CPSC 532D — 15. IMPLICIT REGULARIZATION AND MARGINS

Danica J. Sutherland
University of British Columbia, Vancouver
Fall 2024

We've seen some examples so far of settings where there's more than one empirical risk minimizer; this often happens with *interpolation*, when you can achieve $L_S(h) = 0$ in more than one way, some of which are awful, but $\mathcal{A}$ often picks decent ones. In particular, we saw some explicit examples with polynomial regression.

One way to choose between ERMs (or near-ERMs) is regularized loss minimization, where we prefer solutions with e.g. a small norm. But often we don't do that, and we just run gradient descent to minimize $L_S(h)$. Doing this doesn't just get us *any* arbitrary ERM; it gets us a particular one, decided on by our choice of algorithm. The idea that our optimization algorithm or other such "implementation details" can actually choose for us which of the "equally valid solutions" we end up with is called the *implicit regularization* of the algorithm: we don't explicitly write down a regularizer, but the choice of algorithm has a similar effect.

It's also sometimes called the implicit bias *of the algorithm, in the sense that the algorithm has a certain inductive bias towards certain kinds of solutions. That can sometimes cause confusion with the concept of the same name from social science, though, and just generally kind of imply that it's "bad" when actually often the presence of this implicit regularization is "good."*

In our discussion of neural tangent kernels, we mentioned that we could solve the ODE for gradient flow to say *which* ERM we end up at in (14.8). We didn't prove this, though, and it only applied to "kernel gradient flow" which is not really the algorithm we usually use. What happens for actual problems, with finite learning rates?

15.1 GRADIENT DESCENT FOR LINEAR REGRESSION

Let's think about optimizing the function

$$f(w) = L_S^{sq}(x \mapsto w \cdot x) = \frac{1}{m} \left\| Xw - y \right\|^2,$$

where $X \in \mathbb{R}^{m \times d}$ is the matrix stacking up S_x and $y \in \mathbb{R}^m$ is the vector form of S_y.

It's possible to use this form to handle kernels, too. If there's a finite-dimensional embedding ϕ, we could just collect $\phi(x_i)$ in rows of X and find w. If we instead write $f_\alpha(x) = \sum_i \alpha_i k(x_i, x)$ and do gradient descent on α, notice the training set loss becomes $L_S(f_\alpha) = \frac{1}{m} \left\| K\alpha - y \right\|^2$) and so the rest of the analysis will apply with $X = K$ – which will potentially give a *different* solution than the kernel gradient descent version. Implicit regularization is highly algorithm-specific.

This agrees with "kernel gradient descent" as in Chapter 14 for finite-dimensional kernels.

In any case, we have

$$\nabla f(w) = \frac{2}{m} X^T (Xw - y),$$

which notice is $\frac{2}{m} \left\| X^T X \right\|$-smooth, so f is convex and β-smooth, thus small-learning-rate gradient descent finds a global optimum (Proposition 13.6). In the traditional $m > d$ case when X is full-rank, there's a unique solution to this problem, typically with $Xw \neq y$ but always having $X^T(Xw - y) = 0$. In high-dimensional settings $d > m$,

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

though, it's possible to achieve $Xw = y$ (interpolation) in infinitely many ways. Which one does gradient descent find?

There's a more explicit (but longer) analysis for least squares, which gives some more details without relying on any general gradient descent analyses, in last year's notes.

Proposition 15.1. *Let $X \in \mathbb{R}^{m\times d}$ be of rank m (implying $d \geq m$), and $y \in \mathbb{R}^m$. Suppose that $l(h,(x,y)) = l_y(h(x))$ for a differentiable function l_y such that $l_y(\hat{y}) \to 0$ implies $\hat{y} \to y$.*

Consider any iterative optimization method which begins at a point w_0 and then has updates of the form $w_{t+1} - w_t \in \operatorname{span}\{\nabla L_S(x \mapsto w_k \cdot x) : 0 \leq k \leq t\}$. If this method converges to a global minimizer w_∞ of $L_S(x \mapsto w \cdot x)$, then

$$w_\infty = X^\top (XX^\top)^{-1} y + (I - X^\top (XX^\top)^{-1} X) w_0 = \operatorname*{arg\,min}_{w: Xw = y} \|w - w_0\|.$$

Proof. $XX^\top$ is $m \times m$ of rank m, and so $(XX^\top)^{-1}$ exists; then $X(X^\top (XX^\top)^{-1} y) = y$. The matrix $X^\top (XX^\top)^{-1}$ is the pseudoinverse of X, written $X^\dagger$; this then implies that $L_S(x \mapsto (X^\dagger y) \cdot x) = 0$. Since w_∞ is optimal, we must have $Xw_\infty = y$.

