ARTIFICIAL NEURAL NETWORKS

BY MARTIN GENZEL AND GITTA KUTYNIOK

We are currently witnessing an unprecedented success of *artificial neural networks* in both public life and various areas of sciences. Within a very few years, neural-network-based algorithms have mastered practical tasks which until recently were considered to be very difficult for machines, thereby fundamentally changing our way of thinking about artificial intelligence: Nowadays, self-driving cars are controlled by algorithms based on neural networks [1]. At the same time, similar methods can beat the world top players not only at chess, but also at the much more complex game of Go [2]; and even more impressive, the recent AlphaZero computer program is able to train itself from scratch, reaching a superhuman level of play just within 24 hours [3]. In the public sector, neural network methodologies are just starting to revolutionize the healthcare industry: for instance, they show a great potential to classify types of skin cancer [4]; and meanwhile, the U.S. Food and Drug Administration (FDA) has already approved the marketing of the first medical device for detecting diabetic retinopathy which is based on artificial intelligence [5]. But also beyond medical sciences, these technologies are affecting every single one of us, for example, when using a voice assistant on our smartphone or an online translator. These are only very few applications from a long list, and without doubt, it is still growing at a stunning speed. Finally, even when it comes to (classical) mathematical problems, many ideas from neural network theory proved very useful, such as in solving partial differential equations or ill-posed inverse problems, oftentimes forming the current state-of-the-art methods.

With regard to these very recent success stories, it may come as a surprise that artificial neural networks are by far not a new invention. In fact, they have undergone many fluctuations in popularity during the last decades, dating back to original work by McCulloch and Pitts in 1943 [6]. At that time, a key goal was to develop learning algorithms by mimicking the human brain – which can be seen a *real* neural network – ultimately aiming at a foundation of (artificial) intelligence. The "failure" of this approach can be particularly attributed to a lack of computational power in those days as well as a very limited amount of data sets available for training. Today, however, the data deluge and tremendously increased hardware power have largely eliminated these limitations, which has led to an impressive comeback of neural networks in the 2010's. With modern hardware technology, it is now possible to train *deep* neural networks of more than hundreds of layers and millions of neurons.

Nevertheless, most of the related research is still empirically driven and a sound theoretical foundation is almost completely missing. This was most prominently noted by Ali Rahimi, a researcher in artificial intelligence at Google, who claimed that machine learning algorithms which are based on trial-and-error engineering have become a form of "alchemy" [7]. In fact, the theoretical understanding of this field is still years behind empirical progress, and especially in view of many critical applications (some of which are mentioned above), a lot of fundamental research needs to be done in this direction. Not least because of this, a rapidly growing number of researchers from various areas of mathematics, such as approximation theory, optimization, and statistics, devote themselves to contribute to the development of a theory for artificial neural networks.

The purpose of this article is to provide an introduction to artificial neural networks and how they can be used to solve real-world (learning) problems. Beyond that, we will outline several aspects of a theoretical foundation – rather leading to open questions than answers – as well as recent applications to inverse problems and partial differential equations.

What Are Neural Networks?

To stand on more solid ground, let us begin with a formal definition of artificial neural networks. In its vanilla form, a neural network can be regarded as a highly structured function $\Phi: \mathbb{R}^d \to \mathbb{R}^{N_L}$ that arises from a cascade of simpler functions, taking the form

$$\Phi(x) = T_L \circ \rho \circ T_{L-1} \circ \rho \circ T_{L-2} \circ \ldots \circ \rho \circ T_1 (x).$$

Here, each function $T_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$, $l=1,\ldots,L$ (with $N_0 := d$) is assumed to be affine linear, i.e., $T_l(x) = A_l x + b_l$ for some *weight matrix* $A_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and *bias vector* $b_l \in \mathbb{R}^{N_l}$, while $\rho : \mathbb{R} \to \mathbb{R}$ is a non-linear *activation function,* which is applied entry-wise. In order to indicate that this definition strongly depends the parameter set $\theta = (A_l, b_l)_{l=1}^L$, we may also write Φ_θ instead of Φ. Let us also point out that the total number L of composition steps corresponds to the number of *layers* of the network. More specifically, the last layer is called the *output layer,* while all preceding layers are referred to as *hidden layers,* which are $L-1$ many in our case; note that the very first layer is sometimes also called the *input layer*.

