Lecture Notes 3: Non-Convex Gradient Descent on Smooth Losses

Instructor: Ashok Cutkosky

Starting in this section, and for much of this course, we will consider exclusively losses $\mathcal{L}$ that are *smooth*:

Definition 1. *A function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is H-smooth if $\mathcal{L}$ is differentiable at all $x \in \mathbb{R}^d$, and for all $x, y \in W$,*

$$\|\nabla \mathcal{L}(x) - \nabla \mathcal{L}(y)\| \le H\|x - y\|$$

Let's be clear: there are plenty of losses out there that are not smooth, or are only smooth with a very large value of H. However, without *some* kind of assumption about the loss it is very difficult to prove anything about how an algorithm will operate. Similarly to the way that G-Lipschitzness is the same as the gradient being bounded by G when $\mathcal{L}$ is differentiable, H-smoothness is the same as the Hessian being bounded by H when $\mathcal{L}$ is twice-differentiable:

Proposition 2. *Suppose $\mathcal{L}$ is twice differentiable. Then $\mathcal{L}$ is H-smooth if and only if $\|\nabla^2 \mathcal{L}(\mathbf{w})\|_{op} \le H$ for all $\mathbf{w}$.*

Proof. The proof is essentially the same as in the Lipschitz case. Suppose $\mathcal{L}$ is H-smooth. Let x be some arbitrary point and let v be an arbitary unit vector. Define $f : \mathbb{R} \to \mathbb{R}$ by $f(t) = \mathcal{L}(x + tv)$. Then by definition of directional derivatives, $f'(0) = \langle \nabla \mathcal{L}(x + tv), v\rangle$ and $f''(0) = \nabla^2 \mathcal{L}(x + tv)v$. Now we have:

$$\begin{aligned}
\|\nabla^2 \mathcal{L}(x)v\| &= |f''(0)| \\
&= \left|\lim_{\delta \to 0} \frac{f'(\delta) - f'(0)}{\delta}\right| \\
&= \left|\lim_{\delta \to 0} \frac{\langle \nabla \mathcal{L}(x + \delta v) - \nabla \mathcal{L}(x), v\rangle}{\delta}\right| \\
&\le \lim_{\delta \to 0} \frac{\|\nabla \mathcal{L}(x + \delta v) - \nabla \mathcal{L}(x)\|\|v\|}{\delta} \\
&\le H
\end{aligned}$$

Now since x and v were arbitrary, $\|\mathcal{L}(\mathbf{w})\|_{\text{op}} \le H$ for all $\mathbf{w}$.

On the other hand, suppose $\|\mathcal{L}(\mathbf{w})\|_{\text{op}} \le H$ for all $\mathbf{w}$. Then by mean-value theorem, for any x and y:

$$\nabla \mathcal{L}(x) = \nabla \mathcal{L}(y) + \nabla^2 \mathcal{L}(z)(x - y)$$

for some z on the line segment connecting x and y. Then, since $\|\nabla^2 \mathcal{L}(z)\| \le H$, we have $\|\nabla \mathcal{L}(x) - \nabla \mathcal{L}(y)\| \le H\|x - y\|$. □

Smoothness is at least approximately satisfied by essentially any continuous function, in the sense that any continuous function can be replaced with a approximation that is smooth. The quality of the approximation can be traded off for the amount of smoothness. Formally, the following theorem holds:

Theorem 3. *Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be a G-Lipschitz function. Then for any H, consider*

$$\hat{\mathcal{L}}(x) = \inf_y \mathcal{L}(y) + \frac{H}{2}\|x - y\|^2$$

Then $\hat{\mathcal{L}}$ is H-smooth and G-Lipschitz, and for all $x \in W$,

$$|\hat{\mathcal{L}}(x) - \mathcal{L}(x)| \le \frac{G^2}{H}$$

Because we will not be assuming that $\mathcal{L}$ is convex, in general we will not be able to actually show that our algorithms in fact minimizes $\mathcal{L}$. Instead, we will often simply try to approach a *critical point*:

Definition 4. *A* critical point*, or* first-order stationary point *of $\mathcal{L}$ is a point x such that* $\nabla\mathcal{L}(x) = 0$.

