Deep learning theory lecture notes

Matus Telgarsky mjt@illinois.edu

2021-10-27 v0.0-e7150f2d (alpha)

Contents

Preface

Philosophy of these notes. Two key ideas determined what has been included so far.

1. I aim to provide simplified proofs over what appears in the literature, ideally reducing difficult things to something that fits in a single lecture.
2. I have primarily focused on a classical perspective of achieving a low test error for binary classification with IID data via standard (typically ReLU) feedforward networks.

Organization. Following the second point above, the classical view decomposes the test error into three parts.

1. **Approximation (starts in section 1):** given a classification problem, there exists a deep network which achieves low error *over the distribution*.
2. **Optimization (starts in section 6):** given a finite training set for a classification problem, there exist algorithms to find predictors with low training error *and low complexity*.
3. **Generalization (starts in section 11):** the gap between training and testing error is small for low complexity networks.

Remark 0.1 ***(weaknesses of this "classical" approach)***

- Recent influential work suggests that the classical perspective is hopelessly loose, and has poor explanatory power (Neyshabur, Tomioka, and Srebro 2014; Zhang et al. 2017). Follow-ups highlight this looseness and its lack of correlation with good test error performance (Dziugaite and Roy 2017), and even suggest the basic approach is flawed (Nagarajan and Kolter 2019); please see section 11.1 for further discussion and references.
- The reasons for keeping with this approach here are as follows:

1. It appears that all of these negative results consider the consequences of *worst-case* behavior in one of these three terms on the other two. Here instead we study how they inter-connect in a favorable way. A common them is how they all work together with *low complexity models* on reasonable data.
2. Even if the preceding point is overly optimistic at times, this decomposition still gives us a way to organize and categorize much of what is known in the field, and secondly these ideas will always be useful at least as tools in a broader picture.

Formatting.

- These notes use pandoc markdown with various extensions. A current html version is always at https://mjt.cs.illinois.edu/dlt/, and a current pdf version is always at https://mjt.cs.illinois.edu/dlt/index.pdf.
- Owing to my unfamiliarity with pandoc, there are still various formatting bugs.
- [mjt☹: Various todo notes are marked throughout the text like this.]

Feedback. I'm very eager to hear any and all feedback!

How to cite. Please consider using a format which makes the version clear:

```
@misc{mjt_dlt,
       author = {Matus Telgarsky},
        title = {Deep learning theory lecture notes},
 howpublished = {\url{https://mjt.cs.illinois.edu/dlt/}},
         year = {2021},
         note = {Version: 2021-10-27 v0.0-e7150f2d (alpha)},
}
```

Basic setup: feedforward networks and test error decomposition

In his section we outline our basic setup, which can be summarized as follows:

1. We consider standard shallow and deep feedforward networks.
2. We study mainly binary classification in the supervised learning setup.
3. As above, we study an error decomposition into three parts.

Although this means we exclude many settings, as discussed above, much of the work in other settings uses tools from this most standard one.

Basic shallow network. Consider the mapping

$$x \mapsto \sum_{j=1}^{m} a_j \sigma(w_j^{\mathsf{T}} x + b_j).$$

- σ is the *nonlinearity/activation/transfer*. Typical choices: ReLU $z \mapsto \max\{0, z\}$, sigmoid $z \mapsto \frac{1}{1+\exp(-z)}$.
- $((a_j, w_j, b_j))_{j=1}^m$ are *trainable parameters*; varying them defines the function class. Sometimes in this shallow setting we freeze $(a_j)_{j=1}^m$, which gives a simple model that is still difficult to analyze (e.g., nonconvex).
- We can think of this as a directed graph of *width* m: we have a *hidden layer* of m nodes, where the jth computes $x \mapsto \sigma(w_j^{\mathsf{T}} x + b_j)$.
- Define *weight* matrix $W \in \mathbb{R}^{m \times d}$ and *bias* vector $v \in \mathbb{R}^m$ as $W_{j:} = w_j^{\mathsf{T}}$ and $v_j := b_j$. The first *layer* computes $h := \sigma(Wx + b) \in \mathbb{R}^m$ (σ applied coordinate-wise), the second computes $h \mapsto a^{\mathsf{T}} h$.

Basic deep network. Extending the matrix notation, given parameters $w = (W_1, b_1, \ldots, W_L, b_L)$,

$$f(x; w) := \sigma_L(W_L \sigma_{L-1}(\cdots W_2 \sigma_1(W_1 x + b_1) + b_2 \cdots) + b_L). \tag{1}$$

- σ_j is now a multivariate mapping; in addition to coordinate-wise ReLU and sigmoid, we can do *softmax* $z' \propto \exp(z)$, max-pooling (a few coordinates of input replaced with their maximum), attention layers, and many others.
- We can replace $x \mapsto Wx + b$ with some compact representation while still preservering linearity, e.g., the standard implementation of a convolution layer. [mjt☹: Maybe I will add the explicit formalisms somewhere?].

- Often biases $(b_1, \ldots, b_L)$ are dropped; the handling of these biases can change many elements of the story.
- Typically σ_L is identity, so we refer to L as the number of affine layers, and $L-1$ the number of activation or hidden layers.
- Width now means the maximum output dimension of each activation. (For technical reasons, sometimes need to also take max or input dimension, or treat inputs as a fake layer.)
- Once again we can describe the computation via an acyclic graph. Classically, the activations were univariate mappings applied coordinate-wise, and single rows of the weight matrix were composed with univariate activations to give a *node*. Now, however, activations are often multivariate (and in particular can not be written as identical univariate mappins, applied coordinate-wise), and for computation reasons we prefer not to break the matrices into vectors, giving a more general graph with each matrix and activation as its own node.

Basic supervised learning setup; test error decomposition.

- Given pairs $((x_i, y_i))_{i=1}^n$ (training set), our job is to produce a mapping $x \mapsto y$ which performs well on future examples.
- If there is no relationship between past and future data, we can't hope for much.
- The standard classical learning assumption is that both the training set, and future data, are drawn IID from some distribution on (x, y).
- This IID assumption is *not practical*: it is not satisfied by real data. Even so, the analysis and algorithms here have many elements that carry over to more practical settings.

How do we define "performs well on future examples?"

- Given one (x, y) and a prediction $\hat{y} = f(x)$, we suffer a *loss* $\ell(\hat{y}, y)$, e.g., logistic $\ln(1+\exp(-\hat{y}y))$, or squared $(\hat{y} - y)^2/2$.
- On a training set, we suffer *empirical risk* $\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_i \ell(f(x_i), y_i)$.
- For future (random!) data, we consider *(population) risk* $\mathcal{R}(f) = \mathbb{E}\,\ell(f(x), y) = \int \ell(f(x), y)\mathrm{d}\mu(x, y)$.

"Performs well on future examples" becomes "minimize $\mathcal{R}(f)$." We can decompose $\mathcal{R}(f)$ into three separate concerns: given a training algorithm's choice $\hat{f}$ in some class of functions/predictors $\mathcal{F}$, as well as some reference solution $\bar{f} \in \mathcal{F}$,

$$\begin{aligned}
\mathcal{R}(\hat{f}) &= \mathcal{R}(\hat{f}) - \widehat{\mathcal{R}}(\hat{f}) && \text{(generalization)} \\
&\quad + \widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(\bar{f}) && \text{(optimization)} \\
&\quad + \widehat{\mathcal{R}}(\bar{f}) - \mathcal{R}(\bar{f}) && \text{(concentration/generalization)} \\
&\quad + \mathcal{R}(\bar{f}). && \text{(approximation)}
\end{aligned}$$

These notes are organized are organized into separately considering these three terms (treating "generalization" and "concentration/generalization" together).

Remark 0.2 ***(sensitivity to complexity)*** As discussed, we aim to circumvent the aforementioned pitfalls by working with notions of *low complexity model* which work well with all three parts. There is still very little understanding of the right way to measure complexity, however here are some informal comments.

- First suppose there exists a low complexity $\bar{f} \in \mathcal{F}$ so that the **approximation term** $\mathcal{R}(\bar{f})$ is small. Since the complexity is low, then the **concentration/generalization term** $\widehat{\mathcal{R}}(\bar{f}) - \mathcal{R}(\bar{f})$ is small.
- Since $\bar{f}$ has low complexity, then hopefully we can find $\hat{f}$ with not much larger complexity via an algorithm that balances the **optimization term** $\widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(\bar{f})$ with the complexity of $\hat{f}$; if $\hat{f}$ has low complexity, then the **generalization term** $\mathcal{R}(\hat{f}) - \widehat{\mathcal{R}}(\hat{f})$ will be small.

Remark 0.3 The two-argument form $\ell(\hat{y}, y)$ is versatile. We will most often consider binary classification $y \in \{\pm 1\}$, where we always use the product $\hat{y}y$, even for the squared loss:

$$[\hat{y} - y]^2 = [y(y\hat{y} - 1)]^2 = (y\hat{y} - 1)^2.$$

This also means binary classification networks have output dimension one, not two.

Highlights

Here are a few of the shortened and/or extended proofs in these notes.

1. **Approximation.**
   - (Section 2.2) Succinct universal approximation via Stone-Weierstrass.
   - (Section 3) Succinct Barron's theorem (Fourier representation), with an explicit infinite width form.
   - (Section 5) Shorter depth separation proof.
2. **Optimization.**
   - (Section 8.1) Short re-proof of gradient flow convergence in the shallow NTK regime, due to (Chizat and Bach 2019).
   - (Section 10.3) Short proof that smooth margins are non-decreasing for homogeneous networks; originally due to (Lyu and Li 2019), this short proof is due to (Ji 2020).
3. **Generalization.**
   - (Section 16.2) Shortened "spectrally-normalized bound" proof (P. Bartlett, Foster, and Telgarsky 2017).
   - (Section 17.3) Shortened ReLU network VC dimension proof.

Missing topics and references

Due to the above philosophy, many topics are currently omitted. Over time I hope to fill the gaps.

Here are some big omissions, hopefully resolved soon:

- Architectures:
  - Non-feedforward, e.g., recurrent (Siegelmann and Sontag 1994).
  - Specific feedforward architecture choices like convolutional layers and skip connections.

  - Continuous depth, for instance various neural ODE frameworks (R. T. Q. Chen et al. 2018; Tzen and Raginsky 2019).

- Other learning paradigms:
  - Data augmentation, self-training, and distribution shift.
  - Unsupervised learning (e.g., GANs), Adversarial ML, RL.

Further omitted topics, in a bit more detail, are discussed separately for approximation (section 1.1), optimization (section 6.1), and generalization (section 11.1).

Acknowledgements

Thanks to Ziwei Ji for extensive comments, discussion, and the proof of Theorem 10.3; thanks to Daniel Hsu for extensive comments and discussion; thanks to Francesco Orabona for detailed comments spanning many sections; thanks to Ohad Shamir for extensive comments on many topics; thanks to Karolina Dziugaite and Dan Roy for extensive comments on the generalization material; thanks to Thien Nguyen for extensive and detailed comments and corrections on many sections. Further thanks to Nadav Cohen, Quanquan Gu, Suriya Gunasekar, Frederic Koehler, Justin Li, Akshayaa Magesh, Maxim Raginsky, David Rolnick, Kartik Sreenivasan, Matthieu Terris, and Alex Wozniakowski for various comments and feedback.

1 Approximation: preface

As above, we wish to ensure that our predictors $\mathcal{F}$ (e.g., networks of a certain architecture) have some element $\bar{f} \in \mathcal{F}$ which simultaneously has small $\mathcal{R}(f)$ and small complexity; we can re-interpret our notation and suppose $\mathcal{F}$ already is some constrained class of low-complexity predictors, and aim to make $\inf_{f\in\mathcal{F}} \mathcal{R}(f)$ small.

What is $\mathcal{F}$? In keeping with the earlier theme, it should be some convenient notion of "low complexity model"; but what is that?

1. **Models reached by gradient descent.** Since standard training methods are variants of simple first-order methods, it seems this might be a convenient candidate for $\mathcal{F}$ which is tight with practice. Unfortunately, firstly we only have understanding of these models very close to initialization and very late in training, whereas practice seems to lie somewhere between. Secondly, we can't just make this our definition as it breaks things in the standard approach to generalization.

2. **Models of low norm**, where norm is typically measured layer-wise, and also typically the "origin" is initialization. This is the current most common setup, though it doesn't seem to be able to capture the behavior of gradient descent that well, except perhaps when very close to initialization.

3. **All models of some fixed architecture**, meaning the weights can be arbitrary. This is the classical setup, and we'll cover it here, but it can often seem loose or insensitive to data, and was a key part of the criticisms against the general learning-theoretic approach (Zhang et al. 2017). The math is still illuminating and still key parts can be used as tools in a more sensitive analysis, e.g., by compressing a model and then applying one of these results.

The standard classical setup ("all models of some fixed architecture") is often stated with a goal of competing with all continuous functions:

$$\inf_{f \in \mathcal{F}} \mathcal{R}(f) \qquad \text{vs.} \qquad \inf_{g \text{ continuous}} \mathcal{R}(g).$$

E.g.,

$$\sup_{g \text{ cont.}} \inf_{f \in \mathcal{F}} \mathcal{R}(f) - \mathcal{R}(g).$$

To simplify further, if ℓ is ρ-Lipschitz (and still $y = \pm 1$),

$$\begin{aligned} \mathcal{R}(f) - \mathcal{R}(g) &= \int (\ell(yf(x)) - \ell(yg(x))) \, \mathrm{d}\mu(x, y) \\ &\leq \int \rho |yf(x) - yg(x)| \mathrm{d}\mu(x, y) = \rho \int |f(x) - g(x)| \mathrm{d}\mu(x, y), \end{aligned}$$

and in particular we have reduced the approximation question to one about studying $\|f - g\|$ with function space norms.

Remark 1.1 **(Is this too strenuous?)** Most of the classical work uses the *uniform norm*: $\|f - g\|_{\mathrm{u}} = \sup_{x \in S} |f(x) - g(x)|$ where S is some compact set, and compares against continuous functions. Unfortunately, already if the target is Lipschitz continuous, this means our function class needs complexity which scales exponentially with dimension (Luxburg and Bousquet 2004): this highlights the need for more refined target functions and approximation measures.

(Lower bounds.) The uniform norm has certain nice properties for proving upper bounds, but is it meaningful for a lower bound? Functions can be well-separated in uniform norm even if they are mostly the same: they just need one point of large difference. For this reason, L_1 norms, for instance $\int_{[0,1]^d} |f(x) - g(x)| \mathrm{d}x$ are prefered for lower bounds.

Remark 1.2 While norms have received much recent attention as a way to measure complexity, this idea is quite classical. For instance, a resurgence of interest in the 1990s led to the proof of many deep network VC dimension bounds, however very quickly it was highlighted (and proved) in (P. L. Bartlett 1996) that one has situations where the architecture (and connection cardinality) stays fixed (along with the VC dimension), yet the norms (and generalization properties) vary.

1.1 Omitted topics

- Full proofs for sobolev space approximation (Yarotsky 2016; Schmidt-Hieber 2017). [mjt☹: Planning to add in Fall 2021!!!]
- Approximation of distributions and other settings.
- Approximation power of low-norm functions.

2 Classical approximations and "universal approximation"

We start with two types of standard approximation results, in the "classical" regime where we only care about the number of nodes and not the magnitude of the weights, and also the worst-case goal of competing with an arbitrary continuous function using some function space norm.

1. Elementary folklore results: univariate approximation with one hidden layer, and multivariate approximation with two hidden layers, just by stacking bricks. Latter use L_1 metric, which is disappointing.
2. Celebrated "universal approximation" result: fitting continuous functions over compact sets in uniform norm with a single hidden layer (Hornik, Stinchcombe, and White 1989).

There are weaknesses in these results (e.g., curse of dimension), and thus they are far from the practical picture. Still, they are very interesting and influential.

2.1 Elementary folklore constructions

We can handle the univariate case by gridding the line and taking steps appropriately.

Proposition 2.1 Suppose $g : \mathbb{R} \to \mathbb{R}$ is ρ-Lipschitz. For any $\epsilon > 0$, there exists a 2-layer network f with $\lceil \frac{\rho}{\epsilon} \rceil$ threshold nodes $z \mapsto \mathbf{1}[z \geq 0]$ so that $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \epsilon$.

Proof. Define $m := \lceil \frac{\rho}{\epsilon} \rceil$, and for and $b_i := i\epsilon/\rho$ for $i \in \{0, \dots, m-1\}$, and

$$a_0 = g(0), \qquad a_i = g(b_i) - g(b_{i-1}),$$

and lastly define $f(x) := \sum_{i=0}^{m-1} a_i \mathbf{1}[x_i \geq b_i]$. Then for any $x \in [0,1]$, letting k be the largest index so that $b_k \leq x$, then f is constant along $[b_k, x]$, and

$$\begin{aligned}
|g(x) - f(x)| &\leq |g(x) - g(b_k)| + |g(b_k) - f(b_k)| + |f(b_k) - f(x)| \\
&\leq \rho |x - b_k| + \left| g(b_k) - \sum_{i=0}^{k} a_i \right| + 0 \\
&\leq \rho(\epsilon/\rho) + \left| g(b_k) - g(b_0) - \sum_{i=1}^{k} (g(b_i) - g(b_{i-1})) \right| \\
&= \epsilon.
\end{aligned}$$

Remark 2.1 This is standard, but we've lost something! We are paying for flat regions, which are a specialty of standard networks! A more careful proof only steps when it needs to and pays in *total variation*.

Now let's handle the multivariate case. We will replicate the univariate approach: we will increment function values when the target function changes. In the univariate case, we could "localize" function modifications, but in the multivariate case by default we will modify an entire halfspace at once. To get around this, we use an additional layer.

Remark 2.2

- Note the problem is easy for finite point-sets: can reduce to univariate apx after projection onto a random line (homework 1?). But our goal is approximation over a *distribution* of points.
- We will not get any nice theorem that says, roughly: "the exact complexity of shallow approximation depends on this function of the first $\mathcal{O}(d)$ derivatives" (see also (Yarotsky 2016) for the deep case). This is part of why I like discussing the univariate case, where we have nice characterizations with total variation distance.

Theorem 2.1 Let continuous $g : \mathbb{R}^d \to \mathbb{R}$ and an $\epsilon > 0$ be given, and choose $\delta > 0$ so that $\|x - x'\|_\infty \leq \delta$ implies $|g(x) - g(x')| \leq \epsilon$. Then there exists a 3-layer network f with $\Omega(\frac{1}{\delta^d})$ ReLU with $\int_{[0,1]^d} |f(x) - g(x)| \mathrm{d}x \leq 2\epsilon$.

Remark 2.3

- Note the *curse of dimension* (exponential dependence on d, which also appears in lower bounds (Luxburg and Bousquet 2004)). Note CIFAR has $d = 3072$. This issue is inherent in approximating arbitrary continuous functions, and makes this irrelevant in practice.
- Construction also has large weights and Lipschitz constant.
- Later in Theorem 2.3 ((Hornik, Stinchcombe, and White 1989)) we'll give another approach that controls $\sup_x |f(x) - g(x)|$ and uses only one activation layer, but it will not be a constructive proof, and trying to obtain estimates from it has all the preceding weaknesses as well.

The proof uses the following lemma (omitted in class), approximating continuous functions by piecewise constant functions.

Lemma 2.1 Let g, δ, ϵ be given as in Theorem 2.1. Let any set $U \subset \mathbb{R}^d$ be given, along with a partition $\mathcal{P}$ of U into rectangles (products of intervals) $\mathcal{P} = (R_1, \ldots, R_N)$ with all side lengths not exceeding δ. Then there exist scalars $(\alpha_1, \ldots, \alpha_N)$ so that

$$\sup_{x \in U} |g(x) - h(x)| \leq \epsilon, \qquad \text{where} \qquad h = \sum_{i=1}^{N} \alpha_i \mathbf{1}_{R_i}.$$

Proof. Let partition $\mathcal{P} = (R_1, \ldots, R_N)$ be given, and for each R_i, pick some $x_i \in R_i$, and set $\alpha_i := g(x_i)$. Since each side length of each R_i is at most δ,

$$\begin{aligned}
\sup_{x \in U} |g(x) - h(x)| &= \sup_{i \in \{1,\ldots,N\}} \sup_{x \in R_i} |g(x) - h(x)| \\
&\leq \sup_{i \in \{1,\ldots,N\}} \sup_{x \in R_i} (|g(x) - g(x_i)| + |g(x_i) - h(x)|) \\
&\leq \sup_{i \in \{1,\ldots,N\}} \sup_{x \in R_i} (\epsilon + |g(x_i) - \alpha_i|) = \epsilon.
\end{aligned}$$

Proof of Theorem 2.1. For convenience, throughout this proof define a norm $\|f\|_1 = \int_{[0,2)^d} |f(x)| \mathrm{d}x$. Let $\mathcal{P}$ denote a partition of $[0,2)^d$ into rectangles of the form $\prod_{j=1}^d [a_j, b_j)$ with $b_j - a_j \leq \delta$; the final result follows by restricting consideration to $[0,1]^d$, but we include an extra region to work with half-open intervals in a lazy way. Let $h = \sum_i \alpha_i \mathbf{1}_{R_i}$ denote the piecewise-constant function provided by Lemma 2.1 with the given partition $\mathcal{P}$, which satisfies $\|g - h\|_1 \leq \epsilon$. Our final network f will be of the form $f(x) := \sum_i \alpha_i g_i(x)$, where each g_i will be a ReLU network with two hidden layers and $\mathcal{O}(d)$ nodes; since $|\mathcal{P}| \geq 1/\delta^d$, then f also uses at least $1/\delta^d$ nodes as stated. Our goal is to show $\|f - g\|_1 \leq 2\epsilon$; to this end, note by the preceding

choices and the triangle inequality that

$$\begin{aligned}\|f-g\|_1 &\leq \|f-h\|_1 + \|h-g\|_1 \\ &= \left\|\sum_i \alpha_i(\mathbf{1}_{R_i} - g_i)\right\|_1 + \epsilon \\ &\leq \sum_i |\alpha_i| \cdot \|\mathbf{1}_{R_i} - g_i\|_1 + \epsilon.\end{aligned}$$

As such, if we can construct each g_i so that $\|\mathbf{1}_{R_i} - g_i\|_1 \leq \frac{\epsilon}{\sum_i |\alpha_i|}$, then the proof is complete. (If $\sum_i |\alpha_i| = 0$, we can set f to be the constant 0 network and the proof is again complete.)
Now fix i and let rectangle R_i be given of the form $R_i := \times_{j=1}^d [a_j, b_j)$, and define g_i as follows. Letting $\gamma > 0$ denote a free parameter to be optimized at the end of the proof, for each $j \in \{1, \ldots, d\}$ define

$$\begin{aligned}g_{\gamma,j}(z) &:= \sigma\left(\frac{z-(a_j-\gamma)}{\gamma}\right) - \sigma\left(\frac{z-a_j}{\gamma}\right) - \sigma\left(\frac{z-b_j}{\gamma}\right) + \sigma\left(\frac{z-(b_j+\gamma)}{\gamma}\right) \\ &\in \begin{cases}\{1\} & z \in [a_j, b_j], \\ \{0\} & x \notin [a_j-\gamma, b_j+\gamma], \\ [0,1] & \text{otherwise},\end{cases}\end{aligned}$$

and additionally

$$g_\gamma(x) := \sigma(\sum_j g_{\gamma,j}(x_j) - (d-1)).$$

(Note that a second hidden layer is crucial in this construction, it is not clear how to proceed without it, certainly with only $\mathcal{O}(d)$ nodes. Later proofs can use only a single hidden layer, but they are not constructive, and need $\mathcal{O}(d)$ nodes.) Note that $g_\gamma \approx \mathbf{1}_{R_i}$ as desired, specifically

$$g_\gamma(x) = \begin{cases}1 & x \in R_i, \\ 0 & x \notin \times_j [a_j-\gamma, b_j+\gamma], \\ [0,1] & \text{otherwise},\end{cases}$$

From which it follows that

$$\begin{aligned}\|g_\gamma(x) - \mathbf{1}_{R_i}(x)\|_1 &= \int_{R_i} |g_\gamma - \mathbf{1}_{R_i}| + \int_{\times_j [a_j-\gamma, b_j+\gamma] \setminus R_i} |g_\gamma - \mathbf{1}_{R_i}| + \int_{[0,2)^d \setminus \times_j [a_j-\gamma, b_j+\gamma]} |g_\gamma - \mathbf{1}_{R_i}| \\ &\leq 0 + \prod_{j=1}^d (b_j - a_j + 2\gamma) - \prod_{j=1}^d (b_j - a_j) + 0 \\ &\leq \mathcal{O}(\gamma),\end{aligned}$$

which means we can ensure $\|\mathbf{1}_{R_i} - g_\gamma\|_1 \leq \frac{\epsilon}{\sum_i |\alpha_i|}$ by choosing sufficiently small γ, which completes the proof.

2.2 Universal approximation with a single hidden layer

The proof of Theorem 2.1 use two layers to construct g_γ such that $g_\gamma(x) \approx \mathbf{1}[x \in \times_i [a_i, b_i]]$. If instead we had a way to approximate multiplication we could instead approximate

$$x \mapsto \prod_i \mathbf{1}[x_i \in [a_i, b_i]] = \mathbf{1}[x \in \times_i [a_i, b_i]].$$

Can we do this and then form a linear combination, all with just one hidden layer?

The answer will be yes, and we will use this to resolve the classical *universal approximation* question with a single hidden layer.

Definition 2.1 A class of functions $\mathcal{F}$ is a *universal approximator* over a compact set S if for every continuous function g and target accuracy $\epsilon > 0$, there exists $f \in \mathcal{F}$ with

$$\sup_{x \in S} |f(x) - g(x)| \leq \epsilon.$$

Remark 2.4 Typically we will take $S = [0, 1]^d$; we can then reduce arbitrary compact sets to this case by defining a new function which re-scales the input. The compactness is in a sense necessary: as in the homework, consider approximating the sin function with a finite-size ReLU network over all of $\mathbb{R}$. Lastly, universal approximation is often stated more succinctly as some class being dense in all continuous functions over compact sets.

Consider *unbounded width networks with one hidden layer:*

$$\mathcal{F}_{\sigma,d,m} := \mathcal{F}_{d,m} := \left\{ x \mapsto a^\intercal \sigma(Wx + b) : a \in \mathbb{R}^m, W \in \mathbb{R}^{m \times d}, b \in \mathbb{R}^m \right\}.$$
$$\mathcal{F}_{\sigma,d} := \mathcal{F}_d := \bigcup_{m \geq 0} \mathcal{F}_{\sigma,d,m}.$$

Note that $\mathcal{F}_{\sigma,m,1}$ denotes networks with a single node, and $\mathcal{F}_{\sigma,d}$ is the linear span (in function space) of single-node networks.

First consider the (unusual) activation $\sigma = \cos$. Since $2\cos(y)\cos(z) = \cos(y+z) + \cos(y-z)$, then

$$2\left[\sum_{i=1}^m a_i \cos(w_i^\intercal x + b_i)\right] \cdot \left[\sum_{j=1}^n c_j \cos(u_j^\intercal x + v_j)\right] =$$
$$\sum_{i=1}^m \sum_{j=1}^n a_i c_j \left(\cos((w_i + u_j)^\intercal x + (b_i + v_j)) + \cos((w_i - u_j)^\intercal x + (b_i - v_j))\right),$$

thus $f, g \in \mathcal{F}_{\cos,d} \implies fg \in \mathcal{F}_{\cos,d}$! In other words, $\mathcal{F}_{\cos,d}$ is closed under multiplication, and since we know we can approximate univariate functions arbitrarily well, this suggests that we can approximate $x \mapsto \prod_i \mathbf{1}[x_i \in [a_i, b_i]] = \mathbf{1}[x \in \times_i [a_i, b_i]]$, and use it to achieve our more general approximation goal.

We're in good shape to give the general universal approximation result. The classical Weierstrass theorem establishes that polynomials are universal approximators (Weierstrass 1885), and its generalization, the Stone-Weierstrass theorem, says that any family of functions satisfying some of the same properties as polynomials will also be a universal approximator. Thus we will show $\mathcal{F}_{\sigma,d}$ is a universal approximator via Stone-Weierstrass, a key step being closure under multiplication as above; this proof scheme was first suggested in (Hornik, Stinchcombe, and White 1989), but is now a fairly standard way to prove universal approximation.

First, here is the statement of the Stone-Weierstrass Theorem.

Theorem 2.2 ***(Stone-Weierstrass; (Folland 1999, Theorem 4.45))*** Let functions $\mathcal{F}$ be given as follows.

1. Each $f \in \mathcal{F}$ is continuous.
2. For every x, there exists $f \in \mathcal{F}$ with $f(x) \neq 0$.
3. For every $x \neq x'$ there exists $f \in \mathcal{F}$ with $f(x) \neq f(x')$ ($\mathcal{F}$ *separates points*).
4. $\mathcal{F}$ is closed under multiplication and vector space operations ($\mathcal{F}$ is an algebra).

Then $\mathcal{F}$ is a universal approximator: for every continuous $g : \mathbb{R}^d \to \mathbb{R}$ and $\epsilon > 0$, there exists $f \in \mathcal{F}$ with $\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq \epsilon$.

Remark 2.5

- This is a heavyweight tool, but a convenient way to quickly check universal approximation.
- Proofs are not constructive, but the size lower bound $\Omega(\frac{1}{\epsilon^d})$ seems to naturally appear in various places; e.g., to show closure under products as above, we double (or more) the number of terms for each dimension.
- Weierstrass theorem itself has interesting proofs:
  - The modern standard one is due to Bernstein; it picks a fine grid and then a convenient set of interpolating polynomials which behave stably off the grid.
  - Weierstrass's original proof convolved the target with a Gaussian, which makes it analytic, and also leads to good polynomial approximation.
- The second and third conditions in Stone-Weierstrass are necessary; if there exists x so that $f(x) = 0 \forall f \in \mathcal{F}$, then we can't approximate g with $g(x) \neq 0$; if we can't separate points $x \neq x'$, then we can't approximate functions with $g(x) \neq g(x')$.

First, we go back to cos activations, which was the original choice in (Hornik, Stinchcombe, and White 1989); we can then handle arbitrary activations by univariate approximation of cos, without increasing the depth (but increasing the width).

Lemma 2.2 ***((Hornik, Stinchcombe, and White 1989))*** $\mathcal{F}_{\cos,d}$ is universal.

Proof. Let's check the Stone-Weierstrass conditions:

1. Each $f \in \mathcal{F}_{\cos,d}$ is continuous.
2. For each x, $\cos(0^\intercal x) = 1 \neq 0$.
3. For each $x \neq x'$, $f(z) := \cos((z - x')^\intercal (x - x')/\|x - x'\|^2) \in \mathcal{F}_d$ satisfies
$$f(x) = \cos(1) \neq \cos(0) = f(x').$$
4. $\mathcal{F}_{\cos,d}$ is closed under products and vector space operations as before.

We can work it out even more easily for $\mathcal{F}_{\exp,d}$.

Lemma 2.3 $\mathcal{F}_{\exp,d}$ is universal.

Proof. Let's check the Stone-Weierstrass conditions:

1. Each $f \in \mathcal{F}_{\exp,d}$ is continuous.
2. For each x, $\exp(0^\intercal x) = 1 \neq 0$.

3. For each $x \neq x'$, $f(z) := \exp((z-x')^\intercal(x-x')/\|x-x'\|^2) \in \mathcal{F}_d$ satisfies

$$f(x) = \exp(1) \neq \exp(0) = f(x').$$

4. $\mathcal{F}_{\exp,d}$ is closed under VS ops by construction; for products,

$$\left(\sum_{i=1}^n r_i \exp(a_i^\intercal x)\right)\left(\sum_{j=1}^m s_j \exp(b_j^\intercal x)\right) = \sum_{i=1}^m \sum_{j=1}^m r_i s_j \exp((a+b)^\intercal x).$$

Now let's handle arbitrary activations.

Theorem 2.3 ***((Hornik, Stinchcombe, and White 1989))*** Suppose $\sigma : \mathbb{R} \to \mathbb{R}$ is *sigmoidal*: it is continuous, and

$$\lim_{z\to-\infty} \sigma(z) = 0, \qquad \lim_{z\to+\infty} \sigma(z) = 1.$$

Then $\mathcal{F}_{\sigma,d}$ is universal.

Proof sketch (details in hw1)**.** Given $\epsilon > 0$ and continuous g, use Lemma 2.2 ((Hornik, Stinchcombe, and White 1989)) (or Lemma 2.3) to obtain $h \in \mathcal{F}_{\cos,d}$ (or $\mathcal{F}_{\exp,d}$) with $\sup_{x\in[0,1]^d} |h(x) - g(x)| \leq \epsilon/2$. To finish, replace all appearances of cos with an element of $\mathcal{F}_{\sigma,1}$ so that the total additional error is $\epsilon/2$.

Remark 2.6

- ReLU is fine: use $z \mapsto \sigma(z) - \sigma(z-1)$ and split nodes.
- exp didn't need bias in the proof, but this seems natural due to $\exp(a^\intercal x + b) = e^b \cdot \exp(a^\intercal x)$. On the other hand, approximating exp with ReLU uses bias terms, so we don't obtain a trick via exp to remove biases in general.
- Weakest conditions on σ (Leshno et al. 1993): universal apx iff **not** a polynomial.
- Carefully accounting within the proof seems to indicate curse of dimension again (size $\Omega(\frac{1}{\epsilon^d})$), due for instance to expanding all terms in a product of d terms.

Remark 2.7 ***(other universal approximation proofs)***

- (Cybenko 1989) Assume contradictorily you miss some functions. By duality, $0 = \int \sigma(a^\intercal x - b)\mathrm{d}\mu(x)$ for some signed measure μ, all (a, b). Using Fourier, can show this implies $\mu = 0 \ldots$
- (Leshno et al. 1993) If σ a polynomial, $\ldots$; else can (roughly) get derivatives and polynomials of all orders (we'll have homework problems on this).
- (Barron 1993) Use inverse Fourier representation to construct an infinite-width network; we'll cover this next. It can beat the worst-case curse of dimension!
- (Funahashi 1989) [mjt☹: I'm sorry, I haven't read it. Also uses Fourier.]

3 Infinite-width Fourier representations and the Barron norm

This section presents two ideas which have recently become very influential again.

1. Using infinite-width networks. This may seem complicated, but in fact it simplifies many things, and better captures certain phenomena.
2. Barron's approximation theorem and norm (Barron 1993). Barron's original goal was an approximation result which requires few nodes in some favorable cases. Interestingly, his construction can be presented as an infinite-width representation *with equality*, and furthermore the construction gives approximation guarantees near initialization (e.g., for the NTK, the topic of the next section).

We will finish the section with a more general view of these infinite-width constructions, and a technique to sample finite-width networks from them.

3.1 Infinite-width univariate approximations

Let's warm up with some univariate constructions.

Proposition 3.1 Suppose $g : \mathbb{R} \to \mathbb{R}$ is differentiable, and $g(0) = 0$. If $x \in [0, 1]$, then $g(x) = \int_0^1 \mathbf{1}[x \geq b] g'(b) \mathrm{d}b$.

Proof. By FTC and $g(0) = 0$ and $x \in [0, 1]$,

$$g(x) = g(0) + \int_0^x g'(b)\mathrm{d}b = 0 + \int_0^1 \mathbf{1}[x \geq b] g'(b)\mathrm{d}b.$$

That's really it! We've written a differentiable function as a shallow infinite-width network, *with equality*, effortlessly.

Remark 3.1 In the last subsection, when we sample from infinite-width networks, The error for this univariate case will scale with $\int_0^1 |g'(x)|\mathrm{d}x$. This quantity is adaptive, e.g., correctly not paying for flat regions, which was discussed after our basic grid-based univariate approximation in Proposition 2.1. As mentioned before, this is a big point of contrast with polynomial approximation.

3.2 Barron's construction for infinite-width multivariate approximation

This approach uses Fourier transforms; for those less familiar, it might seem daunting, but:

- The approach will turn out to be natural.
- There is extensive literature on Fourier transforms, so it's an important connection to make.
- The original paper (Barron 1993) is over 30 years old now, and still this seems to be one of the best approaches, even with modern considerations like staying near initialization!

Let's first argue it's natural. Recall the Fourier transform (e.g., Folland 1999, Chapter 8):

$$\hat{f}(w) := \int \exp(-2\pi i w^\mathsf{T} x) f(x) \mathrm{d}x.$$

We also have Fourier inversion: if $f \in L^1$ and $\hat{f} \in L^1$,

$$f(x) = \int \exp(2\pi i w^\mathsf{T} x) \hat{f}(w) \mathrm{d}w.$$

The inversion formula rewrite f as an infinite-width network! The only catch is that the activations are not only non-standard, they are over the complex plane.

Remark 3.2 Unfortunately, there are different conventions for the Fourier transform (in fact, the original work we reference uses a different one (Barron 1993)).

Barron's aproach is to convert these activations into something more normal; here we'll use threshold nodes, but others are fine as well. If our starting function f is over the reals, then using $\Re$ to denote the real part of a complex number, meaning $\Re(a+bi)=a$, then

$$f(x) = \Re f(x) = \int \Re \exp(2\pi i w^\intercal x)\hat{f}(w)\mathrm{d}w.$$

If we expand with $e^{iz} = \cos(z) + i\sin(z)$, we're left with cos, which is not compactly supported; to obtain an infinite-width form with threshold gates using a density which is compactly supported, Barron uses two tricks.

1. **Polar decomposition.** Let's split up the Fourier transform $\hat{f}$ into magnitude and radial parts: write $\hat{f}(w) = |\hat{f}(w)|\exp(2\pi i\theta(w))$ with $|\theta(w)| \le 1$. Since f is real-valued,

$$\begin{aligned} f(x) &= \Re \int \exp(2\pi i w^\intercal x)\hat{f}(w)\mathrm{d}w \\ &= \int \Re\left(\exp(2\pi i w^\intercal x)\exp(2\pi i\theta(w))|\hat{f}(w)|\right)\mathrm{d}w \\ &= \int \Re\left(\exp(2\pi i w^\intercal x + 2\pi i\theta(w))\,|\hat{f}(w)|\mathrm{d}w\right. \\ &= \int \cos\left(2\pi w^\intercal x + 2\pi\theta(w)\right)|\hat{f}(w)|\mathrm{d}w. \end{aligned}$$

   We've now obtained an infinite width network over real-valued activations! cos is neither compactly supported, no approaches a limit as its argument goes $\pm\infty$, which is where Barron's second trick comes in.

2. **Turning cosines into bumps!** We'll do two things to achieve our goal: subtracting $f(0)$, and scaling by $\|w\|$:

$$\begin{aligned} &f(x) - f(0) \\ &= \int \left[\cos(2\pi w^\intercal x + 2\pi\theta(w)) - \cos(2\pi w^\intercal 0 + 2\pi\theta(w))\right]|\hat{f}(w)|\mathrm{d}w \\ &= \int \frac{\cos(2\pi w^\intercal x + 2\pi\theta(w)) - \cos(2\pi\theta(w))}{\|w\|}\|w\|\cdot|\hat{f}(w)|\mathrm{d}w. \end{aligned}$$

   The fraction does not blow up: since cos is 1-Lipschitz,

$$\begin{aligned} &\left|\frac{\cos(2\pi w^\intercal x + 2\pi\theta(w)) - \cos(2\pi\theta(w))}{\|w\|}\right| \\ &\le \frac{|2\pi w^\intercal x + 2\pi\theta(w) - 2\pi\theta(w)|}{\|w\|} \le \frac{2\pi|w^\intercal x|}{\|w\|} \le 2\pi\|x\|. \end{aligned}$$

   This quantity is therefore well-behaved for bounded $\|x\|$ so long as $\|w\||\hat{f}(w)|$ is well-behaved.

Barron combined these ideas with the sampling technique in Lemma 3.1 (Maurey (Pisier 1980)) to obtain estimates on the number of nodes needed to approximate functions whenever $\|w\|\cdot|\hat{f}(w)|$

is well-behaved. We will follow a simpler approach here: we will give an explicit infinite-width form via only the first trick above and some algebra, and only then invoke sampling. The quantity $\|w\| \cdot |\hat{f}(w)|$ will appear in the estimate of the "mass" of the infinite-width network as used to estimate how much to sample, analogous to the quantity $\int_0^1 |g'(x)|\mathrm{d}x$ from Proposition 2.1.

Before continuing, let's discuss $\|w\| \cdot |\hat{f}(w)|$ a bit more, which can be simplified via $\widehat{\nabla f}(w) = 2\pi i w \hat{f}(w)$ into a form commonly seen in the literature.

Definition 3.1 The quantity

$$\int \left\| \widehat{\nabla f}(w) \right\| \mathrm{d}w = 2\pi \int \|w\| \cdot |\hat{f}(w)| \mathrm{d}w$$

is the *Barron norm* of a function f. The corresponding *Barron class with norm* C is

$$\mathcal{F}_C := \left\{ f : \mathbb{R}^d \to \mathbb{R} \quad : \quad \hat{f} \text{ exists}, \int \left\| \widehat{\nabla f}(w) \right\| \mathrm{d}w \leq C \right\}.$$

Remark 3.3 Barron's approximation bounds were on $\mathcal{F}_C$, and in particular the number of nodes needed scaled with C/ϵ^2, where ϵ is the target accuracy. As we will see later, since threshold units are (kindof) the derivatives of ReLUs, then the Barron norm can also be used for complexity estimates of shallow networks near initialization (the NTK regime) (Ji, Telgarsky, and Xian 2020). [mjt☹: My friend Daniel Hsu told me that related ideas are in (Bach 2017) as well, though I haven't read closely and fleshed out this connection yet.]

Here is our approach in detail. Continuing with the previous Barron representation and using $\|x\| \leq 1$,

$$\begin{aligned}
\cos(2\pi w^\mathsf{T} x + 2\pi\theta(w)) - \cos(2\pi\theta(w)) &= \int_0^{w^\mathsf{T} x} -2\pi \sin(2\pi b + 2\pi\theta(w)) \mathrm{d}b \\
&= -2\pi \int_0^{\|w\|} \mathbf{1}[w^\mathsf{T} x - b \geq 0] \sin(2\pi b + 2\pi\theta(w)) \mathrm{d}b \\
&\quad + 2\pi \int_{-\|w\|}^0 \mathbf{1}[-w^\mathsf{T} x + b \geq 0] \sin(2\pi b + 2\pi\theta(w)) \mathrm{d}b.
\end{aligned}$$

Plugging this into the previous form (before dividing by $\|w\|$),

$$\begin{aligned}
f(x) - f(0) &= -2\pi \int\int_0^{\|w\|} \mathbf{1}[w^\mathsf{T} x - b \geq 0] \left[\sin(2\pi b + 2\pi\theta(w)) |\hat{f}(w)| \right] \mathrm{d}b \mathrm{d}w \\
&\quad + 2\pi \int\int_{-\|w\|}^0 \mathbf{1}[-w^\mathsf{T} x + b \geq 0] \left[\sin(2\pi b + 2\pi\theta(w)) |\hat{f}(w)| \right] \mathrm{d}b \mathrm{d}w,
\end{aligned}$$

an infinite width network with threshold nodes!

We'll tidy up with $\widehat{\nabla f}(w) = 2\pi i w \hat{f}(w)$ whereby $\|\widehat{\nabla f}(w)\| = 2\pi \|w\| \cdot |\hat{f}(w)|$ as mentioned before. Lastly, to estimate the "mass" of this infinite width network (the integral of the density part of the

integrand),

$$\begin{aligned}
&\left|2\pi \int\int_0^{\|w\|} \left[\sin(2\pi b + 2\pi\theta(w))|\hat{f}(w)|\right] \mathrm{d}b\mathrm{d}w\right| \\
&+ \left|2\pi \int\int_{-\|w\|}^{0} \left[\sin(2\pi b + 2\pi\theta(w))|\hat{f}(w)|\right] \mathrm{d}b\mathrm{d}w\right| \\
&\leq 2\pi \int\int_{-\|w\|}^{\|w\|} |\sin(2\pi b + 2\pi\theta(w))|\,|\hat{f}(w)|\mathrm{d}b\mathrm{d}w \\
&\leq 2\pi \int 2\|w\| \cdot |\hat{f}(w)|\mathrm{d}b\mathrm{d}w \\
&= 2\int \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w.
\end{aligned}$$

Summarizing this derivations gives the following version of Barron's approach.

Theorem 3.1 ***(based on (Barron 1993))*** Suppose $\int \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w < \infty$, $f \in L_1$, $\hat{f} \in L_1$, and write $\hat{f}(w) = |\hat{f}(w)| \exp(2\pi i\theta(w))$. For $\|x\| \leq 1$,

$$\begin{aligned}
f(x) - f(0) &= \int \frac{\cos(2\pi w^\mathsf{T} x + 2\pi\theta(w)) - \cos(2\pi\theta(w))}{2\pi\|w\|} \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w \\
&= -2\pi \int\int_0^{\|w\|} \mathbf{1}[w^\mathsf{T} x - b \geq 0] \left[\sin(2\pi b + 2\pi\theta(w))|\hat{f}(w)|\right] \mathrm{d}b\mathrm{d}w \\
&\quad + 2\pi \int\int_{-\|w\|}^{0} \mathbf{1}[-w^\mathsf{T} x + b \geq 0] \left[\sin(2\pi b + 2\pi\theta(w))|\hat{f}(w)|\right] \mathrm{d}b\mathrm{d}w.
\end{aligned}$$

The corresponding measure on weights has mass at most

$$2\int \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w.$$

When combined with the sampling tools in 3.3, we will recover Barron's full result that the number of nodes needed to approximate f to accuracy $\epsilon > 0$ is roughly $\left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w/\epsilon^2$.

Ideally, the Barron norm is small, for instance polynomial (rather than exponential) in dimension for interesting examples. Here are a few, mostly taken from (Barron 1993).

- **Gaussians.** Since (e.g., Folland 1999, Prop 8.24)

$$\begin{aligned}
& f(x) = (2\pi\sigma^2)^{d/2} \exp(-\frac{\|x\|^2}{2\sigma^2}) \\
\implies \quad & \hat{f}(w) = \exp(-2\pi^2\sigma^2\|w\|^2),
\end{aligned}$$

  meaning $\hat{f}$ is an unnormalized Gaussian with variance $(4\pi^2\sigma^2)^{-1}$. Using normalization $Z = (2\pi\sigma^2)^{-d/2}$ and Holder gives

$$\begin{aligned}
\int \|w\||\hat{f}(w)|\mathrm{d}w &= Z \int Z^{-1}\|w\||\hat{f}(w)|\mathrm{d}w \\
&\leq Z \left(\int Z^{-1}\|w\|^2|\hat{f}(w)|\mathrm{d}w\right)^{1/2} \\
&= Z \left(\frac{d}{4\pi^2\sigma^2}\right)^{1/2} = \frac{\sqrt{d}}{\sqrt{2\pi}(2\pi\sigma^2)^{(d+1)/2}}.
\end{aligned}$$

Consequently, if $2\pi\sigma^2 \geq 1$, then $\int \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w = \mathcal{O}(\sqrt{d})$. On the other hand, general radial functions have exponential $\|\widehat{\nabla f}(w)\|$ (Comment IX.9, Barron 1993); this is circumvented here since $\|x\| \leq 1$ and hence the Gaussian is quite flat.

- Further brief example $\int \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w$ calculations:
  - A few more from (Barron 1993, sec. IX): radial functions (IX.9), compositions with polynomials (IX.12) and analytic functions (IX.13), functions with $\mathcal{O}(d)$ bounded derivatives (IX.15).
  - Barron also gives a lower bound for a specific set of functions which is exponential in dimension.
  - Further comments on Barron's constructions can be found in (H. Lee et al. 2017).
  - General continuous functions can fail to satisfy $\int \left\|\widehat{\nabla f}(w)\right\| \mathrm{d}w < \infty$, but we can first convolve them with Gaussians and sample the resulting nearby function; this approach, along with a Barron theorem using ReLUs, can be found in (Ji, Telgarsky, and Xian 2020).

3.3 Sampling from infinite width networks

Now we will show how to obtain a finite-width representation from an infinite-width representation. Coarsely, given a representation $\int \sigma(w^\mathsf{T} x) g(w) \mathrm{d}w$, we can form an estimate

$$\sum_{j=1}^{m} s_j \tilde{\sigma}(w_j^\mathsf{T} x), \qquad \text{where } s_j \in \pm 1,\ \tilde{\sigma}(z) = \sigma(z) \int |g(w)| \mathrm{d}w,$$

by sampling $w_j \sim |g(w)| / \int |g(w)| \mathrm{d}w$, and letting $s_j := \operatorname{sgn}(g(w_j))$, meaning the sign corresponding to whether w fell in a negative or positive region of g. In expectation, this estimate is equal to the original function.

Here we will give a more general construction where the integral is not necessarily over the Lebesgue measure, which is useful when it has discrete parts and low-dimensional sets. This section will follow the same approach as (Barron 1993), namely using Maurey's sampling method (cf. Lemma 3.1 (Maurey (Pisier 1980))), which gives an L_2 error; it is possible to use these techniques to obtain an L_∞ error via the "co-VC dimension technique" (Gurvits and Koiran 1995), but this is not pursued here.

To build this up, first let us formally define these infinite-width networks and their mass.

Definition 3.2 An *infinite-width shallow network* is characterized by a *signed measure* ν over weight vectors in $\mathbb{R}^p$:

$$x \mapsto \int \sigma(w^\mathsf{T} x) \mathrm{d}\nu(w).$$

The *mass* of ν is the total positive and negative weight mass assigned by ν: $|\nu|(\mathbb{R}^p) = \nu_-(\mathbb{R}^p) + \nu_+(\mathbb{R}^p)$.

Remark 3.4 We can connect this to the initial discussion of $\int \sigma(w^\mathsf{T} x) g(w) \mathrm{d}w$ by defining a signed measure ν via $\mathrm{d}\nu = g$, and the mass is once again $|\nu|(\mathbb{R}^p) = \int |g(w)| \mathrm{d}w$, and the positive and negative parts ν_- and ν_+ are simply the regions where g is respectively negative (or just non-positive) and positive.

In the case of general measures, a decomposition into ν_- and ν_+ is guaranteed to exist (Jordan decomposition, Folland 1999), and is unique up to null sets.
The notation here uses $\mathbb{R}^p$ not $\mathbb{R}^d$ since we might bake in biases and other feature mappings.

