535520: Optimization Algorithms

Lecture 6 — Stochastic Gradient Descent

Ping-Chun Hsieh (謝秉均)

October 14, 2024

This Lecture

1. Nesterov's Accelerated Gradient and Lower Bound

2. Stochastic Gradient Descent

- Reading Material:
  - Chapter 4 of the textbook "Optimization Methods for Large-Scale Machine Learning" by Leon Bottou, Frank Curtis, and Jorge Nocedal.
    - Available at https://arxiv.org/abs/1606.04838

Gradient-Based Learning Applied to Document Recognition

YANN LECUN, MEMBER, IEEE, LÉON BOTTOU, YOSHUA BENGIO, AND PATRICK HAFFNER

Invited Paper

Multilayer neural networks trained with the back-propagation algorithm constitute the best example of a successful gradient-based learning technique. Given an appropriate network architecture, gradient-based learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns, such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional neural networks, which are specifically designed to deal with the variability of two dimensional (2-D) shapes, are shown to outperform all other techniques.

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called graph transformer networks (GTN's), allows such multimodule systems to be trained globally using gradient-based methods so as to minimize an overall performance measure.

Two systems for online handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of graph transformer networks.

A graph transformer network for reading a bank check is also described. It uses convolutional neural network character recognizers combined with global training techniques to provide record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

Keywords— *Convolutional neural networks, document recognition, finite state transducers, gradient-based learning, graph transformer networks, machine learning, neural networks, optical character recognition (OCR).*

NN	Neural network.
OCR	Optical character recognition.
PCA	Principal component analysis.
RBF	Radial basis function.
RS-SVM	Reduced-set support vector method.
SDNN	Space displacement neural network.
SVM	Support vector method.
TDNN	Time delay neural network.
V-SVM	Virtual support vector method.

I. Introduction

Over the last several years, machine learning techniques, particularly when applied to NN's, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning and less on hand-designed heuristics. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-crafted feature extraction can be advantageously replaced by carefully designed learning

[Proceedings of IEEE, 1998]

Large-Scale Machine Learning with Stochastic Gradient Descent

[COMPSTATS 2010]

Léon Bottou

NEC Labs America, Princeton NJ 08542, USA
leon@bottou.org

Abstract. During the last decade, the data sizes have grown faster than the speed of processors. In this context, the capabilities of statistical machine learning methods is limited by the computing time rather than the sample size. A more precise analysis uncovers qualitatively different tradeoffs for the case of small-scale and large-scale learning problems. The large-scale case involves the computational complexity of the underlying optimization algorithm in non-trivial ways. Unlikely optimization algorithms such as stochastic gradient descent show amazing performance for large-scale problems. In particular, second order stochastic gradient and averaged stochastic gradient are asymptotically efficient after a single pass on the training set.

Keywords: Stochastic gradient descent, Online learning, Efficiency

Leon Bottou (1965-present)
Currently at Meta AI

Wasserstein Generative Adversarial Networks

[ICML 2017]

Martin Arjovsky [1] **Soumith Chintala** [2] **Léon Bottou** [1 2]

Abstract

We introduce a new algorithm named WGAN, an alternative to traditional GAN training. In this new model, we show that we can improve the stability of learning, get rid of problems like mode collapse, and provide meaningful learning curves useful for debugging and hyperparameter searches. Furthermore, we show that the corresponding optimization problem is sound, and provide extensive theoretical work highlighting the deep connections to different distances between distributions.

The typical remedy is to add a noise term to the model distribution. This is why virtually all generative models described in the classical machine learning literature include a noise component. In the simplest case, one assumes a Gaussian noise with relatively high bandwidth in order to cover all the examples. It is well known, for instance, that in the case of image generation models, this noise degrades the quality of the samples and makes them blurry. For example, we can see in the recent paper (Wu et al., 2016) that the optimal standard deviation of the noise added to the model when maximizing likelihood is around 0.1 to each pixel in a generated image, when the pixels were al-

ON THE LIMITED MEMORY BFGS METHOD FOR LARGE SCALE OPTIMIZATION

by

Dong C. Liu and Jorge Nocedal

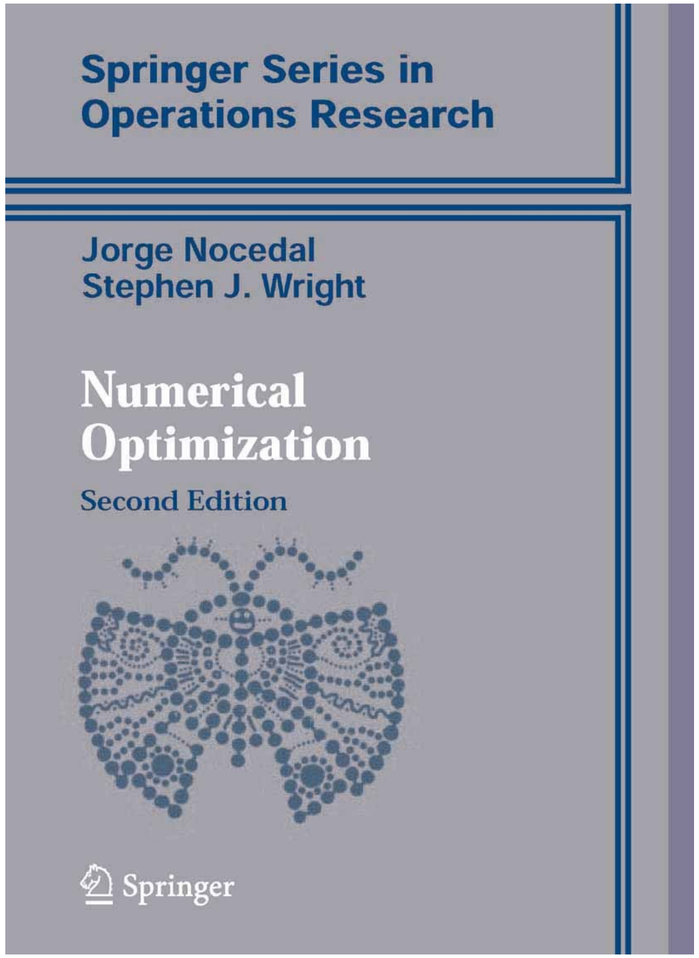
ABSTRACT

We study the numerical performance of a limited memory quasi-Newton method for large scale optimization, which we call the L-BFGS method. We compare its performance with that of the method developed by Buckley and LeNir (1985), which combines cyles of BFGS steps and conjugate direction steps. Our numerical tests indicate that the L-BFGS method is faster than the method of Buckley and LeNir, and is better able to use additional storage to accelerate convergence. We show that the L-BFGS method can be greatly accelerated by means of a simple scaling. We then compare the L-BFGS method with the partitioned quasi-Newton method of Griewank and Toint (1982a). The results show that, for some problems, the partitioned quasi-Newton method is clearly superior to the L-BFGS method. However we find that for other problems the L-BFGS method is very competitive due to its low iteration cost. We also study the convergence properties of the L-BFGS method, and prove global convergence on uniformly convex problems.

Key words: large scale nonlinear optimization, limited memory methods, partitioned quasi-Newton method, conjugate gradient method.

Abbreviated title: Limited memory BFGS.

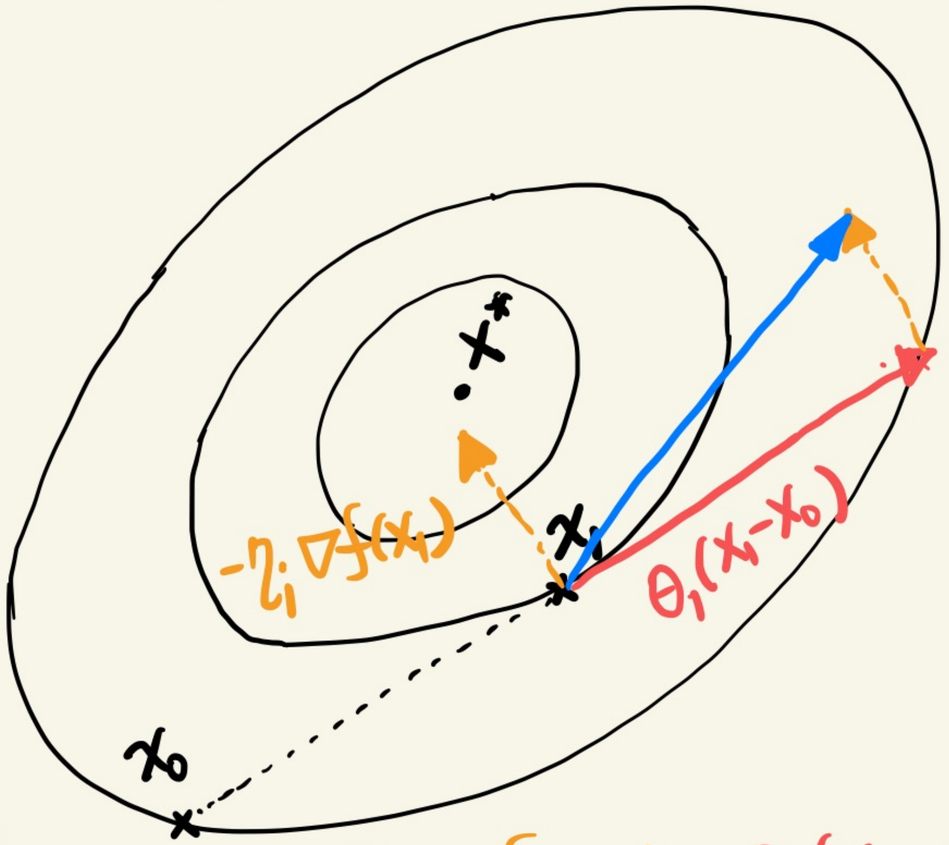
Limited-memory BFGS
[Mathematical Programming, 1989]



