An Introduction to the Mathematics of Deep Learning

Gitta Kutyniok

Abstract

Despite the outstanding success of deep neural networks in real-world applications, ranging from science to public life, most of the related research is empirically driven and a comprehensive mathematical foundation is still missing. At the same time, these methods have already shown their impressive potential in mathematical research areas such as imaging sciences, inverse problems, or numerical analysis of partial differential equations, sometimes by far outperforming classical mathematical approaches for particular problem classes.

The goal of this paper, which is based on a plenary lecture at the 8th European Congress of Mathematics in 2021, is to first provide an introduction into this new vibrant research area. We will then showcase some recent advances in two directions, namely the development of a mathematical foundation of deep learning and the introduction of novel deep learning-based approaches to solve inverse problems and partial differential equations.

1 Introduction

During the last years, deep neural networks have been key to spectacular successes in diverse applications such as autonomous driving, medical diagnosis, speech recognition, and telecommunication. It is by now evident that deep learning and, in general, artificial intelligence, will change in the future both public life and science in an unprecedented way; and this future has already begun. As an example in the sciences, Google's DeepFold 2 has recently led to a breakthrough in highly accurate prediction of protein structures [20].

A strongly increasing impact on mathematics itself can also be witnessed. The field of inverse problems, predominantly in imaging science, was one of the first areas in mathematics, which embraced these novel methodologies. This area, which focusses on problems such as denoising, inpainting, superresolution, or computed tomography, is particularly accessible to learning methods, since there does not exist a precise model for what an image is. Almost all novel contributions, which improved the state of the art, employ such techniques. This, by now, already led to a change in paradigm in this field. We will discuss further details in Section 4.1.

Besides inverse problems, another large area of mathematical problem settings are partial differential equations. One can in general imagine using learning methods in solvers. It is however not immediately evident what the advantage of such an approach would be. The ability of deep neural networks to beat the curse of dimensionality then led to a change of paradigm in this area as well, and research at the intersection of numerical analysis of partial differential equations and deep learning accelerated since about 2017. Several milestones could already be celebrated as will be presented in Section 4.2.

As bright as the deep learning future appears to be, one has to also be aware of various major obstacles still waiting to be overcome. This was very prominently stated during the plenary talk at the main conference in artificial intelligence and machine learning, namely NIPS (today called NeurIPS) in 2017 on behalf of the Test of Time Award, in which Ali Rahimi from Google claimed that "Machine learning has become a form of alchemy". And, indeed, as we will discuss later, a fundamental understanding of deep learning algorithms is still missing, posing a great—and exciting—challenge to, in particular, mathematics.

This problem becomes even more severe when observing that in addition to a lack of theoretical foundation, causing, for instance, a very time-consuming and delicate training process, deep learning approaches also sometimes fail dramatically. One example of such failures are so-called adversarial examples, when small changes in the data lead to a radically different decision; a well-known problem in this regime is the sensitivity of self-driving

cars to minor adaptions of traffic signs such as the placement of stickers. Another example is fairness, when biased training data causes deep learning approaches to, for instance, reach racist decisions.

Summarizing, there is a tremendous need for mathematics in the area of deep learning. One can identify two different research directions:

- *Mathematics for Deep Learning.* This direction aims for deriving a deep mathematical understanding of deep learning and asks questions such as "How can we make deep learning more robust?"
- *Deep Learning for Mathematics.* This direction focusses on mathematical problem settings such as inverse problems and numerical analysis of partial differential equations with the goal to employ deep learning techniques for superior solvers.

In this article we will touch upon both research directions, showcasing some novel results and pointing out key future challenges for mathematics. In Section 2, we will first provide an introduction into deep learning from a mathematics viewpoint. We will then delve deeper into the first direction, namely *Mathematics for Deep Learning* and discuss the subarea of expressivity in more detail (Section 3). This will be followed in Section 4 by highlighting examples of the second direction, namely *Deep Learning for Mathematics.* Finally, Section 5 is devoted to future perspectives for mathematics.

2 Deep Neural Networks

In 1943, McCulloch and Pitts had the vision to introduce artificial intelligence to the world [28]. At that time, their idea was to develop an algorithmic approach to learning by mimicking the functionality of the human brain. Due to the structure of the brain being composed of neurons with numerous interconnections, they introduced so-called artificial neurons as building blocks. The structure of a neuron in the human brain, in its most simple form, consists of dendrites through which signals are transmitted to its soma, while being scaled/amplified due to the structural properties of the respective dendrites. In the soma of the neuron, those incoming signals are accumulated, and a decision is reached whether to fire to other neurons or not, and also with which strength.

A mathematical definition of an artificial neuron is consequently defined as follows. In the sequel, we will build a neural network from such components with the weights and biases being the free parameters, which need to be trained.

Definition 2.1. An *artificial neuron* with *weights* $w_1, ..., w_n \in \mathbb{R}$, *bias* $b \in \mathbb{R}$ and *activation function* $\rho : \mathbb{R} \to \mathbb{R}$ is defined as the function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(x_1, ..., x_n) = \rho\left(\sum_{i=1}^{n} x_i w_i + b\right) = \rho(\langle x, w\rangle + b),$$

where $w = (w_1, ..., w_n)$ and $x = (x_1, ..., x_n)$.

