The Annals of Statistics
2020, Vol. 48, No. 4, 1916–1921
https://doi.org/10.1214/19-AOS1931
Main article: https://doi.org/10.1214/19-AOS1875

REJOINDER: "NONPARAMETRIC REGRESSION USING DEEP NEURAL NETWORKS WITH RELU ACTIVATION FUNCTION"

BY JOHANNES SCHMIDT-HIEBER

Department of Applied Mathematics, University of Twente, a.j.schmidt-hieber@utwente.nl

The author is very grateful to the discussants for sharing their viewpoints on the article. The discussant contributions highlight the gaps in the theoretical understanding and outline many possible directions for future research in this area. The rejoinder is structured according to topics. We refer to [GMMM], [K], [KL] and [S] for the discussant contributions by Ghorbani et al., Kutyniok, Kohler & Langer and Shamir, respectively.

1. Overparametrization and implicit regularization. One of the general claims about deep learning is that, even for extreme overfitting, the method still generalizes well. There are numerous experiments showing that running the training error to zero and, therefore, interpolating all data points results in state-of-the-art generalization performance. The rationale behind this is that among all solutions interpolating the data points, of which most result in bad generalization behavior, stochastic gradient descent (SGD) picks a minimum norm interpolant. This is also known as implicit regularization. While this is well known for stochastic gradient descent applied to linear regression, for deep networks some progress has been made recently in finding the norm minimized by (S)GD; see [10, 23].

It is now reasonable to wonder whether the notion of network sparsity could be removed in the article if implicit regularization would have been taken into account. [GMMM] write that "Model complexity is not controlled by an explicit penalty or procedure, but by the dynamics of stochastic gradient descent (SGD) itself." [S] mentions implicit regularization to show that statistical guarantees should involve specific learning methods.

We conjecture that for additive error models, such as the nonparametric regression model considered in the article, implicit regularization in the overfitted regime is insufficient to achieve even consistency. To support our conjecture, we provide the following two-step argument. In the first step we argue that for one-dimensional input and shallow networks with fixed parameters in the first layer, SGD will converge to a variant of the natural cubic spline interpolant. In the second step we show that this reconstruction leads to an inconsistent estimator if additive noise is present.

A shallow ReLU network with one input and one output node can be written as $x \mapsto \sum_{j=1}^m a_j(b_j x - c_j)_+$. We now study an even more simplified setup where b_j is always one. For small $\delta > 0$, $(x - c_j)_+ \approx \int_{c_j}^{c_j+\delta} (x-u)_+ \, du/\delta$. This motivates to study smoothed shallow ReLU networks of the form

$$x \mapsto f_{\mathbf{a}}(x) = \sum_{j=1}^{m} \frac{a_j}{\sqrt{t_j - t_{j-1}}} \int_{t_{j-1}}^{t_j} (x-u)_+ \, du$$

with parameter vector $\mathbf{a} = (a_1, \ldots, a_m)$ and fixed $t_0 < t_1 < \cdots < t_m$. For convenience, we have rescaled the parameters a_j so that the normalization factor becomes $1/\sqrt{t_j - t_{j-1}}$. We consider the overparametrized regime $m \geq n$ assuming that, for any i, there lies at least one t_j in the interval $[X_{(i-1)}, X_{(i)})$ with $X_{(i)}$ the ith order statistic of the sample $X_1, \ldots, X_n$ and $X_{(0)} = -\infty$. Under overparametrization this is a rather weak assumption and ensures

Received November 2019.

existence of a shallow ReLU network $f_{\mathbf{a}}^*$ perfectly interpolating the data in the sense that $f_{\mathbf{a}}^*(X_i) = Y_i$ for all i.

