arXiv:1802.01396v3 [stat.ML] 14 Jun 2018

To Understand Deep Learning We Need to Understand Kernel Learning

Mikhail Belkin, Siyuan Ma, Soumik Mandal
Department of Computer Science and Engineering
Ohio State University
{mbelkin, masi}@cse.ohio-state.edu, mandal.32@osu.edu

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, even when the labels are corrupted with a high level of noise. 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. We point out that this is difficult to reconcile with the existing generalization bounds. Moreover, none of the bounds produce non-trivial results for interpolating solutions.

Second, we 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.

Certain 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 classical Empirical Risk Minimization (ERM) [Vap95]. 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., [AB09]). 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

Early Stop：https://zhuanlan.zhihu.com/p/28104973

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 by parameter choice, and result in similar bounds. Closely related implicit regularization methods, such as early stopping for gradient descent [YRC07, RWY14, CARR16], 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 [CPC16]. This over-parametrization allows for convergence to global optima, where the training error is zero or nearly zero. Yet these "overfitted[1]" 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 [ZBH+16] or, much earlier, in [Bre95]). There have been a number of recent efforts to understand generalization and overfitting in deep networks including [BFT17, LPRS17, PKL+17].

In this paper we make the case that progress on understanding deep learning is unlikely to move forward until similar phenomena in classical kernel machines are recognized and understood. Kernel machines 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 [SSSSC11, TBRS13, ZBH+16, GOSS16, RCR17], 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 [SFBL98], 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. We would like to point out an insightful (but seemingly little noticed) recent paper [WOBM17] which proposed an alternative explanation for the success of Adaboost, much closer to our discussion here.

RKHS是什么：https://zhuanlan.zhihu.com/p/54704957

来自Chatgpt的解释：

过拟合分类器（overfitting classifiers）和插值分类器（interpolation classifiers）是机器学习领域中常用的术语，两者的区别如下：

过拟合分类器：指的是在训练集上表现出色，但在测试集或新数据上表现较差的分类器。过拟合发生时，分类器过于复杂或过度拟合了训练集的噪声和细节，导致模型无法泛化到新的数据。过拟合分类器具有较高的方差，即对训练集中的变动敏感，但对于新数据的泛化能力较差。

插值分类器：指的是在训练集和测试集上都表现良好的分类器。插值分类器能够准确地捕捉训练集和测试集之间的模式和关系，它们具有较低的方差和较高的偏差。插值分类器能够有效地拟合数据，并在新数据上表现出较好的泛化能力。

总结起来，过拟合分类器在训练集上表现良好但在测试集上表现较差，而插值分类器在训练集和测试集上都能表现良好。插值分类器是目标，而过拟合分类器是需要避免的问题。

Below we will show that most bounds for smooth kernels will, indeed, diverge with increasing data. On the other hand, 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 contribution 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) as well as two synthetic datasets (Section 4) to demonstrate the ubiquity of this behavior. Additionally, we observe that regularization by early stopping provides at most a minor improvement to classifier performance.
  2. It was recently observed in [ZBH+16] 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

[1] We use *overfitting* as a purely technical term to refer to zero classification error as opposed to interpolation which has zero regression error.

similar behavior exhibited by (non-smooth) Laplacian (exponential) kernels, which are easily able to fit random labels. In contrast, as expected from the theoretical considerations of fat shattering dimension [Bel18], 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 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 existing bounds (including potential logarithmic 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 data have some level of label noise, it is not usually possible to ascertain 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 additional random label noise. In both cases we see that empirical test performance of interpolated kernel classifiers decays at slightly below the noise level, as it would, if the classifiers were nearly optimal. This finding holds even for very high levels of label noise. We thus conclude that the existing bounds are unlikely to provide insight into the generalization performance of kernel classifiers. Moreover, since the empirical risk is zero, any potential non-trivial bound for the generalization gap, aiming to describe noisy data, must have tight constants to produce a value between the (non-zero) Bayes risk and 1. To the best of our knowledge, no examples of such bounds exist.

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, in Section 3 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 [ZBH+16]. 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 [CH67], 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 [ZBH+16]). In contrast (as suggested by the theoretical considerations of fat shattering dimension in [Bel18]) 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 and loss functions. 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 [YRC07, RWY14, NTS14, CARR16]. 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 [YRC07] 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 trade-off between train and test performance, cannot provide an explanation for their generalization performance. While overfitted (zero classification loss) classifiers can, in principle, be taking advantage of regularization by introducing regression loss not reflected in the classification error (cf. [SFBL98]), we see (Section 3,4) that their performance does not significantly differ from that for interpolated classifiers for which margin-based explanations to not apply.

Another interesting point is that any strictly convex loss function leads to the same interpolated solution. Thus, it is unlikely that the choice of loss function relates to the generalization properties of classifiers[2].

Since deep networks are also trained to fit the data exactly, the similarity to kernel methods suggests that implicit regularization or the specifics of the loss function used in training, are not the basis of their generalization properties.

Inductive bias and minimum norm solutions. While the notions of *regularization* and *inductive bias* are frequently used interchangeably in the literature, we feel it would be useful 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 the 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 infinitely many RKHS functions are capable of interpolating the data[3]. However, the Representer Theorem [Aro50] 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 [CH67].

