COMMENTARY

Another step toward demystifying deep neural networks

Michael Elad[a,1], Dror Simon[a], and Aviad Aberdam[b]

The field of deep learning has positioned itself in the past decade as a prominent and extremely fruitful engineering discipline. This comeback of neural networks in the early 2000s swept the machine learning community, and soon after found itself immersed in practically every scientific, social, and technological front. A growing series of contributions established this field as leading to state-of-the-art results in nearly every task, recognizing image content, understanding written documents, exposing obscure connections in massive datasets, facilitating efficient search in large repositories, translating languages, enabling a revolution in transportation, revealing new scientific laws in physics and chemistry, and so much more. Deep neural networks not only solve known problems but offer, in addition, unprecedented results in deploying learning to problems that until recently were considered as hopeless or only weakly successful. These include automatically synthesizing text–media, creating musical art pieces, synthesizing realistic images and video, enabling competitive game-playing, and this list goes on and on.

Amazingly, all these great empirical achievements are obtained with hardly any theoretical foundations that could provide a clear justification for the architectures used, an understanding of the algorithms that accompany them, a clear mathematical reasoning behind the various tricks employed, and above all, the impressive results obtained. The quest for a theory that could explain these ingredients has become the Holy Grail of data sciences. Various impressive attempts to provide such a theory have started to appear, relying on ideas from various disciplines (see, e.g., refs. 1–9). The paper by Papyan et al. (10) in PNAS adds an important layer to this vast attempt of developing a comprehensive theory that explains the behavior of deep learning solutions. In this Commentary, we provide a wide context to their results, highlight and clarify their contribution, and raise some open questions as a roadmap for further investigation.

Deep Neural Networks: The Great Riddle

We start with a primer to the core concepts behind deep learning and narrow the discussion to the context of "supervised classification" tasks, as highlighted in ref. 10. We are given a large training dataset of N signal examples, $\{x_i\}_{i=1}^N \in \mathbb{R}^n$, each belonging to one of C categories, and our goal is to use these for designing a machine that operates on new signals from the same origin, aiming to classify them as accurately as possible. Deep neural networks offer one such design strategy (among many alternatives), which has proven to perform extremely well. The network itself can be described as a function $z = h(x, \Theta)$ that operates on the input signal x while being parametrized by Θ, creating a feature vector $z \in \mathbb{R}^p (p \geq C)$. This feature vector is then fed to a linear classifier of the form $\text{SoftMax}\{\mathbf{W}z + b\} \in \mathbb{R}^C$ for getting the actual classification assignment.

Networks of the above form—$h(\cdot)$ and the linear classification layer that follows—tend to have many (often millions) free parameters to be tuned, and their architectures typically consist of many layers (thus the term "deep") of familiar computational steps that include convolutions, subsampling, pulling, rectified linear units, batch-normalization, general matrix multiplications, and more.

The common approach toward setting the network parameters $\{\Theta, \mathbf{W}, b\}$ is via supervised learning, leveraging the known labels of the training examples in the design of the classifier. The learning itself consists of a minimization of a loss function that relates to the accuracy of the classification on the training set. By minimizing this loss via a stochastic gradient descent algorithm, the network gradually improves its performance, both on the training and the test sets, suggesting generalization ability to unseen data.

While all of the above represents a common practice that seems empirically clear, almost nothing in this chain of operations/decisions is theoretically well-founded. The network architecture is chosen in an arbitrary trial-and-error fashion with no clear

[a]Computer Science Department, Technion–Israel Institute of Technology, Haifa 3200003, Israel; and [b]Electrical Engineering Department, Technion–Israel Institute of Technology, Haifa 3200003, Israel

Author contributions: M.E., D.S., and A.A. wrote the paper.

The authors declare no competing interest.

See companion article, "Prevalence of neural collapse during the terminal phase of deep learning training," 10.1073/pnas.2015509117.

[1]To whom correspondence may be addressed. Email: elad@cs.technion.ac.il.

First published October 15, 2020.

reasoning or justification; the loss function that drives the learning may assume many forms, and yet all are highly nonconvex and with no guarantees for getting the truly optimal network; and even if we have found a good minima, nothing supports our expectation for a good generalization behavior.

In addition, if all of the above is not complicated enough, a closer inspection at some of the better-performing classification neural networks reveals a troubling phenomenon: These networks are overparametrized, consisting of more parameters than the training data could ever accommodate. To better clarify this, consider a least-squares problem with more unknowns (these parallels our parameters) than equations (representing our training data). Classically, we refer to this situation as ill-posed, leading to infinitely many possible solutions with no clear way of distinguishing between them. Back to machine learning, this implies that there is a possibility to obtain a network that perfectly classifies the training data, while performing very poorly on test data, a phenomenon known as overfitting. However, deep neural networks that are overparametrized tend to work very well. How come? This brings us to the "double-descent" effect as discovered in ref. 11, which is a key property in explaining the results in ref. 10.

