PNAS RESEARCH ARTICLE | COMPUTER SCIENCES

Wide and deep neural networks achieve consistency for classification

Adityanarayanan Radhakrishnan[a,b,c], Mikhail Belkin[d,e], and Caroline Uhler[a,b,c,1]

Edited by David Donoho, Stanford University, Stanford, CA; received May 21, 2022; accepted March 2, 2023

While neural networks are used for classification tasks across domains, a long-standing open problem in machine learning is determining whether neural networks trained using standard procedures are consistent for classification, i.e., whether such models minimize the probability of misclassification for arbitrary data distributions. In this work, we identify and construct an explicit set of neural network classifiers that are consistent. Since effective neural networks in practice are typically both wide and deep, we analyze infinitely wide networks that are also infinitely deep. In particular, using the recent connection between infinitely wide neural networks and neural tangent kernels, we provide explicit activation functions that can be used to construct networks that achieve consistency. Interestingly, these activation functions are simple and easy to implement, yet differ from commonly used activations such as ReLU or sigmoid. More generally, we create a taxonomy of infinitely wide and deep networks and show that these models implement one of three well-known classifiers depending on the activation function used: 1) 1-nearest neighbor (model predictions are given by the label of the nearest training example); 2) majority vote (model predictions are given by the label of the class with the greatest representation in the training set); or 3) singular kernel classifiers (a set of classifiers containing those that achieve consistency). Our results highlight the benefit of using deep networks for classification tasks, in contrast to regression tasks, where excessive depth is harmful.

neural networks | classification | consistency | neural tangent kernel

Deep learning has produced state-of-the-art results across several application domains including computer vision (1), natural language processing (2), and biology (3). Despite these empirical successes, our understanding of basic theoretical properties of deep networks is far from satisfactory. In fact, for the fundamental problem of classification, it has not been established whether neural networks trained with standard optimization methods can achieve consistency, i.e., whether they minimize the probability of misclassification for arbitrary data distributions (a property also referred to as Bayes optimality in the statistics literature).*

There is a vast literature on the consistency of statistical machine learning methods, which has traditionally focused on methods that do not interpolate or fit training data exactly (4, 5). Given the recent successes of interpolating neural networks (6–8), there is renewed interest in understanding the consistency of interpolating machine learning models including nearest neighbor methods and kernel methods (9–13). While such methods can be universally consistent in the noninterpolating regime, these models are generally not consistent in the interpolating regime (12–14). Moreover, little is known about the consistency of interpolating deep neural networks. Classical work (15) analyzing the consistency of neural networks utilizes the results of Cybenko (16) and Hornik (17) to show that the Bayes optimal classifier can be approximated by a neural network that is sufficiently wide; i.e., these prior results are concerned with the existence of networks that achieve consistency and do not present computationally feasible algorithms for finding such networks.

By establishing a connection between interpolating kernel smoothers and deep neural networks, we identify and construct an explicit class of interpolating neural networks that, when trained with gradient descent, achieve consistency for classification problems. Our results utilize the recent neural tangent kernel (NTK) connection between training wide neural networks and using kernel methods. Several works (18–21) established conditions under which using a kernel method with the NTK is equivalent to training neural networks, as network width approaches infinity. Given the conceptual simplicity of kernel methods, the NTK has been widely used as a tool

Significance

While neural networks used in practice are often very deep, the benefit of depth is not well understood. Interestingly, it is known that increasing depth is often harmful for regression tasks. In this work, we show that, in contrast to regression, very deep networks can be Bayes optimal for classification. In particular, we provide simple and explicit activation functions that can be used with standard neural network architectures to achieve consistency. This work provides fundamental understanding of classification using deep neural networks, and we envision it will help guide the design of future neural network architectures.

Author affiliations: [a]Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02142; [b]Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA 02142; [c]Broad Institute, Massachusetts Institute of Technology, Cambridge, MA 02142; [d]Halicioğlu Data Science Institute, University of California, San Diego, CA 92093; and [e]Computer Science and Engineering, University of California, San Diego, CA 92093

Author contributions: A.R., M.B., and C.U. designed research; performed research; analyzed data; and wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: cuhler@mit.edu.

This article contains supporting information online at http://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2208779120/-/DCSupplemental.

Published March 30, 2023.

*Consistency refers to a property that holds in an asymptotic sense as the number of training samples approaches infinity.

