OPTIMAL APPROXIMATION WITH SPARSELY CONNECTED DEEP NEURAL NETWORKS

HELMUT BÖLCSKEI, PHILIPP GROHS, GITTA KUTYNIOK, AND PHILIPP PETERSEN

ABSTRACT. We derive fundamental lower bounds on the connectivity and the memory requirements of deep neural networks guaranteeing uniform approximation rates for arbitrary function classes in $L^2(\mathbb{R}^d)$. In other words, we establish a connection between the complexity of a function class and the complexity of deep neural networks approximating functions from this class to within a prescribed accuracy.

Additionally, we prove that our lower bounds are achievable for a broad family of function classes. Specifically, all function classes that are optimally approximated by a general class of representation systems—so-called *affine systems*—can be approximated by deep neural networks with minimal connectivity and memory requirements. Affine systems encompass a wealth of representation systems from applied harmonic analysis such as wavelets, ridgelets, curvelets, shearlets, α-shearlets, and more generally α-molecules. This result elucidates a remarkable universality property of neural networks and shows that they achieve the optimum approximation properties of all affine systems combined. As a specific example, we consider the class of $1/\alpha$-cartoon-like functions, which is approximated optimally by α-shearlets.

We also explain how our results can be extended to the case of functions on low-dimensional immersed manifolds.

Finally, we present numerical experiments demonstrating that the standard stochastic gradient descent algorithm generates deep neural networks providing close-to-optimal approximation rates at minimal connectivity. Moreover, these results show that stochastic gradient descent actually learns approximations that are sparse in the representation systems optimally sparsifying the function class the network is trained on.

1. INTRODUCTION

Neural networks arose from the seminal work by McCulloch and Pitts [34] in 1943 which, inspired by the functionality of the human brain, introduced an algorithmic approach to learning with the aim of building a theory of artificial intelligence. Roughly speaking, a neural network consists of neurons arranged in layers and connected by weighted edges; in mathematical terms this boils down to a concatenation of (potentially learned) affine linear functions and relatively simple non-linearities.

Despite significant theoretical progress in the 1990s [28, 7], the area has seen practical advances only during the past decade, triggered by the drastic improvements in computing power, and, in particular, the availability of vast amounts of training data. Specifically, deep neural networks, i.e., networks with large numbers of layers are now state-of-the-art technology for a wide variety of real-world applications, such as image classification [30], speech recognition [27], or game intelligence [8], to name a few. For an in-depth overview, we refer to the survey paper by LeCun, Bengio, and Hinton [33] and the recent book [19].

A neural network effectively implements a non-linear mapping and can be used to either perform classification directly or to extract features that are then fed into a classifier, such as a support vector machine [44]. In the former case, the primary goal is to approximate an unknown classification function based on a given set of corresponding input-output value pairs. This is typically accomplished by learning the network's weights through, e.g., the standard gradient descent (via backpropagation) algorithm [41]. In a classification task with, say, two classes, the function to be learned would take only two values, whereas in the case of, e.g., the prediction of the temperature in a certain environment, it would be real-valued. It is therefore clear that characterizing to what extent deep neural networks are capable of approximating general functions is a question of significant practical relevance.

Deep neural networks employed in practice often consist of hundreds of layers and may depend on billions of parameters, see for example the work [26] on image classification. Training and operation of networks of this scale entail formidable computational challenges which often still present a bottleneck. As a case in point, we mention speech recognition on a smart phone such as, e.g., Apple's SIRI-system, which operates in the cloud. Android's speech recognition system has meanwhile released an offline version that is based on a neural network with sparse connectivity, meaning that the number of edges with nonzero weights is small.

2010 *Mathematics Subject Classification.* 41A25, 82C32, 42C40, 42C15, 41A46, 68T05, 94A34, 94A12.

Key words and phrases. Deep neural networks, function approximation, optimal sparse approximation, sparse connectivity, shearlets.

The desire to reduce the complexity of network training and operation naturally leads to the question of function approximation through neural networks with sparse connectivity. In addition, the network's memory requirements in terms of the number of bits needed to store its topology and quantized weights are of concern in practice.

The purpose of this paper is to characterize the approximation-theoretic properties of deep neural networks under connectivity and memory constraints. Specifically, defining the complexity of a signal class $\mathcal{C}$ as the number of bits needed to describe any element in $\mathcal{C}$ to within a prescribed accuracy, we shall ask the following question:

> *Given a signal class $\mathcal{C}$, how does the complexity of a neural network that approximates every function in $\mathcal{C}$ to within a prescribed accuracy depend on the complexity of $\mathcal{C}$?*

Interpreting the network as an encoder in Donoho's min-max rate distortion theory [14], we establish fundamental lower bounds on connectivity and memory requirements for a network to guarantee uniform approximation rates for a given signal class $\mathcal{C}$. Moreover, we demonstrate that these bounds are saturated by a broad family of signal classes, namely those that can be optimally approximated by a general class of representation systems—so-called affine systems. Affine systems encompass a wealth of representation systems from applied harmonic analysis such as wavelets, ridgelets, curvelets, shearlets, α-shearlets, and more generally α-molecules. Our result therefore reveals an interesting universality property of neural networks; they achieve the optimum approximation properties of all affine systems combined. The technique we develop to derive this result is interesting in its own right as it constitutes a more general framework for transferring results on function approximation through representation systems to results on approximation by neural networks.

1.1. **Deep Neural Networks.**

While various network architectures exist in the literature, we focus on the following setup.

Definition 1.1. *Let $L, d, N_1, \ldots, N_L \in \mathbb{N}$. A map $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ given by*

$$\Phi(x) = W_L \rho\left(W_{L-1} \rho\left(\ldots \rho\left(W_1(x)\right)\right)\right), \quad x \in \mathbb{R}^d, \tag{1}$$

is called a neural network. *It is composed of affine linear maps $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$, $1 \le \ell \le L$, and non-linear functions—often referred to as* rectifiers*—ρ acting component-wise. Here, d is the* dimension of the input layer*, L denotes the* number of layers*, $N_1, \ldots, N_{L-1}$ stands for the* dimensions of the $L-1$ hidden layers*, and N_L is the* dimension of the output layer.

The term "network" arises from the interpretation of the mapping Φ as a weighted acyclic directed graph with nodes arranged in L hierarchical layers and edges only between adjacent layers. In fact, the affine linear map W_ℓ is defined by a matrix $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and an affine part $b_\ell \in \mathbb{R}^{N_\ell}$ via $W_\ell(x) = A_\ell x + b_\ell$. $(A_\ell)_{i,j}$ is the *weight* associated with the edge between the j-th node in the $(\ell-1)$-th layer and the i-th node in the ℓ-th layer, while $(b_\ell)_i$ is the weight associated with the i-th neuron in the ℓ-th layer. This assignment is depicted in Figure 1. We refer to the nodes of the graph as neurons and note that the total number of neurons is given by $N := d + \sum_{j=1}^{L} N_j$.

The real numbers $(A_\ell)_{i,j}$ and $(b_\ell)_i$ are said to be the network's weights and the total number of non-zero edge weights, denoted by M, is the network's connectivity. If M is small relative to the number of connections possible (i.e., the number of edges in the graph that is fully connected between adjacent layers), we say that the network is *sparsely connected*.

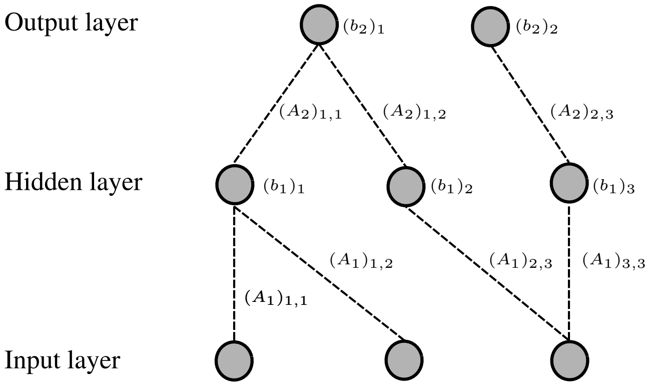
Throughout the paper, we consider the case $\Phi : \mathbb{R}^d \to \mathbb{R}$, i.e., $N_L = 1$, which includes situations such as the classification problem described above. Nonetheless, the results of Sections 2 and 3 as well as Theorem 4.12 can be readily generalized to $N_L > 1$.

We denote the class of networks $\Phi : \mathbb{R}^d \to \mathbb{R}$ with no more than L layers, no more than M non-zero edge weights, and rectifier ρ by $\mathcal{NN}_{L,M,d,\rho}$. Moreover, we let

$$\mathcal{NN}_{\infty,M,d,\rho} := \bigcup_{L \in \mathbb{N}} \mathcal{NN}_{L,M,d,\rho}, \quad \mathcal{NN}_{L,\infty,d,\rho} := \bigcup_{M \in \mathbb{N}} \mathcal{NN}_{L,M,d,\rho}, \quad \mathcal{NN}_{\infty,\infty,d,\rho} := \bigcup_{L \in \mathbb{N}} \mathcal{NN}_{L,\infty,d,\rho}.$$

Now, given a function $f : \mathbb{R}^d \to \mathbb{R}$, we can ask how well a neural network $\Phi \in \mathcal{NN}_{L,M,d,\rho}$ can approximate f. Clearly, this depends on the algorithm chosen to learn the network's weights. But one can also take the following vantage point: The best possible approximation of f by $\mathcal{NN}_{L,M,d,\rho}$ provides a fundamental lower bound on the approximation error *independently of the learning algorithm*.