To develop sampling bounds, first we give the classical general Maurey sampling technique, which is stated as sampling in Hilbert spaces.

Suppose $X = \mathbb{E}\, V$, where r.v. V is supported on a set S. A natural way to "simplify" X is to instead consider $\hat{X} := \frac{1}{k}\sum_{i=1}^k V_i$, where $(V_1, \ldots, V_k)$ are sampled iid. We want to argue $\hat{X} \approx X$; since we're in a Hilbert space, we'll try to make the Hilbert norm $\|X - \hat{X}\|$ small.

Lemma 3.1 ***(Maurey (Pisier 1980))*** Let $X = \mathbb{E}\, V$ be given, with V supported on S, and let $(V_1, \ldots, V_k)$ be iid draws from the same distribution. Then

$$\mathop{\mathbb{E}}_{V_1,\ldots,V_k} \left\| X - \frac{1}{k}\sum_i V_i \right\|^2 \leq \frac{\mathbb{E}\,\|V\|^2}{k} \leq \frac{\sup_{U\in S}\|U\|^2}{k},$$

and moreover there exist $(U_1, \ldots, U_k)$ in S so that

$$\left\| X - \frac{1}{k}\sum_i U_i \right\|^2 \leq \mathop{\mathbb{E}}_{V_1,\ldots,V_k} \left\| X - \frac{1}{k}\sum_i V_i \right\|^2.$$

Remark 3.5 After proving this, we'll get a corollary for sampling from networks.
This lemma is widely applicable; e.g., we'll use it for generalization too.
First used for neural networks by (Barron 1993) and (Jones 1992), attributed to Maurey by (Pisier 1980).

Proof of Lemma 3.1 (Maurey (Pisier 1980)). Let $(V_1, \ldots, V_k)$ be IID as stated. Then

$$\begin{aligned}
&\mathop{\mathbb{E}}_{V_1,\ldots,V_k} \left\| X - \frac{1}{k}\sum_i V_i \right\|^2 \\
&= \mathop{\mathbb{E}}_{V_1,\ldots,V_k} \left\| \frac{1}{k}\sum_i (V_i - X) \right\|^2 \\
&= \mathop{\mathbb{E}}_{V_1,\ldots,V_k} \frac{1}{k^2}\left[\sum_i \|V_i - X\|^2 + \sum_{i\neq j} \langle V_i - X, V_j - X\rangle \right] \\
&= \mathop{\mathbb{E}}_{V} \frac{1}{k}\|V - X\|^2 \\
&= \mathop{\mathbb{E}}_{V} \frac{1}{k}\left(\|V\|^2 - \|X\|^2 \right) \\
&\leq \mathop{\mathbb{E}}_{V} \frac{1}{k}\|V\|^2 \leq \sup_{U\in S} \frac{1}{k}\|U\|^2.
\end{aligned}$$

To conclude, there must exist $(U_1, \ldots, U_k)$ in S so that $\left\|X - k^{-1}\sum_i U_i\right\|^2 \leq \mathbb{E}_{V_1,\ldots,V_k}\left\|X - k^{-1}\sum_i V_i\right\|^2$. ("Probabilistic method.")

Now let's apply this to infinite-width networks in the generality of Definition 3.2. We have two issues to resolve.

- **Issue 1:** what is the appropriate Hilbert space?
  - **Answer:** We'll use $\langle f, g\rangle = \int f(x)g(x)\mathrm{d}P(x)$ for some probability measure P on x, so $\|f\|_{L_2(P)}^2 = \int f(x)^2\mathrm{d}P(x)$.
- **Issue 2:** our "distribution" on weights is not a probability!
  - **Example:** consider $x \in [0,1]$ and $\sin(2\pi x) = \int_0^1 \mathbf{1}[x \geq b]2\pi\cos(2\pi b)\mathrm{d}b$. There are two issues: $\int_0^1 |2\pi\cos(2\pi b)|\mathrm{d}b \neq 1$, and $\cos(2\pi b)$ takes on negative and positive values.
  - **Answer:** we'll correct this in detail shortly, but here is a sketch; recall also the discussion in Definition 3.2 of splitting a measure into positive and negative parts. First, we introduce a fake parameter $s \in \{\pm 1\}$ and multiply $\mathbf{1}[x \geq b]$ with it, simulating positive and negative weights with only positive weights; now our distribution is on pairs (s, b). Secondly, we'll normalize everything by $\int_0^1 |2\pi\cos(2\pi b)|\mathrm{d}b$.

Let's write a generalized shallow network as $x \mapsto \int g(x;w)\mathrm{d}\mu(w)$, where μ is a nonzero signed measure over some abstract parameter space $\mathbb{R}^p$. E.g., $w = (a, b, v)$ and $g(x; w) = a\sigma(v^\mathsf{T}x + b)$.

- Decompose $\mu = \mu_+ - \mu_-$ into nonnegative measures $\mu_\pm$ with disjoint support (this is the *Jordan decomposition* (Folland 1999), which was mentioned in Definition 3.2).
- For nonnegative measures, define total mass $\|\mu_\pm\|_1 = \mu_\pm(\mathbb{R}^p)$, and otherwise $\|\mu\|_1 = \|\mu_+\|_1 + \|\mu_-\|_1$.
- Define $\widetilde{\mu}$ to sample $s \in \{\pm 1\}$ with $\Pr[s = +1] = \frac{\|\mu_+\|_1}{\|\mu\|_1}$, and then sample $g \sim \frac{\mu_s}{\|\mu_s\|_1} =: \tilde{\mu}_s$, and output $\tilde{g}(\cdot; w, s) = s\|\mu\|_1 g(\cdot; w)$.

This sampling procedure has the correct mean:

$$\begin{aligned}
\int g(x;w)\mathrm{d}\mu(w) &= \int g(x;w)\mathrm{d}\mu_+(w) - \int g(x;w)\mathrm{d}\mu_-(w) \\
&= \|\mu_+\|_1 \mathop{\mathbb{E}}_{\tilde{\mu}_+} g(x;w) - \|\mu_-\|_1 \mathop{\mathbb{E}}_{\tilde{\mu}_-} g(x;w) \\
&= \|\mu\|_1 \left[\mathop{\Pr}_{\widetilde{\mu}}[s=+1] \mathop{\mathbb{E}}_{\widetilde{\mu}_+} g(x;w) - \mathop{\Pr}_{\widetilde{\mu}}[s=-1] \mathop{\mathbb{E}}_{\widetilde{\mu}_-} g(x;w)\right] = \mathop{\mathbb{E}}_{\widetilde{\mu}} \tilde{g}(x;w,s).
\end{aligned}$$

Lemma 3.2 ***(Maurey for signed measures)*** Let μ denote a nonzero signed measure supported on $S \subseteq \mathbb{R}^p$, and write $g(x) := \int g(x;w)\mathrm{d}\mu(w)$. Let $(\tilde{w}_1, \ldots, \tilde{w}_k)$ be IID draws from the corresponding $\widetilde{\mu}$, and let P be a probability measure on x. Then

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tilde{w}_1,\ldots,\tilde{w}_k} \left\| g - \frac{1}{k}\sum_i \tilde{g}(\cdot;\tilde{w}_i)\right\|_{L_2(P)}^2 &\leq \frac{\mathbb{E}\,\|\tilde{g}(\cdot;\tilde{w})\|_{L_2(P)}^2}{k} \\
&\leq \frac{\|\mu\|_1^2 \sup_{w\in S}\|g(\cdot;w)\|_{L_2(P)}^2}{k},
\end{aligned}$$

and moreover there exist $(w_1, \ldots, w_k)$ in S and $s \in \{\pm 1\}^m$ with

$$\left\| g - \frac{1}{k}\sum_i \tilde{g}(\cdot;w_i,s_i)\right\|_{L_2(P)}^2 \leq \mathop{\mathbb{E}}_{\tilde{w}_1,\ldots,\tilde{w}_k} \left\| g - \frac{1}{k}\sum_i \tilde{g}(\cdot;\tilde{w}_i)\right\|_{L_2(P)}^2.$$

Proof. By the mean calculation we did earlier, $g = \mathbb{E}_{\widetilde{\mu}} \|\mu\| s g_w = \mathbb{E}_{\widetilde{\mu}} \tilde{g}$, so by the regular Maurey applied to $\widetilde{\mu}$ and Hilbert space $L_2(P)$ (i.e., writing $V := \tilde{g}$ and $g = \mathbb{E}\, V$),

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tilde{w}_1,\ldots,\tilde{w}_k} \left\| g - \frac{1}{k}\sum_i \tilde{g}(\cdot;\tilde{w}_i) \right\|_{L_2(P)}^2 &\leq \frac{\mathbb{E}\,\|\tilde{g}(\cdot;\tilde{w})\|_{L_2(P)}^2}{k} \\
&\leq \frac{\sup_{s\in\{\pm 1\}} \sup_{w\in\mathcal{W}} \left\{ \|\mu\|_1 s g(\cdot;w) \right\}_{L_2(P)}^2}{k}. \\
&\leq \frac{\|\mu\|_1^2 \sup_{w\in S} \|g(\cdot;w)\|_{L_2(P)}^2}{k},
\end{aligned}$$

and the existence of the fixed (w_i, s_i) is also from Maurey.

Example 3.1 ***(various infinite-width sampling bounds)***

1. Suppose $x \in [0,1]$ and f is differentiable. Using our old univariate calculation,

$$f(x) - f(0) = \int_0^1 \mathbf{1}[x \geq b] f'(b) \mathrm{d}b.$$

Let μ denote $f'(b)\mathrm{d}b$; then a sample $((b_i, s_i))_{i=1}^k$ from $\widetilde{\mu}$ satisfies

$$\begin{aligned}
\left\| f(\cdot) - f(0) - \frac{\|\mu\|_1}{k}\sum_i s_i \mathbf{1}[\cdot \geq b_i] \right\|_{L_2(P)}^2 &\leq \frac{\|\mu\|_1^2 \sup_{b\in[0,1]} \|\mathbf{1}[\cdot \geq b]\|_{L_2(P)}^2}{k} \\
&= \frac{1}{k}\left(\int_0^1 |f'(b)| \mathrm{d}b \right)^2.
\end{aligned}$$

2. Now consider the Fourier representation via Barron's theorem:

$$\begin{aligned}
f(x) - f(0) = &-2\pi \iint_0^{\|w\|} \mathbf{1}[w^\mathsf{T} x - b \geq 0] \left[\sin(2\pi b + 2\pi\theta(w)) |\hat{f}(w)| \right] \mathrm{d}b \mathrm{d}w \\
&+ 2\pi \iint_{-\|w\|}^0 \mathbf{1}[-w^\mathsf{T} x + b \geq 0] \left[\sin(2\pi b + 2\pi\theta(w)) |\hat{f}(w)| \right] \mathrm{d}b \mathrm{d}w,
\end{aligned}$$

and also our calculation that the corresponding measure μ on thresholds has $\|\mu\|_1 \leq 2\|\widehat{\nabla f}(w)\|$. Then Maurey's lemma implies that there exist $((w_i, b_i, s_i))_{i=1}^m$ such that, for any probability measure P support on $\|x\| \leq 1$,

$$\begin{aligned}
\left\| f(\cdot) - f(0) - \frac{\|\mu\|_1}{k}\sum_i s_i \mathbf{1}[\langle w_i, \cdot\rangle \geq b_i] \right\|_{L_2(P)}^2 &\leq \frac{\|\mu\|_1^2 \sup_{w,b} \|\mathbf{1}[\langle w, \cdot\rangle \geq b]\|_{L_2(P)}^2}{k} \\
&\leq \frac{4\|\widehat{\nabla f}(w)\|^2}{k}.
\end{aligned}$$

4 Approximation near initialization and the Neural Tangent Kernel

In this section we consider networks close to their random initialization. Briefly, the core idea is to compare a network $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$, which takes input $x \in \mathbb{R}^d$ and has parameters $W \in \mathbb{R}^p$, to its first-order Taylor approximation at random initialization W_0:

$$f_0(x; W) := f(x; W_0) + \langle \nabla f(x; W_0), W - W_0 \rangle .$$

The key property of this simplification is that while it is nonlinear in x, it is affine in W, which will greatly ease analysis. This section is roughly organized as follows

- 4.1 gives the basic setup in more detail, including the networks considered and the specific random initialization. The study is almost solely on shallow networks, since the deep case currently only leads to a degraded analysis and is not well understood.
- 4.2 shows that near initialization, with large width, $f \approx f_0$.
- 4.3 provides the "kernel view": since the previous part shows we are effectively linear over some feature space, it is natural to consider the kernel corresponding to that feature space. This provides many connections to the literature (via "neural tangent kernel" (NTK)), and also is used for a short proof that these functions near initialization are already universal approximators!

4.1 Basic setup: Taylor expansion of shallow networks

As explained shortly, we will almost solely consider the shallow case:

$$f(x; W) := \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \sigma(w_j^\mathsf{T} x), \qquad W := \begin{bmatrix} \leftarrow w_1^\mathsf{T} \rightarrow \\ \vdots \\ \leftarrow w_m^\mathsf{T} \rightarrow \end{bmatrix} \in \mathbb{R}^{m \times d}, \tag{2}$$

where σ will either be a smooth activation or the ReLU, and we will treat $a \in \mathbb{R}^m$ as fixed and only allow $W \in \mathbb{R}^{m \times d}$ to vary. There are a number of reasons for this exact formalism, they are summarized below in Remark 4.1.

Now let's consider the corresponding first-order Taylor approximation f_0 in detail. Consider any univariate activation σ which is differentiable except on a set of measure zero (e.g., countably many points), and Gaussian initialization $W_0 \in \mathbb{R}^{m \times d}$ as before. Consider the Taylor expansion at initialization:

$$\begin{aligned} f_0(x; W) &= f(x; W_0) + \langle \nabla f(x; W_0), W - W_0 \rangle \\ &= \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \left(\sigma(w_{0,j}^\mathsf{T} x) + \sigma'(w_{0,j}) x^\mathsf{T} (w_j - w_{0,j}) \right) \\ &= \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \left(\left[\sigma(w_{0,j}^\mathsf{T} x) - \sigma'(w_{0,j}) w_{0,j}^\mathsf{T} x \right] + \sigma'(w_{0,j}) w_j^\mathsf{T} x \right) . \end{aligned}$$

If σ is nonlinear, then this mapping is nonlinear in x, despite being affine in W! Indeed $\nabla f(\cdot; W_0)$ defines a feature mapping:

$$\nabla f(x; W_0) := \begin{bmatrix} \leftarrow & a_1 \sigma'(w_{0,1}^\mathsf{T} x) x^\mathsf{T} & \rightarrow \\ & \vdots & \\ \leftarrow & a_m \sigma'(w_{0,m}^\mathsf{T} x) x^\mathsf{T} & \rightarrow \end{bmatrix};$$

the predictor f_0 is affine is an affine function of the parameters, and is also affine in this feature-mapped data.

Remark 4.1 ***(rationale for eq. 2)*** The factor $\frac{1}{\sqrt{m}}$ will make the most sense in 4.3, it gives a normalization that leads to a kernel. We only vary the inner layer W and keep the outer layer a fixed to have a nontrivial (nonlinear) model which still leads to non-convex training, but is arguably the simplest such. Random initialization is classical and used for many reasons, a classical one being a "symmetry break" which makes nodes distinct and helps with training (Hertz, Krogh, and Palmer 1991). A standard initialization, is to have the first layer Gaussian with standard deviation $1/\sqrt{d}$, and the second layer Gaussian with standard deviation $1/\sqrt{m}$; in the case of the ReLU, positive constants can be pulled through, and equivalently we can use standard Gaussian initialization and place a coefficient $1/\sqrt{md}$ out front; here we drop the $1/\sqrt{d}$ since we want to highlight a behavior that varies with m, whereas $1/\sqrt{d}$ is a constant. To simplify further, we will make the second layer ± 1. `pytorch` initialization defaults to these standard deviations, but defaults to uniform distributions and not Gaussians. Lastly, some papers managed to set up analysis so that the final layer does most of the work (and the training problem is convex for the last layer), thus we follow the convention of some authors to train all but the last layer to rule out this possibility.

Remark 4.2 Many researchers use the term "overparameterization" to refer to a number of phenomena, but rooted at their core in the use of many more parameters than are seemingly necessary. E.g., we know from the earlier sections that some number of nodes suffice to approximate certain types of functions, but in this section we see we might as well take the width m arbitrarily large. Mainly classical perspectives on the behavior of networks (e.g., their generalization properties) worsen with large width, so "overparameterization" also highlights many of these apparent contradictions.

Remark 4.3 ***(main bibliography for NTK)*** The paper that made the term "NTK" is (Jacot, Gabriel, and Hongler 2018), which also argued gradient descent follows the NTK; a kernel connection was observed earlier in (Cho and Saul 2009).

Another very influential work is (Allen-Zhu, Li, and Song 2018), which showed that one can achieve arbitrarily small training error by running gradient descent on a large-width network, which thus stays within the NTK regime (close to initialization).

A few other early optimization references are (Simon S. Du et al. 2018) (Arora, Du, Hu, Li, and Wang 2019) (Allen-Zhu, Li, and Liang 2018). Also nearly parallel with (Jacot, Gabriel, and Hongler 2018) were (Li and Liang 2018; Simon S. Du et al. 2018).

Estimates of empirical infinite width performance are in (Arora, Du, Hu, Li, Salakhutdinov, et al. 2019) and (Novak et al. 2018).

(Various further works.) The NTK has appeared in a vast number of papers (and various papers use linearization and study the early stage of training, whether they refer to it as the NTK or not). Concurrent works giving general convergence to global minima are (Simon S. Du et al. 2018; Allen-Zhu, Li, and Liang 2018; Oymak and Soltanolkotabi 2019; Zou et al. 2018). Many works subsequently aimed to reduce the width dependence (Zou and Gu 2019; Oymak and Soltanolkotabi 2019); in the classification case, a vastly smaller width is possible (Ji and Telgarsky 2019a; Z. Chen et al. 2019). Another subsequent direction (in the regression case) was obtaining test error and not just training error bounds (Cao and Gu 2020b, 2020a; Arora, Du, Hu, Li, and Wang 2019). Lastly, another interesting point is the use of noisy gradient

desceent in some of these analyses (Allen-Zhu, Li, and Liang 2018; Z. Chen et al. 2020).

Some works use the term F_2 to refer to the kernel space we get after taking a Taylor expansion, and also constrast this with the space F_1 we get byu considering all possible neural networks (e.g., those that are a discrete sum of nodes, which can not be represented exactly with a finite-norm element of the RKHS F_2); this term mostly appears in papers with Francis Bach, see for instance Chizat and Bach (2020).

Remark 4.4 ***(scaling and temperature)*** Some authors including a multiplicative factor $\epsilon > 0$ on the network output, meaning

$$\frac{\epsilon}{\sqrt{m}} \sum_{j=1}^{m} a_j \sigma(w_j^\intercal x).$$

Considering the effect of introducing ϵ in f_0 as well, one can interpret this as having two effects:

- Scaling down the initial random predictions $f(x; W_0) = f_0(x; W_0)$. These initial predictions are random and, without $\epsilon > 0$, are of order $\mathcal{O}(1)$; it therefore takes gradient descent quite a bit of work just to zero out this initial random noise. All together, papers, deal with this random noise in effectively four ways:
  (a) using $\epsilon > 0$ as described here,
  (b) using "symmetric" initialization which forces $f(x; W_0) = 0$ in various ways,
  (c) simply running more gradient descent to clear the noise, which in turn may require larger width,
  (d) considering the well-separated classification setting, which does not need to fully clear the noise, but rather just "push it to one side."
- Scaling down the gradient. As such, many works which do not have $\epsilon > 0$ instead use a small step size, e.g., $1/\sqrt{m}$.

Some authors fix ϵ as a function of m and consider a resulting "scaling" behavior, namely that by taking $m \to 0$, the Taylor expansion "zooms in," and this provides one explanation of the behavior of the NTK; this perspective was summarized in (Chizat and Bach 2019).

Remark 4.5 ***(Practical regimes)*** "NTK regime" or "near initialization" are not well-defined, though generally the proofs in this setup require some combination of $\|W - W_0\|_{\mathrm{F}} = \mathcal{O}(1)$ (or the stronger form $\max_j \|\mathbf{e}_j^\intercal (W - W_0)\|_2 = \mathcal{O}(1/\sqrt{m})$), and/or at most $1/\sqrt{m}$ fraction of the activations change. In practice, these all seem to be violately almost immediately (e.g., just one or two steps of gradient descent), but still the idea captures many interesting phenomena near initialization and do not degrade with overparameterization as do other approaches.

Remark 4.6 ***(multi-layer case)*** Let $\vec{W} = (W_L, \ldots, W_1)$ denote a tuple of the parameters for each layer, whereby the Taylor expansion at initial values $\vec{W}_0$ now becomes

$$x \mapsto f(x; \vec{W}) + \left\langle \vec{W} - \vec{W}_0, \nabla f(x; \vec{W}_0) \right\rangle.$$

The inner product with $\nabla f(x; \vec{W}_0)$ decomposes over layers, giving

$$\left\langle \vec{W} - \vec{W}_0, \nabla f(x; \vec{W}_0) \right\rangle = \sum_{k=1}^{L} \left\langle W_k - W_{0,k}, \nabla_{W_k} f(x; \vec{W}_0) \right\rangle.$$

We will revisit this multi-layer form later when discussing kernels.

Remark 4.7 ***(Taylor expansion around*** 0***)*** There are a few reasons why we do the Taylor expansion around initialization; the main one is that Taylor approximation improves the closer you get to the point you are approximating, another one is that bounds that scale with $\|W\|_{\mathrm{F}}$ can be re-centered to now scale with the potentially much smaller quantity $\|W - W_0\|_{\mathrm{F}}$, and lastly we get to invoke Gaussian concentration tools. Note however how things completely break down if we do what might initially seem a reasonable alternative: Taylor expansion around 0. Then we get

$$f(x;0) + \langle \nabla f(x;0), W - 0 \rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \left(\sigma(0) + \sigma'(0) x^{\mathsf{T}} w_j \right).$$

This is once again affine in the parameters, but it is also affine in the inputs! So we don't have any of the usual power of neural networks.

Remark 4.8 ***(simplification with the ReLU)*** If we use the ReLU $\sigma(z) = \max\{0, z\}$, then the property $\sigma(z) = z\sigma'(z)$ (which is fine even at 0!) means

$$\sigma(w_{0,j}^{\mathsf{T}} x) - \sigma'(w_{0,j}^{\mathsf{T}} x) w_{0,j}^{\mathsf{T}} x = 0,$$

and thus f_0 as above simplifies to give

$$\begin{aligned} f_0(x;W) &= \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \left(\left[\sigma(w_{0,j}^{\mathsf{T}} x) - \sigma'(w_{0,j}) w_{0,j}^{\mathsf{T}} x \right] + \sigma'(w_{0,j}) w_j^{\mathsf{T}} x \right) \\ &= \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \sigma'(w_{0,j}) w_j^{\mathsf{T}} x = \langle \nabla f(x; W_0), W \rangle . \end{aligned}$$

4.2 Networks near initialization are almost linear

Our first step is to show that $f - f_0$ *shrinks* as m increases, which has a few immediate consequences.

- It gives one benefit of "overparameterization."
- It gives us an effective way to do universal approximation with small $\|W - W_0\|$: we simply make m as large as needed and get more functions inside our RKHS.

First we handle the case that σ is smooth, by which we mean σ'' exists and satisfies $|\sigma''| \le \beta$ everywhere. This is not satisfied for the ReLU, but the proof is so simple that it is a good motivator for other cases.

Proposition 4.1 If $\sigma : \mathbb{R} \to \mathbb{R}$ is β-smooth, and $|a_j| \le 1$, and $\|x\|_2 \le 1$, then for any parameters $W, V \in \mathbb{R}^{m \times d}$,

$$|f(x;W) - f_0(x;V)| \le \frac{\beta}{2\sqrt{m}} \|W - V\|_{\mathrm{F}}^2 .$$

Proof. By Taylor's theorem,

$$|\sigma(r) - \sigma(s) - \sigma'(s)(r - s)| = \left| \int_r^s \sigma''(z)(s - z) \mathrm{d}z \right| \le \frac{\beta (r - s)^2}{2}.$$

Therefore

$$\begin{aligned}
&|f(x;W) - f(x;V) - \langle \nabla f(x;V), W - V\rangle| \\
&\le \frac{1}{\sqrt{m}} \sum_{j=1}^{m} |a_j| \cdot \left|\sigma(w_j^\intercal x) - \sigma(v_j^\intercal x) - \sigma'(v_j^\intercal x) x^\intercal (w_j - v_j)\right| \\
&\le \frac{1}{\sqrt{m}} \sum_{j=1}^{m} \frac{\beta (w_j^\intercal x - v_j^\intercal x)^2}{2} \\
&\le \frac{\beta}{2\sqrt{m}} \sum_{j=1}^{m} \|w_j - v_j\|^2 \\
&= \frac{\beta}{2\sqrt{m}} \|W - V\|_{\mathrm{F}}^2 .
\end{aligned}$$

Remark 4.9 The preceding lemma holds for any W, and doesn't even need the Gaussian structure of W_0. This is unique to this shallow case, however; producing an analogous inequality with multiple layers of smooth activations will need to use random initialization.

Now we switch to the ReLU. The proof is much more complicated, but is instructive of the general calculations one must perform frequently with the ReLU.

Remark 4.10 A multi-layer version of the following originally appared in (Allen-Zhu, Li, and Song 2018); there, the multiple layers only *hurt* the bound, introducing factors based on depth. Moreover, the proof is much more complicated. Due to this, we only use a straightforward single-layer version, which appeared later in (Ji, Li, and Telgarsky 2021).

Lemma 4.1 For any radius $B \ge 0$, for any fixed $x \in \mathbb{R}^d$ with $\|x\| \le 1$, with probability at least $1 - \delta$ over the draw of W_0, for any $W \in \mathbb{R}^{m \times d}$ with $\|W - W_0\|_{\mathrm{F}} \le B$,

$$|f(x;W) - f_0(x;W)| \le \frac{2B^{4/3} + B\ln(1/\delta)^{1/4}}{m^{1/6}},$$

and given any additional $V \in \mathbb{R}^{m \times d}$ with $\|V - W_0\|_{\mathrm{F}} \le B$,

$$|f(x;V) - (f(x;W) + \langle \nabla_W f(x;W), V - W\rangle)| \le \frac{6B^{4/3} + 2B\ln(1/\delta)^{1/4}}{m^{1/6}}.$$

Remark 4.11 ***(incorrect approach)*** Let's see how badly things go awry if we try to brute-force the proof. By similar reasoning to the earlier ReLU simplification,

$$\begin{aligned}
|f(x;W) - f_0(x;W)| &= |\langle \nabla f(x;W), W\rangle - \langle \nabla f(x;W_0), W\rangle| \\
&= \left| \frac{1}{\sqrt{m}} \sum_j a_j \left(\mathbf{1}[w_j^\intercal x \ge 0] - \mathbf{1}[w_{0,j}^\intercal x \ge 0 \right) w_j^\intercal x \right|.
\end{aligned}$$

A direct brute-forcing with no sensitivity to random initialization gives

$$|f(x;W) - f_0(x;W)| \le \frac{1}{\sqrt{m}} \sum_j \|w_j\| \le \|W\|_{\mathrm{F}}.$$

We can try to save a bit by using the randomness of $(a_j)_{j=1}^m$, but since Lemma 4.1 is claimed to hold for every $\|W - W_0\|_{\mathrm{F}} \le B$, the argument might be complicated. Our eventual proof will only use randomness of W_0.

The proof will use the following concentration inequality.

Lemma 4.2 For any $\tau > 0$ and $x \in \mathbb{R}^d$ with $\|x\| > 0$, with probability at least $1 - \delta$,

$$\sum_{j=1}^{m} \mathbf{1}\left[|w_j^\mathsf{T} x| \le \tau \|x\|\right] \le m\tau + \sqrt{\frac{m}{2} \ln \frac{1}{\delta}}.$$

Proof. For any row j, define an indicator random variable

$$P_j := \mathbf{1}[|w_j^\mathsf{T} x| \le \tau \|x\|].$$

By rotational invariance, $P_j = \mathbf{1}[|w_{j,1}| \le \tau]$, which by the form of the Gaussian density gives

$$\Pr[P_j = 1] = \int_{-\tau}^{+\tau} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \mathrm{d}z \le \frac{2\tau}{\sqrt{2\pi}} \le \tau.$$

By Hoeffding's inequality, with probability at least $1 - \delta$,

$$\sum_{j=1}^{m} P_j \le m \Pr[P_1 = 1] + \sqrt{\frac{m}{2} \ln \frac{1}{\delta}} \le m\tau + \sqrt{\frac{m}{2} \ln \frac{1}{\delta}}.$$

Proof of Lemma 4.1. Fix $x \in \mathbb{R}^d$. If $\|x\| = 0$, then for any $W \in \mathbb{R}^d$, $f(x; W) = 0 = f_0(x; W)$, and the proof is complete; henceforth consider the case $\|x\| > 0$.
The proof idea is roughly as follows. The Gaussian initialization on W_0 concentrates around a rather large shell, and this implies $|w_{0,j}^\mathsf{T} x|$ is large with reasonably high probability. If $\|W - W_0\|_\mathrm{F}$ is not too large, then $\|w_j - w_{0,j}\|$ must be small for most coordinates; this means that $w_j^\mathsf{T} x$ and $w_{0,j}^\mathsf{T} x$ must have the same sign for most j.
Proceeding in detail, fix a parameter $r > 0$, which will be optimized shortly. Let W be given with $\|W - W_0\| \le B$. Define the sets

$$\begin{aligned} S_1 &:= \left\{ j \in [m] : |w_j^\mathsf{T} x| \le r\|x\| \right\}, \\ S_2 &:= \left\{ j \in [m] : \|w_j - w_{0,j}\| \ge r \right\}, \\ S &:= S_1 \cup S_2. \end{aligned}$$

By Lemma 4.2, with probability at least $1 - \delta$,

$$|S_1| \le rm + \sqrt{m \ln(1/\delta)}.$$

On the other hand,

$$B^2 \ge \|W - W_0\|^2 \ge \sum_{j \in S_2} \|w_j - w_{0,j}\|^2 \ge |S_2| r^2,$$

meaning $|S_2| \le B^2/r^2$. For any $j \notin S$, if $w_j^\mathsf{T} x > 0$, then

$$w_{0,j}^\mathsf{T} x \ge w_j^\mathsf{T} x - \|w_j - w_{0,j}\| \cdot \|x\| > \|x\| (r - r) = 0,$$

meaning $\mathbf{1}[w_j^\mathsf{T} x \ge 0] = \mathbf{1}[w_{0,j}^\mathsf{T} x \ge 0]$; the case that $j \notin S$ and $w_j^\mathsf{T} x < 0$ is analogous. Together,

$$|S| \le rm + \sqrt{m \ln(1/\delta)} + \frac{B^2}{r^2} \quad \text{and} \quad j \notin S \Longrightarrow \mathbf{1}[w_j^\mathsf{T} x \ge 0] = \mathbf{1}[w_{0,j}^\mathsf{T} x \ge 0].$$

Lastly, we can finally choose r to balance terms in $|S|$: picking $r := B^{2/3}/m^{1/3}$ gives

$$|S| \leq (Bm)^{2/3} + \sqrt{m \ln(1/\delta)} + (Bm)^{2/3} \leq m^{2/3}\left(2B^{2/3} + \sqrt{\ln(1/\delta)}\right).$$

Now that $|S|$ has been bounded, the proof considers the two different statements separately, though their proofs are similar.

1. As in the above remark,

$$\begin{aligned}
|f(x;W) - f_0(x;W)| &= |\langle \nabla f(x;W) - \nabla f(x;W_0), W\rangle| \\
&= \frac{1}{\sqrt{m}}\left|\sum_j a_j\left(\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[w_{0,j}^{\mathsf{T}}x \geq 0]\right) w_j^{\mathsf{T}}x\right| \\
&\leq \frac{1}{\sqrt{m}}\sum_j \left|\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[w_{0,j}^{\mathsf{T}}x \geq 0]\right| \cdot |w_j^{\mathsf{T}}x|.
\end{aligned}$$

   To simplify this, as above $\left|\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[w_{0,j}^{\mathsf{T}}x \geq 0]\right|$ is only nonzero for $j \in S$. But when it is nonzero, this means $\operatorname{sgn}(w_j^{\mathsf{T}}x) \neq \operatorname{sgn}(w_{0,j}^{\mathsf{T}}x)$, and thus $|w_j^{\mathsf{T}}x| \leq |w_j^{\mathsf{T}}x - w_{0,j}^{\mathsf{T}}x|$, and together with Cauchy-Schwarz (two applications!), and the above upper bound on $|S|$ gives

$$\begin{aligned}
|f(x;W) - f_0(x;W)| &\leq \frac{1}{\sqrt{m}}\sum_j \left|\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[w_{0,j}^{\mathsf{T}}x \geq 0]\right| \cdot |w_j^{\mathsf{T}}x| \\
&\leq \frac{1}{\sqrt{m}}\sum_{j\in S} \left|w_j^{\mathsf{T}}x - w_{0,j}^{\mathsf{T}}x\right| \\
&\leq \frac{1}{\sqrt{m}}\sum_{j\in S} \|w_j - w_{0,j}\| \\
&\leq \frac{1}{\sqrt{m}}\sqrt{|S|} \cdot \|W - W_0\|_{\mathrm{F}} \\
&\leq B\sqrt{\frac{|S|}{m}} \\
&\leq B\sqrt{\frac{2B^{2/3} + \sqrt{\ln(1/\delta)}}{m^{1/3}}} \\
&\leq \frac{2B^{4/3} + B\ln(1/\delta)^{1/4}}{m^{1/6}}.
\end{aligned}$$

2. Following similar reasoning,

$$\begin{aligned}
&|f(x;V) - (f(x;W) + \langle \nabla_W f(x;W), V - W\rangle)| \\
&= |\langle \nabla f(x;V) - \nabla f(x;W), V\rangle| \\
&= \frac{1}{\sqrt{m}}\left|\sum_j a_j\left(\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[v_j^{\mathsf{T}}x \geq 0]\right) v_j^{\mathsf{T}}x\right| \\
&\leq \frac{1}{\sqrt{m}}\left|\sum_j a_j\left(\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[v_j^{\mathsf{T}}x \geq 0]\right)\right| \cdot \left|w_j^{\mathsf{T}}x - v_j^{\mathsf{T}}x\right| \\
&\leq \frac{1}{\sqrt{m}}\left|\sum_j a_j\left(\mathbf{1}[w_j^{\mathsf{T}}x \geq 0] - \mathbf{1}[v_j^{\mathsf{T}}x \geq 0]\right)\right| \cdot \|w_j - v_j\|.
\end{aligned}$$

Now define S_3 analogously to S_2, but for the new matrix V:

$$S_3 := \{j \in [m] : \|v_j - w_{0,j}\| \geq r\},$$

and additionally define

$$S_4 := S_1 \cup S_2 \cup S_3.$$

By the earlier choice of r and related calculations, with probability at least $1 - \delta$,

$$|S| \leq rm + \sqrt{m \ln(1/\delta)} + \frac{2B^2}{r^2} \leq m^{2/3}\left(3B^{2/3} + \sqrt{\ln(1/\delta)}\right).$$

Plugging this back in and continuing as before,

$$\begin{aligned}
|\langle \nabla f(x; V) - \nabla f(x; W), V\rangle| &\leq \frac{1}{\sqrt{m}}\left|\sum_j a_j \left(\mathbf{1}[w_j^T x \geq 0] - \mathbf{1}[v_j^T x \geq 0]\right)\right| \cdot \|w_j - v_j\|. \\
&\leq \frac{1}{\sqrt{m}} \sum_{j \in S_4} \|w_j - v_j\|_{\mathrm{F}} \\
&\leq \frac{1}{\sqrt{m}} \sqrt{|S_4|} \|V - W\|_{\mathrm{F}} \\
&\leq 2B \sqrt{\frac{3B^{2/3} + \sqrt{\ln(1/\delta)}}{m^{1/3}}} \\
&\leq \frac{6B^{4/3} + 2B \ln(1/\delta)^{1/4}}{m^{1/6}}.
\end{aligned}$$

4.3 Properties of the kernel at initialization

So far, we've said that $f - f_0$ is small when the width is large. Now we will focus on f_0, showing that it is a large class of functions; thus, when the width is large, f obtained with small $\|W - W_0\|_{\mathrm{F}}$ can also capture many functions.

Remark 4.12 ***(kernel view)*** This analysis will take the kernel/RKHS view of f_0. The amount that this perspective appears varies by treatments near initialization, including papers which never explicitly use any kernel concepts. In the original paper giving the name "NTK" (Jacot, Gabriel, and Hongler 2018), only f_0 (and not f) was considered, indeed in the multi-layer case, and in the infinite-width case, using a Gaussian process with a kernel given as here. We won't use this perspective here.

To start, let us see how to define a kernel. In the standard kernel setup, the kernel can be written

as the inner product between feature mappings for two data points:

$$\begin{aligned}
k_m(x,x') &:= \langle \nabla f(x;W_0), \nabla f(x';W_0) \rangle \\
&= \left\langle \begin{bmatrix} \longleftarrow & a_1 x^\mathsf{T} \sigma'(w_{1,0}^\mathsf{T} x)/\sqrt{m} & \longrightarrow \\ & \vdots & \\ \longleftarrow & a_m x^\mathsf{T} \sigma'(w_{m,0}^\mathsf{T} x)/\sqrt{m} & \longrightarrow \end{bmatrix}, \begin{bmatrix} \longleftarrow & a_1 (x')^\mathsf{T} \sigma'(w_{1,0}^\mathsf{T} x')/\sqrt{m} & \longrightarrow \\ & \vdots & \\ \longleftarrow & a_m (x')^\mathsf{T} \sigma'(w_{m,0}^\mathsf{T} x')/\sqrt{m} & \longrightarrow \end{bmatrix} \right\rangle \\
&= \frac{1}{m} \sum_{j=1}^m a_j^2 \left\langle x\sigma'(w_{j,0}^\mathsf{T} x), x'\sigma'(w_{j,0}^\mathsf{T} x') \right\rangle \\
&= x^\mathsf{T} x' \left[\frac{1}{m} \sum_{j=1}^m \sigma'(w_{j,0}^\mathsf{T} x) \sigma'(w_{j,0}^\mathsf{T} x') \right].
\end{aligned}$$

This gives one justification of the $1/\sqrt{m}$ factor: now this kernel is an average and not a sum, and we should expect it to have a limit as $m \to \infty$. To this end, and noting that the rows $(w_{0,j}^\mathsf{T})_{j=1}^m$ are iid, then each term of the summation is iid, so by the SLLN, almost surely

$$k_m(x,x') \xrightarrow{m\to\infty} k(x,x') := x^\mathsf{T} x' \mathop{\mathbb{E}}_w \left[\sigma'(w^\mathsf{T} x) \sigma'(w^\mathsf{T} x') \right].$$

In homework we will (a) provide a more explicit form as *dot product kernel*, and (b) bound the difference exactly. [mjt☹: add explicit ref.]

For now, let us calculate the closed form for the ReLU; let's do this geometrically. [mjt☹: need to include picture proof]

- Consider the plane spanned by x and x'. Since projections of standard Gaussians are again standard Gaussians, we can consider a Gaussian random vector $v \in \mathbb{R}^2$ in this plane.
- The integrand in the expectation is 1 iff $v^\mathsf{T} x \geq 0$ and $v^\mathsf{T} x' \geq 0$. Since $\|v\|$ does not affect these expressions, we can simplify $v \in \mathbb{R}^2$ further to be sampled uniformly from the surface of the sphere.
- Suppose $\|x\| = 1 = \|x'\|$, and define $\theta := \arccos(x^\mathsf{T} x')$; then the integrand is 1 if v has positive inner product with both x and x', which has probability

$$\frac{\pi - \theta}{2\pi}.$$

Together, still using $\|x\| = 1 = \|x'\|$,

$$k(x,x') = x^\mathsf{T} x' \mathop{\mathbb{E}}_w \mathbf{1}[w^\mathsf{T} x \geq 0] \cdot \mathbf{1}[w^\mathsf{T} x' \geq 0] = x^\mathsf{T} x' \left(\frac{\pi - \arccos(x^\mathsf{T} x')}{2\pi} \right).$$

Remark 4.13 ***(multi-layer kernel)*** Let's revisit the multi-layer case, and develop the multi-layer kernel. Suppose the width of every layer except the final one is m, specifically $W_1 \in \mathbb{R}^{m\times d}$, and $W_L \in \mathbb{R}^{1\times m}$, and otherwise $W_i \in \mathbb{R}^{m\times m}$. Then the kernel also decomposes over layers, giving

$$\begin{aligned}
\tilde{k}_m(x,x') &:= \left\langle \nabla f(x;\vec{W}_0), \nabla f(x';\vec{W}_0) \right\rangle \\
&:= \sum_{i=1}^L \left\langle \nabla_{W_i} f(x;\vec{W}_0), \nabla_{W_i} f(x';\vec{W}_0) \right\rangle.
\end{aligned}$$

It is not clear how powerful this representation is, and if it is fundamentally more powerful than the single-layer version. On the one hand, it decomposes over layers and is thus a sum (and not composition) of kernels; on the other hand, each layer does work with the forward mapping of previous layers. There is some work on this topic, though it is far from closing the question (Bietti and Bach 2020); meanwhile, the linearization inequalities in section 4.2 seemingly degrade with depth, so the tradeoffs could be intricate, and also could put serious question on how much the early phase near initialization is relevant in practice.

Remark 4.14 ***(kernel of Taylor expansion at*** 0***)*** Let's also revisit the Taylor expansion at 0, but now with kernels. Before, we noted that the feature expansion is *linear*, rather than non-linear, in the data:

$$\nabla f(x;0) = \begin{bmatrix} \leftarrow & a_1\sigma'(0)x^\mathsf{T}/\sqrt{m} & \rightarrow \\ & \vdots & \\ \leftarrow & a_m\sigma'(0)x^\mathsf{T}/\sqrt{m} & \rightarrow \end{bmatrix} = \frac{\sigma'(0)}{\sqrt{m}} \begin{bmatrix} \leftarrow & a_1x^\mathsf{T} & \rightarrow \\ & \vdots & \\ \leftarrow & a_mx^\mathsf{T} & \rightarrow \end{bmatrix};$$

as mentioned before, this is in contrast to the Taylor expansion at initialization, which is nonlinear in the data. Moreover, the corresponding kernel is a rescaling of the linear kernel:

$$\begin{aligned} \langle \nabla f(x;0), \nabla f(x';0) \rangle &= \frac{\sigma'(0)^2}{m} \left\langle \begin{bmatrix} \leftarrow & a_1x^\mathsf{T} & \rightarrow \\ & \vdots & \\ \leftarrow & a_mx^\mathsf{T} & \rightarrow \end{bmatrix}, \begin{bmatrix} \leftarrow & a_1(x')^\mathsf{T} & \rightarrow \\ & \vdots & \\ \leftarrow & a_m(x')^\mathsf{T} & \rightarrow \end{bmatrix} \right\rangle \\ &= \frac{\sigma'(0)^2}{m} \sum_{j=1}^m a_j^2 x^\mathsf{T}x' = \sigma'(0)^2 x^\mathsf{T}x'. \end{aligned}$$

Now let's return to the task of assessing how many functions we can represent near initialization. For this part, we will fix one degree of freedom in the data to effectively include a bias term; this is not necessary, but gives a shorter proof by reducing to standard kernel approximation theorems. We will show that this class is a universal approximator. Moreover $\|W - V\|$ will correspond to the RKHS norm, thus by making the width large, we can approximate elements of this large RKHS arbitrarily finely.

Proceeding in detail, first let's define our domain

$$\mathcal{X} := \left\{ x \in \mathbb{R}^d : \|x\| = 1, x_d = 1/\sqrt{2} \right\},$$

and our predictors

$$\mathcal{H} := \left\{ x \mapsto \sum_{j=1}^m \alpha_j k(x, x_j) \ : \ m \geq 0, \alpha_j \in \mathbb{R}, x_j \in \mathcal{X} \right\}.$$

This might look fancy, but is the same as the functions we get by starting with $x \mapsto \langle \nabla f(x; W_0), W - W_0 \rangle$ and allowing the width to go to infinity, and $\|W - W_0\|$ be arbitrarily large; by the results in section 4.2, we can always choose an arbitrarily large width so that $f - f_0 \approx 0$ even when $\|W - W_0\|$ is large, and we will also show that large width approximates infinite width in the homework. As such, it suffices to show that $\mathcal{H}$ is a universal approximator over $\mathcal{X}$.

Theorem 4.1 $\mathcal{H}$ is a universal approximator over $\mathcal{X}$; that is to say, for every continuous $g : \mathbb{R}^d \to \mathbb{R}$ and every $\epsilon > 0$, there exists $h \in \mathcal{H}$ with $\sup_{x \in \mathcal{X}} |g(x) - h(x)| \leq \epsilon$.

Remark 4.15 The use of a bias is only to conveniently reduce to an existing result about kernels which are universal approximators. This result is stated over full-dimensional sets, and for this case the bias seems necessary. However, if we restrict to $\|x\| = 1$, the bias should not be necessary, though none of these automatic kernel theorems seem to apply (Notes to chapter 4, Steinwart and Christmann 2008).

Proof. Consider the set $U := \{u \in \mathbb{R}^{d-1} : \|u\|^2 \leq 1/2\}$, and the kernel function

$$k(u, u') := f(u^\intercal u'), \qquad f(z) := \frac{(z + 1/2)}{2} - \frac{(z + 1/2)\arccos(z + 1/2)}{2\pi}.$$

We will show that this kernel is a universal approximator over U, which means it is also a universal approximator on its boundary $\{u \in \mathbb{R}^{d-1} : \|u\|^2 = 1/2\}$, and thus the kernel

$$(x, x') \mapsto \frac{x^\intercal x'}{\pi} - \frac{x^\intercal x'}{\arccos(x^\intercal x')} 2\pi$$

is a universal approximator over $\mathcal{X}$.
Going back to the original claim, first note that arccos has the Maclaurin series

$$\arccos(z) = \frac{\pi}{2} - \sum_{k \geq 0} \frac{(2k)!}{2^{2k}(k!)^2} \left(\frac{z^{2k+1}}{2k + 1} \right),$$

which is convergent for $z \in [-1, +1]$. From here, it can be checked that f has a Maclaurin series where every term is not only nonzero, but positive (adding the bias ensured this). This suffices to ensure that k is a universal approximator (Corollary 4.57, Steinwart and Christmann 2008).

We have not quite closed the loop, as we have not combined the pieces to show that for any continuous function g, we can select a large width m and W so that $g \approx f_0(\cdot; W) \approx f(\cdot; W)$, but we've done most of the work, and a few remaining steps will be in homework. For a direct argument about this using a different approach based on (Barron 1993), see (Ji, Telgarsky, and Xian 2020).

5 Benefits of depth

So far we have given no compelling presentation of depth; in particular we have not justified the high depths used in practice.

In this section, we will give constructions of interesting functions by deep networks which can not be approximated by polynomially-sized shallow networks. These are only constructions, and it is unlikely these network structure are found by gradient descent and other practical methods, so the general question of justifying the high depth and particular architectures used in practice is still open.

There are four subsections to these notes.

1. First we will construct a simple piecewise-affine function, $\Delta : \mathbb{R} \to \mathbb{R}$, which will be our building block of more complex behavior. When Δ is composed with itself, it builds complexity exponentially fast in a variety of natural notions (e.g., exponentially many copies of itself).
2. Then we will show that Δ^{L^2} can be easily written as a deep but constant width network,

whereas a shallow network needs exponential width even for approximation within a constant.

3. Then we will use Δ^L to approximate x^2; this is meaningful because it leadsto many other approximations, and may seem more natural than Δ^L.
4. Lastly we will use x^2 to approximate polynomials and Taylor expansions (Sobolev spaces).

5.1 The humble Δ mapping.

Consider the Δ function:

$$\Delta(x) = 2\sigma_r(x) - 4\sigma_r(x-1/2) + 2\sigma_r(x-1) = \begin{cases} 2x & x \in [0, 1/2), \\ 2-2x & x \in [1/2, 1), \\ 0 & \text{otherwise.} \end{cases}$$

How does Δ look? And how about $\Delta^2 := \Delta \circ \Delta$? And Δ^3? [mjt☹: Picture drawn in class; figures forthcoming.]

The pattern is that Δ^L has 2^{L-1} copies of it self, uniformly shrunk down. In a sense, complexity has increased exponentially as a function of the the number of nodes and layers (both $\mathcal{O}(L)$). Later, it will matter that we not only have many copies, but that they are identical (giving uniform spacing). For now, here's one way to characerize this behavior.

Let $\langle x \rangle = x - \lfloor x \rfloor$ denote fractional part.

Proposition 5.1 Let $\langle x \rangle := x - \lfloor x \rfloor$ denote the fractional part of $x \in \mathbb{R}$. Then

$$\Delta^L(x) = \Delta(\left\langle 2^{L-1}x \right\rangle) = \Delta(2^{L-1}x - \lfloor 2^{L-1}x \rfloor).$$

Remark 5.1 ***(applications of Δ)***

- Δ^L creates 2^L (forward and backward) copies of its input, and thus is generally useful to replicate its input.
- Parity on the hypercube in dimension $d = 2^L$: $\prod_{i=1}^d x_i = \Delta^{L-1}\left(\frac{d+\sum_i x_i}{2d}\right)$.
- We'll use Δ when constructing $(x, y) \mapsto xy$.
- Digit extraction! (Which appears a lot in deep network lower and upper bounds!) (See also the Turing machine constructions in (Siegelmann and Sontag 1994, Figure 3) and elsewhere.)

Remark 5.2 ***(bibliography)*** I'm not sure what to cite for the study of the iterated composition Δ^L and its interesting properties. The perspective here is the one from (Telgarsky 2015, 2016), but probably it exists somewhere earlier. E.g., Δ^L is similar to iterated applications of the logistic map in dynamical systems,which was studied at latest in the 1940s.

