Math. Stat. Learn. 4 (2021), 1–86
DOI 10.4171/MSL/28

Deep learning architectures for nonlinear operator functions and nonlinear inverse problems

Maarten V. de Hoop, Matti Lassas, and Christopher A. Wong

Abstract. We develop a theoretical analysis for special neural network architectures, termed *operator recurrent neural networks*, for approximating nonlinear functions whose inputs are linear operators. Such functions commonly arise in solution algorithms for inverse boundary value problems. Traditional neural networks treat input data as vectors, and thus they do not effectively capture the multiplicative structure associated with the linear operators that correspond to the data in such inverse problems. We therefore introduce a new family that resembles a standard neural network architecture, but where the input data acts multiplicatively on vectors. Motivated by compact operators appearing in boundary control and the analysis of inverse boundary value problems for the wave equation, we promote structure and sparsity in selected weight matrices in the network. After describing this architecture, we study its representation properties as well as its approximation properties. We furthermore show that an explicit regularization can be introduced that can be derived from the mathematical analysis of the mentioned inverse problems, and which leads to certain guarantees on the generalization properties. We observe that the sparsity of the weight matrices improves the generalization estimates. Lastly, we discuss how operator recurrent networks can be viewed as a deep learning analogue to deterministic algorithms such as boundary control for reconstructing the unknown wave speed in the acoustic wave equation from boundary measurements.

1. Introduction

In standard deep learning, the input data are represented by vectors, and each layer of a deep neural network applies an affine transformation (a matrix-vector product plus a shift) composed with nonlinear activation functions. However, for functions for which the input data are linear operators, vectorizing the input destroys the underlying operator structure. Functions whose inputs are linear operators, which we term *nonlinear operator functions*, are present in a broad class of nonlinear inverse problems for partial differential equations (PDE). That is, the possible reconstructions associated with

2020 Mathematics Subject Classification. 35R30, 62M45, 68T05.
Keywords. Inverse problems, neural networks, wave equation, sparse matrices.

such problems involve nonlinear, nonlocal functions between spaces of data operators and function spaces of "images". Optimality of reconstruction algorithms can be studied with statistical decision theory; however, machine learning offers data-driven approaches that make such studies computationally feasible.

1.1. Nonlinear operator functions, inverse problems and reconstruction

We focus our attention on *nonlinear operator functions*, meaning nonlinear functions whose input consists of linear operators, and whose structure consists of a holomorphic function of an operator composed with a very regular function. This type of function structure is found in a variety of existing solution procedures for nonlinear inverse problems arising from hyperbolic PDEs. The model problem is reconstruction of, or "imaging" the unknown speed $c = c(x)$ of waves inside a body, based on from boundary measurements. In this problem, the body is probed by multiple boundary sources, h, generating waves; the waves that come back are measured at the boundary. The boundary measurements corresponding to an operator $X_c : h \mapsto X_c(h)$, and the inverse problem of determining c from X_c is highly nonlinear. This inverse problem has been extensively studied, e.g., in [5, 13, 14, 45, 46, 53, 61, 75, 83, 86, 89] and the stability of the solution with data containing errors is considered in [4, 5, 15]. The inverse problems for the wave equation with given boundary measurements X_c corresponds to the case when we observe the complete wave patterns on the boundary. This inverse problem is closely related to the inverse travel time problem where only the first arrival times of the waves are observed, see [22, 54, 76, 84, 85, 89]. Even though the underlying physical system, for example, the wave equation, is a linear equation, the inverse problem of finding the coefficient function of this equation is a nonlinear problem. In general, we consider X_c as data given to us and denote it by $X = X_c$.

Established uniqueness proofs, based on boundary control [17, 28, 48] and scattering control [23, 24], for the above mentioned inverse problems lead to solution procedures that are recursive in the data operator, X. These procedures can be viewed as applying an operator-valued series expansion in terms of X followed by some elementary operations such as taking inner products and divisions. Typically, one starts with a boundary source h_0, measures the wave $X(h_0)$ at the boundary and computes a new source h_1 using both h_0 and $X(h_0)$. The process is iterated to thus produce a sequence of sources that converge to an optimal source, called a control, which can effectively determine information about the interior. However, the convergence is typically very slow while the intrinsic stability of the inverse problem is poor. Therefore, a natural question is whether the procedures can be replaced by learned procedures that are adapted to the data, taking advantage of working on a low-dimensional manifold of linear operators. The iterative nature of the procedures suggests the introduction of recurrent neural networks (RNNs). Mathematical properties of the inverse problems

can be used to reduce the number of weights to be learned. Notably, a crucial feature of boundary control is that each iteration involves linear operators that smooth source signals by a finite order, meaning that such operators are compact operators. The compactness is used in a crucial way in the solution of the inverse problem. Moreover, when the data operator and operators appearing in the boundary or scattering control based procedures are discretized and approximated by finite $n \times n$ matrices, one obtains good approximations using sparse and low-rank matrices.

The main goal of this paper is to develop a mathematical framework for supervised learning to solve nonlinear inverse problems, whose underlying structure is that of nonlinear operator functions. Based on the structure of known, constructive uniqueness proofs, we introduce general operator recurrent neural networks that take data in as a linear operator. We further introduce an explicit regularization scheme for training such networks based on compactness, sparsity and rank properties of certain operators embedded in the network. The result is a principled network architecture for which crucial analytic features can be controlled tightly. This stands in contrast to more traditional applications of deep neural networks, such as computer vision and speech recognition, in which little mathematical information about the behavior of the underlying "function" is known. To highlight the potential of deep learning in the context of inverse problems, we prove that our type of network, the weights of which are obtained via training with simulated data, solves the inverse problems *at least as well as* the classical, partial-differential-equation based reconstruction procedures. We analyze the approximation and detailed expressivity properties of our operator recurrent neural networks, and provide generalization estimates and rates with increasing training sets to the best possible network. The universal approximation theorems only guarantee a small approximation error for a sufficiently large network, but do not consider the optimization (training) and generalization errors, which are equally important [43]. From the viewpoint of studying inverse problem, the deep learning framework provides a novel integration of analysis and statistics. In this framework, the architecture is derived from the analysis as a domain adaptive ingredient, while statistical decision theory is used to define what is meant by an "optimal" solution method involving regularization with a finite set of training "data".

Formally, we consider inverse problems of the form $X = F(z)$, where F is a direct operator acting on real-valued vectors z generating linear operators X, and are concerned with determining z given X. The vector z models a real-valued function that is digitized in m points whereas the functions on which X acts are digitized with n points. Thus, we view z as a vector in $\mathbb{R}^m$ and X as a matrix in $\mathbb{R}^{n\times n}$ with $m > n$. We will assume uniqueness. In the digitized framework, we let $z \in B^m(\rho_0)$ and

$$\mathcal{X} = F(B^m(\rho_0)) \subset \mathcal{B}^{n\times n}(\rho_1);$$

here, $B^m(\rho_0)$ denotes a ball with radius ρ_0 in $\mathbb{R}^m$ equipped with the standard Euclidean norm and $\mathcal{B}^{n\times n}(\rho_1)$ a ball with radius ρ_1 in $\mathbb{R}^{n\times n}$ equipped with the operator norm of linear operators $\mathbb{R}^n \to \mathbb{R}^n$, that is,

$$\|X\|_{\mathbb{R}^{n\times n}} = \max_{\|v\|_{\mathbb{R}^n}\le 1} \|Xv\|_{\mathbb{R}^n}.$$

When the map F is injective and one is given (or measures) the matrix X as data, and this data does not contain errors, one can consider a map H that is the left inverse of F on $\mathrm{Ran}(F) = \mathcal{X} = F(B^m(\rho_0)) \subset \mathbb{R}^{n\times n}$, that is, one consider the sequence

$$B^m(\rho_0) \xrightarrow{F} \mathcal{X} \xrightarrow{H} \mathbb{R}^m, \quad H(F(z)) = z \quad \text{for } z \in B^m(\rho_0).$$

However, one has to deal with two challenges: Computing the map H may be difficult and the data $X = F(z)$ may contain errors.

We consider a strategy that is rooted in the analysis of inverse problems, when the reconstruction is obtained in two steps. In the first step, one constructs an intermediate quantity, $y \in \mathbb{R}^n$, from X, which is typically relatively unstable; this construction may have to be repeated for a variable parameter which, upon discretization, yields $(y^1, \dots, y^T)$. From $(y^1, \dots, y^T)$ one then obtains z, typically in a stable manner. To formalize this, we assume that there are functions

$$\boldsymbol{f} = (f^1, \dots, f^T)\colon \mathbb{R}^{n\times n} \to (\mathbb{R}^n)^T, \quad \text{where } f^t\colon \mathbb{R}^{n\times n} \to \mathbb{R}^n, \tag{1}$$

and

$$g\colon (\mathbb{R}^n)^T \to \mathbb{R}^m \tag{2}$$

having the property that $g \circ \boldsymbol{f}\colon \mathbb{R}^{n\times n} \to \mathbb{R}^m$ is an extension of the inverse map H of F defined on $\mathcal{X}$, that is, we consider the sequence

$$B^m(\rho_0) \xrightarrow{F} \mathbb{R}^{n\times n} \xrightarrow{\boldsymbol{f}} (\mathbb{R}^n)^T \xrightarrow{g} \mathbb{R}^m, \tag{3}$$

$$g(\boldsymbol{f}(F(z)) = z \quad \text{for } z \in B^m(\rho_0). \tag{4}$$

As discussed above, the intermediate quantities are denoted by

$$f^t(X) = y^t, \quad t = 1, \dots, T.$$

We will approximate f^t by f^t_θ as an operator recurrent network and g by g_θ as a shallow network with fully connected layers motivated, again, by the analysis of inverse problems. We consider the model, where parameter θ consists of two parts, $\theta = (\theta', \theta'')$, and f_θ depends on θ' and g_θ depends on θ'', that is, the parameters determining f_θ and g_θ are unrelated. This structure, as well as the regularization introduced in the later analysis, could be viewed as the inductive bias.

We will consider how the functions f^t and g can be approximated by operator recurrent neural network with appropriately chosen weights. We will also analyze the case when the data are contaminated with noise, $\mathcal{E}$ say, such that $X + \mathcal{E}$ no longer belongs to $\mathrm{Ran}(F)$. Our goal is to use to use recurrent operator neural networks to find a trainable solution algorithm for an inverse problem so that the architecture is informed by the PDE-based solution methods but in which the measurement noise can be take into account in the training. Moreover, we will show that optimal (general) operator recurrent network under the expected loss can be identified as a Bayes estimator.

Remark 1. In the above, the direct map F is an approximation of a map $\mathcal{F}$ that maps between infinite-dimensional Banach spaces and the map H is an approximate inverse of the map $\mathcal{F}$. In practice, F can be obtained using a numerical discretization, such as the finite element method, to approximate solutions of partial differential equations. When the discretization of the model is taken in to account, the sequence (4) needs to be replaced by the sequence

$$g(\boldsymbol{f}(F(z)) = I_{\mathrm{app}}(z) \quad \text{for } z \in B^m(\rho_0), \tag{5}$$

where $\|I_{\mathrm{app}}(z) - z\| \leq \varepsilon_0$. However, in this paper we assume that the finite-dimensional approximation of function F is so precise that the approximation error ε_0 is negligible, and assume that the identity (4) is valid.

1.2. Related work

There has been a substantial amount of progress concerning applying machine learning techniques to linear or linearized inverse problems, particularly in the domain of natural image processing. However, nonlinear hyperbolic inverse problems are an entirely different class of problems, see e.g., [24, 45, 51, 52, 89] and references therein. A closely related recent work is [32], in which a neural network is trained as an additive term to regularize each iteration of a truncated Neumann series as a way to solve a linear reconstruction task. Our paper also uses truncated Neumann series as an approximation to the holomorphic operator function, but the introduced deep learning architecture is directly adapted from the Neumann series structure rather than regularizing it. There have been other prior works in the area of nonlinear inverse problems. In [42], a deep neural network is constructed mimicking the structure of the filtered back projection algorithm for computerized tomography. In [57], neural networks are used for learning a nonlinear regularization term, also in the context of tomography. Deep neural networks have further been employed for inverse scattering problems, such as in [47, 58, 92] and other related inverse problems in [6–8, 21, 42, 64].

Unrolled deep neural network architectures were first used to solve optimization problems [36], in particular, the iterative shrinkage algorithm (ISTA) [27]; for a recent review, see [68]. Unrolling is a way to select a domain specific architecture for deep neural networks that approximates an operator given implicitly by an iterative scheme [8, Sections 4.9.1 and 4.9.4]. Usage of such architectures for solving inverse problems was outlined in [2] and [78], while further developments came in [3]. Our work has some similarities to unrolling, as we take an existing iterative algorithm and use it as the basis for developing a deep learning strategy.

A crucial feature of our approach is that properties derived from the mathematical analysis provide insight as to how to efficiently and sparsely parametrize the neural network that learns the inverse map. Such sparsity bounds are important because fully general neural network models are heavily over parametrized, making them both difficult to analyze as well as computationally resource intensive. Reducing the parameter space as a way to improve learning also has connections to nascent information-theoretic formulations of deep learning, such as through the information bottleneck method [87]. There is a wide array of existing literature on studying sparsity in neural networks. One popular technique to achieve sparsity is to take a pre-trained dense network and prune parameters with low importance; an early example of this technique is [56], with later examples studying pruning including [30, 37, 62]. However, it is desirable to achieve sparsity without needing to first train a dense network. Indeed, in our work, sparsity bounds are directly imposed for the network parameters that encode the linear transformations across layers. Studies of sparsity promotion either before or during network training include [16,20,66,70]. As will be seen later, sparsity in the network parameters has an interpretation in terms of low-rank approximation of the compact operators appearing in the original iterative scheme that is unrolled. The use of low-rank weight matrices in deep learning has become popular for a variety of applications; see for example [41, 55, 60, 93]. However, these works all exploit low-rank structure that is empirically found rather than mathematically derived. Our sparsity bounds also provide improved generalization bound. This is independent of (regularization) techniques employed to improve upon training [50].

2. Principled architecture

In this section, we focus on the network architecture for f_θ while suppressing the parameter t. This network represents f, which is the main component of the inverse map, H. The design is domain adapted in the sense that it utilizes structure that the inverse map may possess. We exemplify this with the inverse boundary value problem associated with the wave equation and boundary control in the final section of

this paper; however, we expect the architecture to adapt equally well to electrical impedance tomography and the $\overline{\partial}$ method.

2.1. Operator recurrent architecture

We define a specialized neural network architecture, the *operator recurrent network*, that we propose as a suitable architecture for learning certain classes of nonlinear functions whose inputs are linear operators and whose outputs are functions. As mentioned in the introduction, we invoke a discretization turning operators into matrices and functions into vectors.

2.1.1. Standard deep neural network. To draw a comparison with the operator recurrent architecture we will introduce shortly, we first define the standard neural network. This is a function $f_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ with depth L and set of weights θ defined by

$$f_\theta(x) = h_L, \tag{6}$$

$$h_\ell = A_\theta^{\ell,0} h_{\ell-1} + \phi_\ell\big[b_\theta^\ell + A_\theta^{\ell,1} h_{\ell-1}\big], \tag{7}$$

$$h_0 = x. \tag{8}$$

The index $\ell = 0, \ldots, L$ indicates the layer of the neural network. Each vector $h_\ell \in \mathbb{R}^{d_\ell}$ is the output of layer ℓ, where d_ℓ is the width of that layer. For each layer ℓ, the functions $\phi_\ell : \mathbb{R}^{d_\ell} \to \mathbb{R}^{d_\ell}$ are the activation functions, which apply a scalar function to each component, that is, for $x = (x_j)_{j=1}^{d_\ell} \in \mathbb{R}^{d_\ell}$, $\phi_\ell(x) = (\phi_\ell(x_j))_{j=1}^{d_\ell} \in \mathbb{R}^{d_\ell}$.

The matrices $A_\theta^{\ell,0} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$, which typically have an identity matrix as a sub-block, encode skip connections by passing outputs from layer $\ell - 1$ to layer ℓ without being operated on by any activation functions. The $\mathbb{R}^{d_\ell}$-vectors b_θ^ℓ are the bias vectors and the $d_\ell \times d_{\ell-1}$ matrices $A_\theta^{\ell,1}$ are the weight matrices. Each of b_θ^ℓ, $A_\theta^{\ell,0}$, $A_\theta^{\ell,1}$ are dependent (in a context-specific way) on parameters θ to be learned. For example, in the case of convolutional neural networks, $A_\theta^{\ell,1}$ is a block-sparse matrix whose blocks are Toeplitz matrices, and the parameters θ determine the values of the diagonals and off-diagonals of these blocks.

2.1.2. Operator recurrent network. While standard neural networks have enjoyed widespread success in many applications, they are not efficient at approximating functions that are mathematically known to have a multiplicative and highly nonlinear structure. This is because a standard neural network with rectifier activations is a form of a multivariate linear spline. For example, approximating even a univariate polynomial to high accuracy requires a fairly deep neural network [94]. In nonlinear inverse problems, the situation is even more problematic, since their structure includes opera-

tor polynomials where the polynomial is of high degree and the operator is discretized as a large matrix. This situation motivates our new construction.

An *operator recurrent* network has an internal structure reflecting the linear operator nature of the input by performing matrix-matrix multiplications, rather than vectorizing the input and then performing matrix-vector multiplications. To this end, we consider following neural networks.

Definition 2.1. A *basic operator recurrent network* with depth L, width n, and set of weights (or parameters) θ is defined as a function $f_\theta\colon \mathbb{R}^{n\times n} \to \mathbb{R}^n$ given by

$$f_\theta(X) = h_L, \tag{9}$$

$$h_\ell = b_\theta^{\ell,0} + A_\theta^{\ell,0} h_{\ell-1} + B_\theta^{\ell,0}\, X\, h_{\ell-1} + \phi_\ell\big[b_\theta^{\ell,1} + A_\theta^{\ell,1} h_{\ell-1} + B_\theta^{\ell,1}\, X\, h_{\ell-1}\big], \tag{10}$$

where $h_0 \in \mathbb{R}^n$ is an initial vector not explicitly given by the data, the quantities $b_\theta^{\ell,0}$, $b_\theta^{\ell,1} \in \mathbb{R}^n$ and $A_\theta^{\ell,0}$, $A_\theta^{\ell,1}$, $B_\theta^{\ell,0}$, $B_\theta^{\ell,1} \in \mathbb{R}^{n\times n}$ are dependent on the parameters θ, and the ϕ_ℓ are the activation functions.

We note that h_ℓ should be viewed as a *hidden state*. The typical initialization of the hidden state is $h_0 = 0$, though it could be learned as well. This naturally applies in the context of inverse problems; in Section 7 the hidden states take the role of boundary controls.

Remark 2. We may consider h_0 not as part of the initial layer, but instead as the output of an initial layer whose value is entirely determined by a bias vector $b_\theta^{0,0}$ set to be equal to h_0, with all other terms set to zero.

Remark 3. The data matrix, X, is a digitized counterpart of an operator. In Section 7 we realize this as the outcome of numerical discretization. However, the digitization may be obtained through composition with a data acquisition operator, which may be viewed as a pre-processing operator that can be learned. Learning a data acquisition scheme has been considered in different contexts [10, 26, 63, 82].

2.1.3. Activation function. In general, the activation functions ϕ_ℓ may differ at each layer ℓ. We choose the form of $\phi_\ell\colon \mathbb{R}^n \to \mathbb{R}^n$ to be a rectifier (or ReLU). That is, ϕ_ℓ is given by

$$(\phi_\ell(y))_j = \phi_\eta(y_j) = \max(y_j, \eta y_j), \quad j = 1, \dots, n, \tag{11}$$

where $0 \le \eta \le 1$ is either a hyperparameter that is chosen in advance (the "leaky" ReLU) or could be a parameter that is learned during optimization (the "parametric" ReLU). In either case, this choice of activation function is a piecewise-linear function on each vector component.

The choice of the rectifier as the activation function has both pragmatic and mathematical reasons. Indeed, in the case of standard deep neural networks with $\eta = 0$, there is significant empirical evidence indicating that the use of the rectifier activation function promotes sparsity and accelerates training [34, 65]. Rectifier networks are also closely connected with piecewise-linear splines, which are known to interpolate data points while minimizing the second-order total variation [90, 91]. In Section 2.5, we will show that in our case such activations induce piecewise (operator) polynomial behavior.

We note that a network of the form (9)–(10) with activation functions being rectifiers with leaky parameter $\eta > 0$ can have its activation functions replaced, without loss of generality, by standard rectifier activation functions ($\eta = 0$). We let ϕ_η be the activation function in (11). Then we can write

$$\phi_\eta = \eta\,\mathrm{Id} + (1-\eta)\,\phi_0, \tag{12}$$

where Id is the identity map and the activation function ϕ_0 is the standard rectified linear unit (relu). Then, starting with (10), we have

$$\begin{aligned} h_\ell &= b_\theta^{\ell,0} + A_\theta^{\ell,0} h_{\ell-1} + B_\theta^{\ell,0}\, X\, h_{\ell-1} + \phi_\eta\big[b_\theta^{\ell,1} + A_\theta^{\ell,1} h_{\ell-1} + B_\theta^{\ell,1}\, X\, h_{\ell-1}\big] \\ &= (b_\theta^{\ell,0} + \eta b_\theta^{\ell,1}) + (A_\theta^{\ell,0} + \eta A_\theta^{\ell,1}) h_{\ell-1} + (B_\theta^{\ell,0} + \eta B_\theta^{\ell,1})\, X\, h_{\ell-1} \\ &\quad + (1-\eta)\phi_0\big[b_\theta^{\ell,1} + A_\theta^{\ell,1} h_{\ell-1} + B_\theta^{\ell,1}\, X\, h_{\ell-1}\big], \end{aligned} \tag{13}$$

and thus an operator recurrent network with $\eta > 0$ can be replaced by another one with $\eta = 0$ by relabeling some of the biases and weights.

2.1.4. Recurrence. By inspecting (9)–(10), we observe that the input data X is inserted multiplicatively into the network at every layer, so that each computed intermediate output h_ℓ depends both on X and previous intermediate outputs $h_{\ell-1}, h_{\ell-2}, \ldots$ in an identical fashion for each ℓ. In the finite-dimensional setting, such expansions can be viewed as matrix polynomials, and each layer can be thought of as performing another stage of an iteration in which the degree of the polynomial is raised through multiplication by the matrix variable. Thus, the neural network learns nonlinear perturbations of this process at each iteration.

There may be a reason to expect that every iteration not only has the same structure, but is in fact identical. For example, this holds true for nonlinear operator functions given by a truncated Neumann series. Thus, operator recurrent networks can also be interpreted as the unrolling of an iterative nonlinear process, where the recurrence refers to the fact that the output of each layer is fed back into another layer that may have the same weights.

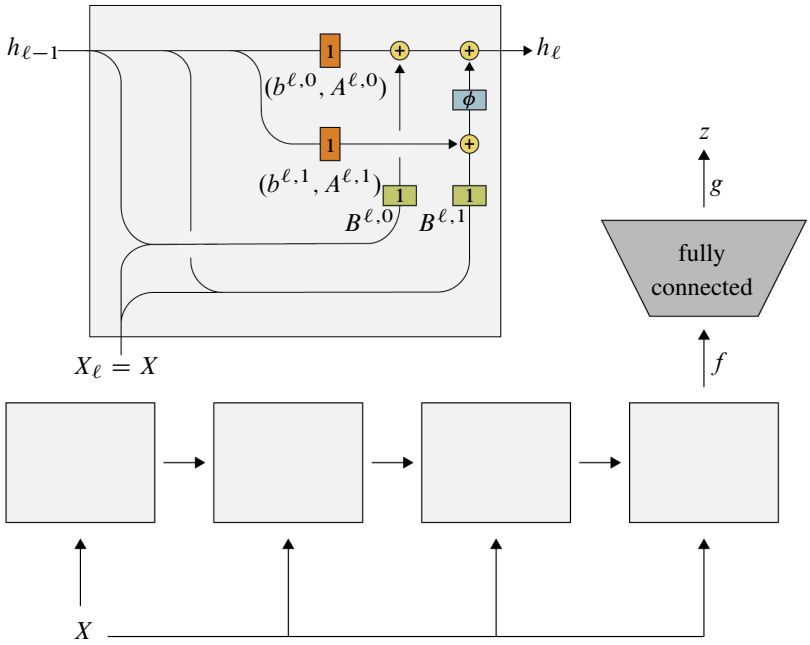

Figure 1. Cell (top) of the operator recurrent network (bottom) architecture. When concatenated with a feed-forward network consisting of a few fully connected layers, the network adapts to inverse problems. The data operator X is inserted multiplicatively into the network at each cell. The initial hidden state h_0 is typically chosen to be zero.

2.2. General operator recurrent networks

A general operator recurrent network is obtained from a basic operator recurrent network merely by adding memory.

Definition 2.2. A *general operator recurrent network* of level K is an extension of the basic operator recurrent network, including terms that contain $h_{\ell-k}$ in the expression for h_ℓ, that is,

$$f_\theta(X) = h_L, \tag{14}$$

$$h_\ell = b_\theta^{\ell,0} + \sum_{k=1,\dots,K;i=0} \left(A_\theta^{\ell,k,i} h_{\ell-k} + B_\theta^{\ell,k,i}\, X\, h_{\ell-k}\right) + \phi_\ell\left[b_\theta^{\ell,1} + \sum_{k=1,\dots,K;i=1} \left(A_\theta^{\ell,k,i} h_{\ell-k} + B_\theta^{\ell,k,i}\, X\, h_{\ell-k}\right)\right], \tag{15}$$

for $\ell \geq 1$, where $h_0 \in \mathbb{R}^n$ is some initial vector not explicitly given by the data, that is, the initial hidden state, $h_{-k} = 0$ for $-k < 0$, and the quantities $b_\theta^{\ell,0}$, $b_\theta^{\ell,1} \in \mathbb{R}^n$ and $A_\theta^{\ell,k,i}$, $B_\theta^{\ell,k,i} \in \mathbb{R}^{n\times n}$ are dependent on the parameters θ, and the ϕ_ℓ are the activation functions.

Basic and general operator recurrent networks can be further generalized upon replacing vectors h_ℓ and biases in $\mathbb{R}^n$ by sets of r vectors, that is, matrices in $\mathbb{R}^{n\times r}$. We will not consider this in the analysis.

In the general operator recurrent network (14)–(15), the dependency of h_ℓ on previous outputs $h_{\ell-m}$ for $m > 1$ is an explicit way to encode skip connections, which feature prominently in applications of standard neural networks [38, 79]. In standard neural networks, however, similar generalizations are fully included in the basic definition since they can be implemented by increasing the width of the network. However, in operator recurrent networks, the width is fixed and so this generalization must be explicitly included. In the following discussions, however, the basic definition (9)–(10) is sufficient as discussed in the example below.

A general operator recurrent network can be written as a basic operator recurrent network by extending the width of the network. We show this explicitly starting from (14)–(15). Let $\widetilde{h}_\ell = (h_\ell, \dots, h_{\ell-K-1})^T \in \mathbb{R}^{nK}$, where $h_{-i} = 0$ for $i > 0$. Also, let

$$\widetilde{A}_\theta^{\ell,i} = \begin{pmatrix} A_\theta^{\ell-1,1,i} & A_\theta^{\ell-1,2,i} & \dots & A_\theta^{\ell-1,K-1,i} & A_\theta^{\ell-1,K,i} \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{pmatrix}, \tag{16}$$

$$\widetilde{B}_\theta^{\ell,i} = \begin{pmatrix} B_\theta^{\ell-1,1,i} & B_\theta^{\ell-1,2,i} & \dots & B_\theta^{\ell-1,K-1,i} & B_\theta^{\ell-1,K,i} \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{pmatrix}, \tag{17}$$

for $i = 1, 2$ and $\widetilde{b}_\theta^{\ell,i} = (b_\theta^{\ell-1,i}, \dots, b_\theta^{\ell-K,i})^T \in \mathbb{R}^{nK}$ for $i = 1, 2$. Also, let $\widetilde{X} = \operatorname{diag}(X, \dots, X) \in \mathbb{R}^{nK\times nK}$. Then the general operator recurrent network f_θ given in (14)–(15) can be written as a basic operator recurrent network $\widetilde{f}_\theta \colon \mathbb{R}^{nK\times nK} \to \mathbb{R}^{nK}$ given by

$$\widetilde{f}_\theta(\widetilde{X}) = \widetilde{h}_L, \tag{18}$$

$$\begin{aligned} \widetilde{h}_\ell = \widetilde{b}_\theta^{\ell,0} + \widetilde{A}_\theta^{\ell,0}\widetilde{h}_{\ell-1} + \widetilde{B}_\theta^{\ell,0}\widetilde{X}\widetilde{h}_{\ell-1} \\ + \phi_\ell\big[\widetilde{b}_\theta^{\ell,1} + \widetilde{A}_\theta^{\ell,1}\widetilde{h}_{\ell-1} + \widetilde{B}_\theta^{\ell,1}\widetilde{X}\widetilde{h}_{\ell-1}\big], \end{aligned} \tag{19}$$

and setting $f_\theta(X) = \Pi_n(\widetilde{f}_\theta(\widetilde{X}))$. Here, $\Pi_n\colon \mathbb{R}^{nK} \to \mathbb{R}^n$ is the operator

$$\Pi_n(y_1, y_2, \dots, y_{nK}) = (y_1, y_2, \dots, y_n).$$

Also, we observe that $\|\widetilde{X}\|_{\mathbb{R}^{nK}\to\mathbb{R}^{nK}} = \|X\|_{\mathbb{R}^n\to\mathbb{R}^n}$.

We can contrast this construction with the standard neural network definitions (6)–(8). In the standard neural network, a vector x is the input, and the intermediate outputs h_ℓ at each layer ℓ are produced by repeatedly applying matrix-vector products as well as activation functions in some order. In contrast, in the operator recurrent network, the input is a matrix X, and it is multiplied on both the left and right by matrices. At the first layer, this is still equivalent to a standard neural network, since the action of a matrix on another matrix is linear. However, at all subsequent layers, this is no longer equivalent, since the matrix X is re-introduced at each layer and is multiplied with the previous output $h_{\ell-1}$.

Remark 4. A standard "additive" neural network (cf. (6)–(8)) with input $x \in \mathbb{R}^n$ can be written as a general operator recurrent network (14)–(15) as follows. We set $X = \operatorname{diag}(x_1, \dots, x_n)$, $h_0 = \mathbf{1} = (1, 1, \dots, 1)^T$ and let $K = 1$. For $\ell = 1$ we choose the weight matrices to be

$$B_\theta^{1,1,0} = A_\theta^{1,0}, \quad B_\theta^{1,1,1} = A_\theta^{1,1}, \quad A_\theta^{1,1,i} = 0, \quad i = 0, 1;$$

for $2 \le \ell \le L$ we choose the weight matrices to be

$$A_\theta^{\ell,1,0} = A_\theta^{\ell,0}, \quad A_\theta^{\ell,1,1} = A_\theta^{\ell,1}, \quad B_\theta^{\ell,1,i} = 0, \quad i = 0, 1.$$

To simplify notation, in particular the indexing of variables, we consider mostly basic operator recurrent networks, that is, the case $K = 1$. However, the results can be generalized in a straightforward way to general operator recurrent networks. The general operator recurrent network will play a fundamental role in Theorem 2.3 only.

2.3. Sparse representation of trained matrices

Next, we specify how the biases and weights depend on the parameters θ. In a typical fully-connected layer for a standard neural network, θ determines the entries of the biases and weights. More precisely,

$$b_\theta^\ell = \theta_0^\ell, \quad A_\theta^{\ell,1} = \begin{bmatrix} \theta_1^\ell & \theta_2^\ell & \cdots & \theta_{d_\ell}^\ell \end{bmatrix}, \tag{20}$$

where

$$\theta = \left\{\theta_p^\ell \in \mathbb{R}^{d_{\ell+1}} : \ell = 1, \dots, L, p = 0, \dots, d_\ell\right\}$$

stands for a set of column vectors. We consider the parametrization of basic operator recurrent networks in terms of θ. The matrices $A_\theta^{\ell,i}$, $B_\theta^{\ell,i}$ in (10) could depend on θ

similarly to (20). However, in our analysis, it is beneficial to provide an alternative quadratic dependence: For each ℓ and $i = 0, 1$ there are $4n$ column vectors

$$\theta_1^{\ell,i}, \dots, \theta_{4n}^{\ell,i} \in \mathbb{R}^n$$

within the parameter set θ such that for $i = 0, 1$,

$$A_\theta^{\ell,i} = A^{\ell,i,(0)} + A_\theta^{\ell,i,(1)}, \quad A_\theta^{\ell,i,(1)} = \sum_{p=1}^{n} \theta_{2p-1}^{\ell,i} (\theta_{2p}^{\ell,i})^T, \tag{21}$$

and similarly for $B_\theta^{\ell,i}$,

$$B_\theta^{\ell,i} = B^{\ell,i,(0)} + B_\theta^{\ell,i,(1)}, \quad B_\theta^{\ell,i,(1)} = \sum_{p=n+1}^{2n} \theta_{2p-1}^{\ell,i} (\theta_{2p}^{\ell,i})^T. \tag{22}$$

Each $A^{\ell,i,(0)}$ and $B^{\ell,i,(0)}$ is a fixed operator that does not depend on parameter θ and is "handcrafted". The resulting deep neural network is illustrated in Figure 1. The fixed operators are typically the zero operator or the identity operator, but they can be also other operators that are chosen depending on the specific application. Examples of such operators suitable for solving the inverse problem for the wave equation are considered later in Section 7, in particular the discussion below (289).