The specific architecture of a neural network can be easily depicted as a graph, where the nodes – usually called *neurons* – visualize the individual variables in each layer and the edges indicate which variables of the current layer contribute to those in the next layer (the non-zero entries of the weight matrices A_l); see Figure 1 for a visualization. Finally, let us note that there exist many more variants of neural networks used in practice, refining and extending the standard definition presented here; popular examples are *convolutional neural networks (CNNs)*, *long short-term memory (LSTM),* or *generative adversarial networks (GANs);* see [8] for further reading.

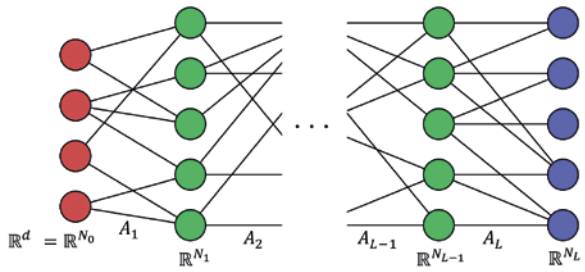

Fig. 1: Visualization of an artificial neural network as a graph. The nodes correspond to the neurons (N_l-many in the l-th layer), while the edges indicate which neurons are connected between the different layers (specified by the weight matrices A_l).

How Are Neural Networks Trained?

Let us now turn to the question of how neural networks may be used to solve practical learning tasks. Broadly speaking (and not exclusively restricting to neural networks), the formal basis of a *supervised learning problem* is a sampling process that arises from a random pair (X,Y) in $\mathbb{R}^d \times \mathbb{R}$; here, the random vector X represents the *input data,* while Y is an *output variable* which one would like to *predict* from X. In other words, we are interested in a reliable prediction of Y based on knowledge of X. Mathematically, this simply corresponds to finding the conditional expectation $\mathbb{E}(Y|X)$ or at least a (deterministic) prediction function $f:\mathbb{R}^d \to \mathbb{R}$ that approximates $\mathbb{E}(Y|X)$ sufficiently well. A prototypical scenario would be an image classification task, where X models real-world image data and Y corresponds to a (discrete) label $\{1,...,K\} \subset \mathbb{R}$, providing semantic information about the image, e.g., if it contains a cat or a dog. In practice, however, the joint probability distribution of (X,Y) is unknown, and neither $\mathbb{E}(Y|X)$ nor f are directly accessible. Instead, one is only given a finite set of (independent) samples $D=(x_i, y_i)_{i=1}^m$ from (X,Y), usually referred to as the *training data*. This restriction gives rise to a central question in learning theory: How to estimate the underlying prediction function f only on the basis of a training data set?

Perhaps the most common approach to tackle this fundamental challenge is based on *empirical risk minimization.* In the special case of neural networks, an empirical risk minimization problem typically looks as follows:

$$\min_\theta \frac{1}{m}\sum_{i=1}^{m}(y_i - \Phi_\theta(x_i))^2 + \lambda \cdot R(\theta), \qquad (1)$$

where the optimization variables $\theta = (A_l, b_l)_{l=1}^L$ are the parameters of the neural network $\Phi_\theta: \mathbb{R}^d \to \mathbb{R}$ (using the notation introduced above). Thus, the optimization in (1) takes place over all possible parameter configurations of the neural network Φ_θ; in learning theory, this set of all candidate solution functions is called the *hypothesis space.* Apart from that, R plays the role of a scalar regularization function, imposing a certain penalty on (undesirable) parameter configurations. Intuitively speaking, a minimizer $\hat{\theta}$ of (1) yields a neural network $\Phi_{\hat{\theta}}$ which fits the training data as well as possible, in the sense that the mean-squared error between $(\Phi_{\hat{\theta}}(x_i))_{i=1}^m$ and $(y_i)_{i=1}^m$ is minimized. However, the actual hope is that $\Phi_{\hat{\theta}}$ *generalizes* well, i.e., $\Phi_{\hat{\theta}}(x)$ does also accurately predict the output y of an unseen *test sample* (x,y), or more formally, the *expected risk* $\mathbb{E}(Y-\Phi_{\hat{\theta}}(X))^2$ gets sufficiently small. In this case, $\Phi_{\hat{\theta}}$ would serve indeed as a good surrogate of the unknown prediction function f, allowing us to perform reliable statistical inference.

Although the concept of empirical risk minimization may appear surprisingly simple, it in fact requires solving a highly challenging optimization problem in general. Let us point out some of the key difficulties that typically arise during the training process of a neural network:

The architecture. A crucial step before optimization is to specify the size of the parameter space: How many layers L should be permitted and how large to choose the widths $N_1,...,N_L$ of the individual layers? Should one make use of *weight sharing,* i.e., putting further restrictions on the weight matrices, such as *convolutional filters?* Although there are no ultimate answers to these questions, a striking phenomenon is that networks are often successfully trained in a highly *over-parameterized regime,* i.e., there are much more free parameters than samples. Finally, an appropriate activation function needs to be selected as well. Perhaps the most popular choice in modern network architectures is the *ReLU (rectified linear unit),* $\rho(t) := \max(0,t)$ for $t \in \mathbb{R}$; and although non-differentiable, this activation function proved surprisingly efficient for the training process.

