To Understand Deep Learning We Need to Understand Kernel Learning[†]

Mikhail Belkin [1] **Siyuan Ma** [1] **Soumik Mandal** [1]

Abstract

Generalization performance of classifiers in deep learning has recently become a subject of intense study. Deep models, which are typically heavily over-parametrized, tend to fit the training data exactly. Despite this "overfitting", they perform well on test data, a phenomenon not yet fully understood. The first point of our paper is that strong performance of overfitted classifiers is not a unique feature of deep learning. Using six real-world and two synthetic datasets, we establish experimentally that kernel machines trained to have zero classification error or near zero regression error (interpolation) perform very well on test data. We proceed to give a lower bound on the norm of zero loss solutions for smooth kernels, showing that they increase nearly exponentially with data size. None of the existing bounds produce non-trivial results for interpolating solutions. We also show experimentally that (non-smooth) Laplacian kernels easily fit random labels, a finding that parallels results recently reported for ReLU neural networks. In contrast, fitting noisy data requires many more epochs for smooth Gaussian kernels. Similar performance of overfitted Laplacian and Gaussian classifiers on test, suggests that generalization is tied to the properties of the kernel function rather than the optimization process. Some key phenomena of deep learning are manifested similarly in kernel methods in the modern "overfitted" regime. The combination of the experimental and theoretical results presented in this paper indicates a need for new theoretical ideas for understanding properties of classical kernel methods. We argue that progress on understanding deep learning will be difficult until more tractable "shallow" kernel methods are better understood.

1 Introduction

The key question in supervised machine learning is that of *generalization*. How will a classifier trained on a certain data set perform on unseen data? A typical theoretical setting for addressing this question is Empirical Risk Minimization (ERM) (Vapnik, 1995). Given data $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$ sampled from a probability distribution P on $\Omega \times \{-1, 1\}$, a class of functions $\mathscr{H} : \Omega \to \mathbb{R}$ and a loss function l, ERM finds a minimizer of the empirical loss:

$$f^* = \arg\min_{f \in \mathscr{H}} L_{emp}(f) := \arg\min_{f \in \mathscr{H}} \sum_i l(f(\boldsymbol{x}_i), y_i)$$

Most approaches work by controlling and analyzing the capacity/complexity of the space $\mathscr{H}$. Many mathematical measures of function space complexity exist, including VC and fat shattering dimensions, Rademacher complexity, covering numbers (see, e.g., (Anthony & Bartlett, 2009)). These analyses generally yield bounds on *the generalization gap*, i.e., the difference between the empirical and expected loss of classifiers. Typically, it is shown that the generalization gap tends to zero at a certain rate as the number of points n becomes large. For example, many of the classical bounds on the generalization gap are of the form $|\mathbb{E}[l(f^*(\boldsymbol{x}), y)] - L_{emp}(f^*)| < O^*(\sqrt{c/n})$, where c is a measure of complexity of $\mathscr{H}$, such as VC-dimension. Other methods, closely related to ERM, include regularization to control bias/variance (complexity) trade-off for parameter choice, and result in similar bounds. Closely related implicit regularization methods, such as early stopping for gradient descent (Yao et al., 2007; Raskutti et al., 2014; Camoriano et al., 2016), provide regularization by limiting the amount of computation, thus aiming to achieve better performance at a lower computational cost. All of these approaches suggest trading off accuracy (in terms of some loss function) on the training data to get performance guarantees on the unseen test data.

In recent years we have seen impressive progress in supervised learning due, in particular, to deep neural architectures. These networks employ large numbers of parameters, often exceeding the size of training data by several orders of magnitude (Canziani et al., 2016). This over-parametrization

† See full version of this paper at arxiv.org/abs/1802.01396.

[1]Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, USA. Correspondence to: Mikhail Belkin <mbelkin@cse.ohio-state.edu>, Siyuan Ma <masi@cse.ohio-state.edu>, Soumik Mandal <mandal.32@buckeyemail.osu.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

allows for convergence to global optima, where the training error is zero or nearly zero. Yet these "overfitted" or even interpolated networks still generalize well to test data, a situation which seems difficult to reconcile with available theoretical analyses (as observed, e.g., in (Zhang et al., 2016) or, much earlier, in (Breiman, 1995)). There have been a number of recent efforts to understand generalization and overfitting in deep networks including (Bartlett et al., 2017; Liang et al., 2017; Poggio et al., 2018).

In this paper we make the case that progress on understanding deep learning is unlikely to move forward until similar phenomena in classical kernel methods are recognized and understood. Kernel methods can be viewed as linear regression in infinite dimensional Reproducing Kernel Hilbert spaces (RKHS), which correspond to positive-definite kernel functions, such as Gaussian or Laplacian kernels. They can also be interpreted as two-layer neural networks with a fixed first layer. As such, they are far more amenable to theoretical analysis than arbitrary deep networks. Yet, despite numerous observations in the literature that very small values of regularization parameters (or even direct minimum norm solutions) often result in optimal performance (Shalev-Shwartz et al., 2011; Takác et al., 2013; Zhang et al., 2016; Gonen et al., 2016), the systematic nature of near-optimality of kernel classifiers trained to have zero classification error or zero regression error has not been recognized. We note that margin-based analyses, such as those proposed to analyze overfitting in boosting (Schapire et al., 1998), do not easily explain performance of interpolated classifiers in the presence of label noise, as sample complexity must scale linearly with the number of data points. Below we will show that most bounds for smooth kernels will, indeed, diverge with increasing data. Besides, empirical evidence shows consistent and robust generalization performance of "overfitted" and interpolated classifiers even for high label noise levels.

We will discuss these and other related issues in detail, providing both theoretical results and empirical data. The contributions of this paper are as follows:

Empirical properties of overfitted and interpolated kernel classifiers.

