{g}_1, \ldots, \mathbf{g}_t) = (1-\beta_2)\mathrm{diag}(\sum_{\tau=1}^t \beta_2^{t-\tau}\mathbf{g}_\tau^2)$$

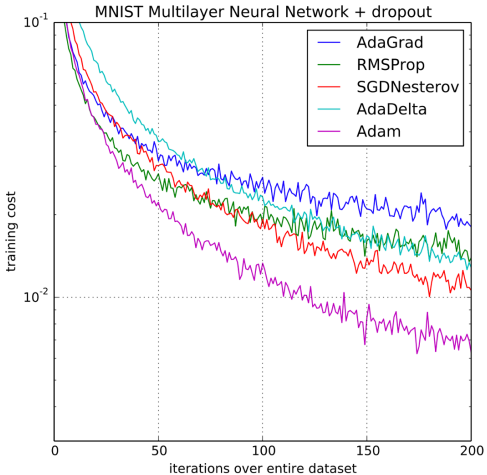
In other words, $\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$, $V_t = \beta_2 V_{t-1} + (1-\beta_2)\mathrm{diag}(\mathbf{g}_t^2)$.

What do we know in practice?

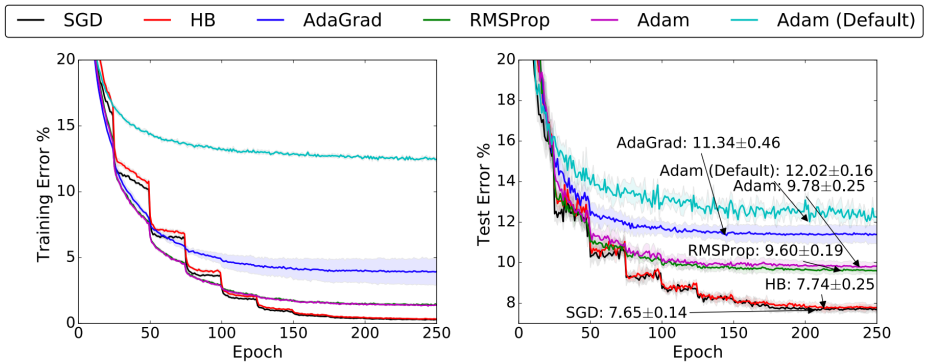
Adaptive methods

- Less sensitive to parameter tuning and adapt to sparse gradients.
- Outperform SGD for NLP tasks, training generative adversarial networks (GANs), deep reinforcement learning, etc., but are less effective in computer vision tasks.
- Tend to overfit and generalize worse than their non-adaptive counterparts [Wil17].
- Often display faster initial progress on the training set, but their performance quickly plateaus on the testing set [Wil17].

Some Good Stories



Some Bad Stories



(a) CIFAR-10 (Train) **(b)** CIFAR-10 (Test)

What do we know in theory?

- SGD with momentum has no acceleration even for some convex quadratic functions.
- For convex problems, Adagrad does converge, but RMSProp and Adam may not converge when $\beta_1 < \sqrt{\beta_2}$ (same for decreasing β_1 over time).

The Non-Convergence of Adam

Counterexample: consider a one-dimensional problem:

$$X = [-1, 1], \quad f(x, \xi) = \begin{cases} Cx, & \text{if } \xi = 1 \\ -x, & \text{if } \xi = 0 \end{cases}, \text{ where } P(\xi = 1) = p = \frac{1+\delta}{C+1}.$$

- Here $F(x) = \mathbb{E}[f(x, \xi)] = \delta x$ and $x^* = -1$.
- The Adam step is $x_{t+1} = x_t - \gamma_0 \Delta_t$ with $\Delta_t = \frac{\alpha m_t + (1-\alpha) g_t}{\sqrt{\beta v_t + (1-\beta) g_t^2}}$
- For large enough $C > 0$, one can show that $\mathbb{E}[\Delta_t] \leq 0$.
- The Adam steps keep drifting away from the optimal solution $x^* = -1$.