Jorge Nocedal (1952-present)
Northwestern University

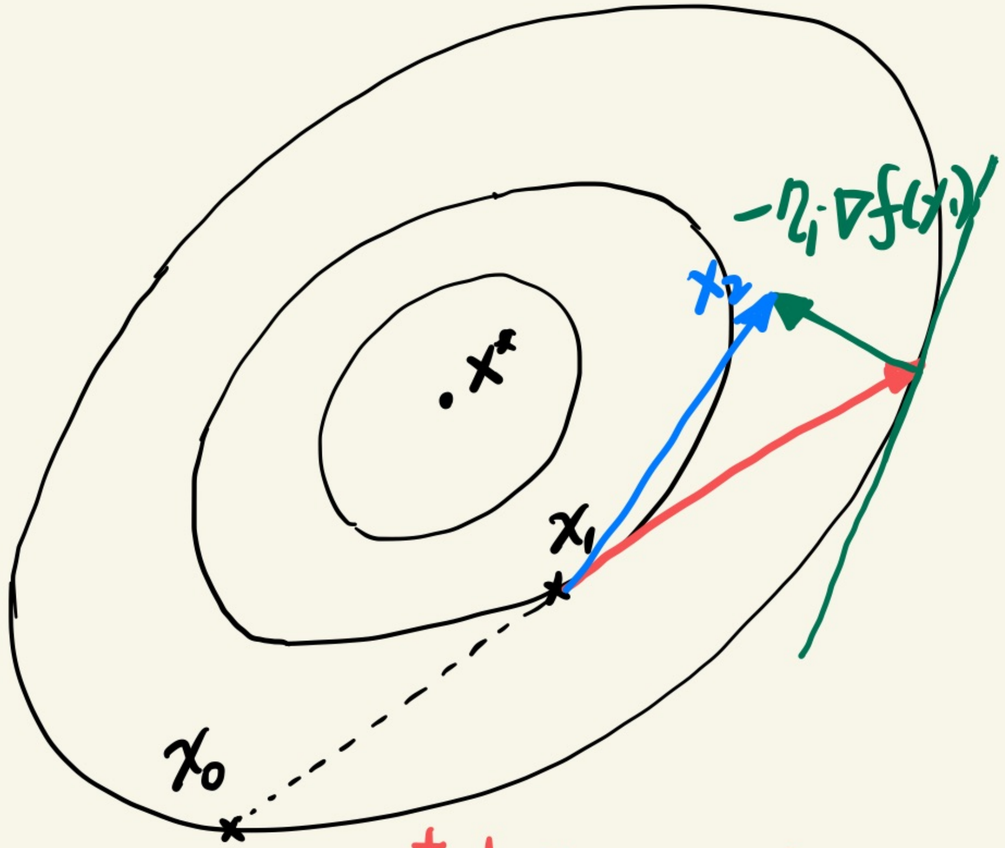
Visualization: Polyak's Momentum vs Nesterov's Method

Polyak's Momentum



$$x_{t+1} = x_t - \eta_t \nabla f(x_t) + \theta_t (x_t - x_{t-1})$$

Nesterov's Method



$$x_{t+1} = \underbrace{x_t + \frac{t-1}{t+2}(x_t - x_{t-1})}_{=y_t} - \eta_t \nabla f(y_t)$$

Comparison: GD and NAG

- For smooth and convex problems, the iteration complexity is:

	Convergence Rate
GD	$f(x_t) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{t^1}$
NAG	$f(x_t) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{(t+1)^2}$

This is a remarkable improvement over GD!

Definition: Iteration Complexity

Definition: Given any $\epsilon > 0$, the **iteration complexity** of an algorithm is defined as the number of iterations needed to reach an ϵ-optimal solution.

Let's quickly verify the following:

	Convergence Rate	Iteration Complexity
GD	$f(x_t) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{t^1}$	$O(\frac{1}{\epsilon})$
Nesterov's	$f(x_t) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{(t+1)^2}$	$O(\frac{1}{\sqrt{\epsilon}})$

A Natural Question: Is NAG optimal?

A Lower Bound

- An Interesting Fact: No first-order methods can improve over Nesterov's method in general (in other words, Nesterov's method is optimal in terms of "first-order oracle complexity")

- Theorem: There exists a convex and L-smooth function such

$$f(x_t) - f(x^*) \geq \frac{3L\|x_0 - x^*\|^2}{32(t+1)^2}$$

where $x_\tau \in x_0 + \text{span}\{\nabla f(x_0), \cdots, \nabla f(x_{\tau-1})\}$, for all $1 \leq \tau \leq t$

- Question: What's the implication of the above condition of x_τ?

Nemirovsky and Yudin, "Problem complexity and method efficiency in optimization", 1983

Constructing an Example for the Lower Bound (1/3)

$$\min_{x \in \mathbb{R}^{2t+1}} f(x) = \frac{L}{4}\left(\frac{1}{2}x^\top A x - e_1^\top x\right)$$

$$A \in \mathbb{R}^{(2t+1)\times(2t+1)} \qquad e_1 \in \mathbb{R}^{2t+1}$$

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & 0 & -1 & 2 \end{bmatrix} \qquad e_1 = \begin{bmatrix} 1 \\ 0 \\ \cdots \\ \cdots \\ \cdots \\ 0 \end{bmatrix}$$

Fact 1: f is convex and L-smooth

Fact 2: $\nabla f(x) = \frac{L}{4}Ax - \frac{L}{4}e_1$

Fact 3: Optimal solution $x^* = (x_1^*, \cdots, x_n^*)$ is given by $x_i^* = 1 - \frac{i}{2t+2}$

Fact 4: $\|x^*\|^2 \leq \frac{2t+2}{3}$ and $f(x^*) = \frac{L}{8}\left(\frac{1}{(2t+1)+1} - 1\right)$

Constructing an Example for the Lower Bound (2/3)

$$\min_{x \in \mathbb{R}^{2t+1}} f(x) = \frac{L}{4}\left(\frac{1}{2}x^\top A x - e_1^\top x\right)$$

$$A \in \mathbb{R}^{(2t+1)\times(2t+1)} \quad e_1 \in \mathbb{R}^{2t+1}$$

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & 0 & -1 & 2 \end{bmatrix} \quad e_1 = \begin{bmatrix} 1 \\ 0 \\ \cdots \\ \cdots \\ \cdots \\ 0 \end{bmatrix}$$

Fact 5: $\underbrace{\text{Span}\{\nabla f(x_0), \cdots \nabla f(x_{t-1})\}}_{=:C_t} = \text{Span}\{e_1, \cdots, e_t\}$ if $x_0 = 0$

Under **any** first-order method, in each iteration, one can expand the search space by **at most 1 dimension**

Constructing an Example for the Lower Bound (3/3)

Let's derive the lower bound!

Step 1: Given that $x_0 = 0$, we have

$$f(x_t) \geqslant \inf_{x \in C_t} f(x) = \frac{L}{8}\left(\frac{1}{t+1} - 1\right)$$

Step 2: Therefore, we have

$$\frac{f(x_t) - f(x^*)}{\|x_0 - x^*\|^2} \geqslant \frac{\frac{L}{8}\left(\frac{1}{t+1} - \frac{1}{2t+2}\right)}{\frac{1}{3}(2t+2)} = \frac{3L}{32(t+1)^2}$$

Another Interpretation of NAG

Aleksandar Botev, Guy Lever, and David Barber, “Nesterov’s Accelerated Gradient and Momentum as approximations to Regularised Update Descent,” IJCNN 2017