Proof of Proposition 5.1. The proof proceeds by induction on $L = i$.
For the base case $i = 1$, if $x \in [0, 1)$ then directly

$$\Delta^1(x) = \Delta(x) = \Delta(\langle x \rangle) = \Delta(\left\langle 2^0 x \right\rangle),$$

whereas $x = 1$ means $\Delta^1(x) = \Delta(0) = \Delta(\langle 2^0 x \rangle)$.
For the inductive step, consider Δ^{i+1}. The proof can proceed by peeling individual Δ from the left or from the right; the choice here is to peel from the right. Consider two cases.

- If $x \in [0, 1/2]$,

$$\Delta^{i+1}(x) = \Delta^i(\Delta(x)) = \Delta^i(2x) = \Delta(\left\langle 2^{i-1} 2x \right\rangle) = \Delta(\left\langle 2^i x \right\rangle).$$

- If $x \in (1/2, 1]$, now additionally using a reflection property of Δ (namely $\Delta(z) = \Delta(1 - z)$ for $z \in [0, 1]$),

$$\begin{aligned}\Delta^{i+1}(x) &= \Delta^i(\Delta(x)) = \Delta^i(2 - 2x) \\ &= \Delta^{i-1}(\Delta(2 - 2x)) = \Delta^{i-1}(\Delta(1 - (2 - 2x))) = \Delta^i(2x - 1) \\ &= \Delta(\left\langle 2^i x - 2^{i-1} \right\rangle) = \Delta(\left\langle 2^i x \right\rangle).\end{aligned}$$

(If $i = 1$, use $\Delta^{1-1}(x) = x$.)

Remark 5.3 ***(how many ReLU?)*** Generally we won't care about inputs outside $[0, 1]$, and can use two ReLUs in place of the three in the definition. But we're taking a linear combination, so the simplest way to write it is with two ReLU in one layer, then a separate ReLU layer with the linear combination. For Δ^L we can be careful and stack and compress further, but that approach is not followed here.

5.2 Separating shallow and deep networks

This section will establish the following separation between constant-width deep networks and subexponential width shallow networks.

Theorem 5.1 ***((Telgarsky 2015, 2016))*** For any $L \geq 2$. $f = \Delta^{L^2+2}$ is a ReLU network with $3L^2 + 6$ nodes and $2L^2 + 4$ layers, but any ReLU network g with $\leq 2^L$ nodes and $\leq L$ layers can not approximate it:

$$\int_{[0,1]} |f(x) - g(x)| \, \mathrm{d}x \geq \frac{1}{32}.$$

Remark 5.4 ***(why L_1 metric?)*** Previously, we used L_2 and L_∞ to state good upper bounds on approximation; for bad approximation, we want to argue there is a large region where we fail, not just a few points, and that's why we use an L_1 norm.

To be able to argue that such a large region exists, we don't just need the hard function $f = \Delta^{L^2+2}$ to have many regions, we need them to be regularly spaced, and not bunch up. In particular, if we replaced Δ with the similar function $4x(1 - x)$, then this proof would need to replace $\frac{1}{32}$ with something decreasing with L.

Proof plan for Theorem 5.1 ((Telgarsky 2015, 2016)):

1. (Shallow networks have low complexity.) First we will upper bound the number of oscillations in ReLU networks. The key part of the story is that oscillations will grow polynomially in width, but *exponentially* in depth. [mjt☹: give explicit lemma ref]

2. (There exists a *regular*, high complexity deep network.) Then we will show there exists a function, realized by a slightly deeper network, which has many oscillations, which are moreover *regularly spaced*. The need for regular spacing will be clear at the end of the proof. We have already handled this part of the proof: the hard function is Δ^{L^2+2}.

3. Lastly, we will use a region-counting argument to combine the preceding two facts to prove the theorem. This step would be easy for the L_∞ norm, and takes a bit more effort for the L_1 norm.

Remark 5.5 ***(bibliographic notes)*** Theorem 5.1 ((Telgarsky 2015, 2016)) was the earliest proof showing that a deep network can not be approximated by a reasonably-sized shallow network, however prior work showed a separation for *exact* representation of deep *sum-product networks* as compared with shallow ones (Bengio and Delalleau 2011). A sum-product network has nodes which compute affine transformations or multiplications, and thus a multi-layer sum-product network is a polynomial, and this result, while interesting, does not imply a ReLU separation.

As above, step 1 of the proof upper bounds the total possible number of affine pieces in a univariate network of some depth and width, and step 2 constructs a deep function which roughly meets this bound. Step 1 can be generalized to the multivariate case, with reasoning similar to the VC-dimension bounds in section 17. A version of step 2 appeared in prior work but for the multivariate case, specifically giving a multivariate-input network with exponentially many affine pieces, using a similar construction (Montúfar et al. 2014). A version of step 2 also appeared previous as a step in a proof that recurrent networks are Turing complete, specifically a step used to perform digit extraction (Siegelmann and Sontag 1994, Figure 3).

Proceeding with the proof, first we want to argue that shallow networks have low complexity. Our notion of complexity is simply the number of affine pieces.

Definition 5.1 For any univariate function $f : \mathbb{R} \to \mathbb{R}$, let $N_A(f)$ denote the number of affine pieces of f: the minimum cardinality (or ∞) of a partition of $\mathbb{R}$ so that f is affine when restricted to each piece.

Lemma 5.1 Let $f : \mathbb{R} \to \mathbb{R}$ be a ReLU network with L layers of widths $(m_1, \ldots, m_L)$ with $m = \sum_i m_i$.

- Let $g : \mathbb{R} \to \mathbb{R}$ denote the output of some node in layer i as a function of the input. Then the number of affine pieces $N_A(g)$ satisfies

$$N_A(g) \leq 2^i \prod_{j<i} m_j.$$

- $N_A(f) \leq \left(\frac{2m}{L}\right)^L$.

Remark 5.6 Working with the ReLU really simplifies this reasoning!

Our proof will proceed by induction, using the following combination rules for piecewise affine functions.

Lemma 5.2 Let functions $f, g, (g_1, \ldots, g_k)$, and scalars $(a_1, \ldots, a_k, b)$ be given.

1. $N_A(f + g) \leq N_A(f) + N_A(g)$.

2. $N_A(\sum_i a_i g_i + b) \le \sum_i N_A(g_i)$.
3. $N_A(f \circ g) \le N_A(f) \cdot N_A(g)$.
4. $N_A\left(x \mapsto f(\sum_i a_i g_i(x) + b)\right) \le N_A(f) \sum_i N_A(g_i)$.

Remark 5.7 This immediately hints a "power of composition": we increase the "complexity" multiplicatively rather than additively!

Remark 5.8 It is natural and important to wonder if this exponential increase is realized in practice. Preliminary work reveals that, at least near initialization, the effective number of pieces is much smaller (Hanin and Rolnick 2019).

Proof of Lemma 5.2.

1. Draw f and g, with vertical bars at the right boundaries of affine pieces. There are $\le N_A(f) + N_A(g) - 1$ distinct bars, and $f + g$ is affine between each adjacent pair of bars.
2. $N_A(a_i g_i) \le N_A(g_i)$ (equality if $a_i \neq 0$), thus induction with the preceding gives $N_A(\sum_i a_i g_i) = \sum_i N_A(g_i)$, and N_A doesn't change with addition of constants.
3. Let $P_A(g)$ denote the pieces of g, and fix some $U \in P_A(g)$; g is a fixed affine function along U. U is an interval, and consider the pieces of $f_{|g(U)}$; for each $T \in P_A(f_{|g(U)})$, f is affine, thus $f \circ g$ is affine (along $U \cap g_{|U}^{-1}(T)$), and the total number of pieces is

$$\sum_{U \in P_A(g)} N_A(f_{|g(U)}) \le \sum_{U \in P_A(g)} N_A(f) \le N_A(g) \cdot N_A(f).$$

4. Combine the preceding two.

Remark 5.9 The composition rule is hard to make tight: the image of each piece of g must hit all intervals of f! This is part of the motivation for the function Δ, which essentially meets this bound with every composition.

Proof of Lemma 5.1.
To prove the second from the first, $N_A(f) \le 2^L \prod_{j \le L} m_j$,

$$\prod_{j \le L} m_j = \exp \sum_{j \le L} \ln m_j = \exp \frac{1}{L} \sum_{j \le L} L \ln m_j \le \exp L \ln \frac{1}{L} \sum_{j \le L} m_j = \left(\frac{m}{L}\right)^L.$$

For the first, proceed by induction on layers. Base case: layer 0 mapping the data with identity, thus $N_A(g) = 1$. For the inductive step, given g in layer $i+1$ which takes $(g_1, \ldots, g_{m_i})$ from the previous layer as input,

$$N_A(g) = N_A(\sigma(b + \sum_j a_j g_j)) \le 2 \sum_{j=1}^{m_i} N_A(g_j)$$
$$\le 2 \sum_{j=1}^{m_i} 2^i \prod_{k<i} m_k = 2^{i+1} m_i \cdot \prod_{k<i} m_k.$$

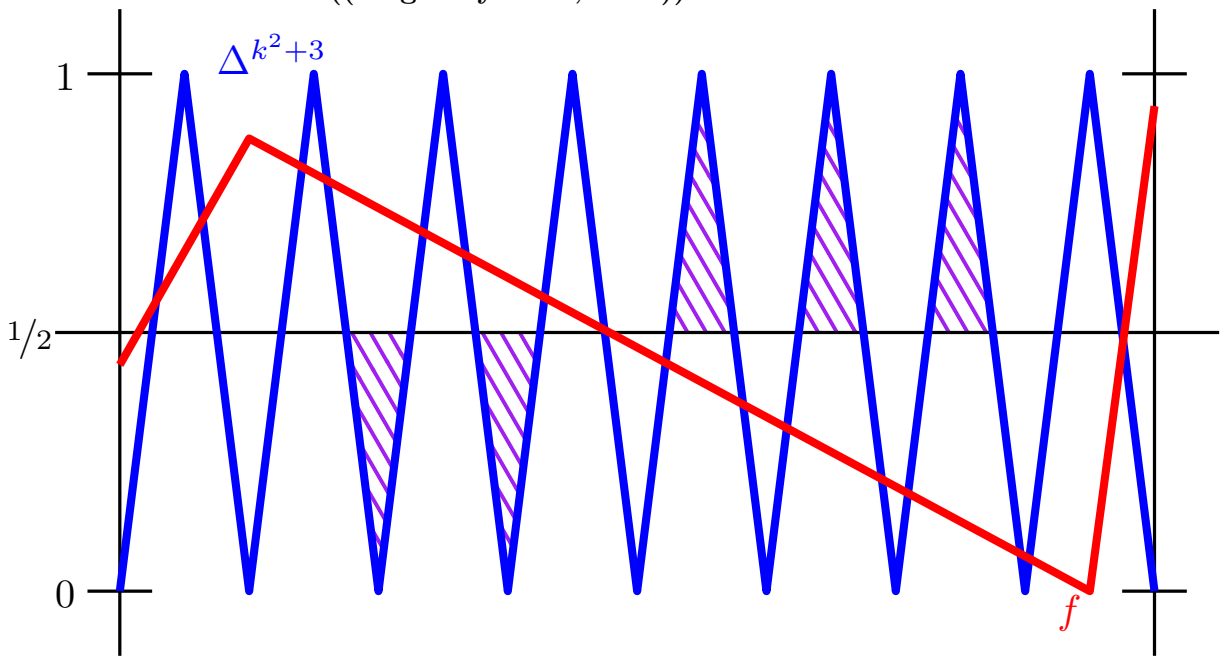
This completes part 1 of our proof plan, upper bounding the number of affine pieces polynomially in width and exponentially in depth.

The second part of the proof was to argue that Δ^L gives a high complexity, regular function: we already provided this in Proposition 5.1, which showed that Δ^L gives exactly 2^{L-1} copies of Δ, each

shrunken uniformly by a factor of 2^{L-1}.

The third part is a counting argument which ensures the preceding two imply the claimed separation in L_1 distance; details are as folllows.

Proof of Theorem 5.1 ((Telgarsky 2015, 2016)).



The proof proceeds by "counting triangles."

- Draw the line $x \mapsto 1/2$ (as in the figure). The "triangles" are formed by seeing how this line intersects $f = \Delta^{L^2+2}$. There are 2^{L^2+1} copies of Δ, which means $2^{L^2+2} - 1$ (half-)triangles since we get two (half-)triangles for each Δ but one is lost on the boundary of $[0, 1]$. Each (half-)triangle has area $\frac{1}{4} \cdot \frac{1}{2^{L^2+2}} = 2^{-L^2-4}$.
- We will keep track of when g passes above and below this line; when it is above, we will count the triangles below; when it is above, we'll count the triangles below. Summing the area of these triangles forms a lower bound on $\int_{[0,1]} |f - g|$.
- Using the earlier lemma, g has $N_A(g) \leq (2 \cdot 2^L/L)^L \leq 2^{L^2}$.
- For each piece, we shouldn't count the triangles at its right endpoint, or if it crosses the line, and we also need to divide by two since we're only counting triangles on one side; together

$$\begin{aligned}
\int_{[0,1]} |f - g| &\geq [\text{number surviving triangles}] \cdot [\text{area of triangle}] \\
&\geq \frac{1}{2}\left[2^{L^2+2} - 1 - 2 \cdot 2^{L^2}\right] \cdot \left[2^{-L^2-4}\right] \\
&= \frac{1}{2}\left[2^{L^2+1} - 1\right] \cdot \left[2^{-L^2-4}\right] \\
&\geq \frac{1}{32}.
\end{aligned}$$

Remark 5.10 ***(other depth separations)***

- Our construction was univariate. Over $\mathbb{R}^d$, there exist ReLU networks with poly(d) notes

in 2 hidden layers which can not be approximated by 1-hidden-layer networks unless they have $\geq 2^d$ nodes (Eldan and Shamir 2015).

  - The 2-hidden-layer function is approximately radial; we also mentioned that these functions are difficult in the Fourier material; the quantity $\int \|w\| \cdot |\hat{f}(w)| \mathrm{d}w$ is generally exponential in dimension for radial functions.
  - The proof by (Eldan and Shamir 2015) is very intricate; if one adds the condition that weights have subexponential size, then a clean proof is known (Daniely 2017).
  - Other variants of this problem are open; indeed, there is recent evidence that separating constant depth separations is hard, in the sense of reducing to certain complexity theoretic questions (Vardi and Shamir 2020).

- A variety of works consider connections to tensor approximation and sum product networks (Cohen and Shashua 2016; Cohen, Sharir, and Shashua 2016).
- Next we will discuss the approximation of x^2.

5.3 Approximating x^2

Why x^2?

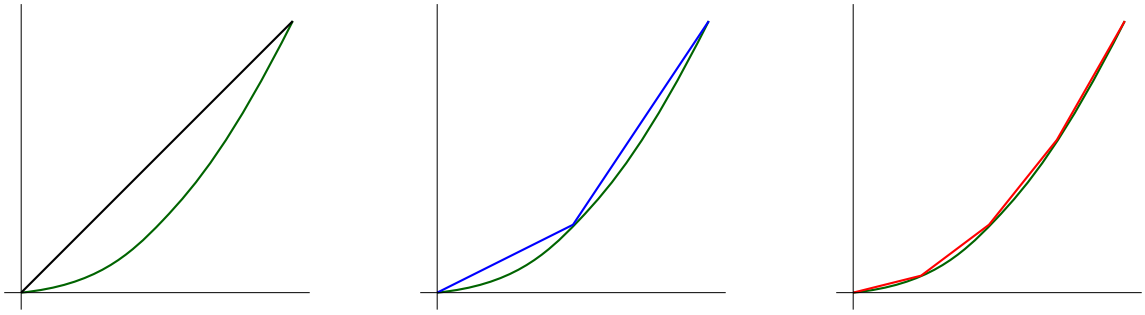
- **Why it should be easy:** because $x^2 = \int_0^\infty 2\sigma(x-b)\mathrm{d}b$, so we need only to uniformly place ReLUs.
  - We'll use an approximate construction due to (Yarotsky 2016). It will need only $\operatorname{poly}\log(1/\epsilon)$ nodes and depth to ϵ-close!
  - By contrast, our *shallow* univariate approximation theorems needed $1/\epsilon$ nodes.
- **Why we care:** with x^2, polarization gives us multiplication:

$$xy = \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right).$$

  From that, we get monomials, polynomials, Taylor expansions.

Remark 5.11 ***(bibliographic notes)*** The ability to efficiently approximate $x \mapsto x^2$, and consequences of this, was observed nearly in parallel by a few authors; in addition to (Yarotsky 2016) as mentioned above (whose approach is roughly followed here), in parallel was the work of (Safran and Shamir 2016), and slightly later the result was also discovered by (Rolnick and Tegmark 2017), all of these with differing perspectives and proofs.

Define $S_i := \left(\frac{0}{2^i}, \frac{1}{2^i}, \ldots, \frac{2^i}{2^i}\right)$; let h_i be the linear interpolation of x^2 on S_i.



Thus:

- $h_i = h_{i+1}$ on S_i.
- For $x \in S_{i+1} \setminus S_i$, defining $\epsilon = 2^{-i-1}$,

$$\begin{aligned} h_i(x) - h_{i+1}(x) &= \frac{1}{2}\left(h_i(x-\epsilon) + h_i(x+\epsilon)\right) - h_{i+1}(x) \\ &= \frac{1}{2}\left((x-\epsilon)^2 + (x+\epsilon)^2\right) - x^2 = \epsilon^2. \end{aligned}$$

  Key point: no dependence on x!

- Thus, for any $x \in S_{i+1}$,

$$h_{i+1}(x) = h_i(x) - \frac{1}{4^{i+1}}\mathbf{1}[x \in S_{i+1} \setminus S_i]$$

- Since h_{i+1} linearly interpolates, then $h_{i+1} - h_i$ must also linearly interpolate. The linear interpolation of $\mathbf{1}[x \in S_{i+1} \setminus S_i]$ is Δ^{i+1} ! Thus

$$h_{i+1} = h_i - \frac{\Delta^{i+1}}{4^{i+1}}.$$

- Since $h_0(x) = x$, then $h_i(x) = x - \sum_{j=1}^{i} \frac{\Delta^j(x)}{4^j}$.

Theorem 5.2 ***(roughly following (Yarotsky 2016))***

1. h_i is the piecewise-affine interpolation of x^2 along $[0,1]$ with interpolation points S_i.
2. h_i can be written as a ReLU network consisting of $2i$ layers and $3i$ nodes using "skip connections," or a pure ReLU network with $2i$ layers and $4i$ nodes.
3. $\sup_{x\in[0,1]} |h_i(x) - x^2| \le 4^{-i-1}$.
4. Any ReLU network f with $\le L$ layers and $\le N$ nodes satisfies

$$\int_{[0,1]} (f(x) - x^2)^2 \mathrm{d}x \ge \frac{1}{5760(2N/L)^{4L}}.$$

Remark 5.12

- Can interpret as: $\mathcal{O}(\ln(1/\epsilon))$ layers are necessary and sufficient if we want size $\mathcal{O}(\ln(1/\epsilon))$. [mjt☹: i need to do this explicitly]
- Last one can be beefed up to a lower bound against strongly convex functions.

Proof.

1. The interpolation property comes from construction/definition.
2. Since $h_i = x - \sum_{j=1}^{i} \frac{\Delta^j}{4^j}$ and since Δ^j requires 3 nodes and 2 layers for each new power, a worst case construction would need $2i$ layers and $3\sum_{j\le i} j = \mathcal{O}(i^2)$ nodes, but we can reuse individual Δ elements across the powers, and thus need only $3i$, though the network has "skip connections" (in the ResNet sense); alternatively we can replace the skip connections with a single extra node per layer which accumulates the output, or rather after layer j outputs h_j, which suffices since $h_{j+1} - h_j = \Delta^{j+1}/4^{j+1}$.
3. Fix i, and set $\tau := 2^{-i}$, meaning τ is the distance between interpolation points. The error between x^2 and h_i is thus bounded above by

$$\begin{aligned}
&\sup_{x\in[0,1-\tau]} \sup_{z\in[0,\tau]} \frac{\tau - z}{\tau}\left(x^2\right) + \frac{z}{\tau}(x+\tau)^2 - (x+z)^2 \\
&= \frac{1}{\tau} \sup_{x\in[0,1-\tau]} \sup_{z\in[0,\tau]} 2xz\tau + z\tau^2 - 2xz\tau - \tau z^2 \\
&= \frac{1}{4\tau} \sup_{x\in[0,1-\tau]} \frac{\tau^3}{4} = \frac{\tau^2}{4} = 4^{-i-1}.
\end{aligned}$$

4. By a bound from last lecture, $N_A(f) \le (2N/L)^L$. Using a symbolic package to differentiate, for any interval $[a,b]$,

$$\min_{(c,d)} \int_{[a,b]} (x^2 - (cx+d))^2 \mathrm{d}x = \frac{(b-a)^5}{180}.$$

Let S index the subintervals of length at least $1/(2N)$ with $N := N_A(f)$, and restrict attention to $[0,1]$. Then

$$\sum_{[a,b]\in S} (b-a) = 1 - \sum_{[a,b]\notin S} (b-a) \ge 1 - N/(2N) = 1/2.$$

Consequently,

$$\begin{aligned}
\int_{[0,1]} (x^2 - f(x))^2 \mathrm{d}x &= \sum_{[a,b]\in P_A(f)} \int_{[a,b]\cap[0,1]} (x^2 - f(x))^2 \mathrm{d}x \\
&\ge \sum_{[a,b]\in S} \frac{(b-a)^5}{180} \\
&\ge \sum_{[a,b]\in S} \frac{(b-a)}{2880N^4} \ge \frac{1}{5760N^4}.
\end{aligned}$$

From squaring we can get many other things (still with $\mathcal{O}(\ln(1/\epsilon))$ depth and size.

- Multiplication (via "polarization"):

$$(x,y) \mapsto xy = \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right).$$

- Multiplications gives polynomials.
- $\frac{1}{x}$ and rational functions (Telgarsky 2017).
- Functions with "nice Taylor expansions" (Sobolev spaces) (Yarotsky 2016); though now we'll need size bigger than $\ln \frac{1}{\epsilon}$:
  - First we approximate each function locally with a polynomial.
  - We multiply each local polynomial by a bump ((Yarotsky 2016) calls the family of bumps a "partition of unity").
  - This was also reproved and connected to statistics questions by (Schmidt-Hieber 2017).

Theorem 5.3 ***(sketch, from (Yarotsky 2016; Schmidt-Hieber 2017))*** Suppose $f : \mathbb{R}^d \to \mathbb{R}$ has all coordinates of all partial derivatives of order up to r within $[-1, +1]$ and let $\epsilon > 0$ be given. Then there exists a $\widetilde{\mathcal{O}}(\ln(1/\epsilon)$ layer and $\widetilde{\mathcal{O}}(\epsilon^{-d/r})$ width network so that

$$\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq \epsilon.$$

[mjt☹: gross and vague, i should clean]

Remark 5.13 There are many papers following up on these; e.g., crawl the citation graph outwards from (Yarotsky 2016).

5.4 Sobolev balls

Here we will continue and give a version of Yarotsky's main consequence to the approximation of x^2: approximating functions with many bounded derivatives (by approximating their Taylor expansions), formally an approximation result against a Sobolev ball in function space.

Remark 5.14 ***(bibliographic notes)*** This is an active area of work; in addition to the original work by (Yarotsky 2016), it's also worth highlighting the re-proof by (Schmidt-Hieber 2017), which then gives an interesting regression consequence. There are many other works in many directions, for instance adjusting the function class to lessen the (still bad) dependence on dimension (Montanelli, Yang, and Du 2020). These approaches all work with polynomials, but it's not clear this accurately reflects approximation power of ReLU networks (Telgarsky 2017).

Theorem 5.4 Suppose $g : \mathbb{R}^d \to \mathbb{R}$ satisfies $g(x) \in [0, 1]$ and all partial derivatives of all orders up to r are at most M. Then there exists a ReLU network with $\mathcal{O}(k(r + d))$ layers and $\mathcal{O}((kd + d^2 + r^2 d^r + krd^r)s^d)$ nodes such that

$$|f(x) - g(x)| \leq Mrd^r \left(s^{-r} + 4d2^d \cdot 4^{-k}\right) + 3d2^d \cdot 4^{-k}. \qquad \forall x \in [0, 1]^d.$$

[mjt☹: This isn't quite right; yarotsky claims a width $c(d, r)/\epsilon^{d/r} \ln(1/\epsilon)$ suffices for error ϵ; need to check what I missed.]

Remark 5.15 ***(not quite right)*** Matus note from Matus to Matus: Yarotsky gets width $c(d, r) \ln(1/\epsilon)/\epsilon^{d/r}$ and mine is worse, need to track down the discrepancy.

The proof consists of the following pieces:

1. Functions in Sobolev space are locally well-approximated by their Taylor expansions; therefore we will expand the approximation of x^2 to give approximation of general monomials in Lemma 5.4.

2. These Taylor approximations really only work locally. Therefore we need a nice way to switch between different Taylor expansions in different parts of $[0,1]^d$. This leads to the construction of a *partition of unity*, and is one of the other very interesting ideas in (Yarotsky 2016) (in addition to the construction of x^2; this is done below in Lemma 5.5.

First we use squaring to obtain multiplication.

Lemma 5.3 For any integers k, l, there exists a ReLU network $\text{prod}_{k,l} : \mathbb{R}^l \to \mathbb{R}$ which requires $\mathcal{O}(kl)$ layers and $\mathcal{O}(kl + l^2)$ nodes such that for any $x \in [0,1]^l$,

$$\left| \text{prod}_{k,l}(x) - \prod_{j=1}^{l} x_j \right| \leq l \cdot 4^{-k},$$

and $\text{prod}_{k,l}(x) \in [0,1]$, and $\text{prod}_{k,l}(x) = 0$ if any x_j is 0.

Proof. The proof first handles the case $l = 2$ directly, and uses $l - 1$ copies of $\text{prod}_{k,2}$ for the general case.
As such, for $(a, b) \in \mathbb{R}^2$, define

$$\text{prod}_{k,2}(a, b) := \frac{1}{2}\left(4h_k((a+b)/2) - h_k(a) - h_k(b)\right).$$

The size of this network follows from the size of h_k given in Theorem 5.2 (roughly following (Yarotsky 2016)), and $\text{prod}_{k,2}(a, b) = 0$ when either argument is 0 since $h_k(0) = 0$. For the approximation guarantee, since every argument to each h_k is within $[0,1]$, then Theorem 5.2 (roughly following (Yarotsky 2016)) holds, and using the polarization identity to rewrite $a \cdot b$ gives

$$\begin{aligned} 2|\text{prod}_{k,l}(a, b) - ab| &= 2|\text{prod}_{k,l}(a, b) - \frac{1}{2}((a+b)^2 - a^2 - b^2)| \\ &\leq 4|h_k((a+b)/2) - ((a+b)/2)^2| + |h_k(a) - a^2| + |h_k(b) - b^2| \\ &\leq 4 \cdot 4^{-k-1} + 4^{-k-1} + 4^{-k-1} \leq 2 \cdot 4^{-k}. \end{aligned}$$

Now consider the case $\text{prod}_{k,i}$ for $i > 2$: this network is defined via

$$\text{prod}_{k,i}(x_1, \ldots, x_i) := \text{prod}_{k,2}(\text{prod}_{k,i-1}(x_1, \ldots, x_{i-1}), x_i).$$

It is now shown by induction that this network has $\mathcal{O}(ki + i^2)$ nodes and $\mathcal{O}(ki)$ layers, that it evaluates to 0 when any argument is zero, and lastly satisfies the error guarantee

$$\left| \text{prod}_{k,i}(x_{1:i}) - \prod_{j=1}^{i} x_j \right| \leq i4^{-k}.$$

The base case $i = 2$ uses the explicit $\text{prod}_{k,2}$ network and gaurantees above, thus consider $i > 2$. The network embeds $\text{prod}_{k,i-1}$ and another copy of $\text{prod}_{k,2}$ as subnetworks, but additionally must pass the input x_i forward, thus requires $\mathcal{O}(ki)$ layers and $\mathcal{O}(ki + i^2)$ nodes, and evaluates to 0 if any argument is 0 by the guarantees on $\text{prod}_{k,2}$ and the inductive hypothesis. For the error

estimate,

$$\begin{aligned}\left|\mathrm{prod}_{k,i}(x_1,\ldots,x_i)-\prod_{j=1}^{i}x_j\right| &\le \left|\mathrm{prod}_{k,2}(\mathrm{prod}_{k,i-1}(x_1,\ldots,x_{i-1}),x_i)-x_i\mathrm{prod}_{k,i-1}(x_1,\ldots,x_{i-1})\right| \\ &\quad+\left|x_i\mathrm{prod}_{k,i-1}(x_1,\ldots,x_{i-1})-x_i\prod_{j=1}^{i-1}x_j\right| \\ &\le 4^{-k}+|x_i|\cdot\left|\mathrm{prod}_{k,i-1}(x_1,\ldots,x_{i-1})-\prod_{j=1}^{i-1}x_j\right| \\ &\le 4^{-k}+|x_i|\cdot\left((i-1)4^{-k}\right)\le i4^{-k}.\end{aligned}$$

From multiplication we get monomials.

Lemma 5.4 Let degree r and input dimension d be given, and let N denote the number of monomials of degree at most r. Then there exists a ReLU network $\mathrm{mono}_{k,r}:\mathbb{R}^d\to\mathbb{R}^N$ with $\mathcal{O}(kr)$ layers and $\mathcal{O}(d^r(kr+r^2))$ nodes so that for any vector of exponents $\vec{\alpha}$ corresponding to a monomial of degree at most r, meaning $\vec{\alpha}\ge 0$, $\sum_i\alpha_i\le r$, and $x^{\vec{\alpha}}:=\prod_{i=1}^d x_i^{\alpha_i}$, then the output coordinate of $\mathrm{mono}_{k,r}$ corresponding to $\vec{\alpha}$, written $\mathrm{mono}_{k,r}(x)_{\vec{\alpha}}$ for convenience, satisfies

$$\left|\mathrm{mono}_{k,r}(x)_{\vec{\alpha}}-x^{\vec{\alpha}}\right|\le r4^{-k}\qquad\forall x\in[0,1]^d.$$

Proof. $\mathrm{mono}_{k,r}$ consists of N parallel networks, one for each monomial. As such, given any $\vec{\alpha}$ of degree $q\le r$, to define coordinate $\vec{\alpha}$ of $\mathrm{mono}_{k,r}$, first rewrite α as a vector $v\in\{1,\ldots,d\}^q$, whereby

$$x^{\vec{\alpha}}:=\prod_{i=1}^{q}x_{v_i}.$$

Define

$$\mathrm{mono}_{k,r}(x)_{\vec{\alpha}}:=\mathrm{prod}_{k,q}(x_{v_1},\ldots,x_{v_q}),$$

whereby the error estimate follows from Lemma 5.3, and the size estimate follows by multiplying the size estimate from Lemma 5.3 by N, and noting $N\le d^r$.

Next we construct the approximate partition of unity.

Lemma 5.5 For any $s\ge 1$, let $\mathrm{part}_{k,s}:\mathbb{R}^d\to\mathbb{R}^{(s+1)^d}$ denote an approximate partition of unity implemented by a ReLU network, detailed as follows.

1. For any vector $v\in S:=\{0,1/s,\ldots,s/s\}^d$, there is a corresponding coordinate $\mathrm{part}_{k,s}(\cdot)_v$, and this coordinate is only supported locally around v, meaning concretely that $\mathrm{part}_{k,s}(x)_v$ is zero for $x\notin\prod_{j=1}^d[v_j-1/s,v_j+1/s]$.
2. For any $x\in[0,1]^d$, $|\sum_{v\in S}\mathrm{part}_{k,s}(x)_v-1|\le d2^d4^{-k}$.
3. $\mathrm{part}_{k,s}$ can be implemented by a ReLU network with $\mathcal{O}(kd)$ layers and $\mathcal{O}((kd+d^2)s^d)$ nodes.

Proof. Set $N:=(s+1)^d$, and let S be any enumeration of the vectors in the grid $\{0,1/s,\ldots,s/s\}^d$.

Define first a univariate bump function

$$h(a) := \sigma(sa+1) - 2\sigma(sa) + \sigma(sa-1) = \begin{cases} 1+sa & a \in [-1/s, 0), \\ 1-sa & a \in [0, 1/s] \\ 0 & \text{o.w.}. \end{cases}$$

For any $v \in S$, define

$$f_v(x) := \text{prod}_{k,d}(h(x_1 - v_1), \ldots, h(x_d - v_d)).$$

By Lemma 5.3,

$$\sup_{x \in [0,1]^d} |f_v(x) - \prod_{j=1}^{d} h(x_j - v_j)| \le d4^{-k}.$$

Each coordinate of the output of $\text{part}_{k,s}$ corresponds to some $v \in S$; in particular, define

$$\text{part}_{k,s}(x)_v := f_v(x).$$

As such, by the definition of f_v, and Lemma 5.3, and since $|S| \le (s+1)^d$, then $\text{part}_{k,s}$ can be written with kd layers and $\mathcal{O}((kd+d^2)s^d)$ nodes. The local support claim for $\text{part}_{k,s}(\cdot)_v$ follows by construction. For the claim of approximate partition of unity, using $U \subseteq S$ to denote the local set of coordinates corresponding to nonzero coordinates of $\text{part}_{k,s}$ (which has $|U| \le 2^d$ by the local support claim),

$$\begin{aligned} |\sum_{v \in S} \text{part}_{k,s}(x)_v - 1| &= |\sum_{v \in U} (\text{part}_{k,s}(x)_v - \prod_{j=1}^{d} h(x_j - v_j) + \prod_{j=1}^{d} h(x_j - v_j) - 1| \\ &\le \sum_{v \in U} |\text{part}_{k,s}(x)_v - \prod_{j=1}^{d} h(x_j - v_j)| + |\sum_{v \in U} \prod_{j=1}^{d} h(x_j - v_j) - 1| \\ &\le 2^d d4^{-k} + |\sum_{v \in U} \prod_{j=1}^{d} h(x_j - v_j) - 1|. \end{aligned}$$

It turns out the last term of the sum is 0, which completes the proof: letting u denote the lexicographically smallest element in U (i.e., the "bottom left corner"),

$$\begin{aligned} |\sum_{v \in U} \prod_{j=1}^{d} h(x_j - v_j) - 1| &= |\sum_{w \in \{0,1/s\}^d} \prod_{j=1}^{d} h((x - u + w)_j) - 1| \\ &= |\prod_{j=1}^{d} \sum_{w_j \in \{0,1/s\}} h((x - u + w)_j) - 1| \\ &= |\prod_{j=1}^{d} (h(x_j - u_j) + h(x_j - u_j + 1/s)) - 1|, \end{aligned}$$

which is 0 because $z := x - u \in [0, 1/s]^d$ by construction, and using the case analysis of h gives

$$h(z_j) + h(z_j + 1/s) = (1 + sz_j) + (1 - s(z_j + 1/s)) = 1$$

as desired.

Finally we are in shape to prove Theorem 5.4.

Proof of Theorem 5.4. The ReLU network for f will combine $\mathrm{part}_{k,s}$ from Lemma 5.5 with $\mathrm{mono}_{k,r}$ from Lemma 5.4 via approximate multiplication, meaning $\mathrm{prod}_{k,2}$ from Lemma 5.3.
In detail, let the grid $S := \{0, 1/s, \ldots, s/s\}^d$ be given as in the statement of Lemma 5.5. For each $v \in S$, let $p_v : \mathbb{R}^d \to \mathbb{R}$ denote the Taylor expansion of degree r at v; by a standard form of the Taylor error, for any $x \in [0,1]^d$ with $\|x - v\|_\infty \leq 1/s$,

$$|p_v(x) - g(x)| \leq \frac{Md^r}{r!}\|v - x\|_\infty^r \leq \frac{Md^r}{r!s^r}.$$

Next, let w_v denote the Taylor coefficients forming p_v, and define $f_v : \mathbb{R}^d \to \mathbb{R}$ as $x \mapsto w_v^{\mathrm{T}}\mathrm{mono}_{k,r}(x - v)$, meaning approximate p_v by taking the linear combination with weights w_v of the approximate monomials in $x \mapsto \mathrm{mono}_{k,r}(x - v)$. By Lemma 5.4, since there are at most d^r terms, the error is at most

$$|f_v(x) - p_v(x)| = |\sum_{\vec{\alpha}}(w_v)_{\vec{\alpha}}(\mathrm{mono}_{k,r}(x - v)_{\vec{\alpha}} - (x - v)^{\vec{\alpha}})| \leq \sum_{\vec{\alpha}}|(w_v)_{\vec{\alpha}}|r4^{-k} \leq Mrd^r 4^{-k}.$$

[mjt☹: just realized a small issue that negative inputs might occur; can do some shifts or reflections or whatever to fix.]
The final network is now obtained by using $\mathrm{prod}_{k,2}$ to approximately multiply each approximate Taylor expansion f_v by the corresponding locally-supported approximate partition of unity element $\mathrm{part}_{k,s}(x)_v$; in particular, define

$$f(x) := \sum_{v \in S}\mathrm{prod}_{k,2}(f_v(x), \mathrm{part}_{k,s}(x)_v).$$

Then, using the above properties and the fact that the partition of unity is locally supported, letting $U \subseteq S$ denote the set of at most 2^d active elements,

$$\begin{aligned}
|f(x) - g(x)| &\leq \left|\sum_{v\in S}\mathrm{prod}_{k,2}(f_v(x), \mathrm{part}_{k,s}(x)_v) - \sum_{v\in S} f_v(x)\mathrm{part}_{k,s}(x)_v\right| \\
&+ \left|\sum_{v\in S} f_v(x)\mathrm{part}_{k,s}(x)_v - \sum_{v\in S} p_v(x)\mathrm{part}_{k,s}(x)_v\right| \\
&+ \left|\sum_{v\in S} p_v(x)\mathrm{part}_{k,s}(x)_v - \sum_{v\in S} g(x)\mathrm{part}_{k,s}(x)_v\right| \\
&+ \left|\sum_{v\in S} g(x)\mathrm{part}_{k,s}(x)_v - g(x)\right| \\
&\leq 2|U|4^{-k} + Mrd^r 4^{-k}(1 + d2^d 4^{-k}) + \frac{Md^r}{r!s^r}(1 + d2^d 4^{-k}) + |f(x)|d2^d 4^{-k} \\
&\leq Mrd^r\left(s^{-r} + 4d2^d \cdot 4^{-k}\right) + 3d2^d \cdot 4^{-k}.
\end{aligned}$$

[mjt☹: The input to $\mathrm{prod}_{k,2}$ can exceed 1. for a maximally lazy fix, I should just clip its input.]

6 Optimization: preface

Classically, the purpose of optimization is to approximately minimize (or maximize) an *objective function* f over a domain S:

$$\min_{w \in S} f(w).$$

A core tension in the use of optimization in machine learning is that we would like to minimize the *population* risk $\mathcal{R}(w) := \mathbb{E}\,\ell(Yf(X;w))$; however, we only have access to the *empirical* risk $\widehat{\mathcal{R}}(w) := n^{-1}\sum_i \ell(y_i f(x_i;w))$.

As a result, when choosing a w_t, we not only care that $\widehat{\mathcal{R}}(w_t)$ is small, but also other good properties which may indicate $\mathcal{R}(w_t)$ is small as well. Foremost amongst these are that w_t has low norm, but there are other possibilities.

Outline.

- We will cover primarily first-order methods, namely gradient descent

  $$w_{t+1} := w_t - \eta_t \nabla \widehat{\mathcal{R}}(w_t),$$

  as well as the gradient flow

  $$\frac{\mathrm{d}w}{\mathrm{d}t} = \dot{w}(t) = -\nabla \widehat{\mathcal{R}}(w(t)).$$

  These dominate machine learning since:

  - They have low per-iteration complexity (which can be reduced further with stochastic gradients); classical optimization developed many methods with higher per-iteration cost but a lower number of iterations, but the high accuracy these give is not important here since our true objective is unknown anyway.
  - It seems they might have additional favorable properties; e.g., we will highlight the preference for low-norm solutions of first-order methods.

- First we'll cover classical smooth and convex opt, including strong convexity and stochastic gradients.

  Here our analysis differs from the literature by generally not requiring boundedness or existence of minima. Concretely, many proofs will use an arbitrary reference point z in place of an optimum $\bar{w}$ (which may not exist); this arbitrary z will be used effectively in the margin maximization lectures.

- Then we will cover topics closer to deep learning, including gradient flow in a smooth shallow NTK case, and a few margin maximization cases, with a discussion of nonsmoothness.

Remark 6.1

- Even though our models are not convex (and $\widehat{\mathcal{R}}$ is not convex in the parameters), our losses will always be convex.
- Analyzing gradient flow simplifies analyses, but in some cases it is difficult or completely unclear how to reproduce the same rates with gradient descent, and secondly it isn't clear that they *should* have the same rates or convergence properties; in deep learning, for instance, the role of step size is not well-understood, whereas approximating gradient flow suggests small step sizes.

- A *regularized ERM* objective has the form $w \mapsto \widehat{\mathcal{R}}(w) + P(w)$, where (for example) $P(w) := \lambda \|w\|^2/2$. We will not discuss these extensively, and we will similarly hardly discuss constrained optimization.
- A good introductory text on various optimization methods in machine learning is (Bubeck 2014); for more on convex optimization, see for instance (Nesterov 2003), and for more on convex analysis, see for instance (Bubeck 2014; Borwein and Lewis 2000).

[mjt☹: . . . maybe I should always use $\widehat{\mathcal{R}}$ or F for objectives]

6.1 Omitted topics

- **Mean-field perspective** (Chizat and Bach 2018; Mei, Montanari, and Nguyen 2018): as $m \to \infty$, gradient descent mimics a *Wasserstein flow* on a distribution over nodes (random features). Many mean-field papers are in the 2-homogeneous case, whereas many NTK papers are in the 1-homogeneous case, which further complicates comparisons.
- **Landscape analysis.** (E.g., all local optima are global.)
  - Matrix completion: solve (under RIP)
    $$\min_{X \in d \times r} \sum_{(i,j) \in S} (M_{i,j} - XX^\intercal)^2.$$
    Recently it was shown that all local optima are global, and so gradient descent from random initialization suffices (Ge, Lee, and Ma 2016).
  - For linear networks optimized with the squared loss, local optima are global, but there are bad saddle points (Kawaguchi 2016).
  - Width n suffices with general losses and networks (Nguyen and Hein 2017).
  - *[There is also work on residual networks but I haven't looked closely.]*
- **Acceleration.** Consider gradient descent *with momentum*: w_0 arbitrary, and thereafter
  $$v_{i+1} := w_i - \eta_i \nabla \widehat{\mathcal{R}}(w_i), \qquad w_{i+1} := v_{i+1} + \gamma_i (v_{i+1} - v_i)$$
  This sometimes seems to help in deep learning (even in stochastic case), but no one knows why (and opinions differ).

  If set $\eta_i = 1/\beta$ and $\gamma_i = i/(i+3)$ **(constants matter)** and $\mathcal{R}$ convex, $\widehat{\mathcal{R}}(w_i) - \inf_w \widehat{\mathcal{R}}(w) \leq \mathcal{O}(1/t^2)$ ("Nesterov's accelerated method"). This rate is tight amongst algorithms outputting iterates in the span of gradients, under some assumptions people treat as standard.
- **Escaping saddle points.** By adding noise to the gradient step, it is possible to exit saddle points (Jin et al. 2017). Some papers use this technique, though it is most useful in settings where all local minima (stationary points that are not saddles) are global minima.
- **Beyond NTK.** A very limited amount of work studies nonlinear cases beyond what is possible with the NTK and/or highlighting ways in which the NTK does not capture the behavior of deep networks in practice, in particular showing sample complexity separations (Allen-Zhu and Li 2019; Daniely and Malach 2020; Ghorbani et al. 2020; Kamath, Montasser, and Srebro 2020; Yehudai and Shamir 2019, 2020).

- **Benefits of depth for optimization.** Most of these works are either for shallow networks, or the analysis allows depth but *degrades* with increasing depth, in contrast with practical observations. A few works now are trying to show how depth can help optimization; one perspective is that sometimes it can accelerate convergence (Arora, Cohen, and Hazan 2018; Arora, Cohen, et al. 2018a).
- **Other first-order optimizers,** e.g., Adam. There is recent work on these but afaik it doesn't capture why these work well on many deep learning tasks.
- **Further analysis of overparameterization.** Overparameterization makes many aspects of the optimization problem nicer, in particular in ways not investigated in these notes (Shamir 2018; S. Du and Hu 2019).
- **Hardness of learning and explicit global solvers.** Even in simple cases, network training is NP-hard, but admits various types of approximation schemes (Goel et al. 2020; Diakonikolas et al. 2020).

7 Semi-classical convex optimization

First we will revisit classical convex optimization ideas. Our presentation differs from the normal one in one key way: we state nearly results without any assumption of a minimizer, but instead use an arbitrary *reference point* $z \in \mathbb{R}^p$. We will invoke these bounds later in settings where the minimum may not exist, but the problem structure suggests good choices for z (see e.g., Lemma 10.1).

[mjt☹: if i include ReLU ntk I can also use it there.]

7.1 Smooth objectives in ML

We say "$\widehat{\mathcal{R}}$ is β-smooth" to mean β-Lipschitz gradients:

$$\|\nabla\widehat{\mathcal{R}}(w) - \nabla\widehat{\mathcal{R}}(v)\| \le \beta\|w - v\|.$$

(The math community says "smooth" for C^∞.) We primarily invoke smoothness via the key inequality

$$\widehat{\mathcal{R}}(v) \le \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), v - w \right\rangle + \frac{\beta}{2}\|v - w\|^2.$$

In words: f can be upper bounded with the *convex* quadratic

$$v \mapsto \frac{\beta}{2}\|v - w\|^2 + \left\langle \nabla\widehat{\mathcal{R}}(w), v - w \right\rangle + \widehat{\mathcal{R}}(w),$$

which shares tangent and function value with $\widehat{\mathcal{R}}$ at w. (The first definition also implies that we are lower bounded by *concave* quadratics.)

Remark 7.1 Smoothness is *trivially* false for standard deep networks: the ReLU is not even differentiable. However, many interesting properties carry over, and many lines of research proceed by trying to make these properties carry over, so at the very least, it's good to understand.

A key consequence: we can guarantee gradient descent does not increase the objective. Consider gradient iteration $w' = w - \frac{1}{\beta}\nabla\widehat{\mathcal{R}}(w)$, then smoothness implies

$$\widehat{\mathcal{R}}(w') \leq \widehat{\mathcal{R}}(w) - \left\langle \widehat{\mathcal{R}}(w), \widehat{\mathcal{R}}(w)/\beta \right\rangle + \frac{1}{2\beta}\|\widehat{\mathcal{R}}(w)\|^2 = \widehat{\mathcal{R}}(w) - \frac{1}{2\beta}\|\nabla\widehat{\mathcal{R}}(w)\|^2,$$

and $\|\nabla\widehat{\mathcal{R}}(w)\|^2 \leq 2\beta(\widehat{\mathcal{R}}(w) - \widehat{\mathcal{R}}(w'))$. With deep networks, we'll produce similar bounds but in other ways.

As an exercise, let's prove the earlier smoothness consequence. Considering the curve $t \mapsto \widehat{\mathcal{R}}(w + t(v-w))$ along $[0,1]$,

$$\begin{aligned}
&\left|\widehat{\mathcal{R}}(v) - \widehat{\mathcal{R}}(w) - \left\langle \nabla\widehat{\mathcal{R}}(w), v - w \right\rangle\right| \\
&= \left| \int_0^1 \left\langle \nabla\widehat{\mathcal{R}}(w + t(v-w)), v - w \right\rangle \mathrm{d}t - \left\langle \nabla\widehat{\mathcal{R}}(w), v - w \right\rangle \right| \\
&\leq \int_0^1 \left| \left\langle \nabla\widehat{\mathcal{R}}(w + t(v-w)) - \nabla\widehat{\mathcal{R}}(w), v - w \right\rangle \right| \mathrm{d}t \\
&\leq \int_0^1 \|\nabla\widehat{\mathcal{R}}(w + t(v-w)) - \nabla\widehat{\mathcal{R}}(w)\| \cdot \|v - w\| \mathrm{d}t \\
&\leq \int_0^1 t\beta\|v - w\|^2 \mathrm{d}t \\
&= \frac{\beta}{2}\|v - w\|^2.
\end{aligned}$$

Example 7.1 Define $\widehat{\mathcal{R}}(w) := \frac{1}{2}\|Xw - y\|^2$, and note $\nabla\widehat{\mathcal{R}}(w) = X^\intercal(Xw - y)$. For any w, w',

$$\begin{aligned}
\widehat{\mathcal{R}}(w') &= \frac{1}{2}\|Xw' - Xw + Xw - y\|^2 \\
&= \frac{1}{2}\|Xw' - Xw\|^2 + \langle Xw' - Xw, Xw - y\rangle + \frac{1}{2}\|Xw - y\|^2 \\
&= \frac{1}{2}\|Xw' - Xw\|^2 + \left\langle w' - w, \widehat{\mathcal{R}}(w) \right\rangle + \widehat{\mathcal{R}}(w).
\end{aligned}$$

Since $\frac{\sigma_{\min}(X)}{2}\|w' - w\|^2 \leq \frac{1}{2}\|Xw' - Xw\|^2 \leq \frac{\sigma_{\max}(X)}{2}\|w' - w\|^2$, thus $\widehat{\mathcal{R}}$ is $\sigma_{\max}(X)$-smooth (and $\sigma_{\min}$-strongly-convex, as we'll discuss).

The smoothness bound holds **with equality** if we use the seminorm $\|v\|_X = \|Xv\|$. We'll (maybe?) discuss smoothness wrt other norms in homework.

[mjt☹: I should use $\mathcal{L}$ not $\widehat{\mathcal{R}}$ since unnormalized.]

