Reconciling modern machine-learning practice and the classical bias–variance trade-off

Mikhail Belkin[a,b,1], Daniel Hsu[c], Siyuan Ma[a], and Soumik Mandal[a]

[a]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210; [b]Department of Statistics, The Ohio State University, Columbus, OH 43210; and [c]Computer Science Department and Data Science Institute, Columbia University, New York, NY 10027

Edited by Peter J. Bickel, University of California, Berkeley, CA, and approved July 2, 2019 (received for review February 21, 2019)

Breakthroughs in machine learning are rapidly changing science and society, yet our fundamental understanding of this technology has lagged far behind. Indeed, one of the central tenets of the field, the bias–variance trade-off, appears to be at odds with the observed behavior of methods used in modern machine-learning practice. The bias–variance trade-off implies that a model should balance underfitting and overfitting: Rich enough to express underlying structure in data and simple enough to avoid fitting spurious patterns. However, in modern practice, very rich models such as neural networks are trained to exactly fit (i.e., interpolate) the data. Classically, such models would be considered overfitted, and yet they often obtain high accuracy on test data. This apparent contradiction has raised questions about the mathematical foundations of machine learning and their relevance to practitioners. In this paper, we reconcile the classical understanding and the modern practice within a unified performance curve. This "double-descent" curve subsumes the textbook U-shaped bias–variance trade-off curve by showing how increasing model capacity beyond the point of interpolation results in improved performance. We provide evidence for the existence and ubiquity of double descent for a wide spectrum of models and datasets, and we posit a mechanism for its emergence. This connection between the performance and the structure of machine-learning models delineates the limits of classical analyses and has implications for both the theory and the practice of machine learning.

machine learning | bias–variance trade-off | neural networks

Machine learning has become key to important applications in science, technology, and commerce. The focus of machine learning is on the problem of prediction: Given a sample of training examples $(x_1, y_1), \ldots, (x_n, y_n)$ from $\mathbb{R}^d \times \mathbb{R}$, we learn a predictor $h_n \colon \mathbb{R}^d \to \mathbb{R}$ that is used to predict the label y of a new point x, unseen in training.

The predictor h_n is commonly chosen from some function class $\mathcal{H}$, such as neural networks with a certain architecture, using empirical risk minimization (ERM) and its variants. In ERM, the predictor is taken to be a function $h \in \mathcal{H}$ that minimizes the empirical (or training) risk $\frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i)$, where ℓ is a loss function, such as the squared loss $\ell(y', y) = (y' - y)^2$ for regression or 0–1 loss $\ell(y', y) = \mathbb{1}_{\{y' \neq y\}}$ for classification.

The goal of machine learning is to find h_n that performs well on new data, unseen in training. To study performance on new data (known as generalization), we typically assume the training examples are sampled randomly from a probability distribution P over $\mathbb{R}^d \times \mathbb{R}$ and evaluate h_n on a new test example (x, y) drawn independently from P. The challenge stems from the mismatch between the goals of minimizing the empirical risk (the explicit goal of ERM algorithms, optimization) and minimizing the true (or test) risk $\mathbb{E}_{(x,y)\sim P}[\ell(h(x), y)]$ (the goal of machine learning).

Conventional wisdom in machine learning suggests controlling the capacity of the function class $\mathcal{H}$ based on the bias–variance trade-off by balancing underfitting and overfitting (cf. refs. 1 and 2): 1) If $\mathcal{H}$ is too small, all predictors in $\mathcal{H}$ may underfit the training data (i.e., have large empirical risk) and hence predict poorly on new data. 2) If $\mathcal{H}$ is too large, the empirical risk minimizer may overfit spurious patterns in the training data, resulting in poor accuracy on new examples (small empirical risk but large true risk).

The classical thinking is concerned with finding the "sweet spot" between underfitting and overfitting. The control of the function class capacity may be explicit, via the choice of $\mathcal{H}$ (e.g., picking the neural network architecture), or it may be implicit, using regularization (e.g., early stopping). When a suitable balance is achieved, the performance of h_n on the training data is said to generalize to the population P. This is summarized in the classical U-shaped risk curve shown in Fig. 1*A* that has been widely used to guide model selection and is even thought to describe aspects of human decision making (3). The textbook corollary of this curve is that "a model with zero training error is overfit to the training data and will typically generalize poorly" (ref. 2, p. 221), a view still widely accepted.

