CPSC 532D — 11. OPTIMIZATION

Danica J. Sutherland
University of British Columbia, Vancouver
Fall 2025

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

Recall our usual decomposition (1.5)

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

and that we've often done as in (1.6) that

$$\begin{aligned} &L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^*) \\ &\quad = \underbrace{L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_S(\mathrm{ERM}_{\mathcal{H}}(S))}_{\le \sup_h L_{\mathcal{D}}(h) - L_S(h)} + \underbrace{L_S(\mathrm{ERM}_{\mathcal{H}}(S)) - L_S(h^*)}_{\le 0} + \underbrace{L_S(h^*) - L_{\mathcal{D}}(h^*)}_{\text{small by Hoeffding}}. \end{aligned}$$

What about if we try to get the ERM, but don't exactly achieve it? We can then do

$$\begin{aligned} L_{\mathcal{D}}(\mathcal{A}(S)) - L_{\mathcal{D}}(h^*) &= \underbrace{L_{\mathcal{D}}(\mathcal{A}(S)) - L_S(\mathcal{A}(S))}_{\le \sup_h L_{\mathcal{D}}(h) - L_S(h)} \\ &\quad + \underbrace{L_S(\mathcal{A}(S)) - L_S(\mathrm{ERM}_{\mathcal{H}}(S))}_{\text{optimization error}} + \underbrace{L_S(\mathrm{ERM}_{\mathcal{H}}(S)) - L_S(h^*)}_{\le 0} + \underbrace{L_S(h^*) - L_{\mathcal{D}}(h^*)}_{\text{small by Hoeffding}}. \end{aligned}$$

Today we'll study this optimization error a bit.

By far the most common optimization algorithm used in machine learning is (stochastic) gradient descent and its variants. We'll cover the foundational ideas, but there is a *lot* more in this area, and generally, learning theory and optimization are becoming much more integrated. For much much more about optimization, some good resources are graduate courses by Michael Friedlander (CPSC 536M) and Mark Schmidt (CPSC "5XX", to *eventually* be offered as an actual class), 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.

11.1 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 parameter vector, for example the flattened vector of all parameters in a network; most of what we'll talk about works for w in a Hilbert space.

For more, visit https://cs.ubc.ca/~dsuth/532D/25w1/.

We start at some initial point w_0, often either 0 or a sample from, say, $\mathcal{N}(0, \sigma^2 \mathrm{I})$. We then update according to the rule

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

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

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

If instead of $\frac{1}{2\eta}\|w - w_t\|^2$ 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.

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

This approximation is useful in that if f is $\frac{1}{\eta}$-*strongly convex*, then g will be a global lower bound for f (Proposition C.3). Even if not, though, it'll be an okay approximation to a lower bound locally, since it's the first-order Taylor expansion plus a term that says "don't go too far.'

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.

We'll usually assume for now 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. In practice, we also probably do set some upper bound T on the optimization time, but will frequently stop sooner if it seems like we're done.

11.2 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 11.1. We say a function f is β-*smooth* if it is differentiable everywhere, and its gradient ∇f is β-Lipschitz.

Proposition 11.2. *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:* $-\beta \mathrm{I} \preceq \nabla^2 f(w) \preceq \beta \mathrm{I}$.

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

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

$$\nabla f(w + \delta) = \nabla f(w) + \nabla^2 f(w)\delta + \|\delta\|^2 \xi,$$

where ξ is an error vector depending on δ but with $\|\xi\| \leq C$ for some constant C. Now, eigendecompose $\nabla^2 f(w)$ into eigenvalues λ_i and corresponding orthonormal eigenvectors v_i, so we can write $\nabla^2 f(w)\delta = \sum_i \lambda_i \langle v_i, \delta \rangle v_i$. Also let $\delta = tv$ for some

$\|v\| = 1$. Then we have that

$$\nabla f(w + tv) - \nabla f(w) = t \sum_i \lambda_i \langle v_i, v \rangle v_i + t^2 \xi.$$

Now suppose that ∇f is β-Lipschitz, meaning that the norm of this expression is at most βt, or equivalently

$$\left\| \sum_i \lambda_i \langle v_i, v \rangle v_i + t\xi \right\| \leq \beta.$$

Choosing $v = v_i$, this becomes $\|\lambda_i v_i + t\xi\| = \sqrt{\lambda_i^2 + t\langle \xi, v_i \rangle + t^2 \|\xi\|^2} \leq \beta$; taking $t \to 0$ and noting that while ξ can depend on t, its norm is bounded for all t, this implies that we must have $|\lambda_i| \leq \beta$.