We shall be interested in the dependence of this lower bound on the connectivity M and on the number of bits available to encode the network topology and the quantized weights. Clearly, smaller M entails lower computational complexity in terms of evaluating (1) and a smaller number of bits translates to smaller memory requirements.



$$A_2 = \begin{pmatrix} (A_2)_{1,1} & (A_2)_{1,2} & 0 \\ 0 & 0 & (A_2)_{2,3} \end{pmatrix}$$

$$A_1 = \begin{pmatrix} (A_1)_{1,1} & (A_1)_{1,2} & 0 \\ 0 & 0 & (A_1)_{2,3} \\ 0 & 0 & (A_1)_{3,3} \end{pmatrix}$$

FIGURE 1. Assignment of the weights $(A_\ell)_{i,j}$ and $(b_\ell)_i$ to the neurons and edges. The network has sparse connectivity, i.e., most of the weights $(A_\ell)_{i,j}$ equal zero.

1.2. **Quantifying Approximation Quality.** We next briefly review a widely used approach for characterizing the approximation quality of functions given restricting conditions on the approximant, in our case the neural network with the restrictions imposed by its connectivity and memory requirements.

Fix $\Omega \subset \mathbb{R}^d$. A common approach to function approximation is to consider a class of functions $\mathcal{C} \subseteq L^2(\Omega)$, termed *signal class* and a corresponding complete system $\mathcal{D} := (\varphi_i)_{i\in I} \subseteq L^2(\Omega)$, termed *representation system*, or *dictionary*, with the restriction on the approximant imposed by a limit on the number of elements in $\mathcal{D}$ allowed to participate in the approximation. One then studies the *error of best M-term approximation* of $f \in \mathcal{C}$:

Definition 1.2. [13] *Given $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, a signal class $\mathcal{C} \subseteq L^2(\Omega)$, and a representation system $\mathcal{D}$, we define, for $f \in \mathcal{C}$ and $M \in \mathbb{N}$,*

$$\Gamma_M^{\mathcal{D}}(f) := \inf_{I_M \subset I, \#I_M = M, (c_i)_{i\in I_M}} \|f - \sum_{i \in I_M} c_i\varphi_i\|_{L^2(\Omega)}. \tag{2}$$

We call $\Gamma_M^{\mathcal{D}}(f)$ the best M-term approximation error of f with respect to $\mathcal{D}$. *Every $f_M = \sum_{i\in I_M} c_i\varphi_i$ attaining the infimum is referred to as a* best M-term approximation *of f in the system $\mathcal{D}$. The supremum of $\gamma > 0$ such that there exists $C > 0$ with*

$$\sup_{f\in\mathcal{C}} \Gamma_M^{\mathcal{D}}(f) \leq CM^{-\gamma}, \qquad \textit{for all } M \in \mathbb{N},$$

determines the optimal M-term approximation rate of $\mathcal{C}$ in the system $\mathcal{D}$, *and will henceforth be referred to as $\gamma^*(\mathcal{C}, \mathcal{D})$.*

A wealth of structured representation systems $\mathcal{D}$ is provided by the area of applied harmonic analysis, starting with wavelets [9], followed by ridgelets [3], curvelets [4], shearlets [25], parabolic molecules [24], and most generally α-molecules [22], which include all previously named systems as special cases. Other examples include tensor product wavelets [10], Gabor frames [20], and wave atoms [11].

For α-shearlet dictionaries $\mathcal{D}$, optimal sparse approximation properties have been completely characterized for the signal class $\mathcal{C}$ of so-called β-cartoon-like functions $\mathcal{E}^\beta([0,1]^2)$, these are piecewise $C^\beta(\mathbb{R}^2)$ functions on the unit square with a C^β discontinuity curve and $\beta \in [1,2]$. Under weak technical conditions on the α-shearlet system, it was shown in [23, 39] that $\gamma^*(\mathcal{C}, \mathcal{D}) = \frac{1}{2\alpha}$ for $\mathcal{C} = \mathcal{E}^{\frac{1}{\alpha}}([0,1]^2)$.

1.3. **Approximation by Deep Neural Networks.** We now substitute the concept of M-term approximation with representation systems by approximation through neural networks with M edges. In other words, sparsity in terms of the number of participating elements of a representation system is replaced by sparsity in terms of connectivity.

More formally, we consider the following setup.

Definition 1.3. *Given $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, a signal class $\mathcal{C} \subseteq L^2(\Omega)$, and $\rho : \mathbb{R} \to \mathbb{R}$, we define, for $f \in \mathcal{C}$ and $M \in \mathbb{N}$,*

$$\Gamma_M^{\mathcal{NN}}(f) := \inf_{\Phi \in \mathcal{NN}_{\infty,M,d,\rho}} \|f - \Phi(f)\|_{L^2(\Omega)}. \tag{3}$$

We call $\Gamma_M^{\mathcal{NN}}(f)$ *the* best M-edge approximation error of f. *The supremal* $\gamma > 0$ *such that a* $C > 0$ *with*

$$\sup_{f\in\mathcal{C}} \Gamma_M(f) \leq CM^{-\gamma}, \qquad \textit{for all } M \in \mathbb{N},$$

exists, determines the optimal M-edge approximation rate of $\mathcal{C}$ achievable by neural networks $\mathcal{NN}_{\infty,\infty,d,\rho}$, *and will henceforth be referred to as* $\gamma^*_{\mathcal{NN}}(\mathcal{C})$.

We emphasize that the infimum in (3) is taken over all networks with no more than M edges of non-zero weight. In particular, this means that the optimum is taken over all possible edge positions and associated edge weights. Knowledge of the optimal M-edge approximation rate hence provides a bound on the approximation rate of a sparsely connected deep neural network. This bound is fundamental as it must be met by all learning algorithms. While we do not evaluate specific learning algorithms, our framework provides a means for assessing the quality of a given learning algorithm in the sense of measuring how close the rate induced by the algorithm comes to the optimal M-edge approximation rate.

1.4. **Previous Work.** The best-known results on approximation by neural networks are the universal approximation theorems by Hornik [28] and Cybenko [7], stating that every measurable function f can be approximated arbitrarily well by a single-hidden-layer ($L = 2$ in our terminology) neural network. The literature on approximation-theoretic properties of shallow networks continuing this work is abundant. Without any claim of comprehensiveness, we mention work on approximation error bounds for functions with bounded first moments [1], bounds in terms of the number of nodes and in terms of the size of the training set [2], the non-existence of localized approximations [5], a fundamental lower bound on approximation rates [12, 3], lower bounds on the network's sparsity—in terms of the number of neurons [42], and on the approximation of specific function classes [36, 35].

Approximation-theoretic results for networks with multiple hidden layers were obtained in [29] for general functions, in [18] for special function classes such as continuous functions, and for functions together with their derivatives in [38].

In [5] it was shown that, although localized approximation is impossible for shallow networks, it can be accomplished using deeper networks. We also highlight two very recent papers, which investigate the benefit—from an approximation-theoretic perspective—of multiple hidden layers. Specifically, in [16] it was shown that there exists a function which although expressible through a small 3-layer network, can only be accurately represented through a very large 2-layer network; here size is measured in terms of the number of nodes. In the setting of deep convolutional neural networks, first results of a nature similar to those in [16] were derived in [37]. For survey articles on approximation-theoretic aspects of neural networks, we refer the interested reader to [17, 40].

Relative to the above-mentioned contributions, the aim of this paper is to derive a universal optimality result concerning approximation properties of deep neural networks in terms of connectivity and memory requirements.

Most closely related to our work is that by Shaham, Cloninger, and Coifman [43], which shows that for functions that are sparse in wavelet frames the best M-edge approximation rate of a 3-layer neural network is at least as good as the best M-term approximation rate by piecewise linear wavelet frames.

1.5. **Contributions.** Our contributions can be grouped into four threads.