However, practitioners routinely use modern machine-learning methods, such as large neural networks and other nonlinear predictors that have very low or zero training risk. Despite the high function class capacity and near-perfect fit to training data, these predictors often give very accurate predictions on new data. Indeed, this behavior has guided a best practice in deep learning for choosing neural network architectures, specifically that the network should be large enough to permit effortless zero-loss training (called interpolation) of the training data (4). Moreover, in direct challenge to the bias–variance trade-off philosophy, recent empirical evidence indicates that neural

Significance

While breakthroughs in machine learning and artificial intelligence are changing society, our fundamental understanding has lagged behind. It is traditionally believed that fitting models to the training data exactly is to be avoided as it leads to poor performance on unseen data. However, powerful modern classifiers frequently have near-perfect fit in training, a disconnect that spurred recent intensive research and controversy on whether theory provides practical insights. In this work, we show how classical theory and modern practice can be reconciled within a single unified performance curve and propose a mechanism underlying its emergence. We believe this previously unknown pattern connecting the structure and performance of learning architectures will help shape design and understanding of learning algorithms.

Author contributions: M.B., D.H., S. Ma, and S. Mandal designed research, performed research, analyzed data, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: mbelkin@cse.ohio-state.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1903070116/-/DCSupplemental.

Published online July 24, 2019.

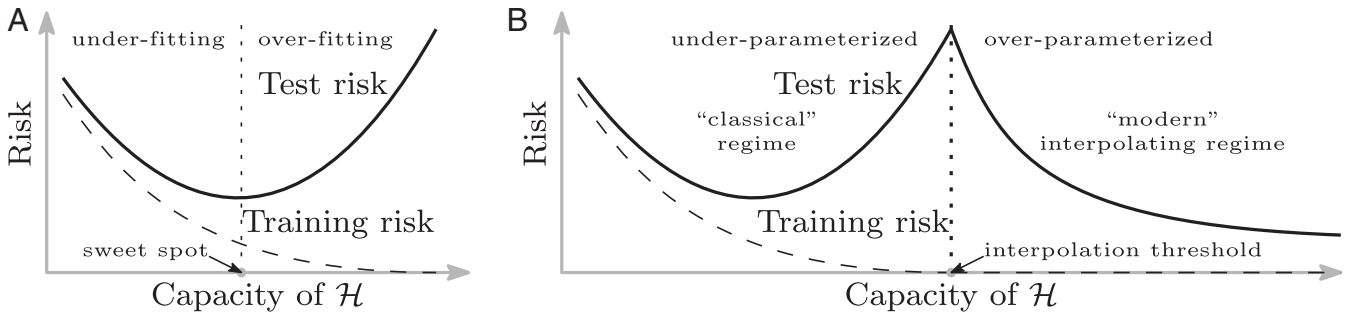

Fig. 1. Curves for training risk (dashed line) and test risk (solid line). (*A*) The classical U-shaped risk curve arising from the bias–variance trade-off. (*B*) The double-descent risk curve, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behavior from using high-capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

networks and kernel machines trained to interpolate the training data obtain near-optimal test results even when the training data are corrupted with high levels of noise (5, 6).

The main finding of this work is a pattern in how performance on unseen data depends on model capacity and the mechanism underlying its emergence. This dependence, empirically witnessed with important model classes including neural networks and a range of datasets, is summarized in the "double-descent" risk curve shown in Fig. 1*B*. The curve subsumes the classical U-shaped risk curve from Fig. 1*A* by extending it beyond the point of interpolation.

When function class capacity is below the "interpolation threshold," learned predictors exhibit the classical U-shaped curve from Fig. 1*A*. (In this paper, function class capacity is identified with the number of parameters needed to specify a function within the class.) The bottom of the U is achieved at the sweet spot which balances the fit to the training data and the susceptibility to overfitting: To the left of the sweet spot, predictors are underfitted, and immediately to the right, predictors are overfitted. When we increase the function class capacity high enough (e.g., by increasing the number of features or the size of the neural network architecture), the learned predictors achieve (near) perfect fits to the training data—i.e., interpolation. Although the learned predictors obtained at the interpolation threshold typically have high risk, we show that increasing the function class capacity beyond this point leads to decreasing risk, typically going below the risk achieved at the sweet spot in the "classical" regime.