Non-convexity. The hypothesis space generated by neural networks is a highly non-convex set in general, which turns (1) into a difficult non-convex optimization problem, mostly with a non-differentiable objective function. The most common solver in practice is *stochastic gradient descent.* A key characteristic of this algorithmic method is that, in each update for θ, a small subset of samples – called a *mini-batch* – is randomly selected and the gradient computation is only performed on this mini-batch instead of the entire sample set. This strategy is much more memory-efficient than standard gradient descent methods, which are mostly intractable for huge training sets. Noteworthy, the gradient can be very efficiently computed by *backpropagation* for neural networks, which relies on a layer-wise application of the ordinary chain rule.

Initialization & Regularization. Committing on stochastic gradient descent comes along with a series of algorithmic issues: How to initialize correctly? How to select the step size (the learning rate) and what is a good choice of the batch size? Should one use additional regularization? All these questions are delicate, and similar to the choice of architecture, there are no universal strategies yet. Common wisdom at least suggests that random initializations may lead to a desirable convergence behavior. Moreover, several types of (explicit or implicit) regularization are popular, such as *drop-out, early stopping, or weight decay* (i.e., R corresponds to an ℓ_2-penalty in (1)). Nevertheless, successful training of neural networks often requires a particular expertise and involves a significant amount of hyperparameter tuning.

In view of all these difficulties, it is an even more astonishing fact that statistical learning with neural networks still works so well in practice. At the latest, since the triumph

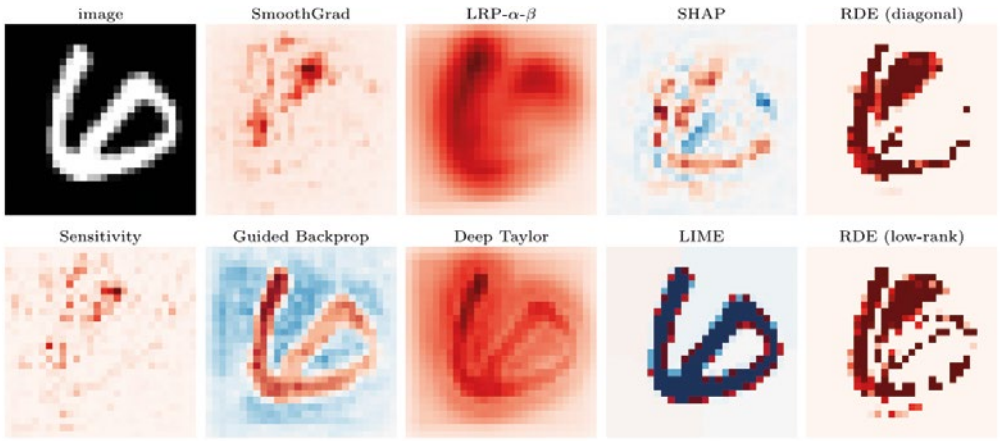

Fig. 2: This figure shows relevance scores for a handwritten digit from the MNIST dataset generated by several explanation methods, visualized as so-called heat maps. Some of these methods focus entirely on "positively" relevant features that speak for the classifier decision (shown in red): Sensitivity Analysis [19], Deep Taylor Decompositions [20], SmoothGrad [21], and Rate Distortion Explanations [22]. The other methods also highlight "negatively" relevant features speaking against the decision (shown in blue): Layer-wise Relevance Propagation [23], Guided Backprop [24], LIME [25], and SHAP [26].

of AlexNet in 2012 [9] – winning the popular ImageNet contest with a CNN-based approach –, people began to realize that especially the ability to train deep networks is a "game-changer" to many problems in machine learning, and far beyond. This fundamental insight can be seen as the birth of *deep learning,* which nowadays influences so many areas in computer science, statistics, and applied mathematics. For further reading, a concise overview on the successes of deep learning can be found in [10] and a comprehensive introduction to this field in [8].