The search for critical points is motivated by the observation that any minimizer of $\mathcal{L}$ must also be a critical point, so that finding a critical point is a necessary but not sufficient condition for actually minimizing $\mathcal{L}$. There is some active research investigating the degree to which this condition may actually be sufficient. In fact, one of the key desirable properties of convex functions is that any critical point must also minimize $\mathcal{L}$.

Since finding a point where the gradient is *exactly* zero may be difficult, we will instead relax the goal slightly to find an *approximate* critical point:

Definition 5. *A ϵ-approximate critical point of $\mathcal{L}$ is a point x such that $\|\nabla\mathcal{L}(x)\| \le \epsilon$*

Now let's re-analyze gradient descent assuming smoothness, but not convexity.

Algorithm 1 Gradient Descent

```
Input: Initial Point $\mathbf{w}_1$, learning rate $\eta$, 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
```

In order to analyze gradient descent this time, we need the following critical Lemma about how the gradient influences the local behavior of a smooth function:

Lemma 6. *Suppose $\mathcal{L}$ is H-smooth. Then for any $\mathbf{w}$ and $\mathbf{v}$,*

$$\mathcal{L}(\mathbf{w}+\mathbf{v}) \le \mathcal{L}(\mathbf{w}) + \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle + \frac{H}{2}\|\mathbf{v}\|^2$$

Proof. Since $\mathcal{L}$ is differentiable (since all smooth functions are differentiable), we can apply the fundamental theorem of calculus to get:

$$\begin{aligned}
\mathcal{L}(\mathbf{w}+\mathbf{v}) &= \mathcal{L}(\mathbf{w}) + \int_0^1 \frac{d}{dt}\mathcal{L}(\mathbf{w}+t\mathbf{v})\, dt\\
&= \mathcal{L}(\mathbf{w}) + \int_0^1 \langle \nabla\mathcal{L}(\mathbf{w}+t\mathbf{v}), \mathbf{v}\rangle\, dt\\
&= \mathcal{L}(\mathbf{w}) + \int_0^1 \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle\, dt + \langle \nabla\mathcal{L}(\mathbf{w}+t\mathbf{v}) - \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle\, dt\\
&= \mathcal{L}(\mathbf{w}) + \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle + \int_0^1 \langle \nabla\mathcal{L}(\mathbf{w}+t\mathbf{v}) - \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle\, dt\\
&\le \mathcal{L}(\mathbf{w}) + \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle + \int_0^1 \|\nabla\mathcal{L}(\mathbf{w}+t\mathbf{v}) - \nabla\mathcal{L}(\mathbf{w})\|\|\mathbf{v}\|\, dt
\end{aligned}$$

Now use the fact that $\mathcal{L}$ is H-smooth:

$$\begin{aligned}
&\le \mathcal{L}(\mathbf{w}) + \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle + \int_0^1 H\|t\mathbf{v}\|\|\mathbf{v}\|\, dt\\
&\le \mathcal{L}(\mathbf{w}) + \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle + H\|\mathbf{v}\|^2\int_0^1 t\, dt\\
&\le \mathcal{L}(\mathbf{w}) + \langle \nabla\mathcal{L}(\mathbf{w}), \mathbf{v}\rangle + \frac{H}{2}\|\mathbf{v}\|^2
\end{aligned}$$

□

This Lemma is a foundational result that enables almost all analysis of gradient-descent based algorithms today. Let us see it in action by using it to understand what happens in one step of gradient descent:

Lemma 7. *Suppose $\mathcal{L}$ is H-smooth. Then every step of gradient descent (Algorithm 1) satisfies:*

$$\mathcal{L}(\mathbf{w}_{t+1}) \le \mathcal{L}(\mathbf{w}_t) - \left(1 - \frac{H\eta}{2}\right)\eta\|\nabla\mathcal{L}(\mathbf{x}_t)\|^2$$

Moreover, if $0 \le \eta \le \frac{1}{H}$,

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

Proof. The second statement of the Lemma follows immediately from the first statement, so let's just prove the first statement. This turns out to be a nearly immediate consequence of the key Lemma 6:

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

now use the fact that $\mathbf{w}_{t+1} - \mathbf{w}_t = -\eta\nabla\mathcal{L}(\mathbf{w}_t)$:

$$\begin{aligned}&= \mathcal{L}(\mathbf{w}_t) - \eta\|\nabla\mathcal{L}(\mathbf{w}_t)\|^2 + \frac{H}{2}\eta^2\|\nabla\mathcal{L}(\mathbf{w}_t)\|^2\\ &= \mathcal{L}(\mathbf{w}_t) - \left(1 - \frac{H\eta}{2}\right)\eta\|\nabla\mathcal{L}(\mathbf{x}_t)\|^2\end{aligned}$$