1. The phenomenon of strong generalization performance of overfitted/interpolated classifiers is not unique to deep networks. We demonstrate experimentally that kernel classifiers that have zero classification or regression error on the training data, still perform well on test. We use six real-world datasets (Section 3) and two synthetic datasets (Section 4) to demonstrate the ubiquity of this behavior. We also observe that regularization by early stopping provides at most a minor improvement to classifier performance.
2. It was recently observed in (Zhang et al., 2016) that ReLU networks trained with SGD easily fit standard datasets with random labels, requiring only about three times as many epochs as for fitting the original labels. Thus the fitting capacity of ReLU network function space reachable by a small number of SGD steps is very high. In Section 5 we demonstrate very similar behavior exhibited by (non-smooth) Laplacian kernels, which are easily able to fit random labels. In contrast, as expected from the theoretical considerations of fat shattering dimension (Belkin, 2018), it is far more computationally difficult to fit random labels using Gaussian kernels. However, we observe that the actual test performance of interpolated Gaussian and Laplacian kernel classifiers on real and synthetic data is very similar, and remains similar even with added label noise.

Theoretical results and the supporting experimental evidence. In Section 4 we show theoretically that performance of interpolated kernel classifiers cannot be explained by the existing generalization bounds available for kernel learning. Specifically, we prove lower bounds on the RKHS norms of overfitted solutions for smooth kernels, showing that they must increase nearly exponentially with the data size. Since most available generalization bounds depend at polynomially on the norm of the solution, this result implies divergence of most bounds as data goes to infinity. Moreover, to the best of our knowledge, none of the bounds apply to interpolated (zero regression loss) classifiers. Note that we need an assumption that the loss of the Bayes optimal classifier (the label noise) is non-zero. While it is usually believed that most real datasets have some level of label noise, it is not possible to verify when this is the case. We address this issue in two ways by analyzing (1) synthetic datasets with a known level of label noise (2) real-world datasets with added random label noise. In both cases we see that empirical test performance of overfitted kernel classifiers decays at slightly below the noise level, as it would, if the classifiers were nearly optimal. As the existing bounds for noisy data diverge, we conclude that they are unlikely to provide insight into the generalization performance of kernel classifiers.

We will now discuss some important points, conclusions and conjectures based on the combination of theoretical and experimental results presented in this paper.

Parallels between deep and shallow architectures in performance of overfitted classifiers. There is extensive empirical evidence, including the experiments in our paper, that "overfitted" kernel classifiers demonstrate strong performance on a range of datasets. Moreover, we see that introducing regularization (by early stopping) provides at most a modest improvement to the classification accuracy. Our findings parallel those for deep networks discussed in (Zhang et al., 2016). Considering that kernel methods can be viewed as a special case of two-layer neural network architectures, we conclude that deep network structure, as such, is unlikely to play a significant role in this surprising phenomenon.

Existing bounds for kernels lack explanatory power in overfitted regimes. Our experimental results show that kernel classifiers demonstrate nearly optimal performance even when the label noise is known to be significant. On the other hand, the existing bounds for overfitted/interpolated kernel methods diverge with increasing data size in the presence of label noise. We believe that a new theory of kernel methods, not dependent on norm-based concentration bounds, is needed to understand this behavior. At this point we know of few candidates for such a theory. A notable (and, to the best of our knowledge, the only) example is 1-nearest neighbor classifier, with expected loss that can be bounded asymptotically by twice the Bayes risk (Cover & Hart, 1967), while its empirical loss (both classification and regression) is identically zero. We conjecture that similar ideas are needed to analyze kernel methods and, potentially, deep learning.

Generalization and optimization. We observe that smooth Gaussian kernels and non-smooth Laplacian kernels have very different optimization properties. We show experimentally that (less smooth) Laplacian kernels easily fit standard datasets with random labels, requiring only about twice the number of epochs needed to fit the original labels (a finding that closely parallels results recently reported for ReLU neural networks in (Zhang et al., 2016)). In contrast (as suggested by the theoretical considerations of fat shattering dimension in (Belkin, 2018) optimization by gradient descent is far more computationally demanding for (smooth) Gaussian kernels. On the other hand, test performance of kernel classifiers is very similar for Laplacian and Gaussian kernels, even with added label noise. Thus the generalization performance of classifiers appear to be related to the structural properties of the kernels (e.g., their radial structure) rather than their properties with respect to the optimization methods, such as SGD.

Implicit regularization. One proposed explanation for the performance of deep networks is the idea of implicit regularization introduced by methods such as early stopping in gradient descent (Yao et al., 2007; Raskutti et al., 2014; Neyshabur et al., 2014; Camoriano et al., 2016). These approaches suggest trading off some accuracy on the training data by limiting the amount of computation, to get better performance on the unseen test data. It can be shown (Yao et al., 2007) that for kernel methods early stopping for gradient descent is effectively equivalent to traditional regularization methods, such as Tikhonov regularization.

As interpolated kernel methods fit the labels exactly (at or close to numerical precision), implicit regularization, viewed as a bias/variance trade-off, cannot provide an explanation for their generalization performance. While overfitted (zero classification loss) classifiers can be taking advantage of regularization by introducing regression loss not reflected in the classification error (cf. (Schapire et al., 1998)), we see (Section 3,4) that their performance does not significantly differ from that for interpolated classifiers.

Since deep networks are also trained to fit the data exactly, the similarity to kernel methods suggests that implicit regularization is not the basis of their generalization properties.

Inductive bias. We would like to draw a distinction between *regularization* which introduces a bias on the training data and *inductive bias*, which gives preferences to certain functions without affecting their output on training data.