The Many Mysteries of Deep Learning

Regardless of the outstanding empirical achievements of deep learning, its theoretical foundation remains widely open. In fact, many phenomena observed in practice are by far not explainable by traditional learning theory, so that trained neural networks often operate as black boxes. It is fair to say that a rigorous understanding of neural network theory is still in its infancy, while most available theoretical results do only address very specific aspects of the learning process or rely on unrealistic assumptions. But perhaps it is precisely this lack of theory which has made deep learning so attractive to many scientists and which has led to a whole new field of research, commonly known as the *mathematics of deep learning.* The literature devoted to the theoretical analysis of deep learning has become extensive by now, going far beyond what can be surveyed here. In the context of this article, we wish to point out some of the most important research questions towards a more profound understanding of artificial neural networks and deep learning. Let us emphasize that it is not necessarily fruitful to consider the following three subjects as independent disciplines; the key difficulty is rather to investigate them within a common mathematical framework and thereby to examine their mutual interplay.

Expressivity & Approximation. One of the fundamental pillars of a learning problem is the *expressive power* of the hypothesis space. Focusing on the setup of neural networks from the previous section, this may be reduced to the following question: How well can one approximate the unknown prediction function f by a neural network Φ_θ of fixed architecture, e.g., when the depth L and width parameters $N_1,\dots,N_L$ are fixed? This research branch is in fact one of the oldest in neural network theory and probably the furthest developed one. Classical results dating back to the 1980's [11] [12] [13] promote so-called *universal approximation theorems,* which essentially state that every continuous function on a compact domain can be arbitrarily well-approximated by a neural network with one hidden layer. Driven by the success of deeper networks, more recent approaches concern the benefit of depth

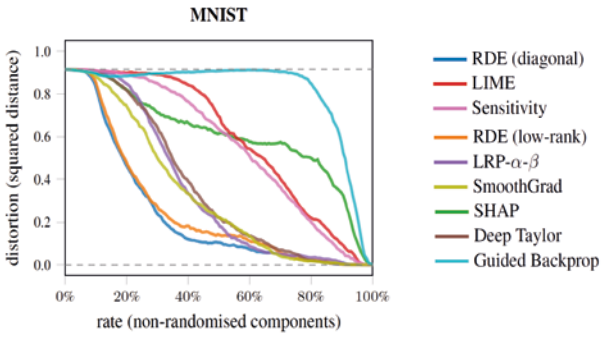

Fig. 3: A quantitative assessment of the approaches visualized in Fig. 2 can be done by a relevance-ordering-based test: going from right to left on the horizontal axis, more and more pixels of the ground truth image are randomized in an order specified by the respective relevance score (from less to more relevant pixels). The vertical axis corresponds to the mean squared error between the classifier output of the ground truth and randomized image. A steep descent of the resulting curve indicates that particularly relevant pixels were identified correctly.

when approximating complicated functions, e.g., see [14] [15] and the references therein. However, most available results do only provide *uniform* approximation guarantees for "generic" function classes which contain "worst-case" functions with much worse approximation properties than the "true" predictor f. Hence, we believe that there is still no satisfactory answer to the following fundamental issues in approximation theory: How to model a "realistic" prediction function f tailored to a specific learning task, and how well can it be approximated by a neural network? Furthermore, what role is played by the domain of f, which may not be the full $\mathbb{R}^d$ but rather samples from the input vector X?

Optimization & Generalization. As already indicated in the course of the empirical risk minimization problem in (1), the "holy grail" of deep learning is the *generalization performance* of trained neural networks, i.e., their ability to correctly predict the output of unseen (test) samples. This fundamental feature is inevitably connected to the underlying optimization task, which is highly non-convex in our case. Indeed, in stark contrast to convex optimization, (1) does not have a unique global minimizer in general, and instead there might be (infinitely) many solutions and spurious local minima. Hence, a key issue is not only whether stochastic gradient descent converges to any global minimizer, but much more importantly, why it yields a minimizer that generalizes well; in other words: what is so special about the method of stochastic gradient descent and why can it learn effective representations of the underlying prediction problem? Another remarkable fact is that this approach often works particularly well in an over-parameterized regime, meaning that there are (much) fewer training samples than free parameters. While this typically leads to a very small training error, it is not clear why stochastic gradient descent often does not suffer from *overfitting* and the generalization error remains moderate. Does a certain type of implicit regularization happen? And to what extent does the initialization help us to operate on a well-behaved part of the optimization landscape? There already exist various attempts to demystify these phenomena, but a complete theory still seems out of reach with currently available tools. We refer the interested reader to the overview paper [16], which may serve as a good starting point for a more comprehensive study of recent advances in this direction.