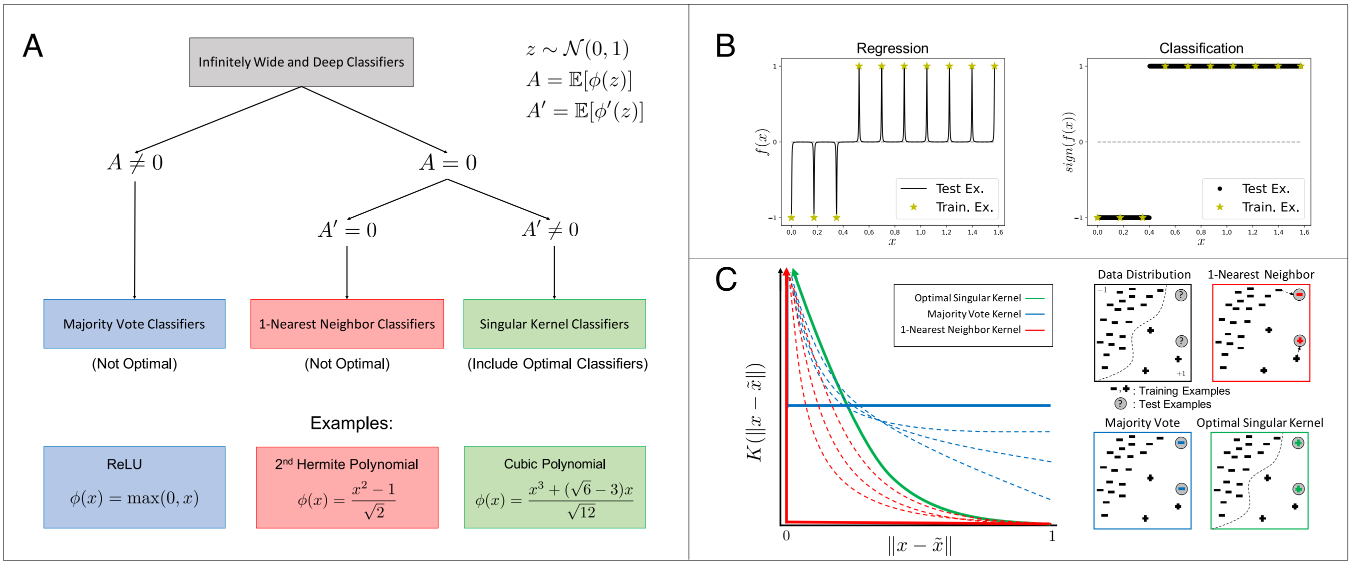

Fig. 1. Behavior of infinitely wide and deep neural networks trained with gradient descent. (*A*) Taxonomy of infinitely wide and deep networks. Depending on the choice of the activation function, $\phi(\cdot)$, these models implement majority vote (blue), 1-nearest neighbor (red), or singular kernel classifiers (green), a subset of which achieve consistency. (*B*) Regression versus classification using infinitely wide and deep networks. While these models are not effective in the regression setting, since their predictions are near zero almost everywhere, they can achieve consistency for classification, where only the sign of the prediction matters. (*C*) Illustration of the different behaviors of infinitely wide and deep networks for varying activation functions. Depending on the activation function, infinitely wide and deep networks implement majority vote (blue), 1-nearest neighbor (red), or singular kernel classifiers that can achieve consistency (green). Singular kernels that grow too slowly are akin to majority vote classifiers (dashed blue), whereas those that grow too quickly are akin to weighted nearest neighbor classifiers (dashed red).

for understanding the theoretical properties of neural networks (19, 21–24). Since neural networks in practice are often both wide and deep, we consider the natural extension of networks that are both infinitely wide and deep.

In particular, we focus on infinitely wide and deep networks in the classification setting and show that they have markedly different behavior than in the regression setting. Indeed, prior work (22, 25) showed that in the regression setting, infinitely wide and deep neural networks simply predict near-zero values at all test samples and, thus, are far from consistent (Fig. 1*B*). As a consequence, these models were dismissed as an approach for explaining the strong performance of deep networks in practice. In stark contrast to regression, we show that the sign of the predictor can be informative even when its numerical output is arbitrarily close to zero (Fig. 1*B* for an illustration). In fact, as we show in this work, this is exactly how infinitely wide and deep neural networks can achieve Bayes optimal classification accuracy even though the output of the network approaches zero.

To characterize the behavior of infinitely wide and deep classifiers, we establish a taxonomy of such models and prove that it includes networks that achieve consistency (Fig. 1*A*). More precisely, we prove that infinitely wide and deep neural network classifiers implement one of the following three well-known classifiers depending on the choice of activation function:

1. *1-nearest neighbor (1-NN) classifiers*: the prediction on a new sample is the label of the nearest sample (under Euclidean distance) in the training set (26).
2. *Majority vote classifiers:* the prediction on a new sample is the label of the class with greater representation in the training set.
3. *Singular kernel classifiers:* the prediction on a new sample is obtained by using the kernel $K(x, \tilde{x}) = \frac{R(\|x-\tilde{x}\|)}{\|x-\tilde{x}\|^{\alpha}}$, where $\alpha > 0$ is the order of the singularity.[†] As is standard when using kernel smoothers for classification, the prediction, $m(x)$, on a new sample x given training data $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ is

$$m(x) = \text{sign}\left(\sum_{i=1}^{n} y^{(i)} K(x^{(i)}, x)\right). \quad \textbf{[1]}$$

As a corollary of a result in ref. 13, it follows that singular kernel classifiers achieve consistency when α is the dimension of the data, d (*SI Appendix*, Appendix C). Hence, our taxonomy and, in particular, Theorem **2** of this work provide exact conditions under which infinitely wide and deep neural network classifiers achieve consistency for any given data dimension. Notably, we identify a simple class of activation functions that yield singular kernel classifiers with $\alpha = d$, and we thus identify concrete examples of neural networks that achieve consistency. For example, for $d = 2$, the infinitely wide and deep classifier with activation function $\phi(x) = (x^3 + (\sqrt{6} - 3)x)/\sqrt{12}$ achieves consistency. Interestingly, the popular rectified linear unit (ReLU) activation $\phi(x) = \max(x, 0)$ leads to an infinitely wide and deep classifier that implements the majority vote classifier and is thus not consistent. Similarly, the activation function $\phi(x) = (x^2 - 1)/\sqrt{2}$ leads to an infinitely wide and deep classifier that implements the 1-NN classifier and is thus also not consistent.

We note that singular kernels provide a natural transition between 1-NN and majority vote classifiers. Namely, as discussed in ref. 13, for $\alpha > d$, singular kernel classifiers behave akin to weighted nearest neighbor classifiers since $\|x - \tilde{x}\|^{\alpha}$ is extremely small for $\tilde{x}$ near x. Similarly, for $\alpha < d$, singular kernel classifiers behave akin to majority vote classifiers since $\|x - \tilde{x}\|^{\alpha}$ is no longer

[†] For this order to be well-defined, $R(\cdot)$ is nonnegative and satisfies $\inf_{|u|<\epsilon} R(u) > 0$ and $|R(u)| < C$ for some $\epsilon, C > 0$.

small for $\tilde{x}$ far from x. We visualize this transition between the three classes established in our taxonomy in Fig. 1*C*.

1. Taxonomy of Infinitely Wide and Deep Neural Networks

In the following, we construct a taxonomy of classifiers implemented by infinitely wide and deep neural networks. Our construction relies on the recent connection between infinitely wide neural networks and kernel methods (18). In particular, this connection involves utilizing a kernel method known as a kernel machine, which is related to the kernel smoother described in Eq. **1**. In contrast to the kernel smoother, a kernel machine with kernel K is given by:

$$\operatorname{sign}\left(yK_n^{-1}K(X,x)\right), \qquad \textbf{[2]}$$

where $X = [x^{(1)}|x^{(2)}|\dots|x^{(n)}] \in \mathbb{R}^{d\times n}$ denotes the training data, $y = [y^{(1)}, y^{(2)}, \dots y^{(n)}] \in \{-1,1\}^{1\times n}$ the labels, $K_n \in \mathbb{R}^{n\times n}$ satisfies $(K_n)_{i,j} = K(x^{(i)}, x^{(j)})$ and $K(X,x) \in \mathbb{R}^n$ satisfies $(K(X,x))_i = K(x^{(i)}, x)$. Both kernel methods can be used as prediction schemes for classification (27). Note that while both algorithms produce predictors with the same functional form, their predictions are generally different. Indeed, understanding the relation between kernel smoothers and kernel machines will be critical to our proof of consistency.

Under certain conditions, training a neural network as width approaches infinity is equivalent[‡] to using a kernel machine with a specific kernel known as the neural tangent kernel (18), which is defined below.