Let us now take a look at some examples of activation functions.

Example 2.2. (1) Heaviside function $\rho(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leq 0. \end{cases}$

(2) Sigmoid function $\rho(x) = \frac{1}{1+e^{-x}}$.

(3) Rectifiable Linear Unit (ReLU) $\rho(x) = \max\{0, x\}$.

The most basic activation function is certainly the Heaviside function, leading to a yes/no decision. The sigmoid function is a smooth alternative. But the by far most extensively used activation function in basically all applications is the ReLU due to its simple piecewise linear structure, which is advantageous in the training process, and still allows superior performance.

2.1 The Mathematical Definition

An (artificial feed-forward) neural network is then build by concatenating artificial neurons to compositions of affine linear maps and activation functions. This leads to the following definition.

Definition 2.3. Let $d \in \mathbb{N}$ be the dimension of the input layer, L the number of layers, $N_0 := d$, N_ℓ, $\ell = 1, \dots, L$, be the dimensions of the hidden and last layer, $\rho : \mathbb{R} \to \mathbb{R}$ a (non-linear) activation function, and, for $\ell = 1, \dots, L$, let T_ℓ be the affine-linear functions

$$T_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}, \qquad T_\ell x = W^{(\ell)} x + b^{(\ell)},$$

with $W^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ being the weight matrices and $b^{(\ell)} \in \mathbb{R}^{N_\ell}$ the bias vectors of the ℓth layer. Then $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ given by

$$\Phi(x) = T_L \rho(T_{L-1} \rho(\dots \rho(T_1(x)) \dots)), \qquad x \in \mathbb{R}^d,$$

is called *(deep) neural network*.

We would like to stress that in many papers a distinction is made between a neural network and its realization, namely the function it realizes. The reason for this is that different architectures can lead to the same function. For this article, we will however avoid such technical delicacies.

2.2 Key Research Directions

Aiming to identify the key mathematical research directions in deep learning, let us take a high-level view of the typical application of a deep neural network; exemplarily we choose classification. One then proceeds in the four—very coarsely explained—steps:

(1) We assume that we are given samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \dots, K\}$, where $\mathcal{M}$ might be a lower-dimensional manifold of $\mathbb{R}^d$. This is a customarily assumed setting in image classification. We then split this set into a training data set $(x_i, f(x_i))_{i=1}^{\tilde{m}}$, say, and a test data set $(x_i, f(x_i))_{i=\tilde{m}+1}^m$, say. The training data set is—as the name indicates—used for training, and the test data set for testing the performance of the trained network. Notice that the test data set stays hidden during the training process.

(2) Then an architecture of a deep neural network needs to be selected, i.e., a choice of L, $(N_\ell)_{\ell=1}^L$, and ρ. Sometimes selected entries of the weight matrices $(W^{(\ell)})_{\ell=1}^L$ are already set to zero at this point, if one does not intend to train a fully connected neural network.

(3) Next, the affine-linear functions $(T_\ell)_{\ell=1}^L = (W^{(\ell)} \cdot + b^{(\ell)})_{\ell=1}^L$ are learnt by solving the optimization problem given by

$$\min_{(W^{(\ell)}, b^{(\ell)})_\ell} \sum_{i=1}^{\tilde{m}} \mathcal{L}(\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x_i), f(x_i)) + \lambda \mathcal{R}((W^{(\ell)}, b^{(\ell)})_\ell),$$

where $\mathcal{L}$ is a loss function to determine a measure of closeness between the network evaluated in the training samples and the (known) function values $f(x_i)$ and $\mathcal{R}$ is a regularization term to impose additional constraints on the weight matrices and bias vectors. The optimization problem is typically solved by stochastic gradient descent, yielding a network $\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell} : \mathbb{R}^d \to \mathbb{R}^{N_L}$, where

$$\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell}(x) = T_L \rho(T_{L-1} \rho(\dots \rho(T_1(x)) \dots)).$$

(4) Finally, one employs the test data set to analyze whether

$$\Phi_{(W^{(\ell)}, b^{(\ell)})_\ell} \approx f,$$

i.e., whether and to which extent the training process was successful.

It is in fact very surprising that this procedure works this well these days, which has two main reasons: First, the drastic improvement of computing power allows the training of networks with hundreds of layers in the sense of *deep* neural networks. And, second, we are living in the age of data, hence vast amounts of training data is available. This being the empirical explanation, a profound mathematical explanation why, for instance, *deep* networks are superior to shallow ones or why the complex training data does not lead to the phenomenon of *overfitting* is to a large extent still missing.

2.2.1 Mathematics for Deep Learning

Based on these considerations, we can now formulate the four key mathematical research directions, first for *Mathematics for Deep Learning*. We will each time also mention the main mathematical fields involved, thereby showing that almost each area of mathematics is touched and required.

- *Expressivity.* This direction aims to understand whether and to which extent aspects of a neural network architecture affect the performance of deep learning. Typically methods from applied harmonic analysis and approximation theory are used.

- *Learning.* The goal here is to analyze the training procedure with a key question being why the typically applied algorithm of stochastic gradient descent does converge to suitable local minima even though the problem itself is highly non-convex. This direction relies on techniques from areas such as algebraic/differential geometry, optimal control, and optimization.

