Early-stopped neural networks are consistent

Anonymous Author(s)
Affiliation
Address
email

Abstract

This work studies the behavior of neural networks trained with the logistic loss via gradient descent on binary classification data where the underlying data distribution is general, and the (optimal) Bayes risk is not necessarily zero. In this setting, it is shown that gradient descent with early stopping achieves population risk arbitrarily close to optimal in terms of not just logistic and misclassification losses, but also in terms of calibration, meaning the sigmoid mapping of its outputs approximates the true underlying conditional distribution arbitrarily finely. Moreover, the necessary iteration, sample, and architectural complexities of this analysis all scale naturally with a certain complexity measure of the true conditional model. Lastly, while it is not shown that early stopping is necessary, it is shown that any classifier satisfying a basic *local interpolation property* is necessarily inconsistent.

1 Overview and main result

Deep networks trained with gradient descent seem to have no trouble adapting to arbitrary prediction problems, and are steadily displacing stalwart methods across many domains. In this work, we provide a mathematical basis for this good performance on arbitrary binary classification problems, considering the simplest possible networks: a shallow architecture where only the inner (input-facing) weights are trained using vanilla gradient descent with a constant step size. The central contributions are as follows.

1. **Fully general classification tasks.** The joint distribution generating the (x, y) pairs only requires x to be bounded, and is otherwise arbitrary. In particular, the underlying distribution may be noisy, meaning the true conditional model of the labels, $\Pr[Y = 1|X = x]$, is arbitrary.

   In this setting, we show that as data, width, and training time increase, the logistic loss *measured over the population* converges to optimality over all measurable functions, which moreover implies that the induced conditional model (defined by a sigmoid mapping) converges to the true model, and the population misclassification rate also converges to optimality. This is in contrast with prior analyses of gradient descent, which either only consider the training risk [Allen-Zhu et al., 2018b, Du et al., 2019, Zou et al., 2018, Oymak and Soltanolkotabi, 2019], or can only handle restricted conditional models [Allen-Zhu et al., 2018a, Arora et al., 2019, Cao and Gu, 2019, Nitanda and Suzuki, 2019, Ji and Telgarsky, 2020b, Chen et al., 2019].

2. **Adaptivity to data simplicity.** The required number of data samples, network nodes, and gradient descent iterations all shrink if the *distribution* satisfies a natural notion of simplicity: the true conditional model $\Pr[Y = 1|X = x]$ is well-approximated by a low-complexity infinite-width random feature model.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Rounding out the story, this work includes a variety of technical contributions powering the above proofs, and additionally a brief toy model hinting towards the necessity of early stopping: concretely, any univariate predictor satisfying a *local interpolation property* can not achieve optimal test error for noisy distributions.

1.1 Main result: low test error via gradient descent

The goal in this work is to minimize the logistic risk over the population: letting μ denote an arbitrary Borel measure over (X, Y) with compactly-supported marginal μ_x and conditional p_y, and a data sample $((x_i, y_i))_{i=1}^n$, define the logistic loss, empirical logistic risk, and logistic risk respectively as

$$\ell(r) := \ln(1 + e^{-z}), \qquad \widehat{\mathcal{R}}(f) := \frac{1}{n}\sum_{k=1}^n \ell(y_k f(x_k)), \qquad \mathcal{R}(f) := \mathbb{E}_{x,y}\ell(yf(x)).$$

We use the logistic loss not only due to its practical prevalence, but also due to an interesting multiplicative error property which improves our main results (cf. Lemma 2.1 and Theorem 1.1), all while being Lipschitz.

We seek to make $\mathcal{R}(f)$ as small as possible: formally, we compare against the Bayes risk

$$\overline{\mathcal{R}} := \inf\left\{\mathcal{R}(f) \ : \ \text{measurable } f : \mathbb{R}^d \to \mathbb{R}\right\}.$$

While competing with $\overline{\mathcal{R}}$ may seem a strenuous goal, in fact it simplifies many aspects of the learning task. Firstly, due to the universal approximation properties of neural networks [Cybenko, 1989, Hornik et al., 1989, Barron, 1993, Funahashi, 1989], we are effectively working over the space of all measurable functions already. Secondly, as will be highlighted in the main result below, via the theory of classification calibration [Zhang, 2004, Bartlett et al., 2006], competing with the Bayes (convex) risk also recovers the true conditional model, and minimizes the misclassification loss; this stands in contrast with the ostensibly more modest goal of minimizing misclassification over a restricted class of predictors, namely the *agnostic learning* setting, which suffers a variety of computational and statistical obstructions [Goel et al., 2020a,b, Yehudai and Shamir, 2020, Frei et al., 2020].

Our predictors are standard shallow networks, trained via gradient descent — the simplest networks which are not convex in their parameters, but satisfy universal approximation. In detail, letting $(a_j)_{j=1}^m$ be uniform random ± 1 signs, $(w_j)_{j=1}^m$ with $w_j \in \mathbb{R}^d$ be standard Gaussians, and $\rho > 0$ be a *temperature*, we predict on an input $x \in \mathbb{R}^d$ with

$$f(x; \rho, a, W) := f(x; W) := \frac{\rho}{\sqrt{m}}\sum_{j=1}^m a_j \sigma_{\mathrm{r}}(w_j^{\intercal} x),$$

where $\sigma_{\mathrm{r}}(z) := \max\{0, z\}$ is the ReLU; since only W is trained, both ρ and a are often dropped. To train, we perform gradient descent with a constant step size on the empirical risk:

$$W_{i+1} := W_i - \eta \nabla \widehat{\mathcal{R}}(W_i), \qquad \text{where } \widehat{\mathcal{R}}(W) := \widehat{\mathcal{R}}\left(x \mapsto f(x; W)\right).$$

For technical convenience, our analysis works with $W_{\leq t} := \arg\min_i \widehat{\mathcal{R}}(W_i)$, the iterate with the best empirical risk over the first t iterations.