Definition 1: Let $f^{(L)}(x;\mathbf{W})$ denote a fully connected network[§] with L hidden layers with parameters $\mathbf{W}$ operating on data $x \in \mathbb{R}^d$. For $x, \tilde{x} \in \mathbb{R}^d$, the **Neural Tangent Kernel** (**NTK**) is given by:

$$K^{(L)}(x,\tilde{x}) = \langle \nabla_{\mathbf{W}} f^{(L)}(x;\mathbf{W}), \nabla_{\mathbf{W}} f^{(L)}(\tilde{x};\mathbf{W})\rangle.$$

To work with a simple closed form for the NTK and to avoid symmetries arising from the activation function, we will consider training data with probability density function on $\mathcal{S}_+^d$, where $\mathcal{S}_+^d$ is the intersection of the unit sphere $\mathcal{S}^d$ in $d+1$ dimensions and the nonnegative orthant.[¶] We also assume that no samples are repeated in the training data.

In this work, we analyze the behavior of infinitely wide and deep networks by analyzing the kernel machine in Eq. **2**, as depth, L, goes to infinity. To perform our analysis, we utilize the recursive formula for the NTK of a deep network originally presented in ref. 18. Namely, $K^{(L)}$ can be expressed as a function of $K^{(L-1)}$ and the network activation function, $\phi(\cdot)$, yielding a discrete dynamical system indexed by L. The exact formula can be found in Eq. **5**, and additional relevant results from prior works that are used in our proofs are referenced in *SI Appendix*, Appendix A.

Remarkably, the properties of the resulting dynamical system as $L \to \infty$ are governed by the mean of $\phi(z)$ and its derivative, $\phi'(z)$, for $z \sim \mathcal{N}(0,1)$. For simplicity, we will assume throughout that $\mathbb{E}[\phi(z)^2] < \infty$ and similarly $\mathbb{E}[\phi'(z)^2] < \infty$, an assumption that holds for many activation functions used in practice including ReLU, leaky ReLU, sigmoid, sinusoids, and polynomials. By defining $A = \mathbb{E}[\phi(z)]$ and $A' = \mathbb{E}[\phi'(z)]$, we break down our analysis into the following three cases:

$$\begin{aligned}&\text{Case 1: } A = 0\,,\ A' \neq 0,\\ &\text{Case 2: } A = 0\,,\ A' = 0,\\ &\text{Case 3: } A \neq 0.\end{aligned}$$

Under cases 1 and 2, 0 is the unique fixed point attractor of the recurrence for $K^{(L)}$ and thus $K^{(L)}(x,\tilde{x}) \to 0$ as $L \to \infty$ for $x \neq \tilde{x}$. As a consequence, cases 1 and 2 lead to infinitely wide and deep neural networks that predict 0 almost everywhere. Hence, these networks are far from Bayes optimal in the regression setting and were thus dismissed as an approach for explaining the strong performance of deep networks. On the other hand, case 3 yields nonzero values for any pair of examples, and thus, prior works that analyzed the regression setting (22, 25) focused on activation functions satisfying case 3.

In stark contrast to the regression setting, we will show that infinitely wide and deep networks with activation functions satisfying case 1 are effective for classification, with a subset achieving consistency. In particular, we will show that networks in case 1 implement singular kernel classifiers, while those in case 2 implement 1-NN classifiers. Notably, we will identify conditions and provide explicit examples of activation functions in case 1 that guarantee consistency. We will then show that infinitely wide and deep classifiers with activations satisfying case 3 generally correspond to majority vote classifiers. A summary of our taxonomy is presented in Fig. 1*A*, and we will now discuss each of the three cases in more depth.

Case 1 ($A = 0, A' \neq 0$) Networks Implement Singular Kernel Classifiers and Can Achieve Optimality. We establish conditions on the activation function under which an infinitely wide and deep network implements a singular kernel classifier (Theorem **1**). We then utilize results of (13) to show that this set of classifiers contains those that achieve consistency for any given data dimension. Lastly, we will present explicit activation functions that lead to infinitely wide and deep classifiers that achieve consistency. We begin with the following theorem, which establishes conditions under which the infinite depth limit of the NTK is a singular kernel.

Theorem 1. *Let $K^{(L)}$ denote the NTK of a fully connected neural network with L hidden layers and activation function $\phi(\cdot)$. For $z \sim \mathcal{N}(0,1)$, define $A = \mathbb{E}[\phi(z)]$, $A' = \mathbb{E}[\phi'(z)]$, and $B' = \mathbb{E}[\phi'(z)^2]$. If $A = 0$ and $A' \neq 0$; then, for $x, \tilde{x} \in \mathcal{S}_+^d$:*

$$\lim_{L\to\infty} \frac{K^{(L)}(x,\tilde{x})}{(A')^{2L}(L+1)} = \frac{R(\|x-\tilde{x}\|)}{\|x-\tilde{x}\|^{\alpha}},$$

where $\alpha = -2\frac{\log(A'^2)}{\log(B')}$, and $R(\cdot)$ is nonnegative, bounded from above, and bounded away from 0 around 0.

The full proof is presented in *SI Appendix*, Appendix B, and we outline its key steps in Section 2. Theorem **2** below characterizes the activation functions for which the infinitely

[‡] This equivalence requires a particular initialization scheme on the weights known as the NTK initialization scheme (18). Formally, this equivalence holds when offset terms corresponding to the predictions of the neural network at initialization are added to those given by using a kernel machine with the NTK (18). Like in prior works, e.g. (22, 23, 25, 28, 29), we will analyze the NTK without such offset. This model corresponds to averaging the predictions of infinitely many infinite width neural networks (30).

[§] Throughout this work, we consider fully connected networks that have no bias terms.