□

This Lemma tells us that the *function progress* of gradient descent is related to the *gradient magnitude*. Instead of being directly proportional, it is proportional to the *square* of the gradient magnitude. This is because there are two compounding effects that make function progress slow down when the gradient is small: first, a small gradient means that $\mathbf{w}_{t+1}$ is closer to $\mathbf{w}_t$. Second, a small gradient means that the change in function value produced by a change in input $\mathbf{w}$ is small.

So, let's use this knowledge to show that gradient descent will find an approximate critical point, and quantify the number of gradient computations required to do so:

Theorem 8. *Suppose $\mathcal{L}$ is H-smooth, and $\eta \le \frac{1}{H}$. Define $\Delta = \mathcal{L}(\mathbf{w}_1) - \inf_{\mathbf{w}}\mathcal{L}(\mathbf{w})$. Then gradient descent guarantees:*

$$\sum_{t=1}^{T}\|\nabla\mathcal{L}(\mathbf{w}_t)\|^2 \le \frac{2\Delta}{\eta}$$

In particular, if $\eta = \frac{1}{H}$ and $\hat{\mathbf{w}}$ is selected from $\mathbf{w}_1, \dots, \mathbf{w}_T$ uniformly at random, we have:

$$\mathbb{E}[\|\nabla\mathcal{L}(\hat{\mathbf{w}})\|] \le \frac{\sqrt{2H\Delta}}{\sqrt{T}}$$

Proof. By Lemma 7, we have:

$$\begin{aligned}\mathcal{L}(\mathbf{w}_{t+1}) &\le \mathcal{L}(\mathbf{w}_t) - \frac{\eta}{2}\|\nabla\mathcal{L}(\mathbf{w}_t)\|^2\\ \mathcal{L}(\mathbf{w}_{t+1}) - \mathcal{L}(\mathbf{w}_t) &\le -\frac{\eta}{2}\|\nabla\mathcal{L}(\mathbf{w}_t)\|^2\end{aligned}$$

now, sum over t from 1 to T. Notice that we never compute $\mathbf{w}_{T+1}$, but it still exists for analysis:

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

The right hand side is a telescoping sum:

$$\mathcal{L}(\mathbf{w}_{T+1}) - \mathcal{L}(\mathbf{w}_1) \leq -\sum_{t=1}^{T} \frac{\eta}{2}\|\nabla \mathcal{L}(\mathbf{w}_t)\|^2$$

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

$$\sum_{t=1}^{T} \|\nabla \mathcal{L}(\mathbf{w}_t)\|^2 \leq \frac{2\Delta}{\eta}$$

Now for the second part of the Theorem, notice that $\hat{\mathbf{w}}$ is chosen at randomfrom $\mathbf{w}_1, \dots, \mathbf{w}_T$, then:

$$\mathbb{E}[\|\nabla \mathcal{L}(\hat{\mathbf{w}})\|^2] = \frac{1}{T}\sum_{t=1}^{T} \|\nabla \mathcal{L}(\mathbf{w}_t)\|^2 \leq \frac{2\Delta}{\eta T}$$

Next, by Jensen inequality,

$$\mathbb{E}[\|\nabla \mathcal{L}(\hat{\mathbf{w}})\|] \leq \sqrt{\mathbb{E}[\|\nabla \mathcal{L}(\hat{\mathbf{w}})\|^2]}$$

so that plugging in the value $\eta = \frac{1}{H}$ finishes the proof. □

0.1 How to Invent Gradient Descent

Although the preceeding analysis shows that gradient descent has some favorable properties, one might wonder why we consider this algorithm at all. Is it just based on some weird physical intuition? There isn't actually any physics happening here, so there's no particular reason to believe that physical intuition is actually useful - maybe we can actually make algorithms that perform better than gradient descent! It turns out that often one *can* indeed do better than gradient descent, and we will look a bit at such algorithms later, but regardless there is a little more reason in the design of the gradient descent algorithm than simply pulling it out of a hat.

