CPSC 532D — 13. OPTIMIZATION

Danica J. Sutherland

University of British Columbia, Vancouver

Fall 2024

We haven't yet really talked in this course about any optimization algorithms to actually *implement* our learning algorithms ERM, RLM, or SRM.

By far the most common optimization algorithm used in machine learning is (stochastic) gradient descent and its variants. Due to time this year, we're only going to talk about full-batch gradient descent, and point to papers that discuss stochastic variants. (This is a major area of research in the intersection between learning theory and optimization, which these days are becoming more integrated.) For much much more about optimization, some good resources are graduate courses by Michael Friedlander (CPSC 536M) and Mark Schmidt (CPSC "5XX"), the books of Boyd and Vandenbreghe [BV04], Nocedal and Wright [NW06], and Bubeck [Bub15], and the recent survey of Garrigos and Gower [GG23]. Chapter 14 of Shalev-Shwartz and Ben-David [SSBD14] also gives an approachable account of projected stochastic subgradient descent, which generalizes what we're talking about here.

13.1 GRADIENT DESCENT

Gradient descent tries to find $\min_w f(w)$ for some function f, such as $L_S(f_w)$. Here w should be some finite-dimensional parameter; in kernel methods, we'd typically use the representer theorem, though there's also something called "kernel gradient descent."

We start at some initial point w_1, often either 0 or a sample from, say, $\mathcal{N}(0, \sigma^2 I)$. We then update according to the rule

$$w_{t+1} = w_t - \eta_t \nabla f(w_t);$$

$\eta_t > 0$ is known as either the "learning rate" or the "step size," although note that it's not actually the size of the step since $\lVert w_{t+1} - w_t \rVert = \eta_t \lVert \nabla f(w_t) \rVert$.

One way to motivate this is to say that we should only "trust" the gradient direction locally, and then should re-check it regularly. Another way is to notice that this update actually minimizes the local quadratic approximation given by

$$g(w) = f(w_t) + \langle \nabla f(w_t), w - w_t \rangle + \frac{1}{2\eta} \lVert w - w_t \rVert^2 ;$$

if f is $\frac{1}{\eta}$-strongly convex, then g will be a global lower bound for f. Even if not, though, it'll be an okay approximation locally.

If instead of $\frac{1}{2\eta} \lVert w - w_t \rVert^2$ we use $\frac{1}{2}(w - w_t)\nabla^2 f(w_t)(w - w_t)$, i.e. the second-order Taylor expansion, this is called Newton's method. Each step of Newton's method often improves your loss much more than gradient descent, but each step is also much more computationally expensive.

We repeat this until we decide to stop, after T steps, and then return a result: this might be w_T (the "last iterate"), $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$ (the "average iterate"), $w_{\hat{t}}$ for $\hat{t} \in \arg\min_{t \in [T]} f(w_t)$ (the "best iterate"), the best iterate according to a validation set, or some other scheme.

For more, visit https://cs.ubc.ca/~dsuth/532D/24w1/.

We'll usually assume that η_t is some constant η, independent of the data, and that we optimize for a fixed number of steps T, also chosen independently of the data. In practice, other schemes are probably better; for instance, it's often better to use a *backtracking* scheme to adaptively choose η_t, or to otherwise have some kind of learning rate schedule that decreases over time.

13.2 β-SMOOTH FUNCTIONS

A common assumption in optimization is that the target function is β-smooth:

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

Definition 13.1. We say a function f is β-*smooth* if it is differentiable everywhere, and its gradient ∇f is β-Lipschitz.

Proposition 13.2. *If f is twice-differentiable, it is β-smooth iff for all w in the interior of its domain, all eigenvalues of the Hessian of f at x have absolute value at most β: $-\beta I \preceq \nabla^2 f(w) \preceq \beta I$.*

The notation $A \succeq 0$ means "is positive-semi-definite"; $A \succeq B$ means that $A - B$ is positive-semi-definite.

Proof. When f is twice-differentiable and β-smooth, we have by Taylor's theorem that for any vector δ,

$$\nabla f(w+\delta) = \nabla f(w) + \nabla^2 f(w)\delta + \mathcal{O}(\|\delta\|^2).$$

Thus by the triangle inequality,

$$\left\|\nabla^2 f(w)\delta\right\| \le \|\nabla f(w+\delta) - \nabla f(w)\| + \mathcal{O}(\|\delta\|^2).$$

Divide through by $\|\delta\|$ and apply that ∇f is β-Lipschitz:

$$\frac{\left\|\nabla^2 f(w)\delta\right\|}{\|\delta\|} \le \frac{\|\nabla f(w+\delta) - \nabla f(w)\|}{\delta} + \mathcal{O}(\|\delta\|) \le \beta + \mathcal{O}(\|\delta\|).$$

Now suppose v is a (unit-norm) eigenvector of $\nabla^2 f(w)$ with eigenvalue λ, and plug in $\delta = tv$ for a scalar t, so that $\|\delta\| = |t|$. Then $\left\|\nabla^2 f(w)\, tv\right\| = \|\lambda t v\| = |\lambda|\,|t|$. This gives us that $|\lambda| \le \beta + \mathcal{O}(t)$. Taking $t \to 0$ gives that $|\lambda| \le \beta$.