Now, for any w we have that

$$\nabla_w L_S(h_w) = \frac{1}{m} \sum_{i=1}^m l'_{y_i}(w \cdot x_i) x_i \in \operatorname{span}\{x_i : i \in [m]\}.$$

For general SGD with unbiased gradients, it will be true for the expected *update, even if not for the actual update.*

This is true for each step, no matter the learning rate; it is also true e.g. if we do stochastic gradient descent based on choosing a subset of the data at each step. Thus the iterates of gradient descent must all be of the form

$$w_t = w_0 + \sum_i \alpha_i^{(t)} x_i = w_0 + X^\top \alpha^{(t)};$$

they can only ever move in the subspace spanned by the data, and otherwise stay where they started. Thus, this must also be true for the limiting point: $w_\infty = w_0 + X^\top \alpha$ for some $\alpha \in \mathbb{R}^m$.

Thus, we know that

$$X(w_0 + X^\top \alpha) = y$$

or equivalently

$$XX^\top \alpha = y - Xw_0.$$

Since we know already that $XX^\top$ is invertible, we have that

$$\begin{aligned} \alpha &= (XX^\top)^{-1}(y - Xw_0) \\ w_\infty &= w_0 + X^\top (XX^\top)^{-1}(y - Xw_0) \\ &= X^\top (XX^\top)^{-1} y + (I - X^\top (XX^\top)^{-1} X) w_0. \end{aligned}$$

This second matrix is the orthogonal projection onto the null space of X, which can be seen e.g. by considering the SVD. The result follows by Lemma 15.2. □

Aside: closest interpolator

Lemma 15.2. *Let $X \in \mathbb{R}^{m\times d}$, $y \in \mathbb{R}^m$, and let $\Pi_\perp$ be the orthogonal projection onto the null space of X. Then*

$$\operatorname*{arg\,min}_{w: Xw = y} \|w - w_0\| = X^\dagger y + \Pi_\perp w_0.$$

Proof. First, the set of possible interpolators must all have $y = Xw$, hence $X^\dagger y = X^\dagger X w$. $X^\dagger X$ is exactly the orthogonal projection onto the row space of X: letting the compact SVD of X be $U\Sigma V^\top$, $X^\dagger = V\Sigma^{-1}U^\top$, and $X^\dagger X = V\Sigma^{-1}U^\top U\Sigma V^\top = V\Sigma^{-1}\Sigma V^\top = VV^\top$, which has $(VV^\top)^\top = VV^\top$ and $(VV^\top)^2 = VV^\top VV^\top = VV^\top$. Notice also that $\Pi_\perp = I - VV^\top$. Thus, $X^\dagger y = VV^\top w$ for any interpolator, and so the set of interpolators is the set $\{X^\dagger y + q : VV^\top q = 0\}$. For any such solution,

$$\begin{aligned}\left\|X^\dagger y + q - w_0\right\|^2 &= \left\|VV^\top(X^\dagger y + q - w_0)\right\|^2 + \left\|(I - VV^\top)(X^\dagger y + q - w_0)\right\|^2 \\ &= \left\|X^\dagger y - VV^\top w_1\right\|^2 + \left\|q - (I - VV^\top)w_0\right\|^2.\end{aligned}$$

The choice of q does not affect the first term, while the second is uniquely minimized by $q = (I - VV^\top)w_0$. □

15.2 SEPARABLE LOGISTIC REGRESSION

There's another major class of loss functions not satisfying the requirement of Proposition 15.1: for instance, with logistic loss $l_y(\hat{y}) = \log(1 + \exp(-y\hat{y}))$, $l_y(\hat{y}) \to 0$ implies $\hat{y} \to y\infty$, not y.

