Lecture Notes 2: Gradient Descent for Convex Losses

Instructor: Ashok Cutkosky

1 Deterministic Convex Losses

The most commonly used algorithm in machine learning today is (stochastic) gradient descent. Let's consider this algorithm first in the *deterministic* setting to gain some idea for how it works.

Algorithm 1 Gradient Descent

Input: Initial Point $\mathbf{w}_1$, learning rate η, time horizon T:
for $t = 1 \dots T$ **do**
  Set $\mathbf{g}_t = \nabla \mathcal{L}(\mathbf{w}_t)$.
  Set $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}_t$.
end for

Recall the definition of a convex function:

Definition 1. *A function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is* convex *if for all x, there exists some vector g_x such that for all y:*

$$\mathcal{L}(y) \geq \mathcal{L}(x) + \langle g_x, y - x \rangle$$

When $\mathcal{L}$ is differentiable, $g_x = \nabla \mathcal{L}(x)$.

The definition of convexity has an interesting consequence: if $\mathbf{w}_\star = \operatorname{argmin} \mathcal{L}(\mathbf{w})$, then we have:

$$\mathcal{L}(\mathbf{w}) - \mathcal{L}(\mathbf{w}_\star) \leq \langle \nabla \mathcal{L}(\mathbf{w}), \mathbf{w} - \mathbf{w}_\star \rangle$$

So, if we can understand the *linear* functions $\langle \nabla \mathcal{L}(\mathbf{w}), \mathbf{w} - \mathbf{w}_\star \rangle$, then we will be able to bound the non-linear suboptimality gap $\mathcal{L}(\mathbf{w}) - \mathcal{L}(\mathbf{w}_\star)$. In order to do this, we'll need to assume that our loss is *Lipschitz*:

Definition 2. *A function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is G-Lipschitz if for all $x, y \in \mathbb{R}^d$,*

$$|\mathcal{L}(x) - \mathcal{L}(y)| \leq G \|x - y\|$$

This definition may be a little strange looking, but it's actually fairly likely to hold. The intuition here is that the G in G-Lipschitzness is actually measuring the degree to which $\mathcal{L}$ is continuous. In fact, if W is a compact set, then any continuous $\mathcal{L}$ must be G-Lipschitz for some G. When $\mathcal{L}$ is differentiable, we can phrase G-Lipschitzness in the following way:

Proposition 3. *If $\mathcal{L}$ is differentiable, then $\mathcal{L}$ is G-Lipschitz if and only if $\|\nabla \mathcal{L}(x)\| \leq G$ for all x.*

Proof. First, suppose $\|\nabla \mathcal{L}(x)\| \geq G$ for some $x \in \mathbb{R}^d$. By definition of gradient for any vector v, we have:

$$\lim_{\delta \to 0} \frac{\mathcal{L}(x + \delta v) - \mathcal{L}(x) - \delta \langle v, \nabla \mathcal{L}(x) \rangle}{\delta} = 0$$

Let $v = \frac{\nabla \mathcal{L}(x)}{\|\nabla \mathcal{L}(x)\|}$. Then this implies:

$$\lim_{\delta \to 0} \frac{\mathcal{L}(x + \delta v) - \mathcal{L}(x)}{\delta} = \|\nabla \mathcal{L}(x)\|$$

Thus, if $\|\nabla \mathcal{L}(x)\| \geq G$, there must be some δ such that

$$|\mathcal{L}(x + \delta v) - \mathcal{L}(x)| \geq G\delta$$

and so $\mathcal{L}$ is no G-Lipschitz. Therefore G-Lipschitzness implies $\|\nabla \mathcal{L}(x)\| \leq G$. Now suppose $\|\nabla \mathcal{L}(x)\| \leq G$ for all x. Then by the fundamental theorem of calculus,

$$\begin{aligned}
\mathcal{L}(x) - \mathcal{L}(y) &= \int_0^1 \frac{d}{dt} \mathcal{L}(y + t(x-y))\, dt \\
&= \int_0^1 \langle \nabla \mathcal{L}(y + t(x-y)), x - y \rangle\, dt \\
&\leq \int_0^1 \|\nabla \mathcal{L}(y + t(x-y))\| \|x - y\|\, dt \\
&\leq G\|x - y\|
\end{aligned}$$

so that $\mathcal{L}$ is G-Lipschitz. □

Here is the standard convergence result for gradient descent:

Theorem 4. *Suppose $\mathcal{L}$ is G-Lipschitz and convex. Suppose $\|\mathbf{w}_\star - \mathbf{w}_1\| \leq D$. Set $\eta = \frac{D}{G\sqrt{T}}$. Then*

$$\sum_{t=1}^T \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star) \leq GD\sqrt{T}$$