The other direction is a vector-valued version of Lemma 4.8. □

Proposition 13.3. *Suppose f is β-smooth. Then for any w and w' in its domain,*

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

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

Proof. Use x_0 for w and x_1 for w'. Then 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

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

since $x_\alpha - x_0 = (1 - \alpha)x_0 + \alpha x_1 - x_0 = \alpha(x_1 - x_0)$, this is

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

Lemma 13.4 (Descent lemma)**.** *Let $w^+ = w - \eta \nabla f(w)$ for a β-smooth function f, where $\eta < 2/\beta$. Then $f(w) - f(w^+) \geq \eta(1 - \frac{1}{2}\eta\beta) \|\nabla f(w)\|^2$, and hence either $\nabla f(w) = 0$ or $f(w^+) < f(w)$.*

Proof. By Proposition 13.3, we have

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

Since we assumed $\eta < 2/\beta$, $1 - \eta\beta/2 > 0$. The claim follows. □

So, this means that gradient descent with a small-enough learning rate is a "descent method": each step decreases the objective.

For convex functions, a point with $\nabla f(w) = 0$ is 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.)

13.3 ASIDE: CONVEX FUNCTIONS

For convex functions in particular (with a slightly smaller learning rate), we can use the following lemma to help in a proof of overall convergence: this lemma relates the improvement of the descent lemma to how much closer a step gets us to some "target point" (presumably the minimizer) w^*:

Lemma 13.5. *Let f be a convex, β-smooth function, and suppose that $\eta \leq 1/\beta$. Let w, w^* be arbitrary points in the interior of the domain of f. Then*

$$f(w^+) - f(w) \leq \frac{1}{2\eta}\left[\|w - w^*\| - \left\|w - \eta\nabla f(w) - w^*\right\|\right].$$

Proof. The first-order characterization of convexity implies that $f(w^*) \geq f(w) - \langle \nabla f(w), w^* - w\rangle$, or equivalently $f(w) \leq f(w^*) + \langle \nabla f(w), w - w^*\rangle$. Thus, starting from Lemma 13.4 and using $\eta \leq 1/\beta$,

$$\begin{aligned}
f(w^+) &\leq f(w) - \eta\left(1 - \frac{1}{2}\beta\eta\right)\|\nabla f(w)\|^2 \\
&\leq f(w) - \frac{1}{2}\eta\|\nabla f(w)\|^2 \\
&\leq f(w^*) + \langle \nabla f(w), w - w^*\rangle - \frac{1}{2}\eta\|\nabla f(w)\|^2 \\
&= f(w^*) + \frac{1}{2\eta}\left[2\eta\langle \nabla f(w), w - w^*\rangle - \eta^2\|\nabla f(w)\|^2\right] \\
&= f(w^*) + \frac{1}{2\eta}\left[\|w - w^*\|^2 - \|w - w^*\|^2 + 2\eta\langle \nabla f(w), w - w^*\rangle - \eta^2\|\nabla f(w)\|^2\right] \\
&= f(w^*) + \frac{1}{2\eta}\|w - w^*\|^2 - \frac{1}{2\eta}\left[\|w - w^*\|^2 - 2\eta\langle \nabla f(w), w - w^*\rangle + \eta^2\|\nabla f(w)\|^2\right] \\
&= f(w^*) + \frac{1}{2\eta}\|w - w^*\|^2 - \frac{1}{2\eta}\left\|(w - w^*) - \eta\nabla f(w)\right\|^2 \\
&= f(w^*) + \frac{1}{2\eta}\left(\|w - w^*\|^2 - \left\|w^+ - w^*\right\|^2\right).
\end{aligned}$$

□

Proposition 13.6. *Let f be convex and β-smooth, with $\eta \leq 1/\beta$. Then the procedure that initializes at w_0 and then sets $w_t = w_{t-1} - \eta\nabla f(w_t)$ satisfies for all $T \geq 1$ that*

$$f(w_T) - f(w^*) \leq \frac{1}{2\eta T}\|w_0 - w^*\|^2,$$

and also that

$$f\left(\frac{1}{T}\sum_{t=1}^{T} w_t\right) - f(w^*) \leq \frac{1}{2\eta T}\|w_0 - w^*\|^2.$$