While interpolated methods fit the data exactly and thus produce no regularization, minimum RKHS norm interpolating solutions introduce inductive bias by choosing functions with special properties. Note that, while infinitely many RKHS functions are capable of interpolating the data[1], the Representer Theorem (Aronszajn, 1950) ensures that the minimum norm interpolant is a linear combination of kernel functions supported on data points $\{K(\boldsymbol{x}_1, \cdot), \ldots, K(\boldsymbol{x}_n, \cdot)\}$. As we observe from the empirical results, these solutions have special generalization properties, which cannot be expected from arbitrary interpolants. While we do not yet understand how this inductive bias leads to strong generalization properties of kernel interpolants, they are obviously related to the structural properties of kernel functions and their RKHS. It is instructive to compare this setting to 1-NN classifier. While no guarantee can be given for piece-wise constant interpolating functions in general, the specific piece-wise constant function chosen by 1-NN has certain optimality properties, guaranteeing the generalization error of at most twice the Bayes risk.

It is well-known that gradient descent (and, in fact, SGD) for any strictly convex loss, initialized at 0 (or any point other point within the span of $\{K(\boldsymbol{x}_1, \cdot), \ldots, K(\boldsymbol{x}_n, \cdot)\}$), converges to the minimum norm solution, which is the unique interpolant for the data within the span of the kernels functions. On the other hand, it can be easily verified[2] that GD/SGD initialized outside of the span of $\{K(\boldsymbol{x}_1, \cdot), \ldots, K(\boldsymbol{x}_n, \cdot)\}$ cannot converge to the minimum RKHS norm solution. Thus the inductive bias corresponding to SGD with initialization at zero, is consistent with that of the minimum norm solution.

Unfortunately, we do not have an analogue of the Representer Theorem for deep networks. Despite a number of recent attempts (see, e.g., (Neyshabur et al., 2017)), it is unclear how best to construct a norm for deep networks similar to the RKHS norm for kernels. Still, it is likely that similarly to kernels, the structure of neural networks in combination with algorithms, such as SGD, introduce an inductive bias[3].

[1] Indeed, the space of RKHS interpolating functions is dense in the space of all functions in L^2!

[2] The component of the initialization vector orthogonal to the span does not change with the iterative updates.

[3] We conjecture that fully connected neural networks have in-

A remark on the importance of accelerated algorithms, hardware and SGD. Finally, we note that the experiments shown in this paper, particularly fitting noisy labels with Gaussian kernels, would be difficult without fast kernel training algorithms (we used EigenPro-SGD (Ma & Belkin, 2017), which provided 10-40x acceleration over the standard SGD/Pegasos (Shalev-Shwartz et al., 2011)) combined with modern GPU hardware. By a remarkably serendipitous coincidence, small mini-batch SGD can be shown to be exceptionally effective (nearly $O(n)$ more effective than GD) for interpolated classifiers (Ma et al., 2017).

To summarize, in this paper we demonstrate significant parallels between the properties of deep neural networks and the classical kernel methods trained in the "modern" overfitted regime. Note that kernel methods can be viewed as a special type of two-layer neural networks with a fixed first layer. Thus, we argue that more complex deep networks are unlikely to be amenable to analysis unless simpler and analytically more tractable kernel methods are better understood. Since the existing bounds seem to provide little explanatory power for their generalization performance, new insights and mathematical analyses are needed.

2 Setup

We recall some properties of kernel methods used in this paper. Let $K(\boldsymbol{x}, \boldsymbol{z}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a positive definite kernel. Then there exists a corresponding Reproducing Kernel Hilbert Space $\mathscr{H}$ of functions on $\mathbb{R}^d$, associated to the kernel $K(x, z)$. Given a data set $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}, \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, let K be the associated kernel matrix, $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and define the minimum norm interpolant

$$f^* = \arg\min_{f \in \mathscr{H},\ f(\boldsymbol{x}_i)=y_i} \|f\|_{\mathscr{H}} \tag{1}$$

Here $\|f\|_{\mathscr{H}}$ is the RKHS norm of f. From the classical representer theorem (Aronszajn, 1950) it follows that f^* exists (if no two data points x_i and x_j have the same features but different labels). Moreover, f^* can be written explicitly as

$$f^*(\cdot) = \sum \alpha_i^* K(\boldsymbol{x}_i, \cdot), \quad (\alpha_1^*, \ldots, \alpha_n^*)^T = K^{-1}\boldsymbol{y} \tag{2}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^T$. The fact that matrix K is invertible follows directly from the positive definite property of the kernel. It is easy to verify that indeed $f(\boldsymbol{x}_i) = y_i$ and hence the function f^* defined by Eq. 2 *interpolates* the data.

An equivalent way of writing Eq. 1 is to observe that f^* minimizes $\sum l(f(\boldsymbol{x}_i), y_i)$ for any non-negative loss function $l(\tilde{y}, y)$, such that $l(y, y) = 0$. If l is strictly convex, e.g., the square loss $l(f(\boldsymbol{x}_i), y_i) = (f(\boldsymbol{x}_i) - y_i)^2$, then $\boldsymbol{\alpha}^*$ is the unique vector satisfying

$$\boldsymbol{\alpha}^* = \arg\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} l\left(\left(\sum_{j=1}^{n} \alpha_i K(\boldsymbol{x}_j, \boldsymbol{x}_i)\right), y_i\right) \tag{3}$$

This is an important formulation as it allows us to define f^* in terms of an unconstrained optimization problem of a finite-dimensional space $\mathbb{R}^n$. We also recall that the RKHS norm of an arbitrary function of the form $f(\cdot) = \sum \alpha_i K(\boldsymbol{x}_i, \cdot)$ is simply $\|f\|_H^2 = \langle \boldsymbol{\alpha}, K\boldsymbol{\alpha} \rangle = \sum_{ij} \alpha_i K_{ij} \alpha_j$. In this paper we will use the popular smooth Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\|\boldsymbol{x} - \boldsymbol{z}\|^2/\sigma^2\right)$ as well as non-smooth Laplacian kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-|\boldsymbol{x} - \boldsymbol{z}\|/\sigma\right)$. We use both direct linear systems solvers and fast iterative methods.