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

[2] It has been long noticed that performance of kernel classifiers does not significantly depend on the choice of loss functions. For example, kernel SVM performs very similarly to kernel least square regression [ZP04].

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

kernels functions. On the other hand, it can be easily verified[4] 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.

This view also provides a natural link to the phenomenon observed in AdaBoost training, where the test error improves even after the classification error on train reached zero [SFBL98]. If we believe that the minimum norm solution (or the related maximum margin solution) has special properties, iterative optimization should progressively improve the classifier, regardless of the training set performance. Furthermore, based on this reasoning, generalizations bounds that connect empirical and expected error are unlikely to be helpful.

Unfortunately, we do not have an analogue of the Representer Theorem for deep networks. Also, despite a number of recent attempts (see, e.g., [NBMS17]), it is not clear how best to construct a norm for deep networks similar to the RKHS norm for kernels. Still, it appears likely that similarly to kernels, the structure of neural networks in combination with algorithms, such as SGD, introduce an inductive bias[5].

We see that kernel machines have a unique analytical advantage over other powerful nonlinear techniques such as boosting and deep neural networks as their minimum norm solutions can be computed analytically and analyzed using a broad range of mathematical analytic techniques. Additionally, at least for smaller data, these solutions can be computed using the classical direct methods for solving systems of linear equations. We argue that kernel machines provide a natural analytical and experimental platform for understanding inference in modern machine learning.

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 to conduct without fast kernel training algorithms (we used EigenPro-SGD [MB17], which provided 10-40x acceleration over the standard SGD/Pegasos [SSSSC11]) 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 full gradient descent) for interpolated classifiers [MBB17].

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 [Aro50] it follows that f^* exists (as long as no two data points x_i and x_j have the same features but different labels).

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

[5] We conjecture that fully connected neural networks have inductive 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 [UVL17].

Moreover, f^* can be written explicitly as

$$f^*(\cdot) = \sum \alpha_i^* K(\boldsymbol{x}_i, \cdot), \text{ where } (\alpha_1^*, \ldots, \alpha_n^*)^T = K^{-1}(y_1, \ldots, y_n)^T \tag{2}$$

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 on a finite-dimensional space $\mathbb{R}^n$. In particular, iterative methods can be used to solve for α^*, often obviating the need to invert the $n \times n$ matrix K. Matrix inversion generally requires n^3 operation, which is prohibitive for large data.

We also recall that the RKHS norm of an arbitrary function of the form $f(\cdot) = \sum \alpha_i K(\boldsymbol{x}_i, \cdot)$ can be easily computed as

$$\|f\|_H^2 = \langle \boldsymbol{\alpha}, K\boldsymbol{\alpha} \rangle = \sum_{ij} \alpha_i K_{ij} \alpha_j$$

In this paper we will primarily use the popular smooth Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}\right)$ as well as non-smooth Laplacian (exponential) kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma}\right)$. We will use both direct linear systems solvers and iterative methods.

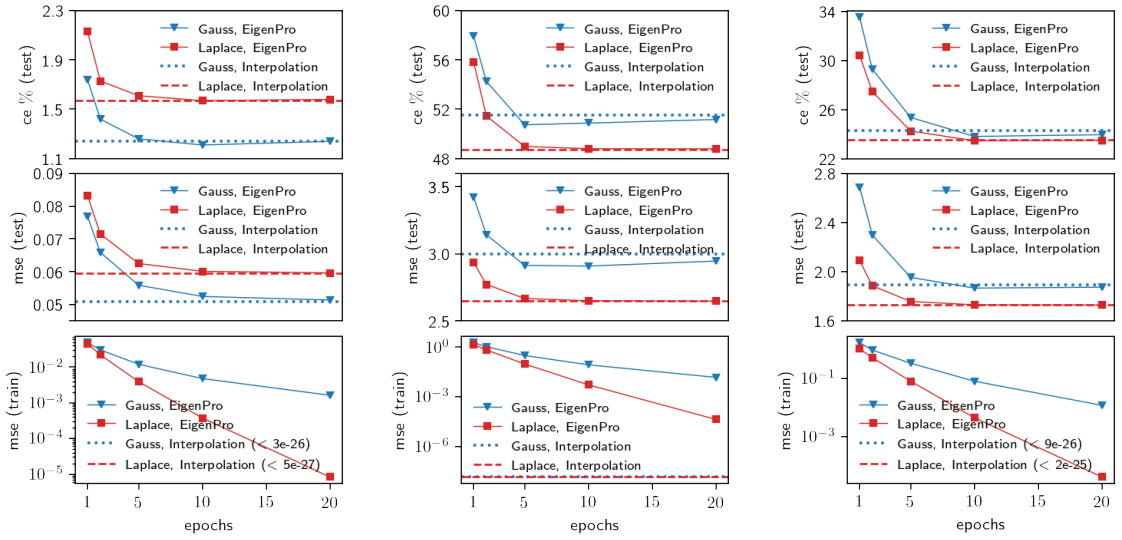
Interpolation versus "overfitting". In this paper we will refer to classifiers as *interpolated* if their square loss on the training error is zero or close to zero. We will call classifiers *overfitted* if the same holds for classification loss (for the theoretical bounds we will additionally require a small fixed margin on the training data). Notice that while interpolation implies overfitting, the converse does not hold.