For the other direction, suppose that each $|\lambda_i| \leq \beta$. Then, via Cauchy-Schwarz,

$$\left\| \sum_i \lambda_i \langle v_i, v \rangle v_i \right\|^2 \leq \sum_i \lambda_i^2 \langle v_i, v \rangle^2 \leq \sum_i \lambda_i^2 \leq \beta^2;$$

thus $\|\nabla f(w + tv) - \nabla f(w) - t^2 \xi\| \leq t\beta$. Rewriting this we get

$$\left\| \frac{\nabla f(w + tv) - \nabla f(w)}{t} - t\xi \right\| \leq \beta;$$

taking the limit as $t \to 0$, this becomes that the directional derivative of ∇f at w in the direction v has norm at most β. Integrating along the line from w to any w', as in the proof of Lemma 4.4, thus shows that $\|\nabla f(w') - \nabla f(w)\| \leq \beta \|w' - w\|$ as desired. □

Proposition 11.3. *Suppose f is β-smooth. Then for any w and w' such that the line segment from w to w' is in its domain,*

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

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

Proof. By the Lagrange form of Taylor's theorem, we have that

$$f(w') = f(w) + \langle \nabla f(w), w' - w \rangle + \frac{1}{2} \langle w' - w, \nabla^2 f(q)(w' - w) \rangle$$

for some q on a line segment between w and w'.

Eigendecompose $\nabla^2 f(q) = \sum_i \lambda_i v_i v_i^\mathsf{T}$, where the v_i are orothonormal; then we have

$$\langle w' - w, \nabla^2 f(q)(w' - w) \rangle = \left\langle w' - w, \sum_i \lambda_i \langle w' - w, v_i \rangle v_i \right\rangle = \sum_i \lambda_i \langle w' - w, v_i \rangle^2,$$

and so

$$\left| \langle w' - w, \nabla^2 f(q)(w' - w) \rangle \right| \leq \sum_i |\lambda_i| \langle w' - w, v_i \rangle^2 \leq \left(\max_i |\lambda_i| \right) \sum_i \langle w' - w, v_i \rangle^2.$$

But since the v_i form an orthonormal basis, that last sum is just $\|w' - w\|^2$: we're taking the coordinates in the basis corresponding to the eigenvectors, and summing

them up. Another way to say that is that for any x,

$$\sum_i \langle x, v_i\rangle^2 = \sum_i \langle x, v_i\rangle\langle v_i, x\rangle = \sum_i \langle x, v_i v_i^\mathsf{T} x\rangle = \left\langle x, \left(\sum_i v_i v_i^\mathsf{T}\right)x\right\rangle = \langle x, x\rangle = \|x\|^2,$$

where $\sum_i v_i v_i^\mathsf{T} = \mathrm{I}$ holds for any orthonormal basis: $\sum_i v_i v_i^\mathsf{T} y = y$ is basically definitional. The desired result follows by Proposition 11.2. □

Lemma 11.4 (Descent lemma)**.** *Let $w^+ = w - \eta\nabla f(w)$ for a β-smooth function f, where $0 < \eta < 2/\beta$. Then*

$$f(w) - f(w^+) \geq \eta\left(1 - \frac{1}{2}\eta\beta\right)\|\nabla f(w)\|^2,$$

and hence either $\nabla f(w) = 0$ or $f(w^+) < f(w)$.

Proof. By Proposition 11.3, we have

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

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

So, this means that gradient descent with a small-enough learning rate is a "descent method": each step decreases the objective, unless $\nabla f(w_t) = 0$ for some t, in which case $w_{t+1} = w_t$ and we're stuck forever. If we never hit such a point, then $f(w_t)$ must strictly decrease forever.

Suppose $\inf_w f(w) \geq a$ for some finite a, for example L_S with a bounded loss. Then we necessarily have $f(w_t) \to f_\infty \geq a$, called the monotone convergence theorem. If so, then since $f(w_t) - f(w_{t+1}) \to 0$, the descent lemma implies $\|\nabla f(w)\| \to 0$; we either converge to a stationary point, or else our iterates diverge in a way where the loss still converges. The latter case happens e.g. for logistic regression with separable data: no finite w achives zero logistic loss, but we can get closer and closer to zero loss by letting $\|w_t\| \to \infty$ but $w_t/\|w_t\| \to w^*$ where w^* achieves zero training errors (0-1 loss).

For convex functions, any stationary point – one with $\nabla f(w) = 0$ – is a global min. This is why convex optimization is nice! 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.)

11.3 ASIDE: CONVEX FUNCTIONS

For convex functions in particular (with a slightly smaller learning rate), we can turn the descent lemma into a proof of gradient descent convergence.