Remark 5. Following Remark 4, choosing for $2 \le \ell \le L$ the weight matrices to be $A_\theta^{\ell,1,i,(0)} = I$ and $B_\theta^{\ell,1,i} = 0$, $i = 0, 1$, we obtain a residual network [38].

We now assume that the matrices $A^{\ell,i,(0)}$ and $B^{\ell,i,(0)}$ and the bias vectors satisfy

$$\sum_{i=0}^{1} \big(\|A^{\ell,i,(0)}\|_{\mathbb{R}^n \to \mathbb{R}^n} + \|B^{\ell,i,(0)}\|_{\mathbb{R}^n \to \mathbb{R}^n} + |b_\theta^{\ell,i}| \big) \le c_0, \tag{23}$$

for some $c_0 \ge 1$. The lower bound, 1, arises as we allow the relevant nonlearned matrices to be identity matrices. This makes possible the ResNet-type architectures that contain layers

$$h \to \phi\big(h + A_\theta^{\ell,(1)} h + B_\theta^{\ell,(1)} Xh\big) \quad \text{or} \quad h \to h + \phi\big(A_\theta^{\ell,(1)} h + B_\theta^{\ell,(1)} Xh\big).$$

We parametrize the bias vectors by $b_\theta^{\ell,i} = \theta_0^{\ell,1,i} \in \mathbb{R}^n$, $i = 0, 1$. With these notations f_θ is determined by the set of parameters θ that is given as an ordered sequence

$$\theta = \big[\theta_p^{\ell,i} \in \mathbb{R}^n : \ell = 1, 2, \dots, L,\ p = 1, 2, \dots, 4n,\ i = 0, 1\big] \cup \big[\theta_0^{\ell,i} \in \mathbb{R}^n : \ell = 1, 2, \dots, L,\ i = 0, 1\big]. \tag{24}$$

We denote the index set in the above sequence by

$$\begin{aligned} P &= P_1 \cup P_2, \\ P_1 &= \{(\ell, i, p) : \ell = 1, 2, \dots, L,\ i = 0, 1,\ p = 1, 2, \dots, 4n\}, \\ P_2 &= \{(\ell, i, p) : \ell = 1, 2, \dots, L,\ i = 0, 1,\ p = 0\}. \end{aligned} \tag{25}$$

We note that the indices in P_1 are related to the learnable weight matrices, $A_\theta^{\ell,i,(1)}$ and $B_\theta^{\ell,i,(1)}$ of the basic operator recurrent network, and the indices in P_2 are related to bias vectors. Below, we use the fact that P_1 has $\#P_1 \leq 4nL$ elements, and P_2 has $\#P_2 \leq 2L$ elements. We note that $\#P_2$ is significantly smaller than $\#P_1$ and that $\#P_2$ is independent of n.

For the general recurrent operator networks we add the index $k = 1, \dots, K$ and replace the above parameters by the ordered sequences

$$\begin{aligned} \widetilde{\theta} = \big[\widetilde{\theta}_p^{\ell,i,k} \in \mathbb{R}^n : \ell = 1, 2, \dots, L,\ p = 1, 2, \dots, 4n,\ k = 1, 2, \dots, K,\ i = 0, 1\big] \\ \cup \big[\widetilde{\theta}_0^{\ell,i,0} \in \mathbb{R}^n :\ \ell = 1, 2, \dots, L,\ i = 0, 1\big]. \end{aligned} \tag{26}$$

Also, for the general recurrent operator networks we denote the index set in the above sequence by

$$\begin{aligned} \widetilde{P} &= \widetilde{P}_1 \cup \widetilde{P}_2, \\ \widetilde{P}_1 &= \{(\ell, i, p, k) : \ell = 1, 2, \dots, L,\ i = 0, 1,\ p = 1, 2, \dots, 4n,\ k = 1, 2, \dots, K\}, \\ \widetilde{P}_2 &= \{(\ell, i, p, k) : \ell = 1, 2, \dots, L,\ i = 0, 1,\ p = 0,\ k = 0\}. \end{aligned} \tag{27}$$

Next, we return to considering the basic recurrent operator networks. From a (numerical) linear algebra viewpoint, the decomposition (21) expresses the matrix $A_\theta^{\ell,i,(1)}$ as a sum of rank 1 matrices, similar to a singular value decomposition. This structure is valuable for our analysis, since it means that we essentially learn a factorization of these matrices rather than the explicit matrix elements. We will exploit that in Section 3.1 while introducing low-rank structures.

Remark 6. In the above, the parameters in each layer are allowed to be different and independent. However, it is natural to consider the case that a subset of parameters is shared across layers. We will analyze the impact of shared weights in various estimates below.

2.4. Approximation properties

Estimates for nonlinear operator functions in the holomorphic calculus. Here, we establish the approximation power of operator recurrent networks, within a certain

space of general nonlinear operator functions. We begin by studying the approximation of functions mapping the linear operator $X: \mathbb{R}^n \to \mathbb{R}^n$ to another linear operator $q(X): \mathbb{R}^n \to \mathbb{R}^n$. This map q is holomorphic in X and it is defined by the fundamental formula of holomorphic operator calculus [81],

$$q(X) = \frac{1}{2\pi i} \int_\gamma q(z)\,(X - z)^{-1}\,\mathrm{d}z, \tag{28}$$

where $\gamma \subset \mathbb{C}$ is a circle having radius larger than the norm of X, oriented in the positive direction. We contract the operator $q(X)$ with a vector $v \in \bar{B}^n(1)$, where $\bar{B}^n(R)$ is the closed ball of radius $R > 0$. In the context of inverse problems and reconstruction, $q(X)$ is often polynomial. To emphasize this context, we write $f(X)$ for $q(X)v$. An example of a neural network based on holomorphic operator calculus is considered in Section 6. We then consider a map, H, obtained from the composition with a nonlinear, smooth function $g: \mathbb{R}^n \to \mathbb{R}^m$.

Below, we use the norms

$$\begin{aligned} \|g\|_{\mathbb{C}^k(\bar{B}^n(r);\mathbb{R}^m)} &= \max_{y \in \bar{B}^n(r)} \max_{|\alpha| \le k} \|D^\alpha g(y)\|_{\mathbb{R}^m}, \\ \|f\|_{C(\mathcal{B}^{n\times n};\mathbb{R}^m)} &= \max_{X \in \mathcal{B}^{n\times n}} \|f(X)\|_{\mathbb{R}^m}, \end{aligned} \tag{29}$$

where $D^\alpha g(y) = (\frac{\partial}{\partial y_1})^{\alpha_2} (\frac{\partial}{\partial y_2})^{\alpha_2} \dots (\frac{\partial}{\partial y_n})^{\alpha_n} g(y)$, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathbb{N}^n$, and $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$. We denote the linear operator norm by $\|X\|_L = \|X\|_{\mathbb{R}^n \to \mathbb{R}^n}$ and recall that $\mathcal{B}^{n\times n} = \{X \in \mathbb{R}^{n\times n} : \|X\|_{\mathbb{R}^n \to \mathbb{R}^n} \le 1\}$ is the closed unit ball in the set of matrices.

Theorem 2.1. *Consider a nonlinear operator function $H: \mathbb{R}^{n\times n} \to \mathbb{R}^m$, defined by*

$$H: X \mapsto g(v^\top q(X)), \tag{30}$$

where q is obtained using the holomorphic operator calculus, $v \in \bar{B}^n(1)$ and $g: \mathbb{R}^n \to \mathbb{R}^m$, satisfying

- *q is a holomorphic function whose domain contains a complex disk $\mathbb{D}_{1+r_1}$ having radius $1 + r_1 > 1 + r$ centered at the origin, for some $r \in (0, 1)$,*
- *$g \in C^k(\bar{B}^n(2); \mathbb{R}^m)$ for some $k \ge 1$, and*
- *q and g are both bounded by 1.*

Let $\varepsilon \in (0, 1)$. Then there exists a general operator recurrent network, H_θ, which depth $L \le L_0$, level $K = 2$, and width $W \le W_0$, and constant $C = C(k, n, r)$ such that

$$\|H - H_\theta\|_{C(\mathcal{B}^{n\times n};\mathbb{R}^m)} \le \varepsilon \tag{31}$$

with

$$L_0 = C\left(\log\left(\frac{4\|g\|_{C^1(\bar{B}^n(2))}}{r\varepsilon}\right) + \log\left(\frac{4^{k+1}\|g\|_{C^k(\bar{B}^n(2))}}{\varepsilon}\right) + 1\right), \tag{32}$$

$$W_0 = Cmn\left(\frac{\varepsilon}{4^{k+1}\|g\|_{C^k(\bar{B}^n(2))}}\right)^{-n/k}\left(\log\left(\frac{4^{k+1}\|g\|_{C^k(\bar{B}^n(2))}}{\varepsilon}\right) + 1\right). \tag{33}$$

Proof. In the proof we will first estimate how to approximate a holomorphic function of an operator by a polynomial and represent the obtained polynomial as a general operator recurrent network. After this we adapt Yarotsky's results on quantified approximation of a function pointwise by a deep neural network and represent the obtained network as a recurrent operator network.

To prove the claim, we first approximate $q(X)$ locally by a polynomial $P(X)$. As q is holomorphic on some disk $\mathbb{D}_{1+r_1}$, where $r_1 > r > 0$ and bounded by 1, its derivatives at zero satisfy

$$q^{(j)}(0) = \frac{j!}{2\pi i}\int_{|z|=1+r}\frac{q(z)}{z^{j+1}}\,\mathrm{d}z \tag{34}$$

and, hence, using that $\|q\| \leq 1$, its Taylor coefficients at zero satisfy

$$a_j = \frac{1}{j!}q^{(j)}(0), \quad |a_j| \leq \frac{1}{(1+r)^{j+1}}. \tag{35}$$

Thus, we have the Taylor polynomial

$$P(z) = \sum_{j=0}^{\ell} a_j z^j, \tag{36}$$

which satisfies for $|z| \leq 1$

$$|q(z) - P(z)| \leq \sum_{p=\ell+1}^{\infty}\frac{1}{(1+r)^{p+1}} \leq \frac{(1+r)^{-\ell-1}}{r}. \tag{37}$$

Hence, if $q(X)$ is defined using the holomorphic functional calculus, then it can be approximated by the matrix polynomial $P(X)$, with

$$\|q(X) - P(X)\|_{\mathbb{C}^n\to\mathbb{C}^n} \leq \frac{(1+r)^{-\ell}}{r} = \varepsilon_0. \tag{38}$$

Given $\varepsilon_0 < r$, we choose ℓ to be

$$\ell = 1 + \left\lfloor \frac{\log((r\varepsilon_0)^{-1})}{\log(1+r)} \right\rfloor, \tag{39}$$

where $\lfloor \cdot \rfloor$ is the integer part of the real number s. From the discussion in the previous section, it is thus possible to exactly represent the map

$$X \mapsto v^\top P(X) = P(X)v \tag{40}$$

(see (36)) that approximates the map $X \mapsto v^\top q(X)$, using a general operator recurrent network

$$\begin{aligned} P(X)v &= h_{2\ell+3}, \\ h_0 &= 0, \\ h_1 &= v, \\ h_{2j} &= a_{j-1}h_{2j-1} + h_{2j-2}, \quad j = 1, 2, \ldots, \ell+1, \\ h_{2j+1} &= Xh_{2j-1}, \qquad\qquad j = 1, 2, \ldots, \ell+1 \end{aligned}$$

(so that $h_{2j+1} = X^j v$ and $h_{2j+1} = \sum_{p=0}^{j-1} a_p X^p v$) of depth $2\ell + 3$ and level 2, and whose hidden states are vectors in $\mathbb{R}^n$ and weight matrices are $n \times n$.

Next, we consider the network approximation of g. First, we note that in the exact nonlinear function f, the function g takes in a vector $q(X)v$ whose norm is bounded by 1, since $|q(z)| \leq 1$ on the closed disk of radius $1 + r$, and $\|v\| \leq 1$. Thus, we are in the setting of approximating a nonlinear function $g: \mathbb{R}^n \to \mathbb{R}^n$ uniformly by neural networks on a bounded domain. By Remark 4, the standard neural network (6)–(8) can be written as a general operator recurrent network (14)–(15), and thus to consider the approximation of g we can use the results for standard neural networks. Such approximation problems have been studied in a wide variety of settings. Here, we use the results of Yarotsky [94] applied to the function

$$g_1(y) = \frac{g(4y)}{4^k \|g\|_{C^k(\bar{B}^n(2))}}. \tag{41}$$

This normalization is such that $\|g_1\|_{C^k(\bar{B}^n(1/2))} \leq 1$, where the domain of g_1 is a ball in $\mathbb{R}^n$ of radius $1/2$. With this normalization, then by Theorem 1 of [94], there exists a constant $C = C(k, n)$ such that a standard additive neural network G exists, satisfying

$$\|g - G\|_{L^\infty(\bar{B}^n(2))} \leq \varepsilon_1, \tag{42}$$

and the depth L' and width W' of G satisfy

$$L' \leq C\left(\log\left(\frac{4^k \|g\|_{C^k(\bar{B}^n(2))}}{\varepsilon_1}\right) + 1\right), \tag{43}$$

$$W' \leq Cmn\left(\frac{\varepsilon_1}{4^k \|g\|_{C^k(\bar{B}^n(2))}}\right)^{-n/k} \log\left(\frac{4^k \|g\|_{C^k(\bar{B}^n(2))}}{\varepsilon_1} + 1\right). \tag{44}$$

Concatenating the previous two networks, we can construct an operator recurrent network

$$f_\theta(X) = G(v^\top P(X)) = G(P(X)v).$$

By abuse of earlier notation we absorbed the weights of G in θ. We then prove our main estimate:

$$\begin{aligned}
&\|g(q(X)v) - G(P(X)v)\|_{\mathbb{R}^n} \\
&\quad \le \|g(q(X)v) - g(P(X)v)\| + \|g(P(X)v) - G(P(X)v)\| \\
&\quad \le \|g\|_{C^1}\|(q(X) - P(X))v\| + \|g - G\|_{C^0(\bar{B}^n(1+r))} \\
&\quad \le \|g\|_{C^1}\varepsilon_0 + \varepsilon_1. \qquad (45)
\end{aligned}$$

We choose $\varepsilon/2 = \|g\|_{C^1}\varepsilon_0 = \varepsilon_1$. Then we set

$$\ell = \frac{\log(2\|g\|_{C^1(\bar{B}^n(2))}/(r\varepsilon))}{\log(1+r)}, \qquad (46)$$

and redefine C to include dependencies on r, to find the full depth bound for the network

$$L \le C\left(\log\left(\frac{4\|g\|_{C^1((\bar{B}^n(2)))}}{r\varepsilon}\right) + \log\left(\frac{4^{k+1}\|g\|_{C^k(\bar{B}^n(2))}}{\varepsilon}\right) + 1\right), \qquad (47)$$

while the width W satisfies

$$W \le Cmn\left(\frac{\varepsilon}{4^{k+1}\|g\|_{C^k(\bar{B}^n(2))}}\right)^{-n/k}\left(\log\left(\frac{4^{k+1}\|g\|_{C^k(\bar{B}^n(2))}}{\varepsilon}\right) + 1\right). \qquad (48)$$

This completes the proof. ∎

Because neural networks are naturally compositional, it is straightforward to extend Theorem 2.1 to the case where the function f being approximated is given by a composition of functions of the form (30).

Theorem 2.2. *Let $J \in \mathbb{Z}_+$ and $\varepsilon \in (0,1)$. Suppose there is a sequence of holomorphic functions q_j and smooth functions g_j for $j = 1, \dots, J$, where the q_j and g_j satisfy the same assumptions as functions q and g in Theorem 2.1 with $m = n$, and $v \in \bar{B}^n(1)$. Consider a nonlinear operator function, H, defined by*

$$X \mapsto g_J\big(q_J(X)g_{J-1}\big(q_{J-1}(X)\dots g_2\big(q_2(X)g_1(v^\top q_1(X))\big)\big)\dots\big). \qquad (49)$$

There exists an operator recurrent network H_θ with depth JL and width W, with W and L given by (33) and (32), respectively, such that

$$\|H - H_\theta\|_{C(\mathcal{B}^{n\times n};\mathbb{R}^n)} \le C'\varepsilon \qquad (50)$$

with constant $C' = C'(k, n, r, J)$.

We will introduce a function $\mathcal{R}$ that measures the norms of the network parameters, θ, and provide an upper bound on the value of this function in an approximation of such maps f or H by operator recurrent networks. This additional control over the norms of the weight parameters will later be used to bound the derivatives of f_θ or H_θ, ultimately leading to a generalization bound.

Universal approximation by general operator recurrent networks. Next we show an universal approximation result for general operator recurrent networks. We recall that the general operator recurrent networks can be written also as a basic operator recurrent network with an increased width, as shown in formulas (18)–(19).

Theorem 2.3. *Let $n, K \in \mathbb{Z}_+$ and*

$$\mathcal{F}^{(n,K)} = \left\{ f_\theta^{(L,K)} \colon \mathbb{R}^{n\times n} \to \mathbb{R}^n \mid L \in \mathbb{Z}_+,\ \theta \in (\mathbb{R}^n)^{\#\tilde{P}} \right\}$$

be the space of general operator recurrent networks $f_\theta^{(L,K)}$ of the form (14)–(15) that have the level K, the length L and the width n. Let $\mathcal{Z} \subset \mathbb{R}^{n\times n}$ be a compact set. Then for $K = 2n + 1$, the set $\mathcal{F}^{(n,K)}$ is dense in the space $C(\mathcal{Z}; \mathbb{R}^n)$.

Proof. In the proof we will first consider the matrix X as a vector in R^{n^2} and approximate a function $X \to g(X)$, where $g \in C(\mathcal{Z}; \mathbb{R}^n)$, using a standard neural network. After this we represent the obtained neural network as a general recurrent neural network that has the level $K = 2n + 1$.

Let $X = (x_{jk})_{j,k=1}^n \in \mathbb{R}^{n\times n}$. We can consider X as a vector consisting of n^2 elements and define a single layer neural network $G \colon \mathbb{R}^{n\times n} \to \mathbb{R}^n$ of form

$$\mathcal{G}(X) = (\mathcal{G}_p(X))_{p=1}^n, \tag{51}$$

where

$$\mathcal{G}_p(X) = \sum_{m=1}^M b_p^{(m)} \phi\left(\left(\sum_{j,k=1}^n a_{pj}^{(km)} x_{jk} \right) + c_p^{(m)} \right), \tag{52}$$

$p = 1, 2, \ldots, n$ and $b_p^{(m)}, a_{pj}^{(km)}, c_p^{(m)} \in \mathbb{R}$.

Let now $g \colon \mathbb{R}^{n\times n} \to \mathbb{R}^n$, $g(X) = (g_p(X))_{p=1}^n$ be a continuous function, $\varepsilon > 0$ and $\mathcal{Z} \subset \mathbb{R}^{n\times n}$ be a compact set. By universal approximation results for standard neural networks [40, 77], for each p there is a neural network $\mathcal{G}_p \colon \mathbb{R}^{n\times n} \to \mathbb{R}^n$ of the form (52) (having a sufficiently large width M) such that

$$\|\mathcal{G}_p(X) - g_p(X)\|_{\mathbb{R}^n} \leq \frac{\varepsilon}{n} \quad \text{for all } X \in \mathcal{Z}. \tag{53}$$

We can write $\mathcal{G}(X)$ using matrix notation as

$$\mathcal{G}(X) = \sum_{m=1}^M B^{(m)} \phi\left(c^{(m)} + \sum_{k=1}^n A^{(km)} X v_k \right), \tag{54}$$

where

$$B^{(m)} = \operatorname{diag}(b_1^{(m)}, \dots, b_n^{(m)}) \in \mathbb{R}^{n\times n}, \quad A^{(km)} = (a_{pj}^{(km)})_{p,j=1}^n \in \mathbb{R}^{n\times n},$$
$$c^{(m)} = (c_p^{(m)})_{p=1}^n \in \mathbb{R}^n, \qquad v_k = (\delta_{pk})_{p=1}^n \in \mathbb{R}^n.$$

Moreover, we can write $\mathcal{G}(X)$ in (54) as

$$\begin{aligned} \mathcal{G}(X) &= S_M, \\ S_0 &= 0, \\ S_m &= \phi\Big(B^{(m)} c^{(m)} + \sum_{k=1}^n B^{(m)} A^{(km)} X v_k \Big) + S_{m-1}. \end{aligned} \tag{55}$$

Writing for $m = 0, 1, \dots, M$ and $k = 1, 2, \dots, n$,

$$h_0 = 0, \tag{56}$$
$$h_{m(2n+1)+2k-1} = v_k, \tag{57}$$
$$h_{m(2n+1)+2k} = A^{(km)} X h_{m(2n+1)+2k-1}, \tag{58}$$
$$h_{m(2n+1)+2n+1} = \phi\Big(B^{(m)} c^{(m)} + \sum_{k=1}^n B^{(m)} h_{m(2n+1)+2k} \Big) + h_{m(2n+1)}, \tag{59}$$

so that

$$h_{m(2n+1)} = S_{m-1},$$

we see that $\mathcal{G}(X) = S_M = h_{M(2n+1)}$. Thus, $\mathcal{G}(X)$ can be written as a general operator recurrent network (14)–(15) having depth $L = (M+1)(2n+1)$, level $K = 2n+1$ and width n, and parameters

$$\begin{aligned} &h_0 = 0, \\ &b_\theta^{\ell,0} = v_k,\ b_\theta^{\ell,1} = 0,\ A_\theta^{\ell,k,i} = 0,\ B_\theta^{\ell,k,i} = 0 && \text{for } \ell = m(2n+1)+2k-1, \\ &b_\theta^{\ell,i} = 0,\ A_\theta^{\ell,k,i} = 0,\ B_\theta^{\ell,k,0} = A^{(km)},\ B_\theta^{\ell,k,1} = 0 && \text{for } \ell = m(2n+1)+2k, \end{aligned}$$

and for $\ell = m(2n+1)+2n+1$,

$$\begin{aligned} b_\theta^{\ell,0} = 0,\ b_\theta^{\ell,1} &= B^{(m)} c^{(m)}, \\ A_\theta^{\ell,k,0} = 0,\ A_\theta^{\ell,k,1} &= B^{(m)},\ B_\theta^{\ell,k,i} = 0 && \text{for } k \le K-1, \\ b_\theta^{\ell,i} = 0,\ A_\theta^{\ell,k,0} &= I,\ A_\theta^{\ell,k,1} = 0,\ B_\theta^{\ell,k,i} = 0 && \text{for } k = K. \end{aligned}$$

Moreover, by (53), the inequality

$$\|\mathcal{G}(X) - g(X)\|_{\mathbb{R}^n} \le \varepsilon \quad \text{for all } X \in \mathcal{Z} \tag{60}$$

is satisfied. ∎

Remark 7. Let $\mathcal{K}_{L,K} = \mathcal{K}^n_{L,K}$ be the space of functions $X \to f_\theta(X)$ where f_θ is a general recurrent neural network of depth L, level K and width n and vanishing non-learned parts of the weight matrices, $A^{\ell,i,k,(0)}$ and $B^{\ell,i,k,(0)}$. Moreover, let $\mathcal{K}^{(\mathrm{sp})}_{L,K} \subset \mathcal{K}_{L,K}$ be the space of (special) general recurrent neural network $f_\theta(\mathbf{M}) \in \mathcal{K}_{L,K}$ that of the form (55) and that can be written in the form (14)–(15) with $\theta \in \widetilde{\Theta}_{L,K}$. We note that $\theta \in \widetilde{\Theta}_{L,K}$ implies that the learned parts of the weigh matrices, $A^{\ell,i,k,(1)}_\theta$ and $B^{\ell,i,k,(1)}_\theta$ are bounded.

We observe first that in formula (52) we can multiply numbers $a^{(km)}_{pj}$, $b^{(m)}_p$, and $c^{(m)}_p$ by $0 < \lambda < 1$ then the function $g(X)$ is changed to $\lambda^2 g(X)$. Second, we observe that if $g^{(1)}(X)$ and $g^{(2)}(X)$ are two neural networks in $\mathcal{K}^{(\mathrm{sp})}_{L,K}$, then their sum,

$$g^{(1)}(X) + g^{(2)}(X),$$

can be written as a function $g(X) \in \mathcal{K}^{(\mathrm{sp})}_{2L,K}$ by replacing in definition (52) of $g^{(2)}_p(X)$ the initial value $S_0 = h_0 = 0$ of $g^{(2)}_p(X)$ by the output of the neural network $g^{(1)}_p(X)$. Note that then the sum $g^{(1)}_p(X) + g^{(2)}(X)$ of the two neural networks of length L is represented as a neural network of the double length $2L$.

By combining the above two observations, we conclude that the union

$$\mathcal{K}^{(\mathrm{sp})}_{\infty,K} = \bigcup_{L=1}^{\infty} K^{(\mathrm{sp})}_{L,K} \tag{61}$$

is a linear subspace that is equal to the space of neural networks $\bigcup_{L=1}^\infty \mathcal{K}_{L,K}$ considered in Theorem 2.3. The fact that $\mathcal{K}^{(\mathrm{sp})}_{\infty,K}$ is a linear subspace will be essential in Section 4.3, where we consider Bayes estimators and the orthogonal projection to the subspace $\mathcal{K}^{(\mathrm{sp})}_{\infty,K}$. Moreover, Theorem 2.3 implies that for $K \geq 2n + 1$ the set $\mathcal{K}^{(\mathrm{sp})}_{\infty,K}$ is dense in $L^\infty(\mathcal{B}_{n\times n}(1); \mathbb{R}^n)$.

2.5. Expressivity

One way to assess the representational power of a network architecture is to study its range. More precisely, suppose that one can partition the output space into regions and also locally characterize the network as it is restricted to each of these regions. Networks that partition the output space to a larger number of regions are considered to be more complex, or in other words, possess better representational power. In regular, deep rectifier networks, which are linear splines that can be written as a composition of max-affine spline operators [11], each application of the rectifier activation partitions the output space into regions bounded by hyperplanes. In contrast, we will here show that the corresponding regions for a recurrent operator network have algebraic varieties as their boundaries and within each region, the network is

a polynomial operator function. To make this description precise, we introduce and motivate several definitions.

Definition 2.3. An *operator polynomial* of degree d on $\mathbb{R}^{n\times n}$ is a function

$$P\colon \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$$

defined as

$$P(X) = \big(A_{00} + A_{10}XA_{11} + A_{20}XA_{21}XA_{22} + \ldots + A_{d0}X\ldots XA_{dd}\big) \tag{62}$$

$$= A_{00} + \sum_{j=1}^{d} A_{j0} \prod_{k=1}^{j} (XA_{jk}). \tag{63}$$

with matrix-valued coefficients $A_{ij} \in \mathbb{R}^{n\times n}$.

The definition of an operator polynomial generalizes the usual definition of a polynomial $\mathbb{R} \to \mathbb{R}$, and is equivalent when $n = 1$. We will prove in Theorem 2.4 that locally, all operator recurrent networks behave like operator polynomials. This is analogous to the result that locally, all deep rectifier networks behave like linear functions.

We next introduce the concept of a *polynomial region*. To motivate this definition, let us recall that in an operator recurrent network we have activation function terms of the form

$$\phi_\ell\big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big), \tag{64}$$

where ϕ_ℓ is a leaky rectifier activation. Then the first vector component of this expression is equal to

$$\begin{cases} \big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big)_1, & \big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big)_1 > 0, \\ \eta\big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big)_1, & \big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big)_1 \le 0. \end{cases} \tag{65}$$

Therefore, the activation function partitions the first vector component of the output into two regions, depending on the sign of $(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell)_1$. If we assume that h_ℓ is a continuous function of X, then the resulting output above will also be continuous in X, and therefore the boundary between these two regions is given by

$$\big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big)_1 = 0 \tag{66}$$

under the assumption that the two regions are nonempty and the quantity in (66) does not vanish identically in an open set. This is expected behavior for all neural networks using rectifier activations. In the case of operator recurrent networks, however, this partition is highly nonlinear due to the presence of a multiplication term. Assume that $h_\ell = Q(X)v$, where $Q(X)$ is an operator polynomial and $v \in \mathbb{R}^n$ is a vector. Then one

can observe that $b_\theta^{\ell,1} + B_\theta^\ell X h_\ell$ can also be written as a polynomial $P(X)v$, with P having degree one higher than Q. Thus, the boundaries separating the regions of the output of an activation function in an operator recurrent network are subsets of zero sets of multivariate polynomials (such sets are also called *algebraic varieties*). These observations motivate the following definition and theorem.

Definition 2.4. A *polynomial region* is an open subset $U \subset \mathbb{R}^{n\times n}$ such that for any boundary point $x_0 \in \partial U$, there exists an open set V containing x_0, a finite index set J, operator polynomials P_j and vectors $v_j \in \mathbb{R}^n$ for $j \in J$, such that

$$V \cap U = \big\{X \in V : (P_j(X)v_j)_1 > 0 \text{ for all } j \in J\big\}. \tag{67}$$

Remark 8. Since we can always compose with a permutation matrix, the coordinate index 1 can be replaced by another index without loss of generality, for example, $(P_j(X)v_j)_k > 0$ for any k.

The set $\{X \in \mathbb{R}^{n\times n} : (P(X)v)_1 = 0\}$, if nonempty, is a submanifold of codimension 1 in $\mathbb{R}^{n\times n}$, since the map $X \mapsto (P(X)v)_1$ can be viewed as a real multivariate polynomial $\mathbb{R}^{n^2} \to \mathbb{R}$. Thus, we can consider a polynomial region as being a high-dimensional generalization of a domain in Euclidean space bounded between a collection of polynomial surfaces. As mentioned above, in operator recurrent networks activation functions partition the output space nonlinearly according to zero sets of polynomials. The partitions are precisely described by the polynomial regions defined above. The analogous behavior in standard deep rectifier networks appears in the form of simplices, which originate from activation functions partitioning the output space along hyperplanes. We have

Theorem 2.4. *Let f_θ be an operator recurrent network on $\mathbb{R}^{n\times n}$ with layerwise outputs h_ℓ, $\ell = 0, \dots, L$. Then, for each ℓ, there exists a countable collection of polynomial regions $\{U_i^\ell\}$ in $\mathbb{R}^{n\times n}$ satisfying:*

1. *This collection partitions $\mathbb{R}^{n\times n}$; that is, $U_i^\ell \cap U_j^\ell = \emptyset$ for every $i \neq j$, and $\bigcup \overline{U_i^\ell} = \mathbb{R}^{n\times n}$.*
2. *Every open ball $B \subset \mathbb{R}^{n\times n}$ only nontrivially intersects U_i^ℓ for finitely many i.*
3. *The restriction of h_ℓ to each U_i^ℓ is an operator polynomial of degree at most ℓ, applied to h_0.*

Proof. The result of the theorem characterizes f_θ as a piecewise operator polynomial whose domain is partitioned into polynomial regions, on each of which f_θ is exactly an operator polynomial. Since operator polynomials form a vector space, then this characterization is also closed under addition and scalar multiplication. In particular, if f_θ and g_θ are two such functions, then any linear combination of the two functions

also satisfies the result of the theorem, except with a new partition of polynomial regions which is the mutual refinement of those of each of the original two functions. Therefore, to prove this theorem, it suffices to consider a slightly simplified version of an operator recurrent network, in which

$$f_\theta(X) = h_L, \tag{68}$$

$$h_{\ell+1}(X) = b_\theta^{\ell,0} + \phi_\ell\big(b_\theta^{\ell,1} + B_\theta^\ell X h_\ell\big), \tag{69}$$

where $X \in \mathbb{R}^{n\times n}$ is the input, $h_0 \in \mathbb{R}^n$ is given, and ϕ_ℓ is a leaky rectifier activation function with $\eta > 0$. The network given in (69) is derived by taking (10) and setting several of the weight matrices to zero. This is done to highlight the fact that the nonlinearity is derived by the matrix-vector multiplication term $B_\theta^\ell X h_\ell$. By our above argument, if the theorem holds for this simplified version, then since the general form of an operator recurrent network in (15) is merely a sum of terms of the simplified form, then the result will hold in general.