For initialization at zero and properly chosen learning rate, SGD with respect to the least squares loss converges to the minimum norm interpolant with parameter vector

$$\mathbf{a}^* = \arg\min_{\mathbf{a}\in\mathbb{R}^m} \{\|\mathbf{a}\|_2 : f_{\mathbf{a}}(X_i) = Y_i, \forall i\}$$

(this result is due to [24] for overdetermined linear systems but can be extended to the underdetermined case; see, also, the generalizations in [8, 10, 16]). Because of $f_{\mathbf{a}}''(x) = a_j/\sqrt{t_j - t_{j-1}}$ for all $x \in (t_{j-1}, t_j)$, we find $\|\mathbf{a}\|_2 = \|f_{\mathbf{a}}''\|_{L^2[t_0,t_m]}$. It is known that the natural cubic spline interpolant L is the interpolant with the smallest L^2-norm on the second derivative. Moreover, we have that $\|f''\|_{L^2}^2 = \|L''\|_{L^2}^2 + \|L'' - f''\|_{L^2}^2$ for all twice differentiable interpolating functions f; see equation (2.9) in [9]. Since $f_{\mathbf{a}^*}$ and L are both interpolants, this implies that the SGD limit $f_{\mathbf{a}^*}$ will be close to the natural cubic spline interpolant.

In the nonparametric regression model with additive errors, the distance between the true function values and the response variables Y_i is of the order of the noise level (which is assumed to be fixed). The natural cubic spline interpolates the Y_i's. If, in a neighborhood, the Y_i's lie all on one side of the regression function, the average distance between the natural cubic spline interpolant and the true regression function will be lower bounded by a constant. Since this happens on a subset with Lebesgue measure bounded from below, the natural cubic spline interpolant is inconsistent for estimating the regression function. As the SGD limit approximates the natural cubic spline interpolant, this indicates that stochastic gradient descent should lead to inconsistent estimators.

We believe that this also holds true for deep networks. In this case it is expected that SGD still converges to a spline interpolant but not necessarily to the natural cubic spline interpolant; see, also, [21] for a related argument.

While it has been observed that there are nonparametric estimators that can interpolate and also achieve fast convergence rates in the nonparametric regression model ([3]), the argument above indicates that implicit regularization in the overfitted regime will not do that. To obtain rate optimal estimators, more regularization has to be imposed forcing the network to do smoothing.

2. Network sparsity. The article identifies sparsity of the network weights as a complexity measure to achieve optimal convergence rates under a hierarchical composition assumption. As sparsity is a nonstandard assumption, there are several comments on it in the reports. [GMMM] show that the empirical distribution of the weights in the first fully connected layer of the VGG-19 network is nearly Gaussian. [KL] mention a recent result proving optimal estimation rates for very deep networks with fully connected layers.

After the original version of this article was drafted, a large body of applied work emerged dealing either with compression through sparsifying dense networks or proposing methods that directly train a sparse neural network. Below we briefly summarize some of these approaches.

One method to achieve sparsity in neural networks is by pruning a fully connected network after training. A simple approach would be to replace small network weights by zero, but more sophisticated approaches based on the second derivative have been proposed as well; see [11, 12, 15]. [5] proposes an iterative pruning procedure; see also [7]. These approaches allow us to reduce the number of parameters in fully connected layers by about 90% without loss of efficiency.

Although Theorem 1 is formulated in terms of network sparsity, the proof explicitly constructs a network topology, that is, the graph structure defined by the nonzero connections

between successive layers, for which the minimax estimation rate is attained (up to log-factors). Instead of searching over all s-sparse networks, it is therefore, in principle, possible to start with this network topology and only learn the nonzero weights. By fixing one sparse network topology, a lot of the flexibility of networks to adapt to the underlying structure in the data might be lost. An intermediate constraint would be to impose an individual sparsity parameter for each weight matrix or to bound the indegree and outdegree for each individual unit in the network. In the applied literature choosing a sparse network topology beforehand has been proposed recently in [19, 20]. The latter article makes an interesting connection between sparsely connected neural networks and decision trees. Related to an initial choice of a sparse network topology is the evolutionary algorithm inspired by biological neural networks proposed in [17]. It starts with sparse weight matrices. In every iteration the smallest weights are removed, and new random connections are added so that the network topology changes but the overall network sparsity is kept constant. The method proposed in [1] is also inspired by the sparsity observed in biological networks. It starts with a sparse network topology and increases the sparsity by only keeping the units in each hidden layer that channel most of the signal to the next layer.

The recent work [6] on weight agnostic neural networks takes this one step further. No training is done, and the weights are fixed to the initialized values at all times. Only the network topology is learned by an iterative procedure. In each step of the iteration, we have a set of candidate models. For each of those models a score is computed. "Around" the models with the highest scores a new set of randomly generated candidate models is generated.