3 Generalization Performance of Overfitted/Interpolating Classifiers

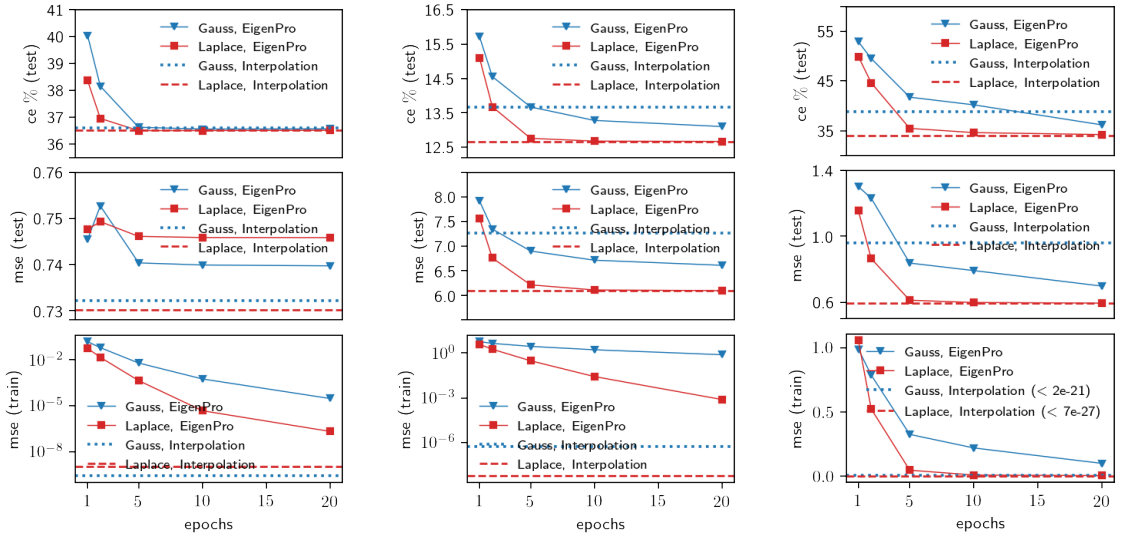
In this section we establish empirically that interpolating kernel methods provide strong performance on a range of standard datasets (see Appendix A for dataset descriptions) both in terms of regression and classification. To construct kernel classifiers we use iterative EigenPro-SGD method [MB17], which is an accelerated version of SGD in the kernel space (cf. Pegasos [SSSSC11]). 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 (see Appendix B for full numerical results including the classification accuracy on the training set).

We see that as the number of epochs increases, training square loss (**mse**) approaches zero[6]. 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 [YRC07, RWY14] provides a small or no benefit in terms of either classification or regression error.

[6]The training classification error (not shown), is similarly small. After 20 epochs of EigenPro it is zero for all datasets, except for 20 Newsgoups with Gaussian/Laplace kernels and HINT-S with Gaussian kernel (see Appendix B).



(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 [MB17] 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.

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 [ZBH+16] which fit the data exactly (only the classification error is reported there, other references also report MSE [CCSL16, HLWvdM16, SEG+17, BFT17]). We note that observations of unexpectedly strong performance of overfitted classifiers 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 [SSSSC11, TBRS13]. Similar observations were also made for Adaboost and Random Forests [SFBL98] (see [WOBM17] for

a recent and quite different take on that). However, we have not seen recognition or systematic exploration of this (apparently ubiquitous) 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 examine in detail why the existing margin 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

Bayes:https://zhuanlan.zhihu.com/p/42991859,https://zhuanlan.zhihu.com/p/397960221,https://zhuanlan.zhihu.com/p/40024110
https://zhuanlan.zhihu.com/p/32480810

In this section we discuss theoretical considerations related to generalization bounds for kernel classification and regression corresponding to smooth kernels. We also provide further supporting experimental evidence. 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, and 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, $\Omega \subset \mathbb{R}^d$ a bounded 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 training 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[7].

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*

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

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

Proof. Let $B_R = \{f \in \mathscr{H}, \|f\|_{\mathscr{H}} < R\} \subset \mathscr{H}$ be a ball of radius R in the RKHS $\mathscr{H}$. We will prove that with high probability B_R contains no functions that t-overfit the data, unless R is large, which will imply our result.

Let l be the hinge loss with margin t: $l(f(\boldsymbol{x}), y) = (t - y f(\boldsymbol{x}))_+$. Let $V_\gamma(B_R)$ be the fat shattering dimension of the function space B_R with the parameter γ. By the classical results on fat shattering dimension (see,e.g.,[AB09]) $\exists C_1, C_2 > 0$ such that with high probability $\forall_{f \in B_R}$:

$$\left| \frac{1}{n} \sum_i l(f(\boldsymbol{x}_i), y_i) - \mathbb{E}_P[l(f(\boldsymbol{x}), y)] \right| \leq C_1 \gamma + C_2 \sqrt{\frac{V_\gamma(B_R)}{n}}$$

Since y is not a deterministic function of x on some subset of non-zero measure, $\mathbb{E}_P[l(f(\boldsymbol{x}), y)]$ is non-zero. Fix γ to be a positive number, such that $C_1 \gamma < \mathbb{E}_P[l(f(\boldsymbol{x}), y)]$.