Interpretability & Safety. While the previously mentioned topics are central parts of traditional statistical learning theory, we now take a different and somewhat more applied perspective: Although practitioners may appreciate theoretical guarantees clarifying the training process, they are usually more interested in assessing the quality and reliability of a *ready-to-use* neural network. Indeed, compared to simple learning architectures such as linear models or decision trees, the "semantic" reasoning of highly non-linear and parameter-rich neural network models is often inaccessible. For example, imagine that a neural network is supposed to assist a brain surgery, recommending which parts of the brain to intervene. Clearly, a surgeon would like to understand the reasoning of the network and how certain it is about its decision. Ideally, such an explanation should be indistinguishable from a human explanation. A first step towards understanding the internal operation of networks would be to specify those variables of the input domain – often called *input features* – that contribute the most to a (classification) decision. In the last years, various methods have been developed that aim to assign relevance scores to input features. Most of them are based on the idea of propagating decisions backwards through the network, similarly to the gradient computation via backpropagation; see Figure 2 and Figure 3 for an illustration of some recent findings. Besides the wish for interpretability, it is also of great importance to investigate the robustness of a network against noisy and corrupted inputs. This topic is closely related to so-called *adversarial examples* [17], which have gained increasing attention in the literature; see Figure 4. However, let us emphasize that most of these observations are of purely empirical nature, whereas a theoretic foundation still amounts to very few

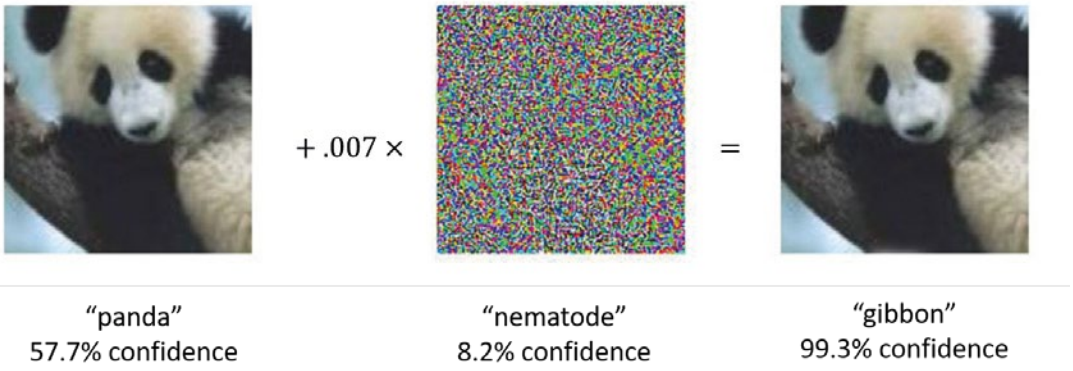

Fig. 4: A typical adversarial example. Trained neural network can be often fooled in such a way that very small corruptions of the input data (adversarial noise) lead to dramatically different outputs; figure taken from [27].

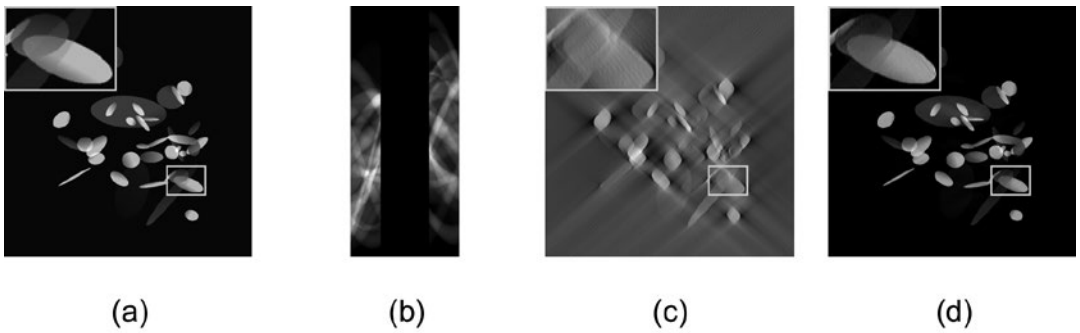

Fig. 5: The inverse problem of limited-angle computed tomography. (a) Ground truth image signal. (b) Noisy Radon transform (sinogram) of the image signal with missing angular measurements. (c) Traditional inversion by filtered back projection. (d) Inversion by "Learning the Invisible" [34].

attempts, e.g., see [18] for recent results on the computational complexity of finding relevance scores.

