CPSC 532D — 15. NONCONVEX OPTIMIZATION

Danica J. Sutherland
University of British Columbia, Vancouver
Fall 2023

In our analysis of gradient descent last time, essentially the first thing we did was assume the target function f was convex. This then allows us to decompose the overall excess error, how much worse we are than the irreducible Bayes error L^*, as

$$L_{\mathcal{D}}(\mathcal{A}(S)) - L^* \le \underbrace{L_{\mathcal{D}}(\mathcal{A}(S)) - L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S))}_{\text{optimization error}} + \underbrace{L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - \inf_{h^* \in \mathcal{H}} L_{\mathcal{D}}(h^*)}_{\text{estimation error}} + \underbrace{\inf_{h^* \in \mathcal{H}} L_{\mathcal{D}}(h^*) - L^*}_{\text{approximation error}}.$$

If the loss is convex (as a function of the parameters we're optimizing), then we know how to bound each of those three terms. But lots of interesting functions we'd like to optimize aren't convex.

For an explicit example, consider the following *deep linear model*:

$$h_{W,v}(x) = v \cdot (Wx),$$

which is a two-layer deep network with *identity* activations $\sigma(t) = t$. This is just a linear model $(W^\top v) \cdot x$, but it's not parameterized as one; these have been used mostly as a stepping stone towards a theory of actual deep learning, because they exhibit some of the properties of deep networks, like nonconvexity. We can see that because $h_{-W,-v}(x) = (-v) \cdot (-Wx) = h_{W,v}(x)$, and so for any typical loss function we have $\ell(h_{W,v}, z) = \ell(h_{-W,-v}, z)$. For this to be a convex function of (W, v), we would therefore need $\frac{1}{2}(W, v) + \frac{1}{2}(-W, -v) = (0, 0)$ to have smaller loss: clearly this is not true in nontrivial settings, since this would imply the constant zero predictor is always globally optimal.

You can do a similar thing with e.g. ReLU nets: permute the order of entries in the first layer's matrix, then take an average of all of those to get a W with all entries constant.

1 GRADIENT DESCENT WITH β-SMOOTH FUNCTIONS

1.1 β-*smooth functions*

DEFINITION 1. We say a function f is β-*smooth* if it is differentiable everywhere, and its gradient ∇f is β-Lipschitz.

Note that this is not what analysts mean when they say a "smooth function" (i.e. infinitely differentiable.

PROPOSITION 2. *If f is twice-differentiable, it is β-smooth iff for all x, $\nabla^2 f(x) \preceq \beta I$.*

This is essentially the same as the characterization that differentiable functions are ρ-Lipschitz iff their gradient norm is at most ρ everywhere.

PROPOSITION 3. *Suppose f is β-smooth. Then for any x and y in its domain,*

$$\left| f(y) - f(x) - \langle \nabla f(x), y - x \rangle \right| \le \frac{1}{2} \beta \left\| x - y \right\|^2 :$$

its deviation from its tangent planes is upper-bounded by a quadratic.

For more, visit https://cs.ubc.ca/~dsuth/532D/23w1/.

Proof. For any x_0, x_1, let $x_\alpha = (1-\alpha)x_0 + \alpha x_1$ for all $\alpha \in (0,1)$, and define $g : [0,1] \to \mathbb{R}$ by $g(\alpha) = f(x_\alpha)$. Notice that $g'(\alpha) = \langle \nabla f(x_\alpha), x_1 - x_0 \rangle$, and so by the fundamental theorem of calculus we have

$$\begin{aligned} f(x_1) - f(x_0) = g(1) - g(0) &= \int_0^1 g'(\alpha) \mathrm{d}\alpha \\ &= \int_0^1 \langle \nabla f(x_\alpha) \underbrace{- \nabla f(x_0) + \nabla f(x_0)}_{0}, x_1 - x_0 \rangle \mathrm{d}\alpha \\ &= \langle \nabla f(x_0), x_1 - x_0 \rangle + \int_0^1 \langle \nabla f(x_\alpha) - \nabla f(x_0), x_1 - x_0 \rangle \mathrm{d}\alpha. \end{aligned}$$

