DISCUSSION OF “NONPARAMETRIC REGRESSION USING DEEP NEURAL NETWORKS WITH RELU ACTIVATION FUNCTION”

BY BEHROOZ GHORBANI, SONG MEI, THEODOR MISIAKIEWICZ AND ANDREA MONTANARI

Stanford University

We congratulate Johannes Schmidt-Hieber for his elegant and thought-provoking results. His article uses deep-learning-inspired methods in the context of nonparametric regression. Schmidt-Hieber defines a rich class of composition-based functions \( G(q, d, t, \beta, K) \) and a class of sparse multi-layer neural networks \( F(L, p, s, F) \). He proves that least square estimation over the class of sparse neural networks (with suitably chosen architecture \( (L, p, s, F) \)) achieves nearly minimax prediction error over \( G(q, d, t, \beta, K) \).

The modeling and analysis in this paper are both elegant and original. They trigger a natural question: how much of the empirical success of deep learning can be understood using this model? As a way to stimulate reflection on this question, we will discuss three challenges: 1. Sparsity and generalization; 2. Curse of dimensionality; 3. Computation.

Throughout, we will denote by \( \epsilon^* = \min_{0 \leq i \leq q} \left[ (2 \beta_i^*/(2 \beta_i^* + t_i)) \right] \in (0, 1) \) the minimax exponent in the class \( G(q, d, t, \beta, K) \). Also, in our discussion we shall focus on multi-layer perceptrons, and in particular we exclude convolutional networks. The latter have entirely different structure, and they do not follow within the scope of the present paper.

1. **Sparsity and generalization.** Modern multi-layer neural networks are highly overparametrized. Schmidt-Hieber uses sparsity of the weights as a gauge to control the model’s complexity and hence to be able to bound the generalization error using tools from empirical process theory.

Is sparsity the right complexity measure in practical deep learning methods? The present paper requires the number of non-zero weights to be \( s \asymp n^{1-\epsilon^*} \log n \). As an example, consider the VGG-19 architecture [13] which is a state-of-the-art deep network trained on ImageNet\(^1\). This network has approximately \( 143 \cdot 10^6 \) parameters, of which \( 123 \cdot 10^6 \) in the fully-connected layers. Figure 1 reports the distribution of these weights after training; we are not able to identify any sparsity structure. Notice that –for ImageNet– the sample size is roughly \( n \approx 2 \cdot 10^6 \), hence much smaller than the number of non-zero coefficients.

The nascent research community in theoretical deep learning is well aware of the fact that some measure of complexity is necessarily controlled by overparametrized neural networks. A popular heuristic explanation uses the notion of ‘implicit regularization’: model complexity is not controlled by an explicit penalty or procedure, but by the dynamics of stochastic gradient descent (SGD) itself [12]. Defining the precise complexity measure that is implicitly controlled by SGD is an open problem, except in some simple examples [14, 8, 6]. A parallel line of work directly analyzes gradient descent, and shows that the generalization error can be controlled even in the presence of infinitely overparametrized networks, as long as gradient descent is stopped early [9, 3].

2. **Curse of dimensionality.** An important achievement of the present paper is to establish a dimension-independent error rate \( R(f_n, f_0) \lesssim n^{-\epsilon} \log^2 n \) (with \( \epsilon^* \) independent of \( d \) for many cases

\(^{\text{Supported by grants NSF DMS-1613091, CCF-1714305, IIS-1741162, and ONR N00014-18-1-2729, NSF DMS-1418362, NSF DMS-1407813.}}\)

\(^{\text{\(1\)}The trained parameters were downloaded from Keras 2.2.4}}\)
the present results are strong enough to break the curse of dimensionality. As a consequence, in the case of additive models, we need

\[ n \rightarrow \infty \]

empirical distribution are

\[ \{ -g \} \]

class

\[ G \]

independent rate appears to be an important guiding principle for the construction of the function

\[ (1) \]

\( f_0(\beta^i_1, \ldots, \beta^i_d, 1, d, (\beta, (\beta \lor 2)d), (K + 1)d) \).

The present paper implies convergence at rate

\[ R(\hat{f}_n, f_0) \lesssim n^{-\frac{2d}{2d+1}} \log^2 n. \]

Examining the proof, we find that this bound holds for

\[ n \geq \max_i (\beta^i_1 + 1)^{2\beta^i_1 + i}. \]

Indeed, the proof of Theorem 1 requires

\[ N = c \cdot \max_{i=0, \ldots, q} n^{i/(2\beta^i_1 + i)} \]

for a c small enough. The statement of Theorem 5 requires

\[ N \geq (\beta_1 + 1)^{i_1}. \]

This gives a lower bound of \( n \) for Theorem 1 to hold, which is

\[ n \geq \max_i (\beta^i_1 + 1)^{2\beta^i_1 + i}. \]

As a consequence, in the case of additive models, we need

\[ n \gtrsim d^d. \]

While the example (1) can be treated via an ad hoc analysis, it raises the question of whether the present results are strong enough to break the curse of dimensionality.