- *Fundamental lower bound on connectivity.* Let $d \in \mathbb{N}$ be the dimension of the input layer, $\Omega \subset \mathbb{R}^d$, $\rho : \mathbb{R} \to \mathbb{R}$ an arbitrary rectifier, and $\mathcal{C}$ an arbitrary subset of $L^2(\Omega)$. Consider a learning algorithm $\mathbf{Learn}$ which maps each accuracy $\epsilon \in (0, 1/2)$ and function $f \in \mathcal{C}$ to a neural network $\mathbf{Learn}(\epsilon, f) \in \mathcal{NN}_{\infty,\infty,d,\rho}$ such that

(4)
$$\sup_{f\in\mathcal{C}} \|f - \mathbf{Learn}(\epsilon, f)\|_{L^2(\Omega)} \leq \epsilon,$$

where the weights of $\mathbf{Learn}(\epsilon, f)$ can each be encoded (i.e., represented) with $O(\log_2(1/\epsilon))$ bits. We demonstrate in Theorem 2.8 that for all $\gamma < \gamma^*(\mathcal{C})$, where $\gamma^*(\mathcal{C})$ is the optimal exponent with respect to the minimax code length of $\mathcal{C}$ as defined in [14, 21] (see Definition 2.1 below), all $C > 0$, and every sequence $(\epsilon_n)_{n\in\mathbb{N}}$ converging to 0 such that $\epsilon_n/\epsilon_{n+1}$ is bounded, there exists a subsequence $(\epsilon_{n'})_{n'\in\mathbb{N}}$ such that

(5)
$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon_{n'}, f)) \geq C\epsilon_{n'}^{-\gamma}, \qquad \text{for all } n' \in \mathbb{N},$$

where $\mathcal{M}(\mathbf{Learn}(\epsilon, f))$ denotes the number of edges of the network $\mathbf{Learn}(\epsilon, f)$. This result quantifies the minimum network connectivity needed to allow approximation of *all* elements in $\mathcal{C}$ to within a prescribed error. In particular, it says that the minimum (complexity) exponent at which the number of edges in the network has to scale is given by $\gamma^*(\mathcal{C})$. On a conceptual level, this statement establishes a universal link between the connectivity of the approximating network and the complexity of the function class that it is to approximate.

- *Optimal M-edge approximation based on representation systems.* We develop an abstract framework for transferring optimal M-term approximation results with representation systems to optimal M-edge approximation results for neural networks. These transfer results hold for dictionaries that are *representable by neural networks* in the following sense: There exists $R \in \mathbb{N}$ such that for every dictionary element and every $\epsilon > 0$, there is a neural network with no more than R edges approximating this element with error at most ϵ. Additionally, we show that a wide class of representation systems, coined *affine systems*, and including as special cases wavelets, ridgelets, curvelets, shearlets, α-shearlets, and more generally, α-molecules, as well as tensor-products thereof, are representable by neural networks. This result suggests an explanation for the "unreasonable effectiveness" of neural networks: They effectively combine the optimum approximation properties of all affine systems taken together.

  For concreteness and to illustrate our general results, we consider the class $\mathcal{C}$ of $1/\alpha$-cartoon-like functions $\mathcal{E}^{\frac{1}{\alpha}}([0,1]^2) \subset L^2([0,1]^2)$ with $\alpha \in [\frac{1}{2}, 1]$ and $\gamma^*(\mathcal{C}) = \frac{2}{\alpha}$, and, for each $\epsilon > 0$ and $f \in \mathcal{C}$, we construct a neural network $\mathbf{Learn}(\epsilon, f)$ satisfying (4). This construction is based on α-shearlets, which are known to provide optimal M-term approximation rate for $\mathcal{C}$. For two classes of rectifiers, namely sigmoidal functions of arbitrary order (Definition 4.1) and smooth approximations of rectified linear units (Definition 4.2), we show that for all $\gamma > \gamma^*(\mathcal{C})$ there exists $C > 0$ such that

  $$\sup_{f \in \mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) \le C\epsilon^{-\gamma}, \qquad \text{for all } \epsilon \in \left(0, \frac{1}{2}\right),$$

  i.e., the number of edges in the constructed approximating neural network grows at most at rate $O(\epsilon^{-\gamma})$ as $\epsilon \to 0$.

  Finally, in Section 6 we show that our techniques and results can be extended to the approximation of functions on immersed submanifolds of $\mathbb{R}^d$.

- *Quantization and memory requirements.* The lower bound (5) holds under the assumption that the weights in the network can each be encoded with $O(\log_2(1/\epsilon))$ bits. That this is, indeed, possible is certainly not clear a priori. To clarify this matter we proceed as follows. We first demonstrate that, if a signal class $\mathcal{C}$ is *effectively representable* (Definition 2.2) with respect to a dictionary $\mathcal{D}$ and the dictionary is *effectively representable by neural networks* (Definition 3.1), a certain best M-term approximation rate in $\mathcal{D}$ implies the same M-edge approximation rate by a neural network with the weights of the network bounded by $p(M)$, where p is a polynomial. Then, we demonstrate that the so-bounded weights can be quantized to $O(\log_2(1/\epsilon))$ bits without incurring a considerable overall approximation error. This implies achievability of the bound (5) and demonstrates that the network at the same time has minimal memory requirements.
- *Realizability of optimal approximation rates.* An important practical question is how networks trained by gradient descent (via backpropagation) [41] perform relative to the fundamental bounds established in the paper. Surprisingly, our numerical experiments indicate that, given a fixed network topology with sparse connectivity, the stochastic gradient descent algorithm yields close-to-optimal approximation rates. Moreover, for the approximation of functions $f \in \mathcal{E}^{\frac{1}{\alpha}}([0,1]^2)$ we observe that stochastic gradient descent generates neural networks which mimic the classical M-term approximation of f in a dictionary of α-molecules.

1.6. **Outline of the Paper.** The fundamental lower bound on connectivity will be developed in Section 2. Section 3 introduces a general framework for transferring sparse approximation results in representation systems to approximation results by neural networks. In Section 4, we apply these techniques first to the broad class of affine systems and then, in Section 5, to the class of $1/\alpha$-cartoon-like functions. Section 6 describes the extension of our main findings to the approximation of functions defined on manifolds. Numerical results assessing the performance of stochastic gradient descent (via backpropagation) relative to our lower bound on connectivity are reported in Section 7.

2. A Fundamental Lower Bound on Connectivity

We start by establishing a lower bound on the worst-case connectivity of neural networks approximating elements from a given signal class $\mathcal{C} \subseteq L^2(\mathbb{R}^d)$ to within a prescribed accuracy of $\epsilon > 0$.

Let us first recall concepts and definitions from min-max rate distortion theory as introduced by Donoho in [14]. Specifically, we review fundamental bounds on the length of lossy compression by encoder-decoder pairs. We then recall in Subsection 2.2 how dictionaries can be interpreted as encoders, and in Subsection 2.3 we analyze neural networks from an encoder point of view.

2.1. **Min-Max Rate Distortion Theory.** Min-max rate distortion theory provides a theoretical foundation for deterministic lossy data compression. We recall the following notions and concepts from [14, 21].

Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$ and consider the function class $\mathcal{C} \subset L^2(\Omega)$. Then, for each $\ell \in \mathbb{N}$, we denote by

$$\mathfrak{E}^\ell := \{E : \mathcal{C} \to \{0,1\}^\ell\}$$

the set of *binary encoders mapping elements of* $\mathcal{C}$ *to bit strings of length* ℓ, and we let

$$\mathfrak{D}^\ell := \{D : \{0,1\}^\ell \to L^2(\Omega)\}$$

be the set of *binary decoders mapping bit strings of length* ℓ *to elements of* $L^2(\Omega)$. An encoder-decoder pair $(E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell$ is said to *achieve distortion* $\epsilon > 0$ *over the function class* $\mathcal{C}$, if

$$\sup_{f \in \mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)} \leq \epsilon.$$

This means that the worst case error incurred by applying the encoder-decoder pair $(E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell$ to an element of $\mathcal{C}$ is upper-bounded by ϵ, often also expressed as the uniform error over $\mathcal{C}$ being bounded by ϵ.

A quantity of central interest is the minimal length $\ell \in \mathbb{N}$ for which there exists an encoder-decoder pair $(E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell$ that achieves distortion $\epsilon > 0$ over the function class $\mathcal{C}$, and its asymptotic behavior as made precise in the following definition.

Definition 2.1. *Let* $d \in \mathbb{N}$ *and* $\mathcal{C} \subset L^2(\Omega)$. *Then, for* $\epsilon > 0$ *the* minimax code length $L(\epsilon, \mathcal{C})$ *is*

$$L(\epsilon, \mathcal{C}) := \min\{\ell \in \mathbb{N} : \exists (E, D) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell : \sup_{f \in \mathcal{C}} \|D(E(f)) - f\|_{L^2(\Omega)} \leq \epsilon\}.$$

Moreover, the optimal exponent $\gamma^*(\mathcal{C})$ *is defined by*

$$\gamma^*(\mathcal{C}) := \inf\{\gamma \in \mathbb{R} : L(\epsilon, \mathcal{C}) = O(\epsilon^{-\gamma})\}.$$

The optimal exponent $\gamma^*(\mathcal{C})$ describes how fast $L(\epsilon, \mathcal{C})$ tends to infinity as ϵ decreases. For function classes $\mathcal{C}_1$ and $\mathcal{C}_2$, $\gamma^*(\mathcal{C}_1) < \gamma^*(\mathcal{C}_2)$ says that asymptotically, i.e., for $\epsilon \to 0$, the length of the encoding bit string for $\mathcal{C}_2$ is larger than that for $\mathcal{C}_1$. In other words, a smaller exponent indicates smaller description complexity. The optimal exponent $\gamma^*(\mathcal{C})$ therefore determines the minimal memory requirements for storing signals $f \in \mathcal{C}$ such that reconstruction with a uniformly bounded error is possible.