Our goal is to show that this iterate $W_{\leq t}$ satisfies $\mathcal{R}(W_{\leq t}) \approx \overline{\mathcal{R}}$. Certain prediction problems may seem simpler than others, and we want our analysis to reflect this in a way which abstracts away as many coincidences of the training process as possible. Concretely, we measure simplicity via the performance and complexity of an infinite-width random feature model over the true distribution, primarily based on the following considerations.

- By measuring performance over the population, random effects of the training sample are removed, and it is impossible for the random feature model to simply revert to memorizing data, as it never sees that training data.
- The random feature model has infinite width, and via sampling can be used as a benchmark for all possible widths simultaneously, but is itself freed from coincidences of random weights.

In detail, our infinite-width random feature model is as follows. Let $\overline{U}_\infty : \mathbb{R}^d \to \mathbb{R}^d$ be an (uncountable) collection of weights (indexed by $\mathbb{R}^d$), and define a prediction mapping via

$$f(x; \overline{U}_\infty) := \int \left\langle \overline{U}_\infty(v), x\mathbb{1}[v^\intercal x \geq 0] \right\rangle \mathrm{d}\mathcal{N}(v), \qquad \mathcal{R}(\overline{U}_\infty) := \mathcal{R}(x \mapsto f(x; \overline{U}_\infty)).$$

Note that for each fixed v, we obtain a *random feature* $x \mapsto x\mathbb{1}[v^\intercal x \geq 0]$. This particular choice is simply the gradient of a corresponding ReLU $\nabla_v \sigma_{\mathrm{r}}(v^\intercal x)$, and is motivated by the NTK literature [Jacot et al., 2018, Li and Liang, 2018, Du et al., 2019]. A similar object has appeared before in NTK convergence analyses [Nitanda and Suzuki, 2019, Ji and Telgarsky, 2020b], but the conditions on $\overline{U}_\infty$ were always strong (e.g., data separation with a margin).

What, then, does it mean for the data to be simple? In this work, it is when there exists a $\overline{U}_\infty$ with $\mathcal{R}(\overline{U}_\infty) \approx \overline{\mathcal{R}}$, and moreover $\overline{U}_\infty$ has low norm; for technical convenience, we measure the norm as the maximum over individual weight norms, meaning $\sup_v \|\overline{U}_\infty(v)\|$. To measure approximability, for sake of interpretation, we use the *binary Kullback-Leibler divergence (KL)*: defining a conditional probability model ϕ_∞ corresponding to $\overline{U}_\infty$ via

$$\phi_\infty(x) := \phi(f(x; \overline{U}_\infty)) \qquad \text{where } \phi(r) := \frac{1}{1 + \exp(-r)},$$

then the binary KL can be written as

$$\mathcal{K}_{\text{bin}}(p_y, \phi_\infty) := \int \left(p_y \ln \frac{p_y}{\phi_\infty} + (1 - p_y) \ln \frac{1 - p_y}{1 - \phi_\infty} \right) \mathrm{d}\mu_x = \mathcal{R}(\overline{U}_\infty) - \overline{\mathcal{R}}.$$

This relationship between binary KL and the excess risk is a convenient property of the logistic loss, which immediately implies calibration as a consequence of achieving the optimal risk.

The pieces are all in place to state our main result.

Theorem 1.1. *Let width m and gradient descent target accuracy ϵ_{gd} be given, choose step size $\eta := 4/\rho^2$, and run gradient descent for $t := 8/\epsilon_{\text{gd}}$ iterations, letting $W_{\leq t} := \arg\min_{i \leq t} \widehat{\mathcal{R}}(W_i)$ denote the iterate with smallest empirical risk. Let reference model $\overline{U}_\infty$ be given with $R := \max\{\rho^2, \sup_v \|\overline{U}_\infty(v)\|\} < \infty$, and define a corresponding conditional model $\phi_\infty(x) := \phi(f(x; \overline{U}_\infty))$. Lastly, define two error quantities τ_0 and τ_1 which go to zero if m increases and ρ decreases, but $m\rho^2$ is increasing:*

$$\tau_0 = \widetilde{\mathcal{O}}\left(\rho d \ln(1/\delta) + \frac{R\sqrt{d \ln(1/\delta)}}{\sqrt{m}} \right), \qquad \tau_1 = \widetilde{\mathcal{O}}\left(\frac{R^{4/3}\sqrt{d \ln(1/\delta)}}{(m\rho^2)^{1/6}} \right),$$

where $\widetilde{\mathcal{O}}$ hides logarithmic dependence on m and d. Then, with probability at least $1 - 40\delta$,