Remark: An Alternative Expression of NAG Updates

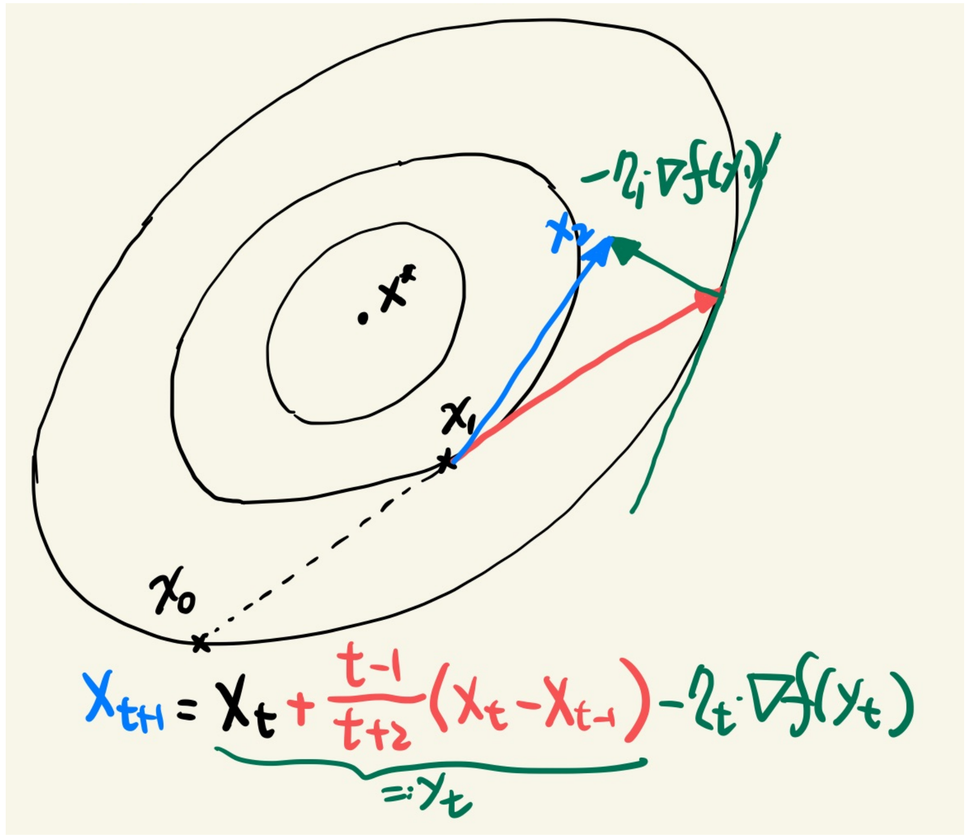
$$x_{t+1} = y_t - \eta_t \nabla f(y_t)$$

$$y_t = x_t + \beta_t (x_t - x_{t-1})$$

$$v_{t+1} = x_{t+1} - x_t \quad \longleftrightarrow$$

$$x_{t+1} = x_t + v_{t+1}$$

$$v_{t+1} = \beta_t v_t - \eta_t \nabla f(x_t + \beta_t v_t)$$



Interpreting NAG as GD on Regularized Function

Define $F(x_t, v_t) := f(x_t + \beta_t v_t) + \frac{\gamma}{2}\|v_t\|^2$

(Clearly, the x^* that minimizes F also minimizes f)

Suppose we take a gradient step of F:

$$v_{t+1} = v_t - \eta_t\big(\nabla f(x_t + \beta_t v_t) + \gamma_t v_t\big) = \underbrace{(1 - \eta_t\gamma_t)}_{=:\beta_t} v_t - \eta_t \nabla f(x_t + \beta_t v_t)$$

Then, we arrive at the NAG update

$$x_{t+1} = x_t + v_{t+1}$$

$$v_{t+1} = \beta_t v_t - \eta_t \nabla f(x_t + \beta_t v_t)$$

Q: Any problems that cannot be easily solved by GD?

Stochastic Optimization in Machine Learning

- Optimization problems in ML arise mainly in two forms:

1. Expected risk/loss minimization

$$F^* := \min_{x \in X} \; F(x), \text{where } F(x) := \mathbb{E}[f(x; \varepsilon)]$$

where ε is the randomness (possibly unknown) in our problem

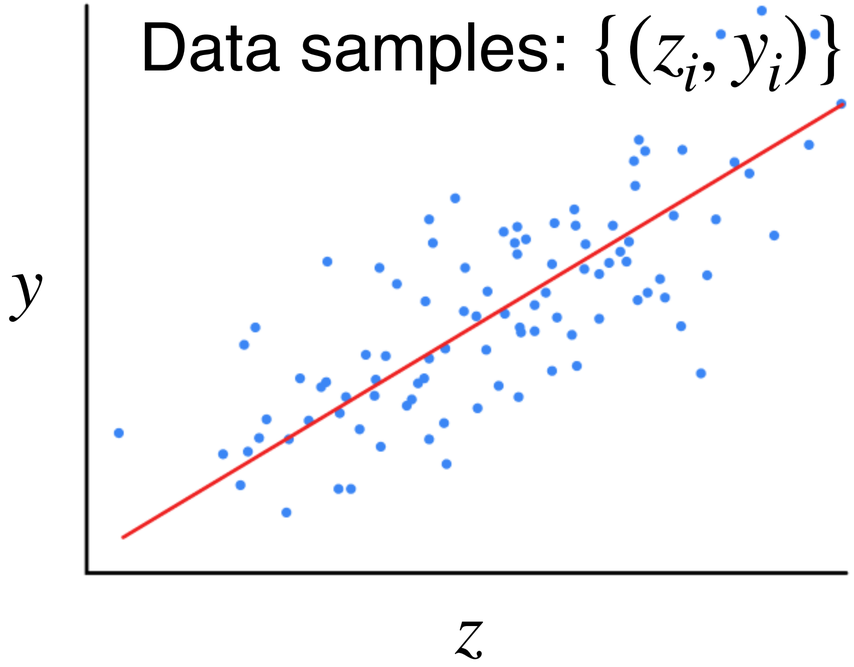
2. Empirical risk minimization (aka Finite-sum problem)

$$F^* := \min_{x \in X} \; F(x), \text{where } F(x) := \frac{1}{n} \sum_{i=1}^{n} f(x; d_i)$$

where $\{d_i\}_{i=1}^{n}$ are n random data samples

Example: Regression With Empirical Risk Minimization

- Example: Regression problems



- Linear model (a, b to be learned):

$$f(z) = az + b$$

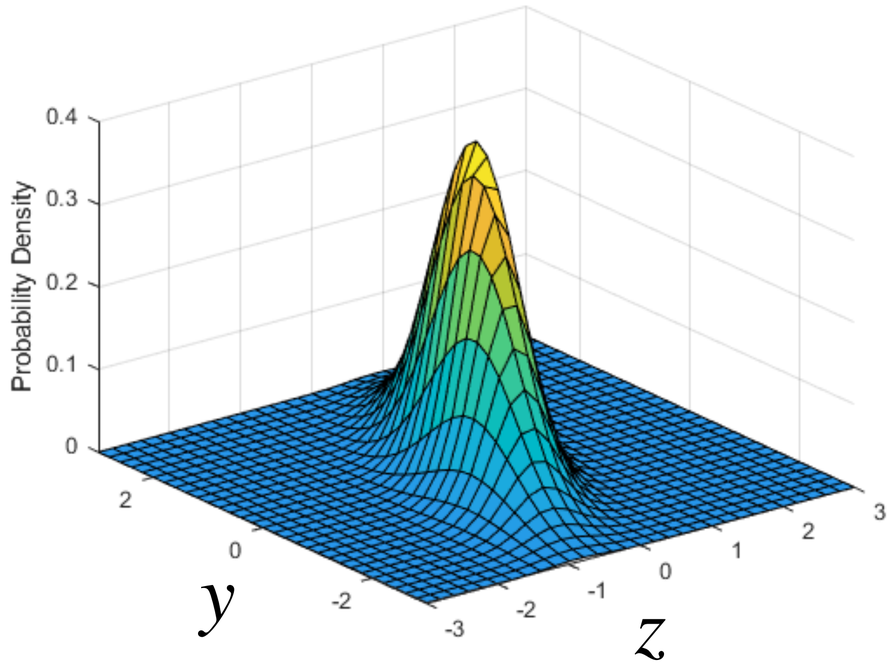
- Empirical risk minimization:

$$(a^*, b^*) \in \arg\min_{a,b} \frac{1}{n} \sum_{i=1}^{n} (f(z_i) - y_i)^2$$

In this problem, $x \equiv (a, b)$

Example: Regression With an Expected Loss

- Example: Regression problems with a generative model



Probability density of feature-label pairs (z, y)

- Linear model (a, b to be learned):

$$f_{a,b}(z) = az + b$$

- Minimization of expected loss:

$$(a^*, b^*) \in \arg\min_{a,b} \mathbb{E}_{(z,y)}\left[(f_{a,b}(z) - y)^2\right]$$

In this problem, $x \equiv (a, b)$

Issues With GD in Empirical Risk Minimization

- Apply GD to minimize the finite-sum loss:

$$x_{k+1} = x_k - \eta_k \cdot \nabla_x F(x)|_{x=x_k}$$
$$= x_k - \eta_k \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla_x f(x_k; d_i)$$

- Question: Any issues?

1. If n is large, then each GD iteration is very expensive

2. Hence, it takes a lot of time before one can make a bit of progress

Stochastic GD for Empirical Risk Minimization

- Idea: Use sampling to estimate expectation [Robbins & Monro, 1951]

At each iteration, we randomly pick an integer $i \in \{1,2,\cdots,n\}$

$$x_{k+1} = x_k - \eta_k \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla f(x_k; d_i) \quad \Rightarrow \quad x_{k+1} = x_k - \eta_k \cdot \nabla f(x_k; d_i)$$

(GD) (SGD)

- The update requires only gradient of one data sample d_i
- $\nabla f(x_k; d_i)$ is an unbiased estimate of $\nabla f(x_k)$, i.e., $\mathbb{E}[\nabla f(x_k; d_i)] = \nabla f(x_k)$
- Advantage: One iteration under SGD is n times faster than GD

H. Robbins, S. Monro, “A stochastic approximation method,” The Annals of Mathematical Statistics, 1951

Intuition: Why Can SGD Work?