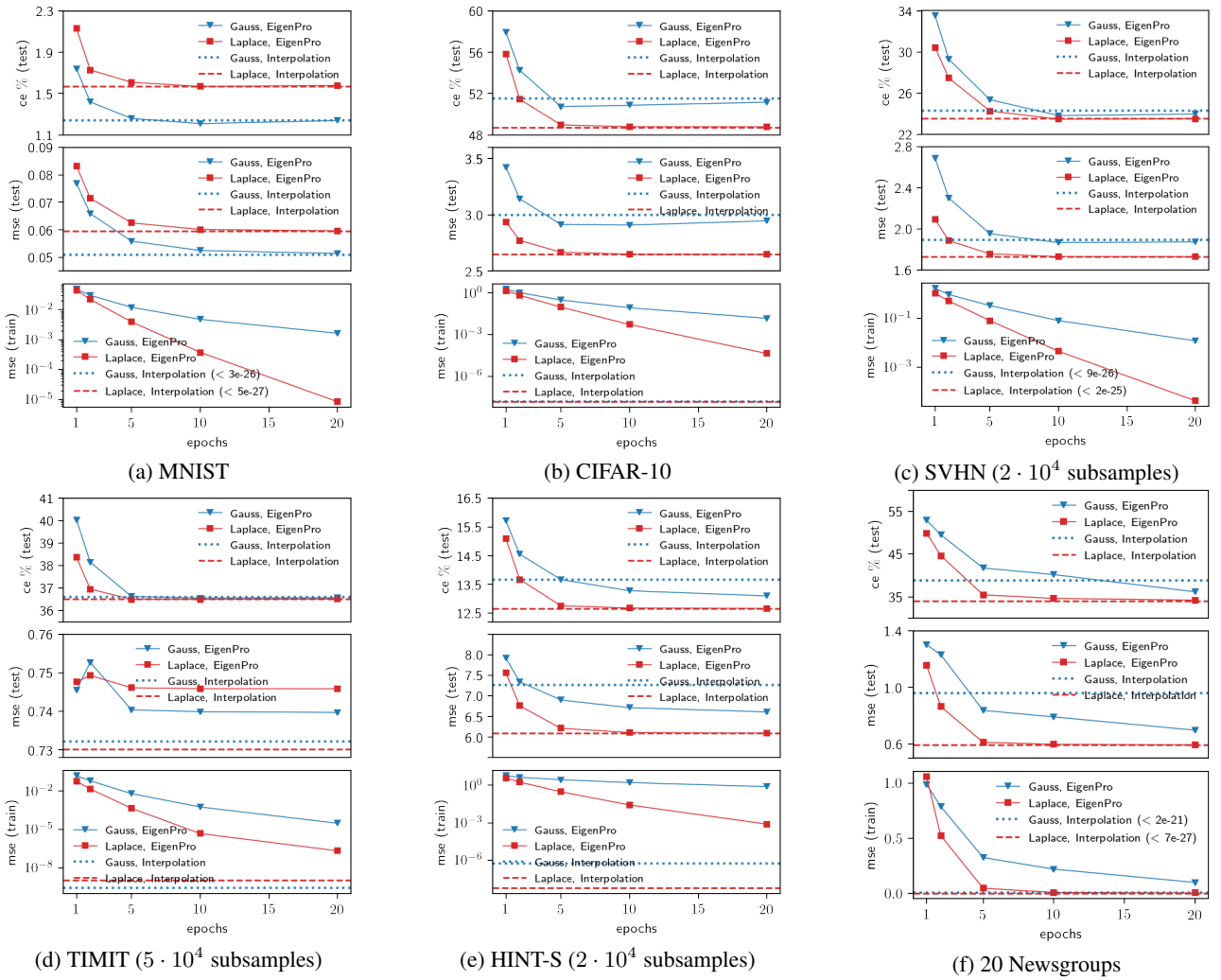
3 Generalization Performance of Overfitted/Interpolating Classifiers

In this section we establish empirically that interpolating kernel methods provide strong performance on a range of standard datasets (see supplementary for descriptions) both in terms of regression and classification. To construct kernel classifiers we use iterative EigenPro-SGD method (Ma & Belkin, 2017), which is an accelerated version of SGD in the kernel space (cf. Pegasos (Shalev-Shwartz et al., 2011)). This provides a highly efficient implementation of kernel methods and, additionally, a setting parallel to neural net training using SGD. Our experimental results are summarized in Fig. 1. We see that as the number of epochs increases, training square loss (**mse**) approaches zero[4]. On the other hand, the test error, both regression (**mse**) and classification (**ce**) remains very stable and, in most cases (in all cases for Laplacian kernels), keeps decreasing and then stabilizes. We thus observe that early stopping regularization (Yao et al., 2007; Raskutti et al., 2014) provides a small or no benefit in terms of either classification or regression error. For comparison, we also show the performance of interpolating solutions given by Eq. 2 and solved using direct methods. As expected, direct solutions always provide a highly accurate interpolation for the *training data* with the error in most cases close to numerical precision. Remarkably, we see that in all cases performance of the interpolated solution on *test* is either optimal or close to optimal both in terms of both regression and classification error.