On this simplified case, we proceed by induction, and in our inductive step we construct a new collection of polynomial regions based on the previous collection of polynomial regions. For the base case, the result of the theorem holds for $\ell = 0$, since the output of the neural network is h_0, which is independent of X. Now, for the induction, suppose the claim is true at output layer ℓ. Then there exists some collection of polynomial regions $\{U_m^\ell\}$ that partitions $\mathbb{R}^{n\times n}$ (that is, disjoint sets such that the union of their closures is $\mathbb{R}^{n\times n}$), such that for any given region U_m^ℓ and for every $X \in U_m^\ell$, $h_\ell(X)$ is expressible as an operator polynomial $P_{\ell,m}(X)$ as applied to h_0, that is,

$$h_\ell(X) = P_{\ell,m}(X) := A_{00}h_0 + \sum_{i=1}^{\ell} A_{i0}\Bigg[\prod_{j=1}^{i}(XA_{ij})\Bigg]h_0, \tag{70}$$

where A_{ij} are the matrix-valued polynomial coefficients of $h_\ell(X)$ in the region U_m^ℓ. Now we apply the iteration (69) to produce the next layer. We first construct the regions and then prove that these partition the matrix space and are polynomial regions. We define

$$D_{1,j}^\ell = \big\{X \in \mathbb{R}^{n\times n} : \big(b_\theta^{\ell,1} + B_\theta^\ell X P_{\ell,m}(X)\big)_j > 0\big\}, \tag{71}$$

$$D_{2,j}^\ell = \operatorname{int}\big((D_{1,j}^\ell)^c\big), \tag{72}$$

meaning that $D_{2,j}^\ell$ is the interior of the complement of $D_{1,j}^\ell$ in $\mathbb{R}^{n\times n}$. Note that a polynomial $P\colon \mathbb{R}^{n\times n} \to \mathbb{R}$, or more generally, a real-analytic function, cannot vanish in an open set unless it is identically zero (see [81]). Thus, if $D_{1,j}^\ell$ is nonempty but also not all of $\mathbb{R}^{n\times n}$, then

$$D_{2,j}^\ell = \big\{X \in \mathbb{R}^{n\times n} : \big(-b_\theta^{\ell,1} - B_\theta^\ell X P_{\ell,m}(X)\big)_j > 0\big\}. \tag{73}$$

We note that $D^{\ell}_{1,j}, D^{\ell}_{2,j}$ are polynomial regions for every ℓ, j. The significance of these regions is that they are formed by the application of the rectifier activation function ϕ_ℓ. We aim to show that if at the inductive step, $h_\ell(X)$ is a piecewise operator polynomial on the partition $\{U^{\ell}_m\}$, then we can use the regions $D^{\ell}_{1,j}, D^{\ell}_{2,j}$ to produce a refinement $\{U^{\ell+1}_m\}$ that satisfies the theorem for the case $\ell + 1$.

To this end, we explicitly construct the new collection of polynomial regions $\{U^{\ell+1}_m\}$, checking that they are indeed polynomial regions. We define this collection of subsets as the collection of all such nonempty sets $U^{\ell+1}_m$ that can be written as

$$U^{\ell+1}_m = U^{\ell}_{m'} \cap \bigcap_{j=1}^{n} D^{\ell}_{k_j,j} \tag{74}$$

for some index m' and some $k_j \in \{1, 2\}$. The collection of sets in (74) are thus a refinement of the original partition $\{U^{\ell}_m\}$, where the refinement is produced by intersecting with the sets $D^{\ell}_{1,j}, D^{\ell}_{2,j}$.

It may be useful to observe that computing the j-th vector component of the next layer gives for $X \in U^{\ell}_{m'}$

$$(h_{\ell+1}(X))_j = \begin{cases} (b^{\ell,0}_\theta)_j + \left(b^{\ell,1}_\theta + B^{\ell}_\theta X P_{\ell,m}(X)\right)_j, & X \in D^{\ell}_{1,j} \cap U^{\ell}_{m'}, \\ (b^{\ell,0}_\theta)_j + \eta\left(b^{\ell,1}_\theta + B^{\ell}_\theta X P_{\ell,m}(X)\right)_j, & X \in D^{\ell}_{2,j} \cap U^{\ell}_{m'}. \end{cases} \tag{75}$$

Having given our explicit construction of the new partition $\{U^{\ell+1}_m\}$, we now must verify that they satisfy the conditions in the statement of the theorem. In particular, we must show that the collection is a finite partition of the domain, that each member is a polynomial region, and that $h_{\ell+1}$ restricted to each such region is an operator polynomial. First we check that this new set $\{U^{\ell+1}_m\}$ partitions the space. Note that

$$D^{\ell}_{1,j} \cap D^{\ell}_{2,j} = \emptyset,$$

and furthermore

$$\overline{D^{\ell}_{1,j}} \cup \overline{D^{\ell}_{2,j}} = \mathbb{R}^{n\times n}.$$

Since $\{D^{\ell}_{1,j}, D^{\ell}_{2,j}\}$ partitions $\mathbb{R}^{n\times n}$, each element $U^{\ell+1}_m$ is constructed by picking one element from each of $n + 1$ different partitions of $\mathbb{R}^{n\times n}$ and taking their intersection. Then it is clear that any two such sets have empty intersection, and

$$\bigcup_m \overline{U^{\ell+1}_m} \subseteq \bigcup \left(\overline{U^{\ell}_{m'}} \cap \bigcap_j \overline{D^{\ell}_{k_j,j}}\right) = \mathbb{R}^{n\times n} \cap \bigcap_{j,k_j} \overline{D^{\ell}_{k_j,j}} = \mathbb{R}^{n\times n}. \tag{76}$$

Furthermore, since the $D^{\ell}_{k,j}$ are finite, and any open ball only finitely intersects $\{U^{\ell}_m\}$ by induction hypothesis, then the same must hold of $\{U^{\ell+1}_m\}$.

Next we show that each set $U_m^{\ell+1}$ is a polynomial region. For any given m let $x \in \partial U_m^{\ell+1}$. Since $U_m^{\ell+1}$ can be expressed by (74), then there are indices m', k_j such that

$$X \in \partial U_{m'}^{\ell} \cup \bigcup_j \partial D_{k_j,j}^{\ell}. \tag{77}$$

Since $U_{m'}^{\ell}$ and $D_{k_j,j}^{\ell}$ are polynomial regions, then there exists a finite collection of open sets containing x satisfying the polynomial region definition (67) for each of the sets $U_{m'}^{\ell}$ and $D_{k_j,j}^{\ell}$. Therefore, taking the intersection of these open sets yields a new open set satisfying the conditions for (67) for the set $U_m^{\ell+1}$.

Lastly, we check that $h_{\ell+1}$ is an operator polynomial applied to h_0 when restricted to each such set. Suppose $h_{\ell+1}$ is restricted to one such polynomial region $U_m^{\ell+1}$. Using the index notation m' and k_j from the decomposition (74), define a vector $b \in \mathbb{R}^n$ by $b = (b_j)_{j=1}^n$ and

$$b_j = (b_\theta^{\ell,0})_j + \gamma_j (b_\theta^{\ell,1})_j, \tag{78}$$

where

$$\gamma_j = \begin{cases} 1 & \text{for } k_j = 1, \\ \eta & \text{for } k_j = 2. \end{cases} \tag{79}$$

Similarly, we define a matrix $B \in \mathbb{R}^{n\times n}$ by $B = (B_{ij})$ and

$$B_{ij} = \gamma_j (B_\theta^{\ell})_{ij}. \tag{80}$$

Then, restricted to $X \in U_m^{\ell+1}$, we can write

$$h_{\ell+1}(X) = b + BX P_{\ell,m}(X). \tag{81}$$

Combining the above with the induction hypothesis, it is clear then that $h_{\ell+1}(X)$ can be expressed as an operator polynomial applied to h_0 when restricted to each $U_m^{\ell+1}$. ∎

The polynomial regions that emerge from an operator recurrent network can have very nonlinear boundaries and thus have a much more complicated geometry compared to the linear regions in standard rectifier networks. In particular, because each polynomial region can be bounded by a number of high-degree polynomial submanifolds, they can be highly irregular and highly nonconvex. This behavior enables the resulting networks to potentially approximate highly nonlinear functions with fewer layers compared with traditional rectifier networks, which must approximate nonlinear behavior through piecewise linear behavior. However, due to this additional complexity, it is nonetheless likely best to employ these networks for nonlinear problems that naturally have operator polynomial or operator analytic behavior, such as hyperbolic inverse problems.

We should also note that since every operator recurrent network has a rectifier network as a special case, then by utilizing the results of [69], we can construct a particular operator recurrent network of depth L and width n that possesses at least $2^{(L+1)n/2}$ distinct polynomial regions. Thus, the expressivity of the network, as measured by the size of the polynomial region partition, increases exponentially with depth L. In the example of representing matrix inversion, the expressivity on a special set of real symmetric matrices is detailed in Theorem 6.3.

3. Regularization function and basic estimates

In this section, we will introduce a sparsity promoting regularization function. This function will later be used as a penalty term in optimization and is employed in training a network; it naturally arises in the analysis of inverse boundary value problems such as the one presented in Section 7 where weight matrices, $A^{\ell,i,(1)}$ and $B^{\ell,i,(1)}$ correspond to compact operators that are in a Schatten class. The regularization yields essentially improved generalization bounds in the later analysis. We note that regularization nowadays is commonly incorporated through the choice of method for nonconvex optimization [50].

3.1. Convex regularizing function

We introduce convex regularization functions, all denoted by $\mathcal{R}$ that, with a slight abuse of notations, are given by

$$\begin{aligned}\mathcal{R}(\theta) &= \frac{1}{2}\sum_{(\ell,i,p)\in P_1} \|\theta_p^{\ell,i}\|_{\mathbb{R}^n},\\ \mathcal{R}(\theta^\ell) &= \frac{1}{2}\sum_{(i,p)\in I^\ell} \|\theta_p^{\ell,i}\|_{\mathbb{R}^n},\\ \mathcal{R}(\theta^{\ell,i}) &= \frac{1}{2}\sum_{p\in I^{\ell,i}} \|\theta_p^{\ell,i}\|_{\mathbb{R}^n},\end{aligned} \tag{82}$$

where $I^\ell = \{(i,p) : \exists(\ell,i,p) \in P_1\}$ and $I^{\ell,i} = \{p : \exists(\ell,i,p) \in P_1\}$ and θ is a set of parameters for neural network f_θ; the index notation was introduced in (24) with $i = 0, 1$ and $P_1 \subset P$ the index set (25) corresponding to the weight matrices. We will use this function as part of an explicit regularization. *The function $\mathcal{R}$ measures the sum of the Schatten seminorms of the learnable weight matrices of the network*, which we will show below.

We consider the value of $\mathcal{R}(\theta)$ for a neural network f_θ when the matrices

$$A^{\ell,i,(1)}\colon \mathbb{R}^n \to \mathbb{R}^n \quad \text{and} \quad B^{\ell,i,(1)}\colon \mathbb{R}^n \to \mathbb{R}^n$$

satisfy

$$A_\theta^{\ell,i,(1)}, B_\theta^{\ell,i,(1)} \in \mathcal{S}_q, \tag{83}$$

where $\mathcal{S}_q$ is the Schatten q class of matrices; here, $q = 1/2$. The Schatten seminorm q, denoted by $\|\cdot\|_{\mathcal{S}_q}$, is the ℓ^q-seminorm of the vector of singular values of a matrix. We note that for $0 < q < 1$ the ℓ^q-seminorms $\|\cdot\|_q$ are not norms but satisfy

$$\|x + y\|_q^q \le \|x\|_q^q + \|y\|_q^q.$$

If $A^{\ell,i,(1)}$ is an $n \times n$ matrix with singular values $\sigma_p^{\ell,i}$ and corresponding singular vectors $u_p^{\ell,i}, v_p^{\ell,i}$ then we can choose parameters (cf. (21)–(22))

$$\theta_{2p-1}^{\ell,i} = (\sigma_p^{\ell,i})^{1/2} u_p^{\ell,i}, \quad \theta_{2p}^{\ell,i} = (\sigma_p^{\ell,i})^{1/2} v_p^{\ell,i}, \tag{84}$$

so that

$$A_\theta^{\ell,i,(1)} = \sum_{p=1}^{n} \theta_{2p-1}^{\ell,i} (\theta_{2p}^{\ell,i})^T = \sum_{p=1}^{n} \sigma_p^{\ell,i} u_p^{\ell,i} (v_p^{\ell,i})^T. \tag{85}$$

We note that the singular values $\sigma_p^{\ell,i}$ are bounded by the norm of $A^{\ell,i,(1)}$ and that the singular vectors $u_p^{\ell,i}$ and $v_p^{\ell,i}$ are orthonormal vectors. We also note that generally the vectors $\theta_p^{\ell,i}$ that parametrize the neural network are not assumed to be orthonormal, but it is possible to choose those to be parallel to the orthogonal vectors that define the singular value decompositions of the weight matrices. Moreover, we have

$$\sum_{i=0}^{1} \sum_{p=1}^{n} \left(\|\theta_{2p-1}^{\ell,i}\|_{\mathbb{R}^n} + \|\theta_{2p}^{\ell,i}\|_{\mathbb{R}^n} \right) = \sum_{i=0}^{1} \sum_{p=1}^{n} 2(\sigma_p^{\ell,i})^{1/2} = 2 \sum_{i=0}^{1} \|A_\theta^{\ell,i,(1)}\|_{\mathcal{S}_{1/2}}^{1/2}. \tag{86}$$

A similar analysis applies to $B_\theta^{\ell,i,(1)}$, $i = 0, 1$ with $p = n + 1, \dots, 2n$. Thus, the function $\mathcal{R}$ measures the sum of the $\mathcal{S}_{1/2}$ seminorms of the matrices of the network as announced above.

Furthermore, we observe that when $\|\theta_p^{\ell,i}\|_{\mathbb{R}^n} \le 1$ for all p, we have

$$\begin{aligned} \|A_\theta^{\ell,i,(1)}\|_{\mathbb{R}^n \to \mathbb{R}^n} + \|B_\theta^{\ell,i,(1)}\|_{\mathbb{R}^n \to \mathbb{R}^n} &\le \sum_{p=1}^{2n} \|\theta_{2p-1}^{\ell,i}\|_{\mathbb{R}^n} \cdot \|\theta_{2p}^{\ell,i}\|_{\mathbb{R}^n} \\ &\le \frac{1}{2} \sum_{p=1}^{2n} \left(\|\theta_{2p-1}^{\ell,i}\|_{\mathbb{R}^n} + \|\theta_{2p}^{\ell,i}\|_{\mathbb{R}^n} \right) \\ &\le \mathcal{R}(\theta^{\ell,i}). \end{aligned} \tag{87}$$

We will use this estimate in the proof of Lemma 3.1 below.

3.2. Truncated network

For our later generalization results, it is important to guarantee that the output of any given network is bounded. Indeed, our goal is to construct an operator recurrent network $f_\theta: \mathcal{B}^{n\times n} \to \mathbb{R}^n$, where $\mathcal{B}^{n\times n} = \mathcal{B}^{n\times n}(1) = \{X \in \mathbb{R}^{n\times n} : \|X\|_{\mathbb{R}^n\to\mathbb{R}^n} \leq 1\}$ is the closed unit ball in the set of matrices, that approximates a bounded, continuous function $f: \mathcal{B}^{n\times n} \to \mathbb{R}^n$. As we know a priori that the function we approximate is bounded by

$$\|f\|_\infty = \|f\|_{L^\infty(\mathcal{B}^{n\times n};\mathbb{R}^n)} = \sup_{X\in\mathcal{B}^{n\times n}} \|f(X)\|_{\mathbb{R}^n}, \tag{88}$$

we can add to the network f_θ two additional layers that cut off any coordinates values that are too large. That is, we introduce a new parameter, $m \in \mathbb{R}_+$, satisfying

$$m \geq \|f\|_\infty \tag{89}$$

and add two layers that implement the function

$$T_m: \mathbb{R}^n \to \mathbb{R}^n,$$

where, for $x = (x_j)_{j=1}^n \in \mathbb{R}^n$,

$$T_m(x) = -b + \phi_0(b + y), \quad y = b - \phi_0(b - x), \tag{90}$$

where $b = (m, m, \dots, m)^T \in \mathbb{R}^n$, and ϕ_0 is the standard rectifier function "ReLU". We note that then $T_m(x) = (T_m(x_j))_{j=1}^n$, where $T_m(x_j) = \max(-m, \min(x_j, m))$.

Definition 3.1. We say that $\bar{f}_\theta: \mathcal{B}^{n\times n} \to \mathbb{R}^n$ is a *truncated basic or, respectively, a truncated general) operator recurrent network* of depth $L + 2$ and width n if

$$\bar{f}_\theta = T_m \circ f_\theta, \tag{91}$$

where T_m is of the form (90) and f_θ is a basic (or, respectively, general) operator recurrent network with depth L and width n.

Truncated neural networks make it possible to effectively use Hoeffding's inequality in studying their generalization properties. Below, we use that for a truncated general operator recurrent network $\bar{f}_\theta$ we have

$$\|\bar{f}_\theta\|_{L^\infty(\mathcal{B}^{n\times n};\mathbb{R}^n)} \leq n^{1/2} m \tag{92}$$

and as the map T_m has Lipschitz constant 1, the Lipschitz constant of $\theta \mapsto T_m(\widetilde{f}_\theta(X))$ is bounded by the Lipschitz constant of $\theta \mapsto \widetilde{f}_\theta(X)$. We note that in (92) the factor $n^{1/2}$ appears due to the fact that we use the Euclidean norm $\|\cdot\|_2$ in $\mathbb{R}^n$. If the norm of $x = (x_j)_{j=1}^n \in \mathbb{R}^n$ is replaced by the norm $\|x\|_{\max} = \|x\|_\infty = \max_j |x_j|$,

that is, if we replace the Euclidean space $(\mathbb{R}^n, \|\cdot\|_{\mathbb{R}^n})$ by $(\mathbb{R}^n, \|\cdot\|_{\max})$ and use $m = \sup_{X\in\mathcal{B}^{n\times n}} \|f(X)\|_{\max}$, we obtain

$$\sup_{X\in\mathcal{B}^{n\times n}} \|\bar{f}_\theta(X)\|_{\max} \leq m. \tag{93}$$

3.3. Intermediate function and regularization determining the loss functions

Here, we assume that the network $\bar{f}_\theta$ is a truncated basic recurrent operator network that satisfies (92). To guarantee a generalization error bound, one needs to avoid the problem of over fitting, in which $\bar{f}_\theta$ accurately approximates $f\colon \mathbb{R}^{n\times n} \to \mathbb{R}^n$ on the training set S, but poorly approximates f away from S. To this end, we introduce a regularizing penalty term using $\mathcal{R}$.

Definition 3.2. For parameter θ and the pair (X, y), we let $\mathcal{L}$ be given by

$$\mathcal{L}(\theta, X, y) = \|\bar{f}_\theta(X) - y\|^2_{\mathbb{R}^n}. \tag{94}$$

Moreover, we let $\mathcal{L}_r$ with regularization parameter $\alpha \in (0, 1)$ be given by

$$\mathcal{L}_r(\theta, X, y) = \|\bar{f}_\theta(X) - y\|^2_{\mathbb{R}^n} + \alpha\mathcal{R}(\theta). \tag{95}$$

We denote by Θ the set of all parameters θ that the weight matrices of the network $\bar{f}_\theta$ depend upon; more precisely

$$\Theta = \Theta_{(L)} = \big\{(\theta_p^{\ell,i})_{(\ell,i,p)\in P} \in (\mathbb{R}^n)^{\#P} : \|\theta_p^{\ell,i}\|_{\mathbb{R}^n} \leq 1\big\}, \tag{96}$$

where P is the index set (25). The regularization term shows up explicitly and independently in the estimate for the Lipschitz constant of the network as well as in the estimate for its derivatives with respect to the weights in P_1 in the next subsection. Also, for the general recurrent operator neural networks we denote the parameter space by

$$\widetilde{\Theta} = \widetilde{\Theta}_{(L,K)} = \big\{(\tilde{\theta}_p^{\ell,i,k})_{(\ell,i,p,k)\in\widetilde{P}} \in (\mathbb{R}^n)^{\#\widetilde{P}} : \|\theta_p^{\ell,i,k}\|_{\mathbb{R}^n} \leq 1\big\}. \tag{97}$$

3.4. Basic estimates of the recurrent operator neural network

3.4.1. Derivative with respect to weights. We show how controlling the norms of the parameter θ provides an upper bound on directional derivatives in a local neighborhood for the neural network f_θ, given by (9)–(10), as a function of θ. Such a bound is crucial to controlling the behavior of the neural network during training. The key intuition here is that estimates of the derivative, which also give upper bounds on the local Lipschitz constant of $\theta \to f_\theta(X)$, provide some knowledge concerning the behavior of the regularized loss function in a neighborhood of its minimum.

In the lemma below we consider a basic operator recurrent network. Recall from (21) that the weight matrices $A_\theta^{\ell,k}$ and $B_\theta^{\ell,k}$ depend quadratically on the column vectors contained within any parameter set θ. This lemma generalizes easily to general operator recurrent networks and is used in the proofs of Theorem 5.2 and Lemma 5.3.

Lemma 3.1. *Let $f_\theta\colon \mathbb{R}^{n\times n} \to \mathbb{R}^n$ be a basic operator recurrent network with leaky rectifier activations, and h_0 satisfy $\|h_0\| \le 1$. Let $\|X\| \le 1$. Then, for $(\ell, i, p) \in P_1$, see (25), the local Lipschitz constant (or the derivative, if it exists) of $f_\theta(X)$ with respect to $\theta_p^{\ell,i}$ is bounded by $K_p^{\ell,i}$ with*

$$K_p^{\ell,i} \le c_0^{L+1} \|\theta_{(p)'}^{\ell,i}\| \exp(\mathcal{R}(\theta)), \tag{98}$$

where $(p)' = p+1$, if p is odd and $(p)' = p-1$, if p is even. For $(\ell, i, p) \in P_2$, see (25), the derivative of $f_\theta(X)$ with respect to $\theta_p^{\ell,i}$ is bounded by $K_p^{\ell,i}$ with

$$K_p^{\ell,i} \le c_0^{L+1} \exp(\mathcal{R}(\theta)), \tag{99}$$

that is, $\operatorname{Lip}(\theta_p^{\ell,i} \to f_\theta(X)) \le c_0^{L+1} \exp(\mathcal{R}(\theta))$ for all $(\ell, i, p) \in P$.

Proof. In the proof we estimate the derivatives of the output of ℓ-th layer and the results are combined in a way that is analogous to the back propagation algorithm. We consider $(\ell, i, p) \in P_1$; that is, we consider derivatives with respect to parameters that determine the weight matrices.

To compute $K_p^{\ell,i}$ we differentiate using the chain rule. We consider the intermediate outputs by h_ℓ. At every point θ, where h_ℓ and $f_\theta(X)$ are differentiable with respect to θ, we have for $\ell' > \ell$

$$\left\| \frac{\partial h_{\ell'}}{\partial \theta_p^{\ell,i}} \right\| \le \left\| \frac{\partial h_{\ell'-1}}{\partial \theta_p^{\ell,i}} \right\| \big(\|A_\theta^{\ell',0}\| + \|B_\theta^{\ell',0}\| + \|A_\theta^{\ell',1}\| + \|B_\theta^{\ell',1}\| \big). \tag{100}$$

Since h_L, the output at layer L, is the same as $f_\theta(X)$, then iterating the above starting from $\ell' = L$ down to ℓ, we obtain

$$\left\| \frac{\partial f_\theta(X)}{\partial \theta_p^{\ell,i}} \right\| \le \left\| \frac{\partial h_\ell}{\partial \theta_p^{\ell,i}} \right\| \prod_{\ell'=\ell+1}^{L} \big(\|A_\theta^{\ell',0}\| + \|B_\theta^{\ell',0}\| + \|A_\theta^{\ell',1}\| + \|B_\theta^{\ell',1}\| \big). \tag{101}$$

We recall that the largest singular value $\sigma_1(A)$ of a matrix $A \in \mathbb{R}^{n\times n}$ satisfies

$$\|A\|_{\mathbb{R}^n \to \mathbb{R}^n} \le \sigma_1(A) \le \|A\|_{s_{1/2}}.$$

Using (87), we find that

$$\|A_\theta^{\ell,i,(1)}\|_{\mathbb{R}^n \to \mathbb{R}^n} + \|B_\theta^{\ell,i,(1)}\|_{\mathbb{R}^n \to \mathbb{R}^n} \le \mathcal{R}(\theta^{\ell,i}). \tag{102}$$

With this inequality, we relate the matrix norms $\|A_\theta^{\ell',i}\|, \|B_\theta^{\ell',i}\|$ to the regularization terms $\mathcal{R}(\theta^{\ell',k})$. By our assumptions

$$\sum_{i=1}^{2} \|A_\theta^{\ell',i}\| + \|B_\theta^{\ell',i}\| \le \sum_{i=1}^{2} \|A^{\ell',i,(0)}\| + \|A_\theta^{\ell',i,(1)}\| + \|B^{\ell',i,(0)}\| + \|B_\theta^{\ell',i,(1)}\|$$
$$\le c_0 + \sum_{i=1}^{2} \|A_\theta^{\ell',i,(1)}\| + \|B_\theta^{\ell',i,(1)}\| \le c_0 + \mathcal{R}(\theta^{\ell'}), \quad (103)$$

cf. (23). We find that

$$\left\| \frac{\partial f_\theta(X)}{\partial \theta_p^{\ell,i}} \right\| \le \left\| \frac{\partial h_\ell}{\partial \theta_p^{\ell,i}} \right\| \prod_{\ell'=\ell+1}^{L} (c_0 + \mathcal{R}(\theta^{\ell'}))$$
$$\le c_0^{L-\ell} \left\| \frac{\partial h_\ell}{\partial \theta_p^{\ell,i}} \right\| \exp\left(\sum_{\ell'=\ell+1}^{L} \mathcal{R}(\theta^{\ell'}) \right), \quad (104)$$

where we used the simple inequality $c_0 + x \le c_0 e^x$ for $x \ge 0$. Viewing $\theta_p^{\ell,i}$ as a column vector,

$$\left\| \frac{\partial h_\ell}{\partial \theta_p^{\ell,i}} \right\| \le \|h_{\ell-1}\| \|\theta_{(p)'}^{\ell,i}\|, \quad (105)$$

where $(p)' = p+1$, if p is odd and $(p)' = p-1$, if p is even and the weight matrices are written in terms of the parameters as a sum of rank-1 matrices as given in (21). This means that every column vector $\theta_p^{\ell,i}$ is "paired" with an adjacent column vector, thus justifying (105). For h_ℓ we find in a similar fashion that

$$\|h_\ell\| \le \|b_\theta^{\ell,0}\| + \|b_\theta^{\ell,1}\| + \big(\|A_\theta^{\ell,0}\| + \|B_\theta^{\ell,0}\| + \|A_\theta^{\ell,1}\| + \|B_\theta^{\ell,1}\|\big)\|h_{\ell-1}\|$$
$$\le \big(c_0 + \|b_\theta^{\ell,0}\| + \|b_\theta^{\ell,1}\| + \|A_\theta^{\ell,0,(1)}\| + \|B_\theta^{\ell,0,(1)}\| + \|A_\theta^{\ell,1,(1)}\| + \|B_\theta^{\ell,1,(1)}\|\big)$$
$$\cdot \big(1 + \|h_{\ell-1}\|\big). \quad (106)$$

Here, when $b_\theta^{\ell,i} = \theta^{\ell,i}$ and

$$A_\theta^{\ell,i,(1)} = \sum_{p=1}^{n} \theta_{2p-1}^{\ell,i} (\theta_{2p}^{\ell,i})^T, \quad B_\theta^{\ell,i,(1)} = \sum_{p=n+1}^{2n} \theta_{2p-1}^{\ell,i} (\theta_{2p}^{\ell,i})^T,$$

and (23) is satisfied, we find that as in (102) and (103)

$$c_0 + \|\theta^{\ell,0}\| + \|\theta^{\ell,1}\| + \|A_\theta^{\ell,0,(1)}\| + \|B_\theta^{\ell,0,(1)}\| + \|A_\theta^{\ell,1,(1)}\| + \|B_\theta^{\ell,1,(1)}\|$$
$$\le c_0 + \mathcal{R}(\theta^\ell), \quad (107)$$

where we recall that $c_0 \geq 1$. As the initial vector h_0 is in the closed unit ball, using the above and that $x \leq e^x$ and $c_0 + x \leq c_0 e^x$, it follows that

$$\|h_\ell\| \leq c_0^\ell \exp\left(\sum_{\ell'=1}^{\ell} \mathcal{R}(\theta^{\ell'}) \right). \tag{108}$$

Using (104) and (108), we therefore find that if $K_p^{\ell,i}$ is the local Lipschitz constant of $f_\theta(X)$ in a neighborhood of θ (when considering only $\theta_p^{\ell,i}$ as a variable), then

$$\begin{aligned} K_p^{\ell,i} &\leq c_0^{L-\ell} \left\| \frac{\partial h_\ell}{\partial \theta_p^{\ell,i}} \right\| \exp\left(\sum_{\ell'=\ell+1}^{L} \mathcal{R}(\theta^{\ell'}) \right) \\ &\leq c_0^{L-\ell} \|h_{\ell-1}\| \|\theta_{(p)'}^{\ell,i}\| \exp\left(\sum_{\ell'=\ell+1}^{L} \mathcal{R}(\theta^{\ell'}) \right) \\ &\leq c_0^{L-\ell} c_0^\ell \|\theta_{(p)'}^{\ell,i}\| \exp\left(\sum_{\ell' \neq \ell} \mathcal{R}(\theta^{\ell'}) \right) \\ &\leq c_0^L \|\theta_{(p)'}^{\ell,i}\| \exp(\mathcal{R}(\theta)). \end{aligned} \tag{109}$$

This yields the claim for $p \in P_1$.

To compute derivatives with respect to bias parameters, in which case $(\ell, i, p) \in P_2$, the result follows similarly to the above by using (104) and replacing (105) by

$$\left\| \frac{\partial h_\ell}{\partial \theta_p^\ell} \right\| \leq 1. \tag{110}$$

This completes the proof. ∎

We point out that the factor c_0^{L+1} in inequality (99) that grows exponentially in L is natural as the nonlearnable parts of the weight matrices of a neural network f_θ are linear operators which norms are bounded by c_0, see (23). Hence, even when the trained parts of the weight matrices vanish, each layer of the neural network can increase the Lipschitz constant of the function f_θ by a multiplicative factor c_0.

3.4.2. Lipschitz constant in X variable. Obtaining sharp Lipschitz constants for networks is essential to assess their robustness against perturbation in their inputs. Such constants were recently derived for feed-forward neural networks in [25] using advanced tools from nonlinear analysis. Here, we provide an upper bound to the Lipschitz constant for a basic operator recurrent network. This bound will play a role in the forthcoming section on generalization.

Lemma 3.2. *Let the set of parameters or weights θ belong to Θ defined in (96). Then the Lipschitz norm of the map $X \to f_\theta(X)$ satisfies*

$$\mathrm{Lip}(f_\theta) \le L c_0^L \exp(\mathcal{R}(\theta)). \tag{111}$$

Proof. We recall that by (103),

$$\begin{aligned}\sum_{i=1}^{2} \|A_\theta^{\ell',i}\| + \|B_\theta^{\ell',i}\| &\le \sum_{i=1}^{2} \|A^{\ell',i,(0)}\| + \|A_\theta^{\ell',i,(1)}\| + \|B^{\ell',i,(0)}\| + \|B_\theta^{\ell',i,(1)}\| \\ &\le c_0 + \sum_{i=1}^{2} \|A_\theta^{\ell',i,(1)}\| + \|B_\theta^{\ell',i,(1)}\| \le c_0 + \mathcal{R}(\theta^{\ell',i}) \end{aligned} \tag{112}$$

cf. (23). As $X \in \mathcal{B}_{n\times n}$ we have $\|X\| \le 1$. In the definition of a basic recurrent operator network (cf. (9)–(10)) we introduced the notation $h_\ell = h_\ell(X) = h_\ell(X; h_{\ell-1})$.