Neural Networks for Mathematical Problems

The impressive gains of neural networks in machine learning have inspired researchers to seek for applications beyond tasks in computer science. In this course, the benefits of deep learning have recently also led to exciting progress in many areas of applied mathematics. Generally speaking, it has turned out that the performance of traditional methods often can increase substantially when combined with data-driven components that were learned from training samples. In a certain sense, this approach takes the "best out of both worlds", as it allows us to face those parts of mathematical problems in which model-based methods would only yield poor outcomes or are not available at all. However, it is worth pointing out that the usage of (deep-)learning-based methods does not simply come for free: As stressed earlier in this article, the training process requires utmost care and its success usually demands huge training data sets, which are not always available in practice. On the other hand, mastering these difficulties promises outstanding results, oftentimes achieving the current state-of-the-art. Let us now showcase the benefits of neural networks in the context of two classical problem settings in mathematics.

Neural Networks for Inverse Problems. The field of *inverse problems,* in particular, the subdomain of imaging sciences is generally enthusiastic about leveraging the advances of machine learning technologies, in particular, deep learning. Given an inverse problem $f=Kg$ with forward operator K, a very straightforward application of neural networks is to generate training data of the form $(Kg_i, g_i)_{i=1}^m$ and then to let a neural network learn an inversion process for K, e.g., see [28]. However, it turned out that, in order to obtain superior results, a crucial step is to pair the learning procedure with model-based knowledge. In [29] and [30], for instance, a neural network is trained on training data of the form $(FKg_i, g_i)_{i=1}^m$ where F corresponds to a model-based inversion operator, such as *filtered back projection*. A more sophisticated approach to ill-posed inverse problems builds upon the fundamental insight that regularization can be achieved by denoising; this conception is commonly known as *plug-and-play priors* [31], which are also very amenable to neural networks as denoiser [32]. Yet another popular direction is learning iterative schemes, which aim to combine the mathematical structure of variational methods with neural network architectures, e.g., see [33].

Let us now take a closer look at the problem of *limited-angle computed tomography,* which gives a more precise idea of how model- and data-driven methods may be combined in a controllable manner. In simple words, the inverse problem of computed tomography requires an inversion of the *Radon transform,* which computes line integrals of an image signal; see Figure 5: (a) + (b). This measurement process becomes heavily ill-posed if, for instance, only a limited range of angular line orientations is accessible, such as it is the case in electron tomography. In this situation, traditional inversion methods typically suffer from substantial artifacts; see Figure 5: (c). Fortunately, it is theoretically well understood which singularities (in the sense of wavefront sets) can be stably reconstructed and which cannot, allowing us to speak of *visible* and *invisible* singularities, respectively. The recent work [34] leverages this insightful observation: in a first model-based step, a sophisticated sparse regularization is employed, which is based on a (directionally sensitive) shearlet system [35] and thereby enables a separation into the "visible" and unknown "invisible" components; in a second data-driven step, a deep neural network is trained to fill in the missing part of the data, without affecting the already reconstructed visible part. This procedure, referred to as "Learning the Invisible", offers a clear interpretation of the neural network's task in limited-angle computed tomography and shows unprecedented reconstruction quality compared to classical methods; see Figure 5: (d).

Neural Networks for Partial Differential Equations. Interestingly, a first and very common approach for solving *partial differential equations (PDEs)* with neural networks actually dates back to 1998 [36], suggesting a numerical approximation of the solution function u of a PDE $\mathcal{L}(u)=f$ by a neural network; more precisely, one aims at finding

a neural network Φ such that $\mathcal{L}(\Phi)\approx f$, rather than working within a classical function space. A much more recent and important observation was that such a strategy can be even implemented with neural networks whose size does *not* scale exponentially with the underlying dimension, e.g., see [37].

Another very promising research direction concerns *parametric PDEs,* which are encountered in many different areas of science and engineering, such as in complex design problems, optimization tasks, or uncertainty quantification. The key assumption here is that there exists a certain *parametric map* $y\mapsto u_y$, assigning a parameter vector $y\in\mathcal{Y}\subset\mathbb{R}^p$ to the solution function u_y of a parametric PDE of the form $\mathcal{L}(u_y,y)=f_y$. In practice, this typically involves multiple evaluations of the parametric map, which can be a tremendous computational burden, especially when the dimension of the parameter space $\mathcal{Y}\subset\mathbb{R}^p$ is high. Most classical approaches rely on model order reduction methods such as the reduced basis method. With the advent of deep learning, a new exciting line of research has emerged, attempting to mimic the parametric map by a neural network and thereby allowing for a significantly faster computation of the solution for a given parameter vector y; see [38] [39] [40] [41] [42] [43]. Besides very promising numerical results, first theoretical guarantees have been established as well, verifying that the replacement of the parametric map by a neural network can indeed overcome the curse of dimensionality [44] [45].