[7] The results also apply to other classes of smooth kernels, such as inverse multi-quadrics.

Suppose now that a function $h \in B_R$ t-overfits the data. Then $\frac{1}{n}\sum_i l(h(\boldsymbol{x}_i), y_i) = 0$ and hence

$$0 < \mathbb{E}_P[l(f(\boldsymbol{x}), y)] - C_1\gamma < C_2\sqrt{\frac{V_\gamma(B_R)}{n}}$$

Thus the ball B_R with high probability contains no function that t-overfits the data unless

$$V_\gamma(B_R) > \frac{n}{C_2}\left(\mathbb{E}_P[l(f(\boldsymbol{x}), y)] - C_1\gamma\right)^2$$

On the other hand, [Bel18] gives a bound on the V_γ dimension of the form $V_\gamma(B_R) < O\left(\log^d\left(\frac{R}{\gamma}\right)\right)$. Expressing R in terms of $V_\gamma(B_R)$, we see that B_R with high probability contains no function that t-overfits the data unless R is at least $Ae^{B\,n^{1/d}}$ for some $A, B > 0$. That completes the proof. □

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., [SC08, RCR15]) 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 [Bel18] (used above) and eigenvalue-dependent bounds, which are potentially logarithmic, e.g., Theorem 13 of [GK17]. 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 is zero (e.g., [SHS17]). 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[8] 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 high levels of label noise. Thus, while classical bounds may describe generalization in zero noise regimes, they cannot explain performance in noisy regimes. We 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 high noise levels.

[8] Note that even when y is a deterministic function of x, the norm of the interpolated solution will diverge to infinity unless $y(\boldsymbol{x}) \in \mathscr{H}$. Since $y(\boldsymbol{x})$ for classification is discontinuous, $y(\boldsymbol{x})$ is never in RKHS for smooth kernels. However, in this case, the growth of the norm of the interpolant as a function of n requires other techniques to analyze.

2. By adding label noise to real-world datasets we can guarantee non-zero Bayes risk. However, as we will see, performance of overfitted/interpolated kernel methods decays at or 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.

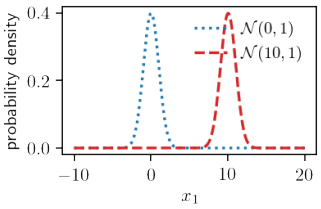
Proposition 1. *Let P be a multiclass probability distribution on $\Omega \times \{1, \ldots, k\}$. Let P_ϵ be the same distribution with the ϵ fraction of the labels flipped at random with equal probability. Then the following holds:*
1. The Bayes optimal classifier c^* for P_ϵ is the same as the Bayes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

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

Remark. Note that adding label noise to a probability distribution increases the error rate of the optimal classifier by at most ϵ. In particular, when $k = 2$ and P has no label noise, the Bayes risk of P_ϵ is simply $\epsilon/2$. In addition, the loss of the Bayes optimal classifier is linear in 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[9]. Interpolated classifiers are constructed by solving Eq. 2 directly[10].

Synthetic dataset 1: Separable classes+noise. We start by considering a synthetic dataset in $\mathbb{R}^{50}$. Each data point $(\boldsymbol{x}, y)$ is sampled as follows: randomly sample label y from $\{-1, 1\}$ with equal probability; for a given y, draw the first coordinate 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 $[-1, 1]$:



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

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

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

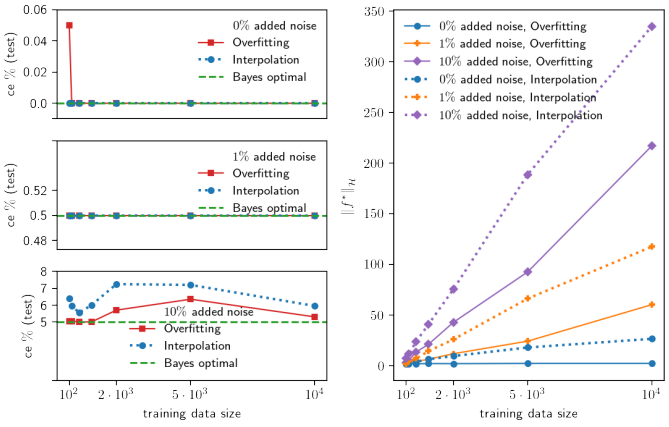


Figure 2: Overfitted/interpolated Gaussian kernel classifiers. Synthetic dataset 1. Left top to bottom: test error 0%, 1%, 10% added label noise. Right: RKHS norms.

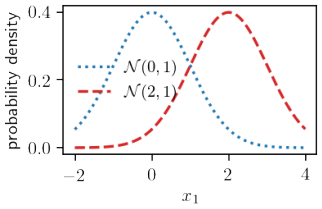
[9] We stop iteration when classification error reaches zero.