We mention that sometimes in the literature the reciprocal of $\gamma^*(\mathcal{C})$ is termed the optimal exponent. The optimal exponent is known for several function classes, such as subsets of Besov spaces $B^s_{p,q}(\mathbb{R}^d)$ with $1 \leq p, q < \infty$, $s > 0$, and $q > (s + 1/2)^{-1}$, namely all functions in $B^s_{p,q}(\mathbb{R}^d)$ whose norm is bounded by a constant $C > 0$ [6]. For this class we have $\gamma^*(\mathcal{C}) = \frac{d}{s}$. We will be particularly interested in so-called β-cartoon-like functions, for which the optimal exponent is given by $\frac{2}{\beta}$, see Theorem 5.3.

2.2. **Representation Systems as Encoders.** For later use we recall well-known relations between the optimal M-term approximation rate $\gamma^*(\mathcal{C}, \mathcal{D})$ for a signal class $\mathcal{C}$ with respect to a dictionary $\mathcal{D} = (\varphi_i)_{i \in I}$ and the optimal exponent $\gamma^*(\mathcal{C})$ per Definition 2.1. In particular, we treat the question of how to build efficient encoders based on best M-term approximations in a given dictionary.

This is formalized by the following definition which is a variation of the concept of "best M-term approximation subject to polynomial-depth search" introduced in [14].

Definition 2.2. *Let* $\Omega \subset \mathbb{R}^d$. *Consider the signal class* $\mathcal{C} \subset L^2(\Omega)$ *and the representation system* $\mathcal{D} = (\varphi_i)_{i \in \mathbb{N}} \subset L^2(\Omega)$. $\mathcal{C}$ *is said to be* effectively representable in $\mathcal{D}$, *if there exist a univariate polynomial* π *and a constant* D *such that for all* $\gamma < \gamma^*(\mathcal{C}, \mathcal{D})$ *there is a constant* $C > 0$ *so that for all* $M \in \mathbb{N}$ *and* $f \in \mathcal{C}$: *there exist* $I_M \subset \{1, \ldots, \pi(M)\}$ *with* $\#I_M = M$ *and* $(c_i)_{i \in I_M}$ *with* $\max_{i \in I_M} |c_i| \leq D$ *such that*

$$\|f - \sum_{i \in I_M} c_i \varphi_i\|_{L^2(\Omega)} \leq C M^{-\gamma}. \tag{6}$$

Remark 2.3. *We note that the notion of effective representability depends on the ordering of the representation system $\mathcal{D}$.*

The condition of having the index set I_M of a best M-term approximation be contained in the set of the first $\pi(M)$ indices implies the following fundamental relationship between the best M-term approximation rate in representation systems and the optimal exponent $\gamma^*(\mathcal{C})$.

Theorem 2.4 ([14, 21])**.** *Let $\Omega \subset \mathbb{R}^d$, $\mathcal{C} \subset L^2(\Omega)$ and assume that $\mathcal{C}$ is effectively representable in $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subset L^2(\Omega)$. Then, we have*

$$\gamma^*(\mathcal{C}) \leq \frac{1}{\gamma^*(\mathcal{C}, \mathcal{D})}.$$

In light of this result, the following definition is natural (see also [21]).

Definition 2.5. *Let $\Omega \subset \mathbb{R}^d$ and assume that the signal class $\mathcal{C} \subset L^2(\Omega)$ is effectively representable in the system $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subset L^2(\Omega)$ with*

$$\gamma^*(\mathcal{C}) = \frac{1}{\gamma^*(\mathcal{C}, \mathcal{D})}.$$

Then, the representation system $\mathcal{D}$ is said to be optimal for the signal class $\mathcal{C}$.

2.3. **Neural Networks as Encoders.** The present section examines to which extent efficient encoders can be constructed from neural networks with limited connectivity. Interpreting neural networks as encoders in the sense of Donoho's min-max rate distortion theory leads us to the first main result of this paper, namely a fundamental lower bound on the connectivity of a neural network approximating the elements of a given function class $\mathcal{C}$ to within accuracy $\epsilon > 0$. We say that this lower bound is fundamental as it has to be met by all learning algorithms.

Remark 2.6. *Throughout the paper, we are concerned with network connectivity in terms of the number of edges of non-zero weight. Looking at the number of nodes instead would not be conceptually different as nodes that are not part of any edge do not influence the underlying mapping Φ and we can bound the number of nodes by twice the number of edges.*

The main result of this section is as follows.

Theorem 2.7. *Let $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, $\rho : \mathbb{R} \to \mathbb{R}$, $c > 0$, and $\mathcal{C} \subset L^2(\Omega)$. Further, let*

$$\mathbf{Learn} : \left(0, \frac{1}{2}\right) \times \mathcal{C} \to \mathcal{NN}_{\infty,\infty,d,\rho}$$

be a map such that, for each pair $(\epsilon, f) \in (0, \frac{1}{2}) \times \mathcal{C}$, every weight of the neural network $\mathbf{Learn}(\epsilon, f)$ can be encoded with no more than $-c\log_2(\epsilon)$ bits while guaranteeing that

$$\sup_{f\in\mathcal{C}} \|f - \mathbf{Learn}(\epsilon, f)\|_{L^2(\Omega)} \leq \epsilon. \tag{7}$$

Then,

$$\sup_{\epsilon\in(0,\frac{1}{2})} \epsilon^{\gamma} \cdot \sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) = \infty, \quad \textit{for all } \gamma < \gamma^*(\mathcal{C}). \tag{8}$$

Proof. Let $\gamma < \gamma^*(\mathcal{C})$ and assume to the contrary that

$$\sup_{\epsilon\in(0,\frac{1}{2})} \epsilon^{\gamma} \cdot \sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) < \infty. \tag{9}$$

In the course of the proof we show that the networks $\mathbf{Learn}(\epsilon, f)$ lead to encoder-decoder pairs $(E, D) \in \mathfrak{E}^{\ell(\epsilon)} \times \mathfrak{D}^{\ell(\epsilon)}$ achieving distortion ϵ over $\mathcal{C}$ with

$$\ell(\epsilon) \leq C_0 \cdot \sup_{f\in\mathcal{C}} \left(\mathcal{M}(\mathbf{Learn}(\epsilon, f)) \log_2(\mathcal{M}(\mathbf{Learn}(\epsilon, f))) \log(1/\epsilon),\right.$$

where C_0 is a constant.

A contradiction to (9) is then produced through the following argument: Take $\delta > 0$ with $\gamma < \gamma + \delta < \gamma^*(\mathcal{C})$. By assumption there exists $C' > 0$ such that

$$C_0 \sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) \log_2(\mathcal{M}(\mathbf{Learn}(\epsilon, f))) \log(1/\epsilon) \leq C'\epsilon^{-(\gamma+\delta)}, \quad \text{for all } \epsilon \in \left(0, \frac{1}{2}\right).$$

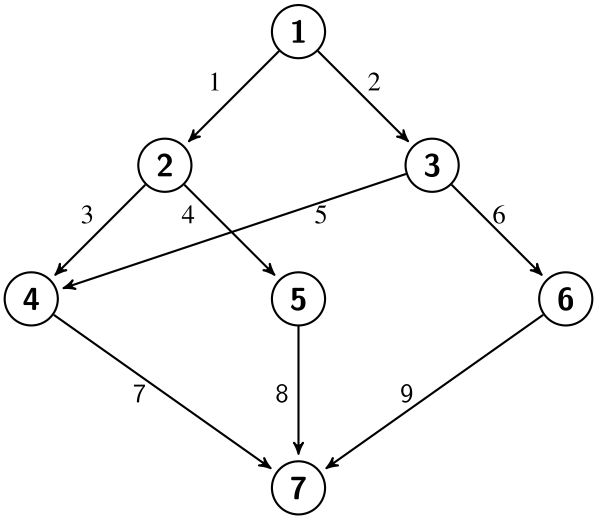


FIGURE 2. Example of a natural numbering of neurons and edges.

This would, however, imply that $L(\epsilon, \mathcal{C}) \leq \ell(\epsilon) = O(\epsilon^{-(\gamma+\delta)})$ which constitutes a contradiction to the optimality of $\gamma^*(\mathcal{C})$ and consequently yields (8).

The remainder of the proof is now devoted to establishing that the networks $\mathbf{Learn}(\epsilon, f)$ lead to encoder-decoder pairs $(E, D) \in \mathfrak{E}^{\ell(\epsilon)} \times \mathfrak{D}^{\ell(\epsilon)}$ with the aforementioned properties. We start by analyzing the code length of a neural network-based encoder operating on $\mathcal{C}$. More precisely, for given $f \in \mathcal{C}$ and $\epsilon > 0$, we aim at encoding the neural network $\mathbf{Learn}(\epsilon, f)$, more specifically its topology and weights, by a bit string of length $O(\mathcal{M}(\mathbf{Learn}(\epsilon, f)))$. To this end, we will frequently invoke the observation that the number of nodes of $\mathbf{Learn}(\epsilon, f)$ is upper-bounded by $2\mathcal{M}(\mathbf{Learn}(\epsilon, f))$, see Remark 2.6. Also, trivially, the number of layers is upper-bounded by $\mathcal{M}(\mathbf{Learn}(\epsilon, f))$.