[¶] For example, min–max scaling followed by projection onto the sphere results in the data lying in this region.

wide and deep network achieves consistency. In particular, we establish the consistency of the infinitely wide and deep neural network classifier, $m_n(\cdot)$, given by taking the limit as $L \to \infty$ of the kernel machine in Eq. **2** with $K = K^{(L)}$, i.e.

$$m_n(x) = \lim_{L \to \infty} \operatorname{sign}\left(y\big(K_n^{(L)}\big)^{-1} K^{(L)}(X, x)\right). \qquad [3]$$

Theorem 2. *Let m_n denote the classifier in Eq.* ***3*** *corresponding to training an infinitely wide and deep network with activation function $\phi(\cdot)$ on n training examples. For $z \sim \mathcal{N}(0,1)$, define $A = \mathbb{E}[\phi(z)]$, $A' = \mathbb{E}[\phi'(z)]$, and $B' = \mathbb{E}[\phi'(z)^2]$. If*

$$A = 0 \quad \text{and} \quad A' \neq 0 \quad \text{and} \quad -\frac{\log(A'^2)}{\log(B')} = \frac{d}{2},$$

then this classifier is Bayes optimal.[#]

While the full proof of Theorem **2** is presented in *SI Appendix*, Appendixes B and C, we outline its key steps in Section 2. In particular, the proof follows by using Theorem **1** above, proving that m_n is a singular kernel classifier, and then using the results of (13), which establish conditions under which singular kernel estimators achieve optimality. The following corollaries (proofs in *SI Appendix*, Appendix D) present concrete classes of activation functions that satisfy the conditions of Theorem **2** for any given data dimension d.

Corollary 1. *Let m_n denote the classifier in Eq.* ***3*** *corresponding to training an infinitely wide and deep network with activation function*

$$\phi(x) = \begin{cases} \frac{1}{\sqrt{2}} h_7(x) + \frac{1}{\sqrt{2}} x & \mathit{if} d = 1, \\ \frac{1}{2^{d/4}} h_3(x) + \sqrt{1 - \frac{2}{2^{d/2}}}\, h_2(x) + \frac{1}{2^{d/4}} x & \mathit{if} d \geq 2, \end{cases}$$

where $h_i(x)$ is the i^{th} probabilist's Hermite polynomial.[‖] *Then, the classifier m_n is Bayes optimal.*

Corollary 2. *For $d \geq 2$, let m_n denote the classifier in Eq.* ***3*** *corresponding to training an infinitely wide and deep network with activation function*

$$\phi(x) = \frac{\sin(ax)}{\sqrt{\sinh(a^2)}}; \qquad -\frac{\log \frac{a^2}{\sinh(a^2)}}{\log \frac{a^2 \cosh(a^2)}{\sinh(a^2)}} = \frac{d}{2}.$$

Then, the classifier m_n is Bayes optimal.

We note the remarkable simplicity of the above activation functions yielding infinitely wide and deep networks that achieve consistency. In particular, for $d \geq 2$, Corollary **1** gives activations are simply cubic polynomials, and Corollary **2** gives sinusoidal activations where the frequency a increases with dimension (e.g., $a^2 \approx 2.676$ leads to consistency for $d = 2$ and $a^2 \approx 6.135$ leads to consistency for $d = 3$). Lastly, we note that our results are easily extended to the case where data have density on a submanifold of $\mathcal{S}_+^d$ by simply selecting α to be the dimension of the data manifold in Theorem **1**.

Case 2 ($A = 0, A' = 0$) Networks Implement 1-NN. We now identify conditions on the activation function under which infinitely wide and deep networks implement the 1-NN classifier.

Theorem 3. *Let m_n denote the classifier in Eq.* ***3*** *corresponding to training an infinitely wide and deep network with activation function $\phi(\cdot)$ on n training examples. For $z \sim \mathcal{N}(0,1)$, define $A = \mathbb{E}[\phi(z)]$ and $A' = \mathbb{E}[\phi'(z)]$. If $A = A' = 0$, then $m_n(x)$ implements 1-NN classification for almost all $x \in \mathcal{S}_+^d$.*

The proof of Theorem **3** is provided in *SI Appendix*, Appendix E. The proof strategy is to show that the value of the kernel between a test example and its nearest training example dominates the prediction as $L \to \infty$. In particular, assuming without loss of generality that $x^T x^{(1)} > x^T x^{(j)}$ for $j \in \{2, 3, \ldots, n\}$, we prove that:

$$\lim_{L \to \infty} \frac{K^{(L)}(x, x^{(j)})}{K^{(L)}(x, x^{(1)})} = 0.$$

As a result, after rescaling by $K^{(L)}(x, x^{(1)})$, we obtain that $m_n(x) = \operatorname{sign}(y^{(1)})$. We note that this proof is analogous to the standard proof that the Gaussian kernel $K(x, \tilde{x}) = \exp\left(-\gamma \|x - \tilde{x}\|^2\right)$ converges to the 1-NN classifier as $\gamma \to \infty$.

Case 3 ($A \neq 0$) Networks Implement Majority Vote Classifiers. We now analyze infinitely wide and deep networks when the activation function satisfies $\mathbb{E}[\phi(z)] \neq 0$ for $z \sim \mathcal{N}(0,1)$. In this setting, we establish conditions under which the infinitely wide and deep network implements majority vote classification, i.e., the prediction on test samples is simply the label of the class with the greatest representation in the training set. More precisely, the following proposition (proof in *SI Appendix*, Appendix F) implies that when the infinite depth NTK is a constant nonzero value for any two nonequal inputs, the resulting classifier is the majority vote classifier.

Proposition 1. *Let m_n denote the classifier in Eq.* ***3*** *corresponding to training an infinitely wide and deep network with activation function $\phi(\cdot)$ on n training examples such that the sum of the labels $y^{(i)}$ is not 0. For any $x, \tilde{x} \in \mathcal{S}_+^d$ with $x \neq \tilde{x}$, if the NTK $K^{(L)}$ satisfies*

$$\lim_{L \to \infty} \frac{K^{(L)}(x, \tilde{x})}{C(L)} = C_1 \text{ and } \lim_{L \to \infty} \frac{K^{(L)}(x, x)}{C(L)} \neq C_1, \qquad [4]$$

with $C_1 > 0$ and $0 < C(L) < \infty$ for any L, then m_n implements the majority vote classifier, i.e.,

$$m_n(x) = \operatorname{sign}\left(\sum_{i=1}^{n} y^{(i)}\right).$$

We now analyze which activation functions satisfy Eq. **4**. As described in ref. 31–34, under case 3, the value of $B' = \mathbb{E}[\phi'(z)^2]$

[#] Let $m(x) = \operatorname*{arg\,max}_{\tilde{y} \in \{-1,1\}} \mathbb{P}(y = \tilde{y} | x)$ denote the Bayes optimal classifier. Let X_n denote the training data in $\mathcal{S}_+^d$, let $f(\cdot)$ denote the density on $\mathcal{S}_+^d$, and let $m_{X_n} := m_n$ denote the classifier in Eq. **3**. Formally, Theorem **2** implies that at almost all $x \in \mathcal{S}_+^d$ with $f(x) > 0$ and for any $\epsilon > 0$, $m_{X_n}(x)$ converges to $m(x)$ in probability as $n \to \infty$, i.e.,

$$\lim_{n \to \infty} \mathbb{P}_{X_n}\left(\left|m_{X_n}(x) - m(x)\right| > \epsilon\right) = 0.$$

This is the same notion of consistency, i.e., weak consistency, established for the Hilbert kernel estimator in ref. 13.