[10] As interpolated classifiers are constructed by solving a poorly conditioned system of equation, the reported norm should be taken as a lower bound on the actual norm.

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 classes. 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%.



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.

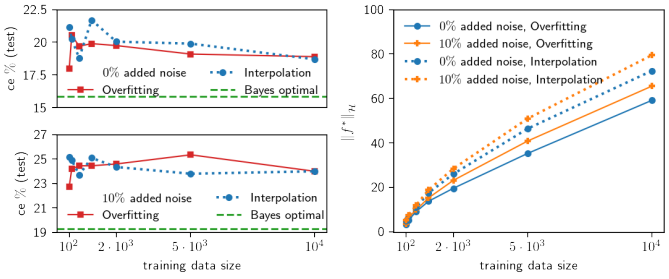
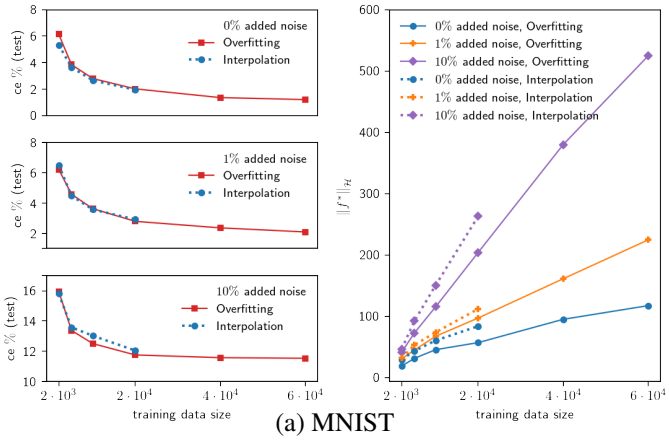


Figure 3: Overfitted and interpolated Gaussian classifiers for non-separable synthetic dataset with added label noise. Left: test error, Right: RKHS norms.

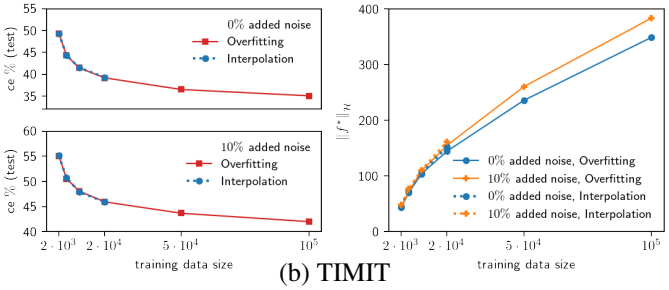
Real data + noise. We consider two real-data multiclass datasets (MNIST and TIMIT). MNIST labels are arguably close to a deterministic function of the features, as most (but not all) digit images are easily recognizable. On the other hand, phonetic classification task in TIMIT seem to be significantly more uncertain and inherently noisier.

This is reflected in the state-of-the-art error rates, less than 0.3% for (10-class) MNIST [WZZ+13] and over 30% for (48-class) TIMIT [MGL+17]. 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



(a) MNIST

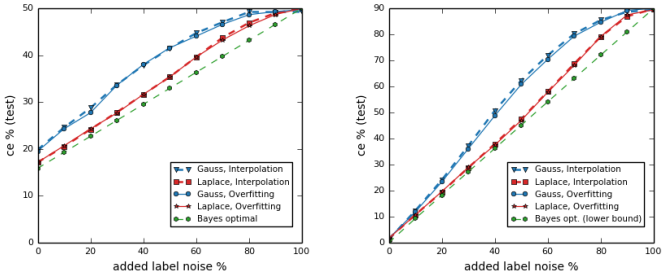


(b) TIMIT

Figure 4: Overfitted/interpolated Gaussian classifiers (MNIST/TIMIT), added label noise. Left: test error, Right: RKHS norms.

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).

High label noise Bayes risk comparison. 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[11] 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.



(a) Synthetic-2 (b) MNIST

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

5 Fitting noise: Laplacian and Gaussian kernels, connections to ReLU Networks

Laplacian kernels and ReLU networks. We will now point out some interesting similarities between Laplacian kernel machines and ReLU networks. In [ZBH+16] 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. We note that Laplacian kernels are non-smooth, with a discontinuity of the derivative reminiscent of that for ReLU units. We conjecture that optimization performance is controlled by the type of non-smoothness.

Table 1: Epochs to overfit (Laplacian)

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

Table 2: Epochs to overfit (Gaussian)

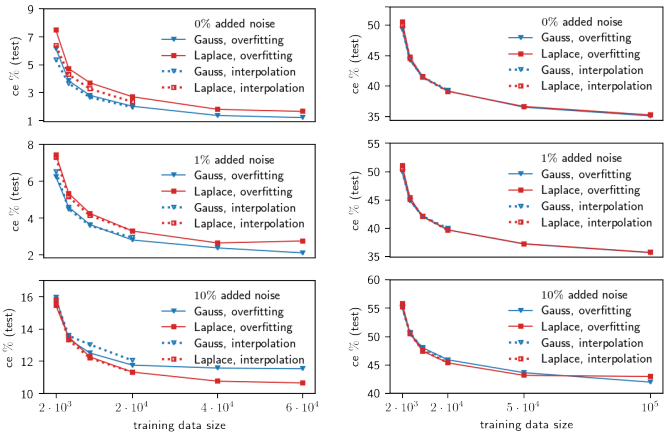
Label	MNIST	SVHN	TIMIT
Original	20	46	7
Random	873	1066	22

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 [Bel18]. 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.