In particular, if $\hat{\mathbf{w}}$ is selected uniformly at random from $\mathbf{w}_1, \ldots, \mathbf{w}_T$,

$$\mathbb{E}[\mathcal{L}(\hat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_t)] \leq \frac{GD}{\sqrt{T}}$$

Proof. The proof is quite short, although the technique may seem a bit magical:

$$\begin{aligned}
\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2 &= \|\mathbf{w}_t - \eta \nabla \mathcal{L}(\mathbf{w}_t) - \mathbf{w}_\star\|^2 \\
&= \|\mathbf{w}_t - \mathbf{w}_\star\|^2 - 2\eta \langle \nabla \mathcal{L}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_\star \rangle + \eta^2 \|\nabla \mathcal{L}(\mathbf{w}_t)\|^2
\end{aligned}$$

rearranging terms:

$$\langle \nabla \mathcal{L}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_\star \rangle \leq \frac{\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2}{2\eta} + \frac{\eta \|\nabla \mathcal{L}(\mathbf{w}_t)\|^2}{2}$$

using convexity:

$$\mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star) \leq \frac{\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2}{2\eta} + \frac{\eta \|\nabla \mathcal{L}(\mathbf{w}_t)\|^2}{2}$$

Use G-Lipschitzness:

$$\frac{\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2}{2\eta} + \frac{\eta G^2}{2}$$

now, sum over all t:

$$\sum_{t=1}^T \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star) \leq \sum_{t=1}^T \frac{\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2}{2\eta} + \frac{\eta G^2}{2}$$

The sum telescopes:

$$= \frac{\|\mathbf{w}_1 - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{T+1} - \mathbf{w}_\star\|^2}{2\eta} + \frac{\eta T G^2}{2}$$
$$\leq \frac{D^2}{2\eta} + \frac{\eta T G^2}{2}$$

Now, use our setting for η:

$$DG\sqrt{T}$$

Now, if $\hat{\mathbf{w}}$ is chosen uniformly at random from $\mathbf{w}_1, \ldots, \mathbf{w}_T$, then by definition:

$$\mathbb{E}[\mathcal{L}(\hat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_\star)] = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star) \leq \frac{GD}{\sqrt{T}}$$

□

This result has a few lessons for us:

1. This is an *upper bound*. This means that the actual performance of the algorithm might be much better than we have shown here. In practice, this is often true.
2. We only make a statement about a randomly chosen iterate, rather than the last iterate. This is very common in analysis of optimization methods (at least in the stochastic case). In practice however, it is common to simply take $\hat{\mathbf{w}} = \mathbf{w}_T$.
3. Most of the analysis is done without using the value of η - instead it is only substituted at the end. You can see from the analysis that we chose $\eta = \frac{D}{G\sqrt{T}}$ specifically to minimize the final expression. Other choices would still converge.
4. The learning rate is $O(1/\sqrt{T})$. This will show up in the non-convex case as well.

2 Stochastic Convex Losses

Now that we've seen how to analyze gradient descent for deterministic convex losses, let's consider the more practical *stochastic* case. Remember in the stochastic setting we assume:

$$\mathcal{L}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim P_z}[\ell(\mathbf{w}, z)]$$

Let's assume that $\ell(\mathbf{w}, z)$ is differentiable as a function of $\mathbf{w}$ for all z. Then, we have the following:

Proposition 5. *If $\mathcal{L}(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$ is given by $\mathcal{L}(\mathbf{w}) = \mathbb{E}_{z\sim P_z}[\ell(\mathbf{w}, z)]$ and $\ell(\mathbf{w}, z)$ is differentiable as a function of $\mathbf{w}$ for all z, then:*

$$\nabla\mathcal{L}(\mathbf{w}) = \mathop{\mathbb{E}}_{z}[\nabla\ell(\mathbf{w}, z)]$$

Proof. Define $p(z)$ to be the probability density function of z. Then we have:

$$\nabla\mathcal{L}(\mathbf{w}) = \nabla\,\mathbb{E}[\ell(\mathbf{w}, z)]$$
$$= \nabla\int \ell(\mathbf{w}, z)p(z)dz$$

interchanging integration and differentation:

$$= \int \nabla\ell(\mathbf{w}, z)p(z)dz$$
$$= \mathbb{E}[\nabla\ell(\mathbf{w}, z)]$$

For the very technical minded reader, this proof has a couple holes: we assumed z has a density $p(z)$, and we assumed that it is indeed possible to exchange integration and differentiation. Both of these are not fatal issues: instead of $p(z)dz$, we may integrate with respect to the probability measure $d\mu(z)$ if there is not density. For the interchange of integration and differentiation, we can overcome this by adding the assumption that for any $\mathbf{w}$ $\nabla\ell(\mathbf{w}, z)$ is bounded on some closed ball centered at $\mathbf{w}$. This assumption is true for all functions you are likely to ever encounter in practice. □

This Lemma gives us a way to provide an *unbiased estimate* of the gradient $\nabla\mathcal{L}(\mathbf{w})$ at any point $\mathbf{w}$: simply sample some $z \sim P_z$ and return $\nabla\ell(\mathbf{w}, z)$. An *unbaised estimator* for a quantity X is a random variable Y such that $\mathbb{E}[Y] = X$. Unbiased estimators are useful because if you can generate many independent unbiased estimates $Y_1, \dots, Y_N$, then their average $\frac{1}{N}\sum_{i=1}^{N} Y_i$ will converge to X as N grows.