Fig. 1 presents this behavior in one of its simplest manifestations. First, observe the loss function value as it evolved over the training iterations, and as expected, drops (nearly) monotonically. In parallel to the training process, we inspect both the train and the test errors—actual errors obtained on example signals. Note that the train error and the loss value, while closely related, are different. The train error gets consistently smaller until it reaches (near) zero, implying that the network has memorized the data it is trained on, performing optimally on it. From here on, this error remains very small or even zero, while the network's parameters still wander around, updated by the training procedure. The double-descent behavior refers to the test error—in the first phase and until reaching the interpolation threshold, the behavior is familiar and well understood, showing a tendency to improve for a while, and then worsen due to overfitting. If training proceeds after the interpolation point, the test error resumes its descent and this time to a (much) lower error value. This demonstrates a typical behavior of highly overparametrized networks and their tendency to lead to state-of-the-art performance. Recent attempts to explain this theoretically are impressive, but still incomplete (12, 13).

Papyan et al.'s Contribution

All of this background takes us to the results brought in ref. 10. Papyan et al. conjecture that overparametrized networks converge to an ideal classification behavior after sufficient training iterations. Posed as four properties, their claims are the following:

1) The feature vectors $z = h(x, \Theta)$ of all of the examples belonging to each class concentrate in isolated points.
2) These concentration points are maximally distant and located on a sphere centered at the origin.
3) The linear classifier matrix **W** is adequately perfectly matched to these concentration points, and
4) The linear classification tends to a simple nearest-neighbor procedure.

If indeed true, these properties are nothing short of fascinating, as the outcome is a seemingly perfect classifier, both in terms of its generalization ability, and its robustness to attacks and noise. This also sheds light on the general engineering belief that deep neural networks are touching the ceiling in terms of their performance.

It is important to note that the claims made in ref. 10 are empirically oriented conjectures, obtained after a massive series of simulations on several well-known classification tasks and using several popular network architectures. As a side philosophical note, one should wonder if this is a representative of our current era, exposing a new way of performing research by relying on considerable experimental setting. Building on the above, the work in ref. 10 precedes by theoretically tying this conjectured behavior to well-known results on optimal classification and maximal margin performance.

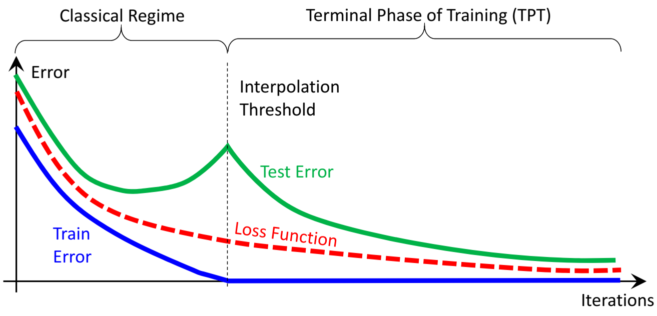

Fig. 1. The double-descent phenomenon for highly overparametrized networks.

Discussion and Thoughts: Open Questions

Origins of the Proposed Behavior. The conjectures in ref. 10 about the perfect classification behavior of deep neural networks are illuminating, yet they are brought without identifying the mechanisms that drive toward this outcome. One may argue that the loss function considered in the paper (Cross-Entropy) is explicitly pushing to concentration and maximal margin, but it is unclear why such ideal outcome is achievable in the first place. What is the role of the optimization strategy used (stochastic gradient descent)? How does the chosen architecture influence this behavior? Do these conjectures apply to any classification task? Does this behavior take place even if we replace the cross-entropy by another loss function (say mean squared error)? All these are open questions that call for further investigation.

Questioning the Concentration. A delicate matter we would like to bring up refers to the first conjecture about the concentration of the feature vectors. Handling C classes and using features of length $p > C$, we follow the notations in ref. 10 and define the matrix $\mathbf{M} \in \mathbb{R}^{p \times C}$ containing these centers as its columns. The rank of **M** is C, i.e., the centers span a subspace of dimension C embedded in $\mathbb{R}^p$. This suggests that these centers can be contaminated by arbitrary vectors orthogonal to this subspace, without impairing their later classification. As a consequence, one should wonder why perfect concentration is necessary. This question strengthens as we look at figure 6 in ref. 10, and observe a weaker concentration in some of the experiments. An interesting connection may exist between the claimed concentration and the information bottleneck (IB) concept advocated in ref. 3. At steady state after training, the IB suggests that the neural network features should carry no information about the source signals apart from content immediately relevant to the classification goal.