- Consider the following toy example: $\min_x \quad f(x) = \frac{1}{2}\sum_{i=1}^{n}(a_i x - b_i)^2$
- To find x^*, we solve $f'(x) = 0$ and get $x^* = \frac{\sum_i a_i b_i}{\sum_i a_i^2} = \frac{\sum_i a_i^2 (b_i/a_i)}{\sum_i a_i^2}$
- Minimum of a single $f(x; d_i) = \frac{1}{2}(a_i x - b_i)^2$ is $x_i^* = b_i/a_i$
- Note that $x^* \in [\min_i x_i^*, \max_i x_i^*] =: S$
- If we have a scalar x outside S, then $\nabla f(x; d_i)$ has the same sign as $\nabla f(x)$

$$\nabla f(x; d_i) = a_i(a_i x - b_i) \qquad \nabla f(x) = \sum_i a_i(a_i x - b_i)$$

- In this case, using $\nabla f(x; d_i)$ instead of $\nabla f(x)$ still ensures improvement
- However, no guarantee when $x \in S$

(Example Credit: Suvrit Sra)

Issues With GD in Expected Risk Minimization

- Apply GD to minimize the expected loss:

$$x_{k+1} = x_k - \eta_k \cdot \nabla_x F(x)|_{x=x_k}$$

$$= x_k - \eta_k \cdot \mathbb{E}_\varepsilon[\nabla_x f(x_k; \varepsilon)]$$

- Question: Any issues?

1. Distribution / statistics of ε is unknown (i.e., a learning setting)

2. Expectation usually involves multi-dimensional integral, which is computationally expensive

Stochastic GD for Expected Risk Minimization

- Idea: Use sampling to estimate expectation [Robbins & Monro, 1951]

$$x_{k+1} = x_k - \eta_k \cdot \mathbb{E}[\nabla f(x_k; \varepsilon_k)] \quad \Rightarrow \quad x_{k+1} = x_k - \eta_k \cdot g(x_k; \varepsilon_k)$$

(GD) (SGD)

- $g(x_k; \varepsilon_k)$ is an unbiased estimate of $\mathbb{E}[\nabla f(x; \varepsilon_k)]$
- Equivalently: $g(x_k; \varepsilon_k) = \mathbb{E}[\nabla f(x_k; \varepsilon_k)] + \underbrace{\varepsilon}_{\text{zero mean noise}}$
- $g(x_k; \varepsilon_k)$ is constructed using one or multiple samples (= mini-batch)
- Advantage: SGD has a low computational cost in each iteration

H. Robbins, S. Monro, “A stochastic approximation method,” The Annals of Mathematical Statistics, 1951

A Broader View of SGD

- SGD is used for finding a stationary point x with $\nabla F(x) = 0$ (why?)

- More generally, SGD can be used for finding the roots of a function defined as $G(x) := \mathbb{E}[g(x; \varepsilon)]$

- This is known as "*stochastic approximation*" (SA) in general
  - SGD is a special case of SA
  - SGD has a "*Markovian*" structure!
  - Many RL algorithms are based on SA!

Other Algorithms That Look Like SGD (I): TD Learning

Temporal Difference (TD) Learning in RL

(Bellman Equation) Under a stationary policy π, for every state s

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s),s'\sim P(\cdot|s,a)}\Big[R(s,a) + \gamma V^\pi(s')\Big]$$

(Equivalently) $$\mathbb{E}_{a\sim\pi(\cdot|s),s'\sim P(\cdot|s,a)}\Big[\underbrace{R(s,a) + \gamma V^\pi(s') - V^\pi(s)}_{g(V^\pi)}\Big] = 0$$

TD(0) algorithm: learn the value function $V^\pi(s)$ without knowing P and R

$$V(s_t) \leftarrow V(s_t) - \eta_t\Big(V(s_t) - (r_{t+1} + \gamma V(s_{t+1}))\Big)$$

- $r_{t+1} + \gamma V(s_{t+1})$ is the estimated return (called TD target)
- $r_{t+1} + \gamma V(s_{t+1}) - V(s_t) =: \delta_t$ is called the TD error

Other Algorithms That Look Like SGD (II): Q-Learning (1/2)

Q-Learning in RL

(Bellman Optimality Equation) Under an optimal policy π^*, for every state s

$$V^{\pi^*}(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V^{\pi^*}(s') \right] \right\}$$

Question: Can we apply the same technique to $V^{\pi^*}(s)$ as in TD(0)?

Nope! There is a “max” operator!

Solution: Let’s leverage the Q-function $Q(s,a)$ instead!

Other Algorithms That Look Like SGD (II): Q-Learning (2/2.)

Q-Learning in RL

(Bellman Optimality Equation) Under an optimal policy π^*, for every state s

$$Q^{\pi^*}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \Big[\max_{a' \in A} Q^{\pi^*}(s',a') \Big]$$

(Equivalently) $$\mathbb{E}_{s' \sim P(\cdot|s,a)} \Big[Q^{\pi^*}(s,a) - R(s,a) - \gamma \max_{a' \in A} Q^{\pi^*}(s',a') \Big] = 0$$

Q-learning algorithm: learn optimal Q function $Q^{\pi^*}(s,a)$ without knowing P and R

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) - \eta_t \Big(Q(s_t,a_t) - (R(s_t,a_t) + \gamma \max_{a \in A} Q(s_{t+1},a)) \Big)$$

Next, let's study the convergence of SGD!

(To begin with, let's introduce 3 useful lemmas)

1. PL condition under strong convexity
2. First stochastic descent lemma
3. Second stochastic descent lemma

A Useful Lemma of Strong Convexity

Lemma: Suppose $F(x)$ is μ-strongly convex with minimum value F^* and minimizer x^*

Then, $$F(x) - F^* \leq \frac{1}{2\mu} \cdot \|\nabla F(x)\|^2$$

Pf: By strong convexity, we have that for all $\tilde{x} \in X$.

$$F(\tilde{x}) \geq F(x) + \nabla F(x)^T(\tilde{x} - x) + \frac{\mu}{2}\|\tilde{x} - x\|^2 \geq F(x) - \frac{1}{2\mu}\|\nabla F(x)\|^2$$

When is this minimized? At $\tilde{x} = x - \frac{1}{\mu}\nabla F(x)$

Hence, $F^* \geq F(x) - \frac{1}{2\mu}\|\nabla F(x)\|^2$ $\square$

Descent Lemma for Stochastic Updates

Lemma: Let F be an L-smooth function. Then, the iterates of SGD satisfy the following: Given x_k, we have

expected directional derivative along $g(x_k;\varepsilon_k)$

$$\mathbb{E}[F(x_{k+1}) \mid x_k] - F(x_k) \leq -\eta_k \cdot \nabla F(x_k)^T \mathbb{E}_{\varepsilon_k}[g(x_k;\varepsilon_k) \mid x_k] + \frac{1}{2}\eta_k^2 L \cdot \mathbb{E}_{\varepsilon_k}[\| g(x_k;\varepsilon_k)\|^2 \mid x_k]$$

Question: Implication of the above descent lemma ?

Proof of Stochastic Descent Lemma

Step 1:

$$F(X_{k+1}) - F(X_k) \leq \nabla F(X_k)^T (X_{k+1} - X_k) + \frac{1}{2} L \cdot \| X_{k+1} - X_k \|_2^2 \quad \text{(why?)}$$

$$\leq -\eta_k \cdot \nabla F(X_k)^T \cdot g(X_k; \xi_k) + \frac{1}{2} \eta_k^2 \cdot L \cdot \| g(X_k; \xi_k) \|^2$$

↑ why?

Step 2: By taking the "conditional expectation" w.r.t. ξ_k, we have

$$E[F(X_{k+1}) | X_k] - E[F(X_k) | X_k] \leq -\eta_k \cdot \nabla F(X_k)^T \cdot E[g(X_k; \xi_k) | X_k] + \frac{1}{2} \eta_k^2 \cdot L \cdot E[\| g(X_k; \xi_k) \|^2 | X_k]$$

Some Mild Technical Assumptions

- Quick recap:

(Objective) $F^* := \min_{x \in X} F(x)$ (where $F(x) = \mathbb{E}[f(x;\varepsilon)]$)

(SGD) $x_{k+1} = x_k - \eta_k \cdot \underbrace{g(x_k;\varepsilon_k)}_{\text{unbiased estimate}}$

- To prove convergence of SGD, consider the following mild technical assumptions:

(A1) $F(x)$ is bounded below, i.e. $F^* > -\infty$

(A2) $F(x)$ is L-smooth

(A3) $g(x_k;\varepsilon_k)$ has bounded variance, i.e.

$$\mathbb{V}\left[g(x_k;\varepsilon_k)\,|\,x_k\right] \leq M + M_V \|\nabla F(x_k)\|^2$$

2nd Stochastic Descent Lemma

Lemma: Under Assumptions (A_1)-(A_3), the iterates of SGD satisfy

$$\mathbb{E}\left[F(X_{k+1}) \mid X_k \right] - F(X_k) \leq -\left(1 - \frac{1}{2}\eta_k \cdot L \cdot (M_V + 1) \right) \cdot \eta_k \cdot \| \nabla F(X_k) \|^2 + \frac{1}{2}\eta_k^2 \cdot L \cdot M$$

Question: What step sizes "η_k" lead to descent in (expectation)?

