MIT 6.7220/15.084 — Nonlinear Optimization (Spring '25) Tue, Apr 15th 2025

Lecture 15
Stochastic gradient descent

Instructor: Prof. Gabriele Farina (✉ gfarina@mit.edu)*

As we have seen in the past few lectures, gradient descent and its family of algorithms (including *accelerated* gradient descent, *projected* gradient descent and *mirror* descent) are first-order methods that can compute approximate minima of differentiable functions. The cost of each iteration of these algorithms is dominated (linearly) by the time it takes to evaluate the gradient of the objective function at each point generated by the algorithm.

L15.1 Preliminary words on stochastic methods

In some cases, including in most machine learning applications, the objective function contains a huge number of terms (usually, a summation over training data points), rendering the gradient computation prohibitively expensive. Today, we will investigate an approach to sidestep this difficulty in practice. The idea is simple, and it is a general principle of algorithm design broadly:

> *If exact computation is expensive, replace it with a cheaper estimate.*

In particular, as is often the case, we will investigate replacing the computation of an exact gradient with the computation of a *cheap, unbiased random estimator* of it. This approach, based on randomness (aka. *stochasticity*) leads to the family of algorithms that go under the name of *stochastic gradient descent.*

As one might expect, such a stochastic approach is *not* free. We will be trading an *exact* quantity (the exact gradient of the objective function) for a *cheap* estimate that is subject to *variance*, subjecting the resulting algorithm to random (erratic) steps rather than clean descent steps. This tradeoff is nonetheless extremely favorable for multiple reasons:

- As a matter of pragmatism, stochastic gradient descent algorithms can perform a significantly larger number of steps in the same time that it takes (exact) gradient descent to perform even a single step. In certain cases, the exact algorithm might not be able to perform a *single* step in the allotted computation budget. By this metric, the choice between an exact and a stochastic method reduces to the choice between

*These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

an algorithm that is not able to start, and one that—as erratic as it may be—at least makes *some* progress.

- As a subtler comment, the use of stochasticity, as opposed to using a powerful exact method, has been observed empirically to lead to better solutions in machine learning settings, even when the exact algorithm can be run fast enough. In machine learning, minimizing the objective function on the training data *correlates*, but is *conceptually different from*, finding a good model for the task at hand. In other words, the role of optimization in machine learning is to (i) produce models that interpolate well the training dataset, but also (ii) do *not overfit*, that is, generalize well to unseen but in-distribution ("similar") examples. Using stochastic gradient descent has been linked with a reduction in overfitting and increased success on this second goal, partly due to the presence of noise, which enables the algorithm to escape local minima and saddle points.

L15.2 Empirical risk minimization (ERM) problems in machine learning

Many problems in machine learning can be abstracted as follows:

- We are given a dataset of k examples $(z_1, y_1), ..., (z_k, y_k)$, where $z_i \in \mathbb{R}^n$ is a feature vector and $y_i \in \mathbb{R}$ is a label. For example:
  - we might be interested in building a *classifier* for handwritten digits, where each z_i is a vector containing 28×28 grayscale pixel intensities, and each y_i is the digit itself (a multinomial classification problem); or
  - we might be interested in *predicting* the price of a house, where each z_i is a vector containing features of the house (e.g., number of bedrooms, square footage, etc.), and each y_i is the price of the house (a regression problem); or
  - we might be interested in predicting the next word in a sentence, where each z_i is a vector containing the previous words in the sentence, and each y_i is the next word (a multinomial classification problem); or
  - we might be interested in predicting the probability of a user clicking on an ad, where each z_i is a vector containing features of the user and the ad, and each y_i is the probability of clicking (a binary—that is, two-class—classification problem); *et cetera*.
- We have selected a *model class* of functions $g_\theta : \mathbb{R}^n \to \mathbb{R}^m$ that we believe can capture the relationship between the features and the labels. These functions are *parameterized* by a vector $\theta \in \mathbb{R}^d$. For example, in the case of neural networks, θ would be the set of layer weights and biases.
  - The output dimension m of the model is typically $m = 1$ in the case of regression and binary classification problems (indicating the probability of the, say, positive class).
  - In the case of multinomial classification, m is typically the number of classes and indicates the (unnormalized) log-probability the model assigns to the event that z_i belongs to each possible class.