So, let's consider logistic regression in particular: for $y_i \in \{-1, 1\}$,

$$f(w) = \frac{1}{m}\sum_{i=1}^m \log(1 + \exp(-y_i x_i^\top w)).$$

We're also going to assume that the data is *linearly separable*: there is some w^* such that $y_i x_i^\top w^* > 0$ for all i. Then, it's possible to drive $f(w)$ arbitrarily close to zero, but never to actually reach it: we only get $\log(1 + \exp(-t)) \to 0$ for $t \to \infty$, so we need $\|w\| \to \infty$. A solution of the form cw^* for $c \to \infty$ would work, but potentially so would many other solutions, since there are probably many possible perfect linear separators on this dataset. Which one does gradient descent find?

We're going to approach this informally, for time and simplicity. Soudry et al. [Sou+18] and Gunasekar et al. [GLSS18] handle it in full, and Ji and Telgarsky [JT19] approach the non-separable case; Bach [Bach24, Section 11.1.2] gives an overview including a few things we aren't covering here.

Notice that

$$\nabla f(w) = -\frac{1}{m}\sum_{i=1}^m \frac{\exp(-y_i x_i^\top w)}{1 + \exp(-y_i x_i^\top w)} y_i x_i.$$

We know that we'll get $\|w_t\| \to \infty$ from the argument above; it's reasonable to expect, then, that we'll have $\frac{w_t}{\|w_t\|} \to v$ for some $\|v\| = 1$, and $y_i x_i^\top v > 0$ for all i since otherwise we wouldn't approach a minimizer. This gives us, roughly speaking,

$$\nabla f(\|w_t\| v) \sim -\frac{1}{m}\sum_{i=1}^m \frac{\exp(-y_i \|w_t\| x_i^\top v)}{1 + \exp(-y_i \|w_t\| x_i^\top v)} y_i x_i \sim -\frac{1}{m}\sum_{i=1}^m \exp(-y_i \|w_t\| x_i^\top v) y_i x_i,$$

since $\frac{t}{1+t} = t + \mathcal{O}(t^2)$ and we'll eventually have $\exp(-y_i \|w_t\| x_i^\top v) \ll 1$.

So, eventually each gradient term gets small. Which ones are bigger than the others? The asymptotic ratio between the size of the gradient contributions from x_i and x_j is

$$\frac{\exp(-y_i \|w_t\| x_i^\top v)\left|y_i\right| \|x_i\|}{\exp(-y_j \|w_t\| x_j^\top v)\left|y_j\right| \left\|x_j\right\|} = \frac{\|x_i\|}{\left\|x_j\right\|} \exp\left(-\|w_t\| (y_i x_i^\top v - y_j x_j^\top v)\right).$$

As $\|w_t\| \to \infty$, this ratio goes to 0 if $y_i x_i^\top v > y_j x_j^\top v$, or ∞ if the order is reversed; it is $\|x_i\| / \|x_j\| \in (0, \infty)$ if and only if $y_i x_i^\top v = y_j x_j^\top v$. So, for whatever v we have, let $\mathcal{I}_v$ be the set of indices such that $y_i x_i^\top v$ is minimized. Only these terms really matter:

$$\nabla f(\|w_t\| v) \sim -\frac{1}{m} \sum_{i \in \mathcal{I}_v} \exp(-y_i \|w_t\| x_i^\top v) y_i x_i.$$

So, if gradient descent diverges in a direction v, the dominant direction in which w_t moves is a (positive) linear combination of the points $\{x_i : i \in \mathcal{I}_v\}$. Let's define $\rho = \min_i y_i x_i^\top v$; then, summarizing,

$$v = \sum_{i=1}^{m} \alpha_i y_i x_i \quad \text{with } \forall i,\ (\alpha_i \geq 0 \text{ and } y_i x_i^\top v = \rho) \text{ or } (\alpha_i = 0 \text{ and } y_i x_i^\top v > \rho). \quad (15.1)$$

In fact, ρ is a quantity known as the *geometric margin* of the linear separator v; it is exactly the smallest distance from any of the x_i to the hyperplane $\{x : v^\top x = 0\}$, the decision boundary of the linear classifier with weights v.