- *Generalization.* This research direction is the least explored and maybe also the most difficult, sometimes called the "holy grail" of deep learning. It targets the out-of-sample error, and asks questions such as "Why is depth beneficial" or "Why does high overparametrization not lead to overfitting?". Required methods belong in particular to the following areas: Learning theory, probability theory, and statistics.

Notice that these three research directions are precisely related to the three components of the error of a statistical learning problem (cf. [4, (1.4) and Fig. 1.2]), namely the approximation error from the hypothesis class, the optimization error from the algorithm itself, and the out-of-sample error.

Besides these more classical problem complexes, new directions have evolved. One of the most exciting directions might be the following, which until now lacks almost entirely a mathematical foundation.

- *Explainability.* Given a trained neural network, this area aims to analyze why certain decisions were reached, and which components of the input data were crucial for those. The range of required approaches is quite broad, including areas such as information theory or uncertainty quantification.

In practice, this direction is invaluable, since one often encounters the situation that a neural network is given and decisions have to be explained, for instance, to a customer. In the imaging situation, typical explanations are relevance maps assigning each pixel a relevance score for the decision such as Layerwise-Relevance Propagation (LRP) [3] or Rate-Distortion Explanation (RDE) [15]. For an example of such an explanation, we refer to Figure 1.

Figure 1: Illustration of an explanation for the classification as a "black swan" using RDE.

However, from a mathematical standpoint, one truly aims for a mathematical definition of the term "relevance" and an according theory of optimal relevance maps. Ideally, one would also like to have explanations beyond the

pixel-based setting and for more challenging modalities. For a survey of some recent work in this direction, we refer to [21].

2.2.2 Deep Learning for Mathematics

As said before, the second main research thread is *Deep Learning for Mathematics* in the sense of deep learning for mathematical problem settings. The two key research subfields are as follows.

- *Inverse Problems.* The main goal is to improve classical model-based approaches by deep learning techniques. Since it is often highly beneficial to not entirely neglect domain knowledge such as the physics of the problem, one crucial question is how to optimally combine deep learning with model-based approaches. This direction relies on tools from imaging science, inverse problems, and microlocal analysis, to name a few.
- *Partial Differential Equations.* Research in this area targets foremost the question of how and to which extent deep neural networks are able to beat the curse of dimensionality. This direction obviously requires methods from areas such as numerical mathematics and partial differential equations.

3 Mathematics for Deep Learning

Deep learning-based methodologies for inverse problems and partial differential equations exploit deep neural networks as approximators. Thus, the first question to ask is whether deep neural networks are at least as good as all previous mathematical methods. This question belongs in the realm of the previously introduced area of expressivity, which will be the focus of this section, aiming to provide a (partial) answer.

3.1 Revisiting Classical Approximation Theory

We start by revisiting classical approximation theory, and, in the sequel, analyze whether deep neural networks have at least similar approximation properties as classical methods.

In a nutshell, function approximation has the following goal. Given a class $\mathcal{C} \subseteq L^2(\mathbb{R}^d)$ of interest — for later use it is sufficient for us to consider $L^2(\mathbb{R}^d)$ — and a representation system $(\varphi_i)_{i\in I} \subseteq L^2(\mathbb{R}^d)$, which can be an orthonormal basis or, more generally, a frame, one aims to measure the suitability of $(\varphi_i)_{i\in I}$ for uniformly approximating functions from $\mathcal{C}$. For a budget N, the approximating function has then typically the form of a linear combination of N terms of the representation system. This leads to the following definition.

Definition 3.1. The *error of best N-term approximation* of some $f \in \mathcal{C}$ is given by

$$\sigma_N(f) := \inf_{I_N \subset I, \#I_N = N, (c_i)_{i\in I_N}} \|f - \sum_{i\in I_N} c_i \varphi_i\|_2.$$

The largest $\gamma > 0$ such that

$$\sup_{f\in\mathcal{C}} \sigma_N(f) = O(N^{-\gamma}) \qquad \text{as } N \to \infty$$

determines the *optimal (sparse) approximation rate* of $\mathcal{C}$ by $(\varphi_i)_{i\in I}$.

A closer look reveals that this viewpoint relates approximation accuracy to the complexity of the approximating system in terms of sparsity.

Also for later use, we will now introduce one example of a class $\mathcal{C}$ and a representation system $(\varphi_i)_{i\in I}$ along with an analysis of its optimal (sparse) approximation rate. The model class we will consider, called *cartoon-like functions* (see Figure 2), was first introduced in imaging science [10], since the predominant features of images are edge structures. Such anisotropic features also occur in other settings such as the solution of transport dominated equations, leading to a model class with much larger applicability.

Figure 2: Illustration of a cartoon-like function

Definition 3.2. The set of *cartoon-like functions* $\mathcal{E}^2(\mathbb{R}^2)$ is defined by

$$\mathcal{E}^2(\mathbb{R}^2) = \{f \in L^2(\mathbb{R}^2) : f = f_0 + f_1 \cdot \chi_B\},$$

where $\emptyset \neq B \subset [0,1]^2$ is simply connected with a C^2-curve with bounded curvature as its boundary, and $f_i \in C^2(\mathbb{R}^2)$ with supp $f_i \subseteq [0,1]^2$ and $\|f_i\|_{C^2} \leq 1$, $i = 0, 1$.