A Convergent Adam-type Algorithm

AMSGrad [Reddi, Kale, & Kumar (2018)]

Algorithm 2 AMSGRAD

Input: $x_1 \in \mathcal{F}$, step size $\{\alpha_t\}_{t=1}^T$, $\{\beta_{1t}\}_{t=1}^T$, β_2
Set $m_0 = 0$, $v_0 = 0$ and $\hat{v}_0 = 0$
for $t = 1$ **to** T **do**
$g_t = \nabla f_t(x_t)$
$m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$
$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ and $\hat{V}_t = \mathrm{diag}(\hat{v}_t)$
$x_{t+1} = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(x_t - \alpha_t m_t / \sqrt{\hat{v}_t})$
end for

- Use maximum value for normalizing the running average of the gradient.
- Ensure non-increasing stepsize and avoid pitfalls of Adam and RMSProp.
- Allow long-term memory of past gradients.

Summary

Learning objective: master the design principles and convergence analysis of SGD and its adaptive variants

- What is SGD?
- How to select stepsize and make it adaptive?
- When and Why does SGD or its variant converge?
- Where does SGD converge to?
- Which SGD optimizer to use in practice?

SGD Tricks Beyond

- SGD with random shuffling
- Grading clipping
- Noisy SGD
- Parallelizing SGD
- Variance reduction (Next Lecture!)
- ... ...

Bibliography


L. Bottou, F. E. Curtis, and J. Nocedal.
Optimization methods for large-scale machine learning.
SIAM Review, 2016.


A. Nemirovski and D. Yudin.
Problem complexity and method efficiency in optimization.
Wiley-Interscience, 1983.


A. Agarwal, P. Bartlett, P. Ravikumar, M. Wainwright.
Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization.
IEEE Transactions on Information Theory, 2012.


A. Nemirovski, A. Juditsky, G. Lan, A. Shapiro.
Robust stochastic approximation approach to stochastic programming.
SIAM Journal on optimization, 2009.


M. Belkin
Fit without fear: remarkable mathematical phenomena of deep learning through the prism of Interpolation.
Acta Numerica, 2021.

Bibliography (cont'd)

J. Duchi, E. Hazan, Y. Singer

Adaptive subgradient methods for online learning and stochastic optimization.

Journal of machine learning research, 12(7), 2011.

D. P. Kingma, and J. Ba.

Adam: A method for stochastic optimization.

arXiv preprint arXiv:1412.6980, 2014.

S.J. Reddi, S. Kale, S. Kumar

On the convergence of adam and beyond.

International Conference on Learning Representations,2018.

A. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht.

The Marginal Value of Adaptive Gradient Methods in Machine Learning.

Neural Information Processing Systems, 2017.

Supplementary Material

Proof of Theorem 6.1

Proof.

- First, $\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2 = \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 - 2\gamma_t \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^T(\mathbf{x}_t - \mathbf{x}^*) + \gamma_t^2 \|\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)\|_2^2$.
- By law of total expectation,

$$\begin{aligned}
\mathbb{E}[\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^T(\mathbf{x}_t - \mathbf{x}^*)] &= \mathbb{E}[\mathbb{E}[\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^T(\mathbf{x}_t - \mathbf{x}^*)|\mathbf{x}_t]] \\
&= \mathbb{E}[\mathbb{E}[\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)|\mathbf{x}_t]^T(\mathbf{x}_t - \mathbf{x}^*)] \\
&= \mathbb{E}[\nabla F(\mathbf{x}_t)^T(\mathbf{x}_t - \mathbf{x}^*)] \\
&\geq \mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^*)].
\end{aligned}$$

- This leads to the recursion:

$$\gamma_t \mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^*)] \leq \frac{1}{2}\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2] - \frac{1}{2}\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2] + \frac{1}{2}\gamma_t^2 B^2.$$

- The result follows by telescoping the sum from $t = 1$ to T.

□

Proof of Theorem 6.2

Proof.

- First, $\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2 = \|\mathbf{x}_t - \mathbf{x}^*\|_2^2 - 2\gamma_t \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^T(\mathbf{x}_t - \mathbf{x}^*) + \gamma_t^2 \|\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)\|_2^2$.
- By law of total expectation and strong convexity,

$$\mathbb{E}[\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)^T(\mathbf{x}_t - \mathbf{x}^*)] = \mathbb{E}[\nabla F(\mathbf{x}_t)^T(\mathbf{x}_t - \mathbf{x}^*)] \geq \mu \mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2].$$

- This leads to the recursion:

$$\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2^2] \leq (1 - \frac{2\mu\gamma}{t})\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2] + \frac{\gamma^2 B^2}{t^2}.$$

- The result follows by induction.

□