7.1.1 Convergence to stationary points

Consider first the gradient iteration

$$w' := w - \eta\nabla\widehat{\mathcal{R}}(w),$$

where $\eta \geq 0$ is the step size. When f is β smooth but not necessarily convex, the smoothness inequality directly gives

$$\begin{aligned}\widehat{\mathcal{R}}(w') &\leq \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle + \frac{\beta}{2}\|w' - w\|^2 \\ &= \widehat{\mathcal{R}}(w) - \eta\|\nabla\widehat{\mathcal{R}}(w)\|^2 + \frac{\beta\eta^2}{2}\|\nabla\widehat{\mathcal{R}}(w)\|^2 \\ &= \widehat{\mathcal{R}}(w) - \eta\left(1 - \frac{\beta\eta}{2}\right)\|\nabla\widehat{\mathcal{R}}(w)\|^2.\end{aligned} \tag{3}$$

If we choose η appropriately ($\eta \leq 2/\beta$) then: either we are near a critical point ($\nabla\widehat{\mathcal{R}}(w) \approx 0$), or we can decrease $\widehat{\mathcal{R}}$.

Let's refine our notation to tell iterates apart:

1. Let w_0 be given.
2. Recurse: $w_{i+1} := w_i - \eta_i \nabla\widehat{\mathcal{R}}(w_i)$.

[mjt☹: I changed indexing (2021-09-23), need to update everywhere. . .]

Rearranging our iteration inequality eq. 3 and summing over $i < t$,

$$\begin{aligned}\sum_{i<t} \eta_i \left(1 - \frac{\beta\eta_i}{2}\right)\|\nabla\widehat{\mathcal{R}}(w_i)\|^2 &\leq \sum_{i<t}\left(\widehat{\mathcal{R}}(w_i) - \widehat{\mathcal{R}}(w_{i+1})\right) \\ &= \widehat{\mathcal{R}}(w_0) - \widehat{\mathcal{R}}(w_t).\end{aligned}$$

We can summarize these observations in the following theorem.

Theorem 7.1 Let $(w_i)_{i\geq 0}$ be given by gradient descent on β-smooth $\widehat{\mathcal{R}}$.

- If $\eta_{i+1} \in [0, 2/\beta]$, then $\widehat{\mathcal{R}}(w_{i+1}) \leq \widehat{\mathcal{R}}(w_i)$.
- If $\eta_i := \eta \in [0, 2/\beta]$ is constant across i,

$$\begin{aligned}\min_{i<t}\|\nabla\widehat{\mathcal{R}}(w_i)\|^2 &\leq \frac{1}{t}\sum_{i<t}\|\nabla\widehat{\mathcal{R}}(w_i)\|^2 \\ &\leq \frac{2}{t\eta(2-\eta\beta)}\left(\widehat{\mathcal{R}}(w_0) - \widehat{\mathcal{R}}(w_t)\right) \\ &\leq \frac{2}{t\eta(2-\eta\beta)}\left(\widehat{\mathcal{R}}(w_0) - \inf_w \widehat{\mathcal{R}}(w)\right).\end{aligned}$$

  This final expression is minimized by $\eta := \frac{1}{\beta}$, which gives

$$\min_{i<t}\|\nabla\widehat{\mathcal{R}}(w_i)\|^2 \leq \frac{1}{t}\sum_{i<t}\|\nabla\widehat{\mathcal{R}}(w_i)\|^2 \leq \frac{2\beta}{t}\left(\widehat{\mathcal{R}}(w_0) - \widehat{\mathcal{R}}(w_t)\right) \leq \frac{2\beta}{t}\left(\widehat{\mathcal{R}}(w_0) - \inf_w \widehat{\mathcal{R}}(w)\right).$$

Remark 7.2

- We have no guarantee about the last iterate $\|\nabla\widehat{\mathcal{R}}(w_t)\|$: we may get near a flat region at some $i < t$, but thereafter bounce out. With a more involved proof, we can guarantee we bounce out (J. D. Lee et al. 2016), but there are cases where the time is exponential in dimension.

- This derivation is at the core of many papers with a "local optimization" (stationary point or local optimum) guarantee for gradient descent.
- In a bit more detail, the step size $1/\beta$ is the result of minimizing the quadratic provided by smoothness:

$$\begin{aligned} w - \frac{1}{\beta}\nabla\widehat{\mathcal{R}}(w) &= \arg\min_{w'} \left(\widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle + \frac{\beta}{2}\|w' - w\|^2 \right) \\ &= \arg\min_{w'} \left(\left\langle \nabla\widehat{\mathcal{R}}(w), w' \right\rangle + \frac{\beta}{2}\|w' - w\|^2 \right). \end{aligned}$$

  This relates to *proximal descent* and *mirror descent* generalizations of gradient descent.
- In t iterations, we found a point w with $\|\nabla\widehat{\mathcal{R}}(w)\| \leq \sqrt{2\beta/t}$. We can do better with Nesterov-Polyak cubic regularization: by choosing the next iterate according to

$$\begin{aligned} \arg\min_{w'} \Big(&\widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle \\ &+ \frac{1}{2}\left\langle \nabla^2\widehat{\mathcal{R}}(w)(w' - w), w' - w \right\rangle + \frac{L}{6}\|w' - w\|^3 \Big) \end{aligned}$$

  where $\|\nabla^2\widehat{\mathcal{R}}(x) - \nabla^2\widehat{\mathcal{R}}(y)\| \leq L\|x - y\|$, then after t iterations, some iterate w_j with $j \leq t$ satisfies

$$\|\nabla\widehat{\mathcal{R}}(w_j)\| \leq \frac{\mathcal{O}(1)}{t^{2/3}}, \qquad \lambda_{\min}\left(\nabla^2\widehat{\mathcal{R}}(w_j)\right) \geq -\frac{\mathcal{O}(1)}{t^{1/3}}.$$

  Note: it is not obvious that the above cubic can be solved efficiently, but indeed there are various ways. If we go up a few higher derivatives, it becomes NP-hard. Original used an eigenvalue solver for this cubic polynomial (Nesterov and Polyak 2006). Other approaches are given by (Carmon and Duchi 2018; Jin et al. 2017), amongst many others.

Gradient flow version. Using FTC, chain rule, and definition,

$$\begin{aligned} \widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(w(0)) &= \int_0^t \left\langle \nabla\widehat{\mathcal{R}}(w(s)), \dot{w}(s) \right\rangle \mathrm{d}s \\ &= -\int_0^t \|\nabla\widehat{\mathcal{R}}(w(s))\|^2 \mathrm{d}s \\ &\leq -t \inf_{s\in[0,t]} \|\nabla\widehat{\mathcal{R}}(w(s))\|^2, \end{aligned}$$

which can be summarized as follows.

Theorem 7.2 For the gradient flow,

$$\inf_{s\in[0,t]} \|\nabla\widehat{\mathcal{R}}(w(s))\|^2 \leq \frac{1}{t}\left(\widehat{\mathcal{R}}(w(0)) - \widehat{\mathcal{R}}(w(t))\right).$$

Remark 7.3 GD: $\min_{i<t} \|\nabla\widehat{\mathcal{R}}(w_i)\|^2 \leq \frac{2\beta}{t}\left(\widehat{\mathcal{R}}(w_0) - \widehat{\mathcal{R}}(w_t)\right)$.

- β is from step size.
- "2" is from the order smoothness term (avoided in GF).

7.1.2 Convergence rate for smooth & convex

If $\widehat{\mathcal{R}}$ is differentiable and *convex*, then it is bounded below by its first-order approximations:

$$\widehat{\mathcal{R}}(w') \geq \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle \qquad \forall w, w'.$$

Theorem 7.3 Suppose $\widehat{\mathcal{R}}$ is β-smooth and convex, and $(w_i)_{\geq 0}$ given by GD with $\eta_i := 1/\beta$. Then for any z,

$$\widehat{\mathcal{R}}(w_t) - \widehat{\mathcal{R}}(z) \leq \frac{\beta}{2t}\left(\|w_0 - z\|^2 - \|w_t - z\|^2\right).$$

Remark 7.4 The reference point z allows us to use this bound effectively when $\widehat{\mathcal{R}}$ lacks an optimum, or simply when the optimum is very large. For an example of such an application of z, see the margin maximization material (e.g., Lemma 10.1).

Proof. By convexity and the earlier smoothness inequality $\|\nabla\widehat{\mathcal{R}}(w)\|^2 \leq 2\beta(\widehat{\mathcal{R}}(w) - \widehat{\mathcal{R}}(w'))$,

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

Rearranging and applying $\sum_{i<t}$,

$$\frac{2}{\beta}\sum_{i<t}(\widehat{\mathcal{R}}(w_{i+1}) - \widehat{\mathcal{R}}(z)) \leq \sum_{i<t}\left(\|w_i - z\|^2 - \|w_{i+1} - z\|^2\right)$$

The final bound follows by noting $\widehat{\mathcal{R}}(w_i) \geq \widehat{\mathcal{R}}(w_t)$, and since the right hand side telescopes.

Remark 7.5 ***(characterizing convexity)*** There are many ways to characterize convexity. As follows are a few different versions; standard texts with more characterizations and more generality (e.g., using infinite output values to model constraint sets, and using subdifferentials as a meaningful surrogate for gradients for nondifferentiable convex functions) are Hiriart-Urruty and Lemaréchal (2001).

- **(Epigraph view.)** Let $\operatorname{epi}(f)$, the *epigraph* of $f : \mathbb{R}^d \to \mathbb{R}$, denote the subset of $\mathbb{R}^{n+1}$ that is equal to or above f:

  $$\operatorname{epi}(f) := \left\{(x, y) \in \mathbb{R}^{d+1} : y \geq f(x)\right\}.$$

  f is *convex* when $\operatorname{epi}(f)$ is a convex set (meaning $[x, x'] := \{\alpha x + (1-\alpha)x' : \alpha \in [0,1]\} \subseteq \operatorname{epi}(f)$ whenever $\{x, x'\} \subseteq \operatorname{epi}(f)$), f is *strictly convex* when $\operatorname{epi}(f)$ is convex and tangents to $\operatorname{epi}(f)$ intersect only one point, and f is *strongly convex* when at any point (x, y) on the boundary of $\operatorname{epi}(f)$ meaning $f(x) = y$)), we can find a quadratic $Q : \mathbb{R}^d \to \mathbb{R}$ with $Q(x) = y$ and $\operatorname{epi}(f) \subseteq \operatorname{epi}(Q)$.

- **(Function value ("zeroth-order") view.)** Given $\alpha \in [0,1]$ and any x, x', f is convex when
$$f(\alpha x + (1-\alpha)x') \leq \alpha f(x) + (1-\alpha) f(x'),$$
strictly convex when for any $\alpha \in (0,1)$ and $x \neq x'$
$$f(\alpha x + (1-\alpha)x') < \alpha f(x) + (1-\alpha) f(x'),$$
and λ-strongly-convex when
$$f(\alpha x + (1-\alpha)x') \leq \alpha f(x) + (1-\alpha) f(x') - \frac{\lambda \alpha (1-\alpha)}{2} \|x - x'\|^2.$$
We alsop have f is λ-strongly convex iff $f - \frac{\lambda}{2}\|\cdot\|^2$ is convex.
- **(Gradient ("first-order") view.)** When f is differentiable, it is convex when
$$f(x') \geq f(x) + \langle \nabla f(x), x' - x \rangle \qquad \forall x, x',$$
strictly convex when
$$f(x') > f(x) + \langle \nabla f(x), x' - x \rangle \qquad \forall x \neq x',$$
and λ-strongly-convex when
$$f(x') \geq f(x) + \langle \nabla f(x), x' - x \rangle + \frac{\lambda}{2}\|x - x'\|^2 \qquad \forall x \neq x'.$$
We can add instantiate these inequalities for any pair (x, x') and the reverse (x', x) and combine them and get that convexity implies
$$0 \leq \langle \nabla f(x') - \nabla f(x), x' - x \rangle \qquad \forall x, x',$$
strict convexity implies
$$0 < \langle \nabla f(x') - \nabla f(x), x' - x \rangle \qquad \forall x \neq x',$$
and strong convexity implies
$$\lambda \|x - x'\|^2 \leq \langle \nabla f(x') - \nabla f(x), x' - x \rangle \qquad \forall x, x'.$$
There are also versions of all of these for nondifferentiable convex functions using *subdifferentials*, see (Hiriart-Urruty and Lemaréchal 2001).
- **(Hessian ("first-order") view.)** When f is twice-differentiable, convexity implies
$$\nabla^2 f(x) \succeq 0 \qquad \forall x,$$
strict convexity implies
$$\nabla^2 f(x) \succ 0 \qquad \forall x,$$
and λ-strong-convexity implies
$$\nabla^2 f(x) \succeq \lambda I \qquad \forall x.$$

For GF, we use the same potential, but indeed start from the telescoping sum, which can be viewed as a Riemann sum corresponding to the following application of FTC:

$$\begin{aligned}\frac{1}{2}\|w(t)-z\|_2^2-\frac{1}{2}\|w(0)-z\|_2^2 &= \frac{1}{2}\int_0^t \frac{\mathrm{d}}{\mathrm{d}s}\|w(s)-z\|_2^2 \mathrm{d}s \\ &= \int_0^t \left\langle \frac{\mathrm{d}w}{\mathrm{d}s}, w(s)-z \right\rangle \mathrm{d}s \\ &= \int_0^t \left\langle \nabla\widehat{\mathcal{R}}(w(s)), z-w(s) \right\rangle \mathrm{d}s \\ &\le \int_0^t \left(\widehat{\mathcal{R}}(z)-\widehat{\mathcal{R}}(w(s))\right) \mathrm{d}s.\end{aligned}$$

Theorem 7.4 For any $z \in \mathbb{R}^d$, GF satisfies

$$\begin{aligned}t\widehat{\mathcal{R}}(w(t))+\frac{1}{2}\|w(t)-z\|_2^2 &\le \int_0^t \widehat{\mathcal{R}}(z)\mathrm{d}s+\frac{1}{2}\|w(0)-z\|_2^2 \\ &= t\widehat{\mathcal{R}}(z)+\frac{1}{2}\|w(0)-z\|_2^2.\end{aligned}$$

Remark 7.6 ***("units" of GD and GF: t vs $\frac{t}{\beta}$)*** Here's a back-of-the-envelope calculation to see why t becomes t/β and why they are really the same, and *not* a sloppiness of the analysis.

- Suppose $\|\nabla\widehat{\mathcal{R}}(w)\| \approx 1$ for sake of illustration.
- The "distance traveled" by GD is

$$\|w_t - w_0\| = \|\frac{1}{\beta}\sum_i \nabla\widehat{\mathcal{R}}(w_i)\| \le \sum_i \frac{1}{\beta}\|\nabla\widehat{\mathcal{R}}(w_i)\| \approx \frac{t}{\beta}.$$

- The "distance traveled" by GF is (via Jensen)

$$\begin{aligned}\|w(t)-w(0)\| = \|\int_0^t \nabla\widehat{\mathcal{R}}(w(s))\mathrm{d}s\| &= \|\frac{1}{t}\int_0^t t\nabla\widehat{\mathcal{R}}(w(s))\mathrm{d}s\| \\ &\le \frac{1}{t}\int_0^t \|t\nabla\widehat{\mathcal{R}}(w(s))\|\mathrm{d}s \approx t.\end{aligned}$$

Remark 7.7 ***(potential functions)***

- For critical points, the potential was $\widehat{\mathcal{R}}(w(s))$ (or arguably $\|\nabla\widehat{\mathcal{R}}(w(s))\|_2^2$).
- Here, the potential was $\|w(s)-z\|_2^2$. This particular choice is widespread in optimization. It is interesting since it is not part of the objective function; it's some gradient descent magic?

We can use similar objective functions with deep learning, without smoothness (!).

Remark 7.8 ***(rates)*** Some rules of thumb (not comprehensive, and there are other ways).

- $\frac{1}{t}$ is often a smoothness argument as above.
- $\frac{1}{\sqrt{t}}$ uses Lipschitz (thus $\|\nabla\widehat{\mathcal{R}}\| = \mathcal{O}(1)$) in place of smoothness upper bound on $\|\nabla\widehat{\mathcal{R}}\|$.
- $\frac{1}{t^2}$ uses "acceleration," which is a fancy momentum inside the gradient.

- $\exp(-\mathcal{O}(t))$ uses strong convexity (or other fine structure on $\widehat{\mathcal{R}}$).
- Stochasticity changes some rates and what is possible, but there are multiple settings and inconsistent terminology.

7.2 Strong convexity

Recall one of our definitions of strong convexity: say that $\widehat{\mathcal{R}}$ is **λ-strongly-convex (λ-sc)** when

$$\widehat{\mathcal{R}}(w') \geq \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle + \frac{\lambda}{2}\|w' - w\|^2;$$

see Remark 7.5 (characterizing convexity) for more forms.

Example 7.2 ***(least squares)*** Earlier we pointed out

$$\frac{1}{2}\|Xw' - w'\|^2 =: \widehat{\mathcal{R}}(w') = \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle + \frac{1}{2}\|Xw' - Xw\|^2$$

and

$$\sigma_{\min}(X)\|w' - w\|^2 \leq \|Xw' - Xw\|^2 \leq \sigma_{\max}(X)\|w' - w\|^2.$$

The latter implies a smoothness upper bound we used, now we know the former implies strong convexity. (We can also say that both hold *with equality* using the special seminorm $\|v\|_X = \|Xv\|$.) We can also verify these properties by noting $\nabla^2\widehat{\mathcal{R}} = X^\intercal X$.

Example 7.3 ***(regularization)*** Define *regularized risk* $\widehat{\mathcal{R}}_\lambda(w) := \widehat{\mathcal{R}}(w) + \lambda\|w\|^2/2$.

If $\widehat{\mathcal{R}}$ is convex, then $\widehat{\mathcal{R}}_\lambda$ is λ-sc:

- A quick check is that if f is twice-differentiable, then $\nabla^2\widehat{\mathcal{R}}_\lambda = \nabla^2\widehat{\mathcal{R}} + \lambda I \succeq 0 + \lambda I$.
- Alternatively, it also follows by summing the inequalities

$$\widehat{\mathcal{R}}(w') \geq \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), w' - w \right\rangle,$$
$$\lambda\|w'\|^2/2 = \lambda\|w\|^2/2 + \langle \lambda w, w' - w\rangle + \lambda\|w' - w\|^2/2.$$

Another very useful property is that λ-sc gives a way to convert gradient norms to suboptimality.

Lemma 7.1 Suppose $\widehat{\mathcal{R}}$ is λ-sc. Then

$$\forall w\,. \qquad \widehat{\mathcal{R}}(w) - \inf_v \widehat{\mathcal{R}}(v) \leq \frac{1}{2\lambda}\|\nabla\widehat{\mathcal{R}}(w)\|^2.$$

Remark 7.9 Smoothness gave $\frac{1}{2\beta}\|\nabla\widehat{\mathcal{R}}(w_i)\|^2 \leq \widehat{\mathcal{R}}(w_i) - \widehat{\mathcal{R}}(w_{i+1})$.

Proof. Let w be given, and define the convex quadratic

$$Q_w(v) := \widehat{\mathcal{R}}(w) + \left\langle \nabla\widehat{\mathcal{R}}(w), v - w \right\rangle + \frac{\lambda}{2}\|v - w\|^2,$$

which attains its minimum at $\bar{v} := w - \nabla\widehat{\mathcal{R}}(w)/\lambda$. By definition λ-sc,

$$\inf_v \widehat{\mathcal{R}}(v) \geq \inf_v Q_w(v) = Q_w(\bar{v}) = \widehat{\mathcal{R}}(w) - \frac{1}{2\lambda}\|\nabla\widehat{\mathcal{R}}(w)\|^2.$$

Remark 7.10 ***(stopping conditions)*** Say our goal is to find w so that $\widehat{\mathcal{R}}(w) - \inf_v \widehat{\mathcal{R}}(v) \leq \epsilon$. When do we stop gradient descent?

- The λ-sc case is easy: by the preceding lemma, we know that we can stop when $\|\nabla \widehat{\mathcal{R}}(w)\| \leq \sqrt{2\lambda\epsilon}$.
- Another easy case is when $\inf_v \widehat{\mathcal{R}}(v)$ is known, and we just watch $\widehat{\mathcal{R}}(w_i)$. E.g., in classification tasks, deep networks are expect to get 0. For things like deep RL, once again it becomes a problem.
- Many software packages use heuristics. Some people just run their methods as long as possible. In convex cases, sometimes we can compute duality gaps.

Remark 7.11 ***(Regularization and boundedness)***

- Given $\widehat{\mathcal{R}}_\lambda(w) = \widehat{\mathcal{R}}(w) + \lambda\|w\|^2/2$ with $\widehat{\mathcal{R}} \geq 0$, optimal point $\bar{w}$ satisfies

$$\frac{\lambda}{2}\|\bar{w}\|_2^2 \leq \widehat{\mathcal{R}}_\lambda(\bar{w}) \leq \widehat{\mathcal{R}}_\lambda(0) = \widehat{\mathcal{R}}(0),$$

  thus it suffices to search over bounded set $\{w \in \mathbb{R}^p : \|w\|^2 \leq 2\widehat{\mathcal{R}}(0)/\lambda\}$. This can often be plugged directly into generalization bounds.
- In deep learning, this style of regularization ("weight decay") is indeed used, but it isn't necessary for generalization, and is much smaller than what many generalization analyses suggest, and thus its overall role is unclear.

[mjt☹: I shuold lemmas lemmas giving level set containment, and existence of minimizers.]

7.2.1 Rates when strongly convex and smooth

Theorem 7.5 Suppose $\widehat{\mathcal{R}}$ is λ-sc and β-smooth, and GD is run with step size $1/\beta$. Then a minimum $\bar{w}$ exists, and

$$\widehat{\mathcal{R}}(w_t) - \widehat{\mathcal{R}}(\bar{w}) \leq \left(\widehat{\mathcal{R}}(w_0) - \widehat{\mathcal{R}}(\bar{w})\right)\exp(-t\lambda/\beta),$$
$$\|w_t - \bar{w}\|^2 \leq \|w_0 - \bar{w}\|^2 \exp(-t\lambda/\beta).$$

Proof. Using previously-proved Lemmas from smooothness and strong convexity,

$$\begin{aligned}
\widehat{\mathcal{R}}(w_{i+1}) - \widehat{\mathcal{R}}(\bar{w}) &\leq \widehat{\mathcal{R}}(w_i) - \widehat{\mathcal{R}}(\bar{w}) - \frac{\|\nabla\widehat{\mathcal{R}}(w_i)\|^2}{2\beta} \\
&\leq \widehat{\mathcal{R}}(w_i) - \widehat{\mathcal{R}}(\bar{w}) - \frac{2\lambda(\widehat{\mathcal{R}}(w_i) - \widehat{\mathcal{R}}(\bar{w}))}{2\beta} \\
&\leq \left(\widehat{\mathcal{R}}(w_i) - \widehat{\mathcal{R}}(\bar{w})\right)(1 - \lambda/\beta),
\end{aligned}$$

which gives the first bound by induction since

$$\prod_{i<t}(1 - \lambda/\beta) \leq \prod_{i<t}\exp\left(-\lambda/\beta\right) = \exp\left(-t\lambda/\beta\right).$$

For the second guarantee, expanding the square as usual,

$$\begin{aligned}\|w' - \bar{w}\|^2 &= \|w - \bar{w}\|^2 + \frac{2}{\beta}\left\langle \nabla\widehat{\mathcal{R}}(w), \bar{w} - w\right\rangle + \frac{1}{\beta^2}\|\nabla\widehat{\mathcal{R}}(w)\|^2 \\ &\leq \|w - \bar{w}\|^2 + \frac{2}{\beta}\left(\widehat{\mathcal{R}}(\bar{w}) - \widehat{\mathcal{R}}(w) - \frac{\lambda}{2}\|\bar{w} - w\|_2^2\right) \\ &\quad + \frac{1}{\beta^2}\left(2\beta(\widehat{\mathcal{R}}(w) - \widehat{\mathcal{R}}(w'))\right) \\ &= (1 - \lambda/\beta)\|w - \bar{w}\|^2 + \frac{2}{\beta}\left(\widehat{\mathcal{R}}(\bar{w}) - \widehat{\mathcal{R}}(w) + \widehat{\mathcal{R}}(w) - \widehat{\mathcal{R}}(w')\right) \\ &\leq (1 - \lambda/\beta)\|w - \bar{w}\|^2,\end{aligned}$$

which gives the argument after a similar induction argument as before.

Remark 7.12

- β/λ is sometimes called the *condition number*, based on linear system solvers, where it is $\sigma_{\max}(X)/\sigma_{\min}(X)$ as in least squares. Note that $\beta \geq \lambda$ and a good condition numbers improves these bounds.
- Setting the bounds to ϵ, it takes a linear number of iterations to learn a linear number of bits of $\bar{w}$.
- Much of the analysis we've done goes through if the norm pair $(\|\cdot\|_2, \|\cdot\|_2)$ is replaced with $(\|\cdot\|, \|\cdot\|_*)$ where the latter *dual norm* is defined as

  $$\|s\|_* = \sup\{\langle s, w\rangle : \|w\| \leq 1\};$$

  for instance, we can define β-smooth wrt $\|\cdot\|$ as

  $$\|\nabla\widehat{\mathcal{R}}(w) - \nabla\widehat{\mathcal{R}}(w')\|_* \leq \beta\|w - w'\|.$$

Next let's handle the gradient flow.

Theorem 7.6 If $\widehat{\mathcal{R}}$ is λ-sc, a minimum $\bar{w}$ exists, and the GF $w(t)$ satisfies

$$\begin{aligned}\|w(t) - \bar{w}\|^2 &\leq \|w(0) - \bar{w}\|^2 \exp(-2\lambda t), \\ \widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(\bar{w}) &\leq \left(\widehat{\mathcal{R}}(w(0)) - \widehat{\mathcal{R}}(\bar{w})\right)\exp(-2t\lambda).\end{aligned}$$

Proof. By first-order optimality in the form $\nabla\widehat{\mathcal{R}}(\bar{w}) = 0$, then

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\|w(t) - \bar{w}\|^2 &= \langle w(t) - \bar{w}, \dot{w}(t)\rangle \\ &= -\left\langle w(t) - \bar{w}, \nabla\widehat{\mathcal{R}}(w(t)) - \nabla\widehat{\mathcal{R}}(\bar{w})\right\rangle \\ &\leq -\lambda\|w(t) - \bar{w}\|^2.\end{aligned}$$

By Grönwall's inequality, this implies

$$\begin{aligned}\|w(t) - \bar{w}\|^2 &\leq \|w(0) - \bar{w}\|^2 \exp\left(-\int_0^t 2\lambda \mathrm{d}s\right) \\ &\leq \|w(0) - \bar{w}\|^2 \exp(-2\lambda t),\end{aligned}$$

which establishes the guarantee on distances to initialization. For the objective function guarantee,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(\bar{w})) &= \left\langle \nabla\widehat{\mathcal{R}}(w(t)), \dot{w}(t) \right\rangle \\ &= -\left\| \nabla\widehat{\mathcal{R}}(w(t)) \right\|^2 \leq -2\lambda(\widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(\bar{w})).\end{aligned}$$

Grönwall's inequality implies

$$\widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(\bar{w}) \leq \left(\widehat{\mathcal{R}}(w(0)) - \widehat{\mathcal{R}}(\bar{w})\right) \exp(-2t\lambda).$$

Remark 7.13 As in all other rates proved for GF and GD, time t is replaced by "arc length units" t/β.

We have strayed a little from our goals by producing laborious proofs that not only separate the objective function and the distances, but also require minimizers. Interestingly, we can resolve this by changing the step size to a large (seemingly worse?) one.

Theorem 7.7 Suppose $\widehat{\mathcal{R}}$ is β-smooth and λ-sc, and a constant step size $\frac{2}{\beta+\lambda}$. Then, for any z,

$$\widehat{\mathcal{R}}(w_t) - \widehat{\mathcal{R}}(z) + \frac{\lambda}{2}\|w_t - z\|^2 \leq \left[\frac{\beta-\lambda}{\beta+\lambda}\right]^t \left(\widehat{\mathcal{R}}(w_0) - \widehat{\mathcal{R}}(z) + \frac{\lambda}{2}\|w_0 - z\|^2\right).$$

Proof. Homework problem ◡̈.

Remark 7.14 ***(standard rates with strong convexity)*** Compared with standard proofs in the literature (Nesterov 2003, chap. 2), the preceding bound with step size $2/(\beta+\lambda)$ is possibly loose: it seems possible to have a $2t$ and not just t in the exponent, albeit after adjusting the other terms (and depending explicitly on minimizers). [mjt☹: I need to resolve what's going on here. . .]

Moreover, another standard rate given in the literature is $1/t$ under just strong convexity (no smoothness); however, this requires a step size $\eta_i := (\lambda(i+1))^{-1}$.

7.3 Stochastic gradients

Let's generalize gradient descent, and consider the iteration

$$w_{i+1} := w_i - \eta_i g_i,$$

where each g_i is merely some vector. If $g_i := \nabla\widehat{\mathcal{R}}(w_i)$, then we have gradient descent, but in general we only approximate it. Later in this section, we'll explain how to make g_i a "stochastic gradient."

Our first step is to analyze this in our usual way with our favorite potential function, but accumulating a big error term: using convexity of $\mathcal{R}$ and choosing a constant step size $\eta_i := \eta \geq 0$ for simplicity,

$$\begin{aligned}\|w_{i+1} - z\|^2 &= \|w_i - \eta g_i - z\|^2 \\ &= \|w_i - z\|^2 - 2\eta_i \langle g_i, w_i - z\rangle + \eta^2\|g_i\|^2 \\ &= \|w_i - z\|^2 + 2\eta \langle g_i - \nabla\mathcal{R}(w_i) + \nabla\mathcal{R}(w_i), z - w_i\rangle + \eta^2\|g_i\|^2 \\ &\leq \|w_i - z\|^2 + 2\eta(\mathcal{R}(z) - \mathcal{R}(w_i) + \underbrace{\langle g_i - \nabla\mathcal{R}(w_i), z - w_i\rangle}_{\epsilon_i}) + \eta^2\|g_i\|^2,\end{aligned}$$

which after rearrangement gives

$$2\eta\mathcal{R}(w_i) \le 2\eta\mathcal{R}(z) + \|w_i - z\|^2 - \|w_{i+1} - z\|^2 + 2\eta\epsilon_i + \eta^2\|g_i\|^2,$$

and applying $\frac{1}{2\eta t}\sum_{i<t}$ to both sides gives

$$\frac{1}{t}\sum_{i<t}\mathcal{R}(w_i) \le \mathcal{R}(z) + \frac{\|w_0 - z\|^2 - \|w_t - z\|^2}{2\eta t} + \frac{1}{t}\sum_{i<t}\left(\epsilon_i + \frac{\eta}{2}\|g_i\|^2\right).$$

The following lemma summarizes this derivation.

Lemma 7.2 Suppose $\mathcal{R}$ convex; set $G := \max_i \|g_i\|_2$, and $\eta := \frac{c}{\sqrt{t}}$. For any z,

$$\mathcal{R}\left(\frac{1}{t}\sum_{i<t} w_i\right) \le \frac{1}{t}\sum_{i<t}\mathcal{R}(w_i) \le \mathcal{R}(z) + \frac{\|w_0 - z\|^2}{2c\sqrt{t}} + \frac{cG^2}{2\sqrt{t}} + \frac{1}{t}\sum_{i<t}\epsilon_i.$$

Proof. This follows from the earlier derivation after plugging in G, $\eta = c/\sqrt{t}$, and applying Jensen's inequality to the left hand side.

Remark 7.15

- We get a bound on the averaged iterate or a minimum iterate, but not the last iterate! (We'll revisit this later.) Averaged iterates are often suggested in theory, but rare in applied classification (afaik), but I've heard of them used in deep RL; OTOH, averaging seems weird with nonconvexity?
- $\eta = c/\sqrt{t}$ trades off between terms. If t not fixed in advance, can use $\eta_i = c/\sqrt{1+i}$, but I'd rather shorten lecture a little by avoiding the needed algebra with non-constant step sizes, and for deep learning at least this style seems to not work well.
- This analysis works fine with $\nabla\mathcal{R}(w_i)$ replaced with subgradient $s_i \in \partial\mathcal{R}(w_i)$.
- Suppose $\|\nabla\mathcal{R}(w_i)\| \le G$ and set $D := \max_i \|w_i - z\|$, then by Cauchy-Schwarz

  $$\frac{1}{t}\sum_{i<t}\epsilon_i \le \frac{1}{t}\sum_{i<t}\langle g_i - \nabla\mathcal{R}(w_i), z - w_i\rangle \le 2GD,$$

  which does not go to 0 with t! Thus more structure needed on ϵ_i, this worst-case argument is bad.
- This proof easily handles projection to convex closed sets: replace $w_i - \eta g_i$ with $\Pi_S(w_i - \eta g_i)$, and within the proof use the non-expansive property of Π_S. This can be used to ensure that D up above is not too large. (We'll return to this point.)

Now let us define the standard stochastic gradient oracle:

$$\mathbb{E}[g_i | w_{\le i}] = \nabla\mathcal{R}(w_i),$$

where $w_{\le i}$ signifies all randomness in $(w_1, \ldots, w_i)$.

Remark 7.16

- We can't use an unconditional expectation because gradient at w_i should rely upon random variable w_i !

- One way to satisfy this: sample (x, y), and set $g_i := \ell'(yf(x; w_i))y\nabla_w f(x; w_i)$; conditioned on $w_{\leq i}$, the only randomness is in (x, y), and the conditional expectation is a gradient *over the distribution*!

Indeed, this setup allows the expectation to be nicely interpreted as an iterated integral over (x_1, y_1), then (x_2, y_2), and so on. The stochastic gradient g_i depends on (x_i, y_i) and w_i, but w_i does not depend on (x_i, y_i), rather on $((x_j, y_j))_{j=1}^{i-1}$.

- It's standard to sample a *minibatch* and average the g_i obtained from each, which ostensibly has the same conditional mean as g_i, but improved variance. It can be hard to analyze this.
- Stochastic minibatch gradient descent is standard for deep networks. However, there is a delicate interplay between step size, minibatch size, and number of training epochs (Shallue et al. 2018).
- Annoyingly, there are many different settings for stochastic gradient descent, but they refer to themselves in the same way and it requires a closer look to determine the precise setting.
- Previous slide suggested (x, y) is a fresh sample from the distribution; in this case, we are doing stochastic gradient descent on he population directly!
- We can also resample the training set, in which case $\mathcal{R}$ is our usual empirical risk, and now the randomness is under our control (randomized algorithm, not random data from nature). The "SVRG/SDCA/SAG/etc" papers are in this setting, as are some newer SGD papers. Since people typically do multiple passes over the time, perhaps this setting makes sense.
- There are many deep learning papers that claim SGD does miraculous things to the optimization process. Unfortunately, none of these seem to come with a compelling and general theoretical analysis. Personally I don't know if SGD works further miracles (beyond computational benefits), but it's certainly interesting!

Now let's work towards our goal of showing that, with high probability, our stochastic gradient method does nearly as well as a regular gradient method. (We will not show any *benefit* to stochastic noise, other than computation!)

Our main tool is as follows.

Theorem 7.8 ***(Azuma-Hoeffding)*** Suppose $(Z_i)_{i=1}^n$ is a martingale difference sequence $(\mathbb{E}(Z_i|Z_{<i}) = 0)$ and $\mathbb{E}\,|Z_i| \leq R$. Then with probability at least $1 - \delta$,

$$\sum_i Z_i \leq R\sqrt{2t\ln(1/\delta)}.$$

Proof omitted, though we'll sketch some approaches in a few weeks.

We will use this inequality to handle $\sum_{i<t} \epsilon_i$. Firstly, we must show the desired expectations are

zero. To start,

$$\begin{aligned}\mathbb{E}\left[\epsilon_i \;\middle|\; w_{\le i}\right] &= \mathbb{E}\left[\langle g_i - \nabla\mathcal{R}(w_i), z - w_i\rangle \;\middle|\; w_{\le i}\right] \\ &= \left\langle \mathbb{E}\left[g_i - \nabla\mathcal{R}(w_i) \;\middle|\; w_{\le i}\right], z - w_i\right\rangle \\ &= \langle 0, z - w_i\rangle \\ &= 0.\end{aligned}$$

Next, by Cauchy-Schwarz and the triangle inequality,

$$\mathbb{E}\,|\epsilon_i| = \mathbb{E}\left|\left\langle g_i - \nabla\widehat{\mathcal{R}}(w_i), w_i - z\right\rangle\right| \le \mathbb{E}\left(\|g_i\| + \|\nabla\widehat{\mathcal{R}}(w_i)\|\right)\|w_i - z\| \le 2GD.$$

Consequently, by Azuma-Hoeffding, with probability at least $1-\delta$,

$$\sum_i \epsilon_i \le 2GD\sqrt{2t\ln(1/\delta)}.$$

Plugging this into the earlier approximate gradient lemma gives the following. [mjt☹: should give explicit cref]

Lemma 7.3 Suppose $\mathcal{R}$ convex; set $G := \max_i \|g_i\|_2$, and $\eta := \frac{1}{\sqrt{t}}$, $D \ge \max_i \|w_i - z\|$, and suppose g_i is a stochastic gradient at time i. With probability at least $1-\delta$,

$$\begin{aligned}\mathcal{R}\left(\frac{1}{t}\sum_{i<t} w_i\right) &\le \frac{1}{t}\sum_{i<t}\mathcal{R}(w_i) \\ &\le \mathcal{R}(z) + \frac{D^2}{2\sqrt{t}} + \frac{G^2}{2\sqrt{t}} + \frac{2DG\sqrt{2\ln(1/\delta)}}{\sqrt{t}}.\end{aligned}$$

Remark 7.17

- If we tune $\eta = c/\sqrt{t}$ here, we only get a DG term. [mjt☹: I should do it]
- We can ensure D is small by projecting to a small set each iteration. By the contractive property of projections, the analysis still goes through.
- By the tower property of conditional expectation, meaning $\mathbb{E} = \mathbb{E}\,\mathbb{E}[\cdot|w_{\le i}]$, without Azuma-Hoeffding we easily get a bound on the expected average error:

$$\mathbb{E}\left[\frac{1}{t}\sum_{i<t}\mathcal{R}(w_i)\right] \le \mathcal{R}(z) + \frac{\|w_0 - z\|^2}{2\sqrt{t}} + \frac{G^2}{2\sqrt{t}}.$$

- If the preceding bound in expectation is sufficient, expected is enough, a more careful analysis lets us use the last iterate (Shamir and Zhang 2013); AFAIK a high probability version still doesn't exist.
- The Martingale structure is delicate: if we re-use even a single data-point, then we can't treat $\mathcal{R}$ as the population risk, but instead as the empirical risk. [mjt☹: and here my notation is truly frustrating.]
- In practice, randomly sampling a permutation over the training data at the beginning of each epoch is common; it can be hard to analyze.

- **Why SGD in ML?** In statistical problems, we shouldn't expect test error better than $\frac{1}{\sqrt{n}}$ or $\frac{1}{n}$ anyway, so we shouldn't optimize to crazy accuracy. With SGD, the per-iteration cost is low. Meanwhile, heavyweight solvers like Newton methods require a massive per-iteration complexity, with the promise of crazy accuracy; but, again we don't need that crazy accuracy here. [mjt☹: summarize as "computation."]

8 Two NTK-based optimization proofs near initializatikon

Here we will show our first optimization guarantees for (shallow) networks: one based on strong convexity, and one based on smoothness.

under construction.

8.1 Strong convexity style NTK optimization proof

(include preamble saying this looks like + **theorem:sc_smooth?**){.mjt} Theorem 7.5

Finally we will prove (rather than assert) that we can stay close to initialization long enough to get a small risk with an analysis that is essentially convex, essentially following the NTK (Taylor approximation).

- This proof is a simplification of one by Chizat and Bach (2019). There are enough differences that it's worth checking the original.
  - That paper highlights a "scaling phenomenon" as an explanation of the NTK. Essentially, increasing with always decreases initialization variance, and the paper argues this corresponds to "zooming in" on the Taylor expansion in function space, and flattening the dynamics.
  - This "scaling perspective" pervades much of the NTK literature and I recommend looking at (Chizat and Bach 2019) for further discussion; I do not discuss it much in this course or even in this proof, though I keep Chizat's $\alpha > 0$ scale parameter.
- This proof comes after many earlier NTK analyses, e.g., (Jacot, Gabriel, and Hongler 2018; Simon S. Du et al. 2018; Allen-Zhu, Li, and Liang 2018; Arora, Du, Hu, Li, and Wang 2019). I like the proof by (Chizat and Bach 2019) very much and learned a lot from it; it was the most natural for me to teach. OTOH, it is quite abstract, and we'll need homework problems to boil it down further.

Basic notation. For convenience, bake the training set into the predictor:

$$f(w) := \begin{bmatrix} f(x_1; w) \\ \vdots \\ f(x_n; w) \end{bmatrix} \in \mathbb{R}^n.$$

We'll be considering squared loss regression:

$$\widehat{\mathcal{R}}(\alpha f(w)) := \frac{1}{2}\|\alpha f(w) - y\|^2, \qquad \widehat{\mathcal{R}}_0 := \widehat{\mathcal{R}}(\alpha f(w(0))),$$

where $\alpha > 0$ is a scale factor we'll optimize later. [mjt☹: maybe I should use $\mathcal{L}$ not $\widehat{\mathcal{R}}$ since unnormalized.]

We'll consider gradient flow:

$$\dot{w}(t) := -\nabla_w \widehat{\mathcal{R}}(\alpha f(w(t))) = -\alpha J_t^\intercal \nabla \widehat{\mathcal{R}}(\alpha f(w(t))),$$
$$\text{where } J_t := J_{w(t)} := \begin{bmatrix} \nabla f(x_1; w(t))^\intercal \\ \vdots \\ \nabla f(x_n; w(t))^\intercal \end{bmatrix} \in \mathbb{R}^{n \times p}.$$

We will also explicitly define and track a flow $u(t)$ over the tangent model; what we care about is $w(t)$, but we will show that indeed $u(t)$ and $w(t)$ stay close in this setting. (Note that $u(t)$ is *not* needed for the analysis of $w(t)$.)

$$f_0(u) := f(w(0)) + J_0(u - w(0)).$$
$$\dot{u}(t) := -\nabla_u \widehat{\mathcal{R}}(\alpha f_0(u(t))) = -\alpha J_0^\intercal \nabla \widehat{\mathcal{R}}(\alpha f_0(u(t))).$$

Both gradient flows have the same initial condition:

$$u(0) = w(0), \qquad f_0(u(0)) = f_0(w(0)) = f(w(0)).$$

Remark 8.1 *(initialization, width, etc)*

- Notice that the setup so far doesn't make any mention of width, neural networks, random initialization, etc.! It's all abstracted away! This is good and bad: the good is that it highlights the "scale" phenomenon, as α is the only concretely interpretable parameter here. On the downside, we need to do some work to get statements about width etc.

Assumptions.

$$\begin{aligned} &\operatorname{rank}(J_0) = n, \\ &\sigma_{\min} := \sigma_{\min}(J_0) = \sqrt{\lambda_{\min}(J_0 J_0^\intercal)} = \sqrt{\lambda_n(J_0 J_0^\intercal)} > 0, \\ &\sigma_{\max} := \sigma_{\max}(J_0) > 0, \\ &\|J_w - J_v\| \le \beta \|w - v\|. \end{aligned} \tag{4}$$

Remark 8.2 ***($J_0 J_0^\intercal$ has full rank, a "representation assumption")*** This is a "representation assumption" in an explicit sense: it implies the tangent model has exact solutions to the least squares problem, regardless of the choice of y, meaning the training error can always be made 0. In detail, consider the least squares problem solved by the tangent space:

$$\min_{u \in \mathbb{R}^p} \frac{1}{2} \|f_0(u) - y\|^2 = \min_{u \in \mathbb{R}^p} \frac{1}{2} \|J_0 u - y_0\|^2,$$

where we have chosen $y_0 := y + J_0 w(0) - f(w(0))$ for convenience. The normal equations for this least squares problem are

$$J_0^\intercal J_0 u = J_0^\intercal y_0.$$

Let $J_0 = \sum_{i=1}^n s_i u_i v_i^\intercal$ denote the SVD of J_0, which has n terms by the rank assumption; the corresponding pseudoinverse is $J_0^\dagger = \sum_{i=1}^n s_i^{-1} v_i u_i^\intercal$. Multiplying both sides by $(J_0^\dagger)^\intercal$,

$$J_0 u = (J_0^\dagger)^\intercal J_0^\intercal J_0 u = (J_0^\dagger)^\intercal J_0^\intercal y_0 = \left[\sum_{i=1}^n u_i u_i^\intercal\right] y_0 = y_0,$$

where the last step follows since $[\sum_i u_i u_i^\mathsf{T}]$ is idempotent and full rank, and therefore the identity matrix. In particular, we can choose $\hat{u} = J_0^\dagger y_0$, then $J_0\hat{u} = [\sum_i u_i u_i^\mathsf{T}]y_0 = y_0$, and in particular

$$\frac{1}{2}\|f_0(\hat{u}) - y\|^2 = \frac{1}{2}\|J_0\hat{u} - y_0\|^2 = 0.$$

As such, the full rank assumption is explicitly a representation assumption: we are forcing the tangent space least squares problem to always have solutions.

Theorem 8.1 ***(see also (Theorem 3.2, Chizat and Bach 2019))*** Assume eq. 4 and $\alpha \geq \frac{\beta\sqrt{1152\sigma_{\max}^2\widehat{\mathcal{R}}_0}}{\sigma_{\min}^3}$. Then

$$\max\left\{\widehat{\mathcal{R}}(\alpha f(w(t))), \widehat{\mathcal{R}}(\alpha f_0(u(t)))\right\} \leq \widehat{\mathcal{R}}_0 \exp(-t\alpha^2\sigma_{\min}^2/2),$$

$$\max\left\{\|w(t) - w(0)\|, \|u(t) - w(0)\|\right\} \leq \frac{3\sqrt{8\sigma_{\max}^2\widehat{\mathcal{R}}_0}}{\alpha\sigma_{\min}^2}.$$

Remark 8.3 ***(shallow case)*** To get a handle on the various abstract constants and what they mean, consider the shallow case, namely $f(x; w) = \sum_j s_j\sigma(w_j^\mathsf{T}x)$, where $s_j \in \{\pm 1\}$ is not trained, and each w_j is trained.

Smoothness constant. Let $X \in \mathbb{R}^{n\times d}$ be a matrix with the n training inputs as rows, and suppose σ is β_0-smooth. Then

$$\begin{aligned}\|J_w - J_v\|_2^2 &= \sum_{i,j}\|x_i\|^2(\sigma'(w_j^\mathsf{T}x_i) - \sigma'(v_j^\mathsf{T}x_i))^2 \\ &\leq \sum_{i,j}\|x_i\|^4\beta_0^2\|w_j - v_j\|^2 \\ &= \beta_0^2\|X\|_\mathrm{F}^4\|w - v\|^2.\end{aligned}$$

Thus $\beta = \beta_0\|X\|_\mathrm{F}^2$ suffices, which we can ballpark as $\beta = \Theta(n)$.

Singular values. Now that we have an interpretation of the full rank assumption, ballpark the eigenvalues of $J_0J_0^\mathsf{T}$. By definition,

$$(J_0J_0^\mathsf{T})_{i,j} = \nabla f(x_i; w(0))^\mathsf{T}\nabla f(x_j; w(0)).$$

Holding i fixed and letting j vary, we can view the corresponding column of $(J_0J_0^\mathsf{T})$ as another feature representation, and $\mathrm{rank}(J_0) = n$ means none of these examples, in this feature representation, are linear combinations of others. This gives a concrete sense under which these eigenvalue assumptions are *representation assumptions*.

Now suppose each $w_j(0)$ is an iid copy of some random variable v. Then, by definition of J_0,

$$\begin{aligned}\mathop{\mathbb{E}}_{w(0)}(J_0J_0^\mathsf{T})_{i,j} &= \mathop{\mathbb{E}}_{w(0)}\nabla f(x_i; w(0))^\mathsf{T}\nabla f(x_j; w(0)). \\ &= \mathop{\mathbb{E}}_{w(0)}\sum_k s_k^2\sigma'(w_k(0)^\mathsf{T}x_i)\sigma'(w_k(0)^\mathsf{T}x_j)x_i^\mathsf{T}x_j \\ &= m\mathop{\mathbb{E}}_{v}\sigma'(v^\mathsf{T}x_i)\sigma'(v^\mathsf{T}x_j)x_i^\mathsf{T}x_j.\end{aligned}$$

In other words, it seems reasonable to expect $\sigma_{\min}$ and $\sigma_{\max}$ to scale with $\sqrt{m}$.

Initial risk $\widehat{\mathcal{R}}_0$. Let's consider two different random initializations.

In the first case, we use one of the fancy schemes we mentioned to force $f(w(0)) = 0$; e.g., we can make sure that s_j is positive and negative an equal number of times, then sample w_j for $s_j = +1$, and then make w_j for $s_j = -1$ be the negation. With this choice, $\widehat{\mathcal{R}}_0 = \|y\|^2/2 = \Theta(n)$.

On the other hand, if we do a general random initialization of both s_j and w_j, then we can expect enough cancellation that, roughly, $f(x_i; w(0)) = \Theta(\sqrt{m})$ (assuming w_j's variance is a constant and not depending on m: that would defeat the purpose of separating out the scale parameter α). then $\|\alpha f(w(0))\|^2 = \Theta(\alpha^2 mn)$, and $\widehat{\mathcal{R}}_0 = \Theta(\alpha^2 mn)$, and thus the lower bound condition on α will need to be checked carefully.

Combining all parameters. Again let's split into two cases, based on the initialization as discussed immediately above.

- **The case $\widehat{\mathcal{R}}_0 = \Theta(\alpha^2 nm)$.** Using $\beta = \Theta(n)$, the condition on α indeed has α on both sides, and becomes

  $$\sigma_{\min}^3 \geq \Omega(\beta \sigma_{\max} \sqrt{nm}) = \sigma_{\max} \Omega(\sqrt{mn^3}).$$

  Since we said the singular values are of order $\sqrt{m}$, we get roughly $m^{3/2} \geq \sqrt{m^2 n^3}$, thus $m \geq n^3$.