Using (108), we obtain

$$\|h_\ell\| \le c_0^\ell \exp\Big(\sum_{\ell'=1}^{\ell} \mathcal{R}(\theta^{\ell'}) \Big). \tag{113}$$

Moreover,

$$\begin{aligned}\|h_\ell(X_1) - h_\ell(X_2)\| &= \|h_\ell(X_1; h_{\ell-1}(X_1)) - h_\ell(X_2; h_{\ell-1}(X_2))\| \\ &= \|h_\ell(X_1; h_{\ell-1}(X_1)) - h_\ell(X_2; h_{\ell-1}(X_1))\| \\ &\quad + \|h_\ell(X_2; h_{\ell-1}(X_1)) - h_\ell(X_2; h_{\ell-1}(X_2))\| \\ &\le \Big(\sum_{i=0}^{1} \|B_\theta^{\ell,i}\| \Big) \|X_1 - X_2\| \, \|h_{\ell-1}(X_1)\| \\ &\quad + \Big(\sum_{i=0}^{1} \|B_\theta^{\ell,i}\| \Big) \|X_2\| \, \|h_{\ell-1}(X_1) - h_{\ell-1}(X_2)\| \\ &\le \Big(c_0 + \sum_{i=0}^{1} \|B_\theta^{\ell,i,(1)}\| \Big) \|X_1 - X_2\| \, \|h_{\ell-1}(X_1)\| \\ &\quad + \Big(c_0 + \sum_{i=0}^{1} \|B_\theta^{\ell,i,(1)}\| \Big) \|X_2\| \, \|h_{\ell-1}(X_1) - h_{\ell-1}(X_2)\| \\ &\le \big(c_0 + \mathcal{R}(\theta^\ell)\big) \|X_1 - X_2\| \, c_0^{\ell-1} \exp\Big(\sum_{\ell'=1}^{\ell-1} \mathcal{R}(\theta^{\ell'}) \Big) \\ &\quad + \big(c_0 + \mathcal{R}(\theta^\ell)\big) \|h_{\ell-1}(X_1) - h_{\ell-1}(X_2)\| \end{aligned}$$

$$\leq \|X_1 - X_2\| \, c_0^\ell \exp\left(\sum_{\ell'=1}^{\ell} \mathcal{R}(\theta^{\ell'}) \right) + c_0 \exp\big(\mathcal{R}(\theta^\ell)\big) \|h_{\ell-1}(X_1) - h_{\ell-1}(X_2)\|.$$

We observe that $h_0(X_1) - h_0(X_2) = 0$. Using induction, we see from this that

$$\|h_\ell(X_1) - h_\ell(X_2)\| \leq \|X_1 - X_2\| \, \ell c_0^\ell \exp\left(\sum_{\ell'=1}^{\ell} \mathcal{R}(\theta^{\ell'}) \right).$$

Then the Lipschitz norm of the map $X \to f_\theta(X)$ satisfies

$$\mathrm{Lip}(f_\theta) \leq c_0^L \exp\left(\sum_{\ell'=1}^{L} \mathcal{R}(\theta^{\ell'}) \right) = L c_0^L \exp(\mathcal{R}(\theta)). \tag{114}$$

This completes the proof. ∎

Remark 9. We observe that the Lipschitz constant of a truncated neural network $\overline{f}_\theta = T_m \circ f_\theta$ satisfies $\mathrm{Lip}(\overline{f}_\theta) \leq \mathrm{Lip}\, f_\theta)$. This holds both with respect to the X and θ variables.

Remark 10. The Lipschitz constant grows with the number of layers L. This seemingly indicates that deeper networks are expected to generalize more poorly even though they reduce training error. The responsible factor, c_0^L, however, arises from the inequality with *nonlearnable* matrices in the network through

$$\|A^{\ell,i,(0)}\| + \|B^{\ell,i,(0)}\| + |b_\theta^{\ell,0}| + |b_\theta^{\ell,1}|$$

(cf. (23)) with $c_0 \geq 1$.

4. Deep learning and inverse problem from a common statistical viewpoint

4.1. Formulation

The learning problem is finding an approximation to a continuous nonlinear operator function f by an operator recurrent network $\overline{f}_\theta$ given training data. We recall that the inverse operator had the form $H = g \circ \boldsymbol{f}$; that is, H stands for F^{-1} on its range, $\mathcal{X}$. As laid out in the introduction,

$$z = g\big(f^1(X), f^2(X), \ldots, f^T(X)\big), \tag{115}$$

A key objective is to train recurrent operator networks f_θ^t that approximate the functions f^t. To simplify notations, we below drop the index t and consider just one function f.

4.1.1. Definitions of random variables. We let $(\Omega, \Sigma, \mathbb{P})$ be a complete probability space and let $\mathbf{z}\colon \Omega \to \mathbb{R}^m$ be a random variable that models a random object, and $\mathbf{X} = F(\mathbf{z})$ be a random variable modeling the noiseless measurement obtained from the object $\mathbf{z}$ and $\mathbf{y}$ be a random variable modeling the intermediate quantity, $\mathbf{y} = f(\mathbf{X})$. We denote

$$p = f \circ F, \tag{116}$$

so that $\mathbf{y} = p(\mathbf{z})$. Next, we add the measurement error $\mathcal{E}$ to the noiseless measurement. To this end, we consider a random variable $\mathcal{E}\colon \Omega \to \mathbb{R}^{n\times n}$ having the variance

$$\mathbb{E}\|\mathcal{E}\|^2_{\mathbb{R}^n \to \mathbb{R}^n} \le \mathbb{E}\|\mathcal{E}\|^2_{\mathbb{R}^{n^2}} = n^2\sigma^2. \tag{117}$$

Here, $\|\mathcal{E}\|^2_{\mathbb{R}^{n^2}} = \sum_{i,j=1}^n |\mathcal{E}_{ij}|^2$ is the square of the Frobenius norm of the matrix $\mathcal{E}$. The noisy measurement is then defined to be

$$\mathbf{M} = \mathbf{X} + \mathcal{E}. \tag{118}$$

This defines a random variable $\mathbf{M}\colon \Omega \to \mathbb{R}^{n\times n}$. We assume that $\mathbf{X}$ and $\mathcal{E}$ are independent.

We denote by $\pi_{\mathbf{z}}$ the distribution of $\mathbf{z}$, by $\pi_{\mathbf{X}} = F_*\pi_{\mathbf{z}}$ the distribution of $\mathbf{X}$, and by $\pi_{\mathcal{E}}$ the distribution of $\mathcal{E}$. Also, we define that

$$\begin{aligned} &\tau_0 \text{ is the distribution of the pair } (\mathbf{X}, \mathbf{y}),\\ &\tau \text{ is the distribution of the pair } (\mathbf{M}, \mathbf{y}). \end{aligned} \tag{119}$$

When $\mathfrak{F}$ is the map $z \to (F(z), p(z))$ and $\widehat{\mathfrak{F}}$ is the map $(z, \varepsilon) \to (F(z) + \varepsilon, p(z))$, then the distribution τ_0 is given by $\tau_0 = \mathfrak{F}_*\pi_{\mathbf{z}}$ and τ is given by $\tau = \widehat{\mathfrak{F}}_*(\pi_{\mathbf{z}} \times \pi_{\mathcal{E}})$.

Our aim below is to approximate the map $\boldsymbol{f}$ by a recurrent operator neural network. We assume that we are given samples of the *measurement-property* pairs that are samples of the pair $(\mathbf{M}, \mathbf{y})$. We note that adding noise $\mathcal{E}$ to the noiseless data $\mathbf{X}$ gives us noisy data $\mathbf{M}$ that may be outside the range $\mathcal{X}$ of the direct map F and hence outside the domain of definition of the map f. However, the domain of the (trained) neural network is not restricted to the range of the direct map.

To consider neural networks that are defined in a ball of radius 1, we assume that

(i) the distribution $\pi_{\mathcal{E}}$ of $\mathcal{E}$ is supported in the ball of radius $\frac{1}{2}$, that is, $\mathcal{E} \in \mathcal{B}_{n\times n}(\frac{1}{2})$ a.s.,

(ii) the distribution $\pi_{\mathbf{z}}$ of $\mathbf{z}$ is such that $F_*\pi_{\mathbf{z}}$ is supported in the ball of radius $\rho_1 = \frac{1}{2}$, that is, the noiseless measurements satisfy $\mathbf{X} = F(\mathbf{z}) \in \mathcal{B}_{n\times n}(\frac{1}{2})$ a.s.,

Under these assumptions, $\mathbf{M} = F(\mathbf{z}) + \mathcal{E} \in \mathcal{B}_{n\times n}(1)$ a.s.

4.1.2. Expected loss and regularization. Given a network with parameters θ, the expected loss for noisy and noiseless measurements are defined by

$$\mathcal{L}(\theta,\tau)=\mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big[\mathcal{L}(\theta,\mathbf{M},\mathbf{y})\big],\quad \mathcal{L}(\theta,\tau_0)=\mathbb{E}_{(\mathbf{X},\mathbf{y})\sim\tau_0}\big[\mathcal{L}(\theta,\mathbf{X},\mathbf{y})\big] \tag{120}$$

(cf. (94)) and the expected regularized loss for noisy and noise-free measurements are defined by

$$\mathcal{L}_r(\theta,\tau)=\mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big[\mathcal{L}_r(\theta,\mathbf{M},\mathbf{y})\big],\quad \mathcal{L}_r(\theta,\tau_0)=\mathbb{E}_{(\mathbf{X},\mathbf{y})\sim\tau_0}\big[\mathcal{L}_r(\theta,\mathbf{X},\mathbf{y})\big] \tag{121}$$

(cf. (95)).

We remark that many other regularizers have been studied. For example, regularizers that measure the Lipschitz norm of the neural network with respect to their inputs have been shown to give good approximations [74].

4.2. Optimal network subject to sparsity bound

First we consider the case when it is a priori known that some recurrent operator network approximates the target function f with some reasonable accuracy. We formalize this case in

Definition 4.1. We say that the function $f:\mathcal{X}\to\mathbb{R}^n$ can be approximated with accuracy ε_0, that is, in the range of F, by a neural network f_{θ_0} with sparsity bound R_0 if there is $\theta_0\in(\mathbb{R}^n)^P$ with

$$\mathcal{R}(\theta_0)=R_0, \tag{122}$$

such that the neural network $\bar{f}_{\theta_0}$ corresponding to the parameter θ_0 satisfies

$$\sup_{X\in\mathcal{X}}\|f(X)-f_{\theta_0}(X)\|_{\mathbb{R}^n}\le\varepsilon_0. \tag{123}$$

We observe that when f_{θ_0} satisfies (123), and $m=\|f\|_\infty$, then the truncated neural network, $\bar{f}_{\theta_0}=T_m\circ f_{\theta_0}$, satisfies

$$\sup_{X\in\mathcal{X}}\|f(X)-\bar{f}_{\theta_0}(X)\|_{\mathbb{R}^n}\le\varepsilon_0. \tag{124}$$

When f satisfies the assumptions of Theorem 2.1, then by Theorem 2.2 and inequality (50) we have that (123) holds with $\varepsilon_0=C'\varepsilon$ and parameters θ_0 that satisfy (122) for some value R_0. We note that below it is not necessary to assume that (122)–(123) hold. Our aim is to find a neural network $\bar{f}_\theta$ that is a better approximation of f than the neural network $\bar{f}_{\theta_0}$ considered in Definition 4.1.

4.2.1. Optimal neural network when the measurements are noise free. We introduce the following definition of θ_0^* that minimizes the regularized loss function in the noise-free case.

Definition 4.2. The parameters of the optimal network in the noise-free case, θ_0^*, are a solution of

$$\begin{aligned}\theta_0^* &= \arg\min_{\theta\in\Theta} \mathcal{L}_r(\theta,\tau_0) = \arg\min_{\theta\in\Theta} \mathbb{E}_{(\mathbf{X},\mathbf{y})\sim\tau_0}\big(\|\bar{f}_\theta(\mathbf{X})-\mathbf{y}\|^2+\alpha\mathcal{R}(\theta)\big)\\ &= \arg\min_{\theta\in\Theta} \mathbb{E}_{\mathbf{z}\sim\pi_{\mathbf{z}}}\big(\|\bar{f}_\theta(F(\mathbf{z}))-p(\mathbf{z})\|^2+\alpha\mathcal{R}(\theta)\big).\end{aligned} \tag{125}$$

Here, $\bar{f}_\theta$ are truncated basic recurrent operator networks of depth $L+2$ with truncation parameter $m \geq \|f\|_\infty$; see (89)–(90).

Below, for simplicity, we assume that

$$m = \|f\|_\infty. \tag{126}$$

In the case when we do not know the norm $\|f\|_\infty$ but are only given an upper bound m for the norm, all estimates below are valid when $\|f\|_\infty$ are replaced by m.

Remark 11. Minimizers to (125) necessarily exist because the loss function is continuous in θ and the constraint $\mathcal{R}(\theta) \leq R_0$ restricts the allowable set to a compact one. The minimizer may not be unique.

Lemma 4.1. *The optimal parameter for noise-free measurements, θ_0^*, and the noise-free expected loss satisfy*

$$\mathcal{L}_r(\theta_0^*,\tau_0) = \mathbb{E}_{(\mathbf{X},\mathbf{y})\sim\tau_0}\big(\|\bar{f}_{\theta_0^*}(\mathbf{X})-\mathbf{y}\|^2\big)+\alpha\mathcal{R}(\theta_0^*) \leq \mathcal{K}_0, \tag{127}$$

where

$$\mathcal{K}_0 := \begin{cases} \min(4n\|f\|_\infty^2, \varepsilon_0^2+\alpha R_0), & \text{if (122)–(123) hold,}\\ 4n\|f\|_\infty^2, & \text{if (122)–(123) do not hold.}\end{cases} \tag{128}$$

Proof. If (122)–(123) hold, we find that

$$\mathcal{L}_r(\theta_0,\tau_0) = \mathbb{E}_{\mathbf{X}\sim\pi_{\mathbf{X}}}\big(\|\bar{f}_{\theta_0}(\mathbf{X})-f(\mathbf{X})\|^2\big)+\alpha\mathcal{R}(\theta_0) \leq \varepsilon_0^2+\alpha R_0. \tag{129}$$

Moreover, both in the case when (122)–(123) hold or do not hold, we can take $\theta = 0$, in which case by (92) $\|\bar{f}_\theta\| \leq n^{1/2}\|f\|_\infty$ and $\mathcal{R}(\theta) = 0$ and thus

$$\begin{aligned}\mathcal{L}_r(\theta,\tau_0) &= \mathbb{E}_{\mathbf{X}\sim\pi_{\mathbf{X}}}\big(\|\bar{f}_\theta(\mathbf{X})-f(\mathbf{X})\|^2\big)+\alpha\mathcal{R}(\theta)\\ &\leq \mathbb{E}_{\mathbf{X}\sim\pi_{\mathbf{X}}}\big(\big(\|\bar{f}_\theta(\mathbf{X})\|+\|f(\mathbf{X})\|\big)^2\big)+\alpha\mathcal{R}(\theta) \qquad (130)\\ &\leq (n^{1/2}+1)^2\|f\|_\infty^2+0 \leq 4n\|f\|_\infty^2. \qquad (131)\end{aligned}$$

We conclude that $\mathcal{L}_r(\theta_0^*,\tau_0) = \min_{\theta\in\Theta}\mathcal{L}_r(\theta,\tau_0)$ satisfies (127). ∎

Next we consider how adding the measurement error $\mathcal{E}$ changes the behavior of the neural network $\bar{f}_{\theta_0^*}$. To this end, we return to considering the random variable $\mathbf{M}$.

4.2.2. Estimate of expected loss for noisy measurements. Next, we consider the expected loss in the case when measurements contain errors. We recall that adding noise to the data brings us outside the range of the direct map but the domain of the (trained) neural network is not restricted to the range of the direct map.

We introduce the notation,

$$\mathcal{L}_{r,0} := \min\left(4n\|f\|_\infty^2, 2\varepsilon_0^2 + 2\alpha R_0 + 2L^2 c_0^{2L}\exp(2\varepsilon_0^2/\alpha + 2R_0)\cdot n^2\sigma^2\right) \quad (132)$$

if (122)–(123) hold, and

$$\mathcal{L}_{r,0} = 4n\|f\|_\infty^2 \quad (133)$$

if (122)–(123) do not hold. Sometimes, to indicate the parameter α, we denote $\mathcal{L}_{r,0} = \mathcal{L}_{r,0}(\alpha)$. We also write

$$\mathcal{R}_0 := \frac{1}{\alpha}\mathcal{L}_{r,0}, \quad (134)$$

that is,

$$\mathcal{R}_0 = \frac{1}{\alpha}\min\left(4n\|f\|_\infty^2, 2\varepsilon_0^2 + 2\alpha R_0 + 2L^2 c_0^{2L}\exp(2\varepsilon_0^2/\alpha + 2R_0)\cdot n^2\sigma^2\right), \quad (135)$$

if (122)–(123) hold, and

$$\mathcal{R}_0 = \frac{1}{\alpha}4n\|f\|_\infty^2, \quad (136)$$

if (122)–(123) do not hold.

Lemma 4.2. *The optimal parameters for noise-free measurements, θ_0^*, and the noisy expected loss satisfy*

$$\mathcal{L}_r(\theta_0^*, \tau) \le \mathcal{L}_{r,0}. \quad (137)$$

Proof. First, we consider the case when (122)–(123) hold. We have

$$\begin{aligned}\mathcal{L}_r(\theta_0^*, \tau) &= \mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\left(\|\bar{f}_{\theta_0^*}(\mathbf{M}) - \mathbf{y}\|_{\mathbb{R}^n}^2\right) + \alpha\mathcal{R}(\theta_0^*)\\ &= \mathbb{E}_{(\mathbf{X},\mathcal{E})}\left(\|\bar{f}_{\theta_0^*}(\mathbf{X}+\mathcal{E}) - f(\mathbf{X})\|_{\mathbb{R}^n}^2\right) + \alpha\mathcal{R}(\theta_0^*).\end{aligned}$$

Equation (127) implies that

$$\mathcal{R}(\theta_0^*) \le \mathcal{K}_0/\alpha. \quad (138)$$

Furthermore,

$$\begin{aligned}&\mathbb{E}_{(\mathbf{X},\mathcal{E})}\left(\|\bar{f}_{\theta_0^*}(\mathbf{X}+\mathcal{E}) - f(\mathbf{X})\|_{\mathbb{R}^n}^2\right)\\ &\quad\le 2\mathbb{E}_{(\mathbf{X},\mathcal{E})}\left(\|\bar{f}_{\theta_0^*}(\mathbf{X}+\mathcal{E}) - \bar{f}_{\theta_0^*}(\mathbf{X})\|_{\mathbb{R}^n}^2\right) + 2\mathbb{E}_{\mathbf{X}}\left(\|\bar{f}_{\theta_0^*}(\mathbf{X}) - f(\mathbf{X})\|_{\mathbb{R}^n}^2\right). \quad (139)\end{aligned}$$

Using (127), the second term in (139) satisfies the inequality,

$$2\mathbb{E}_{\mathbf{X}}\big(\|\bar{f}_{\theta_0^*}(\mathbf{X}) - f(\mathbf{X})\|_{\mathbb{R}^n}^2\big) + 2\alpha\mathcal{R}(\theta_0^*) \leq 2\mathcal{K}_0. \tag{140}$$

For the first term in (139), we observe that as $\bar{f}_{\theta_0^*}\colon \mathcal{B}_{n\times n}(1) \to \mathbb{R}^n$ is in the space $C^{0,1}(\mathcal{B}_{n\times n}(1))$, we have

$$\|\bar{f}_{\theta_0^*}(\mathbf{X}+\mathcal{E}) - \bar{f}_{\theta_0^*}(\mathbf{X})\|_{\mathbb{R}^n} \leq \|\bar{f}_{\theta_0^*}\|_{C^{0,1}}\|\mathcal{E}\|. \tag{141}$$

Hence,

$$2\mathbb{E}_{(\mathbf{X},\mathcal{E})}\big(\|\bar{f}_{\theta_0^*}(\mathbf{X}+\mathcal{E}) - \bar{f}_{\theta_0^*}(\mathbf{X})\|_{\mathbb{R}^n}^2\big) \leq 2\|\bar{f}_{\theta_0^*}\|_{C^{0,1}}^2 \cdot n^2\sigma^2. \tag{142}$$

Combining these two estimates, we obtain

$$\mathcal{L}_r(\theta_0^*, \tau) \leq 2\mathcal{K}_0 + 2\|\bar{f}_{\theta_0^*}\|_{C^{0,1}}^2 \cdot n^2\sigma^2. \tag{143}$$

As $\mathcal{R}(\theta_0^*) \leq \mathcal{K}_0/\alpha$ we obtain

$$\mathcal{L}_r(\theta_0^*, \tau) = 2\mathcal{K}_0 + 2\|\bar{f}_{\theta_0^*}\|_{C^{0,1}}^2 \cdot n^2\sigma^2 \tag{144}$$

$$\leq 2\mathcal{K}_0 + 2L^2 c_0^{2L} \exp\big(2\mathcal{R}(\theta_0^*)\big) \cdot n^2\sigma^2, \tag{145}$$

which proves the claim when (122)–(123) hold.

Second, we consider the case when (122)–(123) do not hold. In this case, as above, we take $\theta = 0$ and see similarly to (131) that

$$\mathcal{L}_r(\theta, \tau) = \mathbb{E}_{(\mathbf{X},\mathcal{E})}\big(\|\bar{f}_\theta(\mathbf{X}+\mathcal{E}) - f(\mathbf{X})\|_{\mathbb{R}^n}^2\big) + \alpha\mathcal{R}(\theta)$$

$$\leq \mathbb{E}_{\mathbf{X}\sim\pi_{\mathbf{X}}}\Big(\big(\|\bar{f}_\theta(\mathbf{X})\|_{\mathbb{R}^n} + \|f(\mathbf{X})\|_{\mathbb{R}^n}\big)^2\Big) + \alpha\mathcal{R}(\theta) \tag{146}$$

$$\leq (n^{1/2}+1)^2\|f\|_\infty^2 + 0 \leq 4n\|f\|_\infty^2. \tag{147}$$

This proves the claim when (122)–(123) do not hold. ∎

4.2.3. Intrinsic error estimate. In this section we analyze the intrinsic error, that is, the expected error that comes from using the optimal (truncated) recurrent operator network to solve the inverse problem. We consider the case when the data is given with random noise.

Definition 4.3. The intrinsic error for parameters $\theta \in \Theta$ is given by

$$\mathcal{G}_{\text{intrinsic}}(\theta) = \mathcal{L}(\theta, \tau) = \mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\|\bar{f}_\theta(\mathbf{M}) - \mathbf{y}\|^2$$

$$= \mathbb{E}_{(\mathbf{X},\mathcal{E})\sim\pi_{\mathbf{X}}\times\pi_{\mathcal{E}}}\|\bar{f}_\theta(\mathbf{X}+\mathcal{E}) - f(\mathbf{X})\|_{\mathbb{R}^n}^2. \tag{148}$$

The optimal recurrent operator network is defined by the following

Definition 4.4. The optimal parameters for noisy measurements, θ^*, are a solution of

$$\theta^* = \arg\min_{\theta\in\Theta} \mathcal{L}_r(\theta,\tau) = \arg\min_{\theta\in\Theta} \big(\mathcal{G}_{\text{intrinsic}}(\theta) + \alpha\mathcal{R}(\theta)\big). \tag{149}$$

Again, here $\bar{f}_\theta$ are truncated basic recurrent operator networks of depth $L+2$ with the truncation parameter $m = \|f\|_\infty$; see (89)–(90).

Our above considerations yield the following

Lemma 4.3. *The optimal parameters for noisy measurements, θ^*, and the noisy expected loss satisfy*

$$\mathcal{L}_r(\theta^*,\tau) = \mathbb{E}_{(\mathbf{X},\mathcal{E})\sim\pi_{\mathbf{X}}\times\pi_{\mathcal{E}}}\big(\|\bar{f}_{\theta^*}(\mathbf{X}+\mathcal{E}) - f(\mathbf{X})\|^2_{\mathbb{R}^n}\big) \le \mathcal{L}_{r,0} \tag{150}$$

and

$$\mathcal{R}(\theta^*) \le \mathcal{R}_0. \tag{151}$$

In particular, the intrinsic error with parameter θ^* satisfies

$$\mathcal{G}_{\text{intrinsic}}(\theta^*) \le \mathcal{L}_{r,0}. \tag{152}$$

Proof. As θ^* satisfies the minimization problem (149), we see using Lemma 4.2 that

$$\mathcal{L}_r(\theta^*,\tau) = \mathcal{L}(\theta^*,\tau) + \alpha\mathcal{R}(\theta^*) \le \mathcal{L}_r(\theta_0^*,\tau) \le \mathcal{L}_{r,0}$$

and $\mathcal{R}(\theta^*) \le \mathcal{L}_{r,0}/\alpha = \mathcal{R}_0$. ∎

4.3. Optimal operator recurrent network as Bayes estimator

In this subsection, we discuss how the optimal neural networks can be considered a Bayesian estimators.

Below, we consider conditional expectations using σ-algebras. We recall the properties of the conditional expectations in Appendix B. Let $(\Omega,\Sigma,\mathbb{P})$ be an complete probability space and $\mathbf{M}\colon\Omega\to\mathbb{R}^{n\times n}$ and $\mathbf{y}\colon\Omega\to\mathbb{R}^n$ be random variables. We denote by τ the distribution of the pair $(\mathbf{M},\mathbf{y})$.

Let $\mathcal{B}_{\mathbf{M}}\subset\Sigma$ be the σ-algebra generated by the random variable $\mathbf{M}\colon\Omega\to\mathbb{R}^{n\times n}$. When $\mathbf{y}\in L^1(\Omega;\Sigma,d\mathbb{P})^n$ is a random variable, we denote the conditional expectation of $\mathbf{y}$ with respect to σ-algebra $\mathcal{B}_{\mathbf{M}}$ by $\mathbb{E}(\mathbf{y}|\mathcal{B}_{\mathbf{M}})$. Roughly speaking, $\mathbb{E}(\mathbf{y}\mid\mathcal{B}_{\mathbf{M}})$ denotes the expectation of a random variable $\mathbf{y}$ under the condition that $\mathbf{M}$ is known and $\mathbf{M}\to\mathbb{E}(\mathbf{y}\mid\mathcal{B}_{\mathbf{M}})$ can be considered as deterministic, measurable function of $\mathbf{M}$; see Appendix B. Below, we also use the notation

$$\mathbb{E}(\mathbf{y}\mid\mathcal{B}_{\mathbf{M}}) = \mathbb{E}(\mathbf{y}\mid\mathbf{M}).$$

Below, let $\mathbf{M}$ and $\mathbf{y}$ be as above in (116) and (118), that is, $\mathbf{M} = F(\mathbf{z}) + \mathcal{E}$ and $\mathbf{y} = p(\mathbf{z})$ where $p = f \circ F$. Thus, we have that $\mathbf{y} \in L^\infty(\Omega; \Sigma, d\mathbb{P})^n$ and

$$\|\mathbf{y}\|_{L^\infty(\Omega;\Sigma,d\mathbb{P})^n} \leq \|f\|_{L^\infty}.$$

Hence,

$$T_m(\mathbf{y}) = \mathbf{y},$$

where T_m is the truncation operator defined in (90) with $m = \|f\|_{L^\infty}$.

Now let $\mathfrak{H} = L^2(\Omega; \mathcal{B}_{\mathbf{M}}, d\mathbb{P})^n$ be the set of $\mathbb{R}^n$-valued functions that lie in $L^2(\Omega; \Sigma, d\mathbb{P})^n$ and are $\mathcal{B}_{\mathbf{M}}$-measurable. We note that $\mathfrak{H}$ is a closed subspace of the Hilbert space $L^2(\Omega; \Sigma, d\mathbb{P})^n$. For $\mathbf{y} \in L^2(\Omega; \Sigma, d\mathbb{P})^n$, we define

$$P_{\mathfrak{H}}\mathbf{y} = \operatorname*{arg\,min}_{\mathbf{u}\in\mathfrak{H}} \|\mathbf{y} - \mathbf{u}\|^2_{L^2(\Omega;\Sigma,d\mathbb{P})^n}, \tag{153}$$

which is the orthogonal projector onto the set $\mathfrak{H}$. As discussed in Appendix B,

$$P_{\mathfrak{H}}\mathbf{y} = \mathbb{E}(\mathbf{y} \mid \mathbf{M}). \tag{154}$$

As $\|\mathbf{y}\|_{L^\infty(\Omega;\Sigma,d\mathbb{P})^n} \leq m$, we see that

$$\|P_{\mathfrak{H}}\mathbf{y}\|_{L^\infty(\Omega;\Sigma,d\mathbb{P})^n} = \|\mathbb{E}(\mathbf{y} \mid \mathbf{M})\|_{L^\infty(\Omega;\Sigma,d\mathbb{P})^n} \leq \|\mathbf{y}\|_{L^\infty(\Omega;\Sigma,d\mathbb{P})^n} \leq m, \tag{155}$$

and thus $T_m(P_{\mathfrak{H}}\mathbf{y}) = P_{\mathfrak{H}}\mathbf{y}$, too.

We now consider the neural networks. We fix $K = 2n + 1$ and define the optimal truncated general operator recurrent operator network with depth L and level K to be $\overline{f}_{\theta^*_{(L,K)}}(\mathbf{M})$ with

$$\theta^*_{(L,K)} = \operatorname*{arg\,min}_{\widetilde{\theta}\in\widetilde{\Theta}_{L,K}} \mathbb{E}_{(\mathbf{y},\mathbf{M})\sim\tau}\big(|\mathbf{y} - \overline{f}_{\widetilde{\theta}}(\mathbf{M})|^2\big), \quad \overline{f}_{\widetilde{\theta}}(\mathbf{M}) = T_m\big(f_{\widetilde{\theta}}(\mathbf{M})\big). \tag{156}$$

Observe that here the minimized function is the nonregularized loss function for the truncated general recurrent operator network $\overline{f}_{\widetilde{\theta}}$.

Proposition 4.4. *Let $n \in \mathbb{Z}_+$ and $K = 2n + 1$. Then the optimal truncated general operator recurrent operator network with depth L and level K, denoted by $\overline{f}_{\theta^*_{(L,K)}} = T_m \circ f_{\tilde{\theta}(L,K)}$ satisfies*

$$\lim_{L\to\infty} \overline{f}_{\theta^*_{(L,K)}}(\mathbf{M}) = \mathbb{E}(\mathbf{y} \mid \mathbf{M}) \quad \text{in } L^2(\Omega; \Sigma, d\mathbb{P})^n. \tag{157}$$

Proof. From the functional analytic viewpoint, the conditional expectation is a projector onto a suitable function space, namely $\mathfrak{H}$ introduced above. Theorem 2.3 will imply that deep operator recurrent networks are dense in this space. We will combine these facts with an analysis of truncated operator recurrent networks to prove the claim.

Let $\mathcal{K}_{L,K}$ be the space of functions $f_{\tilde{\theta}}(\mathbf{M})$ where $f_{\tilde{\theta}}$ is a general operator recurrent neural network of depth L, level K and width n with $\tilde{\theta} \in \widetilde{\Theta}_{L,K}$. Note that these neural networks are not truncated. We denote $\mathfrak{L}^2 = L^2(\Omega; \Sigma, d\mathbb{P})^n$. Using Theorem 2.3, we observe that

$$\mathcal{K} = \mathrm{cl}\Big(\bigcup_{L=1}^{\infty} \mathcal{K}_{L,K} \Big) \tag{158}$$

is equal to $\mathfrak{S}$ with cl the closure in $\mathfrak{L}^2$; see Remark 7.

Let $\mathbf{u}_L \in \mathcal{K}_{L,K}$ be the nearest point in the set $T_m(\mathcal{K}_{L,K})$ to $\mathbf{y}$ and $\mathbf{u}_\infty \in \mathfrak{S}$ be the nearest point in the set $\mathfrak{S}$ to $\mathbf{y}$, that is,

$$\mathbf{u}_L = Q_{K,L}(\mathbf{y}) \in \operatorname*{arg\,min}_{\mathbf{u} \in T_m(\mathcal{K}_{L,K})} \big(\|\mathbf{y} - \mathbf{u}\|^2_{\mathfrak{L}^2} \big), \quad \mathbf{u}_\infty = P_{\mathfrak{S}}\mathbf{y}. \tag{159}$$

Recall, that by (155) we have $P_{\mathfrak{S}}\mathbf{y} = T_m(P_{\mathfrak{S}}\mathbf{y}) \in T_m(\mathfrak{S})$.

We emphasize that here $\mathbf{u}_L$ may not be uniquely determined as $\mathcal{K}_{L,K}$ are not linear subspaces.

As $T_m(\mathcal{K}_{L,K}) \subset \mathfrak{S}$, we have

$$d_L = \|\mathbf{u}_L - \mathbf{y}\|_{\mathfrak{L}^2} \geq \|\mathbf{u}_\infty - \mathbf{y}\|_{\mathfrak{L}^2} = d_\infty. \tag{160}$$

On the other hand, (158) implies that there are elements $\mathbf{w}_L \in \mathcal{K}_{L,K}$ such that $\mathbf{w}_L \to \mathbf{u}_\infty$ in $\mathfrak{L}^2$ as $L \to \infty$. As $\mathbf{u}_\infty \in T_m(\mathfrak{S})$, it is easy to see that

$$\bar{\mathbf{w}}_L = T_m \circ \mathbf{w}_L \in T_m(\mathcal{K}_{L,K}) \subset \mathfrak{S}$$

and $\bar{\mathbf{w}}_L \to \mathbf{u}_\infty$ in $\mathfrak{L}^2$ as $L \to \infty$.