Performance of overfitted/interpolated kernel classifiers closely parallels behaviors of deep networks noted in (Zhang et al., 2016) which fit the data exactly (only the classification error is reported there, other references also report MSE (Chaudhari et al., 2016; Huang et al., 2016; Sagun et al., 2017; Bartlett et al., 2017)). We note that observations of unexpectedly strong performance of overfitted classifiers

ductive biases similar to those of kernel methods. On the other hand, convolutional networks seem to have strong inductive biases tuned to vision problems, which can be used even in the absence of labeled data (Ulyanov et al., 2017).

[4] Training classification error (not shown), is similarly small. It is exactly zero after 20 epochs for all datasets, except for 20 News with Gaussian/Laplace kernels and HINT-S with Gaussian kernel.



(a) MNIST (b) CIFAR-10 (c) SVHN ($2 \cdot 10^4$ subsamples)

(d) TIMIT ($5 \cdot 10^4$ subsamples) (e) HINT-S ($2 \cdot 10^4$ subsamples) (f) 20 Newsgroups

Figure 1: Comparison of approximate classifiers trained by EigenPro-SGD (Ma & Belkin, 2017) and interpolated classifiers obtained from direct method for kernel least squares regression. † All methods achieve 0.0% classification error on training set. ‡ We use subsampled dataset to reduce the computational complexity and to avoid numerically unstable direct solution.

have been made before. For example, in kernel methods it has been observed on multiple occasions that very small values of regularization parameters frequently lead to optimal performance (Shalev-Shwartz et al., 2011; Takác et al., 2013). Similar observations were also made for Adaboost and Random Forests (Schapire et al., 1998). However, we have not seen recognition or systematic exploration of this phenomenon for kernel methods, and more generally in connection to interpolated classifiers and generalization with respect to the square loss.

In the next section we will examine in detail why the existing margin-based bounds are not likely to provide insight into the generalization properties of classifiers in overfitted and interpolated regimes.

4 Existing Bounds Provide No Guarantees for Interpolated Kernel Classifiers

In this section we discuss theoretical considerations related to generalization bounds for kernel classification/regression corresponding to smooth kernels. We also provide some further supporting experimental evidences. Our main theoretical result shows that the norm of overfitted kernels classifiers increases nearly exponentially with the data size as long as the error of the Bayes optimal classifier (the label noise) is non-zero. Most of the available generalizations bounds depend at most polynomially on the RKHS norm, hence diverge to infinity as data size increases and none apply to interpolated classifiers. On the other hand, we will see that the empirical performance of interpolated classifiers remains nearly optimal, even with added label noise.

Let $(\boldsymbol{x}_i, y_i) \in \Omega \times \{-1, 1\}$ be a labeled dataset, $\Omega \subset \mathbb{R}^d$ domain, and let the data be chosen from some probability measure P on $\Omega \times \{-1, 1\}$. We will assume that the loss of the Bayes optimal classifier (the label noise) is not 0, i.e., y is not a deterministic function of $\boldsymbol{x}$ on a subset of non-zero measure. We will say that $h \in \mathscr{H}$ *t-overfits* the data, if it achieves zero classification loss, and, additionally, $\forall_i y_i h(\boldsymbol{x}_i) > t > 0$ for at least a fixed portion of the train-

ing data. This condition is necessary as zero classification loss classifiers with arbitrarily small norm can be obtained by simply scaling any interpolating solution. The margin condition is far weaker than interpolation, which requires $h(\boldsymbol{x}_i) = y_i$ for all data points. We now provide a lower bound on the function norm of t-overfitted classifiers in RKHS corresponding to Gaussian kernels[5].

Theorem 1. *Let $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$ be data sampled from P on $\Omega \times \{-1, 1\}$. Assume that y is not a deterministic function of x on a subset of non-zero measure. Then, with high probability, any h that t-overfits the data, satisfies*

$$\|h\|_{\mathscr{H}} > Ae^{B\, n^{1/d}}$$

for some constants $A, B > 0$ depending on t.

See the proof in the full version (https://arxiv.org/abs/1802.01396) of this paper.

Remark. The bound in Eq. 1 applies to any t-overfitted classifier, independently of the algorithm or loss function.

We will now briefly discuss the bounds available for kernel methods. Most of the available bounds for kernel methods (see, e.g., (Steinwart & Christmann, 2008; Rudi et al., 2015)) are of the following (general) form:

$$\left| \frac{1}{n} \sum_i l(f(\boldsymbol{x}_i), y_i) - \mathbb{E}_P[l(f(\boldsymbol{x}), y)] \right| \leq$$

$$C_1 + C_2 \frac{\|f\|_{\mathscr{H}}^{\alpha}}{n^{\beta}}, \quad C_1, C_2, \alpha, \beta \geq 0$$

Note that the regularization bounds, such as those for Tikhonov regularization, are also of similar form as the choice of the regularization parameter implies an upper bound on the RKHS norm. We see that our super-polynomial lower bound on the norm $\|f\|_{\mathscr{H}}$ in Theorem 1 implies that the right hand of this inequality diverges to infinity for any overfitted classifiers, making the bound trivial. There are some bounds logarithmic in the norm, such as the bound for the fat shattering in (Belkin, 2018) (used above) and eigenvalue-dependent bounds, which are potentially logarithmic, e.g., Theorem 13 of (Goel & Klivans, 2017). However, as all of these bounds include a non-zero accuracy parameter, they do not apply to interpolated classifiers. Moreover, to account for the experiments with high label noise (below), any potential bound must have tight constants. We do not know of any complexity-based bounds with this property. It is not clear such bounds exist.