Theorem 1 in [KL] considers neural networks with fixed width and depth increasing polynomially in the sample size. It is shown that for such extremely deep networks, the empirical risk minimizer over fully-connected layers achieves the optimal estimation rate, and no sparsity is needed. Such architectures are, however, in many aspects quite different compared to the neural networks considered in practice. In [25], it has been observed that, for such extremely deep networks, one needs discontinuous weight assignments to achieve the best possible approximation rate. This is a strange phenomenon which could hint at some issues with the stability during learning of the network weights.

3. Classification and nonparametric regression. While the article deals with data from the nonparametric regression model, the overwhelming part of the literature on deep learning is on classification. Nonparametric regression and estimation of the conditional class probabilities in classification is similar, if a fraction of the data is mislabeled which prevents the conditional class probabilities to be close to zero or one. For the commonly considered classification tasks in deep learning, this is, however, not the case as most of the data are correctly labeled. As the randomness due to mislabeling is negligible in those cases, the only remaining randomness is in the distribution of the design/inputs and reconstruction becomes rather an interpolation than a denoising problem. If the different classes are also well separated from each other, much faster convergence rates can be achieved. This explains why the sample complexity in the nonparametric regression model is much higher than what is observed in deep learning for object recognition tasks; see also Report [S].

Concerning the statistical properties there are some differences. For image classification problems, deep learning is, for instance, not robust to Gaussian perturbations; see [13]. In the nonparametric regression model, Gaussian perturbations just increase the noise level. Since the noise level appears in the estimation risk bounds through the constants, the estimation rates for the class of estimators considered in the article will not change under additive noise perturbations.

We would like to stress again that the structure of the data is essential for the behavior of deep learning and the properties of the reconstructions. One of the challenges for future research will be to study estimation in models beyond nonparametric regression.

4. Algorithms. [GMMM] and [S] question whether one can disentangle the algorithm from the statistical analysis. We would like to stress that Theorem 1 is not about one fixed estimator. It provides bounds for any estimator which, given data, returns a sparsely connected neural network. The method/estimator determines the term $\Delta_n(\widehat{f}_n, f_0)$ defined in equation (5) and Theorem 1 shows that $\Delta_n(\widehat{f}_n, f_0)$ tightly controls the estimation risk from above and below. This is different than the case of data interpolation and training error zero, where $\Delta_n(\widehat{f}_n, f_0)$ is not sufficient anymore to fully characterize the statistical properties; see, also, Report [S] and [26].

We agree that the difficulty is shifted to a precise estimate of the term $\Delta_n(\widehat{f}_n, f_0)$, and we hope to study this term in more detail in future work. This term might heavily depend on the learning rate, the initialization and the energy landscape. Regarding a question in [K], the expectation in the definition of $\Delta_n(\widehat{f}_n, f_0)$ (equation (5) in the article) can be taken over all the randomness, including additional randomization in the algorithm.

While it would be desirable to have precise theoretical bounds for the performance of the most popular deep learning methods such as Adam, we believe that some amount of idealization and simplification is unavoidable. In statistical theory this seems to be widely accepted. For instance, most of the theory on the LASSO deals with regularization parameters derived from large deviations bounds although the standard software implementations choose the regularization parameter by 10-fold cross-validation.

5. High-dimensional input. [GMMM] mentions that, for the current proof strategy and the case of additive models, the dependence of the dimension on the constants is d^d. As mentioned in the article, the results focus on the convergence rates; no attempt has been made to minimize the constants appearing in the proofs. In fact, by a variation of the original argument the dependence on the dimension for additive models $f(\mathbf{x}) = \sum_{i=1}^d f_i(x_i)$ can be shown to be linear. To see this, we can build for any given $N \geq 1$, d separate networks with $s \asymp N \log N$ parameters, computing the functions $f_1(x_1), \ldots, f_d(x_d)$ up to an approximation error of the order $O(N^{-\beta})$. Using the parallelization rule mentioned on p. 21 of the article, one can then combine the individual networks into a large neural network computing the sum $\sum_{i=1}^n f_i(x_i)$ up to an approximation error of order $O(dN^{-\beta})$ using $s \asymp dN \log N$ many network parameters. It then follows from Theorem 2 that the rate is upper bounded by $dn^{-2\beta/(2\beta+1)} \log^3 n$ if $\Delta_n(\widehat{f}_n, f_0)$ is sufficiently small and d is bounded by a power of the sample size.