So, our stochastic gradient descent algorithm will be essentially the same as regular gradient descent, but we will substitute an unbiased gradient estimate instead of the true gradient (Algorithm 3).

Algorithm 2 Stochastic Gradient Descent

Input: Initial Point $\mathbf{w}_1$, learning rate η, time horizon T:
for $t = 1 \dots T$ **do**
  Sample $z_t \sim P_z$
  Set $\mathbf{g}_t = \nabla\ell(\mathbf{w}_t, z_t)$.
  Set $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{g}_t$.
end for

Let's analyze the convergence of this algorithm under the assumption that $\mathcal{L}$ is convex:

Theorem 6. *Suppose that $\mathcal{L}$ is convex and that $\ell(\mathbf{w}, z)$ is G-Lipschitz. Suppose $\|\mathbf{w}_1 - \mathbf{w}_\star\| \le D$. Then with $\eta = \frac{D}{G\sqrt{T}}$,*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star)\right] \le DG\sqrt{T}$$

If $\hat{\mathbf{w}}$ is selected uniformly at random from $\mathbf{w}_1, \dots, \mathbf{w}_T$,

$$\mathbb{E}[\mathcal{L}(\hat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_\star)] \le \frac{DG}{\sqrt{T}}$$

Proof. The proof is actually very similar to the proof for the deterministic case, but we put expectations around many quantities and use the unbiasedness property $\mathbb{E}[\nabla\ell(\mathbf{w}_t, z)|\mathbf{w}_t] = \nabla\mathcal{L}(\mathbf{w}_t)$. All expectations presented here are over the randomness of the algorithms (i.e. over the choices $z_1, \dots, z_T$) unless otherwise specified:

$$\begin{aligned}\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2] &= \mathbb{E}[\|\mathbf{w}_t - \eta\mathbf{g}_t - \mathbf{w}_\star\|^2] \\ &= \mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - 2\eta\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_\star\rangle + \eta^2\|\mathbf{g}_t\|^2]\end{aligned}$$

rearranging:

$$\mathbb{E}[\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_\star\rangle] = \frac{\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2]}{2\eta} + \frac{\eta\,\mathbb{E}[\|\mathbf{g}_t\|^2]}{2}$$

Now, from unbiasedness and convexity, we have:

$$\mathbb{E}[\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_\star\rangle] = \mathop{\mathbb{E}}_{z_t,\dots,z_{t-1}}[\mathbb{E}[\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_\star\rangle|z_1, \dots, z_{t-1}]]$$

now, observe that w_t is a deterministic function of $z_1, \dots, z_{t-1}$, and that $\mathbf{g}_t$ is independent of $z_1, \dots, z_{t-1}$ given w_t:

$$\begin{aligned}&= \mathop{\mathbb{E}}_{z_t,\dots,z_{t-1}}[\mathbb{E}[\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_\star\rangle|w_t]] \\ &= \mathop{\mathbb{E}}_{z_1,\dots,z_{t-1}}[\langle\nabla\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_\star\rangle]\end{aligned}$$

Further, $\nabla\mathcal{L}(w_t)$ and w_t are independent of $z_t, \ldots, z_T$:

$$\begin{aligned}
&= \mathbb{E}[\langle \nabla\mathcal{L}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_\star\rangle] \\
&\geq \mathbb{E}[\mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star)]
\end{aligned}$$

So, plugging this in:

$$\mathbb{E}[\mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star)] \leq \frac{\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2]}{2\eta} + \frac{\eta\,\mathbb{E}[\|\mathbf{g}_t\|^2]}{2}$$

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star)\right] \leq \sum_{t=1}^{T} \frac{\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_\star\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_\star\|^2]}{2\eta} + \frac{\eta\,\mathbb{E}[\|\mathbf{g}_t\|^2]}{2}$$

Telescoping the first sum, and using $\|\mathbf{g}_t\| \leq G$:

$$\begin{aligned}
&\leq \frac{\mathbb{E}[\|\mathbf{w}_1 - \mathbf{w}_\star\|^2}{2\eta} + \frac{\eta T G^2}{2} \\
&\leq DG\sqrt{T}
\end{aligned}$$

Now, for the last statement in the Theorem, again use the fact that

$$\mathbb{E}[\mathcal{L}(\hat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_\star)] = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_\star)\right]$$

□

There are a couple interesting points about this proof:

1. The result appears essentially the same as in the deterministic case.
2. The expectation in the final statement is now over both the randomness in the choice of $\hat{\mathbf{w}}$ and also the randomness in the z_ts.
3. The G here is a bound on $\|\nabla\ell(\mathbf{w}, z)\|$, which could be much larger than the G in Theorem 4, which is only a bound on $\|\nabla\mathcal{L}(\mathbf{w})\|$.