4.1 Experimental validation

Zero label noise? A potential explanation for the disparity between the consequences of lower norm bound in Theorem 1 for classical generalization results and the performance observed in actual data, is the possibility that the error rate of the Bayes optimal classifier (the "label noise") is zero (e.g., (Soudry et al., 2017)). Since our analysis relies on $\mathbb{E}_P[l(f(\boldsymbol{x}), y)] > 0$, the lower bound in Eq. 1 does not hold if y is a deterministic function of $\boldsymbol{x}$. Indeed, many classical bounds are available for "overfitted" classifiers under zero label noise condition. For example, if two classes are linearly separable, the classical bounds (including those for the Perceptron algorithm) apply to linear classifiers with zero loss. To resolve this issue, we provide experimental results demonstrating that near-optimal performance for overfitted kernel classifiers persists even for significant levels of label noise. Thus, while classical results describe generalization in zero noise regimes, they cannot explain performance in noisy regimes. We will provide several lines of evidence:
1. We study synthetic datasets, where the noise level is known a priori, showing that overfitted and interpolated classifiers consistently achieve error close to that of the Bayes optimal classifier, even for significant noise levels.
2. By adding label noise to real-world datasets we can guarantee non-zero Bayes risk. However, performance of overfitted/interpolated kernel methods decays at below the noise level, as it would for the Bayes optimal classifier.
3. We show that (as expected) for "low noise" synthetic and real datasets, adding small amounts of label noise leads to dramatic increases in the norms of overfitted solutions but only slight decreases in accuracy. For "high noise" datasets, adding label noise makes little difference for the norm but a similar decrease in classifier accuracy, consistent with the noise level. We first need the following (easily proved)

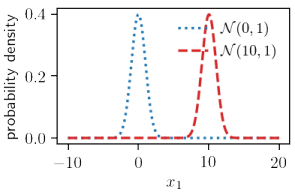
Proposition 1. Let P be a multiclass probability distribution on $\Omega \times \{1, \ldots, k\}$. Let P_ϵ be the same distribution with the ϵ fraction of the labels flipped at random with equal probability. Then the following holds:
1. The Bayes optimal classifier c^* for P_ϵ is the same as the Bayes optimal classifier for P.
2. The error rate ($0-1$ loss)

$$P_\epsilon(c^*(\boldsymbol{x}) \neq y) = \epsilon \frac{k-1}{k} + (1-\epsilon) P(c^*(\boldsymbol{x}) \neq y) \quad (4)$$

Remark. Note that adding label noise increases the error rate of the optimal classifier by at most ϵ.

A note on the experimental setting. In the experimental results in this section we only use (smooth) Gaussian kernels to provide a setting consistent with Theorem 1. Overfitted classifiers are trained to have zero classification error using EigenPro[6]. Interpolated classifiers are constructed by solving Eq. 2 directly.

Synthetic dataset 1: separable. We start by considering a synthetic dataset in $\mathbb{R}^{50}$. Each data point $(\boldsymbol{x}, y)$ is sampled



[5] The results also apply to other classes of smooth kernels, such as inverse multi-quadrics, see (Belkin, 2018).

[6] We stop iteration when classification error reaches zero.

as follows: randomly sample label y from $\{-1, 1\}$ with equal probability; for a given y, draw the first coordinate x_1 of $\boldsymbol{x} = (x_1, \ldots, x_{50}) \in \mathbb{R}^d$ from a univariate normal distribution conditional on the label and the rest uniformly from $[-1, 1]$:

$$x_1 \sim \begin{cases} \mathcal{N}(0,1), & \text{if } y = 1 \\ \mathcal{N}(10,1), & \text{if } y = -1 \end{cases} \qquad (5)$$
$$x_2 \sim U(-1,1), \ldots, x_{50} \sim U(-1,1)$$

We see that the classes are (effectively) linearly separable, with the Bayes optimal classifier $c^*(\boldsymbol{x}) = \text{sign}(x_1 - 5)$.

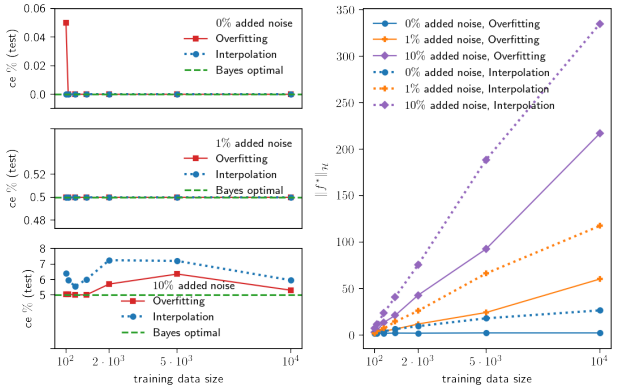
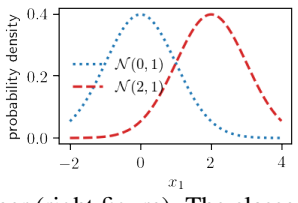


Figure 2: Overfitted and interpolated Gaussian classifiers with added label noise, separable synthetic dataset. Left: test error, Right: RKHS norms.

In Fig. 2, we show classification error rates for Gaussian kernel with a fixed kernel parameter. We compare classifiers constructed to overfit the data by driving the classification error to zero iteratively (using EigenPro) to the direct numerical interpolating solution. We see that, as expected for linearly separable data, an overfitted solution achieves optimal accuracy with a small norm. The interpolated solution has a larger norm yet performs identically. On the other hand adding just 1% label noise increases the norm by more than an order of magnitude. However both overfitted and interpolated kernel classifiers still perform at 1%, the Bayes optimal level. Increasing the label noise to 10% shows a similar pattern, although the classifiers become slightly less accurate than the Bayes optimal. We see that there is little connection between the solution norm and the classifier performance.

Additionally, we observe that the norm of either solution increases quickly with the number of data points, a finding consistent with Theorem 1.