As a side remark, the condition \( n \gtrsim d^d \) is not necessary for learning the model (1), see e.g. [1].

---

2One possible fix of this issue is to treat the function \( g_1 \) as the composition of \( k = \log_2 d \) functions (assuming \( k \) is an integer),

\[ g_1 = g_1,k \circ g_1,k-1 \circ \ldots \circ g_1,1, \]

where \( g_1,i : \mathbb{R}^{d^{i-1}} \to \mathbb{R}^{d^i}, \)

\[ z \mapsto (z_1 + z_2, \ldots, z_1 + z_2 - 1), \]

and

\[ f_0(\beta^i_1, (\beta \lor i)d), (K + 1)d). \]

---
3. Computation. Classical statistical theory views statistical questions as decoupled from computational ones. Schmidt-Hieber’s contribution belongs to this tradition: it postulates an estimator \( \hat{f}_n \) that is a near-minimizer of the empirical risk, and derives statistical rates for this estimator. In contrast, a broad research effort in modern high-dimensional statistics is emphasizing the fundamental role played by computational bottlenecks. For a large number of problems, there are fundamental computational limitations that are dramatically more stringent than statistical ones. A somewhat arbitrary list of examples include high-dimensional regression (for certain types of prior information) [5], matrix factorization [11], community detection [7], sparse principal component analysis [2], tensor principal component analysis [10] and so on.

In practice, multi-layer neural networks are efficiently learnt via SGD or its variants. This seems to us as an important constraint on any statistical theory aiming at explaining the success of deep learning.

We consider the problem of learning a simple ridge function

\[
f_\ell(x) = \varphi_\ell(⟨θ, x⟩)
\]

with the following choices of the nonlinearity \( \varphi_\ell, \ell ∈ \{1, 2\} \):

\[
\varphi_1(x) = \frac{\tanh(x)}{0.628}, \quad \varphi_2(x) = \frac{1}{0.1275} \left( \tanh(x) + c_1 \tanh^3(x) + c_2 \tanh^5(x) \right),
\]

where \( c_1 = -3.422, c_2 = 2.551 \). These coefficients are chosen in such a way that \( \mathbb{E}\{\varphi_1(G)^2\} \approx 1 \) (for \( G ∼ \mathcal{N}(0, 1) \)), and \( \varphi_2 \) has vanishing projection in \( L^2(e^{-x^2/2}dx/\sqrt{2\pi}) \) onto the space of polynomials of degree at most four. (See [4] for a related construction in the statistics literature.) Both of these regression functions belong to the class \( \mathcal{G}(q, d, t, β, K) \) with \( d_0 = t_0 = d, d_1 = t_1 = 1, d_2 = 1, and β_1 = β_2 = ∞ \). The theory developed in the present paper suggests that it should be possible to estimate them at the nearly parametric rate \( (\log n)^2/n \), without distinguishing between \( \varphi_1 \) and \( \varphi_2 \).

We choose the true parameter vector \( θ \in \mathbb{R}^d \) uniformly at random with \( \|θ\|_2 = 1 \). We consider data \( (y_i, x_i) \), where \( x_i ∼ \text{Unif}([-a, a]^d), a = \sqrt{3} \) (to fix the normalization \( \mathbb{E}\{\|x\|_2^2\} = d \)), and \( y_i = f_\ell(x_i) \) for either of the two models \( f_\ell(x) = \varphi_\ell(⟨θ_\ell, x⟩) \), \( \ell ∈ \{1, 2\} \).

We try to learn these functions by fitting fully connected ReLU networks of various depths, using SGD. Figure 2 reports the results of our experiments. We use \( d = 500 \), while the number of neurons in each hidden layer is fixed to 100. We vary the number of training data points, \( n \), from \( 75k \) to \( 250k \). Our data suggest that these networks, independently of their depth, have difficulty in learning \( f_2(·) \). At the same time, \( f_1(·) \) is learnt even with a small amount of training data.

References.


Fig 2. Test (solid lines) and training (dashed lines) loss for fully connected networks trained on the two data distributions $f_\ell(x) = \varphi_\ell(\langle \theta_\ell, x \rangle)$, $\ell \in \{1, 2\}$. We consider networks with \{1, 2, 3\} hidden layers. Each model is trained for 150 epochs via SGD. For reference, the loss of the trivial predictor $\hat{f}(x) = 0$ is 1.