Thus

$$\begin{aligned} |f(x_1) - f(x_0) - \langle \nabla f(x_0), x_1 - x_0 \rangle| &= \left| \int_0^1 \langle \nabla f(x_\alpha) - \nabla f(x_0), x_1 - x_0 \rangle \mathrm{d}\alpha \right| \\ &\le \int_0^1 |\langle \nabla f(x_\alpha) - \nabla f(x_0), x_1 - x_0 \rangle| \, \mathrm{d}\alpha \\ &\le \int_0^1 \|\nabla f(x_\alpha) - \nabla f(x_0)\| \, \|x_1 - x_0\| \, \mathrm{d}\alpha \\ &\le \int_0^1 \beta \, \|x_\alpha - x_0\| \, \|x_1 - x_0\| \, \mathrm{d}\alpha; \end{aligned}$$

since $x_\alpha - x_0 = \alpha(x_1 - x_0)$, this is

$$|f(x_1) - f(x_0) - \langle \nabla f(x_0), x_1 - x_0 \rangle| \le \beta \, \|x_1 - x_0\|^2 \int_0^1 \alpha \mathrm{d}\alpha = \frac{1}{2} \beta \, \|x_1 - x_0\|^2 . \qquad \square$$

1.2 *Descent lemma*

This allows us to characterize what one step of gradient descent does on β-smooth functions. Note that there's no need for subgradients, since β-smooth functions are by definition differentiable, and we won't handle a projection step or stochastic gradients here (though you can do versions of both with some extra work).

Let $x_{t+1} = x_t - \eta \nabla f(x_t)$ for a β-smooth function f. Then by Proposition 3, we have

$$\begin{aligned} f(x_{t+1}) &\le f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{1}{2} \beta \, \|x_{t+1} - x_t\|^2 \\ &= f(x_t) - \eta \langle \nabla f(x_t), \nabla f(x_t) \rangle + \frac{1}{2} \beta \left\| -\eta \nabla f(x_t) \right\|^2 \\ &= f(x_t) - \eta \left(1 - \frac{1}{2} \eta \beta \right) \|\nabla f(x_t)\|^2 . \end{aligned}$$

Thus, if

$$1 - \frac{1}{2}\eta\beta > 0 \qquad \text{iff} \qquad \eta < \frac{2}{\beta},$$

we know that either $\nabla f(x_t) = 0$ (so we're at a local min) or else $f(x_{t+1}) < f(x_t)$. So, this means that gradient descent is a "descent method": each step decreases the objective, and so must eventually reach a point where $\nabla f(x_t) = 0$.

For convex functions, such a point will be a global min. But for nonconvex functions, we can only say that it's a stationary point: it might be a local but non-global minimizer, or a saddle point. (A local max could only happen if we happened to initialize exactly on it.)

Aside: SGD convergence

This type of analysis can be generalized to show that even SGD eventually reaches a stationary point:

PROPOSITION 4 (Corollary 1 of [KR23]). *Let* $\inf_x f(x) \geq f^{\inf} \in \mathbb{R}$ *be* β*-smooth. Let* $\hat{g}_t$ *be independent such that* $\mathbb{E}[\hat{g}_t \mid x_t] = \nabla f(x_t)$ *and*

$$\mathbb{E}[\|\hat{g}_t\|^2 \mid x_t] \leq 2A(f(x_t) - f^{\inf}) + B\|\nabla f(x_t)\|^2 + C$$

for some $A, B, C \geq 0$. *Fix* $\varepsilon > 0$, *and pick* $\eta = \min\left\{\frac{1}{\sqrt{\beta A T}}, \frac{1}{\beta B}, \frac{\varepsilon}{2\beta C}\right\}$. *Initialize stochastic gradient descent at* x_1, *with* $\delta_1 = f(x_1) - f^{\inf}$, *and* $x_{t+1} = x_t - \eta\hat{g}_t$. *As long as* $T \geq \frac{12\delta_1\beta}{\varepsilon^2}\max\left\{B, \frac{12\delta_1 A}{\varepsilon^2}, \frac{2C}{\varepsilon^2}\right\}$, *it holds that* $\min_{1\leq t\leq T}\mathbb{E}[\|\nabla f(x_t)\|] \leq \varepsilon$.