  Since the lower bound on α turned into a lower bound on m, let's plug this $\widehat{\mathcal{R}}_0$ into the rates to see how they simplify:

  $$\begin{aligned}
  \max\left\{\widehat{\mathcal{R}}(\alpha f(w(t))), \widehat{\mathcal{R}}(\alpha f_0(u(t)))\right\} &\leq \widehat{\mathcal{R}}_0 \exp\left(-\frac{t\alpha^2\sigma_{\min}^2}{2}\right) \\
  &= \mathcal{O}\left(\alpha^2 nm \exp\left(-\frac{t\alpha^2\sigma_{\min}^2}{2}\right)\right), \\
  \max\left\{\|w(t) - w(0)\|, \|u(t) - w(0)\|\right\} &\leq \frac{3\sqrt{8\sigma_{\max}^2 \widehat{\mathcal{R}}_0}}{\alpha \sigma_{\min}^2} \\
  &= \mathcal{O}\left(\frac{\sqrt{\sigma_{\max}^2 nm}}{\sigma_{\min}^2}\right).
  \end{aligned}$$

  In these inequalities, the distance to initialization is not affected by α: this makes sense, as the key work needed by the gradient flow is to clear the initial noise so that y can be fit exactly. Meanwhile, the empirical risk rate does depend on α, and is dominated by the exponential term, suggesting that α should be made arbitrarily large. There is indeed a catch limiting the reasonable choices of α, as will be pointed out shortly.

  For now, to pick a value which makes the bounds more familiar, choose $\alpha = \hat{\alpha} := 1/\sigma_{\max}$, whereby additionally simplifying via $\sigma_{\min}$ and $\sigma_{\max}$ being $\Theta(\sqrt{m})$ gives

  $$\begin{aligned}
  \max\left\{\widehat{\mathcal{R}}(\alpha f(w(t))), \widehat{\mathcal{R}}(\alpha f_0(u(t)))\right\} &= \mathcal{O}\left(\sigma_{\max}^{-2} nm \exp\left(-\frac{t\sigma_{\min}^2}{2\sigma_{\max}^2}\right)\right) \\
  &= \mathcal{O}\left(n \exp\left(-\frac{t\sigma_{\min}^2}{2\sigma_{\max}^2}\right)\right), \\
  \max\left\{\|w(t) - w(0)\|, \|u(t) - w(0)\|\right\} &= \mathcal{O}\left(\frac{\sqrt{\sigma_{\max}^2 nm}}{\sigma_{\min}^2}\right) = \mathcal{O}\left(\sqrt{n}\right).
  \end{aligned}$$

Written this way, the empirical risk rate depends on the *condition number* $\sigma_{\max}/\sigma_{\min}$ of the NTK Gram matrix, which is reminiscent of the purely strongly convex and smooth analyses as in Theorem 7.5.

- **The case $\widehat{\mathcal{R}}_0 = \Theta(n)$.** Using $\beta = \Theta(n)$, the condition on α becomes

$$\alpha = \Omega\left(\frac{\beta\sqrt{\sigma_{\max}^2\widehat{\mathcal{R}}_0}}{\sigma_{\min}^3}\right) = \Omega\left(\frac{\sigma_{\max}n^{3/2}}{\sigma_{\min}^3}\right).$$

  We have removed the cancelation from the previous case, and are now constrained in our choice of α; we can still set $\alpha := 1/\sigma_{\max}$, which after using our estimate of $\sqrt{m}$ for $\sigma_{\min}$ and $\sigma_{\max}$ get a similar requirement $m = \Omega(n^3)$. More generally, we get $\alpha = \Omega(n^{3/2}/m)$, which means for large enough m we can treat as close to $1/m$. [mjt☹: Frederic Koehler points out that the first case can still look like $\widehat{\mathcal{R}}_0 = \Theta(\alpha^2 mn + n)$ and even $\Theta(n)$ when α is small; I need to update this story.]

Possible values of α. The two preceding cases considered lower bounds on α. In the case $\widehat{\mathcal{R}}_0 = \Theta(\alpha^2 nm)$, it even seemed that we can make α whatever we want; in either case, the time required to make $\widehat{\mathcal{R}}(\alpha f(w(t)))$ small will decrease as α increases, so why not simply make α arbitrarily large?

An issue occurs once we perform time discretization. Below, we will see that the smoothness of the model looks like $\alpha^2\sigma_{\max}^2$ near initialization; as such, a time discretization, using tools such as in Theorem 7.3, will require a step size roughly $1/(\alpha^2\sigma_{\max}^2)$, and in particular while we may increase α to force the gradient flow to seemingly converge faster, a smoothness-based time discretization will need the same number of steps.

As such, $\alpha = 1/\sigma_{\max}$ seems a reasonable way to simplify many terms in this shallow setup, which translates into a familiar $1/\sqrt{m}$ NTK scaling.

::: **Proof of Theorem 8.1.**

Proof plan.

- First we choose a fortuitous radius $B := \frac{\sigma_{\min}}{2\beta}$, and seek to study the properties of weight vectors w which are B-close to initialization:

$$\|w - w(0)\| \le B;$$

  This B will be chosen to ensure J_t and J_0 are close, amongst other things. Moreover, we choose a T so that all $t \in [0, T]$ are in this good regime:

$$T := \inf\left\{t \ge 0 : \|w(t) - w(0)\| > B\right\}.$$

- Now consider any $t \in [0, T]$. [mjt☹: i should include explicit lemma pointers for each.]
  - First we show that if $J_t J_t^\intercal$ is positive definite, then we rapidly decrease risk, essentially following our old strong convexity proof.
  - Next, since the gradient of the least squares risk is the residual, then decreasing risk implies decreasing gradient norms, and in particular we can not travel far.
  - The above steps go through directly for $u(t)$ due to the positive definiteness of $J_0 J_0^\intercal$; by the choice of B, we can also prove they hold for $J_t J_t^\intercal$.

- As a consequence we also immediately get that we never escape this ball: the gradient norms decay sufficiently rapidly. Consequently, $T = \infty$, and we don't need conditions on t in the theorem!

Remark 8.4 That is to say, in this setting, α large enough (m large enough in the shallow case) ensure we stay in the NTK regime forever! This is *not* the general case.

The evolution in prediction space is

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\alpha f(w(t)) &= \alpha J_t \dot{w}(t) = -\alpha^2 J_t J_t^\intercal \nabla \widehat{\mathcal{R}}(\alpha f(w(t))), \\
&= -\alpha^2 J_t J_t^\intercal (\alpha f(w(t)) - y), \\
\frac{\mathrm{d}}{\mathrm{d}t}\alpha f_0(u(t)) &= \frac{\mathrm{d}}{\mathrm{d}t}\alpha \left(f(w(0) + J_0(u(t) - w(0))\right) = \alpha J_0 \dot{u}(t) \\
&= -\alpha^2 J_0 J_0^\intercal \nabla \widehat{\mathcal{R}}(\alpha f_0(u(t))) \\
&= -\alpha^2 J_0 J_0^\intercal (\alpha f_0(u(t)) - y).
\end{aligned}$$

The first one is complicated because we don't know how J_t evolves.

But the second one can be written

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\alpha f_0(u(t))\right] = -\alpha^2 \left(J_0 J_0^\intercal\right)\left[\alpha f_0(u(t))\right] + \alpha^2 \left(J_0 J_0^\intercal\right) y,$$

which is a concave quadratic *in the predictions* $\alpha f_0(u(t))$.

Remark 8.5 The original NTK paper, (Jacot, Gabriel, and Hongler 2018), had as its story that GF follows a gradient in kernel space. Seeing the evolution of $\alpha f_0(u(t))$ makes this clear, as it is governed by $J_0 J_0^\intercal$, the Gram or kernel matrix!

Let's fantasize a little and suppose $(J_w J_w)^\intercal$ is also positive semi-definite. Do we still have a nice convergence theory?

Lemma 8.1 Suppose $\dot{z}(t) = -Q(t)\nabla \widehat{\mathcal{R}}(z(t))$ and $\lambda := \inf_{t\in[0,\tau]} \lambda_{\min} Q(t) > 0$. Then for any $t \in [0, \tau]$,

$$\widehat{\mathcal{R}}(z(t)) \leq \widehat{\mathcal{R}}(z(0)) \exp(-2t\lambda).$$

Remark 8.6 A useful consequence is

$$\|z(t) - y\| = \sqrt{2\widehat{\mathcal{R}}(z(t))} \leq \sqrt{2\widehat{\mathcal{R}}(z(0)) \exp(-2t\lambda)} = \|z(0) - y\| \exp(-t\lambda).$$

Proof. Mostly just repeating our old strong convexity steps,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\|z(t) - y\|^2 &= \langle -Q(t)(z(t) - y), z(t) - y\rangle \\
&\leq -\lambda_{\min}\left(Q(t)\right)\langle z(t) - y, z(t) - y\rangle \\
&\leq -2\lambda \|z(t) - y\|^2/2,
\end{aligned}$$

and Grönwall's inequality completes the proof.

We can also prove this setting implies we stay close to initialization.

Lemma 8.2 Suppose $\dot{v}(t) = -S(t)^\top \nabla \widehat{\mathcal{R}}(g(v(t)))$, where $S_t S_t^\top = Q_t$, and $\lambda_i(Q_t) \in [\lambda, \lambda_1]$ for $[0, \tau]$. Then for $t \in [0, \tau]$,

$$\|v(t) - v(0)\| \le \frac{\sqrt{\lambda_1}}{\lambda}\|g(v(0)) - y\| \le \frac{\sqrt{2\lambda_1 \widehat{\mathcal{R}}(g(v(0)))}}{\lambda}.$$

Proof.

$$\begin{aligned}
\|v(t) - v(0)\| &= \left\|\int_0^t \dot{v}(s)\mathrm{d}s\right\| \le \int_0^t \|\dot{v}(s)\|\mathrm{d}s \\
&= \int_0^t \|S_t^\top \nabla \mathcal{R}(g(v(s)))\|\mathrm{d}s \\
&\le \sqrt{\lambda_1}\int_0^t \|g(v(s)) - y\|\mathrm{d}s \\
&\overset{(*)}{\le} \sqrt{\lambda_1}\|g(v(0)) - y\|\int_0^t \exp(-s\lambda)\mathrm{d}s \\
&\le \frac{\sqrt{\lambda_1}}{\lambda}\|g(v(0)) - y\| \\
&\le \frac{\sqrt{2\lambda_1 \widehat{\mathcal{R}}(g(v(0)))}}{\lambda},
\end{aligned}$$

where $(*)$ used +Lemma 8.1.

Where does this leave us?

We can apply the previous two lemmas to the *tangent model* $u(t)$, since for any $t \ge 0$,

$$\dot{u}(t) = -\alpha J_0^\top \nabla \widehat{\mathcal{R}}(\alpha f_0(u(t))), \quad \frac{\mathrm{d}}{\mathrm{d}t}\alpha f_0(u(t)) = -\alpha^2 (J_0 J_0^\top)\nabla \widehat{\mathcal{R}}(\alpha f_0(u(t))).$$

Thus since $Q_0 := \alpha^2 J_0 J_0^\top$ satisfies $\lambda_i(Q_0) \in \alpha^2[\sigma_{\min}^2, \sigma_{\max}^2]$,

$$\begin{aligned}
\widehat{\mathcal{R}}(\alpha f_0(u(t))) &\le \widehat{\mathcal{R}}_0 \exp(-2t\alpha^2\sigma_{\min}^2), \\
\|u(t) - u(0)\| &\le \frac{\sqrt{2\sigma_{\max}^2 \widehat{\mathcal{R}}_0}}{\alpha\sigma_{\min}^2}.
\end{aligned}$$

How about $w(t)$?

Let's relate $(J_w J_w^\top)$ to $(J_0 J_0^\top)$.

Lemma 8.3 Suppose $\|w - w(0)\| \le B = \frac{\sigma_{\min}}{2\beta}$. Then

$$\begin{aligned}
\sigma_{\min}(J_w) &\ge \sigma_{\min} - \beta\|w - w(0)\|_2 \ge \frac{\sigma_{\min}}{2}, \\
\sigma_{\max}(J_w) &\le \frac{3\sigma_{\max}}{2}.
\end{aligned}$$

Proof. For the upper bound,

$$\|J_w\| \le \|J_0\| + \|J_w - J_0\| \le \|J_0\| + \beta\|w - w(0)\| \le \sigma_{\max} + \beta B = \sigma_{\max} + \frac{\sigma_{\min}}{2}.$$

For the lower bound, given vector v define $A_v := J_0^\intercal v$ and $B_v := (J_w - J_0)^\intercal v$, whereby

$$\|A_v\| \geq \sigma_{\min}\|v\|, \qquad \|B_v\| \leq \|J_w - J_0\| \cdot \|v\| \leq \beta B\|v\|,$$

and thus

$$\begin{aligned}
\sigma_{\min}(J_w)^2 &= \min_{\|v\|=1} v^\intercal J_w J_w^\intercal v \\
&= \min_{\|v\|=1} \left((J_0 + J_w - J_0)^\intercal v\right)^\intercal (J_0 + J_w - J_0)^\intercal v \\
&= \min_{\|v\|=1} \|A_v\|^2 + 2A_v^\intercal B_v + \|B_v\|^2 \\
&\geq \min_{\|v\|=1} \|A_v\|^2 - 2\|A_v\| \cdot \|B_v\| + \|B_v\|^2 \\
&= \min_{\|v\|=1} (\|A_v\| - \|B_v\|)^2 \geq \min_{\|v\|=1} (\sigma_{\min} - \beta B)^2 \|v\|^2 = \left(\frac{\sigma_{\min}}{2}\right)^2.
\end{aligned}$$

Using this, for $t \in [0, T]$,

$$\dot{w}(t) = -\alpha J_w^\intercal \nabla\widehat{\mathcal{R}}(\alpha f(w(t))), \quad \frac{\mathrm{d}}{\mathrm{d}t}\alpha f(w(t)) = -\alpha^2 (J_w J_w^\intercal)\nabla\widehat{\mathcal{R}}(\alpha f(w(t))).$$

Thus since $Q_t := \alpha^2 J_t J_t^\intercal$ satisfies $\lambda_i(Q_t) \in \alpha^2[\sigma_{\min}^2/4, 9\sigma_{\max}^2/4]$,

$$\begin{aligned}
\widehat{\mathcal{R}}(\alpha f(w(t))) &\leq \widehat{\mathcal{R}}_0 \exp(-t\alpha^2\sigma_{\min}^2/2), \\
\|w(t) - w(0)\| &\leq \frac{3\sqrt{8\sigma_{\max}^2\widehat{\mathcal{R}}_0}}{\alpha\sigma_{\min}^2} =: B'.
\end{aligned}$$

It remains to show that $T = \infty$. Invoke, for the first time, the assumed lower bound on α, namely

$$\alpha \geq \frac{\beta\sqrt{1152\sigma_{\max}^2\widehat{\mathcal{R}}_0}}{\sigma_{\min}^3},$$

which by the above implies then $B' \leq \frac{B}{2}$. Suppose contradictorily that $T < \infty$; since $t \mapsto w(t)$ is continuous, then $t \mapsto \|w(t) - w(0)\|$ is also continuous and starts from 0, and therefore $\|w(T) - w(0)\| = B > 0$ exactly. But due to the lower bound on α, we also have $\|w(T) - w(0)\| \leq \frac{B}{2} < B$, a contradiction.

This completes the proof. :::

Remark 8.7 ***(retrospective)***

- On the downside, the proof is not only *insensitive* to benefits of $w(t)$ over $u(t)$, moreover the guarantees on $w(t)$ are a *degradation* of those on $u(t)$! That is to say, this proof does not demonstrate any benefit to the nonlinear model over the linear one.
- Note that $w(t)$ and $u(t)$ are close by triangle inequality:

$$\begin{aligned}
\|w(t) - u(t)\| &\leq \|w(t) - w(0)\| + \|u(t) - w(0)\|, \\
\|\alpha f(w(t)) - \alpha f_0(u(t))\| &\leq \|\alpha f(w(t)) - y\| + \|\alpha f_0(u(t)) - y\|.
\end{aligned}$$

  [mjt☹: I should move this earlier. Somewhere I should also mention that ideally we'd have a $\|w^\intercal\|_{2,\infty}$ bound, but this proof is architecture agnostic so it wouldn't be natural.]

8.2 Smoothness-based proof

Under construction.

(include pre-amble saying this looks like + **theorem:magic_inequality?**){.mjt} Theorem 7.3

9 Nonsmoothness, Clarke differentials, and positive homogeneity

Smoothness and differentiability do not in general hold for us (ReLU, max-pooling, hinge loss, etc.).

One relaxation of the gradient is the **subdifferential set** ∂_{s} (whose elements are called **subgradients**), namely the set of tangents which lie below the predictor:

$$\partial_{\mathrm{s}}\widehat{\mathcal{R}}(w) := \left\{ s \in \mathbb{R}^p : \forall w' \centerdot \widehat{\mathcal{R}}(w') \geq \widehat{\mathcal{R}}(w) + s^{\mathsf{T}}(w' - w) \right\}.$$

- If $\widehat{\mathcal{R}} : \mathbb{R}^d \to \mathbb{R}$ is convex, then $\partial_{\mathrm{s}}\widehat{\mathcal{R}}$ is nonempty *everywhere*.
- If $\nabla\widehat{\mathcal{R}}$ exists and $\widehat{\mathcal{R}}$ is convex, then $\partial_{\mathrm{s}}\widehat{\mathcal{R}}(w) = \{\nabla\widehat{\mathcal{R}}(w)\}$. [mjt☹: does this need some continuity on $\widehat{\mathcal{R}}$? need to check and provide a reference.]
- Much of convex analysis and convex opt can use subgradients in place of gradients; cf. (Hiriart-Urruty and Lemaréchal 2001; Nesterov 2003). As an example from these notes, Lemma 7.2 can replace gradients with subgradients.

One fun application is a short proof of Jensen's inequality.

Lemma 9.1 ***(Jensen's inequality)*** Suppose random variable X is supported on a set S, and f is convex on S. Then $\mathbb{E}\, f(X) \geq f(\mathbb{E}\, X)$.

Proof. Choose any $s \in \partial_{\mathrm{s}} f(\mathbb{E}\, X)$, and note

$$\mathbb{E}\, f(X) \geq \mathbb{E}\left[f(\mathbb{E}(X)) + s^{\mathsf{T}}(X - \mathbb{E}\, X) \right] = f(\mathbb{E}\, X).$$

Typically, we lack convexity, and the subdifferential set is empty.
Our main formalism is the **Clarke differential** (Clarke et al. 1998):

$$\partial\widehat{\mathcal{R}}(w) := \operatorname{conv}\left(\left\{ s \in \mathbb{R}^p : \exists w_i \to w, \nabla\widehat{\mathcal{R}}(w_i) \to s \right\} \right).$$

Definition 9.1 f is **locally Lipschitz** when for every point x, there exists a neighborhood $S \supseteq \{x\}$ such that f is Lipschitz when restricted to S.

Key properties:

- If $\widehat{\mathcal{R}}$ is locally Lipschitz, $\partial\widehat{\mathcal{R}}$ exists *everywhere*.
- If $\widehat{\mathcal{R}}$ is convex, then $\partial\widehat{\mathcal{R}} = \partial_{\mathrm{s}}\widehat{\mathcal{R}}$ everywhere. [mjt☹: need to check some continuity conditions and add a reference.]
- $\widehat{\mathcal{R}}$ is continuously differentiable at w iff $\partial\widehat{\mathcal{R}}(w) = \{\nabla\widehat{\mathcal{R}}(w)\}$.

We can replace the gradient flow differential equation $\dot{w}(t) = -\nabla\mathcal{R}(w(t))$ with a **differential inclusion**:

$$\dot{w}(t) \in -\partial\widehat{\mathcal{R}}(w(t)) \qquad \text{for a.e. } t \geq 0.$$

If R satisfies some technical structural conditions, then the following nice properties hold; these properties are mostly taken from (Lemma 5.2, Theorem 5.8, Davis et al. 2018) (where the structural condition is C^1 Whitney stratifiability), which was slightly generalized in (Ji and Telgarsky 2020) under o-minimal definability; another alternative, followed in (Lyu and Li 2019), is to simply assume that a chain rule holds.

- **(Chain rule.)** For a.e. $t \geq 0$ and every $v \in \partial \widehat{\mathcal{R}}(w(t))$, then $\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathcal{R}}(w(t)) = -\langle v, \dot{w}(t)\rangle$. This is the key strong property; since it holds for every element v of the Clarke differential simultaneously, it implies the next property.
- **(Minimum norm path.)** For almost every $t \geq 0$, then $\dot{w}(t) = -\arg\min\{\|v\| : v \in \partial\widehat{\mathcal{R}}(w(t))\}$. Consequently,

$$\widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(w(0)) = \int_0^t \frac{\mathrm{d}}{\mathrm{d}s}\widehat{\mathcal{R}}(w(s))\mathrm{d}s = -\int_0^t \min\{\|v\|^2 : v \in \partial\widehat{\mathcal{R}}(w(s))\}\mathrm{d}s;$$

  since the right hand size is nonpositive for all t, the flow never increases the objective.

This allows us to reprove our stationary point guarantee from an earlier lecture: since

$$\widehat{\mathcal{R}}(w(t)) - \widehat{\mathcal{R}}(w(0)) = -\int_0^t \min\{\|v\|^2 : v \in \partial\widehat{\mathcal{R}}(w(s))\}\mathrm{d}s \leq -t \min_{\substack{s\in[0,t]\\ v\in\partial\widehat{\mathcal{R}}(w(s))}} \|v\|^2,$$

then just as before

$$\min_{\substack{s\in[0,t]\\ v\in\partial\widehat{\mathcal{R}}(w(s))}} \|v\|^2 \leq \frac{\widehat{\mathcal{R}}(w(0)) - \widehat{\mathcal{R}}(w(t))}{t},$$

thus for some time $s \in [0,t]$, we have an iterate $w(s)$ which is an approximate stationary point.

Remark 9.1 Let's go back to $\dot{w}(t) := \arg\min\{\|v\| : v \in -\partial\widehat{\mathcal{R}}(w(t))\}$, which we said will hold almost everywhere.

This is *not* satisfied by pytorch/tensorflow/jax/...

(Kakade and Lee 2018) gives some bad examples, e.g.,

$$x \mapsto \sigma(\sigma(x)) - \sigma(-x)$$

with σ the ReLU, evaluated at 0. (Kakade and Lee 2018) also give a randomized algorithm for finding good subdifferentials.

Does it matter? In the NTK regime, few activations change. In practice, many change, but it's unclear what their effect is.

9.1 Positive homogeneity

Another tool we will use heavily outside convexity is *positive homogeneity*.

Definition 9.2 g is *positive homogeneous of degree* L when $g(\alpha x) = \alpha^L g(x)$ for $\alpha \geq 0$. (We will only consider continuous g, so $\alpha > 0$ suffices.)

Example 9.1

- Single ReLU: $\sigma(\alpha r) = \alpha\sigma(r)$.

- Monomials of degree L are positive homogeneous of degree L:

$$\prod_{i=1}^{d}(\alpha x_i)^{p_i} = \alpha^{\sum_i p_i}\prod_i x_i^{p_i} = \alpha^L \prod_i x_i^{p_i}.$$

Remark 9.2 The math community also has a notion of homogeneity without positivity; the monomial example above works with $\alpha < 0$. Homogeneity in math is often tied to polynomials and generalizations thereof.

Example 9.2

- A polynomial $p(x)$ is L-homogeneous when all monomials have the same degree; by the earlier calculation,

$$p(\alpha x) = \sum_{j=1}^{r} m_j(\alpha x) = \alpha^L \sum_{j=1}^{r} m_j(x).$$

  The algebraic literature often discusses "homogeneous polynomials."

- Norms are 1-homogeneous, meaning $\|\alpha x\| = \alpha\|x\|$ for $\alpha > 0$. But they moreover satisfy a stronger property $\|\alpha x\| = |\alpha| \cdot \|x\|$ when $\alpha < 0$. Also, ℓ_p norms are obtained by taking the root of a homogeneous polynomial, which in general changes the degree of a homogeneous function.

- Layers of a ReLU network are 1-homogeneous in the parameters for that layer:

$$\begin{aligned} &f(x; (W_1, \dots, \alpha W_i, \dots, W_L)) \\ &= W_L\sigma(W_{L-1}\sigma(\dots \alpha W_i \sigma(\dots W_1 x \dots)\dots)) \\ &= \alpha W_L\sigma(W_{L-1}\sigma(\dots W_i \sigma(\dots W_1 x \dots)\dots)) \\ &= \alpha f(x; w). \end{aligned}$$

  The *entire network* is L-homogeneous in the full set of parameters:

$$\begin{aligned} f(x; \alpha w) &= f(x; (\alpha W_1, \dots, \alpha W_L)) \\ &= \alpha W_L\sigma(\alpha W_{L-1}\sigma(\dots \sigma(\alpha W_1 x)\dots)) \\ &= \alpha^L W_L\sigma(W_{L-1}\sigma(\dots \sigma(W_1 x)\dots)) \\ &= \alpha^L f(x; w). \end{aligned}$$

  What is the homogeneity as a function of the input?

- Homework will cover some nonsmooth architectures that are *not* positive homogeneous!

9.2 Positive homogeneity and the Clarke differential

Let's work out an element of the Clarke differential for a ReLU network

$$x \mapsto W_L\sigma_{L-1}(\cdots W_2\sigma_1(W_1 x)).$$

As a function of x, this mapping is 1-homogeneous and piecewise affine. As a function of $w = (W_L, \cdot, W_1)$, it is L-homogeneous and piecewise polynomial. The boundary regions form a set of (Lebesgue) measure zero (wrt to either weights or parameters).

Fixing x and considering w, interior to each piece, the mapping is differentiable. Due to the definition of Clarke differential, it therefore suffices to compute the gradients in all adjacent pieces, and then take their convex hull.

Remark 9.3 Note that we are *not* forming the differential by choosing an arbitrary differential element for each ReLU: we are doing a more complicated region-based calculation. However, the former is what pytorch does.

So let's return to considering some w where are differentiable. Let A_i be a diagonal matrix with activations of the output after layer i on the diagonal:

$$A_i = \operatorname{diag}\left(\sigma'(W_i\sigma(\dots\sigma(W_1x)\dots))\right),$$

(note we've baked in x,) and so $\sigma(r) = r\sigma'(r)$ implies layer i outputs

$$x \mapsto A_iW_i\sigma(\dots\sigma(W_1x)\dots)) = A_iW_iA_{i-1}W_{i-1}\cdots A_1W_1x,$$

and the network outputs

$$f(x;w) = W_LA_{L-1}W_{L-1}A_{L-2}\cdots A_1W_1x.$$

and the gradient with respect to layer i is

$$\frac{\mathrm{d}}{\mathrm{d}W_i}f(x;w) = (W_LA_{L-1}\cdots W_{i+1}A_i)^\intercal(A_{i-1}W_{i-1}\cdots W_1x)^\intercal.$$

Additionally

$$\begin{aligned}\left\langle W_i, \frac{\mathrm{d}}{\mathrm{d}W_i}f(x;w)\right\rangle &= \langle W_i, (W_LA_{L-1}\cdots W_{i+1}A_i)^\intercal(A_{i-1}W_{i-1}\cdots W_1x)^\intercal\rangle \\ &= \operatorname{tr}\left(W_i^\intercal(W_LA_{L-1}\cdots W_{i+1}A_i)^\intercal(A_{i-1}W_{i-1}\cdots W_1x)^\intercal\right) \\ &= \operatorname{tr}\left((W_LA_{L-1}\cdots W_{i+1}A_i)^\intercal(W_iA_{i-1}W_{i-1}\cdots W_1x)^\intercal\right) \\ &= \operatorname{tr}\left((W_iA_{i-1}W_{i-1}\cdots W_1x)^\intercal(W_LA_{L-1}\cdots W_{i+1}A_i)^\intercal\right) \\ &= \operatorname{tr}\left(W_LA_{L-1}\cdots W_{i+1}A_iW_iA_{i-1}W_{i-1}\cdots W_1x\right) \\ &= f(x;w),\end{aligned}$$

and

$$\left\langle W_i, \frac{\mathrm{d}}{\mathrm{d}W_i}f(x;w)\right\rangle = f(x;w) = \left\langle W_{i+1}, \frac{\mathrm{d}}{\mathrm{d}W_{i+1}}f(x;w)\right\rangle.$$

This calculation can in fact be made much more general (indeed with a simpler proof!).

Lemma 9.2 Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is locally Lipschitz and L-positively homogeneous. For any $w \in \mathbb{R}^d$ and $s \in \partial f(w)$,

$$\langle s, w\rangle = Lf(w).$$

Remark 9.4 This statement appears in various places (Lyu and Li 2019); the version here is somewhat more general, and appears in (Ji and Telgarsky 2020).

Proof. If $w = 0$, then $\langle s, w\rangle = 0 = Lf(w)$ for every $s \in \partial f(w)$, so consider the case $w \neq 0$. Let D denote those w where f is differentiable, and consider the case that $w \in D \setminus \{0\}$. By the definition of gradient,

$$\lim_{\delta\downarrow 0}\frac{f(w+\delta w) - f(w) - \langle\nabla f(w), \delta w\rangle}{\delta\|w\|} = 0,$$

and by using homogeneity in the form $f(w+\delta w) = (1+\delta)^L f(w)$ (for any $\delta > 0$), then

$$0 = \lim_{\delta \downarrow 0} \frac{\left((1+\delta)^L - 1\right) f(w) - \langle \nabla f(w), \delta w \rangle}{\delta} = -\langle \nabla f(w), w \rangle + \lim_{\delta \downarrow 0} f(w) \left(L + \mathcal{O}(\delta)\right),$$

which implies $\langle w, \nabla f(w) \rangle = L f(w)$.
Now consider $w \in \mathbb{R}^d \setminus D \setminus \{0\}$. For any sequence $(w_i)_{i \geq 1}$ in D with $\lim_i w_i = w$ for which there exists a limit $s := \lim_i \nabla f(w_i)$, then

$$\langle w, s \rangle = \lim_{i \to \infty} \langle w_i, \nabla f(w_i) \rangle = \lim_{i \to \infty} L f(w_i) = L f(w).$$

Lastly, for any element $s \in \partial f(w)$ written in the form $s = \sum_i \alpha_i s_i$ where $\alpha_i \geq 0$ satisfy $\sum_i \alpha_i = 1$ and each s_i is a limit of a sequence of gradients as above, then

$$\langle w, s \rangle = \left\langle w, \sum_i \alpha_i s_i \right\rangle = \sum_i \alpha_i \langle w, s_i \rangle = \sum_i \alpha_i L f(w) = L f(w).$$

9.3 Norm preservation

If predictions are positive homogeneous with respect to each layer, then gradient flow preserves norms of layers.

Lemma 9.3 ***(Simon S. Du, Hu, and Lee (2018))*** Suppose for $\alpha > 0$, $f(x; (W_L, \ldots, \alpha W_i, \ldots, W_1)) = \alpha f(x; w)$ (predictions are 1-homogeneous in each layer). Then for every pair of layers (i, j), the gradient flow maintains

$$\frac{1}{2}\|W_i(t)\|^2 - \frac{1}{2}\|W_i(0)\|^2 = \frac{1}{2}\|W_j(t)\|^2 - \frac{1}{2}\|W_j(0)\|^2.$$

Remark 9.5 We'll assume a risk of the form $\mathbb{E}_k \, \ell(y_k f(x_k; w))$, but it holds more generally. We are also tacitly assuming we can invoke the chain rule, as discussed above.

Proof. Defining $\ell_k'(s) := y_k \ell'(y_k f(x_k; w(s)))$, and fixing a layer i,

$$\begin{aligned}
\frac{1}{2}\|W_i(t)\|^2 - \frac{1}{2}\|W_i(0)\|^2 &= \int_0^t \frac{\mathrm{d}}{\mathrm{d}t} \frac{1}{2} \|W_i(s)\|^2 \mathrm{d}s \\
&= \int_0^t \left\langle W_i(s), \dot{W}_i(s) \right\rangle \mathrm{d}s \\
&= \int_0^t \left\langle W_i(s), -\mathop{\mathbb{E}}_k \ell_k'(s) \frac{\mathrm{d} f(x_k; w)}{\mathrm{d} W_i(s)} \right\rangle \mathrm{d}s \\
&= -\int_0^t \mathop{\mathbb{E}}_k \ell_k'(s) \left\langle W_i(s), \frac{\mathrm{d} f(x_k; w)}{\mathrm{d} W_i(s)} \right\rangle \mathrm{d}s \\
&= -\int_0^t \mathop{\mathbb{E}}_k \ell_k'(s) f(x_k; w) \mathrm{d}s.
\end{aligned}$$

This final expression does not depend on i, which gives the desired equality.

Remark 9.6 One interesting application is to classification losses like $\exp(-z)$ and $\ln(1 + \exp(-z))$, where $\widehat{\mathcal{R}}(w) \to 0$ implies $\min_k y_k f(x_k; w) \to \infty$.

This by itself implies $\|W_j\| \to \infty$ for *some* j; combined with norm preservation, $\min_j \|W_j\| \to \infty$!

[mjt☹: need to update this in light of the new material i've included?]

9.4 Smoothness inequality adapted to ReLU

Let's consider: single hidden ReLU layer, only bottom trainable:

$$f(x;w) := \frac{1}{\sqrt{m}} \sum_j a_j \sigma(\langle x, w_j \rangle), \qquad a_j \in \{\pm 1\}.$$

Let $W_s \in \mathbb{R}^{m \times d}$ denote parameters at time s, suppose $\|x\| \leq 1$.

$$\frac{\mathrm{d}f(x;W)}{\mathrm{d}W} = \begin{bmatrix} a_1 x \sigma'(w_1^{\mathsf{T}} x)/\sqrt{m} \\ \vdots \\ a_m x \sigma'(w_m^{\mathsf{T}} x)/\sqrt{m} \end{bmatrix},$$

$$\left\| \frac{\mathrm{d}f(x;W)}{\mathrm{d}W} \right\|_{\mathrm{F}}^2 = \sum_j \left\| a_j x \sigma'(w_j^{\mathsf{T}} x)/\sqrt{m} \right\|_2^2 \leq \frac{1}{m} \sum_j \|x\|_2^2 \leq 1.$$

We'll use the logistic loss, whereby

$$\begin{aligned} \ell(z) &= \ln(1 + \exp(-z)), \\ \ell'(z) &= \frac{-\exp(-z)}{1 + \exp(-z)} \in (-1, 0), \\ \widehat{\mathcal{R}}(W) &:= \frac{1}{n} \sum_k \ell(y_k f(x_k; W)). \end{aligned}$$

A key fact (can be verified with derivatives) is

$$|\ell'(z)| = -\ell'(z) \leq \ell(z),$$

whereby

$$\begin{aligned} \frac{\mathrm{d}\widehat{\mathcal{R}}}{\mathrm{d}W} &= \frac{1}{n} \sum_k \ell'(y_k f(x_k; W)) y_k \nabla_W f(x_k W), \\ \left\| \frac{\mathrm{d}\widehat{\mathcal{R}}}{\mathrm{d}W} \right\|_{\mathrm{F}} &\leq \frac{1}{n} \sum_k |\ell'(y_k f(x_k; W))| \cdot \|y_k \nabla_W f(x_k W)\|_{\mathrm{F}} \\ &\leq \frac{1}{n} \sum_k |\ell'(y_k f(x_k; W))| \leq \min\left\{1, \widehat{\mathcal{R}}(W)\right\}. \end{aligned}$$

Now we can state a non-smooth, non-convex analog to +Theorem 7.3

Lemma 9.4 ***((Lemma 2.6, Ji and Telgarsky 2019a))*** If $\eta \leq 1$, for any Z,

$$\|W_t - Z\|_{\mathrm{F}}^2 + \eta \sum_{i<t} \widehat{\mathcal{R}}^{(i)}(W_i) \leq \|W_0 - Z\|_{\mathrm{F}}^2 + 2\eta \sum_{i<t} \widehat{\mathcal{R}}^{(i)}(Z),$$

where $\widehat{\mathcal{R}}^{(i)}(W) = \frac{1}{n} \sum_k \ell(y_k \langle W, \nabla f(x_k; W_i) \rangle)$.

Remark 9.7

- $\widehat{\mathcal{R}}^{(i)}(W_i) = \widehat{\mathcal{R}}(W_i)$.
- $\widehat{\mathcal{R}}^{(i)}(Z) \approx \widehat{\mathcal{R}}(Z)$ if W_i and Z have similar activations.
- (Ji and Telgarsky 2019a) uses this in a proof scheme like (Chizat and Bach 2019): consider those iterations where the activations are similar, and then prove it actually happens a lot. (Ji and Telgarsky 2019a), with additional work, can use this to prove low *test* error.

Proof. Using the squared distance potential as usual,

$$\|W_{i+1} - Z\|_{\mathrm{F}}^2 = \|W_i - Z\|_{\mathrm{F}}^2 - 2\eta \left\langle \nabla\widehat{\mathcal{R}}(W_i), W_i - Z \right\rangle + \eta^2 \|\nabla\widehat{\mathcal{R}}(W_i)\|_{\mathrm{F}}^2,$$

where $\|\nabla\widehat{\mathcal{R}}(W_i)\|_{\mathrm{F}}^2 \leq \|\nabla\widehat{\mathcal{R}}(W_i)\|_{\mathrm{F}} \leq \widehat{\mathcal{R}}(W_i) = \widehat{\mathcal{R}}^{(i)}(W_i)$, and

$$\begin{aligned}
&n \left\langle \nabla\widehat{\mathcal{R}}(W_i), Z - W_i \right\rangle \\
&= \sum_k y_k \ell'(y_k f(x_k; W_i)) \left\langle \nabla_W f(x_k; W_i), Z - W_i \right\rangle \\
&= \sum_k \ell'(y_k f(x_k; W_i)) \left(y_k \left\langle \nabla_W f(x_k; W_i), Z \right\rangle - y_k f(x_k; W_i) \right) \\
&\leq \sum_k \left(\ell(y_k \left\langle \nabla_W f(x_k; W_i), Z \right\rangle) - \ell(y_k f(x_k; W_i)) \right) \\
&= n \left(\widehat{\mathcal{R}}^{(i)}(Z) - \widehat{\mathcal{R}}^{(i)}(W_i) \right).
\end{aligned}$$

Together,

$$\|W_{i+1} - Z\|_{\mathrm{F}}^2 \leq \|W_i - Z\|_{\mathrm{F}}^2 + 2\eta \left(\widehat{\mathcal{R}}^{(i)}(Z) - \widehat{\mathcal{R}}^{(i)}(W_i) \right) + \eta \widehat{\mathcal{R}}_i(W_i);$$

applying $\sum_{i<t}$ to both sides gives the bound.

10 Margin maximization and implicit bias

During 2015-2016, various works pointed out that deep networks generalize well, even though parameter norms are large, and there is no explicit generalization (Neyshabur, Tomioka, and Srebro 2014; Zhang et al. 2017). This prompted authors to study *implicit bias of gradient descent*, the first such result being an analysis of linear predictors with *linearly separable data*, showing that gradient descent on the cross-entropy loss is implicitly biased towards a *maximum margin direction* (Soudry, Hoffer, and Srebro 2017).

This in turn inspired many other works, handling other types of data, networks, and losses (Ji and Telgarsky 2019b, 2018, 2020; Gunasekar et al. 2018a; Lyu and Li 2019; Chizat and Bach 2020; Ji et al. 2020).

Margin maximization of first-order methods applied to exponentially-tailed losses was first proved for coordinate descent (Telgarsky 2013). The basic proof scheme there was pretty straightforward, and based on the similarity of the empirical risk (after the monotone transformation $\ln(\cdot)$) to $\ln \sum \exp$, itself similar to $\max(\cdot)$ and thus to margin maximization; we will use this connection as a basis for all proofs in this section (see also (Ji and Telgarsky 2019b; Gunasekar et al. 2018b)).

Throughout this section, fix training data $((x_i, y_i))_{i=1}^n$, define a (an unnormalized) *margin mapping*

$$m_i(w) := y_i f(x_i; w);$$

by this choice, we can also conveniently write an **unnormalized risk** $\mathcal{L}$:

$$\mathcal{L}(w) := \sum_i \ell(m_i(w)) = \sum_i \ell(y_i f(x_i; w)).$$

Throughout this section, we will always assume f is locally-Lipschitz and L-homogeneous in w, which also means each m_i is locally-Lipschitz and L-homogeneous.

We will also use the exponential loss $\ell(z) = \exp(-z)$. The results go through for similar losses.

Remark 10.1 ***(generalization)*** As hinted before, margin maximization is one way gradient descent prefers a solution which has a hope to generalize well, and not merely achieve low empirical risk. This low generalization error of large-margin predictors will appear explicitly later on in section 13.4.

Remark 10.2 ***(implicit bias)*** As mentioned above, the proofs here will show *implicit margin maximization*, which is enough to invoke the generalization theory in section 13.4. However, in certain cases it is valuable to moreover prove converges rates to the *maximum margin direction*. In the linear case, is is possible to convert a margin maximization rate to an implicit bias rate, however the rate degrades by a factor $\sqrt{\cdot}$ (Ji and Telgarsky 2019b); analyzing the implicit bias without degradation in the rate is more involved, and not treated here (Soudry, Hoffer, and Srebro 2017).

Remark 10.3 ***(squared loss)*** While the focus here is on losses with exponential tails and on bias towards the maximum margin direction, there are also many works (not further discussed here) which consider the squared loss (Gunasekar et al. 2017; Arora, Cohen, et al. 2018b, 2019).

10.1 Separability and margin maximization

We just said "maximum margin" and "separable data." What do these mean?

Consider a linear predictor, meaning $x \mapsto \langle w, x\rangle$ for some $w \in \mathbb{R}^d$. This w "separates the data" if y_i and $\operatorname{sgn}(\langle w, x_i\rangle)$ agree, which we can relax to the condition of *strict separability*, namely

$$\min_i y_i \langle w, x_i\rangle > 0.$$

It seems reasonable, or a nice *inductive bias*, if we are as far from 0 as possible:

$$\max_{w \in ?} \min_i y_i \langle w, x_i\rangle > 0$$

The "?" indicates that we must somehow normalize or constrain, since otherwise, for separable data, this max becomes a sup and has value $+\infty$.

Definition 10.1 Data is *linearly separable* when there exists $w \in \mathbb{R}^d$ so that $\min_i y_i \langle w, x_i\rangle > 0$. In this situation, the (ℓ_2) *maximum margin predictor* (which is unique!) is given by

$$\bar{u} := \arg\max_{\|w\|=1} \min_i y_i \langle w, x_i\rangle,$$

and the margin is $\gamma := \min_i y_i \langle \bar{u}, x_i\rangle$.

Remark 10.4 This concept has a long history. Margins first appeared in the classical perceptron analysis (Novikoff 1962), and maximum margin predictors were a guiding motivation for the SVM [mjt☹: need to add many more refs].

Consider now the general case of L-homogeneous predictors, where $y_i \langle w, x_i \rangle$ is replaced by $m_i(w)$.

Proposition 10.1 Suppose $f(x; w)$ is L-homogeneous in w, ℓ is the exponential loss, and there exists $\hat{w}$ with $\widehat{\mathcal{R}}(\hat{w}) < \ell(0)/n$. Then $\inf_w \widehat{\mathcal{R}}(w) = 0$, and the infimum is not attained.

Proof. Note

$$\max_i \ell(-m_i(\hat{w})) \leq \sum_i \ell(-m_i(\hat{w})) = n\widehat{\mathcal{R}}(\hat{w}) < \ell(0),$$

thus applying ℓ^{-1} to both sides gives $\min_i m_i(\hat{w})) > 0$. Therefore

$$0 \leq \inf_w \widehat{\mathcal{R}}(w) \leq \limsup_{c\to\infty} \widehat{\mathcal{R}}(c\hat{w}) = \sum_i \limsup_{c\to\infty} \ell(-m_i(c\hat{w})) = \sum_i \limsup_{c\to\infty} \ell(-c^L m_i(\hat{w})) = 0.$$

This seems to be problematic; how can we "find" an "optimum," when solutions are off at infinity? Moreover, we do not even have unique directions, nor a way to tell different ones apart!

We can use margins, now appropriately generalized to the L-homogeneous case, to build towards a better-behaved objective function. First note that since

$$\min_i m_i(w) = \|w\|^L \min_i m_i\left(\frac{w}{\|w\|}\right),$$

we can compare different directions by normalizing the margin by $\|w\|^L$. Moreover, again using the exponential loss,

$$\begin{aligned} \frac{\ell^{-1}\left(\mathcal{L}(w)\right)}{\|w\|^L} + \frac{\ln(n)}{\|w\|^L} = \frac{\ell^{-1}\left(\sum_i \ell(m_i(w))/n\right)}{\|w\|^L} &\geq \frac{\min_i m_i(w)}{\|w\|^L} \\ &= \frac{\ell^{-1}\left(\max_i \ell(m_i(w))\right)}{\|w\|^L} \\ &\geq \frac{\ell^{-1}\left(\mathcal{L}(w)\right)}{\|w\|^L}. \end{aligned} \tag{5}$$

This motivates the following definition.

Definition 10.2 Say the data is $\vec{m}$-separable when there exists w so that $\min_i m_i(w) > 0$. Define the margin, maximum margin, and smooth margin respectively as

$$\gamma(w) := \min_i m_i(w/\|w\|) = \frac{\min_i m_i(w)}{\|w\|^L}, \qquad \bar{\gamma} := \max_{\|w\|=1} \gamma(w), \qquad \tilde{\gamma}(w) := \frac{\ell^{-1}(\mathcal{L}(w))}{\|w\|^L}. \tag{6}$$

[mjt☹: decide something about $w = 0 \ldots$]

Remark 10.5 The terminology "smoothed margin" is natural for L-homogeneous predictors, but even so it seems to have only appeared recently in (Lyu and Li 2019). In the 1-homogeneous case, the smoothed margin appeared much earlier, indeed throughout the boosting literature (Schapire and Freund 2012).

Remark 10.6 ***(multiclass margins)*** There is also a natural notion of multiclass margin:

$$\min_i \frac{f(x_i; w)_{y_i} - \max_{j\neq y_i} f(x_i; w)_j}{\|w\|^L}.$$

The natural loss to consider in this setting is the cross-entropy loss.

The basic properties can be summarized as follows.

Proposition 10.2 Suppose data is $\vec{m}$-separable. Then:

- $\bar{\gamma} := \max_{\|w\|\le 1} \gamma(w) > 0$ is well-defined (the maximum is attained).
- For any $w \neq 0$, For any $\hat{w}$ with $\bar{\gamma} = \gamma(\hat{w})$,

$$\lim_{c\to\infty} \tilde{\gamma}(cw) = \gamma(w).$$

In particular, for $\hat{w}$ satisfying $\bar{\gamma} = \gamma(\hat{w})$, then $\lim_{c\to\infty} \tilde{\gamma}(c\hat{w}) = \bar{\gamma}$.

Proof. The first part follows by continuity of $m_i(w)$ and compactness of $\{w \in \mathbb{R}^p : \|w\| = 1\}$, and the second from eq. 6 and eq. 5.

Remark 10.7 For the linear case, margins have a nice geometric interpretation. This is not currently true for the general homogeneous case: there is no known reasonable geometric characterization of large margin predictors even for simple settings.

10.2 Gradient flow maximizes margins of linear predictors

Let's first see how far we can get in the linear case, using one of our earlier convex optimization tools, namely Theorem 7.4.

Lemma 10.1 Consider the linear case, with linearly separable data and the exponential loss, and $\max_i \|x_i y_i\| \le 1$. Then

$$\mathcal{L}(w_t) \le \frac{1 + \ln(2nt\gamma^2)}{2t\gamma^2},$$
$$\|w_t\| \ge \ln(2tn\gamma^2) - \ln\left(1 + \ln(2tn\gamma^2)\right).$$

Remark 10.8 The intuition we will follow for the proof is: *for every unit of norm, the (unnormalized) margin increases by at least γ.* Thus the margin bias affects the entire gradient descent process.

Later, when we study the L-homogeneous case, we are only able to show *for every unit norm (to the power L), the (unnormalized) margin increases by at least the current margin*, which implies nondecreasing, but not margin maximization.

Proof. By Theorem 7.4 with $z = \ln(c)\bar{u}/\gamma$ for some $c > 0$,

$$\mathcal{L}(w(t)) \le \mathcal{L}(z) + \frac{1}{2t}\left(\|z\|^2 - \|w(t) - z\|^2\right) \le \sum_i \ell(m_i(z)) + \frac{\|z\|^2}{2t}$$
$$\le \sum_i \exp(-\ln(c)) + \frac{\ln(c)^2}{2t\gamma^2} = \frac{n}{c} + \frac{\ln(c)^2}{2t\gamma^2},$$

and the first inequality follows from the choice $c := 2tn\gamma^2$. For the lower bound on $\|w_t\|$, using

the preceding inequality,

$$\ell\left(\|w_t\|\right) \leq \min_i \ell(m_i(w_t)) \leq \frac{1}{n}\mathcal{L}(w_t) \leq \frac{1+\ln(2tn\gamma^2)^2}{2tn\gamma^2},$$

and the second inequality follows by applying ℓ^{-1} to both sides.

This nicely shows that we decrease the risk to 0, but not that we maximize margins. For this, we need a more specialized analysis.

Theorem 10.1 Consider the linear case, with linearly separable data and the exponential loss, and $\max_i \|x_i y_i\| \leq 1$. Then

$$\gamma(w_t) \geq \tilde{\gamma}(w_t) \geq \bar{\gamma} - \frac{\ln n}{\ln t + \ln(2n\gamma^2) - 2\ln\ln(2tne\gamma^2)}$$

[mjt☹: need to check some constants. also that denominator is hideous, maybe require slightly larger t to remove it?]