- We are interested in finding a choice of parameters θ that "fits the data well". This is typically done by minimizing the *empirical risk* (or *empirical loss*) of the model, defined as

  $$J_{\text{emp}}(\theta) \coloneqq \frac{1}{k} \sum_{i=1}^{k} \ell(g_\theta(z_i), y_i),$$

  where $\ell : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is a loss function that measures the discrepancy between the model's prediction $g_\theta(z_i)$ and the true label y_i. For example, in regression problems, a reasonable choice might be the squared loss $\ell(a, b) = (a - b)^2$. For classification, in which the model output $g_\theta(z_i) = \left(g_{1,\theta}(z_i), ..., g_{m,\theta}(z_i)\right) \in \mathbb{R}^m$ predicts the unnormalized log-probability of each class and $y_i \in \{1, ..., m\}$, the *log loss* (also known as *cross entropy loss* or *softmax loss*) is often used:

  $$\ell(g_\theta(z_i), y_i) = -\sum_{j=1}^{m} 1[y_i = j] \cdot \log\left(\frac{\exp\{g_{j,\theta}(z_i)\}}{\sum_{j'=1}^{m} \exp\{g_{j',\theta}(z_i)\}}\right).$$

In many cases, an explicit regularization term is added on top of J_{emp} to reduce overfitting.

L15.3 Stochastic gradient descent

The empirical regret minimization objective $f = J_{\text{emp}}$ defined above is a sum of k terms, one for each example in the dataset. When the dataset is large, evaluating the gradient of J_{emp} at each iteration of gradient descent can be computationally expensive. In this case, we can replace at each iteration the exact gradient ∇f with a cheap, unbiased estimator $\tilde{\nabla} f$ of it. This means that, denoting with $\mathbb{E}_t$ the expectation conditioned on all past random choices (that is, all randomization used at times $1, ..., t - 1$), the estimator $\tilde{\nabla} f(x_t)$ satisfies $\mathbb{E}_t\left[\tilde{\nabla} f(x_t)\right] = \nabla f(x_t)$. This leads to the *stochastic gradient descent (SGD)* algorithm,

$$\boxed{x_{t+1} \coloneqq x_t - \eta \tilde{\nabla} f(x_t)} \qquad \text{where} \quad \mathbb{E}_t\left[\tilde{\nabla} f(x_t)\right] = \nabla f(x_t). \tag{1}$$

L15.3.1 Typical unbiased estimator: mini-batches

Different instantiations of SGD differ in the choice of the estimator $\tilde{\nabla} f(x_t)$. For empirical risk minimization problems, the most common choice is to replace the gradient of $f = J_{\text{emp}}$ with the average gradient of a *uniformly randomly* subset of training data (typically sampled without replacement), called a *mini-batch*. The resulting instantiation of SGD is therefore known as *mini-batch* SGD.

Computing the random mini-batch is usually easy to do by using a random shuffle of the dataset, and selecting the first b examples as the mini-batch. The size of the mini-batch b is a hyperparameter of the algorithm. A popular choice is $b = 32$ or $b = 64$.

L15.4 Analysis of stochastic gradient descent

As we have already seen several times, the analysis of variants of the gradient descent algorithm passes through generalizing the two key descent lemmas: the gradient descent lemma and the (Euclidean) mirror descent lemma. In this case, we will need to modify the lemmas to account for the fact that the gradient of the objective has been replaced with an unbiased estimator in (1).

L15.4.1 Stochastic gradient descent lemma

Recall that an L-smooth function $f : \mathbb{R}^n \to \mathbb{R}$ satisfies the quadratic upper bound

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x - y\|_2^2 \qquad \forall x, y \in \mathbb{R}^n.$$

Exactly as we did in Lecture 12, we apply the previous bound for the specific choice $y = x_{t+1}, x = x_t$, and obtain

$$f(x_{t+1}) \le f(x_t) - \eta\big\langle \nabla f(x_t), \tilde{\nabla} f(x_t)\big\rangle + \frac{L}{2}\eta^2\big\|\tilde{\nabla} f(x_t)\big\|_2^2$$

Taking the (conditional) expectation on both sides and using the unbiasedness $\mathbb{E}_t\big[\tilde{\nabla} f(x_t)\big] = \nabla f(x_t)$ we therefore obtain the following stochastic generalization of the gradient descent lemma.

Theorem L15.1 (Stochastic gradient descent lemma)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and $\eta > 0$ be an arbitrary stepsize. Two consecutive iterates (x_t, x_{t+1}) produced by the stochastic gradient descent algorithm (1) satisfy

$$\mathbb{E}_t[f(x_{t+1})] \le f(x_t) - \eta\|\nabla f(x_t)\|_2^2 + \frac{L}{2}\eta^2\ \mathbb{E}_t\Big[\big\|\tilde{\nabla} f(x_t)\big\|_2^2\Big].$$

The quantity term $\mathbb{E}_t\Big[\big\|\tilde{\nabla} f(x_t)\big\|_2^2\Big]$, which controls the quadratic term in the previous bound, is often called the *variance* of the stochastic gradient $\tilde{\nabla} f$.