15.2.1 *Margin maximization*

The equations (15.1) turn out to be equivalent to the KKT conditions of the problem of finding the *max-margin separator*, also known as a hard support vector machine (SVM). This problem is given by

$$\arg\max_{v:\|v\|=1} \min_{i \in [m]} y_i x_i \cdot v \quad \text{s.t. } \forall i \in [m],\ y_i x_i \cdot v > 0$$

Change so that $v = w/\|w\|$ for any w:

$$= \arg\max_{w \in \mathbb{R}^d} \min_{i \in [m]} \frac{y_i x_i \cdot w}{\|w\|} \quad \text{s.t. } \forall i \in [m],\ y_i x_i \cdot w > 0$$

$$= \arg\max_{w \in \mathbb{R}^d} \frac{1}{\|w\|} \min_{i \in [m]} y_i x_i \cdot w \quad \text{s.t. } \forall i \in [m],\ y_i x_i \cdot w > 0$$

The objective is the same for any $w' = cw$ for $c > 0$, so we might as well limit ourselves to solutions where $\min_i y_i x_i \cdot w = 1$:

$$\supseteq \arg\max_{w \in \mathbb{R}^d} \frac{1}{\|w\|} \quad \text{s.t. } \forall i \in [m],\ y_i x_i \cdot w \geq 1$$

$$= \arg\min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 \quad \text{s.t. } \forall i \in [m],\ y_i x_i \cdot w \geq 1. \quad (15.2)$$

Using the definition of the KKT conditions on this problem and rearranging a bit yields (15.1). But, here's a direct argument without appealing to the KKT conditions.

First, the solution to (15.2) is unique: the objective is strictly convex, and the constraints are affine and by assumption feasible.

These constraints will be "active" exactly for the indices $\mathcal{I}_v$; other points will have larger values of $y_i x_i \cdot w$. But if w included some component $w_\perp$ such that $x_i \cdot w_\perp = 0$ for all $i \in \mathcal{I}_v$, then this wouldn't affect the active constraints at all, and (similarly to the representer theorem and Lemma 15.2) would only make the objective $\|w\|^2$ bigger. So solutions to (15.2) must have $w = \sum_{i \in \mathcal{I}_v} \alpha_i y_i x_i$ for some coefficients α.

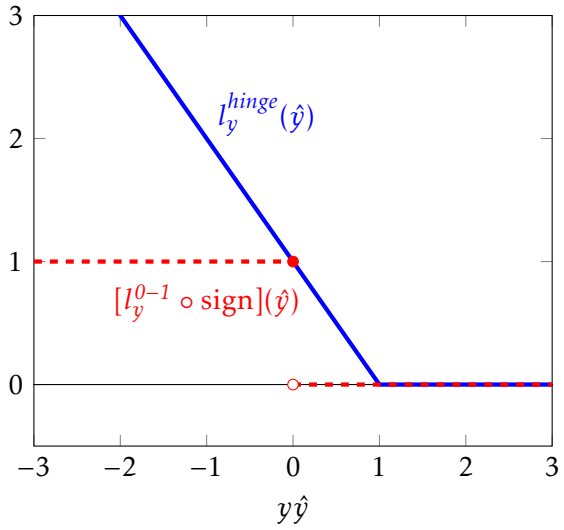
The solution also have that each $\alpha_i \geq 0$: suppose WLOG that $\alpha_1 < 0$; we'll see that setting $\alpha_1 = 0$ to zero would strictly improve the optimization. We have $w \cdot y_j x_j = \alpha_1 y_1 x_1 \cdot y_j x_j + \sum_{i>1} \alpha_i y_i x_i \cdot y_j x_j$, so the only way the margin can be improved by a negative α_1 for any point x_j is if $y_1 y_j \, x_1 \cdot x_j < 0$. But if the data is linearly separable, this is impossible: $\operatorname{sign}(w \cdot x_i) = y_i$ for all i, but if $x_1 \cdot x_j < 0$ then there is no w such that $\operatorname{sign}(x_1 \cdot w) = \operatorname{sign}(x_j \cdot w) \neq 0$. Thus $\alpha_1 < 0$ does not help satisfy any of the constraints. It also does not help with the objective; $\left\|\alpha_1 y_1 x_1 - v\right\|^2 = \alpha_1^2 \|x_1\|^2 + \|v\|^2 - 2\alpha_1 y_1 x_1 \cdot v$, but we just established that $\operatorname{sign}(x_1 \cdot v) = y_1$, so $\alpha_1 < 0$ only hurts the objective.