As another result on high-dimensional input, [S] mentions a theorem proving that basis expansions have difficulties to approximate functions generated by a single neuron. Either huge coefficients are needed or the number of basis functions has to be exponential in the input dimension.

Since the input dimension d in deep learning applications is typically extremely large, a possible future direction would be to analyze neural networks with high-dimensional $d = d_n \uparrow \infty$ and comparing the rates to other nonparametric procedures.

6. Function classes. With respect to the considered function class, [K] emphasizes that the function classes should be detached from the method. On the contrary, [S] favors an alternative approach where the underlying function class consists itself of neural network functions. We believe that both approaches have advantages and disadvantages.

The imposed class of composition functions in the article appears, of course, naturally given the composition structure of deep networks. Compositions are fundamental operations and, as mentioned in the article, many widely studied function classes in nonparametric statistics such as (generalized) additive models occur as special cases of the imposed composition constraint.

For a recent result in the statistical literature with function class consisting of neural network functions, we refer to [2]. One possibility for future research would be to determine

the maxisets for neural networks, that is, the largest possible function class for which a prespecified estimation rate can be obtained; see [14]. The main advantage of generic function spaces, such as Hölder classes, is that we can compare the estimation rates achieved by different methods and, therefore, learn something about the strength and weaknesses of these methods. The article shows, for instance, that wavelet methods have a slower rate of convergence for generalized additive models than sparsely connected deep ReLU networks.

To obtain fast estimation rates, an alternative is to impose structure on the design; see [18, 22].

7. Choice of the activation function. On p. 12 in the article, we highlight several specific properties of the ReLU activation function such as the possibility to easily learn skip connections. [KL] mention that results for ReLU networks automatically transfer to other activation functions. The argument, however, requires that the network parameters will become large. In the meantime we better and better understand how SGD leads to norm control on the parameters. To model this, we think that it is important to control the magnitude of the weights in the network classes. In the article the network parameters are bounded in absolute value by one. This is a convenient choice, but as our understanding of the norm control induced by SGD improves, more realistic constraints are imaginable. It is well known that training does not move the parameters far from the initialized values. To analyze the effect of different initialization strategies, one possibility would be to study network classes generated by all parameters in a neighborhood of a (random) initializer.

8. Real data. [GMMM] report the results of a simulation study which seemingly contradict the theory in the article. They study the noise-free case and up to three hidden layers showing that a certain smooth function cannot be learned. We would like to refer to the simulation study in [4] which finds that, for regression problems, the performance of deep neural networks is not far off from the theoretical bounds. This article also examines the finite sample performance of the multiplication network in Lemma A.2 which forms an essential part in the proof of Theorem 1. To a certain extent, even such specific constructions can be picked up by deep learning. This, however, only works for a careful initialization. It might be necessary to reinitialize the procedure if the algorithm gets stuck in a local minimum with large training error.

Acknowledgments. The author would like to thank Misha Belkin and Dirk Lorentz for fruitful discussions on overfitting and SGD.

REFERENCES

[1] AHMAD, S. and SCHEINKMAN, L. (2019). How can we be so dense? The benefits of using highly sparse representations. arXiv e-prints arXiv:1903.11257.

[2] BARRON, A. R. and KLUSOWSKI, J. M. (2018). Approximation and estimation for high-dimensional deep learning networks. arXiv e-prints arXiv:1809.03090.

[3] BELKIN, M., RAKHLIN, A. and TSYBAKOV, A. B. (2019). Does data interpolation contradict statistical optimality? In *Proceedings of Machine Learning Research* (K. Chaudhuri and M. Sugiyama, eds.). *Proceedings of Machine Learning Research* **89** 1611–1619. PMLR.

[4] ECKLE, K. and SCHMIDT-HIEBER, J. (2019). A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Netw.* **110** 232–242. https://doi.org/10.1016/j.neunet.2018.11.005

[5] FRANKLE, J. and CARBIN, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv:1803.03635.