$$\begin{aligned}
& \mathcal{R}(W_{\leq t}) - \overline{\mathcal{R}} && \textit{(logistic error)} \\
\leq\ & \mathcal{K}_{\text{bin}}(p_y, \phi_\infty) + \left(e^{\tau_0 + \tau_1} - 1\right) \mathcal{R}(\overline{U}_\infty) && \textit{(reference model error)} \\
+\ & e^{\tau_1} \epsilon_{\text{gd}} && \textit{(optimization error)} \\
+\ & \widetilde{\mathcal{O}}\left(\frac{e^{\tau_1}(\rho + R)\left(d \ln(n/\delta)\right)^{3/2}}{\sqrt{n}} \right) && \textit{(generalization error)},
\end{aligned}$$

where the classification risk $\mathcal{R}_{\mathrm{z}}(f) := \Pr[\mathrm{sgn}(f(X)) \neq Y]$ and calibration error satisfy

$$\begin{aligned}
& \mathcal{R}(W_{\leq t}) - \overline{\mathcal{R}} && \textit{(logistic error)} \\
\geq\ & 2 \int \left(\phi(f(x; W_{\leq t})) - p_y(x) \right)^2 \mathrm{d}\mu_x(x) && \textit{(calibration error)} \\
\geq\ & \frac{1}{2} \left(\mathcal{R}_{\mathrm{z}}(W_{\leq t}) - \overline{\mathcal{R}}_{\mathrm{z}} \right)^2 && \textit{(classification error)}.
\end{aligned}$$

Lastly, for any $\epsilon > 0$, there exists $\overline{U}_\infty^{(\epsilon)}$ with $\sup_v \|\overline{U}_\infty^{(\epsilon)}(v)\| < \infty$ and whose conditional model $\phi_\infty^{(\epsilon)}(x) := \phi(f((x, 1)/\sqrt{2}; \overline{U}_\infty^{(\epsilon)}))$ (with a bias term) satisfies $\mathcal{K}_{\text{bin}}(p_y, \phi_\infty^{(\epsilon)}) \leq \epsilon$.

Remark 1.1. The key properties of Theorem 1.1 are as follows.

1. All terms can be made arbitrarily small: in addition to increasing width m (to shrink τ), decreasing temperature ρ, and decreasing optimization error ϵ_{gd}, the final part of Theorem 1.1 guarantees $\mathcal{K}_{\text{bin}}(p_y, \phi_\infty)$ can be arbitrarily small as well. This implies *consistency*, namely the property that all three types of error can be driven to optimality, as is formally stated and proved in Corollary 2.3. The choice to use an infinite-width reference model $\overline{U}_\infty$, and moreover to measure its performance over the population, leads to an easy consistency proof: the proof lets a target accuracy be given, fixes a reference model $\overline{U}_\infty$, and lets the data, width, and training iterations increase.

2. All terms are reflective either of the simplicity of $\overline{U}_\infty$ (via $R = \sup_v \|\overline{U}_\infty(v)\|$) or of the underlying conditional model p_y (by allowing small $\mathcal{R}(\overline{U}_\infty)$, and simultaneously small $\mathcal{K}_{\text{bin}}(p_y, \overline{U}_\infty)$ and R).

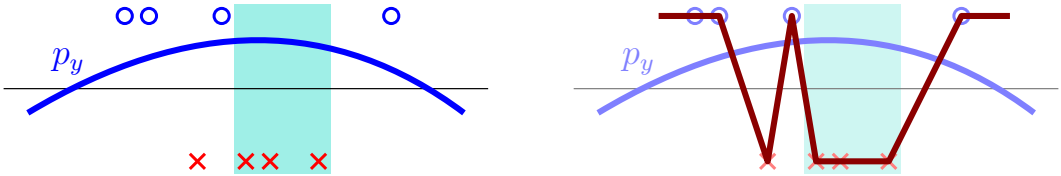
3. One interesting and unusual feature is the multiplicative error on $\mathcal{R}(\overline{U}_\infty)$. Suppose that it is possible to ensure $\mathcal{R}(\overline{U}_\infty) \leq \epsilon_{\text{gd}}$; e.g., this holds under the data separability conditions of [Nitanda and Suzuki, 2019, Ji and Telgarsky, 2020b]. This allows τ_0 to be ignored, and ensures that $\mathcal{K}_{\text{bin}}(p_y, \phi_\infty) = \mathcal{R}(\overline{U}_\infty) - \overline{\mathcal{R}} \leq \mathcal{R}(\overline{U}_\infty) \leq \epsilon_{\text{gd}}$ as well. As such, to achieve total overall population error ϵ_{gd}, it suffices to have $n = 1/\epsilon_{\text{gd}}^2$ and $m = \widetilde{\mathcal{O}}(R^8)$; in this way, the *polylogarithmic width* results of prior work are not only recovered [Ji and Telgarsky, 2020b, Chen et al., 2019], but moreover derived via a unified analysis with the present general nonseparable setting. This multiplicative property is made possible by the logistic loss (cf. Lemma 2.1), and is the main reason that general losses are not treated here.

4. The relationship to classification and calibration errors is merely a restatement of existing results [Zhang, 2004, Bartlett et al., 2006], though it is reproved here in an elementary way for the special case of the logistic loss. Similarly, the guarantee that $\mathcal{K}_{\text{bin}}(p_y, \phi_\infty^{(\epsilon)})$ can be made arbitrarily small is also not a primary contribution, and indeed most of the heavy lifting is provided both by prior work in neural network approximation [Barron, 1993], and by the existing and reliable machinery for proving consistency [Schapire and Freund, 2012]. As such, the consistency result is stated only much later in Corollary 2.3, and our focus is on the exact risk guarantees in Theorem 1.1.