All of the learned predictors to the right of the interpolation threshold fit the training data perfectly and have zero empirical risk. So why should some—in particular, those from richer functions classes—have lower test risk than others? The answer is that the capacity of the function class does not necessarily reflect how well the predictor matches the inductive bias appropriate for the problem at hand. For the learning problems we consider (a range of real-world datasets as well as synthetic data), the inductive bias that seems appropriate is the regularity or smoothness of a function as measured by a certain function space norm. Choosing the smoothest function that perfectly fits observed data is a form of Occam's razor: The simplest explanation compatible with the observations should be preferred (cf. refs. 7 and 8). By considering larger function classes, which contain more candidate predictors compatible with the data, we are able to find interpolating functions that have smaller norm and are thus "simpler." Thus, increasing function class capacity improves performance of classifiers.

Related ideas have been considered in the context of margins theory (7, 9, 10), where a larger function class $\mathcal{H}$ may permit the discovery of a classifier with a larger margin. While the margins theory can be used to study classification, it does not apply to regression and also does not predict the second descent beyond the interpolation threshold. Recently, there has been an emerging recognition that certain interpolating predictors (not based on ERM) can indeed be provably statistically optimal or near optimal (11, 12), which is compatible with our empirical observations in the interpolating regime.

In the remainder of this article, we discuss empirical evidence for the double-descent curve and the mechanism for its emergence and conclude with some final observations and parting thoughts.

Neural Networks

In this section, we discuss the double-descent risk curve in the context of neural networks.

Random Fourier Features. We first consider a popular class of nonlinear parametric models called random Fourier features (RFF) (13), which can be viewed as a class of 2-layer neural networks with fixed weights in the first layer. The RFF model family $\mathcal{H}_N$ with N (complex-valued) parameters consists of functions $h\colon \mathbb{R}^d \to \mathbb{C}$ of the form

$$h(x) = \sum_{k=1}^{N} a_k \phi(x; v_k) \quad \text{where} \quad \phi(x; v) := e^{\sqrt{-1}\langle v_k, x\rangle},$$

and the vectors $v_1, \ldots, v_N$ are sampled independently from the standard normal distribution in $\mathbb{R}^d$. (We consider $\mathcal{H}_N$ as a class of real-valued functions with $2N$ real-valued parameters by taking real and imaginary parts separately.) Note that $\mathcal{H}_N$ is a randomized function class, but as $N \to \infty$, the function class becomes a closer and closer approximation to the reproducing kernel Hilbert space (RKHS) corresponding to the Gaussian kernel, denoted by $\mathcal{H}_\infty$. While it is possible to directly use $\mathcal{H}_\infty$ [e.g., as is done with kernel machines (14)], the random classes $\mathcal{H}_N$ are computationally attractive to use when the sample size n is large but the number of parameters N is small compared with n.

Our learning procedure using $\mathcal{H}_N$ is as follows. Given data $(x_1, y_1), \ldots, (x_n, y_n)$ from $\mathbb{R}^d \times \mathbb{R}$, we find the predictor $h_{n,N} \in \mathcal{H}_N$ via ERM with squared loss. That is, we minimize the empirical risk objective $\frac{1}{n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$ over all functions $h \in \mathcal{H}_N$. When the minimizer is not unique (as is always the case when $N > n$), we choose the minimizer whose coefficients $(a_1, \ldots, a_N)$ have the minimum ℓ_2 norm. This choice of norm is intended as an approximation to the RKHS norm $\|h\|_{\mathcal{H}_\infty}$, which is generally difficult to compute for arbitrary functions in $\mathcal{H}_N$. For problems with multiple outputs (e.g., multiclass classification), we use functions with vector-valued outputs and the sum of the squared losses for each output.