Proof of 2nd Stochastic Descent Lemma

By the 1st Stochastic Descent Lemma, we have

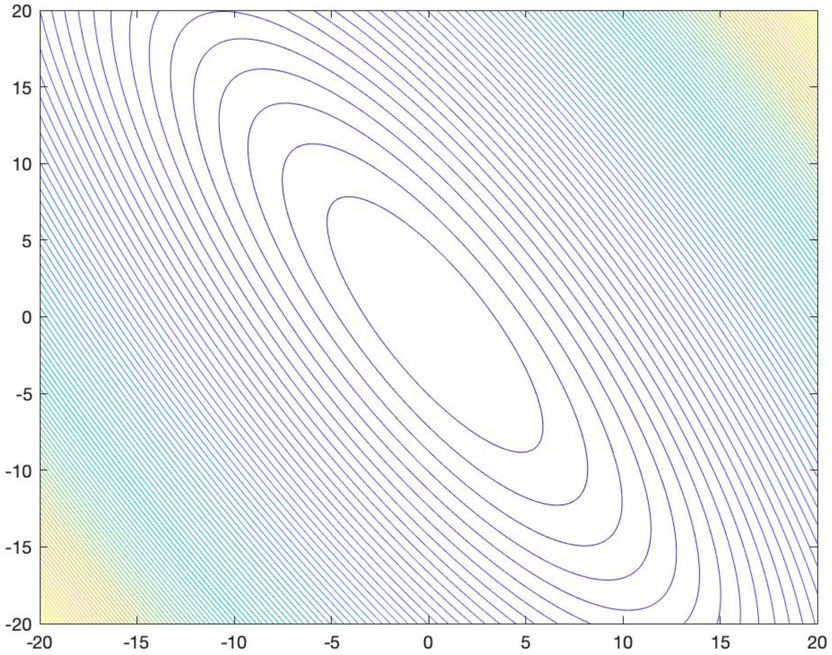
$$\mathbb{E}[F(x_{k+1})|x_k] - F(x_k)$$

$$\leq -\eta_k \cdot \nabla F(x_k)^T \cdot \underbrace{\mathbb{E}[g(x_k;\xi_k)|x_k]}_{=\nabla F(x_k)^T} + \frac{1}{2}\eta_k^2 \cdot L \cdot \underbrace{\mathbb{E}[\|g(x_k;\xi_k)\|^2|x_k]}_{\leq M + (1+M_V)\cdot\|\nabla F(x_k)\|^2}$$

$$= \Big(\qquad\qquad \Big) \cdot \|\nabla F(x_k)\|^2 + \Big(\qquad\qquad \Big)$$

Rethinking the Two Stochastic Descent Lemmas

- Do we really need the "unbiasedness" of $g(x_k; \varepsilon_k)$?



- Why do we need the dependency of $\mathbb{V}\left[g(x_k; \varepsilon_k) \,|\, x_k\right]$ on $\|\nabla F(x_k)\|^2$?
  - Recall that $\mathbb{V}\left[g(x_k; \varepsilon_k) \,|\, x_k\right] \leq M + M_V \|\nabla F(x_k)\|^2$

A Summary of Convergence Results of SGD

	Fixed step size $(\eta_t \equiv \eta)$	Diminishing step sizes $(\eta_t = \Theta(1/t))$
μ-strongly convex functions	$\mathbb{E}[F(\theta_t) - F_*] \leq \frac{\eta L \sigma^2}{2\mu} + (1-\eta\mu)^t \big(F(\theta_0) - F_*\big)$	$\mathbb{E}[F(\theta_t) - F_*] = O\Big(\frac{1}{t}\Big)$
General functions	$\mathbb{E}\Big[\frac{1}{t}\sum_{i=1}^{t} \|\|\nabla F(\theta_i)\|\|^2\Big] \leq \eta L \sigma^2 + \frac{2\big(F(\theta_0) - F_*\big)}{\eta t}$	$\lim_{t\to\infty} \mathbb{E}\Big[\frac{\sum_{i=1}^{t} \eta_i \|\|\nabla F(\theta_i)\|\|^2}{\sum_{i=1}^{t} \eta_i}\Big] = 0$ $\lim_{t\to\infty} \nabla F(\theta_t) = 0$

- For general non-convex functions, SGD converges to a (nearly-)stationary point

Convergence of SGD: Strong Convexity & Fixed Step Sizes

Theorem (SC+F) Suppose the following conditions hold.

(1) Assumptions $(A_1)-(A_3)$

(2) $F(x)$ is L-smooth and μ-strongly convex

(3) Step sizes $\eta_k \equiv \eta$ satisfy that $0 < \eta < \frac{\mu}{L(M_V+1)}$

Then, SGD achieves

$$E[F(x_k)] - F^* \leq \frac{\eta \cdot L \cdot M}{2 \cdot \mu} + (1-\eta\mu)^k \cdot \left(F(x_0) - F^* - \frac{\eta L M}{2\mu}\right)$$

Question: Does SGD achieve convergence to the minimizer under fixed η?

Proof of Theorem (SC+F)

Step1: By the 2nd Stochastic Descent Lemma, we have

$$\mathbb{E}\left[F(x_{k+1})\,|\,x_k\right] - F(x_k)$$

$$\leq -\left(1 - \tfrac{1}{2}\eta \cdot L \cdot (M_v+1)\right) \cdot \eta \cdot \|\nabla F(x_k)\|^2 + \tfrac{1}{2}\eta^2 \cdot L \cdot M$$

$$\leq -\tfrac{1}{2}\eta \cdot \|\nabla F(x_k)\|^2 + \tfrac{1}{2}\eta^2 \cdot L \cdot M$$

$$\leq -\eta \cdot \mu \cdot \left(F(x_k) - F^*\right) + \tfrac{1}{2}\eta^2 \cdot L \cdot M$$

Step2: Subtracting F^* on both sides and taking the "total expectation"

$$\mathbb{E}\left[F(x_{k+1}) - F^*\right] \leq (1-\eta\mu) \cdot \mathbb{E}\left[F(x_k) - F^*\right] + \tfrac{1}{2}\eta^2 \cdot L \cdot M$$

Step3: By subtracting $\frac{\eta L M}{2\mu}$ on both sides, we have

$$E[F(x_{k+1}) - F^*] - \frac{\eta L M}{2\mu} \leq (1 - \eta\mu) \cdot E[F(x_k) - F^*] + \frac{1}{2}\eta^2 L \cdot M - \frac{\eta L \cdot M}{2\mu}$$

$$= (1 - \eta\mu) \cdot \left(E[F(x_k) - F^*] - \frac{\eta L \cdot M}{2\mu} \right)$$

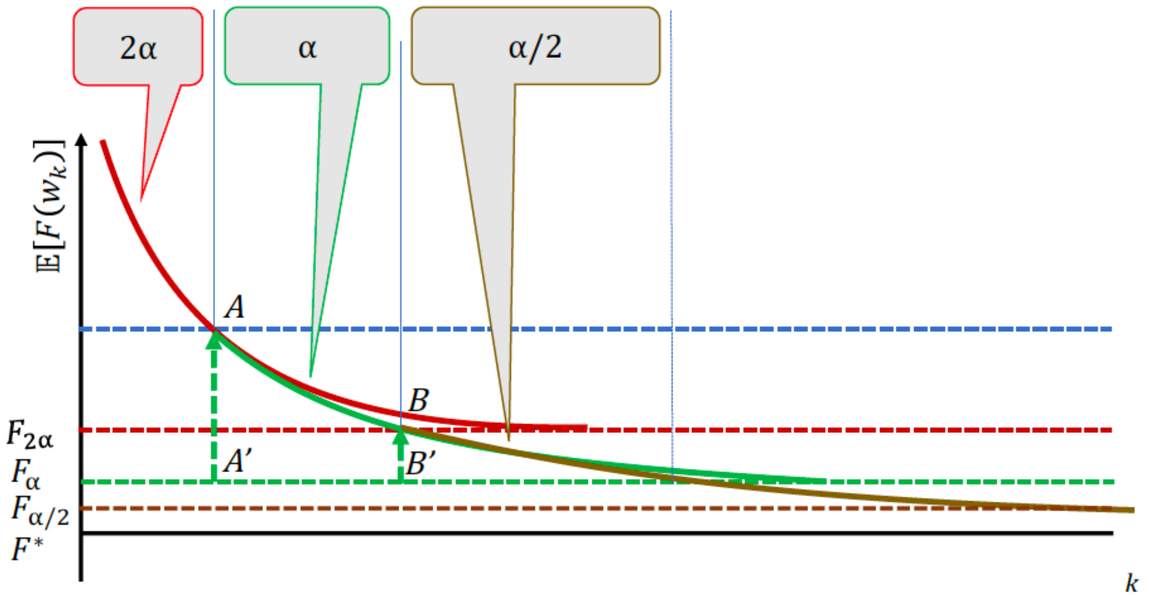
Step4: Hence,

$$E[F(x_k) - F^*] \leq (1 - \eta\mu)^k \cdot \left(F(x_0) - F^* - \frac{\eta L M}{2\mu} \right) + \frac{\eta L M}{2\mu}$$

A Closer Look at Theorem (SC+F)

$$\mathbb{E}[F(x_k) - F_*] \leq \frac{\eta LM}{2\mu} + (1 - \eta\mu)^k \left(F(x_0) - F_*\right)$$

- What if there is no noise in the estimated gradients?
- Why do we need “strong convexity”?
- What’s the role of initial condition?
- A practical strategy of choosing η in SGD:
  - Run SGD with fixed step sizes
  - Whenever the progress stalls, we reduce the step sizes and proceed with SGD



(Figure Source: [Bottou, Curtis, Nocedal, 2018])

Convergence of SGD: Strong Convexity & Diminishing Step Sizes

Theorem (SC+D) Suppose the following conditions hold.