Further discussion of Theorem 1.1, including the formal consistency result (cf. Corollary 2.3) and a proof sketch, all appear in Section 2. Full proofs appear in the appendices.

1.2 Should we early stop?

Theorem 1.1 uses early stopping. Part of this is purely technical, as will be discussed both in the analysis in Section 2, and in the open problems in Section 4. It is of course natural to wonder what happens if one trains indefinitely, entering the actively-studied *interpolation regime* [Belkin et al., 2018, 2019, Bartlett et al., 2019]. Furthermore, there is evidence that gradient descent on shallow



(a) Conditional model p_y and some noisy data. A smoothed prediction rule would perform well.

(b) A *local interpolation rule* working very hard to fit the noisy data.

Figure 1: When data is noisy, it's best to give up on a few points. The shaded region here highlights consecutive points with the wrong label; as in Proposition 1.2, prediction rules that locally interpolate will have a large population risk in these regions.

networks limits towards a particular interpolating choice, one with large margins [Soudry et al., 2018, Lyu and Li, 2020, Ji and Telgarsky, 2020a, Chizat and Bach, 2020]. Is this behavior favorable?

While it is not ruled out here that the interpolating solutions found by neural networks perform well, it is shown that at least in the low-dimensional (univariate!) setting, if a prediction rule perfectly labels the data and is not too wild between training points, then it is guaranteed to achieve poor test loss on noisy problems. This negative observation is not at odds with the interpolation literature, where the performance of some rules improves with dimension [Belkin et al., 2018].

Proposition 1.2. *Given a finite sample* $((x_i, y_i))_{i=1}^n$ *with* $x_i \in \mathbb{R}$ *and* $y_i \in \{\pm 1\}$*, let* $\mathcal{F}_n$ *denote the collection of* local interpolation rules *(cf. Figure 1): letting* $x_{(i)}$ *index examples in sorted order, meaning* $x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)}$*, define* $\mathcal{F}_n$ *as*

$$\mathcal{F}_n := \big\{ f : \mathbb{R}^d \to \mathbb{R} \,:\, \forall i \; f(x_{(i)}) = y_{(i)}, \text{ and } \\ \text{if } y_{(i)} = y_{(i+1)}, \text{ then } \inf_{\alpha \in [0,1]} f\left(\alpha x_{(i)} + (1-\alpha) x_{(i+1)}\right) y_{(i)} > 0 \big\}.$$

Then there exists a constant $c > 0$ *so that with probability at least* $1 - \delta$ *over the draw of* $((x_i, y_i))_{i=1}^n$ *with* $n \ge \ln(1/\delta)/c$*, every* $f \in \mathcal{F}_n$ *satisfies* $\mathcal{R}_z(f) \ge \bar{\mathcal{R}}_z(f) + c$*.*

Although a minor contribution, this result will be discussed briefly in Section 3, with detailed proofs appearing in the appendices.

1.3 Related work

This work and its techniques rely heavily upon prior work from a variety of topics.

Analyses of gradient descent. The proof here shares the most elements with recent works whose width could be polylogarithmic in the sample size and desired target accuracy $1/\epsilon$ [Ji and Telgarsky, 2020b, Chen et al., 2019]. Similarities include using a regret inequality as the core of the proof, using an infinite-width target network [Nitanda and Suzuki, 2019, Ji and Telgarsky, 2020b], and using a *linearization inequality* [Chen et al., 2019, Allen-Zhu et al., 2018b]. On the technical side, the present work differs in the detailed treatment of the logistic loss, and in the linearization inequality which is extended to hold over the population risk; otherwise, the core gradient descent analysis here is arguably simplified relative to these prior works. It should be noted that the use of a regret inequality here and in the previous works crucially makes use of a negated term which was dropped in some classical treatments; this trick is now re-appearing in many places [Orabona and Pál, 2021, Frei et al., 2020].

There are many other, somewhat less similar works in the vast literature of gradient descent on neural networks, in particular in the neural tangent regime [Jacot et al., 2018, Li and Liang, 2018, Du et al., 2019]. These works often handle not only training error, but also testing error [Allen-Zhu et al., 2018a, Arora et al., 2019, Cao and Gu, 2019, Nitanda and Suzuki, 2019, Ji and Telgarsky, 2020b, Chen et al., 2019]. As was mentioned before, these works do not appear to handle arbitrary target models; see for instance the modeling discussion in [Arora et al., 2019, Section 6]. As another interesting recent example, some works explicitly handle certain noisy conditional models, but with error terms that do not go to zero in general [Liang et al., 2021].

Consistency. Consistency of deep networks with classification loss and *some* training procedure is classical; e.g., in [Farago and Lugosi, 1993], the authors show that it suffices to run a computationally intractable algorithm on an architecture chosen to balance VC dimension and universal approximation. Similarly, the work here makes use of Barron's superposition analysis in an infinite-width form to meet the Bayes risk [Barron, 1993, Ji et al., 2020b]. The statistics literature has many other works giving beautiful analyses of neural networks, e.g., even with minimax rates [Schmidt-Hieber, 2017], though it appears this literature generally does not consider gradient descent and arbitrary classification objectives.