Proof. For convenience, define $u(t) := \ell^{-1}(\mathcal{L}(w(t)))$ and $v(t) := \|w(t)\|$, whereby

$$\gamma(w(t)) = \frac{u(t)}{v(t)} = \frac{u(0)}{v(t)} + \frac{\int_0^t \dot{u}(s)\mathrm{d}s}{v(t)}.$$

Let's start by lower bounding the second term. Since $\ell' = -\ell$,

$$\begin{aligned}
\dot{u}(t) &= \left\langle \frac{-\nabla\mathcal{L}(w(t))}{\mathcal{L}(w(t))}, \dot{w}(t) \right\rangle = \frac{\|\dot{w}(t)\|^2}{\mathcal{L}(w(t))}, \\
\|\dot{w}(s)\| &\geq \langle \dot{w}(s), \bar{u}\rangle = \left\langle -\sum_i x_i y_i \ell'(m_i(w(s))), \bar{u} \right\rangle \\
&= \sum_i \ell(m_i(w(s)))\, \langle x_i y_i, \bar{u}\rangle \geq \gamma \sum_i \ell(m_i(w(s))) = \gamma\mathcal{L}(w(s)), \\
v(t) &= \|w(t) - w(0)\| = \left\| \int_0^t \dot{w}(s)\mathrm{d}s \right\| \leq \int_0^t \|\dot{w}(s)\|\,\mathrm{d}s,
\end{aligned}$$

thus

$$\frac{\int_0^t \dot{u}(s)\mathrm{d}s}{v(t)} \geq \frac{\int_0^t \frac{\|\dot{w}(s)\|^2}{\mathcal{L}(w(s))}\mathrm{d}s}{v(t)} = \frac{\int_0^t \|\dot{w}(s)\|\frac{\|\dot{w}(s)\|}{\mathcal{L}(w(s))}\mathrm{d}s}{v(t)} \geq \frac{\gamma\int_0^t \|\dot{w}(s)\|\mathrm{d}s}{v(t)} = \gamma.$$

For the first term $u(0)/v(t)$, note $\mathcal{L}(w(0)) = n$ and thus $u(0) = -\ln n$, whereas by the lower bound on $\|w(t)\|$ from Lemma 10.1,

$$\frac{u(0)}{v(t)} = \frac{-\ln(n)}{\|w(t)\|} \geq \frac{-\ln(n)}{\ln(t) + \ln(2n\gamma^2) - 2\ln\ln(2tne\gamma^2)}.$$

Combining these inequalities gives the bound.

We are maximizing margins, but at a glacial rate of $1/\ln(t)$!

To get some inspiration, notice that we keep running into $\ell^{-1}(\mathcal{L}(w))$ in all the analysis. Why don't we just run gradient flow on this modified objective? In fact, the two gradient flows are the same!

Remark 10.9 ***(time rescaling)*** Let $w(t)$ be given by gradient flow on $\mathcal{L}(w(t))$, and define a time rescaling $h(t)$ via integration, namely so that $\dot{h}(t) = 1/\mathcal{L}(w(h(t)))$. Then, by the substitution rule for integration,

$$\begin{aligned} w(t) - w(0) &= \int_0^t \dot{w}(s)\mathrm{d}s = -\int_0^t \nabla\mathcal{L}(w(s))\mathrm{d}s = \int_{h^{-1}([0,t])} \nabla\mathcal{L}(w(h(s)))|\dot{h}(s)|\mathrm{d}s \\ &= \int_{h^{-1}([0,t])} \frac{\nabla\mathcal{L}(w(h(s)))}{\mathcal{L}(w(h(s)))}\mathrm{d}s = \int_{h^{-1}([0,t])} \nabla \ln \mathcal{L}(w(h(s))\mathrm{d}s \end{aligned}$$

As such, the gradient flow on $\mathcal{L}$ and on $\ell^{-1} \circ \mathcal{L}$ are the same, modulo a *time rescaling*. This perspective was first explicitly stated by Chizat and Bach (2020), though *analyses* using this rescaled time (and alternate flow characterization) existed before Lyu and Li (2019).

Theorem 10.2 ***(time-rescaled flow)*** Consider linear predictors with linearly separable data, and the logistic loss. Suppose $\dot{\theta}(t) := \nabla_\theta \ell^{-1}\mathcal{L}(\theta(t))$. Then

$$\gamma(\theta(t) \geq \tilde{\gamma}(\theta(t)) \geq \gamma - \frac{\ln n}{t\gamma^2 - \ln n}.$$

Proof. We start as before: set $u(t) := \ell^{-1}\mathcal{L}(\theta(t))$ and $v(t) := \|\theta(t)\|$; then

$$\tilde{\gamma}(t) = \frac{u(t)}{v(t)} = \frac{u(0)}{v(t)} + \frac{\int_0^t \dot{u}(s)\mathrm{d}s}{v(t)} = \frac{-\ln n}{v(t)} + \frac{\int_0^t \dot{u}(s)\mathrm{d}s}{v(t)}.$$

Bounding these terms is now much simpler than for the regular gradient flow. Note

$$\begin{aligned} \|\dot{\theta}(s)\| &\geq \langle \nabla \ln \mathcal{L}(\theta(s)), \bar{u}\rangle = \sum_i \frac{\ell'(m_i(\theta(s)))}{\mathcal{L}(\theta(s))} \langle x_i y_i, \bar{u}\rangle \geq \gamma \sum_i \frac{\ell'(m_i(\theta(s)))}{\mathcal{L}(\theta(s))} = \gamma, \\ \dot{u}(s) &= \left\langle \nabla \ln \mathcal{L}(\theta(s)), \dot{\theta}(s)\right\rangle = \|\dot{\theta}(s)\|^2, \end{aligned}$$

thus

$$\begin{aligned} \ell^{-1}\mathcal{L}(\theta(t)) &= \ell^{-1}\mathcal{L}(\theta(0)) + \int_0^t \frac{\mathrm{d}}{\mathrm{d}s}\ell^{-1}\mathcal{L}(\theta(s))\mathrm{d}s \geq -\ln(n) + t\gamma^2, \\ \frac{\int_0^t \dot{u}(s)\mathrm{d}s}{v(t)} &= \frac{\int_0^t \|\dot{\theta}(s)\|^2\mathrm{d}s}{v(t)} \geq \frac{\gamma\int_0^t \|\dot{\theta}(s)\|\mathrm{d}s}{v(t)} \geq \frac{\gamma\|\int_0^t \dot{\theta}(s)\mathrm{d}s\|}{v(t)} = \gamma. \end{aligned}$$

On the other hand,

$$\|\theta(t)\|\gamma \geq \|\theta(t)\|\gamma(\theta(t)) \geq \ell^{-1}\mathcal{L}(\theta(t)) \geq t\gamma^2 - \ln(n).$$

Together,

$$\gamma(t) = \frac{u(t)}{v(t)} \geq \gamma - \frac{\ln(n)}{t\gamma^2 - \ln n}.$$

Remark 10.10 The preceding two proofs are simplified from (Ji and Telgarsky 2019b), but follow a general scheme from the (coordinate descent!) analysis in (Telgarsky 2013); this scheme was also followed in (Gunasekar et al. 2018b). The proof in (Soudry, Hoffer, and Srebro 2017) is different, and is based on an SVM analogy, since $\tilde{\gamma} \to \gamma$.

Note also that the proofs here do not show $w(t)$ converges to (the unique) maximum margin linear separator, which is easy to do with worse rates, and harder to do with good rates. However, large margins is sufficient for generalization in the linear case, as in section 13.4.

10.3 Smoothed margins are nondecreasing for homogeneous functions

In the nonlinear case, we do not have a general result, and instead only prove that smoothed margins are nondecreasing.

Theorem 10.3 ***(originally from (Lyu and Li 2019), simplification due to (Ji 2020))*** Consider the *Clarke flow* $\dot{w}_t \in -\partial \ln \sum_i \exp(-m_i(w_t))$ with $w_0 = 0$, and once again suppose the chain rule holds for almost all $t \geq 0$. If there exists t_0 with $\tilde{\gamma}(w(t_0)) > 0$, then $t \mapsto \tilde{\gamma}(w(t))$ is nondecreasingalong $[t_0, \infty)$.

The proof will use the following interesting approximate homogeneity property of $\ln \sum \exp$.

Lemma 10.2 ***(taken from (Ji and Telgarsky 2020))*** For every w and every $v \in -\partial \ln \sum \exp(-m_i(w))$, assuming the chain rule holds,

$$-L \ln \sum_i \exp(-m_i(w)) \leq \langle v, w \rangle .$$

If $-\ln \sum_i \exp$ were itself homogeneous, this would be an equality; instead, using only the L-homegeneity of m_i, we get a lower bound.

Proof. Let $v \in -\partial \ln \sum \exp(m_i(w))$ be given, whereby (thanks to assuming a chain rule) there exists $v_i \in \partial m_i(w)$ for each i such that

$$v = \sum_{i=1}^{n} \frac{\exp(-m_i(w)) v_i}{\sum_{j=1}^{m} \exp(-m_j(w))}.$$

Since $\exp(-m_k(w)) \geq 0$ for every k and since $-\ln$ is monotone decreasing, Then

$$\begin{aligned}
\langle v, w \rangle &= \left\langle \sum_{i=1}^{n} \frac{\exp(-m_i(w)) v_i}{\sum_{j=1}^{m} \exp(-m_j(w))}, w \right\rangle \\
&= \sum_{i=1}^{n} \frac{\exp(-m_i(w))}{\sum_{j=1}^{m} \exp(-m_j(w))} \langle v_i, w \rangle \\
&= L \sum_{i=1}^{n} \frac{\exp(-m_i(w))}{\sum_{j=1}^{m} \exp(-m_j(w))} m_i(w) \\
&= L \sum_{i=1}^{n} \frac{\exp(-m_i(w))}{\sum_{j=1}^{m} \exp(-m_j(w))} (-\ln \exp(-m_i(w))) \\
&\geq L \sum_{i=1}^{n} \frac{\exp(-m_i(w))}{\sum_{j=1}^{m} \exp(-m_j(w))} \left(-\ln \sum_k \exp(-m_k(w)) \right) \\
&= -L \ln \sum_k \exp(-m_k(w))
\end{aligned}$$

as desired.

Lemma 10.2 (taken from (Ji and Telgarsky 2020)) leads to a fairly easy proof of Theorem 10.3 (originally from (Lyu and Li 2019), simplification due to (Ji 2020)).

Proof of Theorem 10.3 (originally from (Lyu and Li 2019), simplification due to (Ji 2020)). It will be shown that $(\mathrm{d}/\mathrm{d}t)\tilde{\gamma}(t) \geq 0$ whenever $\tilde{\gamma}(t) > 0$, which completes the proof via

the following contradiction. Let $t > t_0$ denote the earliest time where $\tilde{\gamma}(t) < \tilde{\gamma}(t_0)$. But that means $\tilde{\gamma}(t') > 0$ for $t' \in [t_0, t)$, whereby

$$\tilde{\gamma}(t) = \tilde{\gamma}(t_0) + \int_{t_0}^{t} \frac{\mathrm{d}}{\mathrm{d}s}\tilde{\gamma}(s)\mathrm{d}s \geq \tilde{\gamma}(t_0) + 0,$$

a contradiction, thus completing the proof.
To this end, fix any t with $\tilde{\gamma}(t) > 0$, and the goal is to show $(\mathrm{d}/\mathrm{d}t)\tilde{\gamma}(t) \geq 0$. Define

$$u(t) := -\ln \sum_i \exp(-m_i(w(t))), \qquad v(t) := \|w(t)\|^L,$$

whereby $\tilde{\gamma}(t) := \tilde{\gamma}(w(t)) := u(t)/v(t)$, and

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\gamma}(t) = \frac{\dot{u}(t)v(t) - u(t)\dot{v}(t)}{v(t)^2},$$

where $v(t) > 0$ since $\tilde{\gamma}(t) > 0$ is impossible otherwise, which means the ratio is well-defined. Making use of Lemma 10.2 (taken from (Ji and Telgarsky 2020)),

$$\begin{aligned}
\dot{u}(t) &= \|\dot{w}(t)\|^2 \\
&\geq \|\dot{w}(t)\| \left\langle \frac{w(t)}{\|w(t)\|}, \dot{w}(t) \right\rangle \\
&\geq \frac{Lu(t)\|\dot{w}(t)\|}{\|w(t)\|}, \\
\dot{v}(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\|w(t)\|^{L/2} \\
&= L\|w(t)\|^{L-1} \left\langle \frac{w(t)}{\|w(t)\|}, \dot{w}(t) \right\rangle \\
&\leq L\|w(t)\|^{L-1}\|\dot{w}(t)\|,
\end{aligned}$$

whereby

$$\dot{u}(t)v(t) - \dot{v}(t)u(t) \geq \frac{Lu(t)\|\dot{w}(t)\|}{\|w(t)\|}v(t) - u(t)L\|w(t)\|^{L-1}\|\dot{w}(t)\| = 0,$$

which completes the proof.

Remark 10.11 The linear case achieves a better bound by having not only a unique global hard margin solution u, but having the structural property $\langle u, y_i x_i \rangle \geq \gamma$, which for instance implies $\|\dot{w}(t)\| \geq \gamma$.
Instead, the preceding proof uses the much weaker inequality $\|\dot{w}(t)\| \geq \frac{Lu(t)}{\|w(t)\|}$.

Remark 10.12 As mentioned, +Theorem 10.3 (originally from (Lyu and Li 2019), simplification due to (Ji 2020)) was originally presented in (Lyu and Li 2019), though this simplification is due to (Ji 2020), and its elements can be found throughout (Ji and Telgarsky 2020). The version in (Lyu and Li 2019) is significantly different, and makes heavy (and interesting) use of a *polar decomposition* of homogeneous functions and gradient flow on them.
For the case of an infinite-width 2-homogeneous network, assuming a number of convergence properties of the flow (which look technical, but are not "merely technical," and indeed difficult to prove), margins are globally maximized (Chizat and Bach 2020).

11 Generalization: preface

The purpose of this generalization part is to bound the gap between testing and training error for standard (multilayer ReLU) deep networks via the classical uniform convergence tools, and also to present and develop these classical tools (based on Rademacher complexity).

These bounds are very loose, and there is extensive criticism now both of them and of the general approach, as will be discussed shortly (Neyshabur, Tomioka, and Srebro 2014; Zhang et al. 2017; Nagarajan and Kolter 2019; Dziugaite and Roy 2017); this work is ongoing and moving quickly and there are even already many responses to these criticisms (Negrea, Dziugaite, and Roy 2019; L. Zhou, Sutherland, and Srebro 2020; P. L. Bartlett and Long 2020).

11.1 Omitted topics

- Domain adaptation / covariate shift.
- Generalization properties of more architectures. One key omission is of convolution layers; for one generalization analysis, see (Long and Sedghi 2019).
- Other approaches and perspectives on generalization (possibly changing the basic definitions of "generalization"), for instance:
  - PAC-Bayes approaches (Dziugaite and Roy 2017). In the present notes, we only focus on *uniform convergence bounds*, which give high probability bounds between training and test error which hold simultaneously for every element of some class.

    By contrast, PAC-Bayes consider a *distribution* over predictors, and bound the *expected* gap between testing and training error for these predictors in terms of how close this distribution is to some prior distribution over the predictors.
  - The looseness of the uniform-convergence bounds presented in these notes leads many authors to instead use them as *explanatory* tools, e.g., by studying their *correlation* with observed generalization. A correlation was claimed and presented in (P. Bartlett, Foster, and Telgarsky 2017), however it was on a single dataset and architecture. More extensive investigations have appeared recently (Jiang et al. 2020; Dziugaite et al. 2020), and highlight that while some bounds are correlated with generalization (or rather *predictive of generalization*) in some settings, there are other situations (e.g., large width) where no bound is correlated with observed generalization gaps.
  - Compression-based approaches (Arora, Ge, et al. 2018), which bound the generalization of the network *after* applying some compression procedure, with no guarantees on the original network; that said, it is a promising approach, and there has been some effort to recover guarantees on the original network (Suzuki, Abe, and Nishimura 2019).

    Another relevant work, from an explicitly PAC-Bayes perspective, is (W. Zhou et al. 2018). For further connections between PAC-Bayes methodology and compression, see (Blum and Langford 2003), and for more on the concept of compression *schemes*, see for instance (Moran and Yehudayoff 2015).
  - Double descent (Belkin et al. 2018; Belkin, Hsu, and Xu 2019; Hastie et al. 2019), and related "interpolating predictors."

- Various omitted bounds in our uniform deviation framework:
  - (Wei and Ma 2019) give a bound which requires smooth activations; if we convert it to ReLU, it introduces a large factor which does not seem to improve over those presented here. That said, it is an interesting bound and approach. (There are a number of other bounds we don't discuss since similarly they degrade for ReLU.)
  - (Golowich, Rakhlin, and Shamir 2018) have an additional bound over the one of theirs we present here: interestingly, it weakens the depends on $\sqrt{n}$ to $n^{1/4}$ or $n^{1/5}$ but in exchange vastly improves the dependence on norms in the numerator, and is a very interesting bound.

12 Concentration of measure

- **Concentration of measure** studies how certain distribution families and operations on distributions lead to "clumping up" of probability mass. Examples we've seen:
  - Gaussians concentrate around the one-standard-deviation shell; we used this in NTK to say few activations change (so it's concentrated *away* from 0, sometimes this is called "anti-concentration").
  - Azuma-Hoeffding gave us control on the errors in SGD; note that we averaged together many errors before studying concentration!
- We'll see in this section that concentration of measure allows us to handle the generalization gap of *single predictors fixed in advance*, but is insufficient to handle the output of training algorithms.
- We will be **absurdly brief.** Some other resources:
  - Martin Wainwright's lecture notes (Wainwright 2015), now turned into a book (Wainwright 2019).
  - My learning theory class, as well as Maxim Raginsky's.

12.1 sub-Gaussian random variables and Chernoff's bounding technique

Our main concentration tool will be the *Chernoff bounding method*, which works nicely with *sub-Gaussian* random variables.

Definition 12.1 Random variable Z is *sub-Gaussian with mean μ and variance proxy σ^2* when $\mathbb{E}\, e^{\lambda(Z-\mu)} \le e^{\lambda^2\sigma^2/2}$.

Example 12.1

- Gaussian $\mathcal{N}(\mu, \sigma^2)$ is (μ, σ^2)-sub-Gaussian.
- $\sum_i Z_i$ is $(\sum_i \mu, \|\vec{\sigma}\|^2)$ when Z_i is (μ_i, σ^2)-sub-Gaussian.
- If $Z \in [a, b]$ a.s., then Z is $(\mathbb{E}\, Z, (b-a)^2/4)$-sub-Gaussian (this is called the Hoeffding lemma [mjt☹: I should pick a proof.]).

Remark 12.1 There is also "sub-exponential"; we will not use it but it is fundamental.

Sometimes μ is dropped from definition; in this case, one can study $X - \mathbb{E}\,X$, and we'll just say "σ^2-sub-Gaussian."

$\mathbb{E}\exp(\lambda Z)$ is the *moment generating function* of Z; it has many nice properties, though we'll only use it in a technical way.

sub-Gaussian random variables will be useful to us due to their vanishing tail probabilities. This indeed is an equivalent way to define sub-Gaussian (see (Wainwright 2015)), but we'll just prove implication. The first step is Markov's inequality.

Theorem 12.1 ***(Markov's inequality)*** For any nonnegative r.v. X and $\epsilon > 0$,

$$\Pr[X \geq \epsilon] \leq \frac{\mathbb{E}\,X}{\epsilon}.$$

Proof. Apply $\mathbb{E}$ to both sides of $\epsilon \mathbf{1}[X \geq \epsilon] \leq X$.

Corollary 12.1 For any nonnegative, nondecreasing $f \geq 0$ and $f(\epsilon) > 0$,

$$\Pr[X \geq \epsilon] \leq \frac{\mathbb{E}\,f(X)}{f(\epsilon)}.$$

Proof. Note $\Pr[X \geq \epsilon] \leq \Pr[f(X) \geq f(\epsilon)]$ and apply Markov.

The Chernoff bounding technique is as follows. We can apply the proceeding corollary to the mapping $t \mapsto \exp(tX)$ for all $t > 0$: supposing $\mathbb{E}\,X = 0$,

$$\Pr[X \geq \epsilon] = \inf_{t \geq 0} \Pr[\exp(tX) \geq \exp(t\epsilon)] \leq \inf_{t \geq 0} \frac{\mathbb{E}\exp(tX)}{\exp(t\epsilon)}.$$

Simplifying the RHS via sub-Guassianity,

$$\begin{aligned}\inf_{t>0} \frac{\mathbb{E}\exp(tX)}{\exp(t\epsilon)} &\leq \inf_{t>0} \frac{\exp(t^2\sigma^2/2)}{\exp(t\epsilon)} = \inf_{t>0} \exp\left(t^2\sigma^2/2 - t\epsilon\right) \\ &= \exp\left(\inf_{t>0} t^2\sigma^2/2 - t\epsilon\right).\end{aligned}$$

The minimum of this convex quadratic is $t := \frac{\epsilon}{\sigma^2} > 0$, thus

$$\Pr[X \geq \epsilon] = \inf_{t>0} \frac{\mathbb{E}\exp(tX)}{\exp(t\epsilon)} \leq \exp\left(\inf_{t>0} t^2\sigma^2/2 - t\epsilon\right) = \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right). \tag{7}$$

What if we apply this to an average of sub-Gaussian r.v.'s? (The point is: this starts to look like an empirical risk!)

Theorem 12.2 ***(Chernoff bound for subgaussian r.v.'s)*** Suppose $(X_1, \ldots, X_n)$ independent and respectively σ_i^2-subgaussian. Then

$$\Pr\left[\frac{1}{n}\sum_i X_i \geq \epsilon\right] \leq \exp\left(-\frac{n^2\epsilon^2}{2\sum_i \sigma_i^2}\right).$$

In other words ("inversion form"), with probability $\geq 1-\delta$,

$$\frac{1}{n}\sum_i \mathbb{E}\, X_i \leq \frac{1}{n}\sum_i X_i + \sqrt{\frac{2\sum_i \sigma_i^2}{n^2}\ln\left(\frac{1}{\delta}\right)}.$$

Proof. $S_n := \sum_i X_i/n$ is σ^2-subgaussian with $\sigma^2 = \sum_i \sigma_i^2/n^2$; plug this into the sub-Gaussian tail bound in eq. 7.

Remark 12.2 (Gaussian sanity check.) Let's go back to the case $n = 1$. It's possible to get a tighter tail for the Gaussian directly (see (Wainwright 2015)), but it only changes log factors in the "inversion form" of the bound. Note also the bound is neat for the Gaussian since it says the tail mass and density are of the same order (algebraically this makes sense, as with geometric series).

("Inversion" form.) This form is how things are commonly presented in machine learning; think of δ as "confidence"; $\ln(1/\delta)$ term means adding more digits to the confidence (e.g., bound holds with probability 99.999%) means a linear increase in the term $\ln(1/\delta)$.

There are more sophisticated bounds (e.g., Bernstein, Freedman, McDiarmid) proved in similar ways, often considering a Martingale rather than IID r.v.s.

[mjt☹: I should say something about necessary and sufficient, like convex lipschitz bounded vs lipschitz gaussian.]

[mjt☹: maybe give heavy tail poiinter? dunno.]

12.2 Hoeffding's inequality and the need for uniform deviations

Let's use what we've seen to bound misclassifications!

Theorem 12.3 ***(Hoeffding inequality)*** Given independent $(X_1, \ldots, X_n)$ with $X_i \in [a_i, b_i]$ a.s.,

$$\Pr\left[\frac{1}{n}\sum_i (X_i - \mathbb{E}\, X_i) \geq \epsilon\right] \leq \exp\left(-\frac{2n^2\epsilon^2}{\sum_i (b_i - a_i)^2}\right).$$

Proof. Plug Hoeffding Lemma into sub-Gaussian Chernoff bound.

Example 12.2 Fix classifier f, sample $((X_i, Y_i))_{i=1}^n$, and define $Z_i := \mathbf{1}[f(X_i) \neq Y_i]$. With probability at least $1-\delta$,

$$\Pr[f(X) \neq Y] - \frac{1}{n}\sum_{i=1}^n \mathbf{1}[f(x_i) = y_i] = \mathbb{E}\, Z_1 - \frac{1}{n}\sum_{i=1}^n Z_i \leq \sqrt{\frac{1}{2n}\ln\left(\frac{1}{\delta}\right)}.$$

As in, test error is upper bounded by training error plus a term which goes $\downarrow 0$ as $n \to \infty$!

Example 12.3 Classifier f_n memorizes training data:

$$f_n(x) := \begin{cases} y_i & x = x_i \in (x_1, \ldots, x_n), \\ 17 & \text{otherwise.} \end{cases}$$

Consider two situations with $\Pr[Y = +1 | X = x] = 1$.

- Suppose marginal on X has finite support. Eventually (large n), this support is memorized and $\widehat{\mathcal{R}}_{\mathrm{z}}(f_n) = 0 = \mathcal{R}_{\mathrm{z}}(f_n)$.
- Suppose marginal on X is continuous. With probability 1, $\widehat{\mathcal{R}}_{\mathrm{z}}(f_n) = 0$ but $\mathcal{R}_{\mathrm{z}}(f_n) = 1$!

What broke Hoeffding's inequality (and its proof) between these two examples?

- f_n is a *random variable* depending on $S = ((x_i, y_i))_{i=1}^n$. Even if $((x_i, y_i))_{i=1}^n$ are independent, the new random variables $Z_i := \mathbf{1}[f_n(x_i) \neq y_i]$ are not !

This f_n **overfit**: $\widehat{\mathcal{R}}(f_n)$ is small, but $\mathcal{R}(f_n)$ is large.

Possible fixes.

- **Two samples:** train on S_1, evaluate on S_2. But now we're using less data, and run into the same issue if we evaluate multiple predictors on S_2.
- **Restrict access to data within training algorithm:** SGD does this, and has a specialized (martingale-based) deviation analysis.
- **Uniform deviations:** define a new r.v. controlling errors of *all possible predictors* $\mathcal{F}$ *the algorithm might output*:

$$\left[\sup_{f \in \mathcal{F}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right].$$

  This last one is the approach we'll follow here. It can be adapted to data and algorithms by adapting $\mathcal{F}$ (we'll discuss this more shortly).

Remark 12.3 There are measure-theoretic issues with the uniform deviation approach, which we'll omit here. Specifically, the most natural way to reason about

$$\left[\sup_{f \in \mathcal{F}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right]$$

is via uncountably intersections of events, which are not guaranteed to be within the σ-algebra. The easiest fix is to worth with countable subfamilies, which will work for the standard ReLU networks we consider.

13 Rademacher complexity

As before we will apply a brute-force approach to controlling generalization over a function family $\mathcal{F}$: we will simultaneously control generalization for all elements of the class by working with the random variable

$$\left[\sup_{f \in \mathcal{F}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right].$$

This is called "uniform deviations" because we prove a deviation bound that holds *uniformly* over all elements of $\mathcal{F}$.

Remark 13.1 The idea is that even though our algorithms output predictors which depend on data, we circumvent the independence issue by invoking a uniform bound on all elements of $\mathcal{F}$ *before* we see the algorithm's output, and thus generalization is bounded for the algorithm output (and for everything else in the class). This is a brute-force approach because it potentially controls much more than is necessary.

On the other hand, we can adapt the approach to the output of the algorithm in various ways, as we will discuss after presenting the main Rademacher bound.

Example 13.1 ***(finite classes)*** As an example of what is possible, suppose we have $\mathcal{F} = (f_1, \ldots, f_k)$, meaning a finite function class $\mathcal{F}$ with $|\mathcal{F}| = k$. If we apply Hoeffding's inequality to each element of $\mathcal{F}$ and then union bound, we get, with probability at least $1 - \delta$, for every $f \in \mathcal{F}$,

$$\Pr[f(X) \neq Y] - \widehat{\Pr}[f(X) \neq Y] \leq \sqrt{\frac{\ln(k/\delta)}{2n}} \leq \sqrt{\frac{\ln|\mathcal{F}|}{2n}} + \sqrt{\frac{\ln(1/\delta)}{2n}}.$$

Rademacher complexity will give us a way to replace $\ln|\mathcal{F}|$ in the preceding finite class example with something non-trivial in the case $|\mathcal{F}| = \infty$.

Definition 13.1 ***(Rademacher complexity)*** Given a set of vectors $V \subseteq \mathbb{R}^n$, define the **(unnormalized) Rademacher complexity** as

$$\mathrm{URad}(V) := \mathbb{E} \sup_{u \in V} \langle \epsilon, u \rangle, \qquad \mathrm{Rad}(V) := \frac{1}{n}\mathrm{URad}(V),$$

where $\mathbb{E}$ is uniform over the corners of the hyerpcube over $\epsilon \in \{\pm 1\}^n$ (each coordinate ϵ_i is a *Rademacher random variable*, meaning $\Pr[\epsilon_i = +1] = \frac{1}{2} = \Pr[\epsilon_i = -1]$, and all coordinates are iid).

This definition can be applied to arbitrary elements of $\mathbb{R}^n$, and is useful outside machine learning. We will typically apply it to the behavior of a function class on $S = (z_i)_{i=1}^n$:

$$\mathcal{F}_{|S} := \{(f(x_1), \ldots, f(x_n)) : f \in \mathcal{F}\} \subseteq \mathbb{R}^n.$$

With this definition,

$$\mathrm{URad}(\mathcal{F}_{|S}) = \mathop{\mathbb{E}}_{\epsilon} \sup_{u \in \mathcal{F}_{|S}} \langle \epsilon, u \rangle = \mathop{\mathbb{E}}_{\epsilon} \sup_{f \in \mathcal{F}} \sum_i \epsilon_i f(z_i).$$

Remark 13.2 **(Loss classes.)** This looks like fitting random signs, but is not exactly that; often we apply it to the *loss class*: overloading notation,

$$\mathrm{URad}((\ell \circ \mathcal{F})_{|S}) = \mathrm{URad}\left(\{(\ell(y_1 f(x_1)), \ldots, \ell(y_n f(x_n))) : f \in \mathcal{F}\}\right).$$

(Sanity checks.) We'd like $\mathrm{URad}(V)$ to measure how "big" or "complicated" V is. Here are a few basic checks:

1. $\mathrm{URad}(\{u\}) = \mathbb{E}\langle \epsilon, u \rangle = 0$; this seems desirable, as a $|V| = 1$ is simple.
2. More generally, $\mathrm{URad}(V + \{u\}) = \mathrm{URad}(V)$.
3. If $V \subseteq V'$, then $\mathrm{URad}(V) \leq \mathrm{URad}(V')$.
4. $\mathrm{URad}(\{\pm 1\}^n) = \mathbb{E}_\epsilon \epsilon^2 = n$; this also seems desirable, as V is as big/complicated as possible (amongst bounded vectors).
5. $\mathrm{URad}(\{(-1, \ldots, -1), (+1, \ldots, +1)\}) = \mathbb{E}_\epsilon |\sum_i \epsilon_i| = \Theta(\sqrt{n})$. This also seems reasonable: $|V| = 2$ and it is not completely trivial.

(URad vs Rad.) I don't know other texts or even papers which use URad, I only see Rad. I prefer URad for these reasons:

1. The $1/n$ is a nuisance while proving Rademacher complexity bounds.
2. When we connect Rademacher complexity to covering numbers, we need to change the norms to account for this $1/n$.
3. It looks more like a regret quantity.

(Absolute value version.) The original definition of Rademacher complexity (P. L. Bartlett and Mendelson 2002), which still appears in many papers and books, is

$$\mathrm{URad}_{|\cdot|}(V) = \mathop{\mathbb{E}}_{\epsilon} \sup_{u \in V} |\langle \epsilon, u \rangle| .$$

Most texts now drop the absolute value. Here are my reasons:

1. $\mathrm{URad}_{|\cdot|}$ violates basic sanity checks: it is possible that $\mathrm{URad}_{|\cdot|}(\{u\}) \neq 0$ and more generally $\mathrm{URad}_{|\cdot|}(V + \{u\}) \neq \mathrm{URad}_{|\cdot|}(V)$, which violates my basic intuition about a "complexity measure."
2. To obtain $1/n$ rates rather than $1/\sqrt{n}$, the notion of *local Rademacher complexity* was introduced, which necessitated dropping the absolute value essentially due to the preceding sanity checks.
3. We can use URad to reason about $\mathrm{URad}_{|\cdot|}$, since $\mathrm{URad}_{|\cdot|}(V) = \mathrm{URad}(V \cup -V)$.
4. While $\mathrm{URad}_{|\cdot|}$ is more convenient for certain operations, e.g., $\mathrm{URad}_{|\cdot|}(\cup_i V_i) \leq \sum_i \mathrm{URad}_{|\cdot|}(V_i)$, there are reasonable surrogates for URad (as below).

The following theorem shows indeed that we can use Rademacher complexity to replace the $\ln |\mathcal{F}|$ term from the finite-class bound with something more general (we'll treat the Rademacher complexity of finite classes shortly).

Theorem 13.1 Let $\mathcal{F}$ be given with $f(z) \in [a, b]$ a.s. $\forall f \in \mathcal{F}$.

1. With probability $\geq 1 - \delta$,

$$\sup_{f \in \mathcal{F}} \mathbb{E} f(Z) - \frac{1}{n} \sum_i f(z_i) \leq \mathop{\mathbb{E}}_{(z_i)_{i=1}^n} \left(\sup_{f \in \mathcal{F}} \mathbb{E} f(z) - \frac{1}{n} \sum_i f(z_i) \right) + (b-a) \sqrt{\frac{\ln(1/\delta)}{2n}}.$$

2. With probability $\geq 1 - \delta$,

$$\mathop{\mathbb{E}}_{(z_i)_{i=1}^n} \mathrm{URad}(\mathcal{F}_{|S}) \leq \mathrm{URad}(\mathcal{F}_{|S}) + (b-a) \sqrt{\frac{n \ln(1/\delta)}{2}}.$$

3. With probability $\geq 1 - \delta$,

$$\sup_{f \in \mathcal{F}} \mathbb{E} f(Z) - \frac{1}{n} \sum_i f(z_i) \leq \frac{2}{n} \mathrm{URad}(\mathcal{F}_{|S}) + 3(b-a) \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Remark 13.3 To flip which side has an expectation and which side has an average of random variables, replace $\mathcal{F}$ with $-\mathcal{F} := \{-f : f \in \mathcal{F}\}$.

The proof of this bound has many interesting points and is spread out over the next few subsections. It has these basic steps:

1. The *expected* uniform deviations are upper bounded by the *expected* Rademacher complexity. This itself is done in two steps:
    1. The expected deviations are upper bounded by expected deviations between two finite samples. This is interesting since we could have reasonably defined generalization in terms of this latter quantity.
    2. These two-sample deviations are upper bounded by expected Rademacher complexity by introducing random signs.
2. We replace this difference in expectations with high probability bounds via a more powerful concentration inequality which we haven't discussed, *McDiarmid's inequality*.

13.1 Generalization *without* concentration; symmetrization

We'll use further notation throughout this proof.

$$\begin{aligned} Z &\quad \text{r.v.; e.g., } (x, y), \\ \mathcal{F} &\quad \text{functions; e.g., } f(Z) = \ell(g(X), Y), \\ \mathbb{E} &\quad \text{expectation over } Z, \\ \mathop{\mathbb{E}}_{n} &\quad \text{expectation over } (Z_1, \dots, Z_n), \\ \mathbb{E}\, f &= \mathbb{E}\, f(Z), \\ \widehat{\mathbb{E}}_n f &= \frac{1}{n} \sum_i f(Z_i). \end{aligned}$$

In this notation, $\mathcal{R}_\ell(g) = \mathbb{E}\, \ell \circ g$ and $\widehat{\mathcal{R}}_\ell(g) = \widehat{\mathbb{E}} \ell \circ g$.

13.1.1 Symmetrization with a ghost sample

In this first step we'll introduce another sample ("ghost sample"). Let $(Z'_1, \dots, Z'_n)$ be another iid draw from Z; define $\mathbb{E}'_n$ and $\widehat{\mathbb{E}}'_n$ analogously.

Lemma 13.1 $\mathbb{E}_n \left(\sup_{f \in \mathcal{F}} \mathbb{E}\, f - \widehat{\mathbb{E}}_n f \right) \le \mathbb{E}_n\, \mathbb{E}'_n \left(\sup_{f \in \mathcal{F}} \widehat{\mathbb{E}}'_n f - \widehat{\mathbb{E}}_n f \right)$.

Proof. Fix any $\epsilon > 0$ and apx max $f_\epsilon \in \mathcal{F}$; then

$$\begin{aligned} \mathop{\mathbb{E}}_{n} \left(\sup_{f \in \mathcal{F}} \mathbb{E}\, f - \widehat{\mathbb{E}}_n f \right) &\le \mathop{\mathbb{E}}_{n} \left(\mathbb{E}\, f_\epsilon - \widehat{\mathbb{E}}_n f_\epsilon \right) + \epsilon \\ &= \mathop{\mathbb{E}}_{n} \left(\mathop{\mathbb{E}}_{n}' \widehat{\mathbb{E}}'_n f_\epsilon - \widehat{\mathbb{E}}_n f_\epsilon \right) + \epsilon \\ &= \mathop{\mathbb{E}}_{n}' \mathop{\mathbb{E}}_{n} \left(\widehat{\mathbb{E}}'_n f_\epsilon - \widehat{\mathbb{E}}_n f_\epsilon \right) + \epsilon \\ &\le \mathop{\mathbb{E}}_{n}' \mathop{\mathbb{E}}_{n} \left(\sup_{f \in \mathcal{F}} \widehat{\mathbb{E}}'_n f - \widehat{\mathbb{E}}_n f \right) + \epsilon \end{aligned}$$

Result follows since $\epsilon > 0$ was arbitrary.

Remark 13.4 As above, in this section we are working only *in expectation* for now. In the subsequent section, we'll get high probability bounds. But $\sup_{f\in\mathcal{F}} \mathbb{E} f - \mathbb{E}'_n f$ is a random variable; can describe it in many other ways too! (E.g., "asymptotic normality.")

As mentioned before, the preceding lemma says we can instead work with two samples. Working with two samples could have been our starting point (and definition of generalization): by itself it is a meaningful and interpretable quantity!

13.1.2 Symmetrization with random signs

The second step swaps points between the two samples; a magic trick with random signs boils this down into Rademacher complexity.

Lemma 13.2 $\mathbb{E}_n \mathbb{E}'_n \sup_{f\in\mathcal{F}} \left(\widehat{\mathbb{E}}'_n f - \widehat{\mathbb{E}}_n f\right) \le \frac{2}{n} \mathbb{E}_n \mathrm{URad}(\mathcal{F}_{|S})$.

Proof. Fix a vector $\epsilon \in \{-1,+1\}^n$ and define a r.v. $(U_i, U'_i) := (Z_i, Z'_i)$ if $\epsilon = 1$ and $(U_i, U'_i) = (Z'_i, Z_i)$ if $\epsilon = -1$. Then

$$\begin{aligned}\mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \widehat{\mathbb{E}}'_n f - \widehat{\mathbb{E}}_n f\right) &= \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \left(f(Z'_i) - f(Z_i)\right)\right) \\ &= \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(f(U'_i) - f(U_i)\right)\right).\end{aligned}$$

Here's the big trick: since $(Z_1,\dots,Z_n,Z'_1,\dots,Z'_n)$ and $(U_1,\dots,U_n,U'_1,\dots,U'_n)$ have **same distribution**, and ϵ arbitrary, then (with $\Pr[\epsilon_i = +1] = 1/2$ iid "Rademacher")

$$\begin{aligned}\mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \widehat{\mathbb{E}}'_n f - \widehat{\mathbb{E}}_n f\right) &= \mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(f(U'_i) - f(U_i)\right)\right) \\ &= \mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(f(Z'_i) - f(Z_i)\right)\right).\end{aligned}$$

Since similarly replacing ϵ_i and $-\epsilon_i$ doesn't change $\mathbb{E}_\epsilon$,

$$\begin{aligned}&\mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \widehat{\mathbb{E}}'_n f - \widehat{\mathbb{E}}_n f\right) \\ &= \mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(f(Z'_i) - f(Z_i)\right)\right) \\ &\le \mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n \mathop{\mathbb{E}}_n' \left(\sup_{f,f'\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(f(Z'_i) - f'(Z_i)\right)\right) \\ &= \mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n' \left(\sup_{f'\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(f(Z'_i)\right)\right) + \mathop{\mathbb{E}}_\epsilon \mathop{\mathbb{E}}_n' \left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_i \epsilon_i \left(-f'(Z_i)\right)\right) \\ &= 2 \mathop{\mathbb{E}}_n \frac{1}{n} \mathop{\mathbb{E}}_\epsilon \sup_{f\in\mathcal{F}} \sum_i \epsilon_i \left(f(Z_i)\right) = 2 \mathop{\mathbb{E}}_n \frac{1}{n} \mathrm{URad}(\mathcal{F}_{|S}).\end{aligned}$$

13.2 Generalization *with* concentration

We controlled *expected* uniform deviations: $\mathbb{E}_n \sup_{f\in\mathcal{F}} \mathbb{E}\, f - \widehat{\mathbb{E}}_n f$.

High probability bounds will follow via concentration inequalities.

Theorem 13.2 ***(McDiarmid)*** Suppose $F : \mathbb{R}^n \to \mathbb{R}$ satisfies "bounded differences": $\forall i \in \{1,\dots,n\}$ $\exists c_i$,

$$\sup_{z_1,\dots,z_n,z_i'} \left|F(z_1,\dots,z_i,\dots,z_n) - F(z_1,\dots,z_i',\dots,z_n)\right| \le c_i.$$

With pr $\ge 1-\delta$,

$$\mathop{\mathbb{E}}_n F(Z_1,\dots,Z_n) \le F(Z_1,\dots,Z_n) + \sqrt{\frac{\sum_i c_i^2}{2}\ln(1/\delta)}.$$

Remark 13.5 I'm omitting the proof. A standard way is via a *Martingale* variant of the Chernoff bounding method. The Martingale adds one point at a time, and sees how things grow.

Hoeffding follows by setting $F(\vec{Z}) = \sum_i Z_i/n$ and verifying bounded differences $c_i := (b_i - a_i)/n$.

Proof of +Theorem 13.1.
The third bullet item follows from the first two by union bounding. To prove the first two, it suffices to apply the earlier two lemmas on expectations and verify the quantities satisfy bounded differences with constant $(b-a)/n$ and $(b-a)$, respectively.
For the first quantity, for any i and $(z_1,\dots,z_n,z_i')$ and writing $z_j' := z_j$ for $z_j \neq z_i$ for convenience,

$$\begin{aligned}
\left|\sup_{f\in\mathcal{F}} \mathbb{E}\, f - \widehat{\mathbb{E}}_n f - \sup_{g\in\mathcal{F}}(\mathbb{E}\, g - \widehat{\mathbb{E}}_n' g)\right| &= \left|\sup_{f\in\mathcal{F}} \mathbb{E}\, f - \widehat{\mathbb{E}}_n f - \sup_{g\in\mathcal{F}}(\mathbb{E}\, g - \widehat{\mathbb{E}}_n g + g(z_i) - g(z_i'))\right| \\
&\le \sup_{h\in\mathcal{F}}\left|\sup_{f\in\mathcal{F}} \mathbb{E}\, f - \widehat{\mathbb{E}}_n f - \sup_{g\in\mathcal{F}}(\mathbb{E}\, g - \widehat{\mathbb{E}}_n g + h(z_i)/n - h(z_i'))/n\right| \\
&= \sup_{h\in\mathcal{F}} \left|h(z_i) - h(z_i'))\right|/n \\
&\le \frac{b-a}{n}.
\end{aligned}$$

Using similar notation, and additionally writing S and S' for the two samples, for the Rademacher complexity,

$$\begin{aligned}
\left|\mathrm{URad}(\mathcal{F}_{|S}) - \mathrm{URad}(\mathcal{F}_{|S'})\right| &= \left|\mathrm{URad}(\mathcal{F}_{|S}) - \mathop{\mathbb{E}}_{\epsilon}\sup_{f\in\mathcal{F}}\sum_{i=1}^n \epsilon_i f(z_i')\right| \\
&= \left|\mathrm{URad}(\mathcal{F}_{|S}) - \mathop{\mathbb{E}}_{\epsilon}\sup_{f\in\mathcal{F}}\sum_{i=1}^n \epsilon_i f(z_i) - \epsilon_i f(z_i) + \epsilon_i f(z_i')\right| \\
&\le \sup_{h\in\mathcal{F}}\left|\mathrm{URad}(\mathcal{F}_{|S}) - \mathop{\mathbb{E}}_{\epsilon}\sup_{f\in\mathcal{F}}\sum_{i=1}^n \epsilon_i f(z_i) - \epsilon_i h(z_i) + \epsilon_i h(z_i')\right| \\
&\le \sup_{h\in\mathcal{F}} \mathop{\mathbb{E}}_{\epsilon_i}\left|\epsilon_i h(z_i) + \epsilon_i h(z_i')\right| \le (b-a).
\end{aligned}$$

13.3 Example: basic logistic regression generalization analysis

Let's consider logistic regression with bounded weights:

$$\begin{aligned}
\ell(yf(x)) &:= \ln(1+\exp(-yf(x))),\\
|\ell'| &\le 1,\\
\mathcal{F} &:= \left\{ w \in \mathbb{R}^d : \|w\| \le B \right\},\\
(\ell \circ \mathcal{F})_{|S} &:= \{(\ell(y_1 w^\mathsf{T} x_1), \dots, \ell(y_n w^\mathsf{T} x_n)) : \|w\| \le B\},\\
\mathcal{R}_\ell(w) &:= \mathbb{E}\, \ell(Y w^\mathsf{T} X),\\
\widehat{\mathcal{R}}_\ell(w) &:= \frac{1}{n}\sum_i \ell(y_i w^\mathsf{T} x_i).
\end{aligned}$$

The goal is to control $\mathcal{R}_\ell - \widehat{\mathcal{R}}_\ell$ over $\mathcal{F}$ via the earlier theorem; our main effort is in controlling $\mathrm{URad}((\ell \circ \mathcal{F})_S)$.

This has two steps:

- "Peeling" off ℓ.
- Rademacher complexity of linear predictors.

Lemma 13.3 Let $\ell : \mathbb{R}^n \to \mathbb{R}^n$ be a vector of univariate L-lipschitz functions. Then $\mathrm{URad}(\ell \circ V) \le L \cdot \mathrm{URad}(V)$.

Proof. The idea of the proof is to "de-symmetrize" and get a difference of coordinates to which we can apply the definition of L. To start,

$$\begin{aligned}
\mathrm{URad}(\ell \circ V) &= \mathbb{E} \sup_{u \in V} \sum_i \epsilon_i \ell_i(u_i)\\
&= \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon_{2:n}} \sup_{u,w \in V} \left(\ell_1(u_1) - \ell_1(w_1) + \sum_{i=2}^n \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right)\\
&\le \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon_{2:n}} \sup_{u,w \in V} \left(L|u_1 - w_1| + \sum_{i=2}^n \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right).
\end{aligned}$$

To get rid of the absolute value, for any ϵ, by considering swapping u and w,

$$\begin{aligned}
&\sup_{u,w \in V} \left(L|u_1 - w_1| + \sum_{i=2}^n \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right)\\
&= \max \left\{ \sup_{u,w \in V} \left(L(u_1 - w_1) + \sum_{i=2}^n \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right), \right.\\
&\qquad \left. \sup_{u,w} \left(L(w_1 - u_1) + \sum_{i=2}^n \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right) \right\}\\
&= \sup_{u,w \in V} \left(L(u_1 - w_1) + \sum_{i=2}^n \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right).
\end{aligned}$$

As such,

$$\begin{aligned}\operatorname{URad}(\ell \circ V) &\leq \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon_{2:n}} \sup_{u,w \in V} \left(L|u_1 - w_1| + \sum_{i=2}^{n} \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right) \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{\epsilon_{2:n}} \sup_{u,w \in V} \left(L(u_1 - w_1) + \sum_{i=2}^{n} \epsilon_i(\ell_i(u_i) + \ell_i(w_i)) \right) \\ &= \mathop{\mathbb{E}}_{\epsilon} \sup_{u \in V} \left[L\epsilon_1 u_1 + \sum_{i=2}^{n} \epsilon_i \ell_i(u_i) \right].\end{aligned}$$

Repeating this procedure for the other coordinates gives the bound.

Revisiting our overloaded composition notation:

$$\begin{aligned}(\ell \circ f) &= ((x,y) \mapsto \ell(-yf(x))), \\ \ell \circ \mathcal{F} &= \{\ell \circ f : f \in \mathcal{F}\}.\end{aligned}$$

Corollary 13.1 Suppose ℓ is L-lipschitz and $\ell \circ \mathcal{F} \in [a,b]$ a.s.. With probability $\geq 1-\delta$, every $f \in \mathcal{F}$ satisfies

$$\mathcal{R}_\ell(f) \leq \widehat{\mathcal{R}}_\ell(f) + \frac{2L}{n} \operatorname{URad}(\mathcal{F}_{|S}) + 3(b-a)\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Proof. Use the lipschitz composition lemma with

$$\begin{aligned}|\ell(-y_i f(x_i) - \ell(-y_i f'(x_i))| &\leq L|-y_i f(x_i) + y_i f'(x_i))| \\ &\leq L|f(x_i) - f'(x_i))|.\end{aligned}$$

Now let's handle step 2: Rademacher complexity of linear predictors (in ℓ_2).

Theorem 13.3 Collect sample $S := (x_1, \ldots, x_n)$ into rows of $X \in \mathbb{R}^{n \times d}$.

$$\operatorname{URad}(\{x \mapsto \langle w, x \rangle : \|w\|_2 \leq B\}_{|S}) \leq B\|X\|_F.$$

Proof. Fix any $\epsilon \in \{-1,+1\}^n$. Then

$$\sup_{\|w\| \leq B} \sum_i \epsilon_i \langle w, x_i \rangle = \sup_{\|w\| \leq B} \left\langle w, \sum_i \epsilon_i x_i \right\rangle = B \left\| \sum_i \epsilon_i x_i \right\|.$$

We'll bound this norm with Jensen's inequality (only inequality in whole proof!):

$$\mathbb{E} \left\| \sum_i \epsilon_i x_i \right\| = \mathbb{E} \sqrt{\left\| \sum_i \epsilon_i x_i \right\|^2} \leq \sqrt{\mathbb{E} \left\| \sum_i \epsilon_i x_i \right\|^2}.$$

To finish,

$$\mathbb{E} \left\| \sum_i \epsilon_i x_i \right\|^2 = \mathbb{E} \left(\sum_i \|\epsilon_i x_i\|^2 + \sum_{i,j} \langle \epsilon_i x_i, \epsilon_j x_j \rangle \right) = \mathbb{E} \sum_i \|x_i\|^2 = \|X\|_F^2.$$

Remark 13.6 By Khinchine's inequality, the preceding Rademacher complexity estimate is tight up to constants.

Let's now return to the logistic regression example!