(1) Assumptions (A1)-(A3)

(2) $F(x)$ is L-smooth and μ-strongly convex

(3) Step sizes $\eta_k = \frac{\beta}{k+\gamma}$, for some $\beta > \frac{1}{\mu}$ and $\gamma > 0$ such that $\eta_1 \leq \frac{1}{L(M_v+1)}$

Then, SGD achieves $E[F(x_k) - F^*] \leq \frac{\nu}{k+\gamma}$, where

$$\nu := \max\left\{\frac{\beta^2 LM}{2(\beta\mu - 1)}, (\gamma+1)\cdot(F(x_0) - F^*)\right\}$$

Proof of Theorem (SC+D)

Step 1: By the 2nd stochastic descent lemma,

$$\mathbb{E}[F(x_{k+1}) \mid x_k] - F(x_k) \le -\left(1 - \frac{1}{2}\underbrace{\eta_k L (M_v+1)}_{\le 1}\right)\eta_k \|\nabla F(x_k)\|^2 + \frac{1}{2}\eta_k^2 L M$$

$$\le -\frac{1}{2}\eta_k \|\nabla F(x_k)\|^2 + \frac{1}{2}\eta_k^2 L M$$

Step 2: By taking total expectation and subtracting F^*,

$$\mathbb{E}[F(x_{k+1}) - F^*] \le (1 - \eta_k \mu)\,\mathbb{E}[F(x_k) - F^*] + \frac{1}{2}\eta_k^2 L M \quad -(*)$$

Step 3: We prove $\mathbb{E}[F(x_k) - F^*] \le \frac{\nu}{k+\gamma}$ by induction

(Cont.). Want: $\underbrace{E[F(x_k)-F^*] \le \frac{\nu}{k+\gamma}}_{(**)}$, where $\nu := \max\left\{\frac{\beta^2 LM}{2(\beta\mu-1)}, (\gamma+1)\cdot(F(x_0)-F^*)\right\}$

For $k=1$: By (*),

$$E[F(x_1)-F^*] \le (1-\eta_1\mu)\cdot(F(x_0)-F^*) + \frac{1}{2}\eta_1^2 L\cdot M$$

Suppose (**) holds under k and consider $k+1$:

$$E[F(x_{k+1})-F^*] \le \left(1-\frac{\beta}{\gamma+k}\mu\right)\cdot\frac{\nu}{\gamma+k} + \frac{\beta^2\cdot L\cdot M}{2(\gamma+k)^2}$$

$$= \left(\frac{(\gamma+k)-1}{(\gamma+k)^2}\right)\nu - \underbrace{\left(\frac{\beta\cdot\mu-1}{(\gamma+k)^2}\right)\nu}_{\text{non-positive}} + \frac{\beta^2 L\cdot M}{2(\gamma+k)^2} \le \frac{\nu}{(\gamma+k)+1}$$

since $(\gamma+k)^2 \ge (\gamma+k+1)(\gamma+k-1)$

A Closer Look at Theorem (SC+D)

$$\mathbb{E}[F(x_k) - F_*] \leq \frac{\nu}{k+\gamma} \qquad \nu = \max\left\{\frac{\beta^2 LM}{2(\beta\mu - 1)}, (\gamma+1)(F(x_0) - F(x^*))\right\}$$

- Why do we need "strong convexity"?

- Why do we need $\beta \geq \frac{1}{\mu}$? *For sufficiently fast convergence!*
  - A toy example in [Nemirovski et al., 2009]

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming, SIAM Journal on Optimization, 2009

A Toy Example: Slow Convergence Due to Improper β

- Consider $f(x) = x^2/10$ with $x_0 = 1$
  - What is μ in this example?

- Suppose we take the step sizes $\eta_k = 1/k$ (i.e., $\beta = 1 < \frac{1}{\mu} = 10$)

$$x_{k+1} = x_k - \frac{1}{k} f'(x_k) = (1 - \frac{1}{5k}) x_k$$

$$x_k = \prod_{n=0}^{k-1} (1 - \frac{1}{5n}) x_0 > 0.8 \cdot k^{-\frac{1}{5}}$$

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming, SIAM Journal on Optimization, 2009

Next Question: Is $O(\frac{1}{k})$ optimal?

- (Informal) For the minimization of strongly convex functions, there is no algorithm that can achieve an accuracy better than $O(1/k)$ on performing k queries

- Therefore, SGD with step sizes $\Theta(1/k)$ is optimal

A. Nemirovski, D. Yudin, “Problem complexity and method efficiency in optimization,” Wiley, 1983

A. Agarwal, P. Barlett, P Ravikumar, M. Wainright, “Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization,” IEEE Transactions on Information Theory, 2011

Comparison: SGD and Batch GD

- Consider strongly convex "empirical risk minimization" with n samples
- To achieve an accuracy of ϵ, we need: ($\kappa := L/\mu$ is the condition number)

	Iteration Complexity	Per-iteration Cost	Total Computation Cost
Batch GD	$\kappa \log \frac{1}{\epsilon}$	n	$n\kappa \log \frac{1}{\epsilon}$
SGD	$\kappa^2 \frac{1}{\epsilon}$	1	$\kappa^2 \frac{1}{\epsilon}$

- SGD has an advantage for large n and moderate ϵ ("big data" regime!)
  - That is, $\frac{1}{\epsilon} < n \log \frac{1}{\epsilon}$

SGD for Non-Convex and Smooth Functions

Convergence of SGD for Non-Convex, Smooth functions

Theorem Suppose the following conditions hold.

(1) Assumptions (A_1)-(A_3)

(2) $F(x)$ is L-smooth (but not necessarily convex)

(3). Step sizes $\eta_k \equiv \eta$ satisfy that $0 < \eta < \frac{1}{L \cdot (1 + M_v)}$

Then, SGD achieves $\frac{1}{K} E\left[\sum_{k=0}^{K-1} \|\nabla F(x_k)\|^2\right] \leq \eta \cdot L \cdot M + \frac{2 \cdot (F(x_0) - F^*)}{K \cdot \eta}$

Proof:

Step1: By the 2nd stochastic descent lemma and taking "total expectation",

$$\mathbb{E}[F(x_{k+1})] - \mathbb{E}[F(x_k)]$$

$$\leq -\left(1 - \frac{1}{2}\eta \cdot L \cdot (1 + M_v)\right) \cdot \eta \cdot \mathbb{E}[\|\nabla F(x_k)\|^2] + \frac{1}{2}\eta^2 \cdot L \cdot M$$

$$\leq -\frac{1}{2}\eta \cdot \mathbb{E}[\|\nabla F(x_k)\|^2] + \frac{1}{2}\eta^2 \cdot L \cdot M$$

Step2: By taking the "telescoping sum", we have

$$\mathbb{E}[F(x_{K+1})] - F(x_0) \leq -\frac{1}{2}\eta \cdot \sum_{k=0}^{K-1} \mathbb{E}[\|\nabla F(x_k)\|^2] + \frac{1}{2}K \cdot \eta^2 \cdot L \cdot M$$

Convergence of SGD for Non-Convex, Smooth Functions with Diminishing Step Sizes

Theorem Suppose the following conditions hold.

(1) Assumptions (A1)–(A3)

(2) $F(x)$ is L-smooth

(3) Step sizes satisfy $\sum_{k=1}^{\infty} \eta_k = \infty$ and $\sum_{k=1}^{\infty} \eta_k^2 < \infty$

Then, we have $$\lim_{K \to \infty} E\left[\sum_{k=1}^{K} \eta_k \| \nabla F(x_k) \|^2 \right] < \infty$$

and therefore $$E\left[\frac{1}{\sum_{k=1}^{K} \eta_k} \sum_{k=1}^{K} \eta_k \| \nabla F(x_k) \|^2 \right] \to 0, \text{ as } K \to \infty.$$

Proof: Recall from the 2nd descent lemma

Step 1:

$$\mathbb{E}[F(x_{k+1})] - \mathbb{E}[F(x_k)]$$

$$\leq -\left(1 - \frac{1}{2}\eta_k L \cdot (M_v + 1)\right)\eta_k \cdot \mathbb{E}\left[\|\nabla F(x_k)\|^2\right] + \frac{1}{2}\eta_k^2 L M$$

$$\leq -\frac{1}{2}\eta_k \cdot \mathbb{E}\left[\|\nabla F(x_k)\|^2\right] + \frac{1}{2}\eta_k^2 L M \quad \cdots (\ )$$