Calibration. There is an increasing body of work considering the (in)ability of networks trained with the logistic loss to recover the underlying conditional model. Both on the empirical side [Guo et al., 2017] and on the theoretical side [Bai et al., 2021], the evidence is on the side of the logistic loss doing poorly, specifically being *overconfident*, meaning the sigmoid outputs are too close to 0 or 1. This overconfident regime corresponds to large margins; indeed, since gradient descent can be proved in some settings to exhibit unboundedly large unnormalized margins on all training points [Lyu and Li, 2020], the sigmoid mapping of the predictions will necessarily limit to exactly 0 or 1. On the other hand, as mentioned in [Bai et al., 2021], regularization suffices to circumvent the issue. In the present work, a combination of early stopping and small temperature are employed. As mentioned before, calibration is proved here as an immediate corollary of meeting the optimal logistic risk via classification calibration [Zhang, 2004, Bartlett et al., 2006].

1.4 Further notation and technical background

The loss ℓ, risks $\mathcal{R}$ and $\widehat{\mathcal{R}}$, and network f have been defined. The misclassification risk $\mathcal{R}_{\mathrm{z}}(f) = \Pr[\mathrm{sgn}(f(X)) \neq Y]$ appeared in Theorem 1.1, where $\mathrm{sgn}(f(x)) = 2 \cdot \mathbb{1}[f(x) \geq 0] - 1$.

Next, consider the "gradient" of f with respect to weights W:

$$\nabla f(x; W) := \frac{\rho}{\sqrt{m}} \sum_{j=1}^{m} a_j \mathbb{1}[w_j^{\mathsf{T}} x \geq 0] \boldsymbol{e}_j x^{\mathsf{T}};$$

it may seem the nondifferentiability at 0 is concerning, but in analyses close to initialization (as is the one here), few activations change, and their behavior is treated in a worst-case fashion. Note that, as is easily checked with this expression, $\|\nabla f(W)\| \leq \rho$, which is convenient in many places in the proofs. Here $\|\cdot\|$ denotes the Frobenius norm; $\|\cdot\|_2$ will denote the spectral norm.

Given weight matrix W_i at time i, let $(w_{i,j}^{\mathsf{T}})_{j=1}^m$ refer to its rows. Define features $f^{(i)}$ at time i and a corresponding empirical risk $\widehat{\mathcal{R}}^{(i)}$ using the features at time i as

$$\begin{aligned} f^{(i)}(x; V) &:= \left\langle \nabla f(x; W_i), V \right\rangle = \frac{\rho}{\sqrt{m}} \sum_j a_j v_j^{\mathsf{T}} x \mathbb{1}[w_{i,j}^{\mathsf{T}} x \geq 0], \\ \widehat{\mathcal{R}}^{(i)}(x; V) &:= \widehat{\mathcal{R}}(x \mapsto f^{(i)}(x; V)). \end{aligned}$$

By 1-homogeneity of the ReLU, $f^{(i)}(x; W_i) = f(x; W_i)$, which will also be used often. These features at time i, meaning $f^{(i)}$ and $\widehat{\mathcal{R}}^{(i)}$, are very useful in analyses near initialization, as they do not change much.

To be a bit pedantic about the measure μ: as before, there is a joint distribution μ, which is over the Borel σ-algebra on $\mathbb{R}^d \times \{\pm 1\}$, where $\|x\| \leq 1$ almost surely. This condition suffices to grant both a *disintegration* of μ into marginal μ_x and conditional p_y [Kallenberg, 2002, Chapter 6], and also Lusin's theorem [Folland, 1999, Theorem 7.10], which is used to switch from a measurable function to a continuous one in the consistency proof (cf. Corollary 2.3).

2 Discussion and proof sketch of Theorem 1.1

This section breaks down the proof and discussion into four subsections: a section with common technical tools, then sections for the analysis of generalization, optimization, and approximation.

2.1 Key technical lemmas

There are two general techniques used throughout which are worth highlighting here.

The first is the role of the logistic loss. Prior works used many little inequalities like $-\ell' \leq \ell$, but the inequalities used here can be broken into two groups: the multiplicative error property, and the classification calibration bounds.

Lemma 2.1. *1. For any $a \geq b$,*

$$\frac{\phi(a)}{\phi(b)} \leq e^{a-b} \qquad \text{and} \qquad \frac{\ell(-a)}{\ell(-b)} \leq e^{a-b}.$$

In particular, for any g, h with $\sup_{\|x\|\leq 1} |f(x) - g(x)| \leq \tau$,

$$e^{-\tau}\mathcal{R}(h) \leq \mathcal{R}(g) \leq e^{\tau}\mathcal{R}(h)$$

and if $\tau \leq 1$, then $(1-\tau)\mathcal{R}(h) \leq \mathcal{R}(g) \leq (1+2\tau)\mathcal{R}(h)$. If $\max_k |g(x_k) - f(x_k)| \leq \tau$, then $e^{-\tau}\widehat{\mathcal{R}}(h) \leq \widehat{\mathcal{R}}(g) \leq e^{\tau}\widehat{\mathcal{R}}(h)$.

2. For any $f : \mathbb{R}^d \to \mathbb{R}$ and corresponding conditional model $\phi_f(x) := \phi(f(x))$,

$$\frac{1}{2}\left(\mathcal{R}_{\mathrm{z}}(f) - \overline{\mathcal{R}}_{\mathrm{z}}\right)^2 \leq 2\int (\phi_f(x) - p_y(x))^2 \,\mathrm{d}\mu_x(x) \leq \mathcal{K}_{\mathrm{bin}}(p_y, \phi_f) = \mathcal{R}(f) - \overline{\mathcal{R}}.$$

The first property is used crucially as follows. Prior works, when comparing a network to a local linear approximation, incurred an additive error term. Here, the linear approximation is always performed inside the loss, and yields the preceding multiplicative term; crucially, if the loss is small, then the linear approximation error is irrelevant. This approach recovers existing results aimed at analysis of networks of small width in nearly-separable regimes [Ji and Telgarsky, 2020b, Chen et al., 2019], and extends them.