[11] 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.

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 methods using Laplacian kernels hold significant promise for future work on scaling to very large data.



(a) MNIST (b) TIMIT

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

Acknowledgements

We thank Raef Bassily, Daniel Hsu and Partha Mitra for numerous discussions, insightful questions and comments. We thank Like Hui for preprocessing the 20 Newsgroups dataset. We used a Titan Xp GPU provided by Nvidia. We are grateful to NSF for financial support.

References

[AB09] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[Aro50] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[Bel18] Mikhail Belkin. Approximation beats concentration? an approximation view on inference with smooth radial kernels. *arXiv preprint arXiv:1801.03437*, 2018.

[BFT17] Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NIPS*, 2017.

[Bre95] Leo Breiman. Reflections after refereeing papers for nips. 1995.

[CARR16] R. Camoriano, T. Angles, A. Rudi, and L. Rosasco. NYTRO: When subsampling meets early stopping. In *AISTATS*, pages 1403–1411, 2016.

[CCSL16] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, and Yann LeCun. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

[CH67] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[CPC16] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

[GK17] Surbhi Goel and Adam R. Klivans. Eigenvalue decay implies polynomial-time learnability for neural networks. *CoRR*, abs/1708.03708, 2017.

[GLF+93] John S Garofolo, Lori F Lamel, William M Fisher, Jonathon G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. *NIST speech disc*, 1-1.1, 1993.

[GOSS16] Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned svrg. In *ICML*, pages 1397–1405, 2016.

[HLWvdM16] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[HYWW13] Eric W Healy, Sarah E Yoho, Yuxuan Wang, and DeLiang Wang. An algorithm to improve speech recognition in noise for hearing-impaired listeners. *The Journal of the Acoustical Society of America*, 134(4):3029–3038, 2013.

[KH09] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

[Lan95] Ken Lang. Newsweeder: Learning to filter netnews. In *ICML*, 1995.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[LPRS17] Tengyuan Liang, Tomaso A. Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. *CoRR*, abs/1711.01530, 2017.

[MB17] Siyuan Ma and Mikhail Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. *arXiv preprint arXiv:1703.10622*, 2017.

[MBB17] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. *CoRR*, abs/1712.06559, 2017.

[MGL+17] Avner May, Alireza Bagheri Garakani, Zhiyun Lu, Dong Guo, Kuan Liu, Aurélien Bellet, Linxi Fan, Michael Collins, Daniel Hsu, Brian Kingsbury, et al. Kernel approximation methods for speech recognition. *arXiv preprint arXiv:1701.03577*, 2017.

[NBMS17] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *NIPS*, pages 5949–5958, 2017.

[NTS14] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[NWC+11] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop*, volume 2011, page 4, 2011.

[PKL+17] T. Poggio, K. Kawaguchi, Q. Liao, B. Miranda, L. Rosasco, X. Boix, J. Hidary, and H. Mhaskar. Theory of Deep Learning III: explaining the non-overfitting puzzle. *arXiv e-prints*, December 2017.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[RCR15] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *NIPS*, pages 1657–1665, 2015.

[RCR17] A. Rudi, L. Carratino, and L. Rosasco. FALKON: An Optimal Large Scale Kernel Method. *ArXiv e-prints*, May 2017.

[RWY14] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *JMLR*, 15(1):335–366, 2014.

[SC08] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[SEG+17] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[SFBL98] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Statist.*, 26(5), 1998.

[SHS17] D. Soudry, E. Hoffer, and N. Srebro. The Implicit Bias of Gradient Descent on Separable Data. *ArXiv e-prints*, October 2017.

[SSSSC11] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[TBRS13] Martin Takác, Avleen Singh Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for svms. In *ICML*, 2013.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[WOBM17] Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *Journal of Machine Learning Research*, 18(48):1–33, 2017.

[WZZ+13] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, pages 1058–1066, 2013.

[YRC07] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[ZBH+16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[ZP04] P Zhang and Jing Peng. Svm vs regularized least squares classification. 1:176 – 179 Vol.1, 09 2004.

Appendices

A Experimental Setup

Computing Resource. All experiments were run on a single workstation equipped with 128GB main memory, two Intel Xeon(R) E5-2620 processors, and one Nvidia GTX Titan Xp (Pascal) GPU.

Datasets. The table on the right summarizes the datasets used in experiments. We map multiclass labels to multiple binary labels (e.g. one label of c classes to one c-length binary vector). For image datasets including MNIST [LBBH98], CIFAR-10 [KH09], and SVHN [NWC+11], color images are first transformed to grayscale images. We then rescale the range of each feature to $[0, 1]$. For HINT-S [HYWW13] and TIMIT [GLF+93], we normalize each feature by z-score. To efficiently fit the 20 Newsgroups [Lan95] dataset with kernel regression, we transform its sparse feature vector (bag of words) into dense feature vector by summing up the corresponding embeddings of the words from [PSM14].