In Fig. 2, we show the test risk of the predictors learned using $\mathcal{H}_N$ on a subset of the popular dataset of handwritten digits called MNIST. Fig. 2 also shows the ℓ_2 norm of the function coefficients, as well as the training risk. We see that for small values of N, the test risk shows the classical U-shaped curve consistent with the bias–variance trade-off, with a peak occurring at the interpolation threshold $N = n$. Some statistical analyses of RFF suggest choosing $N \propto \sqrt{n} \log n$ to obtain good test risk guarantees (15).

The interpolation regime connected with modern practice is shown to the right of the interpolation threshold, with $N \geq n$. The model class that achieves interpolation with fewest parameters ($N = n$ random features) yields the least accurate predictor. (In fact, it has no predictive ability for classification.) But as the number of features increases beyond n, the accuracy improves dramatically, exceeding that of the predictor corresponding to the bottom of the U-shaped curve. The plot also shows that the predictor $h_{n,\infty}$ obtained from $\mathcal{H}_\infty$ (the kernel machine) outperforms the predictors from $\mathcal{H}_N$ for any finite N.

What structural mechanisms account for the double-descent shape? When the number of features is much smaller than the sample size, $N \ll n$, classical statistical arguments imply that the training risk is close to the test risk. Thus, for small N, adding more features yields improvements in both the training and the test risks. However, as the number of features approaches n (the interpolation threshold), features not present or only weakly present in the data are forced to fit the training data nearly perfectly. This results in classical overfitting as predicted by the bias–variance trade-off and prominently manifested at the peak of the curve, where the fit becomes exact.

To the right of the interpolation threshold, all function classes are rich enough to achieve zero training risk. For the classes $\mathcal{H}_N$ that we consider, there is no guarantee that the most regular, smallest norm predictor consistent with training data (namely $h_{n,\infty}$, which is in $\mathcal{H}_\infty$) is contained in the class $\mathcal{H}_N$ for any finite N. But increasing N allows us to construct progressively better approximations to that smallest norm function. Thus, we expect to have learned predictors with largest norm at the interpolation threshold and for the norm of $h_{n,N}$ to decrease monotonically as N increases, thus explaining the second descent segment of the curve. This is what we observe in Fig. 2, and indeed $h_{n,\infty}$ has better accuracy than all $h_{n,N}$ for any finite N. Favoring small norm interpolating predictors turns out to be a powerful inductive bias on MNIST and other real and synthetic datasets (6). For noiseless data, we make this claim mathematically precise in *SI Appendix*.

Additional empirical evidence for the same double-descent behavior using other datasets is presented in *SI Appendix*. For instance, we demonstrate double descent for rectified linear unit (ReLU) random feature models, a class of ReLU neural networks with a setting similar to that of RFF. We also describe a simple synthetic model, which can be regarded as a 1D version of the RFF model, where we observe the same double-descent behavior.

Neural Networks and Backpropagation. In general multilayer neural networks (beyond RFF or ReLU random feature models), a learning algorithm will tune all of the weights to fit the training data, typically using versions of stochastic gradient descent (SGD), with backpropagation to compute partial derivatives. This flexibility increases the representational power of neural networks, but also makes ERM generally more difficult to implement. Nevertheless, as shown in Fig. 3, we observe that increasing the number of parameters in fully connected 2-layer neural networks leads to a risk curve qualitatively similar to that observed with RFF models. That the test risk improves beyond the interpolation threshold is compatible with the conjectured "small norm" inductive biases of the common training algorithms for neural networks (16, 17). We note that this transition from under- to overparameterized regimes for neural networks was also previously observed by refs. 18–21. In particular, ref. 21 draws a connection to the physical phenomenon of "jamming" in particle systems.