Synthetic dataset 2: Non-separable. Consider the same setting as above, except that the Gaussian classes are moved within two standard deviations of each other (right figure). The classes are no longer separable, with the optimal classifier error of approximately 15.9%.



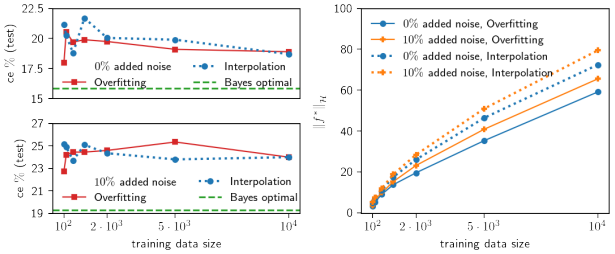
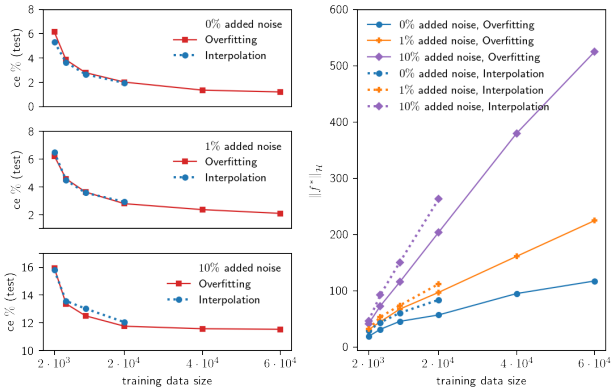
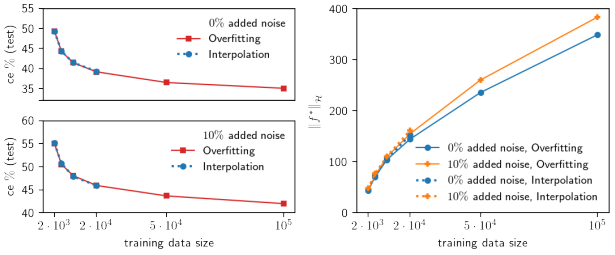


Figure 3: Overfitted and interpolated Gaussian classifiers for non-separable synthetic dataset with added label noise. Left: test error, Right: RKHS norms.

Since the setting is already noisy, we expect that adding additional label noise should have little effect on the norm. This, indeed, is the case: See Fig 3 (bottom left panel). We note that the accuracy of the interpolated classifier is consistently within 5% of the Bayes optimal, even with the added label noise.



(a) MNIST

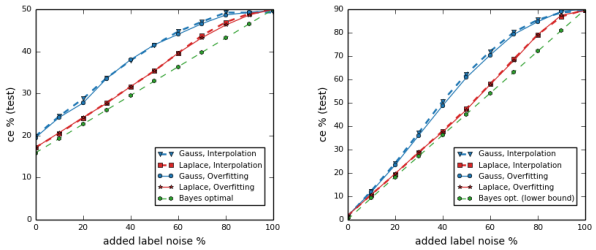


(b) TIMIT

Figure 4: Overfitted and interpolated Gaussian classifiers for real datasets with added label noise. Left:test error, Right:RKHS norm.

Real data + noise. We consider two real-data multiclass datasets (MNIST and TIMIT). MNIST labels are arguably close to a deterministic function of the features, as most (but not all) digit images are easily recognizable. On the other hand, phonetic classification task in TIMIT seems to be inherently noisier. This is reflected in the state-of-the-art error rates, less than 0.3% for (10-class) MNIST (Wan et al., 2013) and over 30% for (48-class) TIMIT (May et al., 2017). While the true Bayes risk for real data cannot be ascertained, we can ensure that it is non-zero by adding label noise. Consistently with the expectations, adding even 1% label noise

significantly increases the norm of overfitted/interpolated solutions norm for "clean" MNIST, while even additional 10% noise makes only marginal difference for "noisy" TIMIT (Fig. 4). On the other hand, the test performance on either dataset decays gracefully with the amount of noise, as it would for optimal classifiers (according to Eq. 4).



(a) Synthetic-2 (b) MNIST

Figure 5: Overfitted/interpolated classifiers, and Bayes optimal for datasets with added label noise. y axis: classification error on test.

High label noise. In Fig. 5 we show performance of Gaussian and Laplacian kernels for different levels of added label noise for Synthetic-2 and MNIST datasets. We see that interpolated kernel classifiers perform well and closely track the Bayes risk[7] even for very high levels of label noise. There is minimal deterioration as the level of label noise increases. Even at 80% label corruption they perform well above chance. Consistently with our observations above, there is very little difference in performance between interpolated and overfitted classifiers. This graph illustrates the difficulty of constructing a non-trivial generalization bound for these noisy regimes, which would have to provide values in the narrow band between the Bayes risk and the risk of a random guess.

We conclude that the theoretical and experimental results in this section suggest that it would be difficult to reconcile performance of overfitted/interpolated kernel classifiers in the noisy regimes with the usual norm-based bounds.

5 Kernels and ReLU Networks

Laplacian kernels and ReLU networks. We will now point out some interesting similarities between Laplacian kernels and ReLU networks. In (Zhang et al., 2016) the authors showed that ReLU neural networks are easily capable of fitting labels randomly assigned to the original features, needing only about three times as many iterations of SGD as for the original labels. In Table 1 we demonstrate a very similar finding for Laplacian kernels. We see that the number of epochs needed to fit random labels is no more than twice that for the original labels. Thus, SGD-type methods with Laplacian kernel have very high computational reach, similar to that of ReLU networks. Note that Laplacian kernels are non-smooth, with a discontinuity of the derivative similar to that of ReLU units. We conjecture that optimization performance is controlled by the type of non-smoothness.