Dataset	n	d	Label
CIFAR-10	5×10^4	1024	{0,...,9}
MNIST	6×10^4	784	{0,...,9}
SVHN	7×10^4	1024	{1,...,10}
HINT-S	2×10^5	425	$\{0, 1\}^{64}$
TIMIT	1.1×10^6	440	{0,...,143}
20 Newsgroups	1.6×10^4	100	$\{0, 1\}^{20}$

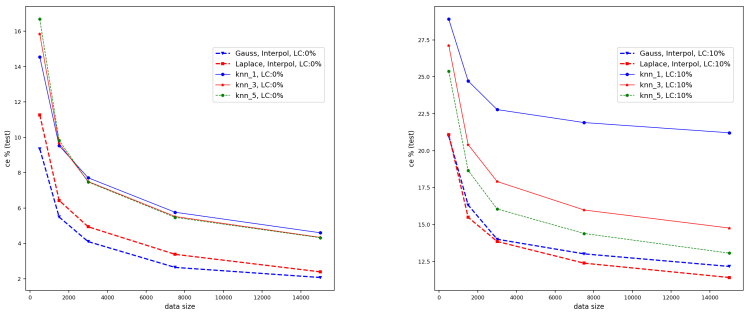
Hyperparameters. For consistent comparison, all iterative methods use mini-batch of size $m = 256$. The EigenPro preconditioner in [MB17] is constructed using the top $k = 160$ eigenvectors of a subsampled training set of size $M = 5000$ (or the training set when its size is less than 5000).

Kernel Bandwidth Selection. For each dataset, we select the bandwidth σ for Gaussian kernel $k(x, y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$ and Laplace kernel $k(x, y) = \exp(-\frac{\|x-y\|}{\sigma})$ through cross-validation on a small subsampled dataset. The final bandwidths used for all datasets are listed in the table on the right side.

Dataset	Gauss	Laplace
CIFAR-10	5	10
MNIST	5	10
SVHN	5	10
HINT-S	11	20
TIMIT	16	20
20 News	0.1	0.1

B Detailed experimental results

Below in Tables 3,4,5 we provide exact detailed numerical results for the graphs given in Section 3. In Table 6 and 7 test classification errors have been compared among different training data size and different methods with 0% and 10% added label noise respectively. Fig. 7 shows this comparison. Fig. 8 shows results with different bandwidths. Three settings for bandwidth have been considered: 50%, 100% and 200% of the bandwidth selected for optimal performance (different for Gaussian and Laplacian kernels).



(a) Label noise 0% (b) Label noise 10%

Figure 7: Interpolated classifiers, and k-NN for MNIST dataset. Added label noise 0% and 10%. y axis: classification error on test data. x axis: training data size.

	Kernel, Method	Epochs (MNIST)					Epochs (CIFAR-10)				
		1	2	5	10	20	1	2	5	10	20
ce % (test)	Gauss, Eipro	1.74	1.42	1.26	1.21	1.24	57.94	54.27	50.74	50.88	51.17
	Laplace, Eipro	2.13	1.73	1.61	1.57	1.58	55.86	51.46	48.98	48.8	48.79
	Gauss, Interp	1.24					51.56				
	Laplace, Interp	1.57					48.75				
ce % (train)	Gauss, Eipro	0.44	0.16	0.018	0.003	0.0	18.73	5.28	0.23	0.03	0.0
	Laplace, Eipro	0.32	0.03	0.0	0.0	0.0	6.95	0.23	0.0	0.0	0.0
	Gauss, Interp	0					0				
	Laplace, Interp	0					0				
mse (test)	Gauss, Eipro	0.077	0.066	0.06	0.05	0.05	3.42	3.14	2.92	2.91	2.95
	Laplace, Eipro	0.083	0.07	0.062	0.06	0.06	2.94	2.77	2.67	2.65	2.65
	Gauss, Interp	0.05					3.00				
	Laplace, Interp	0.06					2.65				
mse (train)	Gauss, Eipro	0.049	0.031	0.012	0.005	0.002	1.88	1.07	0.32	0.09	0.02
	Laplace, Eipro	0.046	0.022	3.9e-3	3.7e-4	8.2e-6	1.44	0.69	0.09	5.4e-3	4.4e-5
	Gauss, Interp	3.2e-27					1.5e-8				
	Laplace, Interp	4.6e-28					1.6e-8				

Table 3: MNIST and CIFAR-10 summary table.