The Future of Neural Networks in the Sciences

Only on the basis of the aforementioned successes in inverse problems and PDEs, it is conceivable that machine learning techniques, and especially deep learning, will lead to a paradigm shift in the mathematical sciences. Already today, about 80% of the talks at imaging science conferences discuss novel approaches involving neural networks in one or the other way. As pointed out before, particularly good outcomes can be expected from a careful combination of model- and learning-based methods, oftentimes establishing the state-of-the-art within their domain. Nevertheless, several major challenges need to be addressed before such a methodology will enjoy a similar reliance as conventional mathematical approaches, based on modeling, simulation, and optimization. First and foremost, the development of a comprehensive theoretical foundation needs to be accelerated, which is an inevitable step towards “whitening” the black box of deep learning. Not less important and of great practical relevance is a more profound quality assessment of training data, as these form the basis of a resultant neural network and its functionality. In this context, it is also crucial to bear in mind that, in most cases, the training data is strongly tailored to a specific application, and therefore requires a certain expertise and sensible treatment. This fact turns the mathematics of deep learning into a highly interdisciplinary field, inviting many other scientific areas and researchers to contribute as well.

References

[1] B. T. Nugraha, S.-F. Su and Fahmizal, "Towards self-driving car using convolutional neural network and road lane detector," in 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), 2017.

[2] D. Silver, A. Huang, C. J. Maddison and others, "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 526, pp. 484-489, 2016.

[3] D. Silver, T. Hubert, J. Schrittwieser and others, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," Science, vol. 362, no. 6419, pp. 1140-1144, 2018.

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, pp. 115-118, 2017.

[5] "FDA permits marketing of artificial intelligence-based device to detect certain diabetes-related eye problems," U.S. Food and Drug Administration, 11 April 2018. [Online]. Available: https://www.fda.gov/news-events/press-announcements/fda-permits-marketing-artificial-intelligence-based-device-detect-certain-diabetes-related-eye.

[6] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys., vol. 5, no. 4, pp. 115-133, 1943.

[7] A. Rahimi and B. Recht, "Reflections on Random Kitchen Sinks," 5 December 2017. [Online]. Available: http://www.argmin.net/2017/12/05/kitchen-sinks/.

[8] A. Courville, I. Goodfellow and Y. Bengio, Deep Learning, MIT Press, 2016.

[9] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012.

[10] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436-444, 2015.

[11] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," Neural Netw., vol. 2, no. 3, pp. 183-192, 1989.

[12] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," Neural Netw., vol. 2, no. 5, pp. 359-366, 1989.

[13] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Math. Control Signals Syst. , vol. 2, no. 4, pp. 303-314, 1989.

[14] H. Bölcskei, P. Grohs, G. Kutyniok and P. Petersen, "Optimal Approximation with Sparsely Connected Deep Neural Networks," SIAM J. Math. Data Sci., vol. 1, no. 1, pp. 8-45, 2019.

[15] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda and Q. Liao, "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review," Int. J. Autom. Comput., vol. 14, no. 5, pp. 503-519, 2017.

[16] J. Fan, C. Ma and Y. Zhong, A Selective Overview of Deep Learning, arXiv preprint:1904.05526, 2019.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, Intriguing properties of neural networks, arXiv preprint:1312.6199, 2013.

[18] S. Wäldchen, J. Macdonald, S. Hauch and G. Kutyniok, The Computational Complexity of Understanding Network Decisions, arXiv preprint:1905.09163, 2019.

[19] K. Simonyan, A. Vedaldi and A. Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, arXiv preprint:1312.6034, 2013.

[20] G. Montavon, W. Samek and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," Digit. Signal Process., vol. 73, pp. 1-15, 2018.

[21] D. Smilkov, N. Thorat, B. Kim, F. Viégas and M. Wattenberg, SmoothGrad: removing noise by adding noise, arXiv preprint:1706.03825, 2017.

[22] J. Macdonald, S. Wäldchen, S. Hauch and G. Kutyniok, A Rate-Distortion Framework for Explaining Neural Network Decisions, arXiv preprint:1905.11092, 2019.

[23] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek, "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," Plos ONE, vol. 10, no. 7, p. e0130140, 2015.

[24] J. T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, Striving for Simplicity: The All Convolutional Net, arXiv preprint:1412.6806, 2014.

[25] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[26] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30, 2017.

[27] L. Borrows, "Noise warfare," 16 February 2018. [Online]. Available: https://www.seas.harvard.edu/news/2018/02/noise-warfare.