As mentioned, the second item above appears in the classification calibration literature, but here is proved directly in an elementary fashion.

A second key technical tool is the approach to linear approximation. Many prior analyses near initialization bound the quantity

$$f(x; V) - f(x; W) - \left\langle \nabla f(x; W), V - W \right\rangle,$$

where V and W are both close to initialization [Allen-Zhu et al., 2018b, Cao and Gu, 2019, Chen et al., 2019]. These proofs are typically performed on a fixed example x_k, and then a union bound carries them over to the whole training set.

Here, instead, a simplified expression, $\left\langle \nabla f(x; W) - \nabla f(x; W_0), W \right\rangle$, is crucially bounded not just on a finite sample, but for all $\|x\| \leq 1$. This technique is used twice: when *de-linearizing* after applying generalization bounds, and when sampling from the infinite-width model $\overline{U}_\infty$. The method of proof is inspired by the concept of *co-VC dimension* [Gurvits and Koiran, 1995]: the inequality is union bounded over a set of points that not only cover the sphere, but also all the activation patterns, and a worst-case analysis suffices for the remaining points due to the density of the cover. This co-VC-inspired technique is presented in full in the appendices.

2.2 Generalization analysis

The generalization analysis ends up being easy thanks to the multiplicative error control in Lemma 2.1, and the aforementioned linearization over all points in the sphere. Specifically, to prove generalization, the network is first linearized, then generalization *of linear predictors* is applied, and then de-linearization is applied on the population risk side. Thanks to this, the generalization bound only pays logarithmically in the width.

Typically the easiest step in proving generalization is to provide a worst-case estimate on the range of the predictor. Here, since there is a goal of controlling the logistic loss over the population, brute-forcing this range estimate incurs a polynomial dependence on network width. The solution here is again to apply the co-VC-inspired technique. Details are deferred to appendices.

2.3 Gradient descent analysis

A common tool in linear prediction is the regret inequality

$$\|v_t - z\|^2 + 2\eta \sum_{i<t} \widehat{\mathcal{R}}(v_{i+1}) \leq \|v_0 - z\|^2 + 2t\eta\widehat{\mathcal{R}}(z),$$

which can be derived by expanding the square in $\|v_t - z\|^2$ and applying smoothness and convexity. The term $\|v_t - z\|^2$ is often dropped, but can be used in a very convenient way: by the triangle inequality, if $\|v_t - v_0\| \geq 2\|z - v_0\|$, then the norm terms above may be canceled from both sides, which leaves only the empirical risk terms; overall, this argument ensures both small norm and small empirical risk. This idea has appeared in a variety of works [Shamir, 2020, Ji et al., 2020a], and is used here to provide a convenient norm control, allowing linearization and all other proof parts to go through. Combining this idea with the earlier generalization analysis and a few other minor tricks gives the following bounds.

Lemma 2.2. *Fix any step size $\eta \leq 4/\rho^2$ and temperature $\rho > 0$. Let reference matrix $Z \in \mathbb{R}^{m\times d}$ and scalar $R_Z \geq \rho$ be given with $\|Z - W_0\| \leq R_Z$, and define*

$$t := \frac{R_Z^2}{2\eta\epsilon_{\text{gd}}}, \qquad \tau := \frac{50\rho R_Z^{4/3}\sqrt{d\ln(em^2d/\delta)}}{m^{1/6}}.$$

Then, with probability at least $1 - (7n + 24)\delta$, the empirical risk minimizer amongst the first t iterations, $W_{\leq t}$, satisfies the empirical risk guarantee

$$e^{-\tau}\widehat{\mathcal{R}}(W_{\leq t}) \leq e^{\tau}\widehat{\mathcal{R}}^{(0)}(Z) + \epsilon_{\text{gd}},$$

and the population risk guarantee

$$e^{-3\tau}\mathcal{R}(W_{\leq t}) \leq e^{\tau}\mathcal{R}^{(0)}(Z) + \epsilon_{\text{gd}} + \frac{26e^{\tau}\rho\,(1 + 3R_Z)\left(d\ln(md^2/\delta)\right)^{3/2}}{\sqrt{n}}.$$

This version of the statement, unlike Theorem 1.1, features an arbitrary reference *matrix* Z. This is powerful, though it can be awkward, since W_0 is a random variable.