Let's compare the *total computation* required by the deterministic and the stochastic versions of gradient descent. Suppose we are performing empirical risk minimization, so that the objective is

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell(\mathbf{w}, z_i)$$

Then, to compute $\nabla\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \nabla\ell(\mathbf{w}, z_i)$ takes N total gradient computations, so $O(N)$ time.
On the other hand, we can re-write $\mathcal{L}$ as:

$$\mathcal{L} = \mathop{\mathbb{E}}_{z}[\ell(\mathbf{w}, z)]$$

where P_z is the uniform distribution over $z_1, \ldots, z_N$. Then, to the gradient estimate $\mathbf{g}_t$, we choose some z uniformly at random from $z_1, \ldots, z_N$ and compute $\mathbf{g}_t = \nabla\ell(\mathbf{w}, z)$. This takes only one gradient evaluation, so $O(1)$ time.

As a result, after T iterations of gradient descent, we have spent $O(NT)$ gradient computations, but with stochastic gradient descent, we have spent only $O(T)$ gradient computations. Since in both cases the converge rate is only a function of T and not N, This makes it seem like stochastic gradient descent is significantly better than deterministic gradient descent.

However, there is a caveat: the G in Theorem 4 is potentially much smaller than the G in Theorem 6, so that it is possible that one would be better off with gradient descent. However, in general this is unlikely: G would have to be $\sqrt{N}$ times larger and as we collect more data (so N grows), this becomes less and less plausible.

3 Generalization of SGD

Remember that one issue that can come up when using ERM to solve machine learning problems is *overfitting*, in which performance on the empirical risk $\frac{1}{N}\sum_{i=1}^{N}\ell(\hat{\mathbf{w}}, z_i)$ is much better than the true loss $\mathcal{L}(\hat{\mathbf{w}}) = \mathbb{E}[\ell(\hat{\mathbf{w}}, z)]$.

One of the magical properties of SGD is that it can in some sense provably avoid overfitting. The procedure is the following: given an i.i.d. dataset $z_1, \ldots, z_N$, run SGD for N iterations (and no more!) using each datapoint one after the other with no repeating (sample without replacement). Then a random selected iterate will satisfy:

$$\mathbb{E}[\mathcal{L}(\hat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}_\star)] \leq \frac{DG}{\sqrt{N}}$$

Notice that this is a statement about the *true* loss $\mathcal{L}$, not the empirical risk! Not only that, the dependency on N is actually statistically optimal up to constant factors in many cases.

3.1 SGD for ERM

However, in practice it seems that minimizing the empirical risk frequently is superior to a single-pass over the dataset. So how can we use SGD to minimize the empirical risk? We have already discussed how if $\hat{\mathcal{L}}$ is the ERM-loss, $\hat{\mathcal{L}}(\mathbf{w}) = \mathbb{E}_{z\sim\hat{P}}[\ell(\mathbf{w}, z)]$, where $\hat{P}$ is the uniform distribution over $z_1, \ldots, z_N$. So you could just run SGD using this distribution.

In practice, however, it is more common to run the following procedure:

Algorithm 3 Stochastic Gradient Descent for ERM with shuffling

Input: Initial Point $\mathbf{w}_1$, learning rate η, time horizon T, dataset $z_1, \ldots, z_N$:
Initialize $t = 1$.
for $e = 1 \ldots T/N$ **do**
  for $i = 1 \ldots N$ **do**
    Set $\mathbf{g}_t = \nabla\ell(\mathbf{w}_t, z_i)$.
    Set $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{g}_t$.
    $t \leftarrow t + 1$.
  end for
  Optionally shuffle the dataset into a new random order.
end for

This algorithm is typically much better in practice than using with-replacement sampling, but the reasons are still somewhat mysterious. Sampling at random for every iteration is much simpler to analyze, and analyses of the in-order scheme are only recently showing improvements. For recent references, see [1, 2].

References

[1] Lam M Nguyen et al. "A Unified Convergence Analysis for Shuffling-Type Gradient Methods". In: *arXiv preprint arXiv:2002.08246* (2020).

[2] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. "Random Reshuffling: Simple Analysis with Vast Improvements". In: *arXiv preprint arXiv:2006.05988* (2020).