L15.4.2 Convergence in gradient norm

Just like the gradient descent lemma for exact gradient descent, the *stochastic* gradient descent lemma guarantees descent in function value, in expectation, when $\eta > 0$ is sufficiently small. In the stochastic case, perhaps unsurprisingly, this threshold value of η depends inversely proportionally on the variance of the unbiased estimator. Specifically, suppose that $\mathbb{E}_t\Big[\big\|\tilde{\nabla} f(x_t)\big\|_2^2\Big] \le G$ at all times t. Then, the gradient descent lemma can be written as

$$\|\nabla f(x_t)\|_2^2 \le \frac{1}{\eta}\mathbb{E}_t[f(x_t) - f(x_{t+1})] + \frac{L}{2}\eta G.$$

Summing over all times $t = 0, 1, ..., T - 1$, we therefore obtain

$$\sum_{t=0}^{T-1}\|\nabla f(x_t)\|_2^2 \le \frac{1}{\eta}\left(\sum_{t=0}^{T-1}\mathbb{E}_t[f(x_t) - f(x_{t+1})]\right) + \frac{L}{2}\eta G T.$$

Taking expectations on both sides, the conditional expectations $\mathbb{E}_t$ decay into regular expectations $\mathbb{E}$, and we can therefore exploit the telescoping nature of the sum $f(x_t) - f(x_{t+1})$, obtaining

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x_t)\|_2^2\right] \le \frac{1}{\eta}(f(x_0) - \mathbb{E}[f(x_T)]) + \frac{L}{2}\eta G T.$$

Assuming the function is lower bounded by the value $f_\star$, we therefore have

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(x_t)\|_2^2\right] \le \frac{f(x_0) - f_\star}{\eta T} + \frac{L}{2}\eta G.$$

Picking $\eta \approx \frac{1}{\sqrt{T}}$ then reveals that within T iterations, at least one $\mathbb{E}\left[\|\nabla f(x_t)\|_2^2\right] \approx \frac{1}{\sqrt{T}}$, and we recover convergence in gradient norm no matter the (bounded) variance of the estimator.

L15.4.3 Stochastic Euclidean mirror descent lemma

When the objective function is further assumed to be *convex*, then we have seen in Lecture 12 that the Euclidean mirror descent lemma applies, and guarantees descent in *distance* to the set of minimizers.

To generalize the Euclidean mirror descent lemma, we operate as in Lecture 12, starting from the linear lower bound property

$$f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle \qquad \forall x, y \in \mathbb{R}^n,$$

that holds for any convex function. Applying the bound for the specific choice $x = x_t$, and rearranging terms, we find

$$\begin{aligned} f(x_t) &\le f(y) - \langle \nabla f(x_t), y - x_t\rangle && \forall y \in \mathbb{R}^n \\ &= f(y) - \left\langle \mathbb{E}_t\left[\tilde{\nabla} f(x_t)\right], y - x_t\right\rangle && \forall y \in \mathbb{R}^n \\ &= f(y) - \mathbb{E}_t\left[\frac{1}{\eta}\langle x_t - x_{t+1}, y - x_t\rangle\right] && \forall y \in \mathbb{R}^n, \end{aligned}$$

where the first equality follows from the unbiasedness of the gradient estimator $\tilde{\nabla} f$, and the second from the definition of the SGD algorithm (1).

Using the three-point equality we discussed in Lecture 12,

$$-\frac{1}{\eta}\langle x_t - x_{t+1}, y - x_t\rangle = \frac{1}{2\eta}\left(\|x_t - y\|_2^2 - \|x_{t+1} - y\|_2^2 + \|x_t - x_{t+1}\|_2^2\right),$$

we obtain the stochastic version of the Euclidean mirror descent lemma.

Theorem L15.2 (Stochastic Euclidean mirror descent lemma)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex. Then, for any stepsize $\eta > 0$ and $y \in \mathbb{R}^n$, two consecutive iterates (x_t, x_{t+1}) produced by the SGD algorithm (1) satisfy

$$f(x_t) \le f(y) + \frac{1}{2\eta}\mathbb{E}_t\left[\|x_t - y\|_2^2 - \|x_{t+1} - y\|_2^2 + \|x_t - x_{t+1}\|_2^2\right].$$

In particular, by setting $y = x_\star$, where $x_\star$ is a minimizer of f, we can extract an *expected* decrease in distance $\|x_t - x_\star\|_2^2$ provided η is chosen small enough, which is again totally analogous to the exact case seen in Lecture 12.