Then, $\|\bar{\mathbf{w}}_L - \mathbf{y}\|_{\mathfrak{L}^2} \to d_\infty$ as $L \to \infty$ and as $\mathbf{u}_L$ are nearest points in $T_m(\mathcal{K}_{L,K}) \subset \mathfrak{S}$ to $\mathbf{y}$, we have

$$\|\bar{\mathbf{w}}_L - \mathbf{y}\|_{\mathfrak{L}^2} \geq d_L \geq d_\infty.$$

These imply that $d_L \to d_\infty$ as $L \to \infty$. As $\mathbf{u}_\infty - \mathbf{y} \perp \mathfrak{S}$ in $\mathfrak{L}^2$, we have

$$\|\mathbf{u}_L - \mathbf{u}_\infty\|^2_{\mathfrak{L}^2} + d^2_\infty = \|\mathbf{u}_L - \mathbf{u}_\infty\|^2_{\mathfrak{L}^2} + \|\mathbf{u}_\infty - \mathbf{y}\|^2_{\mathfrak{L}^2} = \|\mathbf{u}_L - \mathbf{y}\|^2_{\mathfrak{L}^2} = d^2_L,$$

the limit $d_L \to d_\infty$ implies that $\|\mathbf{u}_L - \mathbf{u}_\infty\|_{\mathfrak{L}^2} \to 0$ as $L \to \infty$. This shows that $Q_{K,L}(\mathbf{y}) \to P_{\mathfrak{S}}\mathbf{y}$ as $L \to \infty$. This and formula (154) yield the claim. ∎

This implies that the optimal general operator recurrent network which minimizes the expected loss, $\mathcal{L}(\tilde{\theta}, \tau)$ (represented by a formula analogous to (120) but with general operator recurrent networks) approximate a Bayes estimator for the inverse problem without regularization. Essentially, the deep neural network, here, is used to parametrize the set of decision rules considered in the Bayes estimator.

Remark 12. When f is a random function having a suitable prior distribution it is possible to prove posterior consistency and contraction rates, which give theoretical guarantees that the posterior mean converges to the true solution (determined by f) as the amount of data becomes larger and data error tends to zero [1,33,67,73]. Thus, one expects $f_{\theta^*_{(L,K)}}$ to approximate f.

5. Trained operator recurrent network and generalization

We employ the convex function $\mathcal{R}$ as an explicit regularizer in the loss function for training the network, and we show that this regularizer controls the regularity of the resulting local minimizer. This regularizer also provides a form of norm control, which in conjunction with a concentration inequality allows us to produce a generalization bound based on bounding the difference between the expected loss and the empirical loss. Theoretical bounds for generalization and other regularity properties by controlling the norms of parameters have been studied extensively in the literature for neural networks in many different contexts [12,59,71,72]. We perform a similar analysis, but still distinct from the above works, since operator recurrent networks are different from standard deep neural networks in an essential way. To clarify the presentation, we consider only (truncated) basic operator recurrent neural networks $\bar{f}_\theta$. The generalization for general operator recurrent neural networks and for the additional layers g_θ is possible but we omit these details.

Training data and empirical loss. The training data is the set

$$S = \{(X_j, y_j) : j = 1, 2, \dots, s\}, \tag{161}$$

where $s \in \mathbb{N}$ and (X_j, y_j) are independent samples of the random variable $(\mathbf{M}, \mathbf{y})$ having the distribution τ. As discussed above, using the training data S in (161), our primary aim is to find a recurrent operator network $\bar{f}_\theta : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ that approximates the map $f : \mathbb{R}^{n\times n} \to \mathbb{R}^n$. Incorporating the composition with g and finding a neural network with fully connected layers that approximates it, with training data

$$S' = \{(y_j, z_j) : j = 1, 2, \dots, s\},$$

will be addressed at the end of this section.

For training set S the empirical loss function is given by

$$\mathcal{L}^{(\mathrm{em})}(\theta, S) = \frac{1}{s}\sum_{i=1}^{s} \| \bar{f}_\theta(X_i) - y_i \|^2_{\mathbb{R}^n} \tag{162}$$

and the empirical regularized loss function is given by

$$\mathcal{L}_r^{(\mathrm{em})}(\theta, S) = \frac{1}{s}\sum_{i=1}^{s} \| \bar{f}_\theta(X_i) - y_i \|_{\mathbb{R}^n}^2 + \alpha \mathcal{R}(\theta). \tag{163}$$

Here, $\bar{f}_\theta$ are truncated basic recurrent operator networks of depth $L+2$ with truncation parameter m; see (89)–(90). Below, we assume for simplicity that $m = \|f\|_\infty$.

Typically, training is the optimization problem of finding parameters θ, given a training set S, such that $\mathcal{L}_r^{(\mathrm{em})}(\theta, S)$ is minimized.

5.1. Optimal neural network for sampled data

In this section we consider neural networks that are truncated.

As seen above in (137), there are $\theta \in \Theta$ such that $\mathcal{L}_r(\theta, \tau) \le \mathcal{L}_{r,0}$, where $\mathcal{L}_{r,0}$ was defined in (132)–(133). Thus, when we minimize $\mathcal{L}_r(\theta, \tau)$ subject to condition $\theta \in \Theta$, without loss of generality, we can search only among parameters $\theta \in \Theta$ such that

$$\mathcal{L}_r(\theta, \tau) \le \mathcal{L}_{r,0}. \tag{164}$$

We will see later that when the number of training samples, that is, s is large, it is probable that $\mathcal{L}_r^{(\mathrm{em})}(\theta, S)$ is close to $\mathcal{L}_r(\theta, \tau)$. Due to this we enforce in the optimization of θ the constraint $\mathcal{L}_r^{(\mathrm{em})}(\theta, S) \le \mathcal{L}_{r,0}$ which automatically enforces the constraint

$$\mathcal{R}(\theta) \le \mathcal{R}_0, \tag{165}$$

where $\mathcal{R}_0$ is defined in (134). This yields the minimization problem with inequality constraint,

$$\text{find } \theta \text{ minimizing } \mathcal{L}_r^{(\mathrm{em})}(\theta, S) \text{ subject to } \mathcal{R}(\theta) \le \mathcal{R}_0. \tag{166}$$

Due to this we introduce the following definition.

Definition 5.1. The optimal weights corresponding to the training set S, denoted $\theta(S)$, are a solution of the minimization problem

$$\theta(S) = \operatorname*{arg\,min}_{\theta \in \Theta(\mathcal{R}_0)} \mathcal{L}_r^{(\mathrm{em})}(\theta, S), \tag{167}$$

where

$$\Theta(\mathcal{R}_0) = \{\theta \in \Theta : \mathcal{R}(\theta) \le \mathcal{R}_0\}. \tag{168}$$

We note that when (165) holds (see also (92)), we have

$$\mathcal{L}_r(\theta, \mathbf{M}, \mathbf{y}) \le (1 + n^{1/2})^2 \|f\|_\infty^2 + \alpha \mathcal{R}_0 \le \mathcal{B}_0 \quad \text{for a.e. } (\mathbf{M}, \mathbf{y}) \sim \tau, \tag{169}$$

where

$$\mathcal{B}_0 := 9n\|f\|_\infty^2, \tag{170}$$

see formulas (92) and (135)–(136).

We use below the following technical result.

Lemma 5.1. *If* (122)–(123) *holds and*

$$\alpha \geq \frac{1}{R_0}\big(\varepsilon_0^2 + L^2 c_0^{2L} \exp(4R_0)\cdot n\sigma^2\big), \tag{171}$$

then

$$\mathcal{R}_0 \leq 4R_0. \tag{172}$$

Proof. The proof is a straightforward computation: Inequality (171) implies that $\alpha \geq \frac{1}{R_0}\varepsilon_0^2$, so that $\varepsilon_0^2/\alpha \leq R_0$ and $(\varepsilon_0^2 + \alpha R_0)/\alpha \leq 2R_0$. Thus, (171) implies

$$\begin{aligned} 4R_0\alpha &\geq 2R_0\alpha + \big(2\varepsilon_0^2 + 2L^2 c_0^{2L} \exp\big(2(\varepsilon_0^2 + \alpha R_0)/\alpha\big)\cdot n^2\sigma^2\big) \\ &\geq \big(2\varepsilon_0^2 + 2R_0\alpha + 2L^2 c_0^{2L} \exp\big(2(\varepsilon_0^2 + \alpha R_0)/\alpha\big)\cdot n^2\sigma^2\big) \geq \mathcal{L}_{r,0}. \end{aligned}$$

Hence,

$$\mathcal{R}_0 = \frac{1}{\alpha}\mathcal{L}_{r,0} \leq 4R_0. \qquad \blacksquare$$

For the remainder of this paper, we study (local) minimizers to (166) and show how the resulting neural networks, $\overline{f}_\theta$, enjoy good approximation properties with respect to the true function f. We directly analyze minimizers without studying how to compute them. Typically, the minimization is carried out using variants of stochastic gradient descent. Note that while the architecture of operator recurrent networks differs from that of standard neural networks, gradient descent can still be performed in a straightforward manner with the computation of the gradients via the chain rule. The key difference is that these gradients will contain multiplicative terms with X.

Selecting the parameter α. We later show that a large value of α leads to greater control over the so-called generalization gap. However, a large value of α also leads to large errors, $\|\overline{f}_\theta(X) - f(X)\|^2$, which govern the accuracy of the trained network at approximating the true function f. This is due to the fact that with a large α, the loss function will be best minimized by reducing the regularization term $\mathcal{R}$ rather than reducing the error.

Cross validation is an established technique for selecting the best predictive model among a set of candidates; see [18]. However, we note that this approach may not be practically applicable for large-scale inverse problems. It can be employed to choose the smallest value of α that has good generalization properties as follows:

Given a training set S and particular α, we partition S into equally sized subsets $S_1, S_2, \ldots, S_K$. For each $i = 1, \ldots, K$, we train a network on $S \setminus S_i$ and then evaluate the prediction error on S_i. The arithmetic mean of these errors over all i is computed to produce a *cross-validated error*. Then, given a finite set of candidate parameter values $\alpha_1, \alpha_2, \ldots$, the smallest is chosen such that the corresponding cross-validated error is below some tolerance. These techniques have been used, for example, to regularize solutions to linear systems [31].

5.2. Sparsity bounds

Below, we will show that the regularized minimizer will be found in a set $\Theta(N_0, \mathcal{R}_0)$ that consists of sparse sequences. As the set of sparse sequences is a union of finite dimensional sets, $\Theta(N_0, \mathcal{R}_0)$ can be covered with a "relatively small" number of balls. We will use this and Hoeffding's inequality to obtain improved generalization bounds.

Let $\theta(S)$ be a minimizer that we obtain for (166). We show that $\theta(S)$ enjoys some sparsity bounds which are controlled through the regularizing term $\mathcal{R}(\theta(S))$. We let

$$\mathcal{N}(\theta) := \#\{(\ell, p, i) \in P_1 \cup P_2 : \text{vector } \theta_p^{\ell,i} \text{ is nonzero}\}, \tag{173}$$

$$\mathcal{N}_1(\theta) := \#\{(\ell, p, i) \in P_1 : \text{vector } \theta_p^{\ell,i} \text{ is nonzero}\}, \tag{174}$$

where $\#A$ denotes the cardinality of the set A.

Theorem 5.2. *Let θ satisfy (165). Then*

$$\mathcal{N}_1(\theta(S)) \leq \frac{2Lc_0^L \mathcal{R}_0^{3/2}}{\alpha^{1/2}} \exp(2\mathcal{R}_0) \leq \frac{2Lc_0^L (\mathcal{L}_{r,0})^{3/2}}{\alpha^2} \exp\left(\frac{2\mathcal{L}_{r,0}}{\alpha}\right). \tag{175}$$

Proof. We will use estimates of the directional derivatives to derive sparsity estimates on the parameters. Let $S = \{(X_1, y_1), \ldots, (X_s, y_s)\}$. At the minimizer $\theta(S)$, every directional derivative of $\mathcal{L}_r^{(\mathrm{em})}(\theta, S)$ is nonnegative. Then we compute the derivative of $\mathcal{L}_r^{(\mathrm{em})}(\theta, S)$ with respect to $\theta \in (\mathbb{R}^n)^P$ in direction v and obtain

$$\partial_v \mathcal{L}_r^{(\mathrm{em})}(\theta, S) = \frac{2}{s} \sum_{j=1}^{s} \partial_v \bar{f}_\theta(X_j) \cdot \left(\bar{f}_\theta(X_j) - y_j\right) + \alpha \partial_v \mathcal{R}(\theta). \tag{176}$$

At $\theta = \theta(S)$, we must have $\partial_v \mathcal{L}_r^{(\mathrm{em})}(\theta(S), S) \geq 0$ for every direction v, and, hence,

$$\begin{aligned} -\partial_v \mathcal{R}(\theta)\big|_{\theta(S)} &\leq \frac{2}{s\alpha} \sum_{j=1}^{s} \partial_v \bar{f}_\theta(X_j)\big|_{\theta(S)} \cdot \left(\bar{f}_{\theta(S)}(X_j) - y_j\right) \\ &\leq \frac{2}{s\alpha} \sum_{j=1}^{s} \left\| \partial_v \bar{f}_\theta(X_j)\big|_{\theta(S)} \right\|_{\mathbb{R}^n} \left\| \bar{f}_{\theta(S)}(X_j) - y_j \right\|_{\mathbb{R}^n} \end{aligned} \tag{177}$$

$$\leq \frac{2(\mathcal{L}_{r,0})^{1/2}}{\alpha}\left(\frac{1}{s}\sum_{j=1}^{s}\left\|\partial_v \bar{f}_\theta(X_j)\big|_{\theta(S)}\right\|_{\mathbb{R}^n}^2\right)^{1/2} \tag{178}$$

$$\leq \frac{2\mathcal{R}_0^{1/2}}{\alpha^{1/2}}\left(\frac{1}{s}\sum_{j=1}^{s}\left\|\partial_v \bar{f}_\theta(X_j)\big|_{\theta(S)}\right\|_{\mathbb{R}^n}^2\right)^{1/2}, \tag{179}$$

where we used Hölder's inequality.

Next, we derive a relationship between $\partial_v\ \mathcal{R}(\theta)|_{\theta(S)}$ and the sparsity of $\theta(S)$. For a given $(\ell, i, p) \in P$, for which the corresponding column vector of $\theta(S)$, denoted by $\theta(S)_p^{\ell,i}$, is nonzero, we consider the directional derivative with $v = v_p^{\ell,i}$ signifying the unit vector pointing in the direction of $-\theta_p^{\ell,i}$. Then,

$$w_p^{\ell,i} := \partial_{v_p^{\ell,i}} \mathcal{R}(\theta)\big|_{\theta(S)} = -1. \tag{180}$$

Therefore,

$$\begin{aligned}\mathcal{N}_1(\theta(S)) &\leq -\sum_{(\ell,i,p)\in P_1} w_p^{\ell,i} \\ &\leq \frac{2\mathcal{R}_0^{1/2}}{\alpha^{1/2}}\sum_{i,p:(\ell,i,p)\in P_1}\left(\frac{1}{s}\sum_{j=1}^{s}\left\|\partial_{v_p^{\ell,i}} \bar{f}_\theta(X_j)\big|_{\theta(S)}\right\|_{\mathbb{R}^n}\right)^{1/2} \\ &\leq \frac{2\mathcal{R}_0^{1/2}}{\alpha^{1/2}}\sum_{i,p:(\ell,i,p)\in P_1} K_p^{\ell,i},\end{aligned} \tag{181}$$

where the $K_p^{\ell,i}$ are the derivative estimates obtained in Lemma 3.1. Thus, we have

$$\begin{aligned}\mathcal{N}_1(\theta(S)) &\leq \frac{2Lc_0^L\mathcal{R}_0^{1/2}}{\alpha^{1/2}}\exp(\mathcal{R}_0)\left(\sum_{(\ell,i,p)\in P_1}\|\theta(S)_{(p)'}^{\ell,i}\|_{\mathbb{R}^n}\right) \\ &\leq \frac{2Lc_0^L\mathcal{R}_0^{3/2}}{\alpha^{1/2}}\exp(2\mathcal{R}_0).\end{aligned} \tag{182}$$

This completes the proof. ∎

Using Theorem 5.2 for finding the best parameters $\theta(S)$ given training set S, we may solve (166) with a new constraint: When we define

$$N_1 = \left\lfloor \frac{2Lc_0^L\mathcal{R}_0^{3/2}}{\alpha^{1/2}}\exp(2\mathcal{R}_0)\right\rfloor = \left\lfloor \frac{2Lc_0^L(\mathcal{L}_{r,0})^{3/2}}{\alpha^2}\exp\left(2\frac{\mathcal{L}_{r,0}}{\alpha}\right)\right\rfloor, \tag{183}$$

without loss of generality, we may consider the minimization problem (166) with the constraint that the parameters θ satisfy

$$\mathcal{N}_1(\theta) \leq N_1,$$

where $\mathcal{N}_1(\theta)$, defined in (174), is the number of nonzero parameters determining the weight matrices. Thus, we may consider the minimization problem (166) with adding the constraint that the parameters θ satisfy

$$\mathcal{N}(\theta) \leq N_0, \quad N_0 = N_1 + 2L + 1, \tag{184}$$

where $\mathcal{N}(\theta)$, defined in (173), is the number of nonzero parameters determining the weight matrices and the bias vectors. Effectively, the size of the set of feasible parameters is further reduced by imposing (184). We denote by $\Theta(N_0, \mathcal{R}_0) \subset \Theta$ the set

$$\Theta(N_0, \mathcal{R}_0) = \{\theta \in \Theta : \mathcal{N}_1(\theta) \leq N_1, \ \mathcal{R}(\theta) \leq \mathcal{R}_0\}. \tag{185}$$

Then we redefine $\theta(S)$ to be a solution of a problem analogous to (166), where a minimizer is sought in $\Theta(N_0, \mathcal{R}_0)$, that is,

$$\theta(S) = \underset{\theta \in \Theta(N_0, \mathcal{R}_0)}{\arg\min} \ \mathcal{L}_r^{(\mathrm{em})}(\theta, S). \tag{186}$$

We now estimate the size of $\Theta(N_0, \mathcal{R}_0)$. We recall that in our original construction of operator recurrent networks we proposed that there could be layers with shared parameters. Therefore, we let $L_1 \leq L$ represent the number of independently parametrized layers in the network; in some cases, this quantity may be much smaller than L. Then $\Theta(N_0, \mathcal{R}_0) \subset (\mathbb{R}^n)^P$ is given by a finite union of M_0 compact subsets of linear subspaces,

$$\Theta(N_0, \mathcal{R}_0) = \bigcup_{i=1}^{M_0} V_i, \tag{187}$$

where

$$M_0 = \binom{\#P_1}{N_1} \leq (\#P_1)^{N_1} \leq (4nL)^{2Lc_0^L \mathcal{R}_0^{3/2} \alpha^{-1/2} \exp(2\mathcal{R}_0)}, \tag{188}$$

where $\mathcal{R}_0$ was introduced in (134), and $V_1, V_2, \ldots, V_{M_0}$ are compact subsets of linear subspaces of the full parameter space, such that each V_i, $i = 1, 2, \ldots, M_0$, has dimension $N_0 n$. Indeed, each V_i consists of those $\theta = (\theta_p^{\ell,i})_{\ell,i,p}$ for which all such components are zero except those corresponding to N_1 choices of indices $(\ell, i, p) \in P_1$, along with the condition that $\mathcal{R}(\theta) \leq \mathcal{R}_0$.

We will extensively use the fact that the set $\Theta(N_0, \mathcal{R}_0) \subset (\mathbb{R}^n)^P$ of the form (187) has Hausdorff dimension nN_0 which is significantly smaller than $n \cdot (\#P)$. This means that the assumption that θ is a nN_0-sparse vector implies that θ is in a lower dimensional subset of the parameter space $\mathbb{R}^{nP}$.

In particular, the above means that when regularization parameter α is sufficiently large, we optimize the parameter θ over a set consisting of sparse vectors. Thus, when α grows, the Hausdorff dimension of the parameter set $\Theta(N_0, \mathcal{R}_0)$ (for the

optimization problem (149)) becomes smaller. This property is crucial, and we will see below that generalization estimates become stronger when α grows.

We have assumed that the parameters $\theta_p^{\ell,i}$ with index $(\ell, i, p) \in P_1$, that correspond to the weight matrices, are sparse. However, the parameters $\theta_p^{\ell,i}$ with index $(\ell, i, p) \in P_2$ that correspond to the bias terms, are not assumed to be sparse.

We cover the finite union $\Theta(N_0, \mathcal{R}_0)$ with a finite set of balls of radius ρ with respect to the $\mathcal{R}$-norm. This allows us to further estimate the parameter set $\Theta(N_0, \mathcal{R}_0)$ with a discrete, finite set.

Lemma 5.3. *Let $\Theta(N_0, \mathcal{R}_0)$ be the disjoint union of compact sets given in (187). Then, for every $\rho \in (0, \mathcal{R}_0)$, there exists a finite set Θ_ρ satisfying*

$$\#(\Theta_\rho) \leq 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n}, \tag{189}$$

such that for every $\theta \in \Theta(N_0, \mathcal{R}_0)$, there exists $\widehat{\theta} \in \Theta_\rho$ such that

$$\| \overline{f}_\theta(X) - \overline{f}_{\widehat{\theta}}(X) \| \leq 2L c_0^L \rho (\mathcal{R}_0 + 2L) \exp(2\mathcal{R}_0) \tag{190}$$

for any $X \in \mathcal{B}_{n\times n}$.

Proof. The proof is based on the fact that the set of bounded sparse sequences is a union of bounded finite-dimensional sets that can be covered with a "relatively small" number of balls.

We write $\mathcal{I} = \{1, 2, \ldots, N_0\}$. For each component V_i, $i = 1, 2, \ldots, M_0$, in $\Theta(N_0, \mathcal{R}_0)$ there is an isometry $T_i \colon V_i \to V$, where

$$V = \left\{ (x_i)^{N_0} \in \mathbb{R}^{nN_0} : \|x\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} \leq \mathcal{R}_0 \right\}, \quad \|x\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} = \sum_{i=1}^{N_0} \|x_i\|_{\mathbb{R}^n} \leq \mathcal{R}_0, \tag{191}$$

where each x_i is an element of $\mathbb{R}^n$. Let $m = nN_0$. We call the sets

$$B_{1,2}^m(x_0, r) = \left\{ x \in \mathbb{R}^m : \|x - x_0\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} \leq r \right\}$$

the V-balls of radius r. Then, $V \subset B_{1,2}^m(0, \mathcal{R}_0)$. Let $\rho < \mathcal{R}_0$ and y_i, $i = 1, 2, \ldots, i_0$ be a maximal set of points in V such that

$$\|y_i - y_{i'}\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} > \rho \quad \text{for } i \neq i'.$$

Then the balls $B_{1,2}^m(y_i, \rho/2)$ are disjoint and contained in $B_{1,2}^m(0, \frac{3}{2}\mathcal{R}_0)$. When v_1 is the Euclidean volume of the V-ball $B_{1,2}^m(0, 1)$ in $\mathbb{R}^m$, the sum of volumes of the balls $B_{1,2}^m(y_i, \rho/2)$ is $i_0 v_1 (\rho/2)^m$ and this sum is bounded by $v_1 (\frac{3}{2}\mathcal{R}_0)^m$. Thus, $i_0 \leq (3\mathcal{R}_0/\rho)^m$ and $V \subset B_{1,2}^m(0, \mathcal{R}_0)$ can be covered by i_0 V-balls of radius ρ. Thus, the set $\Theta(N_0, \mathcal{R}_0)$ can be covered by $3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n}$ V-balls of radius ρ, the centers

of which are in $\Theta(N_0, \mathcal{R}_0)$. We let Θ_ρ be the collection of centers of all such V-balls of radius ρ. Then,

$$\#(\Theta_\rho) \leq 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n}.$$

Now, we consider any

$$\theta = (\theta_p^{\ell,i})_{(\ell,i,p)\in P} \in \Theta(N_0, \mathcal{R}_0),$$

where $\theta_p^{\ell,i} = (\theta_{p,j}^{\ell,i})_{j=1}^n \in \mathbb{R}^n$. We see that there exist $i \in \{1, 2, \ldots, M_0\}$, such that $\theta \in V_i$, and there is $\widehat{\theta} \in \Theta_\rho \cap V_i$ such that

$$\|\theta - \widehat{\theta}\|_{V_i} < \rho.$$

Let $\theta^{(q)}$, $q = 0, \ldots, N_0$ be such that $\theta^{(0)} = \theta$, $\theta^{(m)} = \widehat{\theta}$, and when $T_i\theta = \eta = (\eta_j)_{j=1}^m$, and $T_i\widehat{\theta} = \widehat{\eta} = (\widehat{\eta}_j)_{j=1}^m$, and $T_i\theta^{(q)} = \eta^{(q)} = (\eta_j^{(q)})_{j=1}^m$, we have

$$\begin{aligned} \eta_j^{(q)} &= \eta_j \quad \text{if } j \leq m - q, \\ \eta_j^{(q)} &= \widehat{\eta}_j \quad \text{if } j > m - q. \end{aligned} \tag{192}$$

Let $(\ell_q, i_q, p_q) \in P$ be such that T_i maps the unit vector in V_i corresponding to the coordinate with the index (ℓ_q, i_q, p_q) to the unit vector in V_i corresponding to the coordinate with the index q. We note that then

$$\|\theta^{(q+1)} - \theta^{(q)}\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} = \|(\theta^{(q+1)})_{p_q}^{\ell_q,i_q} - (\theta^{(q)})_{p_q}^{\ell_q,i_q}\|_{\mathbb{R}^n}$$

and

$$\sum_{q=0}^{N_0-1} \|\theta^{(q+1)} - \theta^{(q)}\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} \leq \rho.$$

We let $X \in \mathcal{B}_{n\times n}$ and $J_q = \{s\theta^{(q)} + (1-s)\theta^{(q+1)} \in V_i : 0 \leq s \leq 1\}$. Then, by Lemma 3.1,

$$\begin{aligned} \|\overline{f}_\theta(X) - \overline{f}_{\widehat{\theta}}(X)\|_{\mathbb{R}^n} &\leq \sum_{q=0}^{N_0-1} \|\overline{f}_{\theta^{(q+1)}}(X) - \overline{f}_{\theta^{(q)}}(X)\|_{\mathbb{R}^n} \\ &\leq \sum_{q=0}^{N_0-1} \sup_{\theta'\in J_q} \left\| \frac{\partial \overline{f}_\theta(X)}{\partial \theta_{p_q}^{\ell_q,i_q}}\bigg|_{\theta=\theta'} \right\|_{\mathbb{R}^n\to\mathbb{R}^n} \cdot \|(\theta^{(q+1)})_{p_q}^{\ell_q,i_q} - (\theta^{(q)})_{p_q}^{\ell_q,i_q}\|_{\mathbb{R}^n} \\ &\leq \left(\sup_{\theta'\in\Theta(N_0,\mathcal{R}_0)} \sum_{(\ell,i,p)\in P_1\cup P_2} \left\| \frac{\partial \overline{f}_\theta(X)}{\partial \theta_p^{\ell,i}}\bigg|_{\theta=\theta'} \right\|_{\mathbb{R}^n\to\mathbb{R}^n} \right) \\ &\quad \cdot \sum_{q=0}^{N_0-1} \|\theta^{(q+1)} - \theta^{(q)}\|_{\ell^1(\mathcal{I};\mathbb{R}^n)} \end{aligned}$$

$$\leq \sup_{\theta\in\Theta(N_0,\mathcal{R}_0)} \left(\sum_{(\ell,i,p)\in P_1} 2Lc_0^L \|\theta^{\ell,i}_{(p)'}\| \exp(\mathcal{R}(\theta)) + \sum_{(\ell,i,p)\in P_2} 2Lc_0^L \exp(\mathcal{R}(\theta)) \right) \rho$$

$$\leq 2Lc_0^L \rho(\mathcal{R}_0 + 2L) \exp(2\mathcal{R}_0). \tag{193}$$

This proves the claim. ∎

We point out that selecting the finite set Θ_ρ means selecting $\rho > 0$, or conversely, selecting ρ means selecting Θ_ρ. Hence, selecting a different θ means using a different finite set Θ_ρ. Below, we minimize loss functions over $\theta \in \Theta_\rho$ in the proofs of the relevant lemmas and theorems, but the set Θ_ρ is used only as an auxiliary tool so that in the proofs the minimization can be done over a finite set. A suitable value for the parameter ρ is later chosen in formula (211).

Remark 13. If L_0 is the total number of layers and L_1 the number of independent layers, then in the above estimates L is replaced by L_1 in Lipschitz estimates and in (190), and by L_0 in the definition of N_0 and in (189).

5.3. Generalization

In this subsection, we study the probability that a neural network optimized under our regularized empirical loss function can approximate the map f. Given a training set S, we therefore study the generalization error

$$\mathcal{G}(S) := |\mathcal{L}_r^{(\mathrm{em})}(\theta(S), S) - \mathcal{L}_r(\theta(S), \tau)| = |\mathcal{L}^{(\mathrm{em})}(\theta(S), S) - \mathcal{L}(\theta(S), \tau)|. \tag{194}$$

Given that the parameters $\theta(S)$ have been computed, $\mathcal{G}(S)$ measures the difference between the expected loss $\mathcal{L}(\theta(S), \tau)$ and the empirical loss $\mathcal{L}^{(\mathrm{em})}(\theta(S), S)$. If a model over fits the data, the empirical loss is very small while the expected loss remains large. Thus, an upper bound on $\mathcal{G}(S)$ provides some control over the degree of over fitting that is possible.

Considered as a random variable in S, we estimate the probability that $\mathcal{G}(S)$ is small using the following well-known inequality:

Lemma 5.4 (Hoeffding's inequality [39]). *Let $Z_1, \ldots, Z_N$ be N i.i.d. copies of the random variable Z whose range is $[0, Z_{\max}]$, $Z_{\max} > 0$. Then we have for $0 < \delta < \min(\mathbb{E}[Z], Z_{\max} - \mathbb{E}[Z])$,*

$$\mathbb{P}\left[\left| \frac{1}{N} \left(\sum_{i=1}^{N} Z_i \right) - \mathbb{E}[Z] \right| \leq \delta \right] \geq 1 - 2\exp\left(-2N\delta^2 {Z_{\max}}^{-2} \right). \tag{195}$$

To apply Hoeffding's inequality, one requires independent random variables. However, the optimal parameters $\theta(S)$ depend on every element of the training set S. Thus,

we use Lemma 5.3 to apply Hoeffding's inequality on each element of Θ_ρ, and then use the fact that $\theta(S)$ is sufficiently close to at least one element of Θ_ρ. We recall that ρ is the radius of the finite set of balls whose union cover $\Theta(N_0, \mathcal{R}_0)$. At this moment we keep ρ as a free parameter, and will fix its value later in formula (211). This leads to the main generalization result:

Theorem 5.5. *Let $\overline{f}_\theta$ be a truncated basic operator recurrent network with truncation parameter $m = \|f\|_\infty$, of with n and depth L. Consider a random training set $\mathbf{S}$ consisting of s independent samples from distribution τ, and let $\theta(S)$ be a minimizer for (149). Then,*

(i) *For any α and every sufficiently small $\delta > 0$,*

$$\mathbb{P}_{\mathbf{S}\sim\tau^s}\big[\mathcal{E}(\mathbf{S}) \le 2\delta\big] \ge 1 - C_1\Big(\frac{1}{\delta}\Big)^{C_2} \exp\Big(-\frac{2}{(9n)^2\|f\|_\infty^4} s\delta^2\Big), \tag{196}$$

where

$$C_1 = \exp\big(164 n^{3/2} L^2 c_0^{L+1}(1 + \|f\|_\infty)\exp(4\mathcal{L}_{r,0}\alpha^{-1})\big), \tag{197}$$

$$C_2 = 4nLc_0^L \exp(3\mathcal{L}_{r,0}\alpha^{-1}) \tag{198}$$

and $\mathcal{L}_{r,0} \le 4n\|f\|_\infty^2$, cf. (132)–(133).

(ii) *Let the function f be approximated with accuracy ε_0 by some neural network $\overline{f}_{\theta_0}$, where $\theta_0 \in \Theta(N_0, \mathcal{R}_0)$ has sparsity bound $R_0 \ge 1$; that is, conditions (122)–(123) hold with R_0 and ε_0. Then, for all*

$$\alpha \ge \frac{1}{R_0}\big(\varepsilon_0^2 + 2L^2 c_0^{2L}\exp(4R_0)\cdot n^2\sigma^2\big), \tag{199}$$

the inequality (196) holds, where

$$C_1 \le 2\exp\big(n^{3/2}2^{10}(1 + \|f\|_\infty)(1 + R_0)L^3 c_0^{L+1} e^{8R_0}(1 + R_0^2\alpha^{-1/2})\big), \tag{200}$$

$$C_2 \le 16nLc_0^L e^{8R_0}(1 + R_0^2\alpha^{-1/2}). \tag{201}$$

Note that when the depth L grows, the set of functions that the neural networks can represent has an increasingly richer structure and this is reflected by the growths of C_1 and C_2. Naturally s has to increase appropriately to mitigate this growth.