Step 2: By summing the above over $k = 1, \cdots, K$,

$$\underbrace{\mathbb{E}[F(x_{K+1})] - \mathbb{E}[F(x_1)]}_{\| \; F^* - \mathbb{E}[F(x_1)]} \leq -\frac{1}{2}\sum_{k=1}^{K} \eta_k \cdot \mathbb{E}\left[\|\nabla F(x_k)\|^2\right] + \frac{1}{2} L M \cdot \sum_{k=1}^{K} \eta_k^2$$

Recall: Convergence Rates of SGD

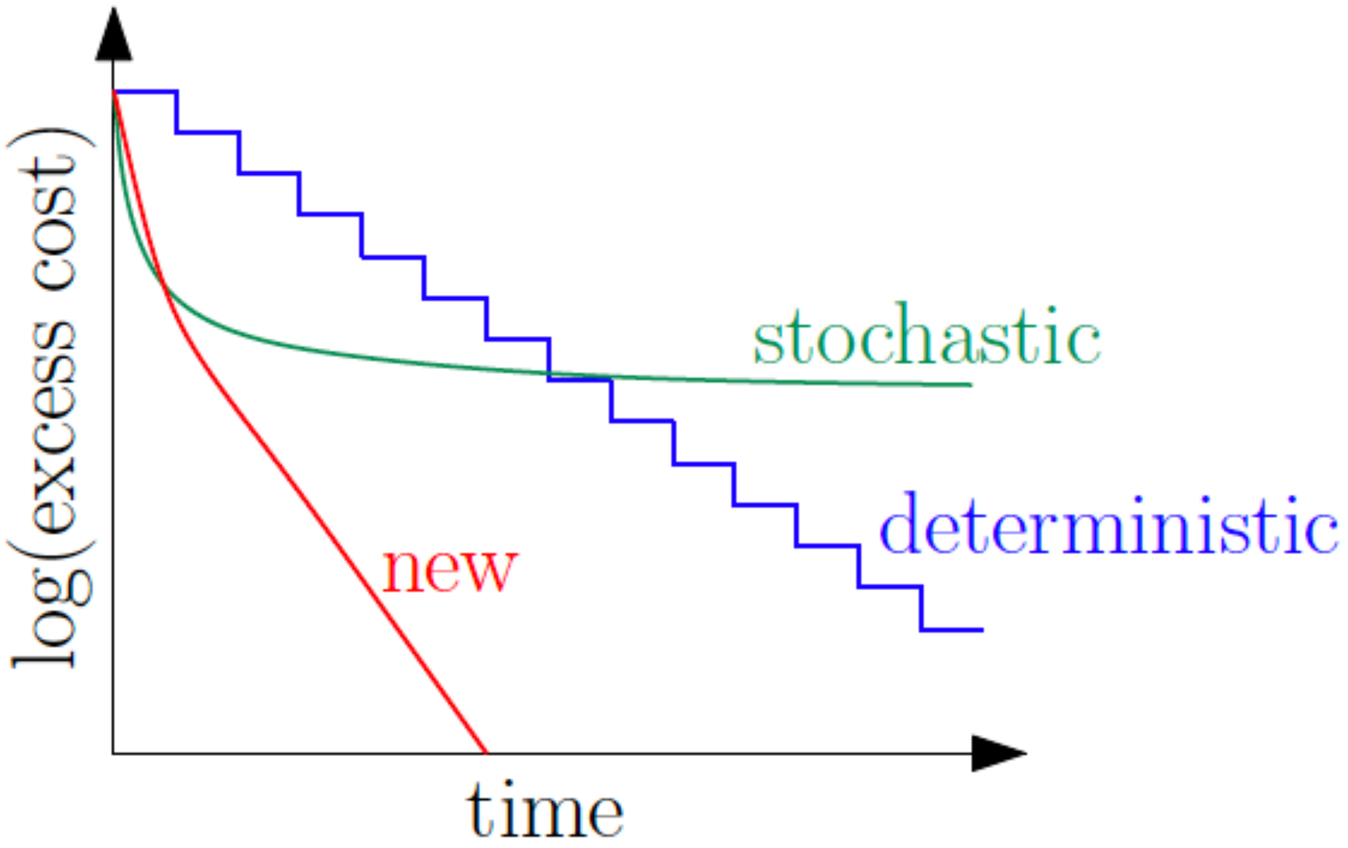
- We know the convergence rate of SGD for smooth strongly-convex functions

	Fixed step size $(\eta_t \equiv \eta)$	Diminishing step sizes $(\eta_t = \Theta(1/t))$
μ-strongly convex functions	$\mathbb{E}[F(\theta_t) - F_*] \leq \frac{\eta L \sigma^2}{2\mu} + (1 - \eta\mu)^t \left(F(\theta_0) - F_*\right)$	$\mathbb{E}[F(\theta_t) - F_*] = O\left(\frac{1}{t}\right)$

- Any issue with the use of a fixed step size? Wandering around a small neighborhood
- Any issue with the use of $\eta_t = \Theta(1/t)$? Slow convergence

Best of Both World?

- Question: Can we have an algorithm that achieves "best of both world”
  - That is, “linear convergence rate” + “low per-iteration cost”



Next topic: Variance reduction

(Figure Credit: Suvrit Sra)

Variance Reduction for SGD

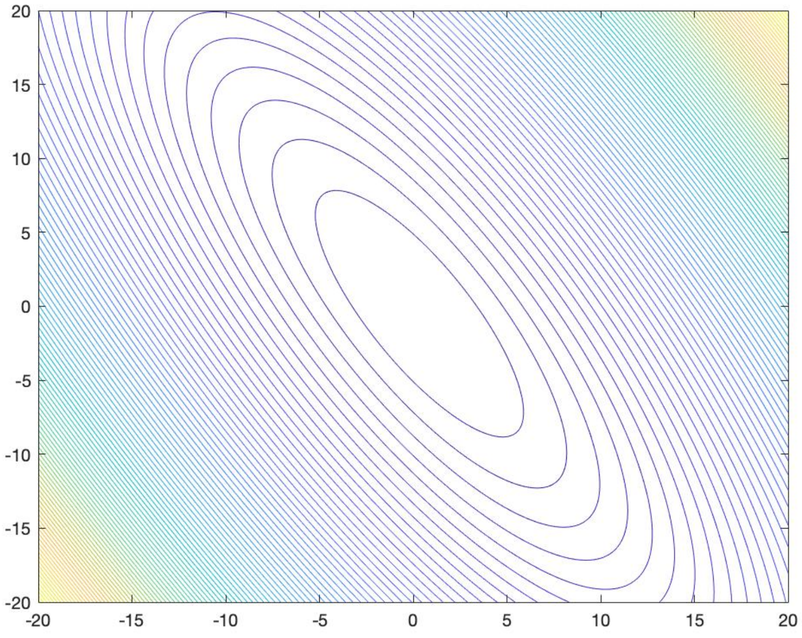
Intuition Behind Variance Reduction

- In vanilla SGD, we use an unbiased estimator $g(x_k, \varepsilon_k)$
  - Question: Can we use $\tilde{g} := g(x_k, \varepsilon_k) + \nu_k$ with $\mathbb{E}[\nu_k] = 0$?
  - Question: What kind of condition do we want about ν_k?
  - Question: What is the ideal ν_k? (By “ideal”, we mean zero variance)

A Intuitive Idea of Variance Reduction: Gradient Aggregation

- Example: SGD for empirical risk minimization

$$\min_{x\in X} F(x) := \frac{1}{n}\sum_{i=1}^{n} f(x; d_i) \qquad (\{d_i\}_{i=1}^{n} \text{ are data samples})$$



Vanilla SGD:

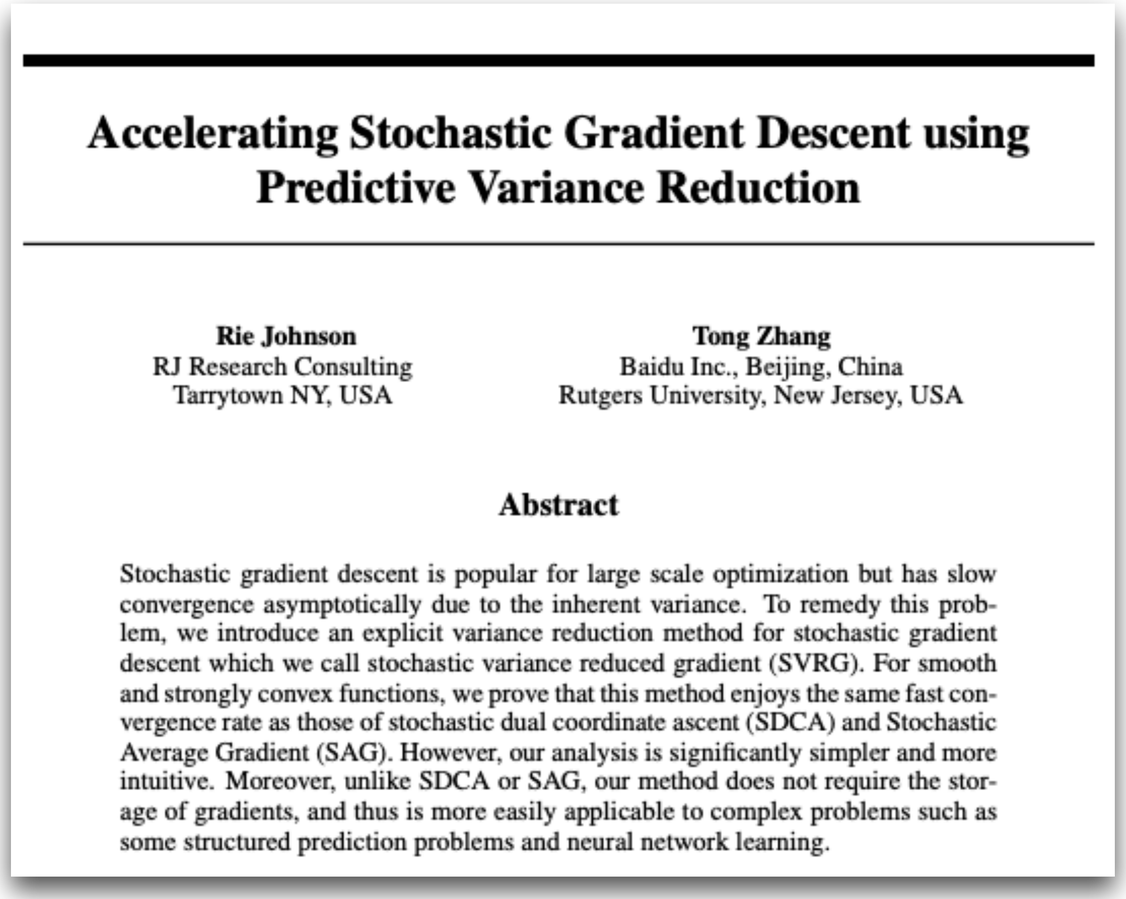
$$x_{k+1} = x_k - \eta \nabla f(x; d_{I_k}), \quad I_k \sim \text{Unif}(1,n)$$