[6] GAIER, A. and HA, D. (2019). Weight agnostic neural networks. arXiv e-prints arXiv:1906.04358.

[7] GALE, T., ELSEN, E. and HOOKER, S. (2019). The state of sparsity in deep neural networks. arXiv e-prints arXiv:1902.09574.

[8] GOWER, R. M. and RICHTÁRIK, P. (2015). Randomized iterative methods for linear systems. *SIAM J. Matrix Anal. Appl.* **36** 1660–1690. MR3432148 https://doi.org/10.1137/15M1025487

[9] GREEN, P. J. and SILVERMAN, B. W. (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach. Monographs on Statistics and Applied Probability* **58**. CRC Press, London. MR1270012 https://doi.org/10.1007/978-1-4899-4473-3

[10] GUNASEKAR, S., LEE, J., SOUDRY, D. and SREBRO, N. (2018). Characterizing implicit bias in terms of optimization geometry. arXiv e-prints arXiv:1802.08246.

[11] HAN, S., POOL, J., TRAN, J. and DALLY, W. (2015). Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems* 28 (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, eds.) 1135–1143. Curran Associates, Red Hook, NY.

[12] HASSIBI, B. and STORK, D. G. (1993). Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems* 5 (S. J. Hanson, J. D. Cowan and C. L. Giles, eds.) 164–171. Morgan Kaufmann, Burlington, MA.

[13] HENDRYCKS, D. and DIETTERICH, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

[14] KERKYACHARIAN, G. and PICARD, D. (2002). Minimax or maxisets? *Bernoulli* **8** 219–253. MR1895892

[15] LECUN, Y., DENKER, J. S. and SOLLA, S. A. (1990). Optimal brain damage. In *Advances in Neural Information Processing Systems* 2 (D. S. Touretzky, ed.) 598–605. Morgan Kaufmann, Burlington, MA.

[16] MANSEL GOWER, R. and RICHTARIK, P. (2015). Stochastic dual ascent for solving linear systems. arXiv e-prints arXiv:1512.06890.

[17] MOCANU, D. C., MOCANU, E., STONE, P., NGUYEN, P. H., GIBESCU, M. and LIOTTA, A. (2018). Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nat. Commun.* **9** 2383. https://doi.org/10.1038/s41467-018-04316-3

[18] NAKADA, R. and IMAIZUMI, M. (2019). Adaptive approximation and estimation of deep neural network with intrinsic dimensionality. arXiv e-prints arXiv:1907.02177.

[19] PRABHU, A., VARMA, G. and NAMBOODIRI, A. (2018). Deep expander networks: Efficient deep networks from graph theory. In *Computer Vision—ECCV* 2018 (V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss, eds.) 20–36. Springer, Cham.

[20] ROBINETT, R. A. and KEPNER, J. (2018). Neural network topologies for sparse training. arXiv e-prints arXiv:1809.05242.

[21] SAVARESE, P., EVRON, I., SOUDRY, D. and SREBRO, N. (2019). How do infinite width bounded norm networks look in function space?. arXiv e-prints arXiv:1902.05040.

[22] SCHMIDT-HIEBER, J. (2019). Deep ReLU network approximation of functions on a manifold. arXiv e-prints arXiv:1908.00695.

[23] SOUDRY, D., HOFFER, E., NACSON, M. S., GUNASEKAR, S. and SREBRO, N. (2018). The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.* **19** Paper No. 70, 57. MR3899772

[24] STROHMER, T. and VERSHYNIN, R. (2009). A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.* **15** 262–278. MR2500924 https://doi.org/10.1007/s00041-008-9030-4

[25] YAROTSKY, D. (2018). Optimal approximation of continuous functions by very deep ReLU networks. In *Proceedings of the* 31*st Conference on Learning Theory* (S. Bubeck, V. Perchet and P. Rigollet, eds.). *Proceedings of Machine Learning Research* **75** 639–649. PMLR.

[26] ZHANG, C., BENGIO, S., HARDT, M., RECHT, B. and VINYALS, O. (2016). Understanding deep learning requires rethinking generalization. arXiv e-prints arXiv:1611.03530.