2.4 Approximation analysis, consistency, and the proof of Theorem 1.1

Rather than trying to reason about good predictors which may happen to be close to random initialization, the approach here is instead to start from deterministic predictors over the population (e.g., $\overline{U}_\infty$), and to use their structure to construct approximants near the initial iterate, the random matrix W_0. Specifically, the approach here is fairly brute-force: given initial weights W_0 with rows $(w_{0,j}^{\intercal})_{j=1}^m$, the rows $(\overline{u}_j)_{j=1}^m$ of the finite width reference matrix $\overline{U} \in \mathbb{R}^{m\times d}$ intended to mimic $\overline{U}_\infty$ (which is after all a mapping $\overline{U}_\infty : \mathbb{R}^d \to \mathbb{R}^d$) are simply

$$\overline{u}_j := \frac{a_j\overline{U}_\infty(w_{0,j})}{\rho\sqrt{m}} + w_{0,j}. \tag{1}$$

By construction, $\|\overline{U} - W_0\| \leq R/\rho$, where $R := \sup_v \|\overline{U}_\infty(v)\|$. To argue that $\mathcal{R}(\overline{U})$ and $\mathcal{R}(\overline{U}_\infty)$ are close, the aforementioned co-VC-inspired technique is again used. Plugging this $\overline{U}$ into Lemma 2.2 and introducing $\mathcal{K}_{\text{bin}}(p_y, \phi_\infty)$ via Lemma 2.1 gives the first part of Theorem 1.1, and the second part of Theorem 1.1 is also from Lemma 2.1.

It remains to prove that for any p_y, there exists $\overline{U}_\infty$ with $\phi(x \mapsto f((x,1)/\sqrt{2}; \overline{U}_\infty)) \approx p_y$ (we must include a bias term). If p_y were continuous, there is a variant of Barron's seminal universal approximation construction which explicitly gives an infinite-width network of the desired form [Barron, 1993, Ji et al., 2020b]. To address continuity is even easier: *Lusin's theorem* [Folland, 1999, Theorem 7.10] lets us take the measurable function p_y, and obtain a continuous function that agrees with it on all but a negligible fraction of the domain. This completes the proof.

As mentioned, a key property of the reference model $\overline{U}_\infty$ is that it depends on neither the random sampling of data, nor the random sampling of weights. This vastly simplifies the proof of consistency, where the proof scheme first fixes an $\epsilon > 0$ and chooses a $\overline{U}_\infty$, and leaves it fixed as m and n vary.

Corollary 2.3. *For each sample size n, let $(W_i^{(n)})_{i\geq 0}$ denote the corresponding sequence of gradient descent iterates obtained with parameter choices $\rho^{(n)} := n^{-1/3}$, and $m^{(n)} := n$, and $\eta^{(n)} := 4/(\rho^{(n)})^2$, and $\epsilon_{\text{gd}}^{(n)} := 1/\sqrt{n}$, and $t^{(n)} := 8\sqrt{n}$, and let $\widehat{W}_n := \arg\min\left\{\widehat{\mathcal{R}}(W_i^{(n)}) : i \leq t^{(n)}\right\}$*

denote the empirical risk minimizer for this sequence, and lastly let $\widehat{\phi}_n(x) := \phi(f((x,1)/\sqrt{2}; \widehat{W}_n))$ *denote the corresponding conditional probability model (with a bias term). Then*

$$\mathcal{R}(\widehat{W}_n) \longrightarrow \overline{\mathcal{R}} \text{ a.s.,} \qquad \mathcal{R}_z(\widehat{W}_n) \longrightarrow \overline{\mathcal{R}}_z \text{ a.s.,} \qquad \widehat{\phi}_n \xrightarrow{L_2(\mu_x)} p_y \text{ a.s.,}$$

where the last convergence is in the $L_2(\mu_x)$ *metric.*

Proofs, as usual, are in the appendices.

3 Discussion and proof sketch of Proposition 1.2

Proposition 1.2 asserts that *local interpolation rules* — predictors which perfectly fit the data, and are not too wild between data points of the same label — will necessarily achieve suboptimal population risk. The proof idea seems simple enough: if the true conditional probability p_y is not one of $\{0, 1/2, 1\}$ everywhere, and is also continuous, then there must exist a region where it is well separated from these three choices. It seems natural that a constant fraction of the data in these regions will form adjacent pairs with the wrong label; a local interpolation rule will fail on exactly these adjacent noisy pairs, which suffices to give the bound. In reality, while this is indeed the proof scheme followed here, the full proof must contend with many technicalities and independence issues. It appears in the appendices.

While the motivation in Section 1.2 focused on neural networks which interpolate, and also maximum margin solutions, the behavior on this noisy univariate data is also well-illustrated by k-nearest-neighbors classifiers (k-nn). Specifically, 1-nn is a local interpolant, and Proposition 1.2 applies. On the other hand, choosing $k = \Theta(\ln(n))$ is known to provide enough smoothing to achieve consistency and avoid interpolation [Devroye et al., 1996].

It should be stressed again that even if the remaining pieces could be proved to apply this result to neural networks, namely necessitating early stopping, it would still be a univariate result only, leaving open many interesting possibilities in higher dimensions.

4 Concluding remarks and open problems

Empirical performance. Does the story here match experiments? E.g., is it often the case that if a neural network performs well, then so does a random feature model? Do neural networks fail on noisy data if care is not taken with temperature and early stopping? Most specifically, is this part of what happens in existing results reporting such failures [Guo et al., 2017]?

Temperature parameter ρ. Another interesting point of study is the temperature parameter ρ. It arises here in a fairly technical way: if p_y is often close to $1/2$, then the random initialization of W_0 gets in the way of learning p_y. The temperature ρ is in fact a brute-force method of suppressing this weight initialization noise. On the other hand, temperature parameters are common across many works which rely heavily on the detailed real-valued outputs of sigmoid and softmax mappings; e.g., in the distillation literature [Hinton et al., 2015]. The temperature also plays the same role as the scale parameter in the *lazy training* regime [Chizat and Bach, 2019].