Gradient Aggregation:

$$x_{k+1} = x_k - \eta\left(\nabla f(x; d_{I_k}) - \nu_k\right), \quad I_k \sim \text{Unif}(1,n)$$

where ν_k and $\nabla f(x; d_{I_k})$ are positively correlated (why?)

Stochastic Variance-Reduced Gradient (SVRG)



Accelerating Stochastic Gradient Descent using Predictive Variance Reduction

Rie Johnson
RJ Research Consulting
Tarrytown NY, USA

Tong Zhang
Baidu Inc., Beijing, China
Rutgers University, New Jersey, USA

Abstract

Stochastic gradient descent is popular for large scale optimization but has slow convergence asymptotically due to the inherent variance. To remedy this problem, we introduce an explicit variance reduction method for stochastic gradient descent which we call stochastic variance reduced gradient (SVRG). For smooth and strongly convex functions, we prove that this method enjoys the same fast convergence rate as those of stochastic dual coordinate ascent (SDCA) and Stochastic Average Gradient (SAG). However, our analysis is significantly simpler and more intuitive. Moreover, unlike SDCA or SAG, our method does not require the storage of gradients, and thus is more easily applicable to complex problems such as some structured prediction problems and neural network learning.

- SVRG serves as a simple and intuitive way to achieve gradient aggregation
- This idea has been applied to various other problem settings, including constrained optimization problems and RL

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction, NIPS 2013

Variance-Reduced and Projection-Free Stochastic Optimization

Elad Hazan EHAZAN@CS.PRINCETON.EDU
Princeton University, Princeton, NJ 08540, USA

Haipeng Luo HAIPENGL@CS.PRINCETON.EDU
Princeton University, Princeton, NJ 08540, USA

Abstract

The Frank-Wolfe optimization algorithm has recently regained popularity for machine learning applications due to its projection-free property and its ability to handle structured constraints. However, in the stochastic learning setting, it is still relatively understudied compared to the gradient descent counterpart. In this work, leveraging a recent variance reduction technique, we propose two stochastic Frank-Wolfe variants which substantially improve previous results in terms of the number of stochastic gradient evaluations needed to achieve $1-\epsilon$ accuracy. For example, we improve from $\mathcal{O}(\frac{1}{\epsilon})$ to $\mathcal{O}(\ln\frac{1}{\epsilon})$ if the objective function is smooth and strongly convex, and from $\mathcal{O}(\frac{1}{\epsilon^2})$ to $\mathcal{O}(\frac{1}{\epsilon^{1.5}})$ if the objective function is smooth and Lipschitz. The theoretical improvement is also observed in experiments on real-world datasets for a multiclass classification application.

more (see for example (Hazan & Kale, 2012; Hazan et al., 2012; Jaggi, 2013; Dudik et al., 2012; Zhang et al., 2012; Harchaoui et al., 2015)).

The Frank-Wolfe algorithm (Frank & Wolfe, 1956) (also known as *conditional gradient*) and it variants are natural candidates for solving these problems, due to its projection-free property and its ability to handle structured constraints. However, despite gaining more popularity recently, its applicability and efficiency in the stochastic learning setting, where computing stochastic gradients is much faster than computing exact gradients, is still relatively understudied compared to variants of projected gradient descent methods.

In this work, we thus try to answer the following question: *what running time can a projection-free algorithm achieve in terms of the number of stochastic gradient evaluations and the number of linear optimizations needed to achieve a certain accuracy?* Utilizing Nesterov's acceleration technique (Nesterov, 1983) and the recent variance reduction idea (Johnson & Zhang, 2013; Mahdavi et al., 2013), we

[Hazan and Luo, ICML 2016]

Stochastic Variance-Reduced Policy Gradient

Matteo Papini [*1] **Damiano Binaghi** [*1] **Giuseppe Canonaco** [*1] **Matteo Pirotta** [2] **Marcello Restelli** [1]

Abstract

In this paper, we propose a novel reinforcement-learning algorithm consisting in a stochastic variance-reduced version of policy gradient for solving Markov Decision Processes (MDPs). Stochastic variance-reduced gradient (SVRG) methods have proven to be very successful in supervised learning. However, their adaptation to policy gradient is not straightforward and needs to account for I) a non-concave objective function; II) approximations in the full gradient computation; and III) a non-stationary sampling process. The result is SVRPG, a stochastic variance-reduced policy gradient algorithm that leverages on importance weights to preserve the unbiasedness of the gradient estimate. Under standard assumptions on the MDP, we provide convergence guarantees for SVRPG with a convergence rate that is linear under increasing batch sizes. Finally, we suggest practical variants of SVRPG, and we empirically evaluate them on continuous MDPs.

a value function, or directly a policy defining the agent's behaviour. Furthermore, when the tasks are characterized by large or continuous state-action spaces, RL needs the powerful function approximators (e.g., neural networks) that are the main subject of study of SL. In a typical SL setting, a performance function $J(\boldsymbol{\theta})$ has to be optimized w.r.t. to model parameters $\boldsymbol{\theta}$. The set of data that are available for training is often a subset of all the cases of interest, which may even be infinite, leading to optimization of finite sums that approximate the expected performance over an unknown data distribution. When generalization to the complete dataset is not taken into consideration, we talk about Empirical Risk Minimization (ERM). Even in this case, stochastic optimization is often used for reasons of efficiency. The idea of stochastic gradient (SG) ascent (Nesterov, 2013) is to iteratively focus on a random subset of the available data to obtain an approximate improvement direction. At the level of the single iteration, this can be much less expensive than taking into account all the data. However, the sub-sampling of data is a source of variance that can potentially compromise convergence, so that per-iteration efficiency and convergence rate must be traded off

[Papini et al., ICML 2018]

Pseudo Code of SVRG

Outer loop for snapshots

1: **for** $s = 1, 2, \ldots$ **do**

2: $\boldsymbol{x}_s^{\text{old}} \leftarrow x_s^j$ and compute $\underbrace{\nabla F(\boldsymbol{x}_s^{\text{old}})}_{\text{batch gradient}}$ // update snapshot

($j \sim \text{Unif}(0, \cdots, m-1)$)

3: initialize $\boldsymbol{x}_s^0 \leftarrow \boldsymbol{x}_s^{\text{old}}$

4: **for** $\underbrace{t = 0, \ldots, m-1}_{\text{each epoch contains } m \text{ iterations}}$ **do**

Inner loop for iterate updates

5: choose i_t uniformly from $\{1, \ldots, n\}$, and

$$\boldsymbol{x}_s^{t+1} = \boldsymbol{x}_s^t - \eta\{\underbrace{\nabla f_{i_t}(\boldsymbol{x}_s^t) - \nabla f_{i_t}(\boldsymbol{x}_s^{\text{old}})}_{\text{stochastic gradient}} + \nabla F(\boldsymbol{x}_s^{\text{old}})\}$$

- Question: How many gradient computations are needed under SVRG in one epoch? How about SGD?

(Slide Credit: Yuxin Chen)

Comparison: SGD / Batch GD / SVRG

- Consider strongly convex "empirical risk minimization" with n samples

	Iteration Complexity	Per-iteration Cost	Total Computation Cost
Batch GD	$\kappa \log \frac{1}{\epsilon}$	n	$n\kappa \log \frac{1}{\epsilon}$
SGD	$\kappa^2 \frac{1}{\epsilon}$	1	$\kappa^2 \frac{1}{\epsilon}$
SVRG			$(n+\kappa)\log \frac{1}{\epsilon}$

($\kappa := L/\mu$ is the condition number)