A lower bound for any optimal (sparse) approximation rate was derived in the same article, i.e., [10]. We would like to remark that the purpose of the technical requirement of "polynomial depth search" in the following theorem is to avoid degenerate cases of representation systems.

Theorem 3.3. *Allowing only polynomial depth search, we have the following optimal behavior for* $f \in \mathcal{E}^2(\mathbb{R}^2)$*:*

$$\sigma_N(f) \asymp N^{-1} \quad \text{as } N \to \infty.$$

The well-known wavelet systems [9] do only provide a suboptimal rate of $N^{-\frac{1}{2}}$ due to the fact that they are isotropic multiscale systems in the sense of scaling in both directions at a similar rate (cf. Figure 3).

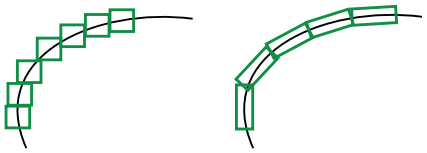

Figure 3: Schematic illustration of wavelet and shearlet approximation.

Various systems were suggested to provide optimal (sparse) approximations for cartoon-like functions. The first successful systems were curvelets [7], which however did not allow faithful implementations. This could be achieved by so-called shearlets, which were introduced in [26], see also the survey article [22]. For an illustration of the benefit of anisotropic scaling, we refer to Figure 3.

Shearlet systems are associated with three parameters: scale j, position m, and orientation k. For the precise definition, let A_{2^j} and $\tilde{A}_{2^j}$, $j \in \mathbb{Z}$, denote the parabolic scaling matrices given by

$$A_{2^j} := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}$$

and $\tilde{A}_{2^j} := \text{diag}(2^{j/2}, 2^j)$, and let S_k, $k \in \mathbb{Z}$, be the shearing matrix given by

$$S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}.$$

(Cone-adapted) discrete shearlet systems can then be defined as follows, cf. [24].

Definition 3.4. The *(cone-adapted) discrete shearlet system* $\mathcal{SH}(\phi, \psi, \tilde{\psi})$ generated by $\phi \in L^2(\mathbb{R}^2)$ and $\psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ is the union of

$$\{\phi(\cdot - m) : m \in \mathbb{Z}^2\},$$

$$\{2^{3j/4}\psi(S_k A_{2^j}\,\cdot\,-m) : j \geq 0, |k| \leq \lceil 2^{j/2}\rceil, m \in \mathbb{Z}^2\},$$
$$\{2^{3j/4}\tilde{\psi}(S_k^T \tilde{A}_{2^j}\,\cdot\,-m) : j \geq 0, |k| \leq \lceil 2^{j/2}\rceil, m \in \mathbb{Z}^2\}.$$

We denote the associated *shearlet transform* by

$$SH(f) := \left(\langle f, g\rangle\right)_{g\in\mathcal{SH}(\phi,\psi,\tilde{\psi})}, \quad f \in L^2(\mathbb{R}^2).$$

This system indeed satisfies the optimal (sparse) approximation rate for cartoon-like functions up to a log-factor, which is often regarded as negligible. The following statement is taken from [24], where also the precise hypotheses can be found.

Theorem 3.5. *Let $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ be compactly supported, and let $\hat{\psi}$, $\hat{\tilde{\psi}}$ satisfy certain decay condition. Then $\mathcal{SH}(\phi, \psi, \tilde{\psi})$ provides an* optimally sparse approximation *of $f \in \mathcal{E}^2(\mathbb{R}^2)$, i.e.,*

$$\sigma_N(f) \lesssim N^{-1}(\log N)^{\frac{3}{2}} \quad \text{as } N \to \infty.$$

Concluding our example for Definition 3.1, shearlet systems provide an (almost) optimal (sparse) approximation rate of N^{-1} for the class $\mathcal{C}$ of cartoon-like functions. For the interested reader, a faithful implementation of the shearlet transform as a 2D&3D (parallelized) fast shearlet transform can be found in `www.ShearLab.org`.

3.2 Universality of Deep Neural Networks

Analyzing approximation problems for deep neural networks immediately bears the question of how to replace the notion of complexity of the approximating term, which was before measured in terms of sparsity. A typical approach for networks is a complexity measure in terms of memory requirements. Recall that the $\|\cdot\|_0$-"norm" counts the number of non-zero entries.

Definition 3.6. Retaining the same notation for deep neural networks as in Definition 2.3, the *complexity* $C(\Phi)$ of a deep neural network Φ is defined by

$$C(\Phi) := \sum_{\ell=1}^{L} \left(\|W^{(\ell)}\|_0 + \|b^{(\ell)}\|_0\right).$$

We will also in the sequel use the notion $\mathcal{NN}_{L,C,d,\rho}$ for the class of deep neural networks with no more than L layers, complexity of at most C, input dimension d, and activation function ρ. If no bound is given, we indicate this by writing ∞.

Thus, the key challenge is now to relate approximation accuracy to the complexity of the approximating network in terms of memory efficiency. A very classical result – and maybe the first main expressivity result from the time of the "first wave" of neural networks – is the Universal Approximation Theorem [8, 17], which states that each continuous function on a compact domain can be approximated up to an arbitrary accuracy by a shallow neural network.