We fix $f \in \mathcal{C}$ and enumerate the neurons and edges of $\mathbf{Learn}(\epsilon, f)$ by assigning unique numbers, henceforth called *indices* of the nodes and edges. This numbering can, for instance, simply assign increasing indices from left to right in every layer as indicated in Figure 2. For the sake of notational simplicity, we also set

$$\Phi := \mathbf{Learn}(\epsilon, f),$$

and

$$M := \mathcal{M}(\Phi).$$

Without loss of generality we assume that M is a power of 2. We recall that the number of layers of Φ is denoted by L, the number of neurons in these layers is $N_1, \ldots, N_L$ (see Definition 1.1), and d is the dimension of the input layer. By Remark 2.6 we have

$$d + \sum_{\ell=1}^{L} N_\ell \leq 2M, \tag{10}$$

which bounds the total number of neurons, and

$$L \leq M, \tag{11}$$

which bounds the total number of layers.

We then construct a bit string representing Φ according to the following steps.

Step 1: We encode the number of edges, M, by starting the overall bit string with M 1's followed by a single 0.

Step 2: We continue by encoding the number of layers in the network. Thanks to (11) this requires no more than $\log_2(M)$ bits. We thus reserve the next $\log_2(M)$ bits for the binary representation of L. [*]

Step 3: For each layer ℓ, we store the number N_ℓ of neurons. Moreover, we store the dimension d of the input layer. As by (10) $N_\ell \leq 2M$, for all ℓ, we can encode (generously) each N_ℓ as well as d using $\log_2(M) + 1$ bits.

[*] Here and in the remainder of the proof, we follow the convention that bit strings are zero-padded from the left, that is, for example, if the number of layers L is equal to 2 and $M = 8$, we encode L by the bit string 010.

In total, Step 3 requires a bit string of length

$$((L+1)\cdot(\log_2(M)+1)) \leq M + M\cdot\log_2(M) + \log_2(M) + 1.$$

In combination with Steps 1 and 2 this yields an overall bit string of length at most

$$2M + M\cdot\log_2(M) + 2\log_2(M) + 2. \tag{12}$$

Step 4: We encode the topology of the graph associated with Φ. Recall that we assigned a unique index i to each neuron. These indices range from 1 to $N := d + \sum_{\ell=1}^{L} N_\ell$ and, by (10), can each be encoded by a bit string of length $\log_2(M)+1$. We denote the bit string corresponding to index i by $b(i) \in \{0,1\}^{\log_2(M)+1}$ and let $n(i)$ be the number of children of neuron i. For each neuron $i = 1,\dots,N$, we form a bit string of length $(n(i)+2)\cdot(\log_2(M)+1)$ by concatenating the representations $b(j)$ of the indices of all its children, followed by an all-zeros bit string of length $2(\log_2(M)+1)$ to signal the transition to the neuron with index $i+1$. Overall, this yields a bit string of length

$$\sum_{i=1}^{N}(n(i)+2)\cdot(\log_2(M)+1) \leq 6M\cdot(\log_2(M)+1), \tag{13}$$

where again we used (10).

Combining (12) and (13) it follows that we have encoded the full topology of the neural network Φ using at most

$$8M + 7M\cdot\log_2(M) + 2\log_2(M) + 2 \tag{14}$$

bits.

Step 5: We encode the weights of Φ. By assumption, each weight can be encoded by a bit string of length $-\lfloor c'\log_2(\epsilon)\rfloor$. For each node $i = 1,\dots,N$, we reserve exactly $-\lfloor c'\log_2(\epsilon)\rfloor$ bits to encode its weight and, for each of its children a bit string of length $-\lfloor c'\log_2(\epsilon)\rfloor$ to encode the weight corresponding to the edge between that child and its parent node. Concatenating the results in a bit string of length $(n(i)+1)\cdot(-\lfloor c'\log_2(\epsilon)\rfloor)$ for each node i, we get in total a bit string of length at most

$$-3M\cdot\lfloor c'\log_2(\epsilon)\rfloor \tag{15}$$

representing the weights of the graph associated with the network Φ.

Summing up (14) for the number of bits needed to encode the topology of Φ and (15) for the number of bits needed to encode the corresponding weights, we conclude that a bit string of length at most

$$8M + 7M\cdot\log_2(M) + 2\log_2(M) + 2 + 1 - 3M\cdot\lfloor c'\log_2(\epsilon)\rfloor, \tag{16}$$

is needed to encode Φ. The network can be recovered uniquely from this bit string by simply reversing the steps above. As (16) can be upper-bounded by

$$C_0\cdot M\cdot\log_2(M)\cdot\log_2(1/\epsilon) \tag{17}$$

for a constant $C_0 > 0$ depending on c' only, we have constructed an encoder-decoder pair $(E,D) \in \mathfrak{E}^{\ell(\epsilon)} \times \mathfrak{D}^{\ell(\epsilon)}$ with $\ell(\epsilon) \leq C_0\cdot M\cdot\log_2(M)\cdot\log_2(1/\epsilon)$ and distortion uniformly bounded (for the class $\mathcal{C}$) by ϵ. This concludes the proof. □

We also offer a quantitative version of the above theorem which yields an explicit lower bound on the connectivity induced by every learning procedure exhibiting a given uniform approximation rate for a given signal class.

Theorem 2.8. *Let $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, $\rho : \mathbb{R} \to \mathbb{R}$, $c > 0$, and $\mathcal{C} \subset L^2(\Omega)$. Further, let*

$$\mathbf{Learn} : \left(0, \frac{1}{2}\right) \times \mathcal{C} \to \mathcal{NN}_{\infty,\infty,d,\rho}$$

be a map such that, for each pair $(\epsilon, f) \in (0,\frac{1}{2}) \times \mathcal{C}$, every weight of the neural network $\mathbf{Learn}(\epsilon, f)$ can be represented with no more than $-c\log_2(\epsilon)$ bits while guaranteeing (7).

Let $\gamma < \gamma^*(\mathcal{C})$ and $C > 0$ and consider a zero-sequence $(\epsilon_n)_{n\in\mathbb{N}} \subset (0,\frac{1}{2})$ with $(\epsilon_n/\epsilon_{n+1})_{n\in\mathbb{N}} \in \ell^\infty$. Then, there exists a subsequence $(\epsilon_{n'})_{n'\in\mathbb{N}}$ such that

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon_{n'}, f)) \geq C\epsilon_{n'}^{-\gamma}, \qquad \textit{for all } n' \in \mathbb{N}. \tag{18}$$

Proof. Let $\gamma < \gamma^*(\mathcal{C})$ and take $(\epsilon_n)_{n\in\mathbb{N}}$ to be a zero-sequence such that $(\epsilon_n/\epsilon_{n+1})_{n\in\mathbb{N}} \in \ell^\infty$. We will prove that there does not exist $C' > 0$ so that

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon_n, f)) \le C'\epsilon_n^{-\gamma}, \qquad \text{for all } n \in \mathbb{N}. \tag{19}$$

It is not hard to see that the non-existence of C' such that (19) holds yields (18). Assume towards a contradiction, that there exists $C' > 0$ such that (19) holds. We can now construct a second learning algorithm

$$\mathbf{Learn}' : \left(0, \frac{1}{2}\right) \times L^2(\Omega) \to \mathcal{NN}_{\infty,\infty,d,\rho}, \quad \mathbf{Learn}'(\epsilon, f) := \mathbf{Learn}(\epsilon_{n(\epsilon)}, f), \tag{20}$$

where $\epsilon_{n(\epsilon)}$ is the largest ϵ_n with $\epsilon_n \le \epsilon < \epsilon_{n-1}$. We get

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}'(\epsilon, f)) = \sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon_{n(\epsilon)}, f))$$

$$\le C'\epsilon_{n(\epsilon)}^{-\gamma} = C'(\epsilon_{n(\epsilon)}/\epsilon)^{-\gamma}\epsilon^{-\gamma} \le C'\|\epsilon_n/\epsilon_{n-1}\|_\infty^{-\gamma}\epsilon^{-\gamma} \le C''\epsilon^{-\gamma}, \tag{21}$$

for a suitable constant $C'' > 0$ and all $\epsilon \in (0, \frac{1}{2})$. Additionally, we have that

$$\sup_{f\in\mathcal{C}} \|f - \mathbf{Learn}'(\epsilon, f)\|_{L^2(\Omega)} \le \epsilon_{n(\epsilon)} \le \epsilon,$$

for every $\epsilon \in (0, \frac{1}{2})$, and every weight of $\mathbf{Learn}'(\epsilon, f)$ can be represented with $-c\log_2(\epsilon_{n(\epsilon)}) \le -c'\log_2(\epsilon)$ bits for some $c' > 0$. Applying Theorem 2.7 to the mapping $\mathbf{Learn}'$ yields the desired contradiction. □

We note that the assumption on the encoding complexity of the weights of $\mathbf{Learn}(\epsilon, f)$ in Theorem 2.8 is necessary in order to be able to store the weights of the network $\mathbf{Learn}(\epsilon, f)$ on a computer. We conclude that *Theorem 2.8 exhibits a fundamental lower bound on the storage complexity of any neural network which uniformly approximates $\mathcal{C}$ to within a given accuracy!*

In Theorem 2.8 we established that a mapping $\mathbf{Learn}$ as in Theorem 2.8 such that for all $\gamma > \gamma^*(\mathcal{C})$ there exists $C > 0$ and

$$\sup_{f\in\mathcal{C}} \mathcal{M}(\mathbf{Learn}(\epsilon, f)) \le C\epsilon^{-\gamma}, \qquad \text{for all } \epsilon \in \left(0, \frac{1}{2}\right), \tag{22}$$

yields neural networks $\mathbf{Learn}(\epsilon, f))$ whose number of edges scales asymptotically optimally in terms of the approximation accuracy ϵ. We cast this property into a definition.