Example 13.2 ***(logistic regression)*** Suppose $\|w\| \le B$ and $\|x_i\| \le 1$, and the loss is the 1-Lipschitz logistic loss $\ell_{\log}(z) := \ln(1 + \exp(z))$. Note $\ell(\langle w, yx\rangle) \ge 0$ and $\ell(\langle w, yx\rangle) \le \ln(2) + \langle w, yx\rangle \le \ln(2) + B$.

Combining the main Rademacher bound with the Lipschitz composition lemma and the Rademacher bound on linear predictors, with probability at least $1 - \delta$, every $w \in \mathbb{R}^d$ with $\|w\| \le B$ satisfies

$$\begin{aligned}\mathcal{R}_\ell(w) &\le \widehat{\mathcal{R}}_\ell(w) + \frac{2}{n}\mathrm{URad}((\ell \circ \mathcal{F})_{|S}) + 3(\ln(2) + B)\sqrt{\ln(2/\delta)/(2n)} \\ &\le \widehat{\mathcal{R}}_\ell(w) + \frac{2B\|X\|_F}{n} + 3(\ln(2) + B)\sqrt{\ln(2/\delta)/(2n)} \\ &\le \widehat{\mathcal{R}}_\ell(w) + \frac{2B + 3(B + \ln(2))\sqrt{\ln(2/\delta)/2}}{\sqrt{n}}.\end{aligned}$$

Remark 13.7 (Average case vs worst case.) Here we replaced $\|X\|_F$ with the looser $\sqrt{n}$.

This bound scales as the SGD logistic regression bound proved via Azuma, despite following a somewhat different route (Azuma and McDiarmid are both proved with Chernoff bounding method; the former approach involves no symmetrization, whereas the latter holds for more than the output of an algorithm).

It would be nice to have an "average Lipschitz" bound rather than "worst-case Lipschitz"; e.g., when working with neural networks and the ReLU, which seems it can kill off many inputs! But it's not clear how to do this. Relatedly: regularizing the gradient is sometimes used in practice?

13.4 Margin bounds

In the logistic regression example, we peeled off the loss and bounded the Rademacher complexity of the predictors.

If most training labels are predicted not only accurately, but with a large margin, as in section 10, then we can further reduce the generalization bound.

Define $\ell_\gamma(z) := \max\{0, \min\{1, 1 - z/\gamma\}\}$, $\mathcal{R}_\gamma(f) := \mathcal{R}_{\ell_\gamma}(f) = \mathbb{E}\,\ell_\gamma(Yf(X))$, and recall $\mathcal{R}_{\mathrm{z}}(f) = \Pr[f(X) \ne Y]$.

Theorem 13.4 For any margin $\gamma > 0$, with probability $\ge 1 - \delta$, $\forall f \in \mathcal{F}$,

$$\mathcal{R}_{\mathrm{z}}(f) \le \mathcal{R}_\gamma(f) \le \widehat{\mathcal{R}}_\gamma(f) + \frac{2}{n\gamma}\mathrm{URad}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Proof. Since

$$\mathbf{1}[\mathrm{sgn}(f(x)) \ne y] \le \mathbf{1}[-f(x)y \ge 0] \le \ell_\gamma(f(x)y),$$

then $\mathcal{R}_{\mathrm{z}}(f) \le \mathcal{R}_\gamma(f)$. The bound between $\mathcal{R}_\gamma$ and $\widehat{\mathcal{R}}_\gamma$ follows from the fundamental Rademacher bound, and by peeling the $\frac{1}{\gamma}$-Lipschitz function ℓ_γ.

[mjt☹: is that using per-example lipschitz? need to restate peeling? also, properly invoke peeling?]

Remark 13.8 ***(bibliographic notes)*** As a generalization notion, this was first introduced for 2-layer networks in (P. L. Bartlett 1996), and then carried to many other settings (SVM, boosting, ...)

There are many different proof schemes; another one uses sparsification (Schapire et al. 1997).

This approach is again being extensively used for deep networks, since it seems that while weight matrix norms grow indefinitely, the margins grow along with them (P. Bartlett, Foster, and Telgarsky 2017).

13.5 Finite class bounds

In our warm-up example of finite classes, our complexity term was $\ln|\mathcal{F}|$. Here we will recover that, via Rademacher complexity. Moreover, the bound has a special form which will be useful in the later VC dimension and especially covering sections.

Theorem 13.5 ***(Massart finite lemma)*** $\mathrm{URad}(V) \le \sup_{u\in V}\|u\|_2\sqrt{2\ln|V|}$.

Remark 13.9 $\ln|V|$ is what we expect from union bound.

The $\|\cdot\|_2$ geometry here is intrinsic here; I don't know how to replace it with other norms without introducing looseness. This matters later when we encounter the Dudley Entropy integral.

We'll prove this via a few lemmas.

Lemma 13.4 If $(X_1,\dots,X_n)$ are c^2-subgaussian, then $\mathbb{E}\max_i X_i \le c\sqrt{2\ln(n)}$.

Proof.

$$\begin{aligned}\mathbb{E}\max_i X_i &= \inf_{t>0}\mathbb{E}\frac{1}{t}\ln\max_i\exp(tX_i) \le \inf_{t>0}\mathbb{E}\frac{1}{t}\ln\sum_i\exp(tX_i)\\ &\le \inf_{t>0}\frac{1}{t}\ln\sum_i\mathbb{E}\exp(tX_i) \le \inf_{t>0}\frac{1}{t}\ln\sum_i\exp(t^2c^2/2)\\ &= \inf_{t>0}(\ln(n)/t + c^2t/2)\end{aligned}$$

and plug in minimizer $t=\sqrt{2\ln(n)/c^2}$

Lemma 13.5 If $(X_1,\dots,X_n)$ are c_i^2-subgaussian and independent, $\sum_i X_i$ is $\|\vec{c}\|_2^2$-subgaussian.

Proof. We did this in the concentration lecture, but here it is again:

$$\mathbb{E}\exp(t\sum_i X_i) = \prod_i\mathbb{E}\exp(tX_i) \le \prod_i\exp(t^2c_i^2/2) = \exp(t^2\|\vec{c}\|_2^2/2).$$

Proof of +Theorem 13.5 (Massart finite lemma).
Let $\vec{\epsilon}$ be iid Rademacher and fix $u\in V$. Define $X_{u,i}:=\epsilon_i u_i$ and $X_u:=\sum_i X_{u,i}$.
By Hoeffding lemma, $X_{u,i}$ is $(u_i--u_i)^2/4 = u_i^2$ -subgaussian, thus (by Lemma) X_u is $\|u\|_2^2$-subgaussian. Thus

$$\mathrm{URad}(V) = \mathop{\mathbb{E}}_{\epsilon}\max_{u\in V}\langle\epsilon,u\rangle = \mathop{\mathbb{E}}_{\epsilon}\max_{u\in V}X_u \le \max_{u\in V}\|u\|_2\sqrt{2\ln|V|}.$$

13.6 Weaknesses of Rademacher complexity

[mjt☹: not an exhaustive list. . .]

- The bounds we will prove shortly are all loose. To some extent, it was argued in (Neyshabur, Tomioka, and Srebro 2014; Zhang et al. 2017) and (Nagarajan and Kolter 2019) that this may be intrinsic to Rademacher complexity, though these arguments can be overturned in various settings (in the former, via *a posteriori* bounds, e.g., as obtained via union bound; in the latter case, by considering a modified set of good predictors for the same problem); as such, that particular criticism is unclear. An alternative approach was highlighted in (Dziugaite and Roy 2017), however the bounds produced there are averages over some collection of predictors, and not directly comparable to the bounds here. Overall, though, many authors are investigating alternatives to the definition of generalization.
- Looking outside the specific setting of neural network generalization, Rademacher complexity has been widely adopted since, to a great extent, it can cleanly re-prove many existing bounds, and moreover elements of Rademacher complexity proofs exist many decades prior to the coining of the term (P. L. Bartlett and Mendelson 2002). However, already in these settings, Rademacher complexity has extensive weaknesses.
  - For many learning problems, extensive effort was put into *fast* or *optimal* learning rates, which often boiled down to replacing a $1/\sqrt{n}$ dependency with a $1/n$. While *Local Rademacher Complexity* is able to recover some of these bounds, it does not seem to recover all of them, and moreover the proofs are often very complicated.
  - In many non-parametric learning settings, for example k-nearest-neighbor, the best bounds all use a direct analysis (Chaudhuri and Dasgupta 2014), and attempts to recover these analyses with Rademacher complexity have been unsuccessful.
  - Closer to the investigation in these lecture notes, there are even cases where a direct Martingale analysis of SGD slightly beats the application of uniform convergence to the output of gradient descent, and similarly to the preceding case, attempts to close this gap have been unsuccessful (Ji and Telgarsky 2019a).

14 Two Rademacher complexity proofs for deep networks

We will give two bounds, obtained by inductively peeling off layers.

- One will depend on $\|W_i^{\mathsf{T}}\|_{1,\infty}$. This bound has a pretty clean proof, and appeared in (P. L. Bartlett and Mendelson 2002).
- The other will depend on $\|W_i^{\mathsf{T}}\|_{\mathrm{F}}$, and is more recent (Golowich, Rakhlin, and Shamir 2018).

[mjt☹: also i didn't mention yet that the other proof techniques reduce to this one?]

14.1 First "layer peeling" proof: $(1,\infty)$ norm

Theorem 14.1 Let ρ-Lipschitz activations σ_i satisfy $\sigma_i(0) = 0$, and

$$\mathcal{F} := \{x \mapsto \sigma_L(W_L\sigma_{L_1}(\cdots\sigma_1(W_1x)\cdots)) : \|W_i^{\mathsf{T}}\|_{1,\infty} \leq B\}$$

Then $\mathrm{URad}(\mathcal{F}_{|S}) \le \|X\|_{2,\infty}(2\rho B)^L\sqrt{2\ln(d)}$.

Remark 14.1 Notation $\|M\|_{b,c} = \|(\|M_{:1}\|_b, \dots, \|M_{:d}\|_b)\|_c$ means apply b-norm to columns, then c-norm to resulting vector.

Many newer bounds replace $\|W_i^{\mathrm{T}}\|$ with a distance to initialization. (The NTK is one regime where this helps.) I don't know how to use distance to initialize in the bounds in this section, but a later bound can handle it.

$(\rho B)^L$ is roughly a Lipschitz constant of the network according to ∞-norm bounded inputs. Ideally we'd have "average Lipschitz" not "worst case," but we're still far from that...

The factor 2^L is not good and the next section gives one technique to remove it.

We'll prove this with an induction "peeling" off layers. This peeling will use the following lemma, which collects many standard Rademacher properties.

Lemma 14.1

1. $\mathrm{URad}(V) \ge 0$.
2. $\mathrm{URad}(cV + \{u\}) = |c|\mathrm{URad}(V)$.
3. $\mathrm{URad}(\mathrm{conv}(V)) = \mathrm{URad}(V)$.
4. Let $(V_i)_{i\ge 0}$ be given with $\sup_{u\in V_i}\langle u, \epsilon\rangle \ge 0\ \forall \epsilon \in \{-1,+1\}^n$. (E.g., $V_i = -V_i$, or $0 \in V_i$.) Then $\mathrm{URad}(\cup_i V_i) \le \sum_i \mathrm{URad}(V_i)$.
5. $\mathrm{URad}(V) = \mathrm{URad}(-V)$.

Remark 14.2

(3) is a mixed blessing: "Rademacher is insensitive to convex hulls,"

(4) is true for $\mathrm{URad}_{|\cdot|}$ directly, where $\mathrm{URad}_{|\cdot|}(V) = \mathbb{E}_\epsilon \sup_{u\in V} |\langle \epsilon, u\rangle|$ is the original definition of (unnormalized) Rademacher complexity: define $W_i := V_i \cup -V_i$, which satisfies the conditions, and note $(\cup_i V_i) \cup -(\cup_i V_i) = \cup_i W_i$. Since $\mathrm{URad}_{|\cdot|}(V_i) = \mathrm{URad}(W_i)$, then $\mathrm{URad}_{|\cdot|}(\cup_i V_i) = \mathrm{URad}(\cup_i W_i) \le \sum_{i\ge 1}\mathrm{URad}(W_i) = \sum_{i\ge 1}\mathrm{URad}_{|\cdot|}(V_i)$. [mjt☹: is this where i messed up and clipped an older Urada remark?]

(6) is important and we'll do the proof and some implications in homework.

Proof of +Lemma 14.1.

1. Fix any $u_0 \in V$; then $\mathbb{E}_\epsilon \sup_{u\in V}\langle \epsilon, v\rangle \ge \mathbb{E}_\epsilon \langle \epsilon, u_0\rangle = 0$.
2. Can get inequality with $|c|$-Lipschitz functions $\ell_i(r) := c\cdot r + u_i$; for equality, note $-\epsilon c$ and ϵc are same in distribution.

3. This follows since optimization over a polytope is achieved at a corner. In detail,

$$\begin{aligned}\mathrm{URad}(\mathrm{conv}(V)) &= \mathop{\mathbb{E}}_{\epsilon} \sup_{\substack{k\geq 1\\ \alpha\in\Delta_k}} \sup_{u_1,\ldots,u_k\in V} \left\langle \epsilon, \sum_j \alpha_j u_j \right\rangle \\ &= \mathop{\mathbb{E}}_{\epsilon} \sup_{\substack{k\geq 1\\ \alpha\in\Delta_k}} \sum_j \alpha_j \sup_{u_j\in V} \langle \epsilon, u_j\rangle \\ &= \mathop{\mathbb{E}}_{\epsilon} \left(\sup_{\substack{k\geq 1\\ \alpha\in\Delta_k}} \sum_j \alpha_j \right) \sup_{u\in V} \langle \epsilon, u\rangle \\ &= \mathrm{URad}(V).\end{aligned}$$

4. Using the condition,

$$\begin{aligned}\mathop{\mathbb{E}}_{\epsilon} \sup_{u\in\cup_i V_i} \langle \epsilon, u\rangle = \mathop{\mathbb{E}}_{\epsilon} \sup_i \sup_{u\in V_i} \langle \epsilon, u\rangle &\leq \mathop{\mathbb{E}}_{\epsilon} \sum_i \sup_{u\in V_i} \langle \epsilon, u\rangle \\ &= \sum_{i\geq 1} \mathrm{URad}(V_i).\end{aligned}$$

5. Since integrating over ϵ is the same as integrating over $-\epsilon$ (the two are equivalent distributions),

$$\mathrm{URad}(-V) = \mathop{\mathbb{E}}_{\epsilon} \sup_{u\in V} \langle \epsilon, -u\rangle = \mathop{\mathbb{E}}_{\epsilon} \sup_{u\in V} \langle -\epsilon, -u\rangle = \mathrm{URad}(V).$$

Proof of +Theorem 14.1.
Let $\mathcal{F}_i$ denote functions computed by nodes in layer i. It'll be shown by induction that

$$\mathrm{URad}((\mathcal{F}_i)_{|S}) \leq \|X\|_{2,\infty}(2\rho B)^i \sqrt{2\ln(d)}.$$

Base case ($i = 0$): by the Massart finite lemma,

$$\begin{aligned}\mathrm{URad}((\mathcal{F}_i)_{|S}) &= \mathrm{URad}\left(\{x \mapsto x_j : j \in \{1,\ldots,d\}\}_{|S}\right) \\ &\leq \left(\max_j \|(x_1)_j, \ldots, (x_n)_j\|_2\right) \sqrt{2\ln(d)} \\ &= \|X\|_{2,\infty}\sqrt{2\ln d} = \|X\|_{2,\infty}(2\rho B)^0\sqrt{2\ln d}.\end{aligned}$$

Inductive step. Since $0 = \sigma(\langle 0, F(x)\rangle) \in \mathcal{F}_{i+1}$, applying both Lipschitz peeling and the preceding multi-part lemma,

$$\begin{aligned}&\mathrm{URad}((\mathcal{F}_{i+1})_{|S}) \\ &= \mathrm{URad}\left(\{x \mapsto \sigma_{i+1}(\|W_{i+1}^{\mathsf{T}}\|_{1,\infty} g(x)) : g \in \mathrm{conv}(-\mathcal{F}_i \cup \mathcal{F}_i)\}_{|S}\right) \\ &\leq \rho B \cdot \mathrm{URad}\left(-(\mathcal{F}_i)_{|S} \cup (\mathcal{F}_i)_{|S}\right) \\ &\leq 2\rho B \cdot \mathrm{URad}\left((\mathcal{F}_i)_{|S}\right) \\ &\leq (2\rho B)^{i+1}\|X\|_{2,\infty}\sqrt{2\ln d}.\end{aligned}$$

Remark 14.3 There are many related norm-based proofs now changing constants and also

$(1, \infty)$; see for instance Neyshabur-Tomioka-Srebro, Bartlett-Foster-Telgarsky (we'll cover this), Golowich-Rakhlin-Shamir (we'll cover this), Barron-Klusowski.

The best lower bound is roughly what you get by writing a linear function as a deep network $\ddot{\frown}$.

The proof does not "coordinate" the behavior of adjacent layers in any way, and worst-cases what can happen.

14.2 Second "layer peeling" proof: Frobenius norm

Theorem 14.2 ***((Theorem 1, Golowich, Rakhlin, and Shamir 2018))*** Let 1-Lipschitz positive homogeneous activation σ_i be given, and

$$\mathcal{F} := \{x \mapsto \sigma_L(W_L \sigma_{L_1}(\cdots \sigma_1(W_1 x) \cdots)) : \|W_i\|_{\mathrm{F}} \leq B\}.$$

Then

$$\mathrm{URad}(\mathcal{F}_{|S}) \leq B^L \|X\|_{\mathrm{F}} \left(1 + \sqrt{2L \ln(2)}\right).$$

Remark 14.4 The criticisms of the previous layer peeling proof still apply, except we've removed 2^L.

The proof technique can also handle other matrix norms (though with some adjustment), bringing it closer to the previous layer peeling proof.

For an earlier version of this bound but including things like 2^L, see Neyshabur-Tomioka-Srebro. [mjt☹: I need a proper citation]

The main proof trick (to remove 2^L) is to replace $\mathbb{E}_\epsilon$ with $\ln \mathbb{E}_\epsilon \exp$; the 2^L now appears inside the ln.

To make this work, we need two calculations, which we'll wrap up into lemmas.

- When we do "Lipschitz peeling," we now have to deal with exp inside $\mathbb{E}_\epsilon$. Magically, things still work, but the proof is nastier, and we'll not include it.
- That base case of the previous layer peeling could by handled by the Massart finite lemma; this time we end up with something of the form $\mathbb{E}_\epsilon \exp(t\|X^{\mathsf{T}}\epsilon\|_2)$.
- Comparing to the $\infty \to \infty$ operator norm (aka $(1, \infty)$) bound, let's suppose $W \in \mathbb{R}^{m \times m}$ has row/node norm $\|W_{j:}\|_2 \approx 1$, thus $\|W_{j:}\|_1 \approx \sqrt{m}$, and

  $$\|W\|_{\mathrm{F}} \approx \sqrt{m} \approx \|W^{\mathsf{T}}\|_{1,\infty},$$

  so this bound only really improves on the previous one by removing 2^L?

Here is our refined Lipschitz peeling bound, stated without proof.

Lemma 14.2 ***((Eq. 4.20, Ledoux and Talagrand 1991))*** Let $\ell : \mathbb{R}^n \to \mathbb{R}^n$ be a vector of univariate ρ-lipschitz functions with $\ell_i(0) = 0$. Then

$$\mathop{\mathbb{E}}_{\epsilon} \exp\left(\sup_{u \in V} \sum_i \epsilon_i \ell_i(u_i)\right) \leq \mathop{\mathbb{E}}_{\epsilon} \exp\left(\rho \sup_{u \in V} \sum_i \epsilon_i u_i\right).$$

Remark 14.5 With exp gone, our proof was pretty clean, but all proofs I know of this are more complicated case analyses. So I will not include a proof ⌢̈.

The peeling proof will end with a term $\mathbb{E}\exp(t\|X^\intercal\epsilon\|)$, and we'll optimize the t to get the final bound. Consequently, we are proving $\|X^\intercal\epsilon\|$ is sub-Gaussian!

Lemma 14.3 $\mathbb{E}\|X^\intercal\epsilon\|_2 \le \|X\|_{\mathrm{F}}$ and $\|X^\intercal\epsilon\|$ is $(\mathbb{E}\|X^\intercal\epsilon\|, \|X\|_{\mathrm{F}}^2)$-sub-Gaussian.

Proof. Following the notation of (Wainwright 2015), define

$$Y_k := \mathbb{E}\left[\|X^\intercal\epsilon\|_2 | \epsilon_1, \dots, \epsilon_k\right], D_k := Y_k - Y_{k-1},$$

whereby $Y_n - Y_0 = \sum_k D_k$. For the base case, as usual

$$\mathbb{E}\|X^\intercal\epsilon\|_2 \le \sqrt{\mathbb{E}\|X^\intercal\epsilon\|^2} = \sqrt{\sum_{j=1}^d \mathbb{E}(X_{:j}^\intercal\epsilon)^2} = \sqrt{\sum_{j=1}^d \|X_{:j}\|^2} = \|X\|_{\mathrm{F}}.$$

Supposing ϵ and ϵ' only differ on ϵ_k,

$$\begin{aligned}\sup_{\epsilon_k} |\|X^\intercal\epsilon\| - \|X^\intercal\epsilon'\||^2 \le \sup_{\epsilon_k} \|X^\intercal(\epsilon - \epsilon')\|^2 &= \sup_{\epsilon_k} \sum_{j=1}^d (X_{:j}^\intercal(\epsilon - \epsilon'))^2 \\ &= \sup_{\epsilon_k} \sum_{j=1}^d (X_{k,j}(\epsilon_k - \epsilon_k'))^2 \le 4\|X_{k:}\|^2,\end{aligned}$$

therefore by the (conditional) Hoeffding lemma, D_k is $\|X_{k:}\|^2$-sub-Gaussian, thus (Theorem 2.3, Wainwright 2015) grants $\sum_k D_k$ is σ^2-sub-Gaussian with $\sigma^2 = \sum_k \|X_{k:}\|^2 = \|X\|_{\mathrm{F}}^2$.

Remark 14.6 ***(pointed out by Ziwei Ji)*** Alternatively, we can use the Lipschitz-convex concentration bound for bounded random variables, and get a variance proxy of roughly $\|X\|_2$. Plugging this into the full peeling proof, we get an interesting bound $B^L\left(\|X\|_{\mathrm{F}} + \|X\|_2\sqrt{128L\ln(2)}\right)$, thus dimension and depth don't appear together.

Proof of +Theorem 14.2 ((Theorem 1, Golowich, Rakhlin, and Shamir 2018)). For convenience, let X_i denote the output of layer i, meaning

$$X_0 = X \quad \text{and} \quad X_i := \sigma_i(X_{i-1}W_i^\intercal).$$

Let $t > 0$ be a free parameter and let w denote all parameters across all layers; the bulk of the proof will show (by induction on layers) that

$$\mathbb{E}\sup_w \exp(t\|\epsilon^\intercal X_i\|) \le \mathbb{E}\, 2^i \exp(tB^i\|\epsilon^\intercal X_0\|).$$

To see how to complete the proof from here, note by the earlier "base case lemma" (setting

$\mu := \mathbb{E}\,\|X_0^\intercal \epsilon\|$ for convenience) and Jensen's inequality that

$$\begin{aligned}
\mathrm{URad}(\mathcal{F}_{|S}) &= \mathbb{E}\sup_w \epsilon^\intercal X_L = \mathbb{E}\,\frac{1}{t}\ln\sup_w \exp\left(t\epsilon^\intercal X_L\right) \\
&\le \frac{1}{t}\ln \mathbb{E}\sup_w \exp\left(t|\epsilon^\intercal X_L|\right) \le \frac{1}{t}\ln \mathbb{E}\, 2^L \exp\left(tB^L\|\epsilon^\intercal X_0\|\right) \\
&\le \frac{1}{t}\ln \mathbb{E}\, 2^L \exp\left(tB^L(\|\epsilon^\intercal X_0\| - \mu + \mu)\right) \\
&\le \frac{1}{t}\ln\left[2^L \exp\left(t^2 B^{2L}\|X\|_{\mathrm{F}}^2/2 + tB^L\mu\right)\right] \\
&\le \frac{L\ln 2}{t} + \frac{tB^{2L}\|X\|_{\mathrm{F}}^2}{2} + B^L\|X\|_{\mathrm{F}},
\end{aligned}$$

whereby the final bound follows with the minimizing choice

$$t := \sqrt{\frac{2L\ln(2)}{B^{2L}\|X\|_{\mathrm{F}}^2}} \implies \mathrm{URad}(\mathcal{F}_{|S}) \le \sqrt{2\ln(2)LB^{2L}\|X\|_{\mathrm{F}}^2} + B^L\|X\|_{\mathrm{F}}.$$

The main inequality is now proved via induction.
For convenience, define $\sigma := \sigma_i$ and $Y := X_{i-1}$ and $V := W_i$ and $\tilde{V}$ has ℓ_2-normalized rows. By positive homogeneity and definition,

$$\begin{aligned}
\sup_w \|\epsilon^\intercal X_i\|^2 &= \sup_w \sum_j (\epsilon^\intercal \sigma(YV^\intercal)_{:j})^2 \\
&= \sup_w \sum_j (\epsilon^\intercal \sigma(YV_{j:}^\intercal))^2 \\
&= \sup_w \sum_j (\epsilon^\intercal \sigma(\|V_{j:}\| Y\tilde{V}_{j:}^\intercal))^2 \\
&= \sup_w \sum_j \|V_{j:}\|^2 (\epsilon^\intercal \sigma(Y\tilde{V}_{j:}^\intercal))^2.
\end{aligned}$$

The maximum over row norms is attained by placing all mass on a single row; thus, letting u denote an arbitrary unit norm (column) vector, and finally applying the peeling lemma, and re-introducing the dropped terms, and closing with the IH,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\epsilon} \exp\left(t\sqrt{\sup_w \|\epsilon^\intercal X_i\|^2}\right) &= \mathop{\mathbb{E}}_{\epsilon} \exp\left(t\sqrt{\sup_{w,u} B^2(\epsilon^\intercal \sigma(Yu))^2}\right) \\
&= \mathop{\mathbb{E}}_{\epsilon} \sup_{w,u} \exp\left(tB|\epsilon^\intercal \sigma(Yu)|\right) \\
&\le \mathop{\mathbb{E}}_{\epsilon} \sup_{w,u} \exp\left(tB\epsilon^\intercal \sigma(Yu)\right) + \exp\left(-tB\epsilon^\intercal \sigma(Yu)\right) \\
&\le \mathop{\mathbb{E}}_{\epsilon} \sup_{w,u} \exp\left(tB\epsilon^\intercal \sigma(Yu)\right) + \mathop{\mathbb{E}}_{\epsilon} \sup_{w,u} \exp\left(-tB\epsilon^\intercal \sigma(Yu)\right) \\
&= \mathop{\mathbb{E}}_{\epsilon} 2 \sup_{w,u} \exp\left(tB\epsilon^\intercal \sigma(Yu)\right) \\
&\le \mathop{\mathbb{E}}_{\epsilon} 2 \sup_{w,u} \exp\left(tB\epsilon^\intercal Yu\right) \\
&\le \mathop{\mathbb{E}}_{\epsilon} 2 \sup_{w} \exp\left(tB\|\epsilon^\intercal Y\|_2\right) \\
&\le \mathop{\mathbb{E}}_{\epsilon} 2^i \sup_{w} \exp\left(tB^i\|\epsilon^\intercal X_0\|_2\right).
\end{aligned}$$

15 Covering numbers

- Covering numbers are another way to do generalization. Covering numbers and Rademacher complexities are in some usual settings nearly tight with each other, though in these lectures we will only produce a way to upper bound Rademacher complexity with covering numbers.
- Covering numbers are a classical concept. The idea is we discretize or cover the function class with some finite collection of representative elements; in this way, it's tight to the "totally bounded" definition of compact set. Their first use in a statistical context is due to (Kolmogorov and Tikhomirov 1959).
- [mjt☹: i should discuss relating it to uniform convergence via rademacher, and how we have two ways, and neither is really tight, need chaining, and pointer to vershynin maybe.]

Definition 15.1 Given a set U, scale ϵ, norm $\|\cdot\|$, $V \subseteq U$ is a **(proper) cover** when

$$\sup_{a \in U} \inf_{b \in V} \|a - b\| \leq \epsilon.$$

Let $\mathcal{N}(U, \epsilon, \|\cdot\|)$ denote the **covering number**: the minimum cardinality (proper) cover.

Remark 15.1 "Improper" covers drop the requirement $V \subseteq U$. (We'll come back to this.)

Most treatments define special norms with normalization $1/n$ baked in; we'll use unnormalized Rademacher complexity and covering numbers.

Although the definition can handle directly covering functions $\mathcal{F}$, we get nice bounds by covering $\mathcal{F}_{|S}$, and conceptually it also becomes easier, just a vector (or matrix) covering problem with vector (and matrix) norms.

15.1 Basic Rademacher-covering relationship

Theorem 15.1 Given $U \subseteq \mathbb{R}^n$,

$$\mathrm{URad}(U) \leq \inf_{\alpha > 0} \left(\alpha \sqrt{n} + \left(\sup_{a \in U} \|a\|_2 \right) \sqrt{2 \ln \mathcal{N}(U, \alpha, \|\cdot\|_2)} \right).$$

Remark 15.2 $\|\cdot\|_2$ comes from applying Massart. It's unclear how to handle other norms without some technical slop.

Proof. Let $\alpha > 0$ be arbitrary, and suppose $\mathcal{N}(U, \alpha, \|\cdot\|_2) < \infty$ (otherwise bound holds trivially).

Let V denote a minimal cover, and $V(a)$ its closest element to $a \in U$. Then

$$\begin{aligned}
\mathrm{URad}(U) &= \mathbb{E} \sup_{a \in U} \langle \epsilon, a \rangle \\
&= \mathbb{E} \sup_{a \in U} \langle \epsilon, a - V(a) + V(a) \rangle \\
&= \mathbb{E} \sup_{a \in U} \left(\langle \epsilon, V(a) \rangle + \langle \epsilon, a - V(a) \rangle \right) \\
&\leq \mathbb{E} \sup_{a \in U} \left(\langle \epsilon, V(a) \rangle + \|\epsilon\| \cdot \|a - V(a)\| \right) \\
&\leq \mathrm{URad}(V) + \alpha\sqrt{n} \\
&\leq \sup_{b \in V} (\|b\|_2) \sqrt{2 \ln |V|} + \alpha\sqrt{n} \\
&\leq \sup_{a \in U} (\|a\|_2) \sqrt{2 \ln |V|} + \alpha\sqrt{n},
\end{aligned}$$

and the bound follows since $\alpha > 0$ was arbitrary.

Remark 15.3 The same proof handles improper covers with minor adjustment: for every $b \in V$, there must be $U(b) \in U$ with $\|b - U(v)\| \leq \alpha$ (otherwise, b can be moved closer to U), thus

$$\sup_{b \in V} \|b\|_2 \leq \sup_{b \in V} \|b - U(b)\|_2 + \|U(b)\|_2 \leq \alpha + \sup_{a \in U} \|a\|_2.$$

To handle other norms, superficially we need two adjustments: Cauchy-Schwarz can be replaced with Hölder, but it's unclear how to replace Massart without slop relating different norms.

15.2 Second Rademacher-covering relationship: Dudley's entropy integral

There is a classical proof that says that covering numbers and Rademacher complexities are roughly the same; the upper bound uses the Dudley entropy integral, and the lower bound uses a "Sudakov lower bound" which we will not include here.

[mjt☹: crappy commment, needs to be improved.]

- The Dudley entropy integral works at *multiple scales*.
  - Suppose we have covers $(V_N, V_{N-1}, ...)$ at scales $(\alpha_N, \alpha_N/2, \alpha_N/4, \ldots)$.
  - Given $a \in U$, choosing $V_i(a) := \arg\min_{b \in V_i} \|a - b\|$,

    $$a = (a - V_N(a)) + (V_N(a) - V_{N-1}(a)) + (V_{N-1}(a) - V_{N-2}(a)) + \cdots .$$

    We are thus rewriting a as a sequence of **increments** at different scales.
  - One way to think of it is as writing a number as its binary expansion

    $$x = (0.b_1 b_2 b_3 \ldots) = \sum_{i \geq 1} \frac{(b_i.b_{i+1} \ldots) - (0.b_{i+1} \ldots)}{2^i} = \sum_{i \geq 1} \frac{b_i}{2^i}.$$

    In the Dudley entropy integral, we are covering these **increments** b_i, rather than the number x directly.

– One can cover increments via covering numbers for the base set, and that is why these basic covering numbers appear in the Dudley entropy integral. But internally, the argument really is about these increments.

[mjt☹: Seems this works with improper covers. I should check carefully and include it in the statement or a remark.]

[mjt☹: citation for dudley? to dudley lol?]

Theorem 15.2 ***(Dudley)*** Let $U \subseteq [-1,+1]^n$ be given with $0 \in U$.

$$\begin{aligned}\operatorname{URad}(U) &\leq \inf_{N \in \mathbb{Z}_{\geq 1}} \left(n \cdot 2^{-N+1} + 6\sqrt{n} \sum_{i=1}^{N-1} 2^{-i} \sqrt{\ln \mathcal{N}(U, 2^{-i}\sqrt{n}, \|\cdot\|_2)} \right) \\ &\leq \inf_{\alpha > 0} \left(4\alpha\sqrt{n} + 12 \int_{\alpha}^{\sqrt{n}/2} \sqrt{\ln \mathcal{N}(U, \beta, \|\cdot\|_2} \mathrm{d}\beta \right).\end{aligned}$$

Proof. We'll do the discrete sum first. The integral follows by relating an integral to its Riemann sum.

- Let $N \geq 1$ be arbitrary.
- For $i \in \{1, \ldots, N\}$, define scales $\alpha_i := \sqrt{n} 2^{1-i}$.
- Define cover $V_1 := \{0\}$; since $U \subseteq [-1,+1]^n$, this is a minimal cover at scale $\sqrt{n} = \alpha_1$.
- Let V_i for $i \in \{2, \ldots, N\}$ denote any minimal cover at scale α_i, meaning $|V_i| = \mathcal{N}(U, \alpha_i, \|\cdot\|_2)$.

Since $U \ni a = (a - V_N(a)) + \sum_{i=1}^{N-1} (V_{i+1}(a) - V_i(a)) + V_1(a)$,

$$\begin{aligned}&\operatorname{URad}(U) \\ &= \mathbb{E} \sup_{a \in U} \langle \epsilon, a \rangle \\ &= \mathbb{E} \sup_{a \in U} \left(\langle \epsilon, a - V_N(a) \rangle + \sum_{i=1}^{N-1} \langle \epsilon, V_{i+1}(a) - V_i(a) \rangle + \langle \epsilon, V_1(a) \rangle \right) \\ &\leq \mathbb{E} \sup_{a \in U} \langle \epsilon, a - V_N(a) \rangle \\ &\quad + \sum_{i=1}^{N-1} \mathbb{E} \sup_{a \in U} \langle \epsilon, V_{i+1} - V_i(a) \rangle \\ &\quad + \mathbb{E} \sup_{a \in U} \langle \epsilon, V_1(a) \rangle .\end{aligned}$$

Let's now control these terms separately.
The first and last terms are easy:

$$\begin{aligned}\mathbb{E} \sup_{a \in U} \epsilon V_1(a) = \mathbb{E} \langle \epsilon, 0 \rangle = 0, \\ \mathbb{E} \sup_{a \in U} \langle \epsilon, a - V_N(a) \rangle \leq \mathbb{E} \sup_{a \in U} \|\epsilon\| \|a - V_N(a)\| \leq \sqrt{n} \alpha_N = n 2^{1-N}.\end{aligned}$$

For the middle term, define **increment class** $W_i := \{V_{i+1}(a) - V_i(a) : a \in U\}$, whereby

$|W_i| \le |V_{i+1}| \cdot |V_i| \le |V_{i+1}|^2$, and

$$\begin{aligned}
\mathbb{E} \sup_{a \in U} \langle \epsilon, V_{i+1}(a) - V_i(a) \rangle &= \mathrm{URad}(W_i) \\
&\le \left(\sup_{w \in W_i} \|w\|_2 \right) \sqrt{2 \ln |W_i|} \le \left(\sup_{w \in W_i} \|w\|_2 \right) \sqrt{4 \ln |V_{i+1}|}, \\
\sup_{w \in W_i} \|w\| &\le \sup_{a \in U} \|V_{i+1}\| + \|a - V_i(a)\| \le \alpha_{i+1} + \alpha_i = 3\alpha_{i+1}.
\end{aligned}$$

Combining these bounds,

$$\mathrm{URad}(U) \le n 2^{1-N} + 0 + \sum_{i=1}^{N} 6\sqrt{n} 2^{-i} \sqrt{\ln \mathcal{N}(U, 2^{-i}\sqrt{n}, \|\cdot\|_2}.$$

$N \ge 1$ was arbitrary, so applying $\inf_{N \ge 1}$ gives the first bound.
Since $\ln \mathcal{N}(U, \beta, \|\cdot\|_2)$ is nonincreasing in β, the integral upper bounds the Riemann sum:

$$\begin{aligned}
\mathrm{URad}(U) &\le n 2^{1-N} + 6 \sum_{i=1}^{N-1} \alpha_{i+1} \sqrt{\ln \mathcal{N}(U, \alpha_{i+1}, \|\cdot\|)} \\
&= n 2^{1-N} + 12 \sum_{i=1}^{N-1} (\alpha_{i+1} - \alpha_{i+2}) \sqrt{\ln \mathcal{N}(U, \alpha_{i+1}, \|\cdot\|)} \\
&\le \sqrt{n}\alpha_N + 12 \int_{\alpha_{N+1}}^{\alpha_2} \sqrt{\ln \mathcal{N}(U, \alpha_{i+1}, \|\cdot\|)} \mathrm{d}\beta.
\end{aligned}$$

To finish, pick $\alpha > 0$ and N with

$$\alpha_{N+1} \ge \alpha > \alpha_{N+2} = \frac{\alpha_{N+1}}{2} = \frac{\alpha_{N+2}}{4},$$

whereby

$$\begin{aligned}
\mathrm{URad}(U) &\le \sqrt{n}\alpha_N + 12 \int_{\alpha_{N+1}}^{\alpha_2} \sqrt{\ln \mathcal{N}(U, \alpha_{i+1}, \|\cdot\|)} \mathrm{d}\beta \\
&\le 4\sqrt{n}\alpha + 12 \int_{\alpha}^{\sqrt{n}/2} \sqrt{\ln \mathcal{N}(U, \alpha_{i+1}, \|\cdot\|)} \mathrm{d}\beta.
\end{aligned}$$

Remark 15.4 Tightness of Dudley: Sudakov's lower bound says there exists a universal C with

$$\mathrm{URad}(U) \ge \frac{c}{\ln(n)} \sup_{\alpha > 0} \alpha \sqrt{\ln \mathcal{N}(U, \alpha, \|\cdot\|)},$$

which implies $\mathrm{URad}(U) = \widetilde{\Theta}$ (Dudley entropy integral). [mjt☹: needs references, detail, explanation.]

Taking the notion of increments to heart and generalizing the proof gives the concept of **chaining**. One key question there is tightening the relationship with Rademacher complexity (shrinking constants and log factors in the above bound).

Another term for covering is "metric entropy."

Recall once again that we drop the normalization $1/n$ from URad and the choice of norm when covering.

16 Two deep network covering number bounds

We will give two generalization bounds.

- The first will be for arbitrary Lipschitz functions, and will be horifically loose (exponential in dimension).
- The second will be, afaik, the tightest known bound for ReLU networks.

16.1 First covering number bound: Lipschitz functions

This bound is intended as a point of contrast with our deep network generalization bounds.

Theorem 16.1 Let data $S = (x_1, \dots, x_n)$ be given with $R := \max_{i,j} \|x_i - x_j\|_\infty$. Let $\mathcal{F}$ denote all ρ-Lipschitz functions from $[-R, +R]^d \to [-B, +B]$ (where Lipschitz is measured wrt $\|\cdot\|_\infty$). Then the **improper** covering number $\widetilde{\mathcal{N}}$ satisfies

$$\ln \widetilde{\mathcal{N}}(\mathcal{F}, \epsilon, \|\cdot\|_{\mathrm{u}}) \leq \max\left\{0, \left\lceil \frac{4\rho(R+\epsilon)}{\epsilon} \right\rceil^d \ln \left\lceil \frac{2B}{\epsilon} \right\rceil \right\}.$$

Remark 16.1 Exponential in dimension!

Revisiting the "point of contrast" comment above, our deep network generalization bounds are polynomial and not exponential in dimension; consequently, we really are doing much better than simply treating the networks as arbitrary Lipschitz functions.

Proof.

- Suppose $B > \epsilon$, otherwise can use the trivial cover $\{x \mapsto 0\}$.
- Subdivide $[-R-\epsilon, +R+\epsilon]^d$ into $\left(\frac{4(R+\epsilon)\rho}{\epsilon}\right)^d$ cubes of side length $\frac{\epsilon}{2\rho}$; call this U.
- Subdivide $[-B, +B]$ into intervals of length ϵ, thus $2B/\epsilon$ elements; call this V.
- Our candidate cover $\mathcal{G}$ is the set of all piecewise constant maps from $[-R-\epsilon, +R+\epsilon]^d$ to $[-B, +B]$ discretized according to U and V, meaning

$$|\mathcal{G}| \leq \left\lceil \frac{2B}{\epsilon} \right\rceil^{\left\lceil \frac{4(R+\epsilon)\rho}{\epsilon} \right\rceil^d}.$$

To show this is an improper cover, given $f \in \mathcal{F}$, choose $g \in \mathcal{G}$ by proceeding over each $C \in U$, and assigning $g_{|C} \in V$ to be the closest element to $f(x_C)$, where x_C is the midpoint of C. Then

$$\begin{aligned}
\|f - g\|_{\mathrm{u}} &= \sup_{C \in U} \sup_{x \in C} |f(x) - g(x)| \\
&\leq \sup_{C \in U} \sup_{x \in C} \left(|f(x) - f(x_C)| + |f(x_C) - g(x)|\right) \\
&\leq \sup_{C \in U} \sup_{x \in C} \left(\rho \|x - x_C\|_\infty + \frac{\epsilon}{2}\right) \\
&\leq \sup_{C \in U} \sup_{x \in C} \left(\rho(\epsilon/(4\rho)) + \frac{\epsilon}{2}\right) \leq \epsilon
\end{aligned}$$

[mjt☹: hmm the proof used uniform norm… is it defined?]

16.2 "Spectrally-normalized" covering number bound

Theorem 16.2 ***(P. Bartlett, Foster, and Telgarsky (2017))*** Fix *multivariate* activations $(\sigma_i)_{i=1}^L$ with $\|\sigma\|_{\text{Lip}} =: \rho_i$ and $\sigma_i(0) = 0$, and data $X \in \mathbb{R}^{n\times d}$, and define

$$\mathcal{F}_n := \left\{ \sigma_L(W_L\sigma_{L-1}\cdots\sigma_1(W_1X^\intercal)\cdots) \ : \ \|W_i^\intercal\|_2 \le s_i, \|W_i^\intercal\|_{2,1} \le b_i \right\},$$

and all matrix dimensions are at most m. Then

$$\ln \mathcal{N}\left(\mathcal{F}_n, \epsilon, \|\cdot\|_{\text{F}}\right) \le \frac{\|X\|_{\text{F}}^2 \prod_{j=1}^L \rho_j^2 s_j^2}{\epsilon^2} \left(\sum_{i=1}^L \left(\frac{b_i}{s_i}\right)^{2/3}\right)^3 \ln(2m^2).$$

Remark 16.2 Applying Dudley,

$$\text{URad}(\mathcal{F}_n) = \widetilde{\mathcal{O}}\left(\|X\|_{\text{F}} \left[\prod_{j=1}^L \rho_j s_j\right] \cdot \left[\sum_{i=1}^L \left(\frac{b_i}{s_i}\right)^{2/3}\right]^{3/2}\right).$$

[mjt☹: that's annoying and should be included/performed rigorously.]

Proof uses $\|\sigma(M) - \sigma(M')\|_{\text{F}} \le \|\sigma\|_{\text{Lip}} \cdot \|M - M'\|_{\text{F}}$; in particular, it allows multi-variate gates like max-pooling! See (P. Bartlett, Foster, and Telgarsky 2017) for $\|\sigma_i\|_{\text{Lip}}$ estimates.

This proof can be adjusted to handle "distance to initialization"; see (P. Bartlett, Foster, and Telgarsky 2017) and the notion "reference matrices."

Let's compare to our best "layer peeling" proof from before, which had $\prod_i \|W_i\|_{\text{F}} \lesssim m^{L/2} \prod_i \|W_i\|_2$. That proof assumed $\rho_i = 1$, so the comparison boils down to

$$m^{L/2}\left(\prod_i \|W_i\|_2\right) \quad \text{vs.} \quad \left[\sum_i \left(\frac{\|W_i^\intercal\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right)\right]^{3/2} \left(\prod_i \|W_i\|_2\right),$$

where $L \le \sum_i \left(\frac{\|W_i^\intercal\|_{2,1}^{2/3}}{\|W_i\|_2^{2/3}}\right) \le Lm^{2/3}$. So the bound is better but still leaves a lot to be desired, and is loose in practice.

It is not clear how to prove exactly this bound with Rademacher peeling, which is a little eerie (independent of whether this bound is good or not).

The proof, as with Rademacher peeling proofs, is an induction on layers, similarly one which does not "coordinate" the behavior of the layers; this is one source of looseness.

Remark 16.3 ***(practical regularization schemes)*** This bound suggests regularization based primarily on the Lipschitz constant of the network; similar ideas appeared in parallel applied work, both for classification problems (Cisse et al. 2017), and for GANs (Arjovsky, Chintala, and Bottou 2017).

Remark 16.4 ***(another proof)*** For an alternate proof a similar fact (albeit requiring univariate gates), see (Neyshabur, Bhojanapalli, and Srebro 2018).

The first step of the proof is a covering number for individual layers.

Lemma 16.1

$$\ln \mathcal{N}(\{WX^\intercal : X \in \mathbb{R}^{m\times d}, \|W^\intercal\|_{2,1} \le b\}, \epsilon, \|\cdot\|_{\mathrm{F}}) \le \left\lceil \frac{\|X\|_{\mathrm{F}}^2 b^2}{\epsilon^2} \right\rceil \ln(2dm).$$

Proof. Let $W \in \mathbb{R}^{m\times d}$ be given with $\|W^\intercal\|_{2,1} \le r$. Define $s_{ij} := W_{ij}/|W_{ij}|$, and note

$$WX^\intercal = \sum_{i,j} \mathbf{e}_i\mathbf{e}_i^\intercal W \mathbf{e}_j\mathbf{e}_j^\intercal X^\intercal = \sum_{i,j} \mathbf{e}_i W_{ij} (X\mathbf{e}_j)^\intercal = \sum_{i,j} \underbrace{\frac{|W_{ij}|\|X\mathbf{e}_j\|_2}{r\|X\|_{\mathrm{F}}}}_{=:q_{ij}} \underbrace{\frac{r\|X\|_{\mathrm{F}} s_{ij}\mathbf{e}_i(X\mathbf{e}_j)^\intercal}{\|X\mathbf{e}_j\|}}_{=:U_{ij}}.$$

Note by Cauchy-Schwarz that

$$\sum_{i,j} q_{ij} \le \frac{1}{r\|X\|_{\mathrm{F}}} \sum_i \sqrt{\sum_j W_{ij}^2}\|X\|_{\mathrm{F}} = \frac{\|W^\intercal\|_{2,1}\|X\|_{\mathrm{F}}}{r\|X\|_{\mathrm{F}}} \le 1,$$

potentially with strict inequality, thus q is not a probability vector, which we will want later. To remedy this, construct probability vector p from q by adding in, with equal weight, some U_{ij} and its negation, so that the above summation form of $WX^\intercal$ goes through equally with p as with q. Now define IID random variables $(V_1, \dots, V_k)$, where

$$\begin{aligned}
\Pr[V_l = U_{ij}] &= p_{ij}, \\
\mathbb{E}\, V_l &= \sum_{i,j} p_{ij} U_{ij} = \sum_{i,j} q_{ij} U_{ij} = WX^\intercal, \\
\|U_{ij}\| &= \left\| \frac{s_{ij}\mathbf{e}_i(X\mathbf{e}_j)}{\|X\mathbf{e}_j\|_2} \right\|_{\mathrm{F}} \cdot r\|X\|_{\mathrm{F}} = |s_{ij}| \cdot \|\mathbf{e}_i\|_2 \cdot \left\| \frac{X\mathbf{e}_j}{\|X\mathbf{e}_j\|_2} \right\|_2 \cdot r\|X\|_{\mathrm{F}} = r\|X\|_{\mathrm{F}}, \\
\mathbb{E}\,\|V_l\|^2 &= \sum_{i,j} p_{ij}\|U_{ij}\|^2 \le \sum_{ij} p_{ij} r^2 \|X\|_{\mathrm{F}}^2 = r^2\|X\|_{\mathrm{F}}^2.
\end{aligned}$$

By +Lemma 3.1 (Maurey (Pisier 1980)), there exist $(\hat{V}_1, \dots, \hat{V}_k) \in S^k$ with

$$\left\| WX^\intercal - \frac{1}{k}\sum_l \hat{V}_l \right\|^2 \le \mathbb{E} \left\| \mathbb{E}\, V_1 - \frac{1}{k}\sum_l V_l \right\|^2 \le \frac{1}{k}\,\mathbb{E}\,\|V_1\|^2 \le \frac{r^2\|X\|_{\mathrm{F}}^2}{k}.$$

Furthermore, the matrices $\hat{V}_l$ have the form

$$\frac{1}{k}\sum_l \hat{V}_l = \frac{1}{k}\sum_l \frac{s_l \mathbf{e}_{i_l}(X\mathbf{e}_{j_l})^\intercal}{\|X\mathbf{e}_{j_l}\|} = \left[\frac{1}{k}\sum_l \frac{s_l \mathbf{e}_{i_l}\mathbf{e}_{j_l}^\intercal}{\|X\mathbf{e}_{j_l}\|}\right] X^\intercal;$$

by this form, there are at most $(2nd)^k$ choices for $(\hat{V}_1, \dots, \hat{V}_k)$.