[‖] The closed forms for these polynomials are as follows: $h_2(x) = \frac{x^2 - 1}{\sqrt{2}}$, $h_3 = \frac{x^3 - 3x}{\sqrt{6}}$, and $h_7(x) = \frac{x^7 - 21x^5 + 105x^3 - 105x}{12\sqrt{35}}$.

for $z \sim \mathcal{N}(0, 1)$ determines the fixed point attractors of $K^{(L)}$ as $L \to \infty$. Thus, the infinite depth behavior under case 3 can be broken down into three cases based on the value of B'. Using the terminology from ref. 31, these cases are:

$$(i)\ B' > 1 \text{ (Chaotic Phase)},$$
$$(ii)\ B' < 1 \text{ (Ordered Phase)},$$
$$(iii)\ B' = 1 \text{ (Edge of Chaos)}.$$

In Lemma **5** in *SI Appendix*, Appendix G, we demonstrate that in the chaotic phase, the resulting infinite depth NTK satisfies the conditions of Proposition **1** and thus implements the majority vote classifier. In Lemma **6** in *SI Appendix*, Appendix G, we similarly show that in the ordered phase, the infinite depth NTK also corresponds to the majority vote classifier.[**] The remaining case known as "edge of chaos" has been analyzed in prior works for specific activation functions; for example, the NTK for networks with ReLU activation satisfies Eq. **4** with $C_1 = \frac{1}{4}$ and $C(L) = L + 1$ (22, 25). Hence, by Proposition **1**, the corresponding infinite depth classifier for ReLU networks corresponds to the majority vote classifier.

Classifiers Implemented by Infinitely Wide and Deep Networks with Standard Activation Functions. We now discuss activation functions commonly used in practice and the classifiers implemented by infinitely wide and deep networks with such activation functions. The conditions of Theorem **1** are satisfied by several commonly used activation functions in practice including sine, erf, tanh, and hard tanh. However, as we prove in *SI Appendix*, Appendix H the order of singularity, α, in Theorem **1** for all of these activation functions is near 0.5, which is the value of α that is required for consistency for data on the unit circle.

On the other hand, activation functions including ReLU, sigmoid, cosid (i.e., $\cos x - x$) (35), and swish (i.e., $\frac{x}{1+e^{-x}}$) (36) satisfy the conditions of Proposition **1** and, thus, implement majority vote.

In *SI Appendix*, Appendix I and Fig. S3, we provide experiments across several data distributions in which we compare the performance of infinitely wide and deep classifiers using standard activation functions such as ReLU, erf, and sine, which have closed forms for the NTK, against those that lead to consistent classifiers according to Theorem **2** above. In all cases, we observe a strong accordance between our experiments and theoretical results, showing that infinitely wide and deep networks using standard activation functions are far from consistent.

Practical relevance of our results. While this work derives activation functions that lead to infinitely wide and deep networks that are consistent for fixed data dimension as the number of training samples approaches infinity, we demonstrate the practical value of the derived activation functions in *SI Appendix*, Appendix J and Fig. S4 on a variety of benchmarking datasets in the context of finitely deep networks and finite sample sizes, concentrating in particular on the small-sample regime. Namely, in *SI Appendix*, Appendix J and Fig. S4, we show that grid searching over the activation functions provided in Corollaries **1** and **2** lead to improved performance over standard classifiers including 179 models from (37), fully connected ReLU networks, as well as ReLU NTKs from (38) on a variety of benchmarking datasets including i) the 90 benchmarking classification tasks in the small-sample regime (with fewer than 5000 training samples) considered in ref. 38 and ii) the 3 classification tasks in the small-dimensional large-sample regime (with fewer than 15 features and greater than 10,000 training samples) considered in ref. 37.

2. Outline of Proof Strategy for Theorems 1 and 2

In the following, we outline the proof strategy for our main results. This involves analyzing infinitely wide and deep networks via the limiting NTK kernel given by $K^{(L)}$ as the number of hidden layers $L \to \infty$. As shown in ref. 18, $K^{(L)}$ can be written recursively in terms of $K^{(L-1)}$ and the so-called dual activation function, which was introduced in ref. 39.

Definition 2: Let $\phi : \mathbb{R} \to \mathbb{R}$ be an activation function satisfying $\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\phi(x)^2] < \infty$. Its **dual activation function** $\check{\phi} : [-1, 1] \to \mathbb{R}$ is given by

$$\check{\phi}(z) = \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\Lambda)}[\phi(u)\phi(v)], \quad \text{where } \Lambda = \begin{bmatrix} 1 & z \\ z & 1 \end{bmatrix}.$$

While all quantities in our theorems are stated in terms of activation functions, these can be restated in terms of dual activations as follows:

$$A^2 = \check{\phi}(0) \quad \text{and} \quad (A')^2 = \check{\phi}'(0) \quad \text{and} \quad B' = \check{\phi}'(1).$$

Assuming that ϕ is normalized such that $\check{\phi}(1) = 1$,[††] the recursive formula for the NTK of a deep fully connected network for data on the unit sphere was described in ref. 18 and 40 in terms of dual activation functions as follows.

A. Recursive Formula for the NTK. Let $f^{(L)}(x; \mathbf{W})$ denote a fully connected neural network with L hidden layers and activation $\phi(\cdot)$. For $x, \tilde{x} \in \mathcal{S}^d$, let $z = x^T\tilde{x}$. Then, $K^{(L)}$ is radial, i.e., $K^{(L)}(x, \tilde{x}) = K^{(L)}(z)$, with $K^{(0)}(z) = z$ and

$$K^{(L)}(z) = \check{\phi}^{(L)}(z) + K^{(L-1)}(z)\check{\phi}'\left(\check{\phi}^{(L-1)}(z)\right), \qquad [5]$$

where $\check{\phi}^{(L)}(z) = \check{\phi}(\check{\phi}^{(L-1)}(z))$ with $\check{\phi}^{(0)}(z) = \check{\phi}(z)$, and $\check{\phi}'(\cdot)$ denotes the derivative of $\check{\phi}(\cdot)$.