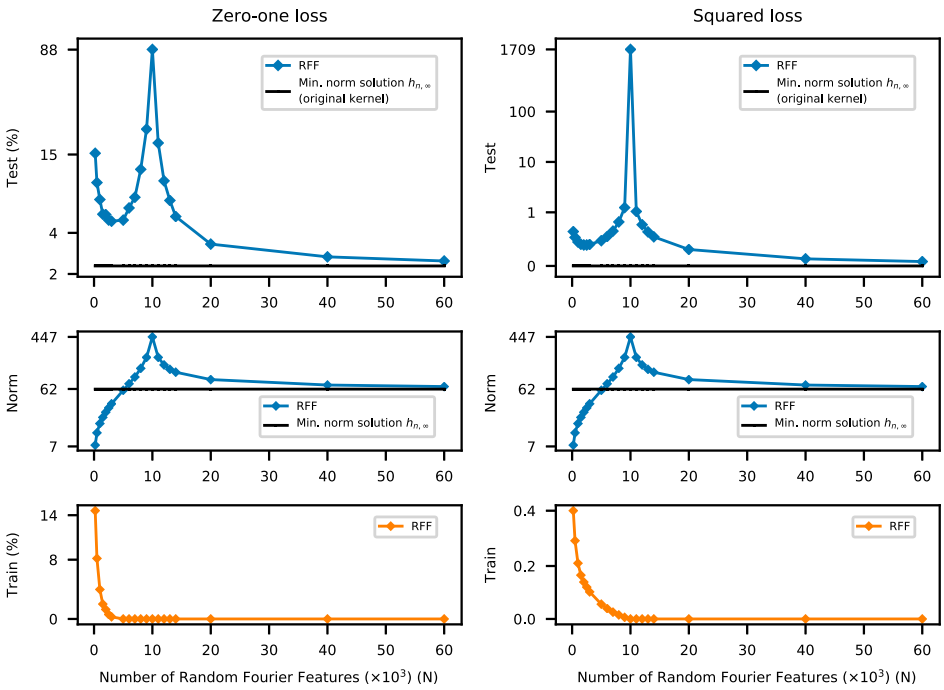

Fig. 2. Double-descent risk curve for the RFF model on MNIST. Shown are test risks (log scale), coefficient ℓ_2 norms (log scale), and training risks of the RFF model predictors $h_{n,N}$ learned on a subset of MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$.

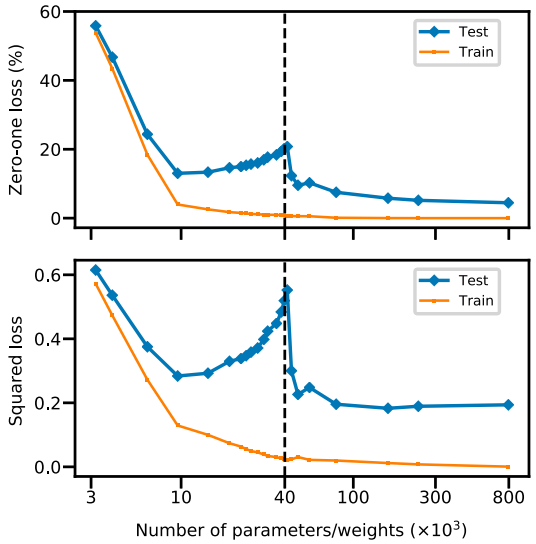

Fig. 3. Double-descent risk curve for a fully connected neural network on MNIST. Shown are training and test risks of a network with a single layer of H hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d + 1) \cdot H + (H + 1) \cdot K$. The interpolation threshold (black dashed line) is observed at $n \cdot K$.

The computational complexity of ERM with neural networks makes the double-descent risk curve difficult to observe. Indeed, in the classical underparameterized regime ($N \ll n$), the nonconvexity of the ERM optimization problem causes the behavior of local search-based heuristics, like SGD, to be highly sensitive to their initialization. Thus, if only suboptimal solutions are found for the ERM optimization problems, increasing the size of a neural network architecture may not always lead to a corresponding decrease in the training risk. This suboptimal behavior can lead to high variability in both the training and test risks that masks the double-descent curve.

It is common to use neural networks with an extremely large number of parameters (22). But to achieve interpolation for a single output (regression or 2-class classification) one expects to need at least as many parameters as there are data points. Moreover, if the prediction problem has more than one output (as in multiclass classification), then the number of parameters needed should be multiplied by the number of outputs. This is indeed the case empirically for neural networks shown in Fig. 3. Thus, for instance, datasets as large as ImageNet (23), which has $\sim 10^6$ examples and $\sim 10^3$ classes, may require networks with $\sim 10^9$ parameters to achieve interpolation; this is larger than many neural network models for ImageNet (22). In such cases, the classical regime of the U-shaped risk curve is more appropriate to understand generalization. For smaller datasets, these large neural networks would be firmly in the overparameterized regime, and simply training to obtain zero training risk often results in good test performance (5).