Definition 2.9. *We call a signal class $\mathcal{C} \subset L^2(\Omega)$* optimally representable by neural networks with rectifier function *ρ, if a mapping* $\mathbf{Learn}$ *as in Theorem 2.8 exists that additionally satisfies: For all $\gamma > \gamma^*(\mathcal{C})$ there exists $C > 0$ such that* (22) *holds.*

We would like to emphasize that optimal representability as just defined is a much stronger statement than simply providing a bound for $\gamma^*_{\mathcal{NN}}(\mathcal{C})$ as per Definition 1.3. If a signal class $\mathcal{C}$ is optimally representable, Theorem 2.8 states that the memory requirements for storing a neural network that approximates any target function $f \in \mathcal{C}$ to within a prescribed accuracy are minimal. We hasten to add that these minimal memory requirements are realized by representing each of the weights in the network by the same number of bits—of order $\log(1/\epsilon)$, a standing assumption throughout the paper.

3. Transitioning from Representation Systems to Neural Networks

Given the benchmark result Theorem 2.8 we now ask the following question:

Which signal classes are optimally representable by (deep) neural networks?

The remainder of the paper is devoted to demonstrating that a large family of signal classes is optimally representable by (deep) neural networks. This will be accomplished by establishing an upper bound on connectivity that matches the lower bound in Theorem 2.8.

The mathematical technique we develop in the process is interesting in its own right as it constitutes a general framework for transferring results on function approximation through representation systems to results for approximation by neural networks. In particular, we prove that for a given signal class and associated representation system, under certain technical conditions, there exists a neural network with no more than $O(M)$ non-zero weights that achieves the same approximation error as a best M-term approximation in that representation system. This is formalized in

Theorem 3.4. By imposing slightly more stringent technical conditions this result can be sharpened to additionally guarantee that the network's weights can be represented with no more than $-c\log_2(\epsilon)$ bits. The proof is constructive in the sense of making the quantizer used to encode the weights explicit. Specifically, we quantize by rounding to $\epsilon^m\mathbb{Z}\cap[-\epsilon^{-k},\epsilon^{-k}]$, for some $m,k\in\mathbb{N}$. These weights can clearly be represented with no more than $-c\log_2(\epsilon)$ bits, with a constant $c>0$ depending upon m and k only.

We start by stating the technical conditions on the representation system needed for the aforementioned transference results to hold. For $X\subset\mathbb{R}^d$ we say that $A:X\to\mathbb{R}$ is *polynomially bounded in* x, if there exists a d-dimensional polynomial p such that $|A(x)|\leq|p(x)|$, for all $x\in X$. We call a collection of functions A_i, $i\in I$, *uniformly polynomially bounded*, if all A_i are polynomially bounded with the same bounding polynomial.

Definition 3.1. *Let $\Omega\subset\mathbb{R}^d$, $\rho:\mathbb{R}\to\mathbb{R}$, and $\mathcal{D}=(\varphi_i)_{i\in\mathbb{N}}\subset L^2(\Omega)$ be a representation system. Then, $\mathcal{D}$ is said to be* representable by neural networks (with rectifier ρ)*, if there exists $R\in\mathbb{N}$ such that for all $\eta>0$ and every $i\in I$ there is a neural network $\Phi_{i,\eta}\in\mathcal{NN}_{\infty,R,d,\rho}$ with*

$$\|\varphi_i-\Phi_{i,\eta}\|_{L^2(\Omega)}\leq\eta.$$

If, in addition, the neural networks $\Phi_{i,\eta}\in\mathcal{NN}_{\infty,C,d,\rho}$ have weights that are uniformly polynomially bounded in (i,η^{-1}), and if ρ is either globally Lipschitz continuous, or differentiable with polynomially bounded first derivative, we call the representation system $(\varphi_i)_{i\in\mathbb{N}}$ effectively representable by neural networks (with rectifier ρ).

Remark 3.2. *We remark that, just like effective representability in a representation system (Definition 2.2), effective representability by neural networks depends on the ordering of the elements in the representation system.*

The following lemma demonstrates that for a network with weights and number of non-zero weights both polynomially bounded in η^{-1}, it is always possible to replace the network by one whose weights can be represented with no more than $-c\log_2(\eta)$ bits per weight without incurring an overall approximation error larger than η.

Lemma 3.3. *Let $d,L\in\mathbb{N}$, $\Omega\subset\mathbb{R}^d$ be bounded and let $\rho:\mathbb{R}\to\mathbb{R}$ be either Lipschitz-continuous or differentiable with polynomially bounded first derivative. Additionally, let $1/2>\eta>0$, $k\in\mathbb{N}$, and consider $\Phi\in\mathcal{NN}_{L,M,d,\rho}$ such that $M\leq\eta^{-k}$ with all weights of Φ bounded by η^{-k}.*

Then, there exist $m\in\mathbb{N}$, depending on k,L,ρ only, and $\widetilde{\Phi}\in\mathcal{NN}_{L,M,d,\rho}$ such that

$$\|\widetilde{\Phi}-\Phi\|_{L^\infty(\Omega)}\leq\eta$$

and all weights of $\widetilde{\Phi}$ are elements of $\eta^m\mathbb{Z}\cap[-\eta^{-k},\eta^{-k}]$.

Proof. We present the proof for Lipschitz-continuous ρ only. The other case follows similarly but is slightly more technical. Let $m\in\mathbb{N}$, to be specified later, and, for a given network Φ, let $\widetilde{\Phi}$ denote the network that results by replacing all weights of Φ by a closest element in $\eta^m\mathbb{Z}\cap[-\eta^{-k},\eta^{-k}]$. Set $C_{max}:=\eta^{-k}$ and denote the total number of non-zero weights in the network by C_W. Note that $C_W\leq 3M\leq 3\eta^{-k}$, where the latter inequality is by assumption. For $\ell=1,\dots,L$, define $\Phi^\ell:\Omega\to\mathbb{R}^{N_\ell}$ as

$$\Phi^\ell(x)=W_\ell\rho\left(W_{\ell-1}\rho\left(\dots\rho\left(W_1(x)\right)\right)\right),\quad x\in\Omega,$$

and let

$$e_\ell:=\|\Phi^\ell-\widetilde{\Phi}^\ell\|_{L^\infty(\Omega,\mathbb{R}^{N_\ell})}.$$

Set $C_0=\max\{1,\sup\{|x|:x\in\Omega\}\}$, denote the Lipschitz constant of ρ by C_ρ, and let

$$C_\ell=\max\{\|\Phi^\ell\|_{L^\infty(\Omega)},\|\tilde{\Phi}^\ell\|_{L^\infty(\Omega)},1\}.$$

Then, it is not difficult to see that

$$e_1\leq\ C_\rho\,C_W\,\eta^m,\ \text{and}\ e_\ell\leq\ e_{\ell-1}\,C_{\ell-1}\,C_{max}\,C_W+C_\rho\,C_W\,\eta^m,\ \text{for all}\ \ell=2,\dots,L. \tag{23}$$

We now bound the quantity C_ℓ for $\ell=1,\dots,L$. A simple computation, exploiting the Lipschitz continuity of ρ, yields that

$$C_\ell\leq(\rho(0)+C_\rho C_W C_{max}C_{\ell-1}),\qquad\text{for}\ \ell=1,\dots,L.$$

Since $\rho(0)<\infty$ there exists $C'>0$ such that

$$C_\ell\leq C'C_0(C_\rho C_W C_{max})^\ell,\quad\text{for}\ \ell=1,\dots,L.$$

As C_W and C_{max} are both polynomially bounded in η^{-1}, it follows that C_ℓ is polynomially bounded in η^{-1}. We can therefore find $n \in \mathbb{N}$ such that $C_{\ell-1} C_\rho C_W C_{max} \leq \eta^{-n}$. Invoking (23), we conclude that there exists a constant $C'' > 0$ so that

$$e_\ell \leq C'' \eta^{m-\ell n}, \quad \text{for } \ell = 1, \ldots, L.$$

Finally, choosing m large enough yields $e_L \leq \eta$, which was to be established. □

The next theorem relates M-term approximation properties of representation systems to M-edge approximation properties for neural networks.