This establishes that the solution to (15.2) must satisfy (15.1). In the other direction: I don't know a concise formal proof without appealing to optimization theory we haven't covered, but geometrically, for suboptimal values of ρ there could be many solutions to (15.1). For the maximal value, however, for points in general position there will only be a single v satisfying these properties.

15.2.2 *Hinge loss interpolation*

The *hinge loss* is given by

$$l_y^{hinge}(\hat{y}) = \begin{cases} 1 - y\hat{y} & \text{if } y\hat{y} \leq 1 \\ 0 & \text{if } y\hat{y} \geq 1. \end{cases}$$



Notice that if $\mathrm{L}_S^{hinge}(x \mapsto w \cdot x) = 0$, then for all $i \in [m]$, $y_i x_i \cdot w \geq 1$. Thus (15.2) is equivalent to

$$\operatorname*{arg\,min}_{w : \mathrm{L}_S^{hinge}(x \mapsto w \cdot x) = 0} \|w\|,$$

the *minimum-norm hinge loss interpolator*. This is kind of a nice analogy to how gradient descent for least squares or similar losses (starting at $w_0 = 0$) finds the minimum-norm interpolator for that loss! But, interestingly, explicitly minimizing logistic loss (with gradient descent) implicitly minimizes hinge loss.

Transforming the hard constraint into a soft one gives us a soft support vector machine,

$$\operatorname*{arg\,min}_h \mathrm{L}_S^{hinge}(h) + \lambda \|h\|^2.$$

15.2.3 *Margin analysis*

How can we think about the 0-1 generalization error of the max-margin predictor?

In dimension d, one option is to use that the VC dimension is either d or $d+1$, depending on if we put an intercept in. But when d is high, e.g. $d > m$, this doesn't really tell us anything.

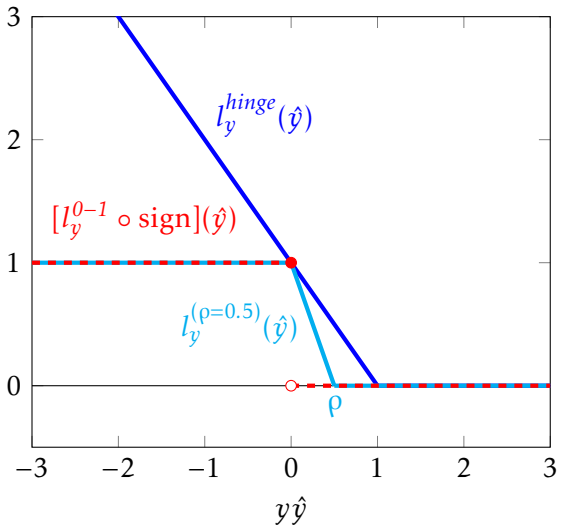
We're finding the minimum-norm interpolator, though, so maybe we can use a Rademacher bound that exploits that the norm isn't too big. So, let's think about $\mathcal{H}_B = \{h \in \mathcal{F} : \|h\| \le B\}$ for some RKHS $\mathcal{F}$, potentially the linear kernel in dimension d but potentially not. We know that $\mathbb{E}_S \operatorname{Rad}(\mathcal{H}_B|_{S_x}) \le \frac{B}{\sqrt{m}}\sqrt{\mathbb{E}\, k(x,x)}$. To use this for a generalization bound on the 0-1 loss, though, we need to convert these soft predictions into hard ones with the sign function, so that the estimation error is bounded in terms of $\operatorname{Rad}((\ell_{0-1} \circ \operatorname{sign} \circ \mathcal{H}_B)|_S)$. But $\ell_{0-1} \circ \operatorname{sign}$ *isn't Lipschitz*; it jumps suddenly from 0 to 1 as the sign of the predictor changes. So we can't use Talagrand's lemma to peel it off at all.