	Kernel, Method	Epochs (SVHN 20k)					Epochs (TIMIT 50k)				
		1	2	5	10	20	1	2	5	10	20
ce % (test)	Gauss, Eipro	33.57	29.32	25.37	23.82	23.96	40.04	38.16	36.63	36.56	36.57
	Laplace, Eipro	30.47	27.48	24.27	23.47	23.51	38.36	36.95	36.5	36.5	36.51
	Gauss, Interp	24.30					36.61				
	Laplace, Interp	23.54					36.51				
ce % (train)	Gauss, Eipro	10.81	4.19	0.71	0.07	0.0	2.13	0.28	0.0	0.002	0.0
	Laplace, Eipro	3.79	0.34	0.0	0.0	0.0	0.39	0.0	0.0	0.0	0.0
	Gauss, Interp	0					0				
	Laplace, Interp	0					0				
mse (test)	Gauss, Eipro	2.69	2.30	1.96	1.87	1.88	0.75	0.75	0.74	0.74	0.74
	Laplace, Eipro	2.09	1.89	1.76	1.73	1.73	0.75	0.75	0.75	0.75	0.75
	Gauss, Interp	1.89					0.73				
	Laplace, Interp	1.73					0.73				
mse (train)	Gauss, Eipro	1.71	0.95	0.34	0.08	0.01	0.163	0.065	0.006	5.8e-4	3.1e-5
	Laplace, Eipro	1.06	0.52	0.08	4.5e-3	4.2e-5	0.059	0.015	4.5e-4	4.8e-6	2.1e-7
	Gauss, Interp	8.9e-27					2.5e-10				
	Laplace, Interp	2.0e-26					1.1e-9				

Table 4: SVHN and TIMIT summary table.

	Kernel, Method	Epochs (HINT-S-20k)					Epochs (20 Newsgroups)				
		1	2	5	10	20	1	2	5	10	20
ce % (test)	Gauss, Eipro	15.71	14.55	13.67	13.28	13.10	52.95	49.55	41.75	40.20	36.15
	Laplace, Eipro	15.09	13.67	12.76	12.68	12.67	49.95	44.65	35.45	34.59	34.20
	Gauss, Interp	13.67					38.75				
	Laplace, Interp	12.65					33.95				
ce % (train)	Gauss, Eipro	10.99	7.60	3.69	1.56	0.37	30.82	24.52	11.71	10.08	2.34
	Laplace, Eipro	5.94	1.13	0.02	0.0	0.0	10.35	2.29	0.11	0.05	0.01
	Gauss, Interp	0					0				
	Laplace, Interp	0					0				
mse (test)	Gauss, Eipro	7.91	7.35	6.90	6.72	6.61	1.30	1.23	0.84	0.79	0.69
	Laplace, Eipro	7.56	6.76	6.22	6.11	6.09	1.15	0.87	0.61	0.59	0.59
	Gauss, Interp	7.26					0.96				
	Laplace, Interp	6.09					0.59				
mse (train)	Gauss, Eipro	5.87	4.49	2.82	1.67	0.78	0.99	0.79	0.32	0.22	0.09
	Laplace, Eipro	3.87	1.82	0.31	0.027	7.7e-4	1.06	0.52	0.04	0.007	0.002
	Gauss, Interp	5.5e-7					2.3e-22				
	Laplace, Interp	5.7e-9					7.5e-28				

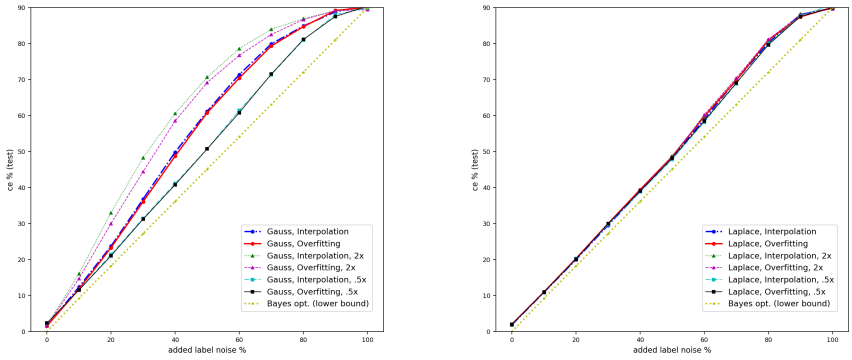
Table 5: HINT-S-20k and 20 Newsgroups summary table.

added label noise = 0%	Methods	Data Size (MNIST)				
		500	1500	3000	7500	15000
ce % (test)	Gauss, Interpolation	9.35	5.49	4.10	2.64	2.07
	Laplace, Interploation	11.27	6.43	4.94	3.38	2.39
	KNN-1	14.55	9.53	7.71	5.76	4.60
	KNN-3	15.84	9.66	7.50	5.52	4.34
	KNN-5	16.68	9.83	7.47	5.47	4.32

Table 6: Classification error (%) for test data vs training data size with 0% added label noise for MNIST dataset.

added label noise = 10%	Methods	Data Size (MNIST)				
		500	1500	3000	7500	15000
ce % (test)	Gauss, Interpolation	20.98	16.32	14.00	13.01	12.16
	Laplace, Interploation	21.08	15.49	13.85	12.38	11.41
	KNN-1	28.90	24.72	22.78	21.90	21.21
	KNN-3	27.13	20.41	17.91	15.98	14.76
	KNN-5	25.37	18.66	16.05	14.39	13.06

Table 7: Classification error (%) for test data vs training data size with 10% added label noise for MNIST dataset.



(a) Gauss (b) Laplace

Figure 8: Overfitted classifiers, interpolated classifiers, and Bayes optimal for MNIST datasets with added label noise with different kernel bandwidth. y axis: classification error on test data.