[28] P. Paschalis, N. D. Giokaris, A. Karabarbounis, G. K. Loudos, D. Maintas, C. N. Papanicolas, V. Spanoudaki, C. Tsoumpas and E. Stiliaris, "Tomographic image reconstruction using Artificial Neural Network," Nucl. Instrum. Methods Phys. Res., vol. 527, no. 1-2, pp. 211-215, 2004.

[29] E. Kang, J. Min and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction," Med. Phys., vol. 44, no. 10, pp. 360-375, 2017.

[30] K. H. Jin, M. T. McCann, E. Froustey and M. Unser, "Deep convolutional neural network for inverse problems in imaging," IEEE Trans. Image Process., vol. 26, no. 9, pp. 4509-4522, 2017.

[31] S. V. Venkatakrishnan, C. A. Bouman and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013), 2013.

[32] T. Meinhardt, M. Moeller, C. Hazirbas and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in IEEE International Conference on Computer Vision (ICCV 2017), 2017.

[33] J. Adler and O. Öktem, "Learned primal-dual reconstruction," IEEE Trans. Medical Imaging, vol. 37, no. 6, pp. 1322-1332, 2.

[34] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen and V. Srinivasan, "Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography," Inverse Probl., vol. 35, no. 6, p. 064002, 2019.

[35] G. Kutyniok and D. Labate, Shearlets: Multiscale Analysis for Multivariate Data, Springer, 2012.

[36] I. Lagaris, A. Likas and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," IEEE Trans. Neural Netw., vol. 9, no. 5, pp. 987-1000, 1998.

[37] W. E, J. Han and A. Jentzen, "Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations," Commun. Math. Stat., vol. 5, no. 4, pp. 349-380, 2017.

[38] Y. Khoo, J. Lu and L. Ying, Solving parametric PDE problems with artificial neural networks, arXiv preprint:1707.03351, 2017.

[39] J. S. Hesthaven and S. Ubbiali, "Non-intrusive reduced order modeling of nonlinear problems using neural networks," J. Comput. Phys., vol. 363, pp. 55-78, 2018.

[40] K. Lee and K. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, arXiv preprint:1812.08373, 2019.

[41] Y. Yang and P. Perdikaris, Physics-informed deep generative models, arXiv preprint:1812.03511, 2018.

[42] M. Raissi, "Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations," J. Mach. Learn. Res., vol. 19, pp. 1-24, 2018.

[43] N. D. Santo, S. Deparis and L. Pegolotti, Data driven approximation of parametrized PDEs by Reduced Basis and Neural Networks, arXiv preprint:1904.01514, 2019.

[44] G. Kutyniok, P. Petersen, M. Raslan and R. Schneider, A Theoretical Analysis of Deep Neural Networks and Parametric PDEs, arXiv preprint:1904.00377, 2019.

[45] C. Schwab and J. Zech, "Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ," Anal. Appl., vol. 17, no. 1, pp. 19-55, 2019.

Gitta Kutyniok, Prof. Dr. rer. nat., has an Einstein Chair in the Institute of Mathematics at Technische Universität Berlin and is the head of the Applied Functional Analysis Group. She also has a courtesy appointment in the Department of Electrical Engineering and Computer Science at TU Berlin and holds an Adjunct Professor Position in Machine Learning in the Faculty of Science and Technology, Department of Physics and Technology at the University of Tromsø. She is Executive and Scientific Director of the Berlin International Graduate School in Model and Simulation based Research (BIMoS), spokesperson of the DFG Research Training Group on Differential Equation- and Data-driven Models in Life Sciences and Fluid Dynamics (DAEDALUS), founder and chair of two GAMM Activity Groups (Mathematical Signal- and Image Processing and Computational and Mathematical Methods in Data Science), chair of the MATH+ Activity Group on Mathematics of Data Science, and chair of the SIAM Activity Group on Imaging Science. Her research was recognized by, for example, a Heisenberg-Fellowship and the von Kaven Prize of the DFG, a membership in the Berlin-Brandenburg Academy of Sciences and Humanities (BBAW), and being elected as SIAM Fellow.

Martin Genzel, Dr. rer. nat., is a Postdoc at the Institute of Mathematics at Technische Universität Berlin. There, he is based at the Applied Functional Analysis Group under the direction of Prof. Dr. Gitta Kutyniok, who also supervised his PhD, which he finished in March 2019. His research is focusing on topics at the interface of applied mathematics, signal processing, and machine learning, in particular, inverse problems, compressed sensing, high-dimensional statistics, and deep learning. He is currently a GAMM junior member, elected in 2016.