L15.4.4 Convergence in convex function value

With the stochastic generalizations of the Euclidean mirror descent lemmas, we can obtain a convergence rate (in terms of function value) by using a telescopic argument, similar to what we did in Lecture 12. However, we will not be able to recover the $\frac{1}{t}$ rate of convergence for SGD.

Let $x_\star$ be a minimizer of the convex function function f, and assume $\mathbb{E}_t\left[\|\tilde{\nabla} f(x)\|_2^2\right] \leq G$ at all x. By summing the bound given in the stochastic Euclidean mirror descent lemma (Theorem L15.2) for all $t = 0, ..., T-1$, we obtain

$$\begin{aligned}\sum_{t=0}^{T-1} f(x_t) &\leq T f(x_\star) + \frac{1}{2\eta}\sum_{t=0}^{T-1}\mathbb{E}_t\left[\|x_t - x_\star\|_2^2 - \|x_{t+1} - x_\star\|_2^2 + \|x_t - x_{t+1}\|_2^2\right] \\ &= T f(x_\star) + \frac{1}{2\eta}\sum_{t=0}^{T-1}\mathbb{E}_t\left[\|x_t - x_\star\|_2^2 - \|x_{t+1} - x_\star\|_2^2 + \eta^2\|\tilde{\nabla} f(x_t)\|_2^2\right] \qquad \text{(from (1))} \\ &\leq T f(x_\star) + \frac{1}{2\eta}\sum_{t=0}^{T-1}\mathbb{E}_t\left[\|x_t - x_\star\|_2^2 - \|x_{t+1} - x_\star\|_2^2 + \eta^2 G\right].\end{aligned}$$

Taking expectations on both sides, the conditional expectations $\mathbb{E}_t$ decay into regular expectations $\mathbb{E}$, and we can therefore exploit the telescoping nature of the sum $\|x_t - x_\star\|_2^2 - \|x_{t+1} - x_\star\|_2^2$:

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}[f(x_t)] &\leq T f(x_\star) + \frac{1}{2\eta}\mathbb{E}\left[\sum_{t=0}^{T-1}\left(\|x_t - x_\star\|_2^2 - \|x_{t+1} - x_\star\|_2^2 + \eta^2 G\right)\right] \\ &\leq T f(x_\star) + \frac{1}{2\eta}\|x_0 - x_\star\|_2^2 + \frac{\eta}{2} G T.\end{aligned}$$

Moving the term $T f(x_\star)$ to the left-hand side and dividing by T, we find

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[f(x_t) - f(x_\star)] \leq \frac{1}{2\eta T}\|x_0 - x_\star\|_2^2 + \frac{\eta}{2} G.$$

Hence, we have the following:

Theorem L15.3. Consider the SGD algorithm (1) run on a convex function. By picking $\eta = \frac{1}{\sqrt{GT}}$, this immediately implies that at least one of the iterates x_t, $t \in \{0, ..., T-1\}$, satisfies

$$\mathbb{E}[f(x_t) - f(x_\star)] \leq \frac{\sqrt{G}}{2}\left(1 + \|x_0 - x_\star\|_2^2\right)\frac{1}{\sqrt{T}}.$$

By convexity, the same bound holds also for $\mathbb{E}\left[f(\overline{x}^T) - f(x_\star)\right]$, where $\overline{x}^T$ is the average of the iterates $x_0, ..., x_{T-1}$.

Remark L15.1. It is important to realize that the above bound is of order $\frac{1}{\sqrt{T}}$ rather than $\frac{1}{T}$ as in the (non-stochastic) gradient descent case. However, in establishing this result, we have not used the gradient descent lemma. So, the above result does not require L -smoothness. As a consequence, the previous result shows that non-stochastic gradient descent, applied to convex functions that are not L-smooth, guarantees $\frac{1}{\sqrt{T}}$ convergence in function value provided that the stepsize is chosen well.

L15.5 Further readings

Several variations of the SGD algorithm and its analysis are known. For one, it is known that when the objective function is *strongly* convex, the SGD algorithm can be shown to converge at a rate of $1/T$ in function value, provided the stepsize is chosen well.

Variance reduction techniques, which aim to reduce the variance of the gradient estimator, have been proposed to accelerate the convergence of SGD. These techniques include the SVRG (stochastic variance reduced gradient) algorithm [JZ13], SAG (stochastic average gradient) [SLB17], and SAGA [DBL14].

[JZ13] Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Neural Information Processing Systems (Neurips).*

[SLB17] Schmidt, M., Le Roux, N., & Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, *162*, 83–112.

[DBL14] Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Neural Information Processing Systems (Neurips).*

Changelog

- Apr 17, 2025: Clarified L-smoothness.