We also observe that in claim (i), making the regularization parameter α larger (that is, forcing the weight matrices to be sparser) makes the probability in (196) larger, but then the error how well the neural network approximates the function f becomes larger.

Proof. The main lines of the proof are the following: The truncated neural networks are bounded, so for each neural network $\overline{f}_\theta$ we can use Hoeffding's inequality. Moreover, the empirical optimizer $\theta(S)$ will be in the set $\Theta(N_0, \mathcal{R}_0)$ with large probability. We use a suitable value ρ which balances approximating an arbitrary element $\theta \in \Theta(N_0, \mathcal{R}_0)$ by an element in Θ_ρ and the number of elements in the set Θ_ρ. Applying Hoeffding's inequality to for all f_θ, $\theta \in \Theta_\rho$ will finalize the proof.

Fix $\theta \in \Theta(N_0, \mathcal{R}_0)$. When $\mathbf{S} = ((\mathbf{M}_i, \mathbf{y}_i))_{i=1}^s$ is a sequence of s independent random samples from distribution τ; see (119). We define the random variable

$$Z_i = \mathcal{L}_r(\theta, \mathbf{M}_i, \mathbf{y}_i). \tag{202}$$

The set of Z_i, $i = 1, \dots, s$, consists of i.i.d. copies of the random variable

$$Z = Z(\mathbf{M}, \mathbf{y}) := \mathcal{L}_r(\theta, \mathbf{M}, \mathbf{y}), \tag{203}$$

where $(\mathbf{M}, \mathbf{y})$ is distributed according to the probability distribution τ. The empirical loss is given by

$$\mathcal{L}_r^{(\mathrm{em})}(\theta, \mathbf{S}) = \frac{1}{s} \sum_{i=1}^{s} Z_i \tag{204}$$

and, by definition, the expected loss is

$$\mathcal{L}_r(\theta, \tau) = \mathbb{E}_{(\mathbf{M},\mathbf{y})}[Z(\mathbf{M}, \mathbf{y})]. \tag{205}$$

Since we assumed that $\overline{f}_\theta$ is a truncated network, we have by (169) that $0 \le Z \le \mathcal{B}_0$; therefore, by Hoeffding's inequality with $Z_{\max} = \mathcal{B}_0$, we have that

$$\mathbb{P}\big[|\mathcal{L}_r^{(\mathrm{em})}(\theta, \mathbf{S}) - \mathcal{L}_r(\theta, \tau)| \le \delta\big] \ge 1 - 2\exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \tag{206}$$

In particular, (206) holds for every element of Θ_ρ. Since

$$\#(\Theta_\rho) \le 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n}, \tag{207}$$

it follows that

$$\begin{aligned} &\mathbb{P}\big[\forall \theta \in \Theta_\rho : |\mathcal{L}_r^{(\mathrm{em})}(\theta, \mathbf{S}) - \mathcal{L}_r(\theta, \tau)| \le \delta\big] \\ &\quad \ge 1 - \sum_{\theta \in \Theta_\rho} \mathbb{P}\big[|\mathcal{L}_r^{(\mathrm{em})}(\theta, \mathbf{S}) - \mathcal{L}_r(\theta, \tau)| > \delta\big] \\ &\quad \ge 1 - 2 \cdot 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n} \exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \end{aligned} \tag{208}$$

Furthermore, in view of (190), for every $\theta \in \Theta(N_0, \mathcal{R}_0)$ there exists $\widehat{\theta} \in \Theta_\rho$ such that for any $X \in B_L^{n\times n}(1)$,

$$\|\overline{f}_\theta(X) - \overline{f}_{\widehat{\theta}}(X)\| \le 2Lc_0^L \rho(\mathcal{R}_0 + 2L)\exp(2\mathcal{R}_0). \tag{209}$$

Using this estimate and (92), we find that for any $X \in \mathcal{B}_{n\times n}$ and $y \in B_n(1)$, we have

$$\begin{aligned}
&|\|\bar{f}_\theta(X) - y\|^2 - \|\bar{f}_{\widehat{\theta}}(X) - y\|^2| \\
&\quad \le \left|\left(\|\bar{f}_\theta(X) - y\| + \|\bar{f}_{\widehat{\theta}}(X) - y\|\right) \cdot \left(\|\bar{f}_\theta(X) - y\| - \|\bar{f}_{\widehat{\theta}}(X) - y\|\right)\right| \\
&\quad \le 2(1 + n^{1/2})\|f\|_\infty 2Lc_0^L \rho(\mathcal{R}_0 + 2L)\exp(2\mathcal{R}_0) \\
&\quad \le 4n^{1/2}(1 + \|f\|_\infty) 2Lc_0^L \rho(\mathcal{R}_0 + 2L)\exp(2\mathcal{R}_0). \qquad (210)
\end{aligned}$$

Below, we denote $Q = 4n^{1/2}(1 + \|f\|_\infty)$.

We next consider the implications of the above estimates when ρ has the value

$$\rho = \frac{\delta}{2Q \cdot 2Lc_0^L(\mathcal{R}_0 + 2L)\exp(2\mathcal{R}_0)}. \qquad (211)$$

Then, for any S, there exists $\widehat{\theta} = \widehat{\theta}(S) \in \Theta_\rho$ such that

$$\mathcal{G}(S) \le |\mathcal{L}_r(\widehat{\theta}, \tau)| + Q \cdot 2Lc_0^L \rho(\mathcal{R}_0 + 2L)\exp(2\mathcal{R}_0). \qquad (212)$$

When we apply this observation for randomly chosen samples $\mathbf{S}$, we obtain that

$$\begin{aligned}
&\mathbb{P}\left[\mathcal{G}(\mathbf{S}) \le \delta + Q \cdot 2Lc_0^L \rho(\mathcal{R}_0 + 2L)\exp(2\mathcal{R}_0)\right] \\
&\qquad \ge 1 - 2 \cdot 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n} \exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \qquad (213)
\end{aligned}$$

We substitute our expressions for $\mathcal{R}_0$, N_0 and M_0 to obtain the estimate

$$\mathbb{P}\left[\mathcal{G}(\mathbf{S}) \le 2\delta\right] \ge 1 - C_0 \exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}), \qquad (214)$$

where

$$\begin{aligned}
C_0 &= 2 \cdot 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{nN_0} \\
&\le 2M_0 \left(\frac{3 \cdot 4Lc_0^L Q \exp(4(\mathcal{R}_0 + 2L))}{\delta}\right)^{nN_0}.
\end{aligned}$$

As

$$M_0 \le (4nL)^{N_0},$$

we have

$$\begin{aligned}
C_0 &\le 2M_0 \left(\frac{12Lc_0^L Q \exp(4(\mathcal{R}_0 + 2L))}{\delta}\right)^{nN_0} \\
&\le 2\left(4nL \frac{(12LQ)^n c_0^{nL} \exp(4n(\mathcal{R}_0 + 2L))}{\delta^n}\right)^{N_0}.
\end{aligned}$$

Using that

$$N_0 \le 2Lc_0^L(1 + \mathcal{R}_0^{3/2}\alpha^{-1/2})\exp(2\mathcal{R}_0),$$

we obtain the estimate

$$C_0 \le 2\left(4nL\frac{(12LQ)^n c_0^{nL}\exp(4n(\mathcal{R}_0+2L))}{\delta^n}\right)^{N_0}$$
$$\le 2\left(4nL\frac{(12LQ)^n c_0^{nL}\exp(4n(\mathcal{R}_0+2L))}{\delta^n}\right)^{2Lc_0^L(1+\mathcal{R}_0^{3/2}\alpha^{-1/2})\exp(2\mathcal{R}_0)}. \quad (215)$$

For claim (i), we can use the facts that $\mathcal{R}_0 = \alpha^{-1}\mathcal{L}_{r,0}$ and $\mathcal{L}_{r,0} \le 4n\|f\|_\infty^2$, so that

$$\begin{aligned}
&C_0\\
&\le 2\left(4nL\frac{(12LQ)^n c_0^{nL}\exp(4n(\mathcal{L}_{r,0}\alpha^{-1}+2L))}{\delta^n}\right)^{2Lc_0^L(1+(\mathcal{L}_{r,0})^{3/2}\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})}\\
&\le 2\big(\delta^{-n}\exp\big(3+nL+nQ+nLc_0\\
&\qquad +4n(\mathcal{L}_{r,0}\alpha^{-1}+2L)\big)\big)^{2Lc_0^L(1+(\mathcal{L}_{r,0})^{3/2}\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})}\\
&\le C_1\Big(\frac{1}{\delta}\Big)^{C_2},
\end{aligned}$$

where, using that $\mathcal{L}_{r,0} \ge 1$, $Q \ge 1$, $c_0 \ge 1$, and $\alpha \le 1$,

$$\begin{aligned}
C_1 &= 2\exp\big[(3+nL+nQ+nLc_0+4n(\mathcal{L}_{r,0}\alpha^{-1}+2L))\\
&\qquad \cdot 2Lc_0^L(1+(\mathcal{L}_{r,0})^{3/2}\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})\big]\\
&\le 2\exp\big[20n(1+Q+c_0L+\mathcal{L}_{r,0}\alpha^{-1})\cdot Lc_0^L(1+(\mathcal{L}_{r,0})^2\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})\big]\\
&\le 2\exp\big[40nQc_0L(1+\mathcal{L}_{r,0}\alpha^{-1})\cdot Lc_0^L(1+\tfrac{1}{2}(\mathcal{L}_{r,0})^2\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})\big]\\
&\le \exp\big[41nQL^2c_0^{L+1}\exp(4\mathcal{L}_{r,0}\alpha^{-1})\big]\\
C_2 &= n\cdot 2Lc_0^L(1+(\mathcal{L}_{r,0})^{3/2}\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})\\
&\le 4nLc_0^L(1+\tfrac{1}{2}(\mathcal{L}_{r,0})^2\alpha^{-2})\exp(2\mathcal{L}_{r,0}\alpha^{-1})\\
&\le 4nLc_0^L\exp(3\mathcal{L}_{r,0}\alpha^{-1}).
\end{aligned}$$

This proves claim (i).

Now, we consider claim (ii). If (122)–(123) hold, and α satisfies the assumption in claim (ii), we have by Lemma 5.1 that

$$\mathcal{R}_0 \le 4R_0. \quad (216)$$

Hence, we find, using (215) and $2R_0 \ge 1$ and $\alpha \le 1$, that

$$\begin{aligned}
C_0 \le 2\big[&\delta^{-n}\exp\big(3+nL+nQ+nLc_0\\
&+4n(4R_0+2L)\big)\big]^{2Lc_0^L(1+(4R_0)^{3/2}\alpha^{-1/2})\exp(2\cdot 4R_0)}
\end{aligned}$$

$$\leq 2\big[\delta^{-n}\exp(3+9nL+nQ+nLc_0+16R_0)\big]^{16Lc_0^L(1+R_0^2\alpha^{-1/2})\exp(8R_0)}$$
$$\leq 2\exp\big[n2^8Q(1+R_0)L^3c_0^{L+1}e^{8R_0}(1+R_0^2\alpha^{-1/2})\big]\delta^{-16nLc_0^Le^{8R_0}(1+R_0^2\alpha^{-1/2})}$$

Thus, we obtain

$$C_1 \leq 2\exp\big[n2^8Q(1+R_0)L^3c_0^{L+1}e^{8R_0}(1+R_0^2\alpha^{-1/2})\big],$$
$$C_2 \leq 16nLc_0^Le^{8R_0}(1+R_0^2\alpha^{-1/2}).$$

This completes the proof of claim (ii). ∎

Estimate (196) quantifies the effect on the generalization error from varying the values of the regularization parameter α and the sample size s. Note that (196) approaches 1 exponentially fast with respect to increasing s. On the other hand, with increasing L the expressivity of the network also rapidly increases, so one may thus expect that the sample size s will need to increase accordingly in order to maintain a good generalization bound. Indeed, (196) decreases super-exponentially away from 1 as L increases. Similarly, increasing the regularization parameter α also reduces the generalization error, as it decreases the variance in the loss function. However, increasing α to improve the generalization competes with the goal of accurately approximating the true function. Furthermore, when (122)–(123) hold then the lower bound $\alpha \geq \varepsilon_0^2/R_0$ indicates when there is sufficient regularization. Additionally, a suitable value for the error lower bound δ can also be tuned to apply the bound meaningfully. If s, α, δ are not chosen judiciously, the resulting probability bound may be potentially meaningless, yielding a probability value close to, or potentially less than, zero.

5.4. Trained neural network versus optimal neural network

The generalization error expresses how efficient the training is. Here, we discuss how close the trained network is to an optimal network. We denote the optimal weights by θ^* and present a "generalization gap" type estimate for the error between networks with weights θ^* and weights $\theta(S)$.

We let θ^* be a solution of

$$\theta^* = \underset{\theta\in\Theta(N_0,\mathcal{R}_0)}{\arg\min}\ \mathcal{L}_r(\theta,\tau) = \underset{\theta\in\Theta(N_0,\mathcal{R}_0)}{\arg\min}\ \mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big(\|\bar{f}_\theta(\mathbf{M})-\mathbf{y}\|^2+\alpha\mathcal{R}(\theta)\big), \quad (217)$$

and write

$$\mathcal{L}_r^* = \mathcal{L}_r(\theta^*,\tau).$$

This means that $X \mapsto \bar{f}_{\theta^*}(X)$ is the neural network having the optimal expected performance for $(X,\mathbf{y})$ sampled from distribution τ. Note that the optimal parameter θ^* depends on the regularization parameter α, and to emphasize this we sometimes

denote it by $\theta^*(\alpha)$. Clearly,

$$\mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big(\|\bar{f}_{\theta^*(\alpha)}(\mathbf{M})-\mathbf{y}\|^2\big)\leq \mathcal{L}_{r,0}(\alpha), \tag{218}$$

cf. (132)–(133). We observe that when (122)–(123) holds, then, when α grows, also the bound $\mathcal{L}_{r,0}(\alpha)$ for the expected error in (218) may grow.

A trivial, but important observation is that when $\bar{f}_{\theta_0}$ is any neural network, for example, a neural network which corresponds to an implementation of the approximation of the analytic solution algorithm, we have

$$\mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big[\mathcal{L}_r(\theta^*,\mathbf{M},\mathbf{y})\big]\leq \mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big[\mathcal{L}_r(\theta_0,\mathbf{M},\mathbf{y})\big]. \tag{219}$$

This means that the optimal neural network $\bar{f}_{\theta^*}$ (or a network trained with a sufficiently large data set as elucidated below) has a better expected performance than the deterministic approximation $\bar{f}_{\theta_0}$ of the analytic solution algorithm.

Next, we estimate the expected performance gap between the optimal neural network and the neural network $\bar{f}_{\theta(S)}$ optimized with the training data S, defined by,

$$\mathcal{G}_{\text{opt}}(S):=\big|\mathbb{E}_{(\mathbf{M},\mathbf{y})\sim\tau}\big(\mathcal{L}_r(\theta(S),\mathbf{M},\mathbf{y})-\mathcal{L}_r(\theta^*,\mathbf{M},\mathbf{y})\big)\big|. \tag{220}$$

Given that the parameters $\theta(S)$ have been generated using the training set S, $\mathcal{G}_{\text{opt}}(S)$ measures the difference between the expected loss $\mathcal{L}_r(\theta(S),\tau)$ and the loss of the optimal neural network, $\mathcal{L}_r(\theta^*,\tau)$.

Using similar methods to those used to prove Theorem 5.5 we obtain the following:

Theorem 5.6. *Let $\bar{f}_\theta$ be truncated basic operator recurrent networks with truncation parameter $m=\|f\|_\infty$. Consider a random training set $\mathbf{S}$ consisting of s independent samples from distribution τ and let $\theta(\mathbf{S})$ be a minimizer for (166) and θ^* be a minimizer for (217) signifying the best possible weights. Then,*

(i) *For any $\alpha>0$ and every sufficiently small $\delta>0$, we have*

$$\mathbb{P}_{\mathbf{S}\sim\tau^n}\big[\mathcal{G}_{\text{opt}}(\mathbf{S})\leq 6\delta\big]\geq 1-2C_1\Big(\frac{1}{\delta}\Big)^{C_2}\exp\Big(-\frac{2}{(9n)^2\|f\|_\infty^4}s\delta^2\Big), \tag{221}$$

where C_1 and C_2 are given as in (198).

(ii) *Let the function f be approximated with accuracy ε_0 by some neural network $\bar{f}_{\theta_0}$, where $\theta_0\in\Theta(N_0,\mathcal{R}_0)$ has sparsity bound $R_0\geq 1$, that is, conditions (122)–(123) hold with R_0 and ε_0. Then, for all α satisfying (199), the inequality (221) holds with the constants C_1 and C_2 given by (200).*

Moreover,

$$\mathbb{P}_{\mathbf{S}\sim\tau^n}\big[\mathbb{E}_{\mathbf{M},\mathbf{y}}\big(\|\overline{f}_{\theta(\mathbf{S})}(\mathbf{M})-\mathbf{y}\|^2\big) \le 6\delta + 4\varepsilon_0^2 + 2\alpha R_0 + 2L^2 c_0^{2L}\exp(8R_0)\cdot n^2\sigma^2\big] \ge 1 - 2C_1\Big(\frac{1}{\delta}\Big)^{C_2}\exp\Big(-\frac{2}{(9n)^2\|f\|_\infty^4}s\delta^2\Big). \tag{222}$$

Roughly speaking, Theorem 5.6 (i) means that the trained neural network performs almost as well as the optimal neural network with large probability. Theorem 5.6 (ii) estimates the probability that training yields a neural network which output is close to that of the target function. We note that the training of the neural network does not require that we know θ_0, and thus Theorem 5.6 (ii) estimates the probability that the trained neural network $\overline{f}_{\theta(S)}$ approximates the function f when some θ_0 is just known to exist.

Proof. The main lines of the proof are the following: We will compare the minimization of empirical and nonempirical loss functions when the parameters θ vary either in the continuous index set $\Theta(N_0,\mathcal{R}_0)$ or in the finite index set Θ_ρ. Thus, we compare four minimization problems. Finally, the claim follows by applying the results for the generalization gap, that is, Theorem 5.5 for the "best" and the "worst" minimization problem.

Let ρ be given by (211). As in Theorem 5.2 above and (184), we find that θ^* satisfies the sparsity estimate

$$\mathcal{N}_1(\theta^*) \le N_1. \tag{223}$$

We will compare the optimal parameter θ^* with an optimal parameter θ_ρ^* in the finite set Θ_ρ, that is, θ_ρ^* is a solution of

$$\theta_\rho^* = \operatorname*{arg\,min}_{\theta_\rho\in\Theta_\rho}\mathcal{L}_r(\theta_\rho,\tau), \tag{224}$$

$$\mathcal{L}_{r,\rho}^* = \mathcal{L}_r(\theta_\rho^*,\tau). \tag{225}$$

As in (193), if $\widehat{\theta}\in\Theta(N_0,\mathcal{R}_0)_\rho$ satisfies $\|\widehat{\theta}-\theta\|_{\ell^1(\mathcal{I};\mathbb{R}^n)}\le\rho$, then for any $X\in\mathcal{B}_{n\times n}$,

$$\|\overline{f}_{\widehat{\theta}}(X)-\overline{f}_\theta(X)\|_{\mathbb{R}^n} \le 2Lc_0^L\rho(\mathcal{R}_0+2L)\exp(2\mathcal{R}_0), \tag{226}$$

$$|\|\overline{f}_{\widehat{\theta}}(X)-y\|^2-\|\overline{f}_\theta(X)-y\|^2| \le Q\,2Lc_0^L\rho(\mathcal{R}_0+2L)\exp(2\mathcal{R}_0)\le\delta, \tag{227}$$

where ρ is given by (211) and $Q=4n^{1/2}(1+\|f\|_\infty)$ as before, cf. (209)–(210). As $\rho\le\delta$, we have $\|\widehat{\theta}-\theta\|_{\ell^1(\mathcal{I};\mathbb{R}^n)}\le\delta$; then $\mathcal{L}_r{}^*(\rho)\le\mathcal{L}_r^*+2\delta$. Clearly, $\mathcal{L}_r^*\le\mathcal{L}_r^*(\rho)$. Thus,

$$\mathcal{L}_r^*\le\mathcal{L}_{r,\rho}^*\le\mathcal{L}_r^*+2\delta, \tag{228}$$

or, equivalently,

$$\mathcal{L}_r(\theta^*, \tau) \le \mathcal{L}_r(\theta^*_\rho, \tau) \le \mathcal{L}_r(\theta^*, \tau) + 2\delta. \tag{229}$$

Let training data S be sampled from τ^s, and let $\theta_\rho(S)$ be an optimal empirical parameter for S in Θ_ρ, that is,

$$\theta_\rho(S) = \arg\min_{\theta_\rho \in \Theta_\rho} \mathcal{L}_r^{(\mathrm{em})}(\theta_\rho, S), \tag{230}$$

$$\mathcal{L}_{r_\rho}(S) = \mathcal{L}_r^{(\mathrm{em})}(\theta_\rho(S), S). \tag{231}$$

We denote, as in the above, an optimal empirical parameter for sample S in the entire parameter set by

$$\theta(S) = \arg\min_{\theta_\rho \in \Theta(N_0, \mathcal{R}_0)} \mathcal{L}_r^{(\mathrm{em})}(\theta, S),$$
$$\mathcal{L}_r(S) = \mathcal{L}_r^{(\mathrm{em})}(\theta(S), S).$$

As in (228), we have

$$\mathcal{L}_r(S) \le \mathcal{L}_{r_\rho}(S) \le \mathcal{L}_r(S) + 2\delta, \tag{232}$$

or, equivalently,

$$\mathcal{L}_r^{(\mathrm{em})}(\theta(S), S) \le \mathcal{L}_r^{(\mathrm{em})}(\theta_\rho(S), S) \le \mathcal{L}_r^{(\mathrm{em})}(\theta(S), S) + 2\delta. \tag{233}$$

We recall that by (208),

$$\mathbb{P}_{\mathbf{S}\sim\tau^n}\left[\forall \theta \in \Theta_\rho : |\mathcal{L}_r^{(\mathrm{em})}(\theta, \mathbf{S}) - \mathcal{L}_r(\theta, \tau)| \le \delta\right] \ge 1 - 2\cdot 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n} \exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \tag{234}$$

By applying (234) when θ has the value $\theta_\rho(\mathbf{S}) \in \Theta_\rho$, we trivially obtain

$$\mathbb{P}_{\mathbf{S}\sim\tau^s}\left[|\mathcal{L}_r(\theta_\rho(\mathbf{S}), \mathbf{S}) - \mathcal{L}_r(\theta_\rho(\mathbf{S}), \tau)| \le \delta\right] \ge 1 - 2\cdot 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n} \exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}) \tag{235}$$

and, by applying (234) when θ has the value $\theta^*_\rho \in \Theta_\rho$, we trivially obtain

$$\mathbb{P}_{\mathbf{S}\sim\tau^s}\left[|\mathcal{L}_r(\theta^*_\rho, \mathbf{S}) - \mathcal{L}_r(\theta^*_\rho, \tau)| \le \delta\right] \ge 1 - 2\cdot 3^{N_0 n} M_0 (\mathcal{R}_0/\rho)^{N_0 n} \exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \tag{236}$$

We recall that for an arbitrary training data S, θ^*_ρ and $\theta_\rho(S)$ are defined to be some solutions of minimization problems (224) and (230), respectively. Thus, we have for all S,

$$\mathcal{L}_r^{(\mathrm{em})}(\theta_\rho(S), S) \le \mathcal{L}_r^{(\mathrm{em})}(\theta^*_\rho, S), \quad \mathcal{L}_r(\theta^*_\rho, \tau) \le \mathcal{L}_r(\theta_\rho(S), \tau). \tag{237}$$

By combining (235), (236), and (237), we obtain

$$\mathbb{P}_{\mathbf{S}\sim\tau^s}\big[|\mathcal{L}_r(\theta_\rho(\mathbf{S}),\tau)-\mathcal{L}_r(\theta_\rho^*,\tau)|\le 2\delta\big] \\ \ge 1-2\cdot 2\cdot 3^{N_0 n}M_0(\mathcal{R}_0/\rho)^{N_0 n}\exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \quad (238)$$

Combining this estimate with (229) and (233), we conclude that

$$\mathbb{P}_{\mathbf{S}\sim\tau^s}\big[|\mathcal{L}_r(\theta(\mathbf{S}),\tau)-\mathcal{L}_r(\theta^*,\tau)|\le 2\delta+2\cdot 2\delta\big] \\ \ge 1-2\cdot 2\cdot 3^{N_0 n}M_0(\mathcal{R}_0/\rho)^{N_0 n}\exp(-2s\delta^2\alpha^{-2}\mathcal{B}_0^{-2}). \quad (239)$$

This yields claim (i).

In claim (ii), the fact that inequality (221) holds with constants C_1 and C_2 given by (200) follows by estimating C_1 and C_2 as in the proof of Theorem 5.5. Finally, using inequalities (122), (123), (219), and (222), it follows that for any S

$$\begin{aligned}\mathbb{E}_{\mathbf{M},\mathbf{y}}\|\bar{f}_{\theta(S)}(\mathbf{M})-\mathbf{y}\|^2 &\le \mathcal{L}_r(\theta(S),\tau)\\ &\le \big(\mathcal{L}_r(\theta(S),\tau)-\mathcal{L}_r(\theta^*,\tau)\big)+4\varepsilon_0^2+2\alpha R_0\\ &\quad +2L^2c_0^{2L}\exp(2R_{\max})\cdot n^2\sigma^2. \end{aligned} \quad (240)$$

This inequality together with claim (i) yields claim (ii). ∎

Remark 14. Above we have considered a truncated basic operator recurrent network f_θ. The results can be generalized for a neural network $\bar{f}_{\vec{\theta}}$, $\vec{\theta}=(\theta_{s_1},\dots,\theta_{s_K})$ of the form

$$f_{\vec{\theta}}(X)=G\big(\bar{f}_\theta^1(X),\bar{f}_\theta^2(X),\dots,\bar{f}_\theta^K(X)\big), \quad (241)$$

where $\bar{f}_\theta^j(X)$, $j=1,2,\dots,K$ are basic operator recurrent networks and $G\colon\mathbb{R}^{Kn}\to\mathbb{R}^d$, $G(z_1,\dots,z_{Kn})=(G^a(z_1,\dots,z_{Kn}))_{a=1}^d$ is a given Lipschitz function, for example a neural network of the form (6)–(8). We call $\bar{f}_{\vec{\theta}}$ in (241) a combination of basic operator recurrent networks. This type of neural network is used below to analyze solution algorithms for inverse problems, cf. (241).

To obtain the generalizations of the above theorems an essential observation is that

$$\left\|\frac{\partial F_{\vec{\theta}}(X)}{\partial\theta_{p,s_j}^{\ell,i}}\right\|\le\|\nabla G\|\cdot\left\|\frac{\partial\bar{f}_\theta^j(X)}{\partial\theta_{p,s_j}^{\ell,i}}\right\|\le\mathrm{Lip}(G)\left\|\frac{\partial\bar{f}_\theta^j(X)}{\partial\theta_{p,s_j}^{\ell,i}}\right\|. \quad (242)$$

Using this and results of Lemma 3.1, we see that if $\mathrm{Lip}(G)\le 1$, then the Lipschitz constants of $F_{\vec{\theta}}(X)$ with respect to the components of $\vec{\theta}$ satisfy the analogous estimates that are given in Lemma 3.1 for a basic operator recurrent network f_θ. Moreover, if we assume that $\|G\|_\infty\le m=\|f\|_\infty$, then the proofs of Theorems 5.5 and 5.6 show that the claims of Theorems 5.5 and 5.6 are valid when the truncated basic

operator recurrent network f_θ is replaced by the combination of basic operator recurrent networks $F_{\vec{\theta}}$, when the number L in the claims of these theorems is replaced by the number KL, the terms $(9n)^2$ are replaced by $(5d)^2$, and the terms $n^{3/2}$ are replaced by $nd^{1/2}$. The first replacement is needed as the number of components of the parameters $\vec{\theta} = (\theta_{s_1}, \dots, \theta_{s_K})$ is increased by a factor K and hence the estimate in formula (188) changes. The second and the third replacements are needed as in the equation (170) and (210) the factor $9n$ is replaced by $9d$.

6. Example: Operator recurrent network for matrix inversion

Before we describe the relationship between operator recurrent neural networks and nonlinear inverse problems for the wave equation, we describe the simpler problem of matrix inversion. For $n > 0$ an integer, suppose we have a data set

$$\{(X_j, y_j); j = 1, \dots, s\}, \tag{243}$$

where each $X_j \in \mathbb{R}^{n\times n}$ is a nonsingular matrix and $y_j \in \mathbb{R}^n$. As before, the learning problem is to construct a function f whose graph $\{(X, y = f(X)\}$ closely fits the data set. However, suppose we also know that the data set comes from an algebraic relationship

$$Xy = h, \tag{244}$$

where $h \in \mathbb{R}^n$ is a fixed vector. Then the problem of constructing f can be solved exactly by $f(X) = X^{-1}h$. In other words, given a matrix X, we are tasked with learning how to apply its inverse to some particular vector h.

Developing efficient methods to solve linear systems under special conditions is a central problem in scientific computing. In the absence of any additional assumptions on the linear system, in practice one must use Gaussian elimination or variations thereof. However, over the decades, a variety of faster methods have been developed for specific families of matrices, such as those that are sparse, low-rank, oscillatory, arising from differential equations, and so forth. Of particular note are iterative methods, such as Krylov methods. Just as deep-learning-driven methods have been shown to be competitive with handmade algorithms in the realm of image processing, it is of similar interest to see whether deep-learning-driven matrix inversion can be competitive with handmade inversion methods.

It is important to reiterate that this problem is distinct from, and significantly more challenging than, a linear inverse problem. In the linear inverse problem, the data set consists of pairs of vectors $\{(x_j, y_j)\}$ which obeys a linear (or approximately linear) relationship $x = Ay$ for a fixed matrix A. In this case, the learning problem is to

construct the linear (or approximately linear) map $f(x) = A^{-1}y$. Traditional rectifier neural networks are well-suited to this task.

We seek to investigate the suitability of operator recurrent networks for learning to solve this problem under certain conditions. From Theorem 2.4, we know that operator recurrent networks are exactly equal to piecewise matrix polynomials, and therefore a natural question is how to approximate the matrix inversion problem with piecewise matrix polynomials. One notable special case is Neumann series, which represents the inverse of X via the matrix power series

$$X^{-1} = \sum_{k=0}^{\infty} (I - X)^k, \tag{245}$$

and this equality holds when $\|I - X\| < 1$, in which case the power series converges. By truncating this power series, we can approximate X^{-1} by a matrix polynomial, which can in turn be represented by an operator recurrent network. To apply Neumann series to any matrix X, we first rescale the matrix so that $\|I - X\| < 1$ is satisfied, before applying the series expansion, and then scale back.

Because it comes from a Taylor expansion, Neumann series is a very simplistic construction and only holds on the disk of convergence given by $\|I - X\| < 1$. When learning matrix inversion, we may have prior knowledge about additional spectral information of X, and this can allow us to produce a polynomial approximation of the matrix inverse that has better approximation properties and which also holds for regions other than a disk centered about identity or a multiple of the identity.

To see this, we further assume that the matrices X_j are drawn from a set U consisting of normal (that is, orthogonally diagonalizable) matrices whose eigenvalues lie in a compact set K that does not contain some open neighborhood of zero. This guarantees that all X_j, as well as their inverses, have uniformly bounded spectral norm.

Lemma 6.1. *Let U consist of the set of orthogonally diagonalizable matrices whose eigenvalues lie in a compact set $K \subset \mathbb{C}$ that does not contain 0, and assume that $\mathbb{C} \setminus K$ is connected. Then there exists a sequence of operator polynomials that approximate the function $X \mapsto X^{-1}$ uniformly on U.*

Proof. Since K does not contain 0, then the complex function $z \mapsto 1/z$ is holomorphic on some open set containing K. Because $\mathbb{C} \setminus K$ is connected, then we can apply the celebrated theorem of Mergelyan [80] to construct a sequence of polynomials $\{p_i(z)\}$ that uniformly approximates $z \mapsto 1/z$ on K. Then, by the holomorphic functional calculus, we have a sequence of operator polynomials $\{p_i(X)\}$ that uniformly approximates $X \mapsto X^{-1}$ on U. ∎

This basic result conveys that it is possible to find a polynomial p such that $p(X)h$ well-approximates $X^{-1}h$ under the assumption that X belongs to the set U. Next, we

construct a toy example that demonstrates how piecewise-linear activation functions σ in a operator recurrent network can be used to separate the space of matrices into separate regions, on each of which a different matrix polynomial is defined by the network.