Random features, and going beyond the NTK. The analysis here early stops before the feature learning begins to occur. How do things fare outside the NTK? Is there an analog of Theorem 1.1, still stopping shy of the interpolation pitfalls of Proposition 1.2, but managing to beat random features generically?

The logistic loss. One reason the logistic is used here is its simple interplay with calibration (e.g., see the elementary proof of Lemma 2.1, as compared with the full machinery of classification calibration [Zhang, 2004, Bartlett et al., 2006]). The other key reason was the multiplicative error property Lemma 2.1. Certainly, the logistic loss is widely used in practice; are the preceding technical points at all related to the widespread empirical use of the logistic loss?

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. `arXiv:1811.04918 [cs.LG]`, 2018a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. `arXiv:1811.03962 [cs.LG]`, 2018b.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. `arXiv:1901.08584 [cs.LG]`, 2019.

Yu Bai, Song Mei, Huan Wang, and Caiming Xiong. Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification. `arXiv:2102.07856 [cs.LG]`, 2021.

Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, May 1993.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. `arXiv:1906.11300 [stat.ML]`, 2019.

Mikhail Belkin, Daniel J. Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *NeurIPS*, 2018.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. 2019. `arXiv:1903.07571 [cs.LG]`.

Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *NeurIPS*, 2019.

Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep relu networks? `arXiv:1911.12360 [cs.LG]`, 2019.

Lénaïc Chizat and Francis Bach. A Note on Lazy Training in Supervised Differentiable Programming. `arXiv:1812.07956v2 [math.OC]`, 2019.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *COLT*, 2020.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and Banach spaces. In *Handbook of the geometry of Banach spaces*, volume 1, pages 317–366, 2001.

L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*, 2019.

A. Farago and G. Lugosi. Strong universal consistency of neural network classifiers. *IEEE Transactions on Information Theory*, 39(4):1146–1151, 1993. doi: 10.1109/18.243433.

Gerald B. Folland. *Real analysis: modern techniques and their applications*. Wiley Interscience, 2 edition, 1999.

Spencer Frei, Yuan Cao, and Quanquan Gu. Agnostic learning of a single neuron with gradient descent. `arXiv:2005.14426 [cs.LG]`, 2020.

K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Netw.*, 2(3):183–192, May 1989. ISSN 0893-6080.

Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *ICML*, 2020a.

Surbhi Goel, Adam R. Klivans, Pasin Manurangsi, and Daniel Reichman. Tight hardness results for training depth-2 relu networks. `arXiv:2011.13550 [cs.LG]`, 2020b.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

Leonid Gurvits and Pascal Koiran. Approximation and learning of convex superpositions. In Paul Vitányi, editor, *Computational Learning Theory*, pages 222–236. Springer, 1995.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. `arXiv:1503.02531 [stat.ML]`, 2015.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, july 1989.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, pages 8571–8580, 2018.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. `arXiv:2006.06657 [cs.LG]`, 2020a.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow ReLU networks. In *ICLR*, 2020b.

Ziwei Ji, Miroslav Dudík, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *COLT*, pages 2109–2136, 2020a.

Ziwei Ji, Matus Telgarsky, and Ruicheng Xian. Neural tangent kernels, transportation mappings, and universal approximation. In *ICLR*, 2020b.

Olav Kallenberg. *Foundations of modern probability*. Probability and its Applications (New York). Springer-Verlag, New York, second edition, 2002.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *NeurIPS*, pages 8157–8166, 2018.

Shiyu Liang, Ruoyu Sun, and R. Srikant. Achieving small test error in mildly overparameterized neural networks. `arXiv:2104.11895 [cs.LG]`, 2021.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *ICLR*, 2020.

Atsushi Nitanda and Taiji Suzuki. Refined generalization analysis of gradient descent for overparameterized two-layer neural networks with smooth activations on classification problems. `arXiv:1905.09870 [stat.ML]`, 2019.

Francesco Orabona and Dávid Pál. Parameter-free stochastic optimization of variationally coherent functions. `arXiv:2102.00236 [math.OC]`, 2021.

Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. `arXiv:1902.04674 [cs.LG]`, 2019.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. 2017. `arXiv:1708.06633 [math.ST]`.

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

Ohad Shamir. Gradient methods never overfit on separable data. `arXiv:2007.00028 [cs.LG]`, 2020.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. In *ICLR*, 2018.

Gilad Yehudai and Ohad Shamir. Learning a single neuron with gradient methods. `arXiv:2001.05205 [cs.LG]`, 2020.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32:56–85, 2004.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. `arXiv:1811.08888 [cs.LG]`, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The title is that early-stopped networks are consistent, and the abstract promises general settings, which together correspond explicitly to Corollary 2.3. A significant portion of the abstract and first page are also devoted to improved performance with favorable data, which is reflected in our main result, namely Theorem 1.1, as elaborated in Remark 1.1.
   (b) Did you describe the limitations of your work? [Yes] Page 1 specifies all the main restrictions: shallow networks, gradient descent, early stopping, classification, logistic loss. Further limitations are discussed throughout, and the open problems in Section 4 are focused on weaknesses and next steps.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The main result, Theorem 1.1, is stated formally, moreover only after all aspects of the setting have been introduced.
   (b) Did you include complete proofs of all theoretical results? [Yes] Full proofs appear in the appendices, and a significant fraction of the body (Sections 2 and 3) is devoted to proof sketches and related technical discussion.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]