(When deriving VC dimension, we pretended the 0-1 loss was Lipschitz, but that only worked because we were working with a hypothesis class mapping to ± 1. There's no similar trick we can play with continuous-output $\mathcal{H}$.)

We can work around this problem with *surrogate losses*. The hinge loss, above, is a good example: $\ell_{0-1}(h,z) \le \ell_{hinge}(h,z)$ for any inputs, so necessarily $\mathrm{L}_{\mathcal{D}}^{0-1}(h) \le \mathrm{L}_{\mathcal{D}}^{hinge}(h)$, and any generalization bound that applies to hinge loss also applies to 0-1 loss.

We can also use a tighter surrogate, though. One choice is *margin loss*:

$$l_y^\rho(\hat{y}) = \begin{cases} 1 & \text{if } y\hat{y} \le 0 \\ 1 - \frac{1}{\rho} y\hat{y} & \text{if } 0 \le y\hat{y} \le \rho \\ 0 & \text{if } y\hat{y} \ge \rho. \end{cases}$$



This is $1/\rho$-Lipschitz, bounded in $[0,1]$, and always an upper bound to the 0-1 loss. If $\min_i y_i h(x_i) \ge \rho$, then $\mathrm{L}_S^\rho(h) = 0$. We get an immediate result:

$$\mathrm{L}_{\mathcal{D}}^{0-1}(\operatorname{sign} \circ h) \le \mathrm{L}_{\mathcal{D}}^{\rho}(h) \le \mathrm{L}_S^{\rho}(h) + \frac{2}{\rho} \mathop{\mathbb{E}}_{S} \operatorname{Rad}(\mathcal{H}|_{S_x}) + \sqrt{\frac{1}{2m} \log \frac{1}{\delta}} \tag{15.3}$$

if $h \in \mathcal{H}$ and we picked ρ independently of S and h.

We can also do a nonuniform analysis to avoid committing in advance to a particular margin ρ, exactly like what we did for SRM:

Proposition 15.3. *Let $\mathcal{H}$ contain functions mapping to $\mathbb{R}$, and fix some $r > 0$. Then for any $\delta \in (0,1)$, we have with probability at least $1-\delta$ over the choice of $S \sim \mathcal{D}^m$ that it holds for all $h \in \mathcal{H}$ and $\rho \in (0, r]$ that*

$$L_{\mathcal{D}}^{0-1}(\operatorname{sign} \circ h) \le L_S^{\rho}(h) + \frac{4}{\rho} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^m} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{m} \log \log_2 \frac{2r}{\rho}} + \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}.$$

Proof. Let $\rho_k = r2^{-k}$ for all $k \ge 0$, and $\delta_k = \frac{6\delta}{\pi^2 k^2}$ for $k \ge 1$; note that $\sum_{k=1}^{\infty} \delta_k = \delta$. By (15.3), it holds with probability at least $1 - \delta_k$ for each ρ_k that

$$\forall h \in \mathcal{H}, \quad L_{\mathcal{D}}^{0-1}(\operatorname{sign} \circ h) \le L_S^{\rho_k - margin}(h) + \frac{2}{\rho_k} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^m} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{2m} \log \frac{1}{\delta_k}}.$$

For any $\rho \in (0, r]$, the smallest k such that $\rho_k \le \rho$ is given by $k = \left\lceil \log_2 \frac{r}{\rho} \right\rceil$.

We have $\ell_{\rho'} \le \ell_{\rho}$ for any $\rho' \le \rho$, so $L_S^{\rho_k}(h) \le L_S^{\rho}(h)$.

We also know that $\rho \le \rho_{k-1} = 2\rho_k$, so $\frac{1}{\rho_k} \le \frac{2}{\rho}$.

Finally, from $\log \frac{1}{\delta_k} = \log \frac{\pi^2}{6\delta} + 2 \log \log_2 \left\lceil \log_2 \frac{r}{\rho} \right\rceil$ we use that $\pi^2/6 < 2$ and $\lceil \log_2 a \rceil < \log_2(a) + 1 = \log_2(2a)$. □

We do have to commit to some predefined upper bound on the margin r, but the resulting bound only depends on it through $\sqrt{\log \log_2 r}$, so we can pick something big.