Lemma 6.2. *Let U consist of real symmetric $n \times n$ matrices of norm at most 1, which are definite (that is, all eigenvalues share the same sign), and which are diagonally dominant. Furthermore, suppose that all matrices in U have inverses whose norms do not exceed $1/\varepsilon$ for some $\varepsilon > 0$. Then there exists an operator recurrent network f such that for every $X \in U$, and for some nonzero vector h,*

$$f(X) = \begin{cases} Xh, & X > 0, \\ 0, & X < 0. \end{cases} \tag{246}$$

In particular, there is a network that can distinguish X as either positive definite or negative definite.

Proof. Because each $X \in U$ is diagonally dominant, then $|X_{ii}| \geq \sum_{j \neq i} |X_{ij}|$. Then the disks D_i centered at X_{ii} of radius $\sum_{j \neq i} |X_{ij}|$ must lie either entirely in the left half of the complex plane, or the right half. It follows from the Gershgorin disk theorem (see [35]) that we can determine whether X is positive or negative definite by determining the sign of any of its diagonal entries.

Let $e_1 = [1, 0, \ldots, 0]^T$ be the first standard coordinate vector, and let E_{11} be the $n \times n$ matrix of all zeros except a 1 in the $(1, 1)$ entry. Then we observe that for any matrix X,

$$E_{11} X e_1 = \begin{bmatrix} X_{11} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{247}$$

If σ is the standard rectifier, then $\sigma(E_{11} X e_1)$ is a nonzero vector if and only if X is positive definite. Next, we claim that the vector

$$\big(\|h\|_2/\varepsilon\big) E_{11} X e_1 + Xh \tag{248}$$

has a positive number in its first component if $X > 0$. If $(v)_1$ denotes the first component of any vector v, then

$$\begin{aligned} \Big(\big(\|h\|_2/\varepsilon\big) E_{11} X e_1 + Xh\Big)_1 &= X_{11}\|h\|_2/\varepsilon + (Xh)_1 \\ &\geq X_{11}\|h\|_2/\varepsilon - \|Xh\|_2 \\ &\geq X_{11}\|h\|_2/\varepsilon - \|h\|_2 \\ &= \|h\|_2(X_{11}/\varepsilon - 1). \end{aligned} \tag{249}$$

Since the norm of X^{-1} is bounded by $1/\varepsilon$, then the smallest eigenvalue of X must be greater than ε. Therefore $X_{11} > \varepsilon$, so then the first entry of (248) must be positive. Next, we consider the value of the first entry of (248) when $X < 0$. Now we consider the sum $b = \sum_{j=1}^{n} b_j$, where $b_j = (\|h\|_2/\varepsilon) E_{jj} X e_j$. We claim it is negative. Using similar manipulations,

$$\begin{aligned}\big(\big(\|h\|_2/\varepsilon\big) E_{11} X e_1 + Xh\big)_1 &= X_{11}\|h\|_2/\varepsilon + (Xh)_1 \\ &\leq X_{11}\|h\|_2/\varepsilon + \|Xh\|_2 \\ &\leq X_{11}\|h\|_2/\varepsilon + \|h\|_2 \\ &= \|h\|_2(X_{11}/\varepsilon + 1). \end{aligned} \tag{250}$$

Since $X < 0$ and its inverse has norm bounded by $1/\varepsilon$, then its largest eigenvalue is at most $-\varepsilon$. Therefore $X_{11}/\varepsilon < -1$, so the result follows.

Now consider the vector

$$Xh + \frac{\|h\|_2}{\varepsilon} \sum_{j=1}^{n} E_{jj} X e_j. \tag{251}$$

From the above, every entry of this vector is either positive or negative, depending on whether X itself is positive or negative definite. Now, applying the standard rectifier σ to (251), the quantity is unchanged if $X > 0$, and is set to zero if $X < 0$. Finally, we consider the function

$$f(X) = -\sigma\left(\frac{\|h\|_2}{\varepsilon} \sum_{j=1}^{n} E_{jj} X e_j\right) + \sigma\left(Xh + \frac{\|h\|_2}{\varepsilon} \sum_{j=1}^{n} E_{jj} X e_j\right). \tag{252}$$

From our above, computations we observe that this f satisfies the property desired for the lemma, and furthermore, f can be constructed using layers of a general operator recurrent network. ∎

The purpose of this lemma is to produce an example that demonstrates how an operator recurrent network can distinguish between two sets of matrices, in particular those are that positive definite or negative definite, in a manner similar to how a standard rectifier network can determine whether a vector lies above or below a particular hyperplane. Next, utilizing the network constructed in the above lemma, we can show that an operator recurrent network exists that represents a different matrix polynomial depending on whether the input matrix is positive definite or negative definite.

Theorem 6.3. *Let U be the set of real symmetric matrices satisfying the same properties as those of Lemma 6.2. Then there exists an operator recurrent network f such that $f(X) = p_1(X)$ when $X > 0$ and $f(X) = p_2(X)$ when $X < 0$, such that p_1, p_2 are operator polynomials applied to the input vector h_0.*

Proof. First we construct a network f_1 representing polynomials of degree 1; in particular

$$f_1(X) = \begin{cases} A_1 X h_0 + h_0, & X > 0, \\ A_2 X h_0 + h_0, & X < 0, \end{cases} \tag{253}$$

where h_0 is some fixed vector. Let the operator network constructed in Lemma 6.2, using initial vector h_0, be relabeled at g. Then the above f_1 can be constructed by

$$f_1(X) = A_1 g(X) - A_2 g(-X) + h_0. \tag{254}$$

We observe that f_1 is a general operator recurrent network. To obtain matrix polynomials of higher degree, we perform a similar construction. By way of example, let us write down a piecewise degree 2 matrix polynomial by

$$f_2(X) = (C_1 X B_1 + A_1) g(X) - (C_2 X B_2 + A_2) g(-X) + h_0. \tag{255}$$

Then,

$$f_2(X) = \begin{cases} C_1 X B_1 X h_0 + A_1 X h_0 + h_0, & X > 0, \\ C_2 X B_2 X h_0 + A_2 X h_0 + h_0, & X < 0. \end{cases} \tag{256}$$

This construction can thus be easily extended to a network f_n, and in each such case, $f_n(X)$ restricted to either $\{X > 0\}$ or $\{X < 0\}$ yields an n-th degree operator polynomial. ∎

Lastly, we can use the above theorem, combined with Lemma 6.1, to construct an operator recurrent network that represents two different operator polynomials, each a different approximation to the matrix inverse, applied to the vector h. Note that the construction in the above theorem has no restrictions on the coefficient matrices A_1, A_2, B_1, B_2, etc. Since the operator polynomials arising from Lemma 6.1 have scalar coefficients, this is equivalent to the matrix-valued coefficients being multiples of the identity matrix, in which case they commute with all X. In this case, it is clear that we can arrange for values of A_1, A_2, B_1, B_2, and so forth, as to produce arbitrary scalar coefficients.

We reiterate that the purpose of Theorem 6.3 is not to give an optimal result for how operator recurrent networks can learn matrix inversion, but to provide a concrete illustration for how such networks can leverage its piecewise-polynomial nature, partitioning its input domain into distinct regions.

Finally, we note that Theorem 5.5 (i) and Theorem 5.6 (i) provide generalization estimates for training an operator recurrent neural network to represent matrix inversion. These imply estimates for sample complexity. Claims (ii) of these theorems, that is the improved estimates, however, are not generally applicable as we do not know whether the weight matrices can have rapidly decaying singular values (that is,

have small ℓ^p norms). Next, we consider an inverse problem for a wave equation in which case there is a solution algorithm which can be approximated by our neural network with such weight matrices that the improved generalization estimates, Theorem 5.5 (ii) and Theorem 5.6 (ii) are applicable.

7. Example: Operator recurrent network for an inverse problem with the wave equation

Here, we establish a direct relationship between operator recurrent neural networks and reconstruction pertaining to an inverse boundary value problems for the wave equation.

7.1. Analytic solution of inverse problem by boundary control method

We summarize the boundary control method used to solve an inverse problem for the wave equation. For the sake of simplicity, we present the one-dimensional case. We consider the wave equation with an unknown wave speed $c = c(x)$,

$$\begin{aligned} \big(\partial_t^2 - c(x)^2\partial_x^2\big)u(x,t) &= 0, && x \in \mathbb{R}_+,\ t \in \mathbb{R}_+, \\ \partial_x u(x,t)\big|_{x=0} &= h(t), \\ u(x,t)\big|_{t=0} &= 0, \quad \partial_t u(x,t)\big|_{t=0} = 0, && x \in \mathbb{R}_+, \end{aligned} \tag{257}$$

where we assume that c is a smooth positive function satisfying $c(0) = 1$. We denote the solutions of the wave equation with Neumann boundary value $h = h(t)$ by $u = u^h(x,t)$. Function h can be viewed as a boundary source. We assume that c is unknown, but that we are given the Neumann-to-Dirichlet map, $\mathcal{M}_{ND} = \mathcal{M}_{ND}^c$,

$$\mathcal{M}_{ND} h = u^h(x,t)\big|_{x=0}, \quad t \in (0, 2T). \tag{258}$$

This map is also called a response operator that maps the source to the boundary value of the produced wave. The Neumann-to-Dirichlet map is a smoothing operator of order one, that is, it is a bounded linear operator

$$\mathcal{M}_{ND} \colon L^2\big([0,2T]\big) \to H^1\big([0,2T]\big),$$

where $H^s([0,2T])$ are Sobolev spaces. An alternative to approximate $\mathcal{M}_{ND}$ by a matrix would be to choose suitable bases in the Hilbert spaces $L^2([0,2T])$ and $H^1([0,2T])$ and represent $\mathcal{M}_{ND}$ with respect to the relevant basis vectors. An alternative that avoids using two different bases, is to consider the bounded operator X_c,

$$X_c = \partial_t \mathcal{M}_{ND}^c \colon L^2\big([0,2T]\big) \to L^2\big([0,2T]\big), \tag{259}$$

and approximate this operator in a basis of the Hilbert space $L^2([0,2T])$. In this paper, we use this option and consider operator (259) as the given data.

The travel time of the waves from the boundary point 0 to the point x is given by

$$\tau(x) = \int_0^x \frac{\mathrm{d}x'}{c(x')}. \tag{260}$$

We consider the set $M = [0,\infty)$ as a manifold with boundary endowed with the distance function $d_M(x,y) = |\tau(x) - \tau(y)|$ that we call the travel time distance. We denote by $M(s) = \{x \in \mathbb{R}_+ : \tau(x) \le s\}$ the set of points which travel time to the boundary is at most s. The set $M(s)$ is called the domain of influence. The function τ is strictly increasing and we denote its inverse by

$$\chi = \tau^{-1} : [0,\infty) \to [0,\infty),$$

that is, $\tau(\chi(s)) = s$. The function $\chi(s)$ is called the travel time coordinate, because for every time s it gives a point x whose travel time to the boundary is s. The function

$$Z(s) = c(\chi(s))$$

is the wave speed in the medium represented in the travel time coordinates and by [48], formula (22), it uniquely determines the wave speed $c(x)$ in Euclidean coordinates. Thus, it also determines the data operator X_c, and thus we can define a nonlinear operator

$$\mathcal{F} : Z \to X_c. \tag{261}$$

In the study of the inverse problems, this map is called the direct map. Below, we approximate the function $Z(s)$ by a finite-dimensional vector $z = (Z(s_j))_{j=1}^m$, where s_j are points in the interval $[0,T]$. Also, X_c will be approximated by a finite-dimensional matrix $X = (\langle X_c \psi_k \rangle)_{j,k=1}^n$, we obtain a finite-dimensional direct map (see (4)),

$$F : B^m(z^{(0)}, \rho_0) \to \mathbb{R}^{n\times n}, \quad F(z) = X, \tag{262}$$

where $B^m(z^{(0)},\rho_0) \subset \mathbb{R}^m$ is a ball centered at a vector $z^{(0)}$ having positive elements.

Next, we return to the continuous setting we explain how the data operator X_c measured on the boundary can be used to compute the wave speed function in the travel time coordinate, that is, $c(\chi(s))$, and after that, how this reconstruction process can be approximated by an algorithm that has the same form as the neural network in (9)–(10).

We define

$$Sf(t) = \int_0^t f(t')\,\mathrm{d}t'. \tag{263}$$

We observe that $\partial_t \mathcal{M}^c_{ND} = \mathcal{M}^c_{ND}\partial_t$, and, hence, we have $\mathcal{M}^c_{ND} = SX_c = X_c S$.

We denote

$$\langle u^f(T), u^h(T)\rangle_{L^2(M)} = \int_M u^f(x,T)u^h(x,T)c(x)^{-2}\,\mathrm{d}x \tag{264}$$

and $\|u^f(T)\|_{L^2(M)} = \langle u^f(T), u^f(T)\rangle_{L^2(M)}^{1/2}$. By the Blagovestchenskii identity (see for example [17,48]), we have

$$\langle u^f(T), u^h(T)\rangle_{L^2(M)} = \int_{[0,2T]} (Kf)(t)h(t)\,\mathrm{d}t, \tag{265}$$

while

$$\langle u^f(T), 1\rangle_{L^2(M)} = \int_{[0,2T]} f(t)\Phi_T(t)\,\mathrm{d}t, \tag{266}$$

where

$$K = JSX_c - RX_cSRJ, \tag{267}$$

$$Rf(t) = f(2T-t) \qquad \text{“time reversal operator”}, \tag{268}$$

$$Jf(t) = \frac{1}{2}\mathbf{1}_{[0,T]}(t)\int_t^{2T-t} f(s)\,\mathrm{d}s \quad \text{“time filter”}, \tag{269}$$

$$\Phi_T(t) = (T-t)\mathbf{1}_{[0,T]}(t). \tag{270}$$

Here, $J\colon L^2([0,2T]) \to L^2([0,2T])$ and $R\colon L^2([0,2T]) \to L^2([0,2T])$.

In the boundary control method the first task is to approximately solve the following blind control problem: Can we find a boundary source f such that

$$u^f(x,T) \approx \mathbf{1}_{M(s)}(x)\,? \tag{271}$$

Here, $\mathbf{1}_A$ is the indicator function of the set A, that is $\mathbf{1}_A(x) = 1$ for $x \in A$, zero otherwise. The problem is called a blind control problem because we do not know the wave speed $c(x)$ that determines how the waves propagate in the medium, and we aim to control the value of the wave at the time $t = T$. This control problem can be solved via regularized minimization problems. In [49] the problem was solved using Tikhonov regularization, while in this paper we consider sparse regularization techniques that are closely related to neural networks.

7.2. Variational formulation and sparse regularization

In sparse regularization, we represent the function $f(t) \in L^2([0,2T])$ in terms of orthogonal functions $\psi_j(t) \in L^2([0,2T])$, $j = 1,2,\ldots,n$, where $n \in \mathbb{N}_+ \cup \{\infty\}$, such that

$$\left\| \sum_{j=1}^n f_j\psi_j \right\|_{L^2([0,2T])} \le C_0 \sum_{j=1}^n |f_j|. \tag{272}$$

Here, the case $n < \infty$ corresponds to numerical approximations with a finite set of basis functions, and the case $n = \infty$ corresponds to the ideal continuous model; we consider these two cases simultaneously. When $n = \infty$, we assume that the functions $\psi_j(t)$, $j = 1, 2, \ldots$ span a dense set in $L^2([0, 2T])$.

For $\mathbf{f} = (f_j)_{j=1}^n$, we denote

$$f(t) = (B\mathbf{f})(t) = \sum_{j=1}^{n} f_j \psi_j(t). \tag{273}$$

For $n = \infty$, we denote $\ell_n^1 = \ell^1$ and $\|\mathbf{f}\|_1 = \sum_{j=1}^{\infty} |f_j|$. For $n < \infty$, we denote $\ell_n^1 = \mathbb{R}^n$ and $\|\mathbf{f}\|_1 = \sum_{j=1}^{n} |f_j|$.

We seek solutions for which $\mathbf{f} = (f_j)_{j=1}^n \in \ell_n^1$ is a sparse vector. Such sparse vectors correspond to sources that are generated by a small number of basis functions ψ_j. We let $P_s : L^2([0, 2T]) \to L^2([0, 2T])$ denote the multiplication by the indicator function of the interval $[0, s]$, that is, $(P_s f)(t) = \mathbf{1}_{[0,s]}(t)\, f(t)$.

To obtain approximate solutions of control problem (271), we consider an ℓ_n^1-regularized version of the minimization problem,

$$\min_{\mathbf{f} \in \ell_n^1} \|u^{P_s B\mathbf{f}}(\cdot, T) - 1\|_{L^2(M)}^2 + \alpha \|\mathbf{f}\|_1, \tag{274}$$

where $\alpha > 0$ is a regularization parameter. This minimization problem is equivalent to finding $\mathbf{f}$ that solves

$$\min_{\mathbf{f} \in \ell_n^1} \langle K P_s B\mathbf{f}, P_s B\mathbf{f} \rangle_{L^2([0,2T])} - 2\langle P_s B\mathbf{f}, \Phi_T \rangle_{L^2([0,2T])} + \alpha \|\mathbf{f}\|_1, \tag{275}$$

where $K = JSX_c - RX_cSRJ$ as before. We denote the solution of this minimization problem by $\mathbf{f}_{\alpha,s}$.

Minimization problem (275) can be solved using the Iterated Soft Thresholding Algorithm (ISTA) [27]. The standard ISTA algorithm is the iteration

$$\mathbf{f}_s^{(m+1)} = \sigma_\alpha\big(\mathbf{f}_s^{(m)} - B^* P_s (JSX_c - RX_cSRJ) P_s B\mathbf{f}_s^{(m)} + B^* P_s \Phi_T\big), \quad m = 1, 2, \ldots, \tag{276}$$

where $\mathbf{f}_s^{(m)} \in \ell_n^1$, $\mathbf{f}_s^{(0)} = 0$ and σ_α is the soft thresholding operator, given by

$$\sigma_\alpha(x) = \max(0, x - \alpha) - \max(0, -x - \alpha) = \mathrm{ReLU}(x - \alpha) - \mathrm{ReLU}(-x - \alpha) \tag{277}$$

for $x \in \mathbb{R}$; for a vector $x = (x_j)_{j=1}^n$ it is defined componentwise.

By [27],

$$\mathbf{f}_{\alpha,s} = \lim_{m \to \infty} \mathbf{f}_s^{(m)}, \tag{278}$$

where the limit is taken in ℓ^1_n, and the convergence in this limit is exponential. We denote $f_{\alpha,s} = B\mathbf{f}_{\alpha,s}$. When $n = \infty$, we have by Appendix A that

$$\lim_{\alpha\to 0} u^{f_{\alpha,s}}(\cdot, T) = \mathbf{1}_{M(s)}(\cdot) \tag{279}$$

in $L^2(M)$.

7.3. Reconstruction

When the minimizers $f_{\alpha,s}$ are found for all $s \in [0, T]$ with small $\alpha > 0$, we continue the reconstruction of the wave speed by computing the volumes of the domains of influence,

$$V(s) = \|1_{M(s)}\|^2_{L^2(M)} = \lim_{\alpha\to 0} \langle u^{f_{\alpha,s}}(T), 1\rangle_{L^2(M)} = \lim_{\alpha\to 0} \langle f_{\alpha,s}, \Phi_T\rangle_{L^2([0,2T])}, \tag{280}$$

where $s \in [0, T]$. We note that $M(s) = [0, \chi(s)]$. In particular, $V(s)$ determines the wave speed in the travel time coordinate,

$$v(s) = \frac{1}{\partial_s V(s)}. \tag{281}$$

That is,

$$v(s) = c(\chi(s)), \quad \chi(s) = \int_0^s v(t)\,\mathrm{d}t. \tag{282}$$

When $v(s)$ is obtained, we can find the wave speed $c(x)$ also in the Euclidean coordinates using the formula,

$$c(x) = v(\chi^{-1}(x)). \tag{283}$$

However, in our reconstruction, we consider the function $v(s)$ as the final result.

7.4. Identification with operator recurrent networks

The ISTA algorithm iteration (278) produces $\mathbf{f}_s^{(n_0)}$ after n_0 steps. We observe that this iteration can be expressed by defining

$$\mathbf{h}_s^{(3m+1)} = \mathbf{f}_s^{(m)}, \quad \mathbf{h}_s^{(3m+2)} = P_s B\mathbf{f}_s^{(m)}, \quad \mathbf{h}_s^{(3m+3)} = RJP_s B\mathbf{f}_s^{(m)} \tag{284}$$

and viewing it as the operator recurrent neural network $X \mapsto f_{(\alpha,s)}(X)$, where

$$f_{(\alpha,s)}(X) = \mathbf{h}_s^{(3\ell_0+1)}, \tag{285}$$

in which, for $m = 0, 1, \dots, \ell_0$,

$$\mathbf{h}_s^{(3m+3+1)} = \sigma_\alpha\big(I\,\mathbf{h}_s^{(3m+1)} - B^* P_s J S X \mathbf{h}_s^{(3m+3)} + B^* P_s R X \mathbf{h}_s^{(3m+2)} + B^* P_s \Phi_T\big), \tag{286}$$

$$\mathbf{h}_s^{(3m+3)} = P_s B \mathbf{h}_s^{(3m+1)}, \tag{287}$$

$$\mathbf{h}_s^{(3m+2)} = SRJP_s B \mathbf{h}_s^{(3m+1)} \tag{288}$$

with the initial state $\mathbf{h}_s^{(1)} = 0$. This is motivated by the notion of unrolling. As the low-pass filter operator J and the integrator S are compact operators in $L^2(0, 2T)$, and moreover, the operators $SRJP_s$ and $P_s JS$ appearing above are in a Schatten class $\mathcal{S}_p$ with index $p > 1/2$, we approximate the above algorithm as a neural network with weight matrices of the form (21), and

$$A^\ell = A^{\ell,(0)} + A_\theta^{\ell,(1)}, \quad B^\ell = B^{\ell,(0)} + B_\theta^{\ell,(1)}, \tag{289}$$

where the $A^{\ell,(0)}$ and $B^{\ell,(0)}$, considered as fixed operators in a suitable basis are zero operators, identity operators, projectors P_s or $P_s R$, and $A_\theta^{\ell,(1)}$ and $B_\theta^{\ell,(1)}$ are operators $SRJP_s B$ and $B^* P_s JS$ appearing in (286)–(288), which are Schatten class operators, in $\mathcal{S}_p$ with index $p > 1/2$. When $n < \infty$, the generalized Hölder inequality implies for a matrix $A \in \mathbb{R}^{n\times n}$ and $p > 1/2$ that

$$\|A\|_{\mathcal{S}_{1/2}(\mathbb{R}^{n\times n})} \le n^{1/r} \|A\|_{\mathcal{S}_p(\mathbb{R}^{n\times n})}, \tag{290}$$

where $r = p/(2p-1)$. Furthermore, $B^* P_s \Phi_T$ in (286)–(288) are the bias vectors.

We have included the fixed operators $A^{\ell,(0)}$ and $B^{\ell,(0)}$ in the network architecture, because then for any given value of $s \in [0, T]$ the computation of $\mathbf{f}_s^{(\ell_0)}$ in the discretized boundary control method can be written as an operator recurrent network $f_\theta^s(X)$ of the form (10). Here, parameters θ, define the operator recurrent networks $f_\theta^s(X)$, depend on s and θ. Also, by (290), when $n < \infty$, it follows that the neural network (286)–(288) of depth $3\ell_0 + 1$ has the sparsity bound

$$\begin{aligned}\mathcal{R}(\theta_s) &\le \ell_0\big(\|B^* SRJP_s B\|_{\mathcal{S}_{1/2}(\mathbb{R}^{n\times n})} + \|B^* P_s JSB\|_{\mathcal{S}_{1/2}(\mathbb{R}^{n\times n})}\big)\\ &\le C_r \ell_0 n^{1/r},\end{aligned} \tag{291}$$

where $r < \infty$ is arbitrary and C_r depends on r.

In the discretized boundary control method we compute the functions $\mathbf{f}_s^{(\ell_0)} = f_{(\alpha,s)}(X)$ that approximate functions f_{α,s_j}, for parameter values $s = s_j$, $j = 1, 2, \dots, K$, given by

$$s_j = jT/K \in [0, T]. \tag{292}$$

Note that $\mathbf{f}_s^{(\ell_0)}$ converge to the functions f_{α,s_j}, as the depth of the neural network, ℓ_0 tends to infinity. Then, we define analogously to (1), we denote

$$f_\theta^j(X) = f_{(\alpha,s_j)}(X) \in \mathbb{R}^n, \qquad j = 1,2,\dots,K, \tag{293}$$

$$\mathbf{f}(X) = \big(f_\theta^1(X),\dots,f_\theta^K(X)\big) \in (\mathbb{R}^n)^K. \tag{294}$$

We also denote $s_0 = 0$ and $f_\theta^0(X) = 0$.

We may add one linear layer G_1 into the neural network that computes the derivative in (281) using finite differences,

$$\begin{aligned} D_\alpha(s_j) &:= \frac{1}{v_\alpha(s_j)} = \frac{V_\alpha(s_j) - V_\alpha(s_{j-1})}{s_j - s_{j-1}} \\ &= \frac{1}{s_j - s_{j-1}}\big(\langle f_{\alpha,s_j}, \Phi_T\rangle_{L^2([0,2T])} - \langle f_{\alpha,s_{j-1}}, \Phi_T\rangle_{L^2([0,2T])}\big), \end{aligned} \tag{295}$$

where $j = 1,2,\dots,K$ and

$$V_\alpha(s_j) = \langle f_{\alpha,s_j}, \Phi_T\rangle_{L^2([0,2T])}, \tag{296}$$

cf. (280). We denote $G_1(f_{\alpha,s_1}, f_{\alpha,s_2},\dots,f_{\alpha,s_K}) = (D_\alpha(s_1),\dots,D_\alpha(s_K))$.

Approximating the componentwise function $s \to s^{-1}$ via a standard neural network $G_2\colon \mathbb{R}^K \to \mathbb{R}^K$, of the form (6)–(8), we obtain a neural network

$$F_{\vec\theta}, \quad \vec\theta = (\theta_{s_1},\dots,\theta_{s_K})$$

of the form

$$H_{\vec\theta}(X) = G_2\big(G_1\big(f_\theta^1(X), f_\theta^2(X),\dots,f_\theta^K(X)\big)\big), \tag{297}$$

which output approximates the values $v(s_j) = c(\chi(s_j))$, $j = 1,2,\dots,K$. By using [94], steps (296) and (295) (see also Theorem 2.1), and the function $s \to s^{-1}$ can be approximated by a neural network G_2 of the form (9)–(10). We observe that formula (297) is analogous to (2). Finally, by (291), the neural network $F_{\vec\theta}$ in (297) can be written as an operator recurrent network that has the sparsity bound

$$\mathcal{R}(\vec\theta) \le C_r' K \ell_0 n^{1/r}, \tag{298}$$

where $r < \infty$ is arbitrary and C_r' depends on r.

The low-pass filter operator J is in a Schatten class. Here, we show that the low-pass filter operator J used above is in a Schatten class with $p>1$. We consider the extension of low pass filter operator $J\colon L^2(0,2T) \to L^2(0,2T)$. It can be written as

$$J = A^{-1/2} \circ (A^{1/2} \circ J), \tag{299}$$

where

$$A = -\frac{\mathrm{d}^2}{\mathrm{d}x^2} + 1,$$

and where $\frac{\mathrm{d}^2}{\mathrm{d}x^2}$ is Laplace operator defined as an unbounded self-adjoint operator in $L^2([0, 2T])$ with Neumann boundary condition,

$$\mathcal{D}(A) = \Big\{ f \in H^2\big([0, 2T]\big) : \frac{\mathrm{d}f}{\mathrm{d}x}(0) = 0,\ \frac{\mathrm{d}f}{\mathrm{d}x}(2T) = 0 \Big\},$$

where $H^s([0, 2T])$ are Sobolev spaces, $\mathcal{D}(A^{1/2}) = H^1([0, 2T])$, and

$$A^{1/2} \circ J^* \colon L^2\big([0, 2T]\big) \to L^2\big([0, 2T]\big)$$

is a bounded operator. As the eigenvalues of A are of the form $\lambda_j = c_T j^2 + 1$, the eigenvalues of $A^{-1/2} \colon L^2([0, 2T]) \to L^2([0, 2T])$ are $(c_T j^2 + 1)^{-1/2}$, and, hence,

$$A^{-1/2} \colon L^2\big([0, 2T]\big) \to L^2\big([0, 2T]\big)$$

is in the Schatten class $\mathcal{S}_p(L^2(0, 2T))$ with $p > 1$. As the Schatten classes are operator ideals, this implies that

$$J \in \mathcal{S}_p(L^2(0, 2T)) \quad \text{with } p > 1. \tag{300}$$

In the same way, we observe that $S \in \mathcal{S}_p(L^2(0, 2T))$ with $p > 1$ and hence the operators SJ and SRJ appearing in (276) satisfy $SJ, SRJ \in \mathcal{S}_p(L^2(0, 2T))$ with $p > 1/2$. Thus, when we approximate these operators by matrices representing operators in a space spanned by finitely many basis functions ψ_j, it is natural to assume that the $\mathcal{S}_{1/2}$-norms of these matrices are bounded with some relatively small constants.

Furthermore, we note that the "bias functions" Φ_T are in the Sobolev space $H^1([0, 2T])$, that is, a compact subset of $L^2([0, 2T])$ and therefore Φ_T can be approximated by a vector which coordinates are a sparse sequence.

In summary, the boundary control method can be approximated by an operator recurrent network of the form (9)–(10), where the weight operators A and B are either Schatten class operators (which we can train with sparsity regularization to obtain a better algorithm), or simple operators, such as the time-reversal operator R or the projector P_s that we may consider as fixed in the neural network and that we do not train. The time reversal operator is extensively used in imaging applications; see for example [9, 19]. Also, the bias vectors can be approximated by sparse vectors. Furthermore, we observe that if we consider sparse regularization leading to activation functions that are linear combinations of ReLU functions, we do not specify in the neural network formulation what the basis function ψ_j are. Thus, the training of the neural network also leads to finding a basis that is optimal for sparse regularization.

7.5. Discretization error versus depth and width of the network

Here, we estimate the error in the point of departure of the network design in the main body of this paper. By stability and error analyses of the boundary control method, we can estimate how well the discretized boundary control method works and what are the error estimates for all wave speeds c in the set

$$\mathcal{V}^3 = \left\{c \in C^3(M) : C_0 \leq c(x) \leq C_1,\ \|c\|_{C^3(M)} \leq M,\ \operatorname{supp}(c-1) \subset I_0\right\}, \quad (301)$$

where $I_0 \subset \mathbb{R}_+$ is a compact interval. We use C as a generic constant which depends on parameters of the space $\mathcal{V}^3$ and which value may be different in each appearance.

We consider the discretization of analytical algorithms that reconstruct $c(x)$, with error $C\delta$ in the $L^\infty(M)$-norm, from the map X, or from the map $\mathcal{M}_{ND}$. To this end, we denote $\varepsilon = \delta^m$, where $m = 270$. In [48], it was shown for the discretized boundary control method that we can compute the wave speed with error $C\delta = C\varepsilon^\gamma$, with Hölder exponent $\gamma = 1/m$, when we discretize the time interval $[0, T]$ with a grid of $N_0(\varepsilon) = C\varepsilon^{-4/7}$ points and measurement operator $\mathcal{M}_{ND}$ is given with an error ε in the operator norm in $L^2(0, 2T)$. In this paper we omit the analysis of the measurement errors in the Neumann-to-Dirichlet map, and consider only the discretization error, that is, the error caused by approximating the infinite dimensional operators by finite dimensional matrices. The discrete BC-method in [48] requires solving $K \leq C\varepsilon^{-1/18} = C\delta^{-270/18}$ minimization problems of the form (274), that is, for each value of s_j in (292). Moreover, as by [27] the iteration in the ISTA algorithm has exponential convergence to the solution of the minimization problem, we conclude that the linear system can be solved with accuracy $C\varepsilon$ using an iteration of $C\log(\varepsilon^{-1})$ steps that each require a composition of linear operators and the operator $\mathcal{M}_{ND}$.