Proof. For each step t,

$$f(w_t) - f(w^*) \leq \frac{1}{2\eta}\left(\|w_{t-1} - w^*\|^2 - \|w_t - w^*\|^2\right).$$

Using the descent lemma and then Lemma 13.5, we know that

$$\begin{aligned} f(w_T) - f(w^*) &\leq \frac{1}{T}\sum_{t=1}^{T}(f(w_t) - f(w^*)) \\ &\leq \frac{1}{2\eta T}\sum_{t=1}^{T}\left(\|w_{t-1} - w^*\|^2 - \|w_t - w^*\|^2\right) \\ &= \frac{1}{2\eta T}\left[\|w_0 - w^*\|^2 - \|w_1 - w^*\|^2 + \|w_1 - w^*\|^2 - \cdots - \|w_T - w^*\|^2\right] \\ &= \frac{1}{2\eta T}\left(\|w_0 - w^*\|^2 - \|w_T - w^*\|^2\right) \\ &\leq \frac{1}{2\eta T}\|w_0 - w^*\|^2 . \end{aligned}$$

By Jensen's inequality, $f\left(\frac{1}{T}\sum_{t=1}^{T} w_t\right) \leq \frac{1}{T}\sum_{t=1}^{T} f(w_t)$, and so the same bound applies. □

13.3.1 *Aside: SGD non-convex convergence*

The analysis above can be pretty-easily extended to SGD; see e.g. Chapter 14 of Shalev-Shwartz and Ben-David [SSBD14] or the recent survey of Garrigos and Gower [GG23]. It can be generalized further, though more complicatedly, to show that even SGD eventually reaches a stationary point, even for non-convex functions:

Proposition 13.7 (Corollary 1 of [KR23]). *Let* $\inf_x f(x) \geq f^{\text{inf}} \in \mathbb{R}$ *be* β*-smooth. Let* $\hat{g}_t \mid x_t$ *be independent such that* $\mathbb{E}[\hat{g}_t \mid x_t] = \nabla f(x_t)$ *and*

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

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* $\delta_1 = f(x_1) - f^{\text{inf}}$, *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.

13.4 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

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

and we have

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

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

Autodiff is nice....

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.

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.

13.5 IS A STATIONARY POINT ENOUGH?

One model we can look at is deep linear nets, $f(x) = w_d W_{d-1} \cdots W_2 W_1 x$. 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.

To see that they're nonconvex: consider just a depth two model on scalars, $f(x) = vwx$ for $v, w \in \mathbb{R}$. Consider square loss with the training set $S = ((1, 1))$. Then $L_S(f) = (vw - 1)^2$, whose minimizers are

$$\{(v, w) : vw = 1\} = \{(v, 1/v) : v \neq 0\}.$$

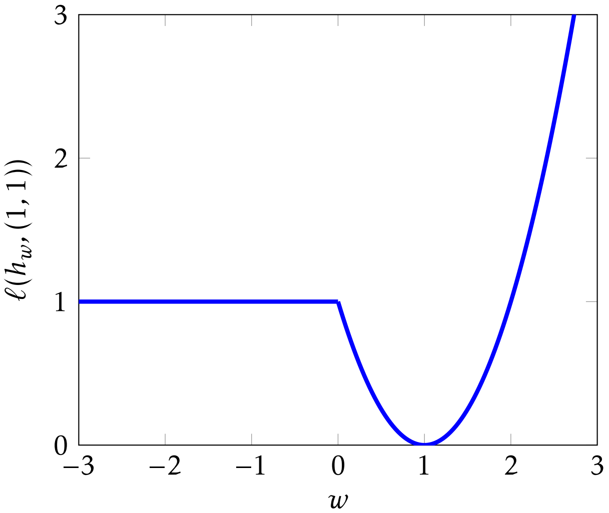
But this is *not* a convex set: it's a line in $\mathbb{R}^2$ with the point $(0, 0)$ cut out of it. The set of minimizers of convex functions must be convex, so therefore L_S is not convex.

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 $h(x) = \operatorname{ReLU}(wx)$, 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 (non-strict) 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, bad (strict) local minima can appear for almost any activation function [DLS20], and with more units, the loss landscape has such points almost all the time.

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

This is very much still an active topic of research, but 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.

[Bub15] Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. *Foundations and Trends in Machine Learning* 8.3-4 (2015). arXiv: 1405.4980.

[BV04] Stephen Boyd and Lieven Vandenbreghe. *Convex Optimization*. Cambridge University Press, 2004.

[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.

[GG23] Guillaume Garrigos and Robert M. Gower. *Handbook of Convergence Theorems for (Stochastic) Gradient Methods*. 2023. arXiv: 2301.11235.

[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.

[NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.