Elad et al. PNAS | November 3, 2020 | vol. 117 | no. 44 | 27071

A closely related issue to the above, still referring to the matrix **M** of the feature centers, has to do with its spatial orientation in $\mathbb{R}^p$. This matrix can be rotated freely by multiplying it by a unitary matrix $\mathbf{Q} \in \mathbb{R}^{C \times C}$ without any change to the later classification behavior. Thus, one should wonder what dictates the choice of **Q**. We would like to offer a complementary conjecture worth exploring, suggesting that **Q** should be such that the obtained centers are maximally simplified or explainable. Put in other words, the centers should be driven to become maximally sparse in order to make them more interpretable.

Extensions. The properties of deep overparametrized networks identified and exposed in ref. 10 are tantalizing, as they offer a simple and intuitive behavior to otherwise very complicated machines. Could one envision similar tendencies in deep neural networks handling regression or synthesis tasks? Indeed, what is the parallel of the ideal classification to this breed of networks? These are important open questions that should be addressed, in our quest to demystify neural network solutions of inverse problems, generative adversarial networks, and more.

Is It Really Ideal Classification? The main message underlying the work in ref. 10 is that the terminal phase of training (TPT) and its expected outcomes are welcome and desired. While this belief is tempting, we should highlight the risks in such a regime of training. Early stopping, or avoiding interpolation, is a well-known regularization technique that has been shown to be beneficial for natural language processing and image classification, and especially so when the dataset includes label noise (14), as common in many applications.

Another consideration has to do with the notion of miscalibration (15): TPT leads to a classifier whose output fails to represent the model's certainty. In practical applications this confidence measure could be crucial in the decision-making process. More broadly, should we push for maximally distant centers? If we classify C classes and some are semantically close to each other (e.g., cars and trucks), maybe it would make more sense to accommodate this in the obtained centers, allowing an easier migration between some of the classes?

Perspective. The discoveries exposed in ref. 10 are inspiring, suggesting an intuitive and simple explanation to deep network machines that are perceived as mysterious. This work points to many important open questions that will, no doubt, occupy the scientific community in coming years.

1 J. Bruna, S. Mallat, Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1872–1886 (2013).

2. A. B. Patel, T. Nguyen, R. Baraniuk, *A Probabilistic Framework for Deep Learning*, (NeurIPS, 2016).

3 N. Tishby, N. Zaslavsky, *Deep Learning and the Information Bottleneck Principle*, (IEEE-ITW, 2015).

4 V. Papyan, Y. Romano, M. Elad, Convolutional neural networks analyzed via convolutional sparse coding. *JMLR* **18**, 2887–2938 (2017).

5 J. Sulam, A. Aberdam, A. Beck, M. Elad, On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 1968–1980 (2020).

6 B. D. Haeffele, R. Vidal, *Global Optimality in Neural Network Training* (CVPR, 2017).

7 P. Chaudhari, S. Soatto, *Stochastic Gradient Descent Performs Variational Inference, Converges to Limit Cycles for Deep Networks* (IEEE-ITA, 2018).

8 H. Bölcskei, P. Grohs, G. Kutyniok, P. Petersen, Optimal approximation with sparsely connected deep neural networks. *SIMODS* **1**, 8–45 (2019).

9 R. Giryes, G. Sapiro, A. M. Bronstein, Deep neural networks with random Gaussian weights: A universal classification strategy? *IEEE-TSP* **64**, 3444–3457 (2016).

10. V. Papyan, X. Y. Han, D. L. Donoho, Prevalence of neural collapse during the terminal phase of deep learning training. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 24652–24663 (2020).

11 M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15849–15854 (2019).

12 S. Ma, R. Bassily, M. Belkin, *The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning* (ICML, 2018).

13 P. Nakkiran *et al.*, *Deep Double Descent: Where Bigger Models and More Data Hurt* (ICLR, 2019).

14 M. Li, M. Soltanolkotabi, S. Oymak, *Gradient Descent with Early Stopping Is Provably Robust to Label Noise for Overparameterized Neural Networks* (ICAIS, 2020).

15 C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, *On Calibration of Modern Neural Networks* (ICML, 2017).

27072 | www.pnas.org/cgi/doi/10.1073/pnas.2018957117 Elad et al.