From the discretization error estimates we may deduce estimates for the depth and width of the operator recurrent neural network based on a scenario without training: The upper bound for the depth is L and the upper bound for the width n is

$$L \leq C\log(\delta^{-1}), \quad n \leq C\varepsilon^{-4/7-1/18} \leq C\varepsilon^{-9/14} \leq C\delta^{-175}. \quad (302)$$

Moreover, as $K \leq C\varepsilon^{-1/18}$, we see that this neural network can be written as $H_{\vec{\theta}}$ given in (297) that has the sparsity bound $\mathcal{R}(\vec{\theta})$ and accuracy bound ε_0 that given by

$$\mathcal{R}(\vec{\theta}) \leq C'KLn^{1/r} \leq C'\delta^{-270/18} \cdot \log(\delta^{-1}) \cdot \delta^{-175/r} \leq C''\delta^{-16}, \quad (303)$$

$$\varepsilon_0 = C\delta, \quad (304)$$

where $r < \infty$ is arbitrary and C, C' and C'' depend on r. Consider now the case when a priori distribution of the data is supported in the set of the Neumann-to-Dirichlet maps corresponding to the wave speeds $c \in \mathcal{V}^3$. Then the above implies, in terms of

Definition 4.1, that the map $X_c \to c$, solving the inverse problem for the wave equation, can be approximated with accuracy $\varepsilon_0 = C\delta$ by a neural network $X \to F_{\vec{\theta}}(X)$, where $\vec{\theta}$ has the sparsity bound $R_0 \leq C''\delta^{-16}$. Note that here we do not require that the absolute values of the components of the vector $\vec{\theta}$ are bounded by one. However, this happens if T or the parameters of the set $\mathcal{V}^3$ are sufficiently small.

The above worst case estimate gives also an upper bound how well an optimally trained neural network performs. However, if one is interested in reconstructing a wave speed c in a subset $\mathcal{W} \subset \mathcal{V}^3$ and uses training data sampled from the set $\mathcal{W}$, then the trained network is by our analysis close to an optimal neural network that will most likely perform better than the neural network with a priori determined parameters approximating the boundary control method for three reasons: First, the optimal neural network is optimized to the subset $\mathcal{W}$, not the larger class $\mathcal{V}^3$. Second, the neural network is based on theoretical estimates that prove worst case errors in all substeps. Third, the algorithm with a priori determined parameters does not estimate the average error in the reconstruction, but absolute error and thus the optimal neural network that optimizes the expected error may perform better.

A. Time reversal algorithm with sparse regularization

In this appendix we consider how the results in [17,48] can be generalized in the case when one regularizes the ℓ^1 term of the source term.

Let $B\colon \ell^1 \to L^2(0,T)$ be an operator such that there is $C_0 > 0$ such that

$$\|Bf\|_{L^2(0,T)} \leq C_0 \|f\|_{\ell^1}.$$

For example, when $s > 1/2$, the Besov space $B^s_{11}(S^1)$ on the unit circle S^1 is a subset of $L^2(S^1)$ (that is isomorphic to $L^2([0,T])$). Moreover, there is an isomorphism $B\colon \ell^1 \to B^s_{11}(S^1)$ of the form (273), where ψ_j are wavelets [88].

Theorem A.1. *Assume that $B(\ell^1) \subset L^2(0,T)$ is a dense subset. Let $r \in [0,T]$ and $\alpha > 0$. Let us define*

$$S_r = \big\{ f \in L^2(0,T) : \operatorname{supp}(f) \subset [T-r, T] \big\}. \tag{A.1}$$

Then the regularized minimization problem

$$\min_{f \in \ell^1} \big(\langle Bf, KBf \rangle_{L^2(0,T)} - 2\langle Bf, \Phi_T \rangle_{L^2(0,T)} + \alpha \|f\|_{\ell^1} \big) \tag{A.2}$$

has a minimizer $f_{\alpha,r}$. Moreover, $u^{Bf_{\alpha,r}}(T)$ converges to the indicator function of the domain of influence,

$$\lim_{\alpha \to 0} \|u^{Bf_{\alpha,r}}(T) - 1_{M(r)}\|_{L^2(M;dV)} = 0. \tag{A.3}$$

Proof of Theorem A.1. Let $\alpha > 0$ and let $f \in \ell^1$. We define the energy function

$$E(f) := \langle P_s Bf, KP_s Bf \rangle_{L^2(0,T)} - 2\langle P_s Bf, \Phi_T \rangle_{L^2(0,T)} + \alpha \|f\|_{\ell^1}. \tag{A.4}$$

The finite speed of wave propagation implies $\operatorname{supp}(u^{P_s Bf}(T)) \subset M(r)$. Moreover, by the Blagovestchenskii formula we have

$$E(f) = \|u^{P_s Bf}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} - \|1_{M(r)}\|^2_{L^2(M;dV)} + \alpha\|f\|_{\ell^1}. \tag{A.5}$$

Let $(f_j)_{j=1}^\infty \subset \ell^1$ be such that

$$\lim_{j\to\infty} E(f_j) = \inf_{f\in\ell^1} E(f) =: E^*.$$

Then,

$$\alpha\|f_j\|_{\ell^1} \le E(f_j) + \|1_{M(r)}\|^2_{L^2(M;dV)} \le E^* + \operatorname{vol}(M) = E^{**},$$

and we see that $(f_j)_{j=1}^\infty$ is bounded in ℓ^1 and satisfies $\|f_j\|_{\ell^1} \le \alpha^{-1}E^{**}$.

The space ℓ^1 is the dual of the space c_0 of sequences converging to zero. Thus, by Banach–Alaoglu theorem, Hilbert space, there is a subsequence of $(f_j)_{j=1}^\infty$ that weak*-converges in ℓ^1. Let us denote the limit by $f_\infty \in \ell^1$ and the subsequence still by $(f_j)_{j=1}^\infty$.

When $y = (y_i)_{i=1}^\infty \in \ell^1$, we denote $p_k(y) = (y_i)_{i=1}^k \in \mathbb{R}^k$. Now, we see that as $(f_j)_{j=1}^\infty$ weak*-converges to f_∞ in ℓ^1, we have for all vectors $g_k = (\delta_{jk})_{j=1}^\infty \in c_0$ such that

$$(f_j, g_k)_{\ell^1,c_0} \to (f_\infty, g_k)_{\ell^1,c_0} \quad \text{as } j \to \infty.$$

Hence, we see that $p_k(f_j)$ converge to $p_k(f_\infty)$ and for all k and

$$\sum_{i=1}^k |(f_\infty)_i| \le \lim_{j\to\infty} \sum_{i=1}^k |(f_j)_i| \le \|f_j\|_{\ell^1} \le \alpha^{-1}E^{**}.$$

Taking limit $k \to \infty$ we see that $\|f_\infty\|_{\ell^1} \le \alpha^{-1}E^{**}$.

The map $U_T\colon L^2(0,T) \to H^1(M)$, mapping $U_T\colon h \mapsto u^h(T)$, is bounded. The embedding $I\colon H^1(M) \hookrightarrow L^2(M)$ is compact, and thus U_T is a compact operator

$$U_T\colon L^2(0,T) \to L^2(M).$$

As $P_s Bf_j$ is a bounded sequence in $L^2(0,T)$, we see that by replacing the sequence $(f_j)_{j=1}^\infty$ by its suitable subsequence, we can assume that

$$u^{P_s Bf_j}(T) \to u^{P_s Bf_\infty}(T)$$

in $L^2(M)$ as $j \to \infty$.

The above yields that

$$\begin{aligned} E(f_\infty) &= \lim_{j\to\infty} \|u^{P_s Bf_j}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} - \|1_{M(r)}\|^2_{L^2(M;dV)} + \alpha \|f_\infty\|_{\ell^1} \\ &\le \lim_{j\to\infty} \|u^{P_s Bf_j}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} - \|1_{M(r)}\|^2_{L^2(M;dV)} + \alpha \liminf_{j\to\infty} \|f_j\|_{\ell^1} \\ &= \liminf_{j\to\infty} E(f_j) = \inf_{f\in S_r} E(f), \end{aligned}$$

and thus $f_\infty \in \ell^1$ is a minimizer for (A.4).

As $B(\ell^1) \subset L^2(0,T)$ is a dense subset, we see by using Tataru's approximate controllability theorem that

$$\left\{u^{P_r Bf}(T) \in L^2(M(r)) : f \in \ell^1\right\}$$

is dense in $L^2(M(r))$. Let $\delta > 0$. For $\varepsilon = \delta^2/2$, let us choose $f_\varepsilon \in \ell^1$, such that

$$\|u^{P_r Bf_\varepsilon}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} \le \varepsilon. \tag{A.6}$$

Using (A.5) we have

$$\|u^{P_r Bf_{\alpha,r}}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} \le E(f_{\alpha,r}) + \|1_{M(r)}\|^2_{L^2(M;dV)}.$$

Because $E(f_{\alpha,r}) \le E(f_\varepsilon)$ we have

$$\begin{aligned} \|u^{P_r Bf_{\alpha,r}}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} &\le \|u^{P_r Bf_\varepsilon}(T) - 1_{M(r)}\|^2_{L^2(M;dV)} + \alpha \|f_\varepsilon\|_{\ell^1} \\ &\le \varepsilon + \alpha \|f_\varepsilon\|_{\ell^1}. \end{aligned}$$

When $0 < \alpha < \alpha_\delta = \delta^2/2\|f_\varepsilon\|_{\ell^1}$, we see that

$$\|u^{P_r Bf_{\alpha,r}}(T) - 1_{M(r)}\|_{L^2(M;dV)} \le \left(\varepsilon + \alpha \|f_\varepsilon\|_{\ell^1}\right)^{\frac{1}{2}} = \delta.$$

Thus,

$$\lim_{\alpha\to 0} \|u^{P_r Bf_{\alpha,r}}(T) - 1_{M(r)}\|_{L^2(M;dV)} = 0. \qquad \blacksquare$$

B. Conditional expectation as a projector

In this appendix, we recall the definition and the properties of conditional expectations using σ-algebras discussed in detail in [44, Ch. 5] and [29].

Let $(\Omega, \Sigma, \mathbb{P})$ be an complete probability space and $Z\colon \Omega \to \mathbb{R}^m$ be a random variable. Below, we consider the case when Z is $\mathbb{R}$-valued, that is, $m = 1$, but the discussion below generalizes in a straight forward way for $m \in \mathbb{Z}_+$.

Let $\mathcal{B}_Z \subset \Sigma$ be a σ-algebra generated by the random variable Z, that is, the smallest σ-algebra that contains the sets $Z^{-1}(S) \subset \Omega$, where $S \subset \mathbb{R}$ is an open set We recall that when $F\colon \Omega \to \mathbb{R}$ satisfies $F = F(\omega) \in L^1(\Omega; d\mathbb{P})$, then $\mathbb{E}(F|\mathcal{B}_Z)(\omega)$ is the $\mathcal{B}_Z$-measurable random variable that satisfies

$$\int_A \mathbb{E}(F \mid \mathcal{B}_Z)(\omega)\, d\mathbb{P}(\omega) = \int_A F(\omega)\, d\mathbb{P}(\omega) \tag{B.1}$$

for all sets $A \in \mathcal{B}_Z$.

Roughly speaking, $\mathbb{E}(F \mid \mathcal{B}_Z)$ denotes the expectation of a random variable $F = F(\omega)$ under the condition that Z is known. More precisely, by [29, Section 10.1 and Theorem 4.2.8], there is a measurable function $g_F\colon \mathbb{R} \to \mathbb{R}$ such that

$$\mathbb{E}(F \mid \mathcal{B}_Z) = g_F(Z) = g_F(Z(\omega)), \quad \mathbb{P}\text{-a.e.}, \tag{B.2}$$

that is, $\mathbb{E}(F \mid \mathcal{B}_Z)$ can be considered as deterministic function of Z. To simplify notations, one uses for the conditional expectation of the random variable F, under the condition that Z is given, the notation

$$\mathbb{E}(F \mid \mathcal{B}_Z) = \mathbb{E}(F \mid Z), \tag{B.3}$$

where the right-hand side is in fact equal to $g_F(Z)$. We emphasize that as Z is a random variable, also $\mathbb{E}(F \mid \mathcal{B}_Z) = \mathbb{E}(F \mid Z)$ is a random variable.

Let $H = L^2(\Omega; \mathcal{B}_Z, d\mathbb{P})$ be the set of $\mathbb{R}$-valued functions $u = u(\omega)$ that satisfy $u \in L^2(\Omega; \Sigma, d\mathbb{P})$ and are $\mathcal{B}_Z$-measurable. Observe that $H \subset L^2(\Omega; \Sigma, d\mathbb{P})$ is a closed subspace of the Hilbert space $L^2(\Omega; \Sigma, d\mathbb{P})$.

By [29, Theorem 4.2.8], for any $u \in L^2(\Omega; \mathcal{B}_Z, d\mathbb{P})$ there is a Borel-measurable function g such that $u(\omega) = g(Z(\omega))$, that is, $u = g \circ Z$, $\mathbb{P}$-a.e. in Ω.

By (B.1), we have

$$\langle \mathbb{E}(F|\mathcal{B}_Z), g(\omega)\rangle_{L^2(\Omega;\Sigma,d\mathbb{P})} = \langle F, g(\omega)\rangle_{L^2(\Omega;\Sigma,d\mathbb{P})} \tag{B.4}$$

for indicator functions $g = \mathbf{1}_A$ with all sets $A \in \mathcal{B}_Z$. As such indicator functions span a dense set in H, we have that (B.4) holds for all $g \in H$. As $\mathbb{E}(F|\mathcal{B}_Z)(\omega) \in H$, this yields that

$$\mathbb{E}(F|\mathcal{B}_Z) = P_H F, \tag{B.5}$$

where

$$P_H\colon L^2(\Omega; \Sigma, d\mathbb{P}) \to L^2(\Omega; \Sigma, d\mathbb{P}) \tag{B.6}$$

is the orthogonal projector onto the set $H = L^2(\Omega; \mathcal{B}_Z, d\mathbb{P})$, that is, $\mathrm{Ran}(P_H) = H$. In the main text we use extensively that fact that

$$P_H F = \arg\min \|F - u\|^2_{L^2(\Omega;\Sigma,d\mathbb{P})} \tag{B.7}$$

subject to the condition $u \in H = L^2(\Omega; \mathcal{B}_Z, d\mathbb{P})$.

Acknowledgments. The authors are indebted to an anonymous referee for many suggestions that greatly improved the paper.

Funding. M. V. d. H. gratefully acknowledges support from the Department of Energy under grant DE-SC0020345, the Simons Foundation under the MATH + X program, and the corporate members of the Geo-Mathematical Imaging Group at Rice University. C. W. was funded by Total. M. L. was supported by Finnish Centre of Excellence in Inverse Modelling and Imaging and Academy of Finland grants 284715, 312110.

References

[1] K. Abraham and R. Nickl, On statistical Calderón problems. *Math. Stat. Learn.* **2** (2019), no. 2, 165–216 Zbl 1445.35144 MR 4130599

[2] J. Adler and O. Öktem, Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems* **33** (2017), no. 12, Art. ID 124007 Zbl 1394.92070 MR 3729789

[3] J. Adler and O. Öktem, Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging* **37** (2018), no. 6, 1322–1332

[4] G. Alessandrini and J. Sylvester, Stability for a multidimensional inverse spectral theorem. *Comm. Partial Differential Equations* **15** (1990), no. 5, 711–736 Zbl 0715.35080 MR 1070844

[5] M. Anderson, A. Katsuda, Y. Kurylev, M. Lassas, and M. Taylor, Boundary regularity for the Ricci equation, geometric convergence, and Gel'fand's inverse boundary problem. *Invent. Math.* **158** (2004), no. 2, 261–321 Zbl 1177.35245 MR 2096795

[6] S. Antholzer, M. Haltmeier, and J. Schwab, Deep learning for photoacoustic tomography from sparse data. *Inverse Probl. Sci. Eng.* **27** (2019), no. 7, 987–1005 Zbl 1465.94008 MR 3938045

[7] S. Arridge and A. Hauptmann, Networks for nonlinear diffusion problems in imaging. *J. Math. Imaging Vision* **62** (2020), no. 3, 471–487 Zbl 1434.68496 MR 4082373

[8] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, Solving inverse problems using data-driven models. *Acta Numer.* **28** (2019), 1–174 Zbl 1429.65116 MR 3963505

[9] G. Bal and L. Ryzhik, Time reversal and refocusing in random media. *SIAM J. Appl. Math.* **63** (2003), no. 5, 1475–1498 Zbl 1126.76360 MR 2001204

[10] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, Learning-based compressive subsampling. *IEEE Journal of Selected Topics in Signal Processing* **10** (2016), no. 4, 809–822

[11] R. Balestriero and R. G. Baraniuk, Mad Max: Affine spline insights Into deep learning. *Proceedings of the IEEE* **109** (2021), no. 5, 704–727

[12] P. Bartlett, D. J. Foster, and M. Telgarsky, Spectrally-normalized margin bounds for neural networks. 2017, arXiv:1706.08498

[13] M. I. Belishev, An approach to multidimensional inverse problems for the wave equation. *Dokl. Akad. Nauk SSSR* **297** (1987), no. 3, 524–527 Zbl 0661.35084 MR 924687

[14] M. I. Belishev and Y. V. Kurylev, To the reconstruction of a Riemannian manifold via its spectral data (BC-method). *Comm. Partial Differential Equations* **17** (1992), no. 5-6, 767–804 Zbl 0812.58094 MR 1177292

[15] M. Bellassoued and D. Dos Santos Ferreira, Stability estimates for the anisotropic wave equation from the Dirichlet-to-Neumann map. *Inverse Probl. Imaging* **5** (2011), no. 4, 745–773 Zbl 1250.58012 MR 2852371

[16] G. Bellec, D. Kappel, W. Maass, and R. Legenstein, Deep rewiring: Training very sparse deep networks. In *ICLR 2018 – 6th International Conference on Learning Representations*, 2018, https://openreview.net/forum?id=BJ_wN01C-

[17] K. Bingham, Y. Kurylev, M. Lassas, and S. Siltanen, Iterative time-reversal control for inverse problems. *Inverse Probl. Imaging* **2** (2008), no. 1, 63–81 Zbl 1141.35468 MR 2375323

[18] C. M. Bishop, *Pattern recognition and machine learning*. Inf. Sci. Stat., Springer, New York, 2006 Zbl 1107.68072 MR 2247587

[19] L. Borcea, G. Papanicolaou, and C. Tsogka, Theory and applications of time reversal and interferometric imaging. Special section on imaging. *Inverse Problems* **19** (2003), no. 6, S139–S164 Zbl 1045.94500 MR 2036525

[20] J. Bruna and S. Mallat, Multiscale sparse microcanonical models. *Math. Stat. Learn.* **1** (2018), no. 3-4, 257–315 Zbl 1426.62111 MR 4059723

[21] 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 Problems* **35** (2019), no. 6, Art. ID 064002 Zbl 1416.92099 MR 3975365

[22] D. Burago and S. Ivanov, Boundary rigidity and filling volume minimality of metrics close to a flat one. *Ann. of Math. (2)* **171** (2010), no. 2, 1183–1211 Zbl 1192.53048 MR 2630062

[23] P. Caday, M. V. de Hoop, V. Katsnelson, and G. Uhlmann, Reconstruction of piecewise smooth wave speeds using multiple scattering. *Trans. Amer. Math. Soc.* **372** (2019), no. 2, 1213–1235 Zbl 1426.35237 MR 3968801

[24] P. Caday, M. V. de Hoop, V. Katsnelson, and G. Uhlmann, Scattering control for the wave equation with unknown wave speed. *Arch. Ration. Mech. Anal.* **231** (2019), no. 1, 409–464 Zbl 1412.35377 MR 3894555

[25] P. L. Combettes and J.-C. Pesquet, Lipschitz certificates for layered network structures driven by averaged activation operators. *SIAM J. Math. Data Sci.* **2** (2020), no. 2, 529–557 MR 4117304

[26] G. Dardikman-Yoffe and Y. C. C. Eldar, Learned SPARCOM: Unfolded deep super-resolution microscopy. *Opt. Express* **28** (2020), no. 19, 27736–27763

[27] I. Daubechies, M. Defrise, and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.* **57** (2004), no. 11, 1413–1457 Zbl 1077.65055 MR 2077704

[28] M. V. de Hoop, P. Kepley, and L. Oksanen, Recovery of a smooth metric via wave field and coordinate transformation reconstruction. *SIAM J. Appl. Math.* **78** (2018), no. 4, 1931–1953 Zbl 1395.35224 MR 3825620

[29] R. M. Dudley, *Real analysis and probability*. Cambridge Stud. Adv. Math. 74, Cambridge Univ. Press, Cambridge, 2002 Zbl 1023.60001 MR 1932358

[30] J. Frankle and M. Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR 2019 – International Conference on Learning Representations*, 2019, https://openreview.net/forum?id=rJl-b3RcF7

[31] J. Friedman, T. Hastie, and R. Tibshirani, Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33** (2010), no. 1, 1–22

[32] D. Gilton, G. Ongie, and R. Willett, Neumann networks for linear inverse problems in imaging. *IEEE Trans. Comput. Imaging* **6** (2020), 328–343 MR 4063269

[33] M. Giordano and R. Nickl, Consistency of Bayesian inference with Gaussian process priors in an elliptic inverse problem. *Inverse Problems* **36** (2020), no. 8, Art. ID 085001 Zbl 1445.35330 MR 4151406

[34] X. Glorot, A. Bordes, and Y. Bengio, Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 315–323, Proceedings of Machine Learning Research 15, PMLR, 2011

[35] G. H. Golub and C. F. Van Loan, *Matrix computations*. Fourth edn., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins Univ. Press, Baltimore, MD, 2013 Zbl 1268.65037 MR 3024913

[36] K. Gregor and Y. LeCun, Learning fast approximations of sparse coding. In *ICML '10 – Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406, Omnipress, 2010

[37] S. Han, J. Pool, J. Tran, and W. J. Dally, Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems 28*, Curran Associates, 2015

[38] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, 2016

[39] W. Hoeffding, Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* **58** (1963), 13–30 Zbl 0127.10602 MR 144363

[40] K. Hornik, Approximation capabilities of multilayer feedforward networks. *Neural Networks* **4** (1991), no. 2, 251–257

[41] M. Jaderberg, A. Vedaldi, and A. Zisserman, Speeding up convolutional neural networks with low rank expansions. In *Proceedings of the British Machine Vision Conference*, BMVA Press, 2014

[42] 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** (2017), no. 9, 4509–4522 Zbl 1409.94275 MR 3670561

[43] P. Jin, L. Lu, and Y. Tang, Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. *Neural Networks* **130** (2020), 85–99

[44] O. Kallenberg, *Foundations of modern probability*. Second edn., Probab. Appl. (N. Y.), Springer, New York, 2002 Zbl 0996.60001 MR 1876169

[45] A. Katchalov, Y. Kurylev, and M. Lassas, *Inverse boundary spectral problems*. Chapman Hall/CRC Monogr. Surv. Pure Appl. Math. 123, Chapman & Hall/CRC, Boca Raton, FL, 2001 Zbl 1037.35098 MR 1889089

[46] A. Katchalov, Y. Kurylev, M. Lassas, and N. Mandache, Equivalence of time-domain inverse problems and boundary spectral problems. *Inverse Problems* **20** (2004), no. 2, 419–436 Zbl 1073.35209 MR 2065431

[47] Y. Khoo and L. Ying, SwitchNet: A neural network model for forward and inverse scattering problems. *SIAM J. Sci. Comput.* **41** (2019), no. 5, A3182–A3201 Zbl 1425.65208 MR 4018415

[48] J. Korpela, M. Lassas, and L. Oksanen, Regularization strategy for an inverse problem for a $1 + 1$ dimensional wave equation. *Inverse Problems* **32** (2016), no. 6, Art. ID 065001 Zbl 1382.35335 MR 3493581

[49] J. Korpela, M. Lassas, and L. Oksanen, Discrete regularization and convergence of the inverse problem for $1 + 1$ dimensional wave equation. *Inverse Probl. Imaging* **13** (2019), no. 3, 575–596 Zbl 1418.35380 MR 3959328

[50] J. Kukačka, V. Golkov, and D. Cremers, Regularization for deep learning: A taxonomy. 2017, arXiv:1710.10686

[51] Y. Kurylev, M. Lassas, and G. Uhlmann, Inverse problems for Lorentzian manifolds and non-linear hyperbolic equations. *Invent. Math.* **212** (2018), no. 3, 781–857 Zbl 1396.35074 MR 3802298

[52] M. Lassas, Inverse problems for linear and non-linear hyperbolic equations. In *Proceedings of the International Congress of Mathematicians—Rio de Janeiro 2018. Vol. IV. Invited lectures*, pp. 3751–3771, World Scientific Publ., Hackensack, NJ, 2018 Zbl 1447.35006 MR 3966550

[53] M. Lassas and L. Oksanen, Inverse problem for the Riemannian wave equation with Dirichlet data and Neumann data on disjoint sets. *Duke Math. J.* **163** (2014), no. 6, 1071–1103 Zbl 1375.35634 MR 3192525

[54] M. Lassas, V. Sharafutdinov, and G. Uhlmann, Semiglobal boundary rigidity for Riemannian metrics. *Math. Ann.* **325** (2003), no. 4, 767–793 Zbl 1331.53066 MR 1974568

[55] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, Speeding-up convolutional neural networks using fine-tuned CP-decomposition. 2014, arXiv:1412.6553

[56] Y. LeCun, J. S. Denker, and S. A. Solla, Optimal brain damage. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, pp. 598–605, MIT Press, Cambridge, MA, USA, 1989

[57] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier, NETT: Solving inverse problems with deep neural networks. *Inverse Problems* **36** (2020), no. 6, Art. ID 065005 Zbl 1456.65038 MR 4115067

[58] L. Li, L. G. Wang, F. L. Teixeira, C. Liu, A. Nehorai, and T. J. Cui, DeepNIS: Deep neural network for nonlinear electromagnetic inverse scattering. *IEEE Transactions on Antennas and Propagation* **67** (2019), no. 3, 1819–1825

[59] X. Li, J. Lu, Z. Wang, J. Haupt, and T. Zhao, On tighter generalization bound for deep neural networks: CNNs, ResNets, and beyond. 2018, arXiv:1806.05159

[60] X. Li and Z. Zhou, Speech command recognition with convolutional neural network. 2017, http://cs229.stanford.edu/proj2017/final-reports/5244201.pdf

[61] S. Liu and L. Oksanen, A Lipschitz stable reconstruction formula for the inverse problem for the wave equation. *Trans. Amer. Math. Soc.* **368** (2016), no. 1, 319–335 Zbl 1329.35355 MR 3413865

[62] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, Rethinking the value of network pruning. In *ICLR 2019 – International Conference on Learning Representations*, 2019, https://openreview.net/forum?id=rJlnB3C5Ym

[63] B. Luijten, R. Cohen, F. J. de Bruijn, H. A. W. Schmeitz, M. Mischi, Y. C. Eldar, and R. J. G. van Sloun, Adaptive ultrasound beamforming using deep learning. *IEEE Transactions on Medical Imaging* **39** (2019), no. 12, 3967–3978

[64] S. Lunz, O. Öktem, and C.-B. Schönlieb, Adversarial regularizers in inverse problems. In *Advances in Neural Information Processing Systems 31*, pp. 8507–8516, Curran Associates, 2018

[65] A. L. Maas, A. Y. Hannun, and A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research 28, PMLR, 2013

[66] D. Mocanu, E. Mocanu, P. Stone, P. Nguyen, M. Gibescu, and A. Liotta, Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications* **9** (2018), Art. ID 2383

[67] F. Monard, R. Nickl, and G. P. Paternain, Statistical guarantees for Bayesian uncertainty quantification in nonlinear inverse problems with Gaussian process priors. *Ann. Statist.* **49** (2021), no. 6, 3255–3298 MR 4352530

[68] V. Monga, Y. Li, and Y. Eldar, Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine* **38** (2021), 18–44

[69] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, On the number of linear regions of deep neural networks. In *NIPS '14 – Proceedings of the 27th International Conference on Neural Information Processing Systems. Volume 2*, pp. 2924–2932, MIT Press, Cambridge, MA, USA, 2014

[70] H. Mostafa and X. Wang, Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4646–4655, Proceedings of Machine Learning Research 97, PMLR, 2019

[71] B. Neyshabur, S. Bhojanapalli, and N. Srebro, A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018, https://openreview.net/forum?id=Skz_WfbCZ

[72] B. Neyshabur, R. Tomioka, and N. Srebro, Norm-based capacity control in neural networks. In *Proceedings of the 28th Conference on Learning Theory*, pp. 1376–1401, Proceedings of Machine Learning Research 40, PMLR, 2015

[73] R. Nickl, On Bayesian inference for some statistical inverse problems with partial differential equations. *Bernoulli News* **24** (2017), no. 2, 5–9

[74] A. M. Oberman and J. Calder, Lipschitz regularized deep neural networks converge and generalize. 2018, arXiv:1808.09540

[75] L. Oksanen, Solving an inverse obstacle problem for the wave equation by using the boundary control method. *Inverse Problems* **29** (2013), no. 3, Art. ID 035004 Zbl 1302.65217 MR 3029503

[76] L. Pestov and G. Uhlmann, Two dimensional compact simple Riemannian manifolds are boundary distance rigid. *Ann. of Math. (2)* **161** (2005), no. 2, 1093–1110 Zbl 1076.53044 MR 2153407

[77] A. Pinkus, Approximation theory of the MLP model in neural networks. In *Acta numerica, 1999*, pp. 143–195, Acta Numer. 8, Cambridge Univ. Press, Cambridge, 1999 Zbl 0959.68109 MR 1819645

[78] P. Putzky and M. Welling, Recurrent inference machines for solving inverse problems. 2017, arXiv:1706.04008

[79] O. Ronneberger, P. Fischer, and T. Brox, U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Springer, Cham, 2015

[80] W. Rudin, *Real and complex analysis*. Third edn., McGraw-Hill Book Co., New York, 1987 Zbl 0925.00005 MR 924157

[81] W. Rudin, *Functional analysis*. Second edn., International Series in Pure and Applied Mathematics, McGraw-Hill, Inc., New York, 1991 Zbl 0867.46001 MR 1157815

[82] T. Sanchez, I. Krawczuk, Z. Sun, and V. Cevher, Closed loop deep Bayesian inversion: Uncertainty driven acquisition for fast MRI. In *ICLR 2020 – International Conference on Learning Representations*, 2020, https://openreview.net/forum?id=BJlPOlBKDB

[83] P. Stefanov and G. Uhlmann, Stability estimates for the hyperbolic Dirichlet to Neumann map in anisotropic media. *J. Funct. Anal.* **154** (1998), no. 2, 330–358 Zbl 0915.35066 MR 1612709

[84] P. Stefanov and G. Uhlmann, Boundary and lens rigidity, tensor tomography and analytic microlocal analysis. In *Algebraic analysis of differential equations. From microlocal analysis to exponential asymptotics*, pp. 275–293, Springer, Tokyo, 2008 Zbl 1138.53039 MR 2758914

[85] P. Stefanov, G. Uhlmann, and A. Vasy, Boundary rigidity with partial data. *J. Amer. Math. Soc.* **29** (2016), no. 2, 299–332 Zbl 1335.53055 MR 3454376

[86] J. Sylvester and G. Uhlmann, A global uniqueness theorem for an inverse boundary value problem. *Ann. of Math. (2)* **125** (1987), no. 1, 153–169 Zbl 0625.35078 MR 873380

[87] N. Tishby and N. Zaslavsky, Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2015

[88] H. Triebel, *Function spaces and wavelets on domains*. EMS Tracts Math. 7, Eur. Math. Soc., Zürich, 2008 Zbl 1158.46002 MR 2455724

[89] G. Uhlmann, Inverse boundary value problems for partial differential equations. Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998). *Doc. Math.* (1998), Extra Vol. III, 77–86 Zbl 0906.35111 MR 1648142

[90] M. Unser, A representer theorem for deep neural networks. *J. Mach. Learn. Res.* **20** (2019), Paper No. 110 Zbl 1434.68526 MR 3990464

[91] M. Unser, J. Fageot, and J. P. Ward, Splines are universal solutions of linear inverse problems with generalized TV regularization. *SIAM Rev.* **59** (2017), no. 4, 769–793 Zbl 1382.41011 MR 3720356

[92] Z. Wei and X. Chen, Deep-learning schemes for full-wave nonlinear inverse scattering problems. *IEEE Transactions on Geoscience and Remote Sensing* **57** (2019), 1849–1860

[93] A. Yaguchi, T. Suzuki, S. Nitta, Y. Sakata, and A. Tanizawa, Scalable deep neural networks via low-rank matrix factorization. 2019, arXiv:1910.13141v1

[94] D. Yarotsky, Error bounds for approximations with deep ReLU networks. 2016, arXiv:1610.01145

Received 23 January 2020; revised 7 November 2021.

Maarten V. de Hoop
Department of Computational and Applied Mathematics and Department of Earth Sciences, Rice University, 6100 Main Street, Houston, TX 77005, USA; mdehoop@rice.edu

Matti Lassas
Department of Mathematics and Statistics, University of Helsinki, P.O. Box 68 (Gustaf Hällströmin katu 2b), 00014 Helsinki, Finland; matti.lassas@helsinki.fi

Christopher A. Wong
Department of Computational and Applied Mathematics and Department of Earth Sciences, Rice University, 6100 Main Street, Houston, TX 77005, USA; cawong89@gmail.com