Theorem 3.7. *Let $d \in \mathbb{N}$, $K \subset \mathbb{R}^d$ compact, $f : K \to \mathbb{R}$ continuous, $\rho : \mathbb{R} \to \mathbb{R}$ continuous and not a polynomial. Then, for each $\epsilon > 0$, there exist $N \in \mathbb{N}$, $a_k, b_k \in \mathbb{R}, w_k \in \mathbb{R}^d$, $1 \leq k \leq N$, such that*

$$\|f - \sum_{k=1}^{N} a_k \rho(\langle w_k, \cdot\rangle - b_k)\|_\infty \leq \epsilon.$$

While this is certainly an interesting result, it is not satisfactory in terms of complexity, since this can be arbitrary large.

Aiming to derive an optimality result, we require a lower bound as a benchmark. One example of such a statement was proven in [5] in terms of a so-called optimal exponent $\gamma^*(\mathcal{C})$ from information theory to measure the complexity of $\mathcal{C} \subset L^2(\mathbb{R}^d)$. We should stress that only the essence of this result is stated without all details.

Theorem 3.8. *Let $d \in \mathbb{N}$, $\rho : \mathbb{R} \to \mathbb{R}$, and let $\mathcal{C} \subset L^2(\mathbb{R}^d)$. Further, let*

$$\mathbf{Learn} : (0,1) \times \mathcal{C} \to \mathcal{NN}_{\infty,\infty,d,\rho}$$

satisfy that, for each $f \in \mathcal{C}$ and $0 < \epsilon < 1$,

$$\sup_{f \in \mathcal{C}} \|f - \mathbf{Learn}(\epsilon, f)\|_2 \leq \epsilon.$$

Then, for all $\gamma < \gamma^*(\mathcal{C})$,

$$\epsilon^\gamma \sup_{f \in \mathcal{C}} C(\mathbf{Learn}(\epsilon, f)) \to \infty, \qquad \text{as } \epsilon \to 0.$$

This now provides a conceptual lower bound independent of the learning algorithm. It in fact allows not only to construct deep neural networks, which are memory-optimal, but also to answer the question with which we started, namely whether deep neural networks are at least as good as all previous mathematical methods. We will affirm this for approximations by affine systems such as wavelets and shearlets.

One can now proceed as follows. Assume that we are given a specific function class such as cartoon-like images, and an associated representation system with an optimal approximation rate such as shearlets. Mimicking classical approximation theory—more specifically best N-term approximations—by neural networks leads to such memory-optimal neural networks, which at the same time perform at least as good as the associated representation system from an approximation standpoint.

One example of a resulting theorem is taken from [5]. Notice that this is in fact the optimal approximation rate (up to some ϵ), implying that the bound in Theorem 3.8 is sharp.

Theorem 3.9. *Let ρ be a suitably chosen activation function, and let $\epsilon > 0$. Then, for all $f \in \mathcal{E}^2(\mathbb{R}^2)$ and $N \in \mathbb{N}$, there exists $\Phi \in \mathcal{NN}_{3,O(N),2,\rho}$ with*

$$\|f - \Phi\|_2 \lesssim N^{-1+\epsilon} \to 0 \quad \text{as } N \to \infty.$$

Thus, one can conclude that deep neural networks achieve optimal approximation properties of all affine systems combined. Intriguingly, training the network architecture of the proof, the neural network does even learn approximations of classical affine systems such as shearlets; for more details see [5].

4 Deep Learning for Mathematics

Having established that deep neural networks are at least as good as various classical approximation methods, we will continue our journey in the deep learning world and next ask whether deep learning methods are even better than classical approaches. For this, we will now enter the area of *Deep Learning for Mathematics* and turn towards the setting of inverse problems.

4.1 Inverse Problems meet Deep Learning

We start by recalling a general classical approach to solve inverse problems. We will later discuss how to best combine it with deep learning in specific problem settings in the sense of taking the best out of the model- and data-world.

Assume that we are given an (ill-posed) inverse problem

$$Kf = g, \qquad \text{where} \quad K : X \to Y,$$

where X and Y are Hilbert spaces, say. In its most classical form in imaging science, K could be an operator which adds noise to an image, leading to a denoising problem. *Sparse regularization* is a conceptually general

approach for recovering f from knowledge of g and K, see also [18]. It computes an approximate solution $f^\alpha \in X$, $\alpha > 0$, by solving

$$f^\alpha := \operatorname*{argmin}_f \Big[\underbrace{\|Kf - g\|^2}_{\text{Data fidelity term}} + \alpha \cdot \underbrace{\|(\langle f, \varphi_i \rangle)_{i \in I}\|_1}_{\text{Penalty term}} \Big],$$

where $(\varphi_i)_{i \in I}$ is a suitably selected — in the sense of providing sparse approximations of f — orthonormal basis or frame for X.

One class of approaches for combining deep learning with solvers such as sparse regularization are supervised approaches, which in their most direct form first apply the solver followed by the neural network [19]. A bit more sophisticated are approaches which replace certain procedures in the solver – such as a denoising part – by a deep neural network in the sense of plug-and-play [31] or using a specifically trained network [1]. Semi-supervised approaches aim to encode the regularization as a neural network, see, e.g., [27], whereas deep image prior [32] are one example of what one might coin unsupervised approaches.