Additional results with neural networks are given in *SI Appendix*.

Decision Trees and Ensemble Methods

Does the double-descent risk curve manifest with other prediction methods besides neural networks? We give empirical evidence that the families of functions explored by boosting with decision trees and random forests also show similar generalization behavior to that of neural nets, both before and after the interpolation threshold.

AdaBoost and random forests have recently been investigated in the interpolation regime by ref. 24 for classification. In particular, they give empirical evidence that, when AdaBoost and random forests are used with maximally large (interpolating) decision trees, the flexibility of the fitting methods yields interpolating predictors that are more robust to noise in the training data than the predictors produced by rigid, noninterpolating methods (e.g., AdaBoost or random forests with shallow trees). This in turn is said to yield better generalization. The averaging of the (near) interpolating trees ensures that the resulting function is substantially smoother than any individual tree, which aligns with an inductive bias that is compatible with many real-world problems.

We can understand these flexible fitting methods in the context of the double-descent risk curve. Observe that the size of a decision tree (controlled by the number of leaves) is a natural way to parameterize the function class capacity: Trees with only 2 leaves correspond to 2-piecewise constant functions with an axis-aligned boundary, while trees with n leaves can interpolate n training examples. It is a classical observation that the U-shaped bias–variance trade-off curve manifests in many problems when the class capacity is considered this way (2). (The interpolation threshold may be reached with fewer than n leaves in many cases, but n is clearly an upper bound.) To further enlarge the function class, we consider ensembles (averages) of several interpolating trees.* So, beyond the interpolation threshold, we use the number of such trees to index the class capacity. When we view the risk curve as a function of class capacity defined in this hybrid fashion, we see the double-descent curve appear just as with neural networks (Fig. 4 and *SI Appendix*). We observe a similar phenomenon using L_2 boosting (26, 27), another popular ensemble method; the results are reported in *SI Appendix*.

Concluding Thoughts

The double-descent risk curve introduced in this paper reconciles the U-shaped curve predicted by the bias–variance trade-off and the observed behavior of rich models used in modern machine-learning practice. The posited mechanism that underlies its emergence is based on common inductive biases and hence can explain its appearance (and, we argue, ubiquity) in machine-learning applications.

We conclude with some final remarks.

Historical Absence. The double-descent behavior may have been historically overlooked on account of several cultural and practical barriers. Observing the double-descent curve requires a parametric family of spaces with functions of arbitrary complexity. The linear settings studied extensively in classical statistics usually assume a small, fixed set of features and hence fixed fitting capacity. Richer families of function classes are typically used in the context of nonparametric statistics, where smoothing and regularization are almost always used (28). Regularization, of all forms, can both prevent interpolation and change the effective capacity of the function class, thus attenuating or masking the interpolation peak.

The RFF model is a popular and flexible parametric family. However, these models were originally proposed as a computationally favorable alternative to kernel machines. This computational advantage over traditional kernel methods holds only for $N \ll n$, and hence models at or beyond the interpolation threshold are typically not considered.

The situation with general multilayer neural networks is slightly different and more involved. Due to the nonconvexity of the ERM optimization problem, solutions in the classical underparameterized regime are highly sensitive to initialization.

*These trees are trained in the way proposed in random forest except without bootstrap resampling. This is similar to the PERT method of ref. 25.