We utilize the dynamical system in Eq. **5** to analyze the behavior of $K^{(L)}(\cdot)$ as $L \to \infty$. Theorem **1** implies that upon normalization by $(L + 1)\check{\phi}'(0)^L$, this dynamical system converges to a singular kernel with singularity of order $\alpha = -\log\left(\check{\phi}'(0)\right) / \log\left(\check{\phi}'(1)\right)$. We now present a sketch of the proof of this result.

We first derive the order of the singularity upon iteration of $\check{\phi}$ since, as we show in *SI Appendix*, Appendix B, the order of the singularity of the infinite depth NTK is the same as that of the iterated $\check{\phi}$. Since we consider that data in $\mathcal{S}^d_+$, $\check{\phi}(\cdot)$ is a function defined on the unit interval $[0, 1]$, understanding the properties of infinitely wide and deep networks reduces to understanding the properties of iterating a function on the unit interval. To provide intuition around how the iteration of a function on the unit interval can give rise to a function with a singularity, we discuss iterating a piecewise linear function as an illuminating example; Fig. 2 for a visualization.

[**] More precisely, we consider the behavior of the infinite depth classifier under ridge regularization, as the regularization term approaches 0.

[††] Such normalization is always possible for any activation function satisfying $\mathbb{E}[\phi(z)^2] < \infty$ for $z \sim \mathcal{N}(0, 1)$ and has been used in various works before including (22, 24, 25, 33, 40, 41).

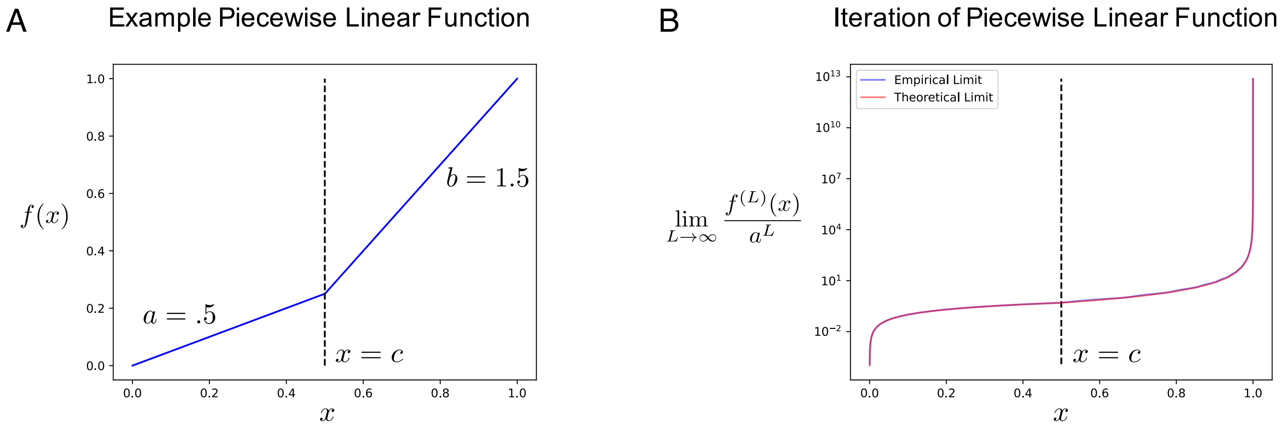

Fig. 2. Iteration of a piecewise linear function on a unit interval leads to a function with a singularity at $x = 1$, upon appropriate normalization. (*A*) We consider the piecewise linear function $f(x)$ given by $1 - b(1 - x)$ on $(c, 1]$ and ax on $[0, c]$, where $a = .5, b = 1.5$ and $c = \frac{b-1}{b-a}$. (*B*) We observe that upon iterating $f(\cdot)$ numerically to the limit of machine precision, the resulting function strongly agrees with the theoretical limit of Lemma **1** given by a function with singularity of order $-\log_b a \approx 1.7$.

Lemma 1. *For $0 < a < 1$ and $b > 1$, let $f : [0, 1] \to \mathbb{R}$ and $c = \frac{b-1}{b-a}$ such that*

$$f(x) = \begin{cases} ax & \text{if } x \in [0, c] \\ 1 - b(1 - x) & \text{if } x \in (c, 1] \end{cases}.$$

Then,

$$\lim_{L\to\infty} \frac{f^{(L)}(x)}{a^L} = \frac{R(x)}{(1 - x)^{-\log_b a}},$$

where $R(x)$ is nonnegative, bounded from above and bounded away from 0 around $x = 1$.

Proof: For any $x \in [0, c]$, we necessarily have:

$$\lim_{L\to\infty} \frac{f^{(L)}(x)}{a^L} = \lim_{L\to\infty} \frac{a^L x}{a^L} = x.$$

Now, for fixed $x \in (c, 1)$, since $x = 0$ is an attractive fixed-point of f, let L_0 denote the smallest integer such that $f^{(L_0)}(x) \leq c$. Hence, since $f^{(L_0)}(x) \in [0, c]$, we obtain:

$$\lim_{L\to\infty} \frac{f^{(L)}(x)}{a^L} = \lim_{L\to\infty} \frac{f^{(L-L_0)}(f^{(L_0)}(x))}{a^{L-L_0}} \frac{1}{a^{L_0}} = f^{(L_0)}(x)a^{-L_0}. \qquad [6]$$

We next solve for L_0 by analyzing the iteration of $g(x) := 1 - b(1 - x)$. In particular, we observe that $g^{(L)}(x) = 1 - b^L(1 - x)$, and thus L_0 is given by

$$\begin{aligned} &1 - b^{L_0}(1 - x) \leq c \\ \implies &L_0 = \left\lceil \log_a \left(\frac{1 - x}{1 - c} \right)^{-\log_b a} \right\rceil \\ \implies &a^{-L_0} \in \left[\left(\frac{1 - c}{1 - x} \right)^{-\log_b a}, \frac{1}{a} \left(\frac{1 - c}{1 - x} \right)^{-\log_b a} \right]. \end{aligned}$$

Hence, by Eq. **6**, we conclude that for $x \in (c, 1)$, it holds that

$$\lim_{L\to\infty} \frac{f^{(L)}(x)}{a^L} = \frac{R(x)}{(1 - x)^{-\log_b a}},$$

where $R(x)$ is nonnegative, bounded from above and bounded away from 0 around $x = 1$, which completes the proof. □

In *SI Appendix*, Appendix B, we extend this analysis to the iteration of dual activations on the unit interval, thereby establishing the order of a singularity obtained by iterating dual activation functions. We then show that this order equals the order of the singularity given by the infinite depth NTK.