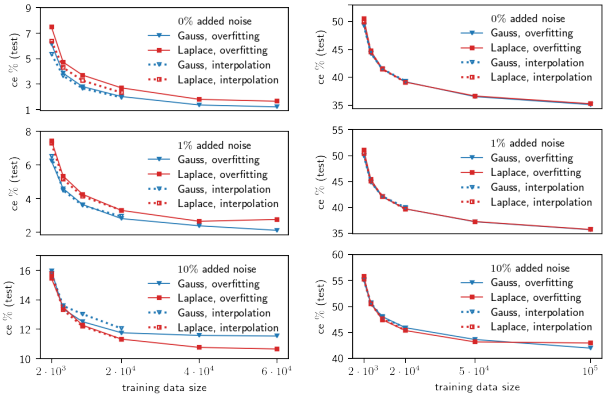
Table 1: Epochs to overfit, Laplace

Label	MNIST	SVHN	TIMIT
Original	4	8	3
Random	7	21	4

Table 2: Epochs to overfit, Gauss

Label	MNIST	SVHN	TIMIT
Original	20	46	7
Random	873	1066	22

[7] As we do not know the true Bayes risk for MNIST, we use a lower bound by simply assuming it is zero. The "true" Bayes risk is likely slightly higher than our curve.



(a) MNIST (b) TIMIT

Figure 6: Overfitted and interpolated classifiers using Gaussian kernel and Laplace kernel for datasets with added label noises (top: 0%, middle: 1%, bottom: 10%)

Laplacian and Gaussian kernels. On the other hand, training Gaussian kernels to fit noise is far more computationally intensive (see Table. 2), as suggested by the bounds on fat shattering dimension for smooth kernels (Belkin, 2018). As we see from the table, Gaussian kernels also require many more epochs to fit the original labels. On the other hand, overfitted/interpolated Gaussian and Laplacian kernels show very similar classification and regression performance on test data (Section 3). That similarity persists even with added label noise, see Fig. 6. Hence it appears that the generalization properties of these classifiers are not related to the specifics of the optimization process. We conjecture that the radial structure of these two kernels plays a key role in ensuring strong classification performance.

A note on computational efficiency. In our experiments EigenPro traced a very similar optimization path to SGD/Pegasos while providing 10X-40X acceleration in terms of the number of epochs (with about 15% overhead). When combined with Laplacian kernels, optimal performance is consistently achieved in under 10 epochs. We believe that accelerated methods with Laplacian kernels hold significant promise for future work on scaling to very large data.

Acknowledgements

We thank Raef Bassily, Daniel Hsu and Partha Mitra for discussions and insightful comments. Like Hui helped with preparing the 20 Newsgroups data. We used a Titan Xp GPU provided by Nvidia. We thank NSF for financial support.

References

Anthony, M. and Bartlett, P. L. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Bartlett, P., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. In *NIPS*, 2017.

Belkin, M. Approximation beats concentration? an approximation view on inference with smooth radial kernels. *arXiv preprint arXiv:1801.03437*, 2018.

Breiman, L. Reflections after refereeing papers for nips. 1995.

Camoriano, R., Angles, T., Rudi, A., and Rosasco, L. NYTRO: When subsampling meets early stopping. In *AISTATS*, pp. 1403–1411, 2016.

Canziani, A., Paszke, A., and Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

Chaudhari, P., Choromanska, A., Soatto, S., and LeCun, Y. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Goel, S. and Klivans, A. R. Eigenvalue decay implies polynomial-time learnability for neural networks. *CoRR*, abs/1708.03708, 2017. URL http://arxiv.org/abs/1708.03708.

Gonen, A., Orabona, F., and Shalev-Shwartz, S. Solving ridge regression using sketched preconditioned svrg. In *ICML*, pp. 1397–1405, 2016.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Liang, T., Poggio, T. A., Rakhlin, A., and Stokes, J. Fisher-rao metric, geometry, and complexity of neural networks. *CoRR*, abs/1711.01530, 2017. URL http://arxiv.org/abs/1711.01530.

Ma, S. and Belkin, M. Diving into the shallows: a computational perspective on large-scale shallow learning. *arXiv preprint arXiv:1703.10622*, 2017.

Ma, S., Bassily, R., and Belkin, M. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. *CoRR*, abs/1712.06559, 2017. URL http://arxiv.org/abs/1712.06559.

May, A., Garakani, A. B., Lu, Z., Guo, D., Liu, K., Bellet, A., Fan, L., Collins, M., Hsu, D., Kingsbury, B., et al. Kernel approximation methods for speech recognition. *arXiv preprint arXiv:1701.03577*, 2017.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In *NIPS*, pp. 5949–5958, 2017.

Poggio, T., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., Hidary, J., and Mhaskar, H. Theory of Deep Learning III: explaining the non-overfitting puzzle. *arXiv e-prints*, December 2018.

Raskutti, G., Wainwright, M. J., and Yu, B. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *JMLR*, 15(1):335–366, 2014.

Rudi, A., Camoriano, R., and Rosasco, L. Less is more: Nyström computational regularization. In *NIPS*, pp. 1657–1665, 2015.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5), 1998.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

Soudry, D., Hoffer, E., and Srebro, N. The Implicit Bias of Gradient Descent on Separable Data. *ArXiv e-prints*, October 2017.

Steinwart, I. and Christmann, A. *Support vector machines*. Springer Science & Business Media, 2008.

Takác, M., Bijral, A. S., Richtárik, P., and Srebro, N. Mini-batch primal and dual methods for svms. In *ICML*, 2013.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *ICML*, pp. 1058–1066, 2013.

Yao, Y., Rosasco, L., and Caponnetto, A. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.