15.3 OTHER MODELS/ALGORITHMS

Lyu and Li [LL20] and Ji and Telgarsky [JT20] study small-learning-rate gradient descent on L-*homogeneous* networks, those satisfying $h(x; \alpha w) = \alpha^L h(x; w)$ for $\alpha > 0$; this is true e.g. for (leaky)-ReLU networks. (We'll describe the [LL20] results.) Their analysis is in terms of the *normalized margin*

$$\bar{\gamma}(w) = \frac{\min_{i \in [m]} y_i h(x_i; w)}{\|w\|_2^L}.$$

This normalization is exactly the one that makes $\bar{\gamma}(\alpha w) = \bar{\gamma}(w)$. They show, using an approach like that of Section 15.2, that gradient flow or small-learning-rate gradient descent (under some additional regularity conditions) monotonically increase the log-sum-exp version of normalized margin, which means they approximately monotonically increase the normalized margin, which roughly means that it finds a local maximum (ish) of the normalized margin.

They talk about convergence to a "KKT point"; this is using the version of the KKT conditions where stationarity is defined by gradients, not subgradients, and hence isn't sufficient for optimality in nonconvex problems.

This is a kind of margin maximization, and Proposition 15.3 applies, but in general it's *not* margin maximization in an RKHS. Compare this to training a very wide network with square loss, in which case the implicit regularization prefers solutions with minimal NTK norm distance from the initialization. Knowing these results, you can ask questions like what this margin maximization actually does on particular models [e.g. Fre+23].

There's been a bunch of recent work trying to figure out the implicit regularization of Adam, rather than SGD, on homogeneous networks; some recent papers are [WMCL21; Wan+22; CKS23; XL24].

There's also a *ton* more work in this area; Vardi [Var22] gives a (now kind of outdated) survey.

REFERENCES

[Bach24] Francis Bach. *Learning Theory from First Principles*. Draft version. August 2024.

[CKS23] Matias D. Cattaneo, Jason M. Klusowski, and Boris Shigida. *On the Implicit Bias of Adam*. 2023. arXiv: 2309.00079.

[Fre+23] Spencer Frei, Gal Vardi, Peter L. Bartlett, Nathan Srebro, and Wei Hu. "Implicit Bias in Leaky ReLU Networks Trained on High-Dimensional Data". *ICLR*. 2023. arXiv: 2210.07082.

[GLSS18] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. "Characterizing Implicit Bias in Terms of Optimization Geometry". *ICML*. 2018. arXiv: 1802.08246.

[JT19] Ziwei Ji and Matus Telgarsky. "The implicit bias of gradient descent on nonseparable data". *COLT*. 2019. arXiv: 1803.07300.

[JT20] Ziwei Ji and Matus Telgarsky. "Directional convergence and alignment in deep learning". *NeurIPS*. 2020. arXiv: 2006.06657.

[LL20] Kaifeng Lyu and Jian Li. "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks". *ICLR*. 2020. arXiv: 1906.05890.

[Sou+18] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The Implicit Bias of Gradient Descent on Separable Data. *JMLR* (2018). arXiv: 1710.10345.

[Var22] Gal Vardi. *On the Implicit Bias in Deep-Learning Algorithms*. 2022. arXiv: 2208.12591.

[Wan+22] Bohan Wang, Qi Meng, Huishuai Zhang, Ruoyu Sun, Wei Chen, Zhi-Ming Ma, and Tie-Yan Liu. "Does Momentum Change the Implicit Regularization on Separable Data?" *NeurIPS*. 2022. arXiv: 2110.03891 [cs.LG].

[WMCL21] Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. "The Implicit Bias for Adaptive Optimization Algorithms on Homogeneous Neural Networks". *ICML*. 2021. arXiv: 2012.06244.

[XL24] Shuo Xie and Zhiyuan Li. "Implicit Bias of AdamW: ℓ_∞ Norm Constrained Optimization". *ICML*. 2024. arXiv: 2404.04454.