Next, we discuss the proof strategy for Theorem **2**, which establishes conditions on the activation function under which infinitely wide and deep networks achieve consistency in the classification setting. The proof builds on results in ref. 13 characterizing the consistency of singular kernel smoothers of the form

$$g(x) = \frac{\sum_{i=1}^n y^{(i)} K(x^{(i)}, x)}{\sum_{i=1}^n K(x^{(i)}, x)}, \text{ where } K(x^{(i)}, x) = \frac{1}{\|x - x^{(i)}\|^{\alpha}}.$$

In particular, it is shown that if $\alpha = d$, then $g(x)$ achieves consistency. Since Theorem **1** establishes conditions under which the infinite depth NTK implements a singular kernel, to complete the proof, we show that infinitely wide and deep classifiers achieve consistency by 1) showing that the classifier m_n implements a singular kernel smoother and 2) selecting ϕ such that $\alpha = d$ for the corresponding singular kernel.

3. Discussion

In this work, we identified and constructed explicit neural networks that achieve consistency for classification when trained using standard procedures. Furthermore, we provided a taxonomy characterizing the behavior of infinitely wide and deep neural network classifiers. Namely, we showed that these models implement one of the following three well-known types of classifiers: 1) 1-NN (test predictions are given by the label of the nearest training example); 2) majority vote (test predictions are given by the label of the class with the greatest representation in the training set); or 3) singular kernel classifiers (a set of classifiers containing those that achieve consistency). We conclude by discussing implications of our work and future extensions.

A. Benefit of Depth in Neural Networks. An emerging trend in machine learning is that larger neural networks capable of interpolating (i.e., perfectly fitting) the training data can generalize to test data (6–8). While the size of neural networks can be increased through width or depth, works such as (6, 7) primarily identified a benefit to increasing network width. Indeed, it remained unclear whether there was any benefit in using extremely deep networks. A line of prior work analyzed the impact of selecting activation functions and initializations to enable the training of deep networks (31, 32, 42), while other works (24, 43, 44) empirically demonstrated that drastically increasing depth in networks with ReLU or tanh activation could lead to worse performance. In this work, we established a remarkable benefit of very deep networks by proving that they achieve consistency with a careful choice of activation function. In line with previous empirical findings, we proved that deep networks with activations such as ReLU or tanh do not achieve consistency.

B. Regression versus Classification. Our results demonstrate the benefit of using infinitely wide and deep networks for classification tasks. We note that this is in stark contrast to the regression setting, where infinitely deep and wide neural networks are far from consistent, as they simply predict a nonnegative, near-zero constant almost everywhere (22, 25). Thus, our work provides concrete examples of neural networks that are effective for classification but not regression. A key difference between regression and classification is that classification requires only the sign of the prediction. In particular, as we show in this work, the sign of the prediction of an infinitely wide and deep network can be meaningful for classification even though the prediction itself is close to 0.

C. Edge of Chaos Regime. An interesting class of models that are only partially characterized by our taxonomy corresponds to networks with activations in the edge of chaos regime, i.e., when the activation function, $\phi(\cdot)$ satisfies $\mathbb{E}[\phi(z)] \neq 0$ and $\mathbb{E}[\phi'(z)^2] = 1$ for $z \sim \mathcal{N}(0, 1)$. We proved that all activations in this class that have been described so far (22, 25), including the popular ReLU activation, give rise to infinitely wide and deep networks that implement the majority vote classifier. While it appears that all activations in this class lead to the majority vote classifier, it remains open to understand whether there exist other activations in this regime that implement alternative classifiers. Moreover, works analyzing the edge of chaos regime typically consider infinite width networks with bias terms. While these bias terms are often set to avoid exponential convergence of predictions with increasing depth, they can be detrimental in the classification setting. For example, the work of ref. 25 shows that with appropriate bias, the tanh activation function leads to an infinitely wide and deep network that satisfies our Proposition **1** and thus implements majority vote. However, without the bias, this activation function satisfies Theorem **1** and thus leads to a singular kernel classifier. It is an interesting question to characterize how the addition of bias terms may influence our taxonomy.

D. Finite vs. Infinite Neural Networks. In this work, we identified and constructed infinitely wide and deep classifiers that achieve consistency. In particular, our results imply weak consistency of infinitely wide and deep networks, which means that these models converge in probability to the Bayes optimal classifier as the number of training samples approaches infinity. While recent work (14) demonstrated that finite depth NTKs are not universally consistent, i.e., they are not consistent for arbitrary distributions, it remains open as to whether these models are consistent in a weaker sense. An important next question is to understand whether interpolating neural networks that are finitely wide and deep can achieve consistency for classification and provide specific activation functions to do so. Some evidence in this direction is given by recent work demonstrating that sufficiently wide and deep ReLU networks correctly classify points on disjoint curves on a sphere (45). We also note that Bayes consistency considers the setting when the number of training examples approaches infinity. Another natural next step is to characterize the number of training examples needed for infinitely wide and deep classifiers to reasonably approximate the Bayes optimal classifier. Recent work (46) identified a slow (logarithmic) rate of convergence for singular kernel classifiers, thereby implying that many training examples are needed for these models to be effective in practice. An important open direction of future work is thus to determine not only whether finitely wide and deep networks are Bayes optimal for classification but also whether these models require fewer samples to perform well in practice.

Data, Materials, and Software Availability. There are no data underlying this work.

ACKNOWLEDGMENTS. A.R. and C.U. were partially supported by NSF (DMS-1651995), ONR (N00014-17-1-2147 and N00014-18-1-2765), the MIT-IBM Watson AI Lab, the Eric and Wendy Schmidt Center at the Broad Institute, and a Simons Investigator Award (to C.U.). M.B. acknowledges support from NSF IIS-1815697, NSF DMS-2031883/Simons Foundation Award 81463 and the TILOS institute (NSF CCF-2112665).

1. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition" in *Computer Vision and Pattern Recognition*, (IEEE, 2016), pp. 770–778.
2. T. Brown *et al*., "Language models are few-shot learners" in *Advances in Neural Information Processing Systems*, (Curran Associates, Red Hook, NY, 2020), vol. 33, pp. 1877–1901.
3. K. Tunyasuvunakool *et al*., Highly accurate protein structure prediction for the human proteome. *Nature* **596**, 1–9 (2021).
4. A. Christmann, I. Steinwart, *Support Vector Machines* (Springer, 2008).
5. L. Devroye, L. Gyorfi, G. Lugosi, *A Probablistic Theory of Pattern Recognition* (Springer Verlag, 1996), vol. 31.
6. M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15849–15854 (2019).
7. P. Nakkiran *et al*., "Deep double descent: Where bigger models and more data hurt" in *International Conference in Learning Representations* (2020).
8. C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, "Understanding deep learning requires rethinking generalization" in *International Conference on Learning Representations* (2017).
9. T. Cover, P. Hart, Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27 (1967).
10. M. Belkin, D. Hsu, P. Mitra, "Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2018), vol. 31.
11. M. Belkin, A. Rakhlin, A. Tsybakov, "Does data interpolation contradict statistical optimality?' in *International Conference on Artificial Intelligence and Statistics*, A. Storkey, F. Perez-Cruz, Eds. (Proceedings of Machine Learning Research, 2018), vol. 84, pp. 1611–1619.
12. A. Rakhlin, X. Zhai, "Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon" in *Conference on Learning Theory*, A. Beygelzimer, D. Hsu, Eds. (Proceedings of Machine Learning Research, 2019), vol. 99, 2595–2623.
13. L. Devroye, L. Györfi, A. Krzyzàk, The Hilbert kernel regression estimate. *J. Multivar. Anal.* **65**, 209–227 (1998).
14. D. Beaglehole, M. Belkin, P. Pandit, Kernel ridgeless regression is inconsistent in low dimensions. arXiv [Preprint] (2022). http://arxiv.org/abs/2205.13525 (Accessed 7 July 2022).
15. A. Faragó, G. Lugosi, Strong universal consistency of neural network classifiers. *IEEE Trans. Inf. Theory* **39** (1993).
16. G. Cybenko, Approximation by superposition of Sigmoidale function. *Math. Control Signal Syst.* **2**, 304–314 (1989).
17. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
18. A. Jacot, F. Gabriel, C. Hongler, "Neural Tangent Kernel: Convergence and generalization in neural networks" in *Advances in Neural Information Processing Systems*, (Curran Associates, Red Hook, NY, 2018), vol. 31, pp. 8571–8580.
19. J. Lee *et al*., "Wide neural networks of any depth evolve as linear models under gradient descent" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2019), vol. 32, pp. 8572–8583.
20. C. Liu, L. Zhu, M. Belkin, "On the linearity of large non-linear models: When and why the tangent kernel is constant" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2020), vol. 33, pp. 15954–15964.

21. C. Liu, L. Zhu, M. Belkin, Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Appl. Comput. Harmon. Anal.* **59** (2022).
22. K. Huang, Y. Wang, M. Tao, T. Zhao, "Why do deep residual networks generalize better than deep feedforward networks? – A Neural Tangent Kernel perspective" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2020), vol. 33, pp. 2698–2709.
23. A. Radhakrishnan, G. Stefanakis, M. Belkin, C. Uhler, Simple, fast, and flexible framework for matrix completion with infinite width neural networks. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2115064119 (2022).
24. L. Xiao, J. Pennington, S. Schoenholz, "Disentangling trainability and generalization in deep learning" in *International Conference on Machine Learning* (2019).
25. S. Hayou, A. Doucet, J. Rousseau, Mean-field behaviour of Neural Tangent Kernel for deep neural networks. arXiv [Preprint] (2021). http://arxiv.org/abs/1905.13654 (Accessed 7 July 2022).
26. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning* (Springer, 2001), vol. 1.
27. B. Schölkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, 2002).
28. S. Arora *et al*., "On exact computation with an infinitely wide neural net" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2019), vol. 32, pp. 8141–8150.
29. J. Lee *et al*., "Finite versus infinite neural networks: An empirical study" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2020), vol. 33, pp. 15156–15172.
30. R. Novak *et al*., "Neural Tangents: Fast and easy infinite neural networks in Python" in *International Conference on Learning Representations* (2020).
31. B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, S. Ganguli, "Exponential expressivity in deep neural networks through transient chaos" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2016), vol. 29, pp. 3360–3368.
32. G. Yang, S. Schoenholz, "Mean field residual networks: On the edge of chaos" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2017), vol. 30, pp. 7103–7114.
33. S. Hayou, A. Doucet, J. Rousseau, "On the impact of the activation function on deep neural networks training" in *International Conference on Machine Learning*, K. Chaudhuri, R. Salakhutdinov, Eds. (Proceedings of Machine Learning Research, 2019), vol. 97, pp. 2672–2680.
34. J. Lee *et al*., "Deep neural networks as Gaussian processes" in *International Conference on Learning Representations* (2017).
35. S. Eger, P. Youssef, I. Gurevych, "Is it time to Swish? Comparing deep learning activation functions across NLP tasks" in *Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, 2018), pp. 4415–4424.
36. P. Ramachandran, B. Zoph, Q. V. Le, "Searching for activation functions" in *International Conference on Learning Representations* (2017).
37. M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**, 3133–3181 (2014).
38. S. Arora *et al*., "Harnessing the power of infinitely wide deep nets on small-data tasks" in *International Conference on Learning Representations* (2020).
39. A. Daniely, R. F. Frostig, Y. Singer, "Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2016), vol. 29, pp. 2253–2261.
40. A. Geifman *et al*., "On the similarity between the Laplace and Neural Tangent Kernels" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2020), vol. 33, pp. 1451–1461.
41. T. Liang, H. Tran-Bach, Mehler's formula, branching process, and compositional kernels of deep neural networks. *J. Am. Stat. Assoc.* **117**, 1–35 (2020).
42. M. Murray, V. Abrol, J. Tanner, Activation function design for deep networks: Linearity and effective initialisation. arXiv [Preprint] (2021). http://arxiv.org/abs/2105.07741 (Accessed 7 July 2022).
43. E. Nichani, A. Radhakrishnan, C. Uhler, "Increasing depth leads to U-shaped test risk in over-parameterized convolutional networks" in *International Conference on Machine Learning Workshop on Over-parameterization: Pitfalls and Opportunities* (2021).
44. L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. Schoenholz, J. Pennington, "Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks" in *International Conference on Machine Learning* J. Dy, A. Krause, Eds. (Proceedings of Machine Learning Research, 2018), vol. 80, pp. 5393–5402.
45. T. Wang, S. Buchanan, D. Gilboa, J. Wright, "Deep networks provably classify data on curves" in *Advances in Neural Information Processing Systems* (Curran Associates, Red Hook, NY, 2021), vol. 34, pp. 28940–28953.
46. P. P. Mitra, C. Sire, Parameter-free statistically consistent interpolation: Dimension-independent convergence rates for Hilbert kernel regression. arXiv [Preprint] (2021). http://arxiv.org/abs/2106.03354 (Accessed 7 July 2022).