We will now focus on one specific inverse problem from imaging science and discuss one exemplary approach in more detail. This approach follows the philosophy to apply the model-based solver as far as it is reliable and only complement it by a deep neural network where necessary. The problem we aim to study is the inverse problem of (limited angle-) computed tomography.

A CT scanner samples the *Radon transform*, which is defined by

$$\mathcal{R}f(s, \theta) = \int_{-\infty}^{\infty} f\big(s\omega(\theta) + t\omega(\theta)^\perp\big) dt, \quad \text{for } (s, \theta) \in \mathbb{R} \times (0, \pi).$$

Here $\omega(\theta) := (\cos\theta, \sin\theta)$ is the unitary vector with orientation described by the angle θ with respect to the x_1-axis and $\omega(\theta)^\perp := (-\sin\theta, \cos\theta)$.

The problem of inverting the Radon transform becomes even harder, if $\mathcal{R}f(s, \cdot)$ is only sampled on a proper subset $[-\phi, \phi]$ of $(0, \pi)$, which is the case in, for instance, electron tomography. In the sequel, we will refer to the respective Radon transform by $\mathcal{R}_\phi$. Classical solvers fail in this case due to the fact that a large connected region of the measurements is missing, while also being too complex for accurate modeling.

The key problem can in fact be regarded as recovering parts of the wavefront set of the original image, where – coarsely speaking – a wavefront set is the set of singularities of a distribution together with their directions; for a precise definition we refer to [16]. Since shearlets resolve the wavefront set [23], the following approach was suggested in [6], following the previously described philosophy:

- Step 1: *Reconstruct the visible.*
  Compute
  $$f^* := \operatorname*{argmin}_{f \geq 0} \| \mathcal{R}_\phi f - g\|_2^2 + \|SH(f)\|_{1,w}.$$
  We then split the set of parameters (j, m, k) of shearlets into a visible set $\mathcal{I}_{\texttt{vis}}$ and an invisible set $\mathcal{I}_{\texttt{inv}}$ related to whether they are associated with shearlets within a range of acquired data or not, leading to:
  - ◇ For $(j, m, k) \in \mathcal{I}_{\texttt{inv}}$: $SH(f^*)_{(j,m,k)} \approx 0$.
  - ◇ For $(j, m, k) \in \mathcal{I}_{\texttt{vis}}$: $SH(f^*)_{(j,m,k)}$ is reliable and near perfect.
- Step 2: *Learn the invisible.*
  Train a neural network (U-net) Φ to compute
  $$\Phi : SH(f^*)_{\mathcal{I}_{\texttt{vis}}} \mapsto F,$$
  where F is an approximation of $SH(f_{\texttt{gt}})_{\mathcal{I}_{\texttt{inv}}}$ and $f_{\texttt{gt}}$ the ground truth image.
- Step 3: *Combine.*
  Finally, compute
  $$f_{\texttt{LtI}} = SH^T \left(SH(f^*)_{\mathcal{I}_{\texttt{vis}}} + F\right).$$

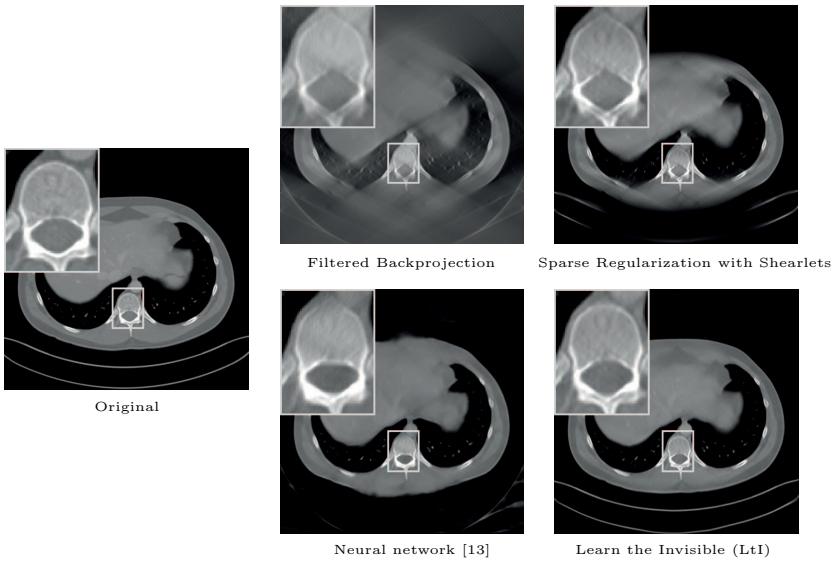

Figure 4: Illustration of the superiority of combined model-deep learning approaches.

The numerical experiments in Figure 4 indicate the superiority of deep learning approaches in general and even more a careful combination of classical solvers with deep neural networks to pure model-based approaches.

This answers the question whether deep neural networks can perform even better than classical methods to the affirmative. We include with Figure 5 one additional example, which follows the same philosophy for the edge detection problem [2]. Without going into the details, this approach first uses shearlets as a coarse edge detector, followed by a deep neural network.