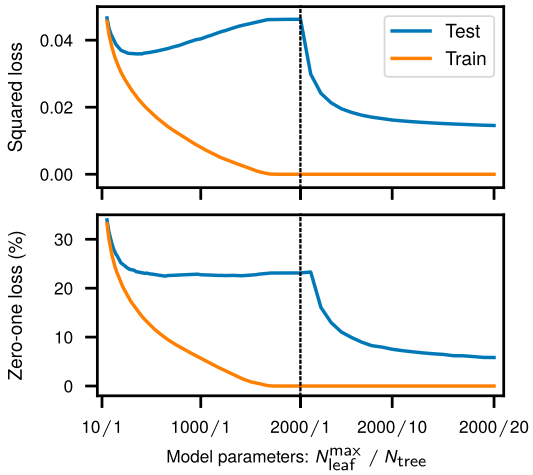

Fig. 4. Double-descent risk curve for random forests on MNIST. The double-descent risk curve is observed for random forests with increasing model complexity trained on a subset of MNIST ($n = 10^4$, 10 classes). Its complexity is controlled by the number of trees N_{tree} and the maximum number of leaves allowed for each tree $N_{\text{leaf}}^{\text{max}}$.

Moreover, as we have seen, the peak at the interpolation threshold is observed within a narrow range of parameters. Sampling of the parameter space that misses that range may lead to the misleading impression that increasing the size of the network simply improves performance. Finally, in practice, training of neural networks is typically stopped as soon as (an estimate of) the test risk fails to improve. This early stopping has a strong regularizing effect that, as discussed above, makes it difficult to observe the interpolation peak.

Inductive Bias. In this paper, we have dealt with several types of methods for choosing interpolating solutions. For random Fourier features, solutions are constructed explicitly by minimum norm linear regression in the feature space. As the number of features tends to infinity they approach the minimum functional norm solution in the reproducing kernel Hilbert space, a solution which maximizes functional smoothness subject to the interpolation constraints. For neural networks, the inductive bias owes to the specific training procedure used, which is typically SGD. When all but the final layer of the network are fixed (as in RFF models), SGD initialized at zero also converges to the minimum norm solution. While the behavior of SGD for more general neural networks is not fully understood, there is significant empirical and some theoretical evidence (e.g., ref. 16) that a similar minimum norm inductive bias is present. Yet another type of inductive bias related to averaging is used in random forests. Averaging potentially nonsmooth interpolating trees leads to an interpolating solution with a higher degree of smoothness; this averaged solution performs better than any individual interpolating tree.

Remarkably, for kernel machines all 3 methods lead to the same minimum norm solution. Indeed, the minimum norm interpolating classifier, $h_{n,\infty}$, can be obtained directly by explicit norm minimization (solving an explicit system of linear equations), through SGD, or by averaging trajectories of Gaussian processes [computing the posterior mean (29)].

Optimization and Practical Considerations. In our experiments, appropriately chosen "modern" models usually outperform the optimal classical model on the test set. But another important practical advantage of overparameterized models is in optimization. There is a growing understanding that larger models are "easy" to optimize as local methods, such as SGD, converge to global minima of the training risk in overparameterized regimes (e.g., ref. 30). Thus, large interpolating models can have low test risk and be easy to optimize at the same time, in particular with SGD (31). It is likely that the models to the left of the interpolation peak have optimization properties qualitatively different from those to the right, a distinction of significant practical import.

Outlook. The classical U-shaped bias–variance trade-off curve has shaped our view of model selection and directed applications of learning algorithms in practice. The understanding of model performance developed in this work delineates the limits of classical analyses and opens additional lines of inquiry to study and compare computational, statistical, and mathematical properties of the classical and modern regimes in machine learning. We hope that this perspective, in turn, will help practitioners choose models and algorithms for optimal performance.

ACKNOWLEDGMENTS. M.B. was supported by NSF Grant RI-1815697. D.H. was supported by NSF Grant CCF-1740833 and a Sloan Research Fellowship.