Theorem 3.4. *Let $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, and $\rho : \mathbb{R} \to \mathbb{R}$. Suppose that $\mathcal{D} = (\varphi_i)_{i \in \mathbb{N}} \subset L^2(\Omega)$ is representable by neural networks in the sense of Definition 3.1. Let $f \in L^2(\Omega)$ and, for $M \in \mathbb{N}$, let $f_M = \sum_{i \in I_M} c_i \varphi_i$, $I_M \subset \mathbb{N}$, $\# I_M = M$, satisfy*

$$\| f - f_M \|_{L^2(\Omega)} \leq \epsilon,$$

where $\epsilon \in (0, 1/2)$. Then, there exists a corresponding neural network $\Phi(f, M) \in \mathcal{NN}_{\infty, M', d, \rho}$ with $M' = O(M)$, satisfying

$$\| f - \Phi(f, M) \|_{L^2(\Omega)} \leq 2\epsilon.$$

In particular, for all signal classes $\mathcal{C} \subset L^2(\Omega)$ it holds that

$$\gamma^*_{\mathcal{NN}}(\mathcal{C}) \geq \gamma^*(\mathcal{C}, \mathcal{D}).$$

Proof. By the triangle inequality it suffices to show that there exists a neural network that approximates f_M to within ϵ. By representability according to Definition 3.1, it follows that there is $R \in \mathbb{N}$ such that for each $i \in I_M$ and for $\eta := \epsilon / M$, there exists a neural network $\Phi_{i,\eta} \in \mathcal{NN}_{\infty, R, d, \rho}$ with

$$\| \varphi_i - \Phi_{i,\eta} \|_{L^2(\Omega)} \leq \eta. \tag{24}$$

Let then $\Phi(f, M)$ be a neural network consisting of the networks $(\Phi_{i,\eta})_{i \in I_M}$ operating in parallel, all with the same input, and summing their one-dimensional outputs with weights $(c_i)_{i \in I_M}$ according to

$$\Phi(f, M)(x) := \sum_{i \in I_M} c_i \Phi_{i,\eta}(x). \tag{25}$$

Then, $\Phi(f, M) \in \mathcal{NN}_{R, M', d, \rho}$, where $M' = RM$ and, using (24), we can conclude that

$$\| f_M - \Phi(f, M) \|_{L^2(\Omega)} \leq \epsilon,$$

which completes the proof. □

This result shows that we can restrict ourselves to the approximation of the individual elements of a representation system by neural networks with the only constraint being that the number of edges in the individual networks must admit a uniform upper bound.

The previous result does, however, not guarantee that the weights of the network $\Phi(f, M)$ can be represented with $-c \log(\epsilon)$ bits for a constant $c > 0$. To understand if the lower bound on the connectivity of Theorem 2.8 is sharp we thus need to extend Theorem 3.4, which will, again, be accomplished through a transfer argument.

Theorem 3.5. *Let $\Omega \subset \mathbb{R}^d$ be bounded and suppose that $\mathcal{C} \subset L^2(\Omega)$ is effectively representable in the dictionary $\mathcal{D} = (\varphi_i)_{i \in \mathbb{N}} \subset L^2(\Omega)$ according to Definition 2.2. Suppose that $\mathcal{D}$ is effectively representable by neural networks according to Definition 3.1. Then, for all $\gamma < \gamma^*(\mathcal{C}, \mathcal{D})$ there exist constants $c, C > 0$ and a map*

$$\mathbf{Learn} : \left(0, \frac{1}{2}\right) \times L^2(\Omega) \to \mathcal{NN}_{\infty, \infty, d, \rho},$$

such that for every $f \in \mathcal{C}$ the following statements hold:

- *(i) there exist $m, k \in \mathbb{N}$ such that each weight of the neural network $\mathbf{Learn}(\epsilon, f)$ is an element of $\epsilon^m \mathbb{Z} \cap [-\epsilon^{-k}, \epsilon^{-k}]$,*
- *(ii) the estimate*

  $$\| f - \mathbf{Learn}(\epsilon, f) \|_{L^2(\Omega)} \leq \epsilon$$

  holds true, and
- *(iii) the neural network $\mathbf{Learn}(\epsilon, f)$ has at most $C \cdot \epsilon^{-\gamma}$ edges with non-zero weights.*

Specifically, if $\mathcal{D}$ is effectively representable by neural networks, then optimality of $\mathcal{D}$ for the signal class $\mathcal{C}$ in the sense of Definition 2.5 implies that $\mathcal{C}$ is optimally representable by neural networks in the sense of Definition 2.9.

In Section 4 we provide a large class of dictionaries that are effectively representable by neural networks and yield effective representations for relevant function classes.

Proof of Theorem 3.5. Let $f \in \mathcal{C}$, $M \in \mathbb{N}$, and $\gamma < \gamma^*(\mathcal{C}, \mathcal{D})$. Pick a subset $I_M \subset \{1, \ldots, \pi(M)\}$ and coefficients $(c_i)_{i \in I_M}$ according to Definition 2.2 such that (6) holds and, in the notation of (6), set $\epsilon_M/2 := CM^{-\gamma}$. Let $\Phi(f, M)$ be constructed as in the proof of Theorem 3.4. Note that $\Phi(f, M) \in \mathcal{NN}_{R,RM,d,\rho}$ for some $R \in \mathbb{N}$, independent of ϵ. Next, note that by effective representability of $\mathcal{C}$ in $\mathcal{D}$, the weights $(c_i)_{i \in I_M}$ in the last layer of $\Phi(f, M)$ are given by $(c_i)_{i \in I_M}$ and, moreover, these weights are uniformly bounded, independently of $f \in \mathcal{C}$. By the assumption of $\mathcal{D}$ being effectively representable by neural networks, the weights in the approximating networks $\Phi_{i,\eta}(x)$ in (25) are uniformly polynomially bounded in $(i, (M/\epsilon_M))$, and since $i \leq \pi(M)$, the weights in the approximating networks $\Phi_{i,\eta}(x)$ in (25) are uniformly polynomially bounded in (M, ϵ_M^{-1}). Since $M = (\epsilon_M/(2C))^{-\gamma}$ it follows that the weights in the approximating networks $\Phi_{i,\eta}(x)$ in (25) are uniformly polynomially bounded in ϵ_M^{-1}. Invoking Lemma 3.3 yields the existence of $m, k \in \mathbb{N}$ such that there is $\tilde{\Phi}(f, M)$ satisfying

$$\|\Phi(f, M) - \tilde{\Phi}(f, M)\|_{L^2(\Omega)} \leq \epsilon_M/2, \quad \text{for all } f \in \mathcal{C},$$

and where all weights in $\tilde{\Phi}(f, M)$ are elements of $\epsilon^m \mathbb{Z} \cap [-\epsilon^{-k}, \epsilon^{-k}]$. Moreover, by construction

$$\|f - \tilde{\Phi}(f, M)\|_{L^2(\Omega)} \leq \epsilon_M.$$

We set $\mathbf{Learn}(\epsilon_M, f) := \tilde{\Phi}(f, M)$. For general $\epsilon \in (0, 1/2)$, we let

$$\mathbf{Learn}(\epsilon, f) := \mathbf{Learn}(\epsilon_{M(\epsilon)}, f),$$

where $\epsilon_{M(\epsilon)}$ is chosen such that $\epsilon_{M(\epsilon)} \leq \epsilon < \epsilon_{M(\epsilon)-1}$. Clearly,

$$\|f - \mathbf{Learn}(\epsilon, f)\|_{L^2(\Omega)} \leq \epsilon,$$

and $\mathbf{Learn}(\epsilon, f)$ has $RM(\epsilon) \leq R((\epsilon/(2C))^{-\gamma} + 1)$ edges which yields the desired result. □

4. All Affine Representation Systems are Effectively Representable by Neural Networks

This section shows that a large class of dictionaries, namely *affine systems*, as defined below, are effectively representable by neural networks. Affine systems include as special cases wavelets, ridgelets, curvelets, shearlets, α-shearlets, and more generally α-molecules. Combined with Theorem 3.5 the results in this section establish the central conceptual fact that any signal class that is effectively representable in an arbitrary affine system is optimally representable by neural networks in the sense of Definition 2.9.

Clearly, such strong statements are possible only under restrictions on the choice of the rectifier function for the approximating neural network.

4.1. Choice of Rectifier Function.

We consider two classes of rectifier functions, namely sigmoidal functions [7] and smooth approximations of rectified linear units, both defined formally next.

We start with sigmoidal rectifiers as considered in [7, 5].