That is, the *best iterate* achieves ε suboptimality (in expectation) with $\mathcal{O}(1/\varepsilon^4)$ steps. The assumption on $\hat{g}_t$ is satisfied for example if the $\hat{g}_t$ have a bounded variance, or if we use subsampling for a Lipschitz loss, or various other settings.

1.3 *Are deep networks* β*-smooth?*

Is $f(w) = L_S(h_w)$ for h_w a class of deep networks β-smooth?

Consider the very simple network

$$h_{W,v}(x) = v \cdot \sigma(Wx),$$

where σ is itself β-smooth. Then the square loss for a single data point is

$$f(W, v) = (v^T\sigma(Wx) - y)^2 = v^T\sigma(Wx)\sigma(Wx)^T v - 2y\sigma(Wx)^T v + y^2,$$

and we have

$$\nabla_v f(W, v) = 2(\sigma(Wx)^T v - y)\sigma(Wx)$$

$$\nabla_v^2 f(W, v) = 2\sigma(Wx)\sigma(Wx)^T.$$

If this is unfamiliar, try looking at individual partial derivatives to see that they line up.

The Jacobian with W is more annoying, since we'd have to flatten W and reshape and stuff. But the overall Hessian of f with respect to its input parameters will have $\nabla_v^2 f$ as a block in it, and so its largest eigenvalue will depend on W: if σ is the ReLU or something similar, then large values of W will result in much larger Hessians. Thus the loss is only going to be fully β-smooth if you bound the set of possible Ws, but for any particular parameters it's going to be "locally" smooth.

Autodiff is nice....

Notice that the descent lemma doesn't actually need a global upper bound on the smoothness, just along the path from x_t to x_{t+1}. So, intuitively, we should roughly expect (stochastic) gradient descent to reach a stationary point of the loss as long as $\nabla^2 f$ doesn't blow up, i.e. in typical situations as long as none of the parameters blows up.

Aside: edge of stability

So, if we're optimizing a deep network with a fixed learning rate η, whether the descent lemma applies or not – whether gradient descent is "stable" or not – depends on whether $\eta < \frac{2}{\beta}$, or more relevantly $\beta < \frac{2}{\eta}$, for the "local" value of β. We can roughly get this local value of β by just checking the largest eigenvalue of $\nabla^2 f(x_t)$, and see whether it stays in a "stable" regime or not.

Note that the "local β" might be larger than $\max(\nabla^2 f(x_t), \nabla^2 f(x_{t+1}))$: you might go through a sharper point on the way. For instance, consider $f(x) = |x|$ on the reals: $f''(x) = 0$ for all $x \neq 0$, but the descent lemma might not apply when you switch signs, since you go through 0 which has "infinite second derivative."

Cohen et al. [Coh+21] demonstrated that in fact, optimization typically exhibits "progressive sharpening" where β increases up to $2/\eta$, then hovers around there on the "edge of stability" [also see Fox23]. Damian, Nichani, and Lee [DNL23] have recently proposed a mechanism for how this happens, based on Taylor expansions of the training process.