4.2 Deep Learning-Based Solvers for Partial Differential Equations

Finally, we will provide a glimpse into the effectiveness of deep neural networks for solving partial differential equations, and provide an answer to the question of why one should use deep learning for solving partial differential equations at all.

Given a partial differential equations $\mathcal{L}(u) = f$, a common approach to solve this equation using a neural network Φ is to approximate the solution u by Φ, i.e., to train Φ such that

$$\mathcal{L}(\Phi) \approx f.$$

This requires to incorporate the partial differential equation into the loss function. Some of the key approaches in this realm are the Deep Ritz Method [11], so-called Physics Informed Neural Networks [29], or using a backwards stochastic partial differential equation reformulation [14].

We will now focus on a more general setting, namely parametric partial differential equations, which in fact arise in basically any branch of science and engineering such as in complex design problems or uncertainty quantification tasks. Given a parametric partial differential equation $\mathcal{L}(u_y, y) = f_y$ with y being a parameter from a parameter space $\mathcal{Y} \subseteq \mathbb{R}^p$ and u_y the associated solution in a Hilbert space $\mathcal{H}$. Since in applications one typically faces a multi-query situation, the so-called *parametric map*, given by

$$\mathcal{Y} \ni y \;\mapsto\; u_y \in \mathcal{H} \quad \text{such that} \quad \mathcal{L}(u_y, y) = f_y,$$

needs to be solved several times. If p is very large, the curse of dimensionality could lead to an exponential computational cost.

It seems natural to ask whether deep neural networks can be of benefit in this situation in the sense of whether a network can approximate the parametric map leading to a flexible, universal approach which is hopefully not

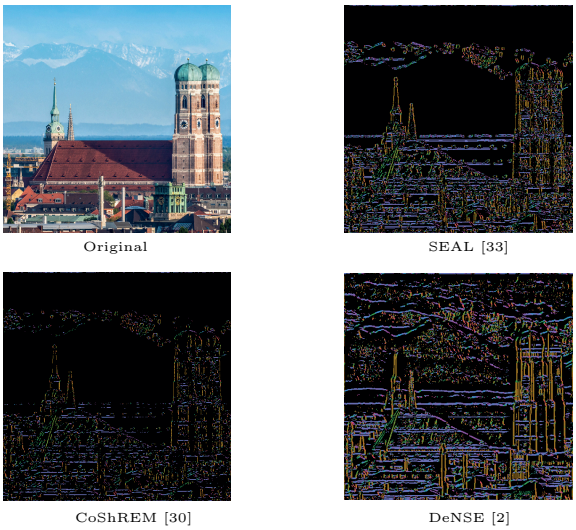

Figure 5: Illustration of another combined model-deep learning approach ([2]) in relation to model-based methods ([33, 30]).

affected by the curse of dimensionality. For this, we first need to bring the problem into a finite-dimensional domain, which is done by a high-fidelity discretization, leading to the problem

$$\mathbb{R}^p \supseteq \mathcal{Y} \ni y \;\mapsto\; \mathbf{u}_y^{\mathrm{h}} \in \mathbb{R}^D \quad \text{such that} \quad b_y\left(u_y^h, v\right) = f_y(v) \;\text{ for all } v$$

with $b_y\left(u_y^h, v\right) = f_y(v)$ being the associated variational form and $\mathbf{u}_y^{\mathrm{h}}$ being the coefficient vector of u_y^h with respect to a suitable basis. We can now ask the following questions.

- Given $\epsilon > 0$, does there exist a neural network Φ such that

$$\|\Phi(y) - \mathbf{u}_y^{\mathrm{h}}\| \leq \epsilon \qquad \text{for all } y \in \mathcal{Y},$$

  and how does the complexity of Φ depend on p and D?

- How do neural networks perform numerically on this task?

The first question falls in the category of expressivity and would need to be complemented by an analysis of the learning procedure as well as the generalization error, as discussed in Section 2.2.1. Mathematical answers to those two questions are however at this point still out of reach, leaving only numerical experiments as an alternative.

The first question was indeed solved by explicitly constructing an associated deep neural network, while careful monitoring its complexity. We state the result from [25] in a high level form.

Theorem 4.1. *There exists a neural network Φ which approximates the parametric map, i.e.,*

$$\|\Phi(y) - \mathbf{u}_y^{\mathrm{h}}\| \leq \epsilon \qquad \textit{for all } y \in \mathcal{Y},$$

and the dependence of $C(\Phi)$ on p and D can be (polynomially) controlled.

With an extensive set-up of numerical experiments such as fixing a specific neural network architecture and the training procedure, it could then be shown in [12] that the numerical performance of deep neural networks for this task does also not suffer from the curse of dimensionality.

5 Conclusions

Deep learning shows impressive performance in real-world applications. However, a theoretical foundation is largely missing. Developing such a foundation requires various areas of mathematics as well as the development of new mathematics. The two main research areas are *Mathematics for Deep Learning* with its subfields expressivity, learning, generalization, and explainability, and *Deep Learning for Mathematics* aiming to apply deep learning to solve inverse problems and partial differential equations.

Let us conclude with seven mathematical key problems of deep learning as they were stated in [4]:

(1) What is the role of depth?

(2) Which aspects of a neural network architecture affect the performance of deep learning?