Let's try to design an iterative algorithm for optimizing $\mathcal{L}$ from first principles. One desirable property might be that we want the function to continuously decrease as the algorithm progresses. That is, $\mathcal{L}(\mathbf{w}_{t+1}) \leq \mathcal{L}(\mathbf{w}_t)$. Further, we want the decrease to happen as fast as possible, so that $\mathcal{L}(\mathbf{w}_t) - \mathcal{L}(\mathbf{w}_{t+1})$ is as large as possible.

To do this, we'll consider a *local upper bound* on $\mathcal{L}$. That is, given $\mathbf{w}_t$, we will define a new function $U_{\mathbf{w}_t}$ such that:

1. $U_{\mathbf{w}_t}(\mathbf{w}_t) = \mathcal{L}(\mathbf{w}_t)$.
2. $U_{\mathbf{w}_t}(\mathbf{w}) \geq \mathcal{L}(\mathbf{w})$ for all $\mathbf{w}$.
3. $U_{\mathbf{w}_t}$ is "simple" in the sense that we can easily compute the minimizer $\text{argmin}_{\mathbf{w}} U_{\mathbf{w}_t}(\mathbf{w})$ in closed form.

The idea is that $U_{\mathbf{w}_t}$ should somehow encapsulate the information available to the algorithm about $\mathcal{L}$ near the point $\mathbf{w}_t$.

Given such an upper bound, our algorithm will be:

1. Form $U_{\mathbf{w}_t}$.
2. Set $\mathbf{w}_{t+1} = \text{argmin}\, U_{\mathbf{w}_t}(\mathbf{w})$.

This algorithm is guaranteed to have $\mathcal{L}(\mathbf{w}_{t+1}) \leq \mathcal{L}(\mathbf{w}_t)$ because:

$$\mathcal{L}(\mathring{w}_{t+1}) \leq U_{\mathbf{w}_t}(\mathbf{w}_{t+1}) \leq U_{\mathbf{w}_t}(\mathbf{w}_t) = \mathcal{L}(\mathbf{w}_t)$$

Furthermore, the "gap" between $\mathcal{L}(\mathbf{w}_{t+1})$ and $\mathcal{L}(\mathbf{w}_t)$ can be lower bounded by $U_{\mathbf{w}_t}(\mathbf{w}_t) - U_{\mathbf{w}_t}(\mathbf{w}_{t+1})$. Thus, by setting $\mathbf{w}_{t+1} = \text{argmin}\, U_{\mathbf{w}_t}$, we are maximizing this gap. It only remains to find a reasonable candidate for $U_{\mathbf{w}_t}$. To this end, we again apply the key Lemma 6:

Proposition 9. *Suppose $\mathcal{L}$ is H-smooth. For any $\mathbf{w}$, define $U_{\mathbf{x}}(\mathbf{y}) = \mathcal{L}(\mathbf{x}) + \langle \nabla\mathcal{L}(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{H}{2}\|\mathbf{x} - \mathbf{y}\|^2$. Then for all $\mathbf{x}$ and $\mathbf{y}$:*

$$\mathcal{L}(\mathbf{x}) = U_{\mathbf{x}}(\mathbf{x})$$
$$\mathcal{L}(\mathbf{y}) \le U_{\mathbf{x}}(\mathbf{y})$$

Furthermore,

$$\underset{\mathbf{y}}{\mathit{argmin}}\, U_{\mathbf{x}}(\mathbf{y}) = \mathbf{x} - \frac{\nabla\mathcal{L}(\mathbf{x})}{H}$$

Proof. First, it is clear from the definition that $\mathcal{L}(\mathbf{x}) = U_{\mathbf{x}}(\mathbf{x})$. Next, the statement $\mathcal{L}(\mathbf{y}) \le U_{\mathbf{x}}(\mathbf{y})$ follows directly from Lemma 6. Finally, differentiating with respect to $\mathbf{y}$ shows that to compute the argmin we need:

$$0 = \nabla\mathcal{L}(\mathbf{x}) - H(\mathbf{x} - \mathbf{y})$$

so solving for $\mathbf{y}$ yields the last part of the result. $\square$

Notice that this Proposition actually recovers the gradient descent algorithm, and automatically is using the best learning rate! This strategy of defining a bounding function U is a common theme in optimization of which gradient descent is simply the first example. This should immediately suggest a way to perhaps improve the algorithm: if we could come up with a better bound U, maybe we could make a better algorithm.