Lemma 16.2 Let $\mathcal{F}_n$ be the same image vectors as in the theorem, and let per-layer tolerances $(\epsilon_1, \dots, \epsilon_L)$ be given. then

$$\ln \mathcal{N}\left(\mathcal{F}_n,\ \sum_{j=1}^L \rho_j \epsilon_j \prod_{k=j+1}^L \rho_k s_k,\ \|\cdot\|_{\mathrm{F}} \right) \le \sum_{i=1}^L \left\lceil \frac{\|X\|_{\mathrm{F}}^2 b_i^2 \prod_{j<i} \rho_j^2 s_j^2}{\epsilon_i^2} \right\rceil \ln(2m^2).$$

Proof. Let X_i denote the output of layer i of the network, using weights $(W_i, \dots, W_1)$, meaning

$$X_0 := X \qquad \text{and} \qquad X_i := \sigma_i(X_{i-1}W_i^\mathsf{T}).$$

The proof recursively constructs cover elements $\hat{X}_i$ and weights $\hat{W}_i$ for each layer with the following basic properties.

- Define $\hat{X}_0 := X_0$, and $\hat{X}_i := \Pi_{B_i}\sigma_i(\hat{X}_{i-1}\hat{W}_i^\mathsf{T})$, where B_i is the Frobenius-norm ball of radius $\|X\|_\mathrm{F}\prod_{j<i}\rho_j s_j$.
- Due to the projection Π_{B_i}, $\|\hat{X}_i\|_\mathrm{F} \le \|X\|_\mathrm{F}\prod_{j\le i}\rho_j s_j$. Similarly, using $\rho_i(0) = 0$, $\|X_i\|_\mathrm{F} \le \|X\|_\mathrm{F}\prod_{j<i}\rho_j s_j$.
- Given $\hat{X}_{i-1}$, choose $\hat{W}_i$ via +Lemma 16.1 so that $\|\hat{X}_{i-1}W_i^\mathsf{T} - \hat{X}_{i-1}\hat{W}_i^\mathsf{T}\|_\mathrm{F} \le \epsilon_i$, whereby the corresponding covering number $\mathcal{N}_i$ for this layer satisfies

$$\ln \mathcal{N}_i \le \left\lceil \frac{\|\hat{X}_{i-1}\|_\mathrm{F}^2 b_i^2}{\epsilon_i^2} \right\rceil \ln(2m^2) \le \left\lceil \frac{\|X\|_\mathrm{F}^2 b_i^2 \prod_{j<i}\rho_j^2 s_j^2}{\epsilon_i^2} \right\rceil \ln(2m^2).$$

- Since each cover element $\hat{X}_i$ depends on the full tuple $(\hat{W}_i, \dots, \hat{W}_1)$, the final cover is the product of the individual covers (and not their union), and the final cover log cardinality is upper bounded by

$$\ln \prod_{i=1}^{L} \mathcal{N}_i \le \sum_{i=1}^{L} \left\lceil \frac{\|X\|_\mathrm{F}^2 b_i^2 \prod_{j<i}\rho_j^2 s_j^2}{\epsilon_i^2} \right\rceil \ln(2m^2).$$

It remains to prove, by induction, an error guarantee

$$\|X_i - \hat{X}_i\|_\mathrm{F} \le \sum_{j=1}^{i} \rho_j \epsilon_j \prod_{k=j+1}^{i} \rho_k s_k.$$

The base case $\|X_0 - \hat{X}_0\|_\mathrm{F} = 0 = \epsilon_0$ holds directly. For the inductive step, by the above ingredients and the triangle inequality,

$$\begin{aligned}
\|X_i - \hat{X}_i\|_\mathrm{F} &\le \rho_i \|X_{i-1}W_i^\mathsf{T} - \hat{X}_{i-1}\hat{W}_i^\mathsf{T}\|_\mathrm{F} \\
&\le \rho_i \|X_{i-1}W_i^\mathsf{T} - \hat{X}_{i-1}W_i^\mathsf{T}\|_\mathrm{F} + \rho_i \|\hat{X}_{i-1}W_i^\mathsf{T} - \hat{X}_{i-1}\hat{W}_i^\mathsf{T}\|_\mathrm{F} \\
&\le \rho_i s_i \|X_{i-1} - \hat{X}_{i-1}\|_\mathrm{F} + \rho_i \epsilon_i \\
&\le \rho_i s_i \left[\sum_{j=1}^{i-1} \rho_j \epsilon_j \prod_{k=j+1}^{i-1} \rho_k s_k \right] + \rho_i \epsilon_i \\
&= \left[\sum_{j=1}^{i-1} \rho_j \epsilon_j \prod_{k=j+1}^{i} \rho_k s_k \right] + \rho_i \epsilon_i \\
&= \sum_{j=1}^{i} \rho_j \epsilon_j \prod_{k=j+1}^{i} \rho_k s_k.
\end{aligned}$$

Proof of +Theorem 16.2 (P. Bartlett, Foster, and Telgarsky (2017)). By solving a

Lagrangian (minimize cover size subject to total error $\leq \epsilon$), choose

$$\epsilon_i := \frac{\alpha_i \epsilon}{\rho_i \prod_{j>i} \rho_j s_j}, \qquad \alpha_i := \frac{1}{\beta} \left(\frac{b_i}{s_i}\right)^{2/3}, \qquad \beta := \sum_{i=1}^{L} \left(\frac{b_i}{s_i}\right)^{2/3}.$$

Invoking the induction lemma with these choices, the resulting cover error is

$$\sum_{i=1}^{L} \epsilon_i \rho_i \prod_{j>i} \rho_j s_j = \epsilon \sum_{j=1}^{L} \alpha_i = \epsilon.$$

and the main term of the cardinality (ignoring $\ln(2m^2)$) satisfies

$$\begin{aligned} \sum_{i=1}^{L} \frac{\|X\|_{\mathrm{F}}^2 b_i^2 \prod_{j<i} \rho_j^2 s_j^2}{\epsilon_i^2} &= \frac{\|X\|_{\mathrm{F}}^2}{\epsilon^2} \sum_{i=1}^{L} \frac{b_i^2 \prod_{j=1}^{L} \rho_j^2 s_j^2}{\alpha_i^2 s_i^2} \\ &= \frac{\|X\|_{\mathrm{F}}^2 \prod_{j=1}^{L} \rho_j^2 s_j^2}{\epsilon^2} \sum_{i=1}^{L} \frac{\beta^2 b_i^{2/3}}{s_i^{2/3}} = \frac{\|X\|_{\mathrm{F}}^2 \prod_{j=1}^{L} \rho_j^2 s_j^2}{\epsilon^2} \left(\sum_{i=1}^{L} \left(\frac{b_i}{s_i}\right)^{2/3}\right)^3. \end{aligned}$$

[mjt☹: I should include the Lagrangian explicitly and also explicit Dudley.]

17 VC dimension

[mjt☹: ok if VC dim is one section, covering numbers should be as well?]

[mjt☹: remainder is copy/pasted from fall2018, was not taught in fall2019.]

[mjt☹: should include in preamble various bounds not taught, and a comment that VC dim proofs are interesting and reveal structure not captured above.]

- VC dimension is an ancient generalization technique; essentially the quantity itself appears in the work of Kolmogorov, and was later rediscovered a few times, and named after Vapnik and Chervonenkis, whose used it for generalization.
- To prove generalization, we will upper bound Rademacher complexity with VC dimension; classical VC dimension generalization proofs include Rademacher averages.
- There is some huge ongoing battle of whether VC dimension is a good measure or not. I think the proofs are interesting and are sensitive to interesting properties of deep networks in ways not capture by many of the bounds we spent time on. Anyway, a discussion for another time...
  - As stated the bounds are worst-case-y; I feel they could be adapted into more average-case-y bounds, though this has not been done yet...

First, some definitions. First, the zero-one/classification risk/error:

$$\mathcal{R}_{\mathrm{z}}(\mathrm{sgn}(f)) = \Pr[\mathrm{sgn}(f(X)) \neq Y], \ \widehat{\mathcal{R}}_{\mathrm{z}}(\mathrm{sgn}(f)) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\mathrm{sgn}(f(x_i)) \neq y_i].$$

The earlier Rademacher bound will now have

$$\mathrm{URad}\left(\{(x,y) \mapsto \mathbf{1}[\mathrm{sgn}(f(x)) \neq y] : f \in \mathcal{F}\}_{|S}\right).$$

This is at most 2^n; we'll reduce it to a combinatorial quantity:

$$\begin{aligned}
\mathrm{sgn}(U) &:= \{(\mathrm{sgn}(u_1), \ldots, \mathrm{sgn}(u_n)) : u \in V\}, \\
\mathrm{Sh}(\mathcal{F}_{|S}) &:= \left|\mathrm{sgn}(\mathcal{F}_{|S})\right|, \\
\mathrm{Sh}(\mathcal{F}; n) &:= \sup_{\substack{S \in ? \\ |S| \leq n}} \left|\mathrm{sgn}(\mathcal{F}_{|S})\right|, \\
\mathrm{VC}(\mathcal{F}) &:= \sup\{i \in \mathbb{Z}_{\geq 0} : \mathrm{Sh}(\mathcal{F}; i) = 2^i\}.
\end{aligned}$$

Remark 17.1 Sh is "shatter coefficient," VC is "VC dimension."

Both quantities are criticized as being too tied to their worst case; bounds here depend on (empirical quantity!) $\mathrm{URad}(\mathrm{sgn}(\mathcal{F}_{|S}))$, which can be better, but throws out the labels.

Theorem 17.1 ***("VC Theorem")*** With probability at least $1 - \delta$, every $f \in \mathcal{F}$ satisfies

$$\mathcal{R}_z(\mathrm{sgn}(f)) \leq \widehat{\mathcal{R}}_z(\mathrm{sgn}(f) + \frac{2}{n}\mathrm{URad}(\mathrm{sgn}(\mathcal{F}_{|S})) + 3\sqrt{\frac{\ln(2/\delta)}{2n}},$$

and

$$\begin{aligned}
\mathrm{URad}(\mathrm{sgn}(\mathcal{F}_{|S})) &\leq \sqrt{2n \ln \mathrm{Sh}(\mathcal{F}_{|S})}, \\
\ln \mathrm{Sh}(\mathcal{F}_{|S}) &\leq \ln \mathrm{Sh}(\mathcal{F}; n) \leq \mathrm{VC}(\mathcal{F}) \ln(n+1).
\end{aligned}$$

Remark 17.2 [mjt☹: Say something like "Need $\mathrm{Sh}(\mathcal{F}_{|s}) = o(n)$"in order to learn"." ?]

Minimizing $\widehat{\mathcal{R}}_z$ is NP-hard in many trivial cases, but those require noise and neural networks can often get $\widehat{\mathcal{R}}_z(\mathrm{sgn}(f)) = 0$.

$\mathrm{VC}(\mathcal{F}) < \infty$ suffices; many considered this a conceptual breakthrough, namely "learning is possible!"

The quantities (VC, Sh) appeared in prior work (not by V-C). Symmetrization apparently too, though I haven't dug this up.

First step of proof: pull out the zero-one loss.

Lemma 17.1 $\mathrm{URad}(\{(x,y) \mapsto \mathbf{1}[\mathrm{sgn}(f(x)) \neq y] : f \in \mathcal{F}\}_{|S}) \leq \mathrm{URad}(\mathrm{sgn}(\mathcal{F}_{|S}))$.

Proof. For each i, define

$$\ell_i(z) := \max\left\{0, \min\left\{1, \frac{1 - y_i(2z-1)}{2}\right\}\right\},$$

which is 1-Lipschitz, and satisfies

$$\ell_i(\mathrm{sgn}(f(x_i))) = \mathbf{1}[\mathrm{sgn}(f(x_i)) \neq y_i].$$

(Indeed, it is the linear interpolation.) Then

$$\begin{aligned}
&\mathrm{URad}(\{(x,y) \mapsto \mathbf{1}[\mathrm{sgn}(f(x)) \neq y] : f \in \mathcal{F}\}_{|S}) \\
&= \mathrm{URad}(\{(\ell_1(\mathrm{sgn}(f(x_1))), \dots, \ell_n(\mathrm{sgn}(f(x_n)))) : f \in \mathcal{F}\}_{|S}) \\
&= \mathrm{URad}(\ell \circ \mathrm{sgn}(\mathcal{F})_{|S}) \\
&\leq \mathrm{URad}(\mathrm{sgn}(\mathcal{F})_{|S}).
\end{aligned}$$

[mjt☹: is that using the fancier per-coordinate vector-wise peeling again?]

Plugging this into our Rademacher bound: w/ pr $\geq 1-\delta$, $\forall f \in \mathcal{F}$,

$$\mathcal{R}_z(\mathrm{sgn}(f)) \leq \widehat{\mathcal{R}}_z(\mathrm{sgn}(f)) + \frac{2}{n}\mathrm{URad}(\mathrm{sgn}(\mathcal{F})_{|S}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

The next step is to apply Massart's finite lemma, giving

$$\mathrm{URad}(\mathrm{sgn}(\mathcal{F}_{|S})) \leq \sqrt{2n\mathrm{Sh}(\mathcal{F}_{|S})}.$$

One last lemma remains for the proof.

[mjt☹: lol why mention warren. should be explicit and not passive-aggressive.]

Lemma 17.2 ***(Sauer-Shelah? Vapnik-Chervonenkis? Warren? . . .)*** Let $\mathcal{F}$ be given, and define $V := \mathrm{VC}(\mathcal{F})$. Then

$$\mathrm{Sh}(\mathcal{F}; n) \leq \begin{cases} 2^n & \text{when } n \leq V, \\ \left(\frac{en}{V}\right)^V & \text{otherwise.} \end{cases}$$

Moreover, $\mathrm{Sh}(\mathcal{F}; n) \leq n^V + 1$.

(**Proof.** Omitted. Exists in many standard texts.)

[mjt☹: okay fine but i should give a reference, and eventually my own clean proof.]

17.1 VC dimension of linear predictors

Theorem 17.2 Define $\mathcal{F} := \left\{x \mapsto \mathrm{sgn}(\langle a, x\rangle - b) : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$ ("linear classifiers"/"affine classifier"/ "linear threshold function (LTF)"). Then $\mathrm{VC}(\mathcal{F}) = d + 1$.

Remark 17.3 By Sauer-Shelah, $\mathrm{Sh}(\mathcal{F}; n) \leq n^{d+1} + 1$. Anthony-Bartlett chapter 3 gives an exact equality; only changes constants of $\ln \mathrm{VC}(\mathcal{F}; n)$.

Let's compare to Rademacher:

$$\begin{aligned}
\mathrm{URad}(\mathrm{sgn}(\mathcal{F}_{|S})) &\leq \sqrt{2nd\ln(n+1)}, \\
\mathrm{URad}(\{x \mapsto \langle w, x\rangle : \|w\| \leq R\}_{|S}) &\leq R\|X_S\|_F,
\end{aligned}$$

where $\|X_S\|_F^2 = \sum_{x \in S} \|x\|_2^2 \leq n \cdot d \cdot \max_{i,j} x_{i,j}$. One is scale-sensitive (and suggests regularization schemes), other is scale-insensitive.

Proof. First let's do the lower bound $\mathrm{VC}(\mathcal{F}) \geq d+1$.

- Suffices to show $\exists S := \{x_1, \ldots, x_{d+1}\}$ with $\mathrm{Sh}(\mathcal{F}_{|S}) = 2^{d+1}$.
- Choose $S := \{\mathbf{e}_1, \ldots, \mathbf{e}_d, (0, \ldots, 0)\}$.

Given any $P \subseteq S$, define (a, b) as

$$a_i := 2 \cdot \mathbf{1}[\mathbf{e}_i \in P] - 1, \qquad b := \frac{1}{2} - \mathbf{1}[0 \in P].$$

Then

$$\mathrm{sgn}(\langle a, \mathbf{e}_i \rangle - b) = \mathrm{sgn}(2\mathbf{1}[\mathbf{e}_i \in P] - 1 - b) = 2\mathbf{1}[\mathbf{e}_i \in P] - 1,$$
$$\mathrm{sgn}(\langle a, 0 \rangle - b) = \mathrm{sgn}(2\mathbf{1}[0 \in P] - 1/2) = 2\mathbf{1}[0 \in P] - 1,$$

meaning this affine classifier labels S according to P, which was an arbitrary subset.
Now let's do the upper bound $\mathrm{VC}(\mathcal{F}) < d+2$.

- Consider any $S \subseteq \mathbb{R}^d$ with $|S| = d+2$.
- By *Radon's Lemma* (proved next), there exists a partition of S into nonempty (P, N) with $\mathrm{conv}(P) \cap \mathrm{conv}(N)$.
- Label P as positive and N as negative. Given any affine classifier, it can not be correct on all of S (and thus $\mathrm{VC}(\mathcal{F}) < d+2$): either it is incorrect on some of P, or else it is correct on P, and thus has a piece of $\mathrm{conv}(N)$ and thus $x \in N$ labeled positive.

[mjt☹: needs ref]

Lemma 17.3 ***(Radon's Lemma)*** Given $S \subseteq \mathbb{R}^d$ with $|S| = d+2$, there exists a partition of S into nonempty (P, N) with $\mathrm{conv}(P) \cap \mathrm{conv}(S) \neq \emptyset$.

Proof. Let $S = \{x_1, \ldots, x_{d+2}\}$ be given, and define $\{u_1, \ldots, u_{d+1}\}$ as $u_i := x_i - x_{d+2}$, which must be linearly dependent:

- Exist scalars $(\alpha_1, \ldots, \alpha_{d+1})$ and a j with $\alpha_j := -1$ so that

$$\sum_i \alpha_i u_i = -u_j + \sum_{i \neq j} \alpha_i u_i = 0;$$

- thus $x_j - x_{d+2} = \sum_{\substack{i \neq j \\ i < d+2}} \alpha_i (x_i - x_{d+2})$ and $0 = \sum_{i<d+2} \alpha_i x_i - x_{d+2} \sum_{i<d+2} \alpha_i =: \sum_j \beta_j x_j$, where $\sum_j \beta_j = 0$ and not all β_j are zero.

Set $P := \{i : \beta_i > 0\}$, $N := \{i : \beta_i \leq 0\}$; where neither set is empty.
Set $\beta := \sum_{i \in P} \beta_i - \sum_{i \in N} \beta_i > 0$.
Since $0 = \sum_i \beta_i x_i = \sum_{i \in P} \beta_i x_i + \sum_{i \in N} \beta_i x_i$, then

$$\frac{0}{\beta} = \sum_{i \in P} \frac{\beta_i}{\beta} x_i + \sum_{i \in N} \frac{\beta_i}{\beta} x_i$$

and the point $z := \sum_{i \in P} \beta_i x_i / \beta = \sum_{i \in N} \beta_i x_i / (-\beta)$ satisfies $z \in \mathrm{conv}(P) \cap \mathrm{conv}(N)$.

Remark 17.4 Generalizes Minsky-Papert "xor" construction.

Indeed, the first appearance I know of shattering/VC was in approximation theory, the papers of Warren and Shapiro, and perhaps it is somewhere in Kolmogorov's old papers.

17.2 VC dimension of threshold networks

Consider iterating the previous construction, giving an "LTF network": a neural network with activation $z \mapsto \mathbf{1}[z \geq 0]$.

We'll analyze this by studying output of all nodes. To analyze this, we'll study not just the outputs, but the behavior of all nodes.

[mjt☹: another suggestion of definition in pandoc-numbering]

Definition.

- Given a sample S of size n and an LTF network with m nodes (in any topologically sorted order), define activation matrix $A := \mathrm{Act}(S; W := (a_1, \ldots, a_m))$ where A_{ij} is the output of node j on input i, with fixed network weights W.
- Let $\mathrm{Act}(S; \mathcal{F})$ denote the set of activation matrices with architecture fixed and weights W varying.

Remark 17.5 Since last column is the labeling, $|\mathrm{Act}(S; \mathcal{F})| \geq \mathrm{Sh}(\mathcal{F}_{|S})$.

Act seems a nice complexity measure, but it is hard to estimate given a single run of an algorithm (say, unlike a Lipschitz constant).

We'll generalize Act to analyze ReLU networks.

Theorem 17.3 For any LTF architecture $\mathcal{F}$ with p parameters,

$$\mathrm{Sh}(\mathcal{F}; n) \leq |\mathrm{Act}(S; \mathcal{F})| \leq (n+1)^p.$$

When $p \geq 12$, then $\mathrm{VC}(\mathcal{F}) \leq 6p \ln(p)$.

Proof.

- Topologically sort nodes, let $(p_1, \ldots, p_m)$ denote numbers of respective numbers of parameters (thus $\sum_i p_i = p$).
- Proof will iteratively construct sets $(U_1, \ldots, U_m)$ where U_i partitions the weight space of nodes $j \leq i$ so that, within each partition cell, the activation matrix does not vary.
- The proof will show, by induction, that $|U_i| \leq (n+1)^{\sum_{j \leq i} p_j}$. This completes the proof of the first claim, since $\mathrm{Sh}(\mathcal{F}_{|S}) \leq |\mathrm{Act}(\mathcal{F}; S)| = |U_m|$.
- For convenience, define $U_0 = \{\emptyset\}$, $|U_0| = 1$; the base case is thus $|U_0| = 1 = (n+1)^0$.

(Inductive step). Let $j \geq 1$ be given; the proof will now construct U_{j+1} by refining the partition U_j.

- Fix any cell C of U_j; as these weights vary, the activation is fixed, thus the input to node $j+1$ is fixed for each $x \in S$.
- Therefore, on this augmented set of n inputs (S with columns of activations appended to each example), there are $(n+1)^{p_{j+1}}$ possible outputs via Sauer-Shelah and the VC dimension of affine classifiers with p_{j+1} inputs.
- In other words, C can be refined into $(n+1)^{p_{j+1}}$ sets; since C was arbitrary,

$$|U_{j+1}| = |U_j|(n+1)^{p_{j+1}} \leq (n+1)^{\sum_{l \leq j+1} p_l}.$$

This completes the induction and establishes the Shattering number bound.

It remains to bound the VC dimension via this Shatter bound:

$$\begin{aligned}
&\mathrm{VC}(\mathcal{F}) < n \\
\Longleftarrow &\forall i \geq n \,.\, \mathrm{Sh}(\mathcal{F}; i) < 2^i \\
\Longleftarrow &\forall i \geq n \,.\, (i+1)^p < 2^i \\
\Longleftrightarrow &\forall i \geq n \,.\, p \ln(i+1) < i \ln 2 \\
\Longleftrightarrow &\forall i \geq n \,.\, p < \frac{i \ln(2)}{\ln(i+1)} \\
\Longleftarrow &p < \frac{n \ln(2)}{\ln(n+1)}
\end{aligned}$$

If $n = 6p\ln(p)$,

$$\begin{aligned}
\frac{n\ln(2)}{\ln(n+1)} \geq \frac{n\ln(2)}{\ln(2n)} &= \frac{6p\ln(p)\ln(2)}{\ln 12 + \ln p + \ln\ln p} \\
&\geq \frac{6p \ln p \ln 2}{3 \ln p} > p.
\end{aligned}$$

Remark 17.6 Had to do handle $\forall i \geq n$ since VC dimension is defined via sup; one can define funky $\mathcal{F}$ where Sh is not monotonic in n.

Lower bound is $\Omega(p \ln m)$; see Anthony-Bartlett chapter 6 for a proof. This lower bound however is for a specific fixed architecture!

Other VC dimension bounds: ReLU networks have $\widetilde{\mathcal{O}}(pL)$, sigmoid networks have $\widetilde{\mathcal{O}}(p^2m^2)$, and there exists a convex-concave activation which is close to sigmoid but has VC dimension ∞.

Matching lower bounds exist for ReLU, not for sigmoid; but even the "matching" lower bounds are deceptive since they hold for a *fixed* architecture of a given number of parameters and layers.

17.3 VC dimension of ReLU networks

Today's ReLU networks will predict with

$$x \mapsto A_L \sigma_{L-1}\left(A_{L-1} \cdots A_2 \sigma_1(A_1 x + b_1) + b_2 \cdots + b_{L-1}\right) + b_L,$$

where $A_i \in \mathbb{R}^{d_i \times d_{i-1}}$ and $\sigma_i : \mathbb{R}^{d_i \to d_i}$ applies the ReLU $z \mapsto \max\{0, z\}$ coordinate-wise.

Convenient notation: collect data as rows of matrix $X \in \mathbb{R}^{n \times d}$, and define

$$\begin{aligned}
X_0 &:= X^\top & Z_0 &:= \text{all 1s matrix}, \\
X_i &:= A_i(Z_{i-1} \odot X_{i-1}) + b_i 1_n^\top, & X_i &:= \mathbf{1}[X_i \geq 0],
\end{aligned}$$

where $(Z_1, \ldots, Z_L)$ are the activation matrices.

[mjt☹: i should double check i have the tightest version? which is more sensitive to earlier layers? i should comment on that and the precise structure/meaning of the lower bounds?]

Theorem 17.4 ***((Theorem 6, P. L. Bartlett et al. 2017))*** Let fixed ReLU architecture $\mathcal{F}$ be given with $p = \sum_{i=1}^L p_i$ parameters, L layers, $m = \sum_{i=1}^L m_i$ nodes. Let examples $(x_1, \ldots, x_n)$

be given and collected into matrix X. There exists a partition U_L of the parameter space satisfying:

- Fix any $C \in U_L$. As parameters vary across C, activations $(Z_1, \ldots, Z_L)$ are fixed.
- $\mathrm{Sh}(\mathcal{F}; n) \leq |\{Z_L(C) : C \in U_L\}| \leq |U_L| \leq (12nL)^{pL}$, where $Z_L(C)$ denotes the sign pattern in layer L for $C \in U_L$.
- If $pL^2 \geq 72$, then $\mathrm{VC}(\mathcal{F}) \leq 6pL \ln(pL)$.

Remark 17.7 ***(on the proof)*** As with LTF networks, the prove inductively constructs partitions of the weights up through layer i so that the activations are fixed across all weights in each partition cell.

Consider a fixed cell of the partition, whereby the activations are fixed zero-one matrices. As a function of the *inputs*, the ReLU network is now *an affine function*; as a function of the *weights* it is *multilinear* or rather *a polynomial of degree* L.

Consider again a fixed cell and some layer i; thus $\sigma(X_i) = Z_i \odot X_i$ is a matrix of polynomials of degree i (in the weights). If we can upper bound the number of possible signs of $A_{i+1}(Z_i \odot X_i) + b_i 1_n^\top$, then we can refine our partition of weight space and recurse. For that we need a bound on sign patterns of polynomials, as on the next slide.

Theorem 17.5 ***(Warren '68; see also Anthony-Bartlet Theorem 8.3)*** Let F denote functions $x \mapsto f(x; w)$ which are r-degree polynomials in $w \in \mathbb{R}^p$. If $n \geq p$, then $\mathrm{Sh}(\mathcal{F}; n) \leq 2(\frac{2enr}{p})^p$.

Remark 17.8 Proof is pretty intricate, and omitted. It relates the VC dimension of F to the zero sets $Z_i := \{w \in \mathbb{R}^p : f(x; w) = 0\}$, which it controls with an application of Bezout's Theorem. The zero-counting technique is also used to obtain an exact Shatter coefficient for affine classifiers.

Proof (of ReLU VC bound).
We'll inductively construct partitions $(U_0, \ldots, U_L)$ where U_i partitions the parameters of layers $j \leq i$ so that for any $C \in U_i$, the activations Z_j in layer $j \leq i$ are fixed for all parameter choices within C (thus let $Z_j(C)$ denote these fixed activations).
The proof will proceed by induction, showing $|U_i| \leq (12nL)^{pi}$.
Base case $i = 0$: then $U_0 = \{\emptyset\}$, Z_0 is all ones, and $|U_0| = 1 \leq (12nL)^{pi}$.
(Inductive step).

- Fix $C \in S_i$ and $(Z_1, \ldots, Z_i) = (Z_1(C), \ldots, Z_i(C))$.
- Note $X_{i+1} = A_{i+1}(Z_i \odot X_i) + b_i 1_n^\top$ is polynomial (of degree $i+1$) in the parameters since $(Z_1, \ldots, Z_i)$ are fixed.
- Therefore

$$|\{\mathbf{1}[X_{i+1} \geq 0] : \text{params} \in C\}| \leq \mathrm{Sh}(i+1 \text{ deg poly}; m_i \cdot n \text{ functions})$$
$$\leq 2\left(\frac{2enm_{i+1}}{\sum_{j \leq i} p_j}\right)^{\sum_{j \leq i+1} p_j} \leq (12nL)^p.$$

[Technical comment: to apply the earlier shatter bound for polynomials, we needed $n \cdot m_{i+1} \geq \sum_j p_j$; but if (even more simply) $p \geq nm_{i+1}$, we can only have $\leq 2^{nm_{i+1}} \leq 2^p$ activation matrices anyway, so the bound still holds.]

- Therefore carving U_i into pieces according to $Z_{i+1} = \mathbf{1}[X_{i+1} \geq 0]$ being fixed gives

$$|U_{i+1}| \leq |U_i|(12nL)^p \leq (12nL)^{p(i+1)}.$$

  This completes the induction and upper bounds the number of possible activation patterns and also the shatter coefficient.

It remains to upper bound the VC dimension via the Shattering bound. As with LTF networks,

$$\begin{aligned}
\mathrm{VC}(\mathcal{F}) < n &\impliedby \forall i \geq n \centerdot \mathrm{Sh}(\mathcal{F}; i) < 2^i \\
&\impliedby \forall i \geq n \centerdot (12iL)^{pL} < 2^i \\
&\iff \forall i \geq n \centerdot pL \ln(12iL) < i \ln 2 \\
&\iff \forall i \geq n \centerdot pL < \frac{i \ln 2}{\ln(12iL)} \\
&\impliedby pL < \frac{n \ln 2}{\ln(12nL)}
\end{aligned}$$

If $n = 6pL\ln(pL)$,

$$\begin{aligned}
\frac{n \ln 2}{\ln(12nL)} &= \frac{6pL\ln(pL)\ln(2)}{\ln(72pL^2\ln(pL))} = \frac{6pL\ln(pL)\ln(2)}{\ln(72) + \ln(pL^2) + \ln\ln(pL)} \\
&\geq \frac{6pL\ln(pL)\ln(2)}{\ln(72) + \ln(pL^2) + \ln(pL) - 1} \geq \frac{6\ln(pL)\ln(2)}{3\ln(pL^2)} \\
&= 2pL\ln 2 > pL.
\end{aligned}$$

Remark 17.9 If ReLU is replaced with a degree $r \geq 2$ piecewise polynomial activation, have r^i-degree polynomial in each cell of partition, and shatter coefficient upper bound scales with L^2 not L. The lower bound in this case still has L not L^2; it's not known where the looseness is.

Lower bounds are based on digit extraction, and for each pair (p, L) require a fixed architecture.

References

Allen-Zhu, Zeyuan, and Yuanzhi Li. 2019. "What Can ResNet Learn Efficiently, Going Beyond Kernels?"

Allen-Zhu, Zeyuan, Yuanzhi Li, and Yingyu Liang. 2018. "Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers." *arXiv Preprint arXiv:1811.04918*.

Allen-Zhu, Zeyuan, Yuanzhi Li, and Zhao Song. 2018. "A Convergence Theory for Deep Learning via over-Parameterization."

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. "Wasserstein Generative Adversarial Networks." In *ICML*.

Arora, Sanjeev, Nadav Cohen, Noah Golowich, and Wei Hu. 2018a. "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks."

———. 2018b. "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks."

Arora, Sanjeev, Nadav Cohen, and Elad Hazan. 2018. "On the Optimization of Deep Networks: Implicit Acceleration by Overparameterization." In *Proceedings of the 35th International Conference on Machine Learning*, edited by Jennifer Dy and Andreas Krause, 80:244–53. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR. http://proceedings.mlr.press/v80/arora18a.html.

Arora, Sanjeev, Nadav Cohen, Wei Hu, and Yuping Luo. 2019. "Implicit Regularization in Deep Matrix Factorization." In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, 32:7413–24. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/c0c783b5fc0d7d808f1d14a6e9c8280d-Paper.pdf.

Arora, Sanjeev, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. 2019. "On Exact Computation with an Infinitely Wide Neural Net." *arXiv Preprint arXiv:1904.11955*.

Arora, Sanjeev, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. 2019. "Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks." *arXiv Preprint arXiv:1901.08584*.

Arora, Sanjeev, Rong Ge, Behnam Neyshabur, and Yi Zhang. 2018. "Stronger Generalization Bounds for Deep Nets via a Compression Approach."

Bach, Francis. 2017. "Breaking the Curse of Dimensionality with Convex Neural Networks." *Journal of Machine Learning Research* 18 (19): 1–53.

Barron, Andrew R. 1993. "Universal Approximation Bounds for Superpositions of a Sigmoidal Function." *IEEE Transactions on Information Theory* 39 (3): 930–45.

Bartlett, Peter L. 1996. "For Valid Generalization, the Size of the Weights Is More Important Than the Size of the Network." In *NIPS*.

Bartlett, Peter L., Nick Harvey, Chris Liaw, and Abbas Mehrabian. 2017. "Nearly-Tight VC-Dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks."

Bartlett, Peter L., and Philip M. Long. 2020. "Failures of Model-Dependent Generalization Bounds for Least-Norm Interpolation."

Bartlett, Peter L., and Shahar Mendelson. 2002. "Rademacher and Gaussian Complexities: Risk Bounds and Structural Results." *JMLR* 3 (November): 463–82.

Bartlett, Peter, Dylan Foster, and Matus Telgarsky. 2017. "Spectrally-Normalized Margin Bounds for Neural Networks." *NIPS*.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2018. "Reconciling Modern Machine Learning Practice and the Bias-Variance Trade-Off."

Belkin, Mikhail, Daniel Hsu, and Ji Xu. 2019. "Two Models of Double Descent for Weak Features."

Bengio, Yoshua, and Olivier Delalleau. 2011. "Shallow Vs. Deep Sum-Product Networks." In *NIPS*.

Bietti, Alberto, and Francis Bach. 2020. "Deep Equals Shallow for ReLU Networks in Kernel Regimes."

Blum, Avrim, and John Langford. 2003. "PAC-MDL Bounds." In *Learning Theory and Kernel Machines*, 344–57. Springer.

Borwein, Jonathan, and Adrian Lewis. 2000. *Convex Analysis and Nonlinear Optimization*. Springer Publishing Company, Incorporated.

Bubeck, Sébastien. 2014. "Theory of Convex Optimization for Machine Learning."

Cao, Yuan, and Quanquan Gu. 2020a. "Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks."

———. 2020b. "Generalization Error Bounds of Gradient Descent for Learning over-Parameterized Deep ReLU Networks."

Carmon, Yair, and John C. Duchi. 2018. "Analysis of Krylov Subspace Solutions of Regularized Nonconvex Quadratic Problems." In *NIPS*.

Chaudhuri, Kamalika, and Sanjoy Dasgupta. 2014. "Rates of Convergence for Nearest Neighbor Classification."

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. "Neural Ordinary Differential Equations."

Chen, Zixiang, Yuan Cao, Quanquan Gu, and Tong Zhang. 2020. "A Generalized Neural Tangent Kernel Analysis for Two-Layer Neural Networks."

Chen, Zixiang, Yuan Cao, Difan Zou, and Quanquan Gu. 2019. "How Much over-Parameterization Is Sufficient to Learn Deep ReLU Networks?"

Chizat, Lénaïc, and Francis Bach. 2018. "On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport." *arXiv e-Prints*, May, arXiv:1805.09545. http://arxiv.org/abs/1805.09545.

———. 2019. "A Note on Lazy Training in Supervised Differentiable Programming."

———. 2020. "Implicit Bias of Gradient Descent for Wide Two-Layer Neural Networks Trained with the Logistic Loss." *arXiv:2002.04486 [math.OC]*.

Cho, Youngmin, and Lawrence K. Saul. 2009. "Kernel Methods for Deep Learning." In *NIPS*.

Cisse, Moustapha, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. "Parseval Networks: Improving Robustness to Adversarial Examples."

Clarke, Francis H., Yuri S. Ledyaev, Ronald J. Stern, and Peter R. Wolenski. 1998. *Nonsmooth Analysis and Control Theory*. Springer.

Cohen, Nadav, Or Sharir, and Amnon Shashua. 2016. "On the Expressive Power of Deep Learning: A Tensor Analysis." In *29th Annual Conference on Learning Theory*, edited by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, 49:698–728. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR. http://proceedings.mlr.press/v49/cohen16.html.

Cohen, Nadav, and Amnon Shashua. 2016. "Convolutional Rectifier Networks as Generalized Tensor Decompositions." In *Proceedings of the 33rd International Conference on Machine Learning*, edited by Maria Florina Balcan and Kilian Q. Weinberger, 48:955–63. Proceedings of Machine Learning Research. New York, New York, USA: PMLR. http://proceedings.mlr.press/v48/cohenb16.html.

Cybenko, George. 1989. "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals and Systems* 2 (4): 303–14.

Daniely, Amit. 2017. "Depth Separation for Neural Networks." In *COLT*.

Daniely, Amit, and Eran Malach. 2020. "Learning Parities with Neural Networks."

Davis, Damek, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D. Lee. 2018. "Stochastic Subgradient Method Converges on Tame Functions."

Diakonikolas, Ilias, Surbhi Goel, Sushrut Karmalkar, Adam R. Klivans, and Mahdi Soltanolkotabi. 2020. "Approximation Schemes for ReLU Regression."

Du, Simon S., Wei Hu, and Jason D. Lee. 2018. "Algorithmic Regularization in Learning Deep Homogeneous Models: Layers Are Automatically Balanced."

Du, Simon S., Xiyu Zhai, Barnabas Poczos, and Aarti Singh. 2018. "Gradient Descent Provably Optimizes over-Parameterized Neural Networks."

Du, Simon S, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. 2018. "Gradient Descent Finds Global Minima of Deep Neural Networks." *arXiv Preprint arXiv:1811.03804*.

Du, Simon, and Wei Hu. 2019. "Width Provably Matters in Optimization for Deep Linear Neural Networks."

Dziugaite, Gintare Karolina, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M. Roy. 2020. "In Search of Robust Measures of Generalization."

Dziugaite, Gintare Karolina, and Daniel M. Roy. 2017. "Computing Nonvacuous Generalization Bounds for Deep (stochastic) Neural Networks with Many More Parameters Than Training Data."

Eldan, Ronen, and Ohad Shamir. 2015. "The Power of Depth for Feedforward Neural Networks."

Folland, Gerald B. 1999. *Real Analysis: Modern Techniques and Their Applications*. 2nd ed. Wiley Interscience.

Funahashi, K. 1989. "On the Approximate Realization of Continuous Mappings by Neural Networks." *Neural Netw.* 2 (3): 183–92.

Ge, Rong, Jason D. Lee, and Tengyu Ma. 2016. "Matrix Completion Has No Spurious Local Minimum." In *NIPS*.

Ghorbani, Behrooz, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. 2020. "When Do Neural Networks Outperform Kernel Methods?"

Goel, Surbhi, Adam Klivans, Pasin Manurangsi, and Daniel Reichman. 2020. "Tight Hardness Results for Training Depth-2 ReLU Networks."

Golowich, Noah, Alexander Rakhlin, and Ohad Shamir. 2018. "Size-Independent Sample Complexity of Neural Networks." In *COLT*.

Gunasekar, Suriya, Jason D Lee, Daniel Soudry, and Nati Srebro. 2018a. "Implicit Bias of Gradient Descent on Linear Convolutional Networks." In *Advances in Neural Information Processing Systems*, 9461–71.

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

Gunasekar, Suriya, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. 2017. "Implicit Regularization in Matrix Factorization."

Gurvits, Leonid, and Pascal Koiran. 1995. "Approximation and Learning of Convex Superpositions." In *Computational Learning Theory*, edited by Paul Vitányi, 222–36. Springer.

Hanin, Boris, and David Rolnick. 2019. "Deep ReLU Networks Have Surprisingly Few Activation Patterns."

Hastie, Trevor, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. 2019. "Surprises in High-Dimensional Ridgeless Least Squares Interpolation."

Hertz, John, Anders Krogh, and Richard G. Palmer. 1991. *Introduction to the Theory of Neural Computation*. USA: Addison-Wesley Longman Publishing Co., Inc.

Hiriart-Urruty, Jean-Baptiste, and Claude Lemaréchal. 2001. *Fundamentals of Convex Analysis*. Springer Publishing Company, Incorporated.

Hornik, K., M. Stinchcombe, and H. White. 1989. "Multilayer Feedforward Networks Are Universal Approximators." *Neural Networks* 2 (5): 359–66.

Jacot, Arthur, Franck Gabriel, and Clément Hongler. 2018. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks." In *Advances in Neural Information Processing Systems*, 8571–80.

Ji, Ziwei. 2020. "Personal Communication."

Ji, Ziwei, Miroslav Dudík, Robert E. Schapire, and Matus Telgarsky. 2020. "Gradient Descent Follows the Regularization Path for General Losses." In *COLT*.

Ji, Ziwei, Justin D. Li, and Matus Telgarsky. 2021. "Early-Stopped Neural Networks Are Consistent."

Ji, Ziwei, and Matus Telgarsky. 2018. "Gradient Descent Aligns the Layers of Deep Linear Networks." *arXiv:1810.02032 [cs.LG]*.

———. 2019a. "Polylogarithmic Width Suffices for Gradient Descent to Achieve Arbitrarily Small Test Error with Shallow ReLU Networks."

———. 2019b. "Risk and Parameter Convergence of Logistic Regression." In *COLT*.

———. 2020. "Directional Convergence and Alignment in Deep Learning." *arXiv:2006.06657 [cs.LG]*.

Ji, Ziwei, Matus Telgarsky, and Ruicheng Xian. 2020. "Neural Tangent Kernels, Transportation Mappings, and Universal Approximation." In *ICLR*.

Jiang, Yiding, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. 2020. "Fantastic Generalization Measures and Where to Find Them." In *ICLR*.

Jin, Chi, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. 2017. "How to Escape Saddle Points Efficiently." In *ICML*.

Jones, Lee K. 1992. "A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection Pursuit Regression and Neural Network Training." *The Annals of Statistics* 20 (1): 608–13.

Kakade, Sham, and Jason D. Lee. 2018. "Provably Correct Automatic Subdifferentiation for Qualified Programs."

Kamath, Pritish, Omar Montasser, and Nathan Srebro. 2020. "Approximate Is Good Enough: Probabilistic Variants of Dimensional and Margin Complexity."

Kawaguchi, Kenji. 2016. "Deep Learning Without Poor Local Minima." In *NIPS*.

Kolmogorov, A. N., and V. M. Tikhomirov. 1959. "ϵ-Entropy and ϵ-Capacity of Sets in Function Spaces." *Uspekhi Mat. Nauk* 14 (86, 2): 3–86.

Ledoux, M., and M. Talagrand. 1991. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer.

Lee, Holden, Rong Ge, Tengyu Ma, Andrej Risteski, and Sanjeev Arora. 2017. "On the Ability of Neural Nets to Express Distributions." In *COLT*.

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

Leshno, Moshe, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. 1993. "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function." *Neural Networks* 6 (6): 861–67. http://dblp.uni-trier.de/db/journals/nn/nn6.html#LeshnoLPS93.

Li, Yuanzhi, and Yingyu Liang. 2018. "Learning Overparameterized Neural Networks via Stochastic Gradient Descent on Structured Data."

Long, Philip M., and Hanie Sedghi. 2019. "Generalization Bounds for Deep Convolutional Neural Networks."

Luxburg, Ulrike von, and Olivier Bousquet. 2004. "Distance-Based Classification with Lipschitz Functions." *Journal of Machine Learning Research*.

Lyu, Kaifeng, and Jian Li. 2019. "Gradient Descent Maximizes the Margin of Homogeneous Neural Networks."

Mei, Song, Andrea Montanari, and Phan-Minh Nguyen. 2018. "A Mean Field View of the Landscape of Two-Layers Neural Networks." *arXiv e-Prints*, April, arXiv:1804.06561. http://arxiv.org/abs/1804.06561.

Montanelli, Hadrien, Haizhao Yang, and Qiang Du. 2020. "Deep ReLU Networks Overcome the Curse of Dimensionality for Bandlimited Functions."

Montúfar, Guido, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. 2014. "On the Number of Linear Regions of Deep Neural Networks." In *NIPS*.

Moran, Shay, and Amir Yehudayoff. 2015. "Sample Compression Schemes for VC Classes."

Nagarajan, Vaishnavh, and J. Zico Kolter. 2019. "Uniform Convergence May Be Unable to Explain Generalization in Deep Learning."

Negrea, Jeffrey, Gintare Karolina Dziugaite, and Daniel M. Roy. 2019. "In Defense of Uniform Convergence: Generalization via Derandomization with an Application to Interpolating Predictors."

Nesterov, Yurii. 2003. *Introductory Lectures on Convex Optimization — a Basic Course*. Springer.

Nesterov, Yurii, and B. T. Polyak. 2006. "Cubic Regularization of Newton Method and Its Global Performance." *Math. Program.* 108 (1): 177–205.

Neyshabur, Behnam, Srinadh Bhojanapalli, and Nathan Srebro. 2018. "A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks." In *ICLR*.

Neyshabur, Behnam, Ryota Tomioka, and Nathan Srebro. 2014. "In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning." *arXiv:1412.6614 [cs.LG]*.

Nguyen, Quynh, and Matthias Hein. 2017. "The Loss Surface of Deep and Wide Neural Networks."

Novak, Roman, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2018. "Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes." *arXiv e-Prints*. http://arxiv.org/abs/1810.05148.

Novikoff, Albert B. J. 1962. "On Convergence Proofs on Perceptrons." *In Proceedings of the Symposium on the Mathematical Theory of Automata* 12: 615–22.

Oymak, Samet, and Mahdi Soltanolkotabi. 2019. "Towards Moderate Overparameterization: Global Convergence Guarantees for Training Shallow Neural Networks." *arXiv Preprint arXiv:1902.04674*.

Pisier, Gilles. 1980. "Remarques Sur Un résultat Non Publié de b. Maurey." *Séminaire Analyse Fonctionnelle (dit)*, 1–12.

Rolnick, David, and Max Tegmark. 2017. "The Power of Deeper Networks for Expressing Natural Functions."

Safran, Itay, and Ohad Shamir. 2016. "Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks."

Schapire, Robert E., and Yoav Freund. 2012. *Boosting: Foundations and Algorithms*. MIT Press.

Schapire, Robert E., Yoav Freund, Peter Bartlett, and Wee Sun Lee. 1997. "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods." In *ICML*, 322–30.

Schmidt-Hieber, Johannes. 2017. "Nonparametric Regression Using Deep Neural Networks with ReLU Activation Function."

Shallue, Christopher J., Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. 2018. "Measuring the Effects of Data Parallelism on Neural Network Training."

Shamir, Ohad. 2018. "Exponential Convergence Time of Gradient Descent for One-Dimensional Deep Linear Neural Networks." *arXiv:1809.08587 [cs.LG]*.

Shamir, Ohad, and Tong Zhang. 2013. "Stochastic Gradient Descent for Non-Smooth Optimization: Convergence Results and Optimal Averaging Schemes." In *ICML*.

Siegelmann, Hava, and Eduardo Sontag. 1994. "Analog Computation via Neural Networks." *Theoretical Computer Science* 131 (2): 331–60.

Soudry, Daniel, Elad Hoffer, and Nathan Srebro. 2017. "The Implicit Bias of Gradient Descent on Separable Data." *arXiv Preprint arXiv:1710.10345*.

Steinwart, Ingo, and Andreas Christmann. 2008. *Support Vector Machines*. 1st ed. Springer.

Suzuki, Taiji, Hiroshi Abe, and Tomoaki Nishimura. 2019. "Compression Based Bound for Non-Compressed Network: Unified Generalization Error Analysis of Large Compressible Deep Neural Network."

Telgarsky, Matus. 2013. "Margins, Shrinkage, and Boosting." In *ICML*.

———. 2015. "Representation Benefits of Deep Feedforward Networks."

———. 2016. "Benefits of Depth in Neural Networks." In *COLT*.

———. 2017. "Neural Networks and Rational Functions." In *ICML*.

Tzen, Belinda, and Maxim Raginsky. 2019. "Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit."

Vardi, Gal, and Ohad Shamir. 2020. "Neural Networks with Small Weights and Depth-Separation Barriers." *arXiv:2006.00625 [cs.LG]*.

Wainwright, Martin J. 2015. "UC Berkeley Statistics 210B, Lecture Notes: Basic tail and concentration bounds." January 2015. https://www.stat.berkeley.edu/%C2%A0mjwain/stat210b/.

———. 2019. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. 1st ed. Cambridge University Press.

Wei, Colin, and Tengyu Ma. 2019. "Data-Dependent Sample Complexity of Deep Neural Networks via Lipschitz Augmentation."

Weierstrass, Karl. 1885. "Über Die Analytische Darstellbarkeit Sogenannter Willkürlicher Functionen Einer Reellen Veränderlichen." *Sitzungsberichte Der Akademie Zu Berlin*, 633–39, 789–805.

Yarotsky, Dmitry. 2016. "Error Bounds for Approximations with Deep ReLU Networks."

Yehudai, Gilad, and Ohad Shamir. 2019. "On the Power and Limitations of Random Features for Understanding Neural Networks." *arXiv:1904.00687 [cs.LG]*.

———. 2020. "Learning a Single Neuron with Gradient Methods." *arXiv:2001.05205 [cs.LG]*.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. "Understanding Deep Learning Requires Rethinking Generalization." *ICLR*.

Zhou, Lijia, D. J. Sutherland, and Nathan Srebro. 2020. "On Uniform Convergence and Low-Norm Interpolation Learning."

Zhou, Wenda, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. 2018. "Non-Vacuous Generalization Bounds at the ImageNet Scale: A PAC-Bayesian Compression Approach."

Zou, Difan, Yuan Cao, Dongruo Zhou, and Quanquan Gu. 2018. "Stochastic Gradient Descent Optimizes over-Parameterized Deep Relu Networks."

Zou, Difan, and Quanquan Gu. 2019. "An Improved Analysis of Training over-Parameterized Deep Neural Networks."