2 IS A STATIONARY POINT ENOUGH?

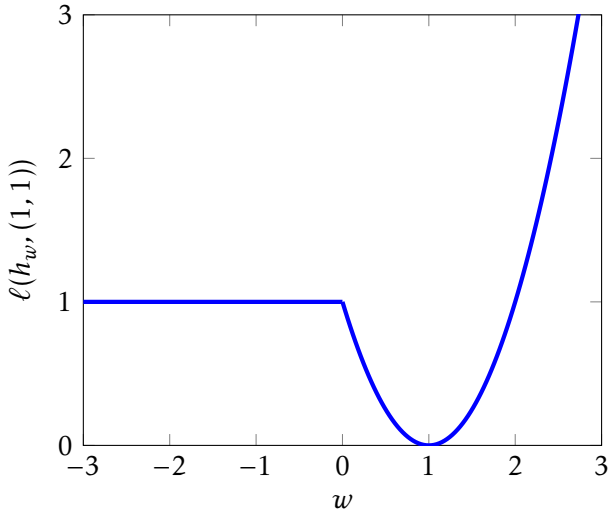
One model we can look at is deep linear nets, $f(x) = w_d \mathrm{W}_{d-1} \cdots \mathrm{W}_2 \mathrm{W}_1 x$. These are just linear models, but they're nonconvex and hierarchical and so exhibit some of the same behaviour as regular deep nets. It's reasonable to expect that, generally speaking, if something doesn't work on deep linear nets, it won't work on deep nonlinear nets either.

It turns out that for deep linear nets:

- Fortunately, all local minima in deep linear nets are global minima [Kaw16; LvB18].
- Unfortunately, stationary points can also be saddle points – including potentially "bad" saddles with $\lambda_{\min}(\nabla^2 f) = 0$ even though they're not local minima. (For example, x^3 has a saddle point like this at $x = 0$; they can be even worse in high dimensions.)
- Fortunately, in general, gradient descent almost surely converges to local minimizers, not saddles (or local maxes) [LSJR16].
- Unfortunately, doing so can take exponential time [Du+17].
- Fortunately, this doesn't happen for deep linear networks, under some conditions [ACGH19].

Unfortunately, there *are* bad local minima in nonlinear networks. For a very simple example, consider the network $h : \mathbb{R} \to \mathbb{R}$ given by $h(x) = \mathrm{ReLU}(wx)$, where $w \in \mathbb{R}$; use square loss with a single example, $(1, 1)$. Then the loss is

$$\ell(h_w, (1,1)) = \begin{cases} (w-1)^2 & w \geq 0 \\ 1 & w \leq 0 \end{cases}.$$



Any negative input is a local min (since $f(w) \geq f(v)$ for all v in a neighbourhood of w), but it's not a global min (since $f(1) = 0$). Thus, if you start gradient descent with a negative w, it's just stuck. In fact, this kind of thing can happen for almost any activation function [DLS20].

But, do bad local minima exist for realistic networks, with realistic data? Even if they do, does SGD find them?

We'll see next that, in one unrealistic (but not *too* ridiculous) setting, gradient descent always finds a local minimum.

REFERENCES

[ACGH19] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks." *ICLR*. 2019. arXiv: 1810.02281.

[Coh+21] Jeremy M. Cohen, Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. "Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability." *ICLR*. 2021. arXiv: 2103.00065.

[DLS20] Tian Ding, Dawei Li, and Ruoyu Sun. *Sub-Optimal Local Minima Exist for Neural Networks with Almost All Non-Linear Activations*. 2020. arXiv: 1911.01413.

[DNL23] Alex Damian, Eshaan Nichani, and Jason D. Lee. "Self-Stabilization: The Implicit Bias of Gradient Descent at the Edge of Stability." *ICLR*. 2023. arXiv: 2209.15594.

[Du+17] Simon S. Du, Chi Jin, Jason D. Lee, Michael I. Jordan, Barnabás Póczos, and Aarti Singh. "Gradient Descent Can Take Exponential Time to Escape Saddle Points." *NeurIPS*. 2017. arXiv: 1705.10412.

[Fox23] Curtis Fox. "A study of the edge of stability in deep learning." MSc. Thesis. University of British Columbia, 2023.

[Kaw16] Kenji Kawaguchi. "Deep Learning without Poor Local Minima." *NeurIPS*. 2016. arXiv: 1605.07110.

[KR23] Ahmed Khaled and Peter Richtárik. "Better Theory for SGD in the Nonconvex World." *TMLR* (2023).

[LSJR16] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. "Gradient Descent Only Converges to Minimizers." *COLT*. 2016.

[LvB18] Thomas Laurent and James von Brecht. "Deep Linear Networks with Arbitrary Loss: All Local Minima Are Global." *ICML*. 2018.