(3) Why does stochastic gradient descent converge to good local minima despite the non-convexity of the problem?

(4) Why do large neural networks not overfit?

(5) Why do neural networks perform well in very high-dimensional environments?

(6) Which features of data are learned by deep architectures?

(7) Are neural networks capable of replacing highly specialized numerical algorithms in natural sciences?

It is thus fair to say that there are exciting future perspectives for mathematics.

Acknowledgments. This research was partly supported by the Bavarian High-Tech Agenda, DFG-SFB/TR 109 Grant C09, DFG-SPP 1798 Grant KU 1446/21-2, DFG-SPP 2298 Grant KU 1446/32-1, and NSF-Simons Research Collaboration on the Mathematical and Scientific Foundations of Deep Learning (MoDL) (NSF DMS 2031985). The author would like to thank Hector Andrade-Loarca, Ron Levie, and Philipp Scholl for their helpful feedback on an early version of this article.

References

[1] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. Inverse Probl. **33**, 124007 (2017).

[2] H. Andrade-Loarca, G. Kutyniok, O. Öktem, and P. Petersen. Extraction of digital wavefront sets using applied harmonic analysis and deep neural networks. SIAM J. Imaging Sci. **12**, 1936–1966 (2019).

[3] 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-wiserelevance propagation. PLoS ONE **10**, e0130140 (2015).

[4] J. Berner, P. Grohs, G. Kutyniok, and P. Petersen. The Modern Mathematics of Deep Learning. In: Mathematical Aspects of Deep Learning, Cambridge University Press, to appear.

[5] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen. Optimal Approximation with Sparsely Connected Deep Neural Networks. SIAM J. Math. Data Sci. **1**, 8–45 (2019).

[6] 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. **35**, 064002 (2019).

[7] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. Comm. Pure Appl. Math. **57**, 219–266 (2004).

[8] G. Cybenko. Approximation by superpositions of a sigmoidal function. Math. Control Signal **2**, 303–314 (1989).

[9] I. Daubechies. Ten lectures on wavelets, volume 61 of CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1992).

[10] D. Donoho. Sparse components of images and optimal atomic decompositions. Constr. Approx. **17**, 353–382 (2001).

[11] W. E and B. Yu. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. Commun. Math. Stat. **6**, 1–12 (2018).

[12] M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks. J. Sci. Comput. **88**, Article number: 22 (2021).

[13] J. Gu and J. C. Ye. Multi-scale wavelet domain residual learning for limited-angle CT reconstruction. Procs Fully3D, 443–447 (2017).

[14] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. USA **115**, 8505–8510 (2018).

[15] C. Heiß, R. Levie, C. Resnick, G. Kutyniok, and J. Bruna. In-Distribution Interpretability for Challenging Modalities. ICML, Interpretability for Scientific Discovery (2020).

[16] L. Hörmander, The analysis of linear partial differential operators. I. Distribution theory and Fourier analysis. Springer-Verlag, Berlin, 2003.

[17] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989).

[18] B. Jin, P. Maaß, and O. Scherzer. Sparsity regularization in inverse problems, Inverse Probl. **33**, 060301 (2017).

[19] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging, IEEE Trans. Image Process. **26**, 4509–4522 (2017).

[20] J. Jumper, R. Evans, A. Pritzel, et al. Highly accurate protein structure prediction with AlphaFold. Nature **596**, 583–589 (2021).

[21] S. Kolek, D. A. Nguyen, R. Levie, J. Bruna, and G. Kutyniok. A rate-distortion framework forexplaining black-box model decisions. In: Springer LNAI Volume: xxAI - Beyond Explainable AI, to appear.

[22] G. Kutyniok and D. Labate. Introduction to Shearlets. In: Shearlets: Multiscale Analysis for Multivariate Data, 1–38, Birkhäuser Boston (2012).

[23] G. Kutyniok and D. Labate. Resolution of the wavefront set using continuous shearlets. Trans. Amer. Math. Soc. **361**, 2719–2754 (2009).

[24] G. Kutyniok and W.-Q Lim. Compactly supported shearlets are optimally sparse. J. Approx. Theory **163**, 1564–1589 (2011).

[25] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider. A Theoretical Analysis of DeepNeural Networks and Parametric PDEs. Constr. Approx. **55**, 73–125 (2022).

[26] D. Labate, W.-Q Lim, G. Kutyniok, and G. Weiss. Sparse multidimensional representation using shearlets. Wavelets XI (San Diego, CA, 2005), 254–262, SPIE Proc. 5914, SPIE, Bellingham, WA (2005).

[27] S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems. NIPS, 8507–8516 (2018).

[28] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. B. Math. Biophys. **5**, 115–133 (1943).

[29] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019).

[30] R. Reisenhofer, J. Kiefer, and E. J. King. Shearlet-based detection of flame fronts. Exp. Fluids **57**, 11 (2015).

[31] Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (red), SIAM J. Imaging Sci. **10**, 1804–1844 (2017).

[32] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. CVPR, 9446–9454 (2018).

[33] Z. Yu, W. Liu, Y. Zou, C. Feng, S. Ramalingam, B. V. Kumar, and J. Kautz. Simultaneous edge alignment and learning. ECCV (2018).