1. S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma. *Neural Comput.* **4**, 1–58 (1992).
2. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning* (Springer, 2001), vol. 1.
3. G. Gigerenzer, H. Brighton, Homo heuristicus: Why biased minds make better inferences. *Top. Cognit. Sci.* **1**, 107–143 (2009).
4. R. Salakhutdinov, Deep learning tutorial at the Simons Institute, Berkeley. https://simons.berkeley.edu/talks/ruslan-salakhutdinov-01-26-2017-1. Accessed 28 December 2018 (2017).
5. C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, "Understanding deep learning requires rethinking generalization" in *Proceedings of International Conference on Learning Representations* (International Conference on Learning Representations, 2017).
6. M. Belkin, S. Ma, S. Mandal, "To understand deep learning we need to understand kernel learning" in *Proceedings of the 35th International Conference on Machine Learning,* J. Dy, A. Krause, Eds. (Proceedings of Machine Learning Research, Stockholm, Sweden 2018), vol. 80, pp. 541–549.
7. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, 1995).
8. A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth, Occam's razor. *Inf. Process. Lett.* **24**, 377–380 (1987).
9. P. L. Bartlett, The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theory* **44**, 525–536 (1998).
10. R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee, Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.* **26**, 1651–1686 (1998).
11. 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*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, Eds. (Curran Associates, Inc., 2018), pp. 2300–2311.
12. M. Belkin, A. Rakhlin, A. B. Tsybakov, "Does data interpolation contradict statistical optimality?" in *Proceedings of Machine Learning Research,* K. Chaudhuri, M. Sugiyama, Eds. (Proceedings of Machine Learning Research, 2019), vol. 89, pp. 1611–1619.
13. A. Rahimi, B. Recht, "Random features for large-scale kernel machines" in *Advances in Neural Information Processing Systems*, J. C. Platt, D. Koller, Y. Singer, S. T. Roweis, Eds. (Curran Associates, Inc., 2008), pp. 1177–1184.
14. B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm for optimal margin classifiers" in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (ACM, New York, NY, 1992), pp. 144–152.
15. A. Rudi, L. Rosasco, "Generalization properties of learning with random features" in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Eds. (Curran Associates, Inc., New York, NY, 2017), pp. 3215–3225.
16. S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, N. Srebro, "Implicit regularization in matrix factorization" in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, Eds. (Curran Associates, Inc., New York, NY, 2017), pp. 6151–6159.
17. Y. Li, T. Ma, H. Zhang, "Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations" in *Proceedings of the 31st Conference On Learning Theory,* S. Bubeck, V. Perchet, P. Rigollet, Eds. (Proceedings of Machine Learning Research, 2018), vol. 75, pp. 2–47.
18. S. Bös, M. Opper, "Dynamics of training" in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, T. Petsche, Eds. (MIT Press, 1997), pp. 141–147.

19. M. S. Advani, A. M. Saxe, High-dimensional dynamics of generalization error in neural networks. arXiv:1710.03667 (10 October 2017).
20. B. Neal *et al.*, A modern take on the bias-variance tradeoff in neural networks. arXiv:1810.08591 (25 January 2019).
21. S. Spigler *et al.*, A jamming transition from under-to over-parametrization affects loss landscape and generalization. arXiv:1810.09665 (22 October 2018).
22. A. Canziani, A. Paszke, E. Culurciello, An analysis of deep neural network models for practical applications. arXiv:1605.07678 (24 May 2016).
23. O. Russakovsky *et al.*, ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
24. A. J. Wyner, M. Olson, J. Bleich, D. Mease, Explaining the success of adaboost and random forests as interpolating classifiers. *J. Mach. Learn. Res.* **18**, 1–33 (2017).
25. A. Cutler, G. Zhao, Pert-perfect random tree ensembles. *Comput. Sci. Stat.* **33**, 490–497 (2001).
26. J. H. Friedman, Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
27. P. Bühlmann, B. Yu, Boosting with the l_2 loss: Regression and classification. *J. Am. Stat. Assoc.* **98**, 324–339 (2003).
28. L. Wasserman, *All of Nonparametric Statistics* (Springer, 2006).
29. C. E. Rasmussen, "Gaussian processes in machine learning" in *Advanced Lectures on Machine Learning*, O. Bousquet, U. von Luxburg, G. Rätsch, Eds. (Springer, Berlin, 2004), pp. 63–71.
30. M. Soltanolkotabi, A. Javanmard, J. D. Lee, Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Trans. Inf. Theory* **65**, 742–769 (2018).
31. S. Ma, R. Bassily, M. Belkin, "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning" in *Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research*, J. Dy, A. Krause, eds. (PMLR, Stockholmsmässan, Stockholm, Sweden, 2018), vol. 80, pp. 3325–3334.