Theorem 11.5. *Let f be a convex, β-smooth function. Begin with w_0 and then let $w_{t+1} = w_t - \eta\nabla f(w_t)$, for some $0 < \eta \leq 1/\beta$. Let $\bar{w}_s = \frac{1}{\mathrm{T}-s+1}\sum_{t=\mathrm{T}-s+1}^{\mathrm{T}} w_t$ be the average*

of the last s iterates; particular examples include $\bar{w}_1 = w_T$ and $\bar{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$. For any $1 \le s \le T$, it holds that

$$f(\bar{w}_s) \le \frac{1}{2\eta T}\|w_0 - w^*\|.$$

Proof. We're going to want to bound the improvement in each step of gradient descent: $f(w_t) - f(w_{t+1})$. By convexity (in particular, Proposition C.3), we have that

$$f(w_t) - f(w^*) \le \langle w_t - w^*, \nabla f(w_t)\rangle.$$

To see what that right-hand side is, we'll use

$$\|w_t - \eta\nabla f(w_t) - w^*\|^2 = \|w_t - w^*\|^2 - 2\eta\langle w_t - w^*, \nabla f(w_t)\rangle + \eta^2\|\nabla f(w_t)\|^2,$$

and rearrange into

$$f(w_t) - f(w^*) \le \langle w_t - w^*, \nabla f(w_t)\rangle = \frac{1}{2\eta}\left[\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2\right] + \frac{\eta}{2}\|\nabla f(w_t)\|^2.$$

Now we can use the descent lemma to deal with that last term: since $\eta \le 1/\beta$, $1 - \frac{1}{2}\eta\beta \ge \frac{1}{2}$, and Lemma 11.4 becomes

$$f(w_t) - f(w_{t+1}) \ge \frac{1}{2}\eta\|\nabla f(w)\|^2.$$

We've therefore obtained

$$f(w_t) - f(w_*) \le \frac{1}{2\eta}\left[\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2\right] + f(w_t) - f(w_{t+1}),$$

which we can simplify to

$$f(w_{t+1}) - f(w^*) \le \frac{1}{2\eta}\left[\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2\right]$$

or equivalently

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

This holds for each t from 1 to T; let's the take the average of them all, obtaining

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

Notice that by the descent lemma, $f(w_t)$ is weakly decreasing; thus

$$\frac{1}{T - s + 1}\sum_{t=T-s+1}^{T} f(w_t) \le \frac{1}{T}\sum_{t=1}^{T} f(w_t) \le f(w^*) + \frac{1}{2\eta T}\|w_0 - w^*\|^2.$$

But by Jensen's inequality, since f is convex,

$$f(\bar{w}_s) = f\left(\frac{1}{T-s+1}\sum_{t=T-s+1}^{T} w_t\right) \le \frac{1}{T-s+1}\sum_{t=T-s+1}^{T} f(w_t). \qquad \square$$

Much faster rates are available if f is smooth and *strongly* convex.

11.3.1 *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 11.6 (Corollary 1 of [KR23]). *Let f be β-smooth, with $\inf_x f(x) \ge f^{\inf} > -\infty$. Let $\hat{g}_t \mid x_t$ be independent such that $\mathbb{E}[\hat{g}_t \mid x_t] = \nabla f(x_t)$ and*

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

for some $A, B, C \ge 0$. 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_0, with $\delta_0 = f(x_0) - f^{\inf}$, and $x_{t+1} = x_t - \eta\hat{g}_t$. As long as $T \ge \frac{12\delta_0\beta}{\varepsilon^2}\max\left\{B, \frac{12\delta_0 A}{\varepsilon^2}, \frac{2C}{\varepsilon^2}\right\}$, it holds that $\min_{1\le t\le T}\mathbb{E}[\|\nabla f(x_t)\|] \le \varepsilon$.

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 choose $\hat{g}_t$ as the gradient of $L_{S'}$ for $S' \subseteq S$ chosen randomly and the loss being Lipschitz, or various other settings.

11.4 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

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

and we have

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

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

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. (All of this also requires that σ itself be β-smooth; ReLU is not.)

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.

11.5 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.

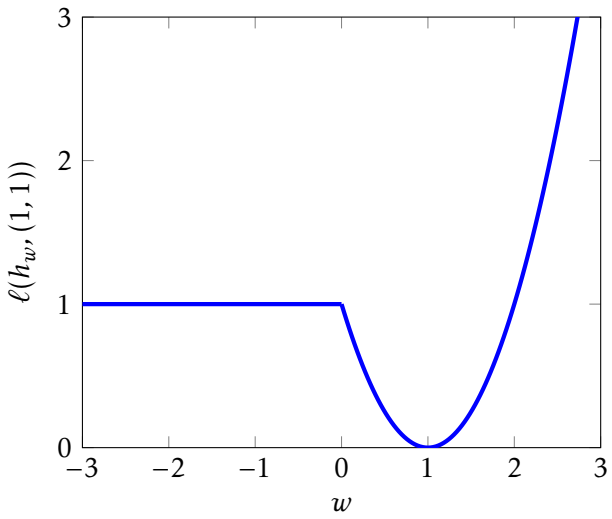
It turns out that for deep linear nets:

- 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 $h(x) = \mathrm{ReLU}(wx)$, 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 (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? Moreover, even if I find a good local min of L_S, does that imply I get a good L_D for realistic networks?

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.