Definition 4.1. *A continuous function $\rho : \mathbb{R} \to \mathbb{R}$ is called a* sigmoidal function of order $k \in \mathbb{N}$, $k \geq 2$, *if there exists $K > 0$ such that*

$$\lim_{x \to -\infty} \frac{1}{x^k}\rho(x) = 0, \quad \lim_{x \to \infty} \frac{1}{x^k}\rho(x) = 1, \quad \textit{and} \quad |\rho(x)| \leq K(1 + |x|)^k, \textit{ for } x \in \mathbb{R}.$$

The function ρ is called a strong sigmoidal function of order k, *if there exist constants $a, b, K > 0$ such that*

$$\left|\frac{1}{x^k}\rho(x)\right| \leq K|x|^{-a}, \textit{for } x < 0, \ \left|\frac{1}{x^k}\rho(x) - 1\right| \leq Kx^{-a}, \textit{for } x \geq 0, |\rho(x)| \leq K(1 + |x|)^k, \textit{ for } x \in \mathbb{R},$$

and

$$\left|\frac{d}{dx}\rho(x)\right| \leq K|x|^b, \quad \textit{for } x \in \mathbb{R}.$$

One of the most commonly used rectifier functions is the so-called rectified linear unit (ReLU) given by $x \mapsto \max\{0, x\}$. The second class of rectifiers we consider are smooth versions of the ReLU.

Definition 4.2. *Let $\rho : \mathbb{R} \to \mathbb{R}^+$, $\rho \in C^\infty(\mathbb{R})$ satisfy*

$$\rho(x) = \begin{cases} 0, & \textit{for } x \leq 0, \\ x, & \textit{for } x \geq K, \end{cases}$$

for some constant $K > 0$. Then, we call ρ an admissible smooth rectifier.

The reason for considering these two specific classes of rectifiers resides in the fact that neural networks based thereon allow effective representations of multivariate bump functions. We demonstrate in the sequel, that the effective representation of multivariate bump functions by neural networks leads to the effective representation of all affine systems by neural networks. Approximation of multivariate bump functions using sparsely connected neural networks is a classical topic in neural network theory [32]. What is new here is the aspect of quantized weights.

In [5] it was shown that B-splines can be well approximated by neural networks with sigmoidal rectifiers. It is instructive to recall this result. To this end, for $m \in \mathbb{N}$, we denote the univariate cardinal B-spline of order $m \in \mathbb{N}$ by N_m, i.e., $N_1 = \chi_{[0,1]}$, and $N_{m+1} = N_m * \chi_{[0,1]}$, for all $m \geq 1$. Multivariate B-splines are simply tensor products of univariate B-splines. Specifically, we define, for $d \in \mathbb{N}$, the d-dimensional B-spline of order m by N_m^d.

Theorem 4.3 ([5], Thm. 4.2)**.** *Let $d, m, k \in \mathbb{N}$, and take ρ to be a sigmoidal function of order $k \geq 2$. Further, let $L := \lceil \log(md - d)/\log(k) \rceil + 1$. Then, there is $M \in \mathbb{N}$, possibly dependent on d, m, k, such that for all $D, \epsilon > 0$, there exists a network $\Phi_{D,\epsilon} \in \mathcal{NN}_{L,M,d,\rho}$ with*

$$\|N_m^d - \Phi_{D,\epsilon}\|_{L^2([-D,D]^d)} \leq \epsilon.$$

This result alone does, however, not establish that dictionaries based on B-splines are effectively representable by neural networks. For this to hold we need the weights in the approximating neural network $\Phi_{D,\epsilon}$ to be polynomially bounded in $1/\epsilon$. We next show that this is, indeed, possible for strongly sigmoidal rectifiers.

Theorem 4.4. *Let $d, m, k \in \mathbb{N}$, and ρ strongly sigmoidal of order $k \geq 2$. Further, let $L := \lceil \log(md-d)/\log(k) \rceil + 1$. Then, there exists $M \in \mathbb{N}$, possibly dependent on s, m, k, such that for all $D, \epsilon > 0$, there is a network $\Phi_{D,\epsilon} \in \mathcal{NN}_{L,M,d,\rho}$ with*

$$\|N_m^d - \Phi_{D,\epsilon}\|_{L^2([-D,D]^d)} \leq \epsilon.$$

Moreover, the weights of the network $\Phi_{D,\epsilon}$ are polynomially bounded in (D, ϵ^{-1}).

Proof. The neural network $\Phi_{D,\epsilon}$ in Theorem 4.3 is explicitly constructed in [5]. Carefully following the steps in that construction and making explicit use of the strong sigmoidality of ρ, as opposed to plain sigmoidality as in [5], yields the desired result. □

Likewise, taking ρ to be an admissible smooth rectifier, we can ensure that a neural network with only 3 layers yields a smooth bump function. The corresponding result is as follows.

Theorem 4.5. *Let ρ be an admissible smooth rectifier. Then, for all $d \in \mathbb{N}$, there exist $M \in \mathbb{N}$ and a neural network $\Phi_\rho \in \mathcal{NN}_{3,M,d,\rho}$ such that*

(i) Φ_ρ is compactly supported,
(ii) $\Phi_\rho \in C^\infty(\mathbb{R})$, and
(iii) $\hat{\Phi}_\rho(\xi) \neq 0$, for all $\xi \in \mathbb{R}^d$, with $|\xi| \leq 3$.

Proof. For simplicity of exposition we consider the case $d = 2$ only; the general case requires minor modifications. We start by constructing an auxiliary function as follows. For $0 < p_1 \leq p_2 \leq p_3$ such that $p_1 + p_2 = p_3$, define $t : \mathbb{R} \to \mathbb{R}$ as

$$t(x) := \rho(x) - \rho(x - p_1) - \rho(x - p_2) + \rho(x - p_3). \tag{26}$$

Then, t is smooth and compactly supported. Letting $q = \|t\|_{L^\infty(\mathbb{R})}$, we choose $g : \mathbb{R}^2 \to \mathbb{R}$ according to

$$g(x_1, x_2) := \rho(t(x_1) + t(x_2) - q). \tag{27}$$

By construction, g is smooth and compactly supported. Moreover, the two-step design of g per (26) and (27) allows the realization of g by a three layer neural network. Since $g > 0$, it follows that $|\hat{g}(0)| > 0$. Hence, there exists a $\delta > 0$ such that $|\hat{g}(\xi)| > 0$ for all $\xi \in B_\delta(0)$. We now set

$$\varphi := g\left(\frac{\cdot}{3\delta}\right),$$

which can then also be constructed by a neural network $\Phi_\rho \in \mathcal{NN}_{3,M,2,\rho}$ for some $M \in \mathbb{N}$. Since, by construction, $|\hat{\varphi}(\xi)| > 0$ for all $\xi \in B_3(0)$, the function φ and hence Φ_ρ indeed satisfy the desired assumptions. □

4.2. **Invariance to Affine Transformations.** We next leverage the approximation results for bump functions to demonstrate that a wide class of representation systems is effectively representable by neural networks.

As a first step towards establishing this general result, we show in this section that representability of a function f by neural networks in the sense of Definition 3.1 is invariant to affine transformations and finite linear combinations of shifts of f. Moreover, we will see that if f can be approximated by a neural network to within an error of ϵ and the network's weights are polynomially bounded in ϵ^{-1}, the same holds true for the approximating networks corresponding to affine transformations and finite linear combinations of f.

Proposition 4.6. *Let $d \in \mathbb{N}$ and $\rho : \mathbb{R} \to \mathbb{R}$. Assume that there exist constants $M, L \in \mathbb{N}$ such that for all $D, \epsilon > 0$ there is $\Phi_{D,\epsilon} \in \mathcal{NN}_{L,M,d,\rho}$ with*

$$\|f - \Phi_{D,\epsilon}\|_{L^2([-D,D]^d)} \le \epsilon. \tag{28}$$

Let $A \in \mathbb{R}^{d\times d}$ be an expanding[†] *matrix and $b \in \mathbb{R}^d$. Then, there exists M' depending on M and d only such that for all $E, \eta > 0$ there is $\Psi_{E,\eta} \in \mathcal{NN}_{L,M',d,\rho}$ with*

$$\|\det(A)^{1/2} f(A\cdot -b) - \Psi_{E,\eta}\|_{L^2([-E,E]^d)} \le \eta. \tag{29}$$

If the weights of $\Phi_{D,\epsilon}$ are polynomially bounded in ϵ^{-1}, then the weights of $\Psi_{E,\eta}$ are polynomially bounded in $(|\det(A)|, E, |b|, \eta^{-1})$.

Proof. The proof proceeds by noting that for every $\Phi \in \mathcal{NN}_{L,M,d,\rho}$, we have

$$\|\det(A)^{1/2} f(A\cdot - b) - \det(A)^{1/2}\Phi(A\cdot - b)\|_{L^2([-E,E]^d)} = \|f - \Phi\|_{L^2(A\cdot[-E,E]^d - b)},$$

and there exists M' depending on M and d only such that $\det(A)^{1/2}\Phi(A\cdot - b) \in \mathcal{NN}_{L,M',d,\rho}$. Since the matrix A is expanding, there exists a constant $c > 0$ independent of d such that

$$A\cdot[-E,E]^d - b \subset [-c\cdot(|\det(A)|\cdot E + |b|), c\cdot(|\det(A)|\cdot E + |b|)]^d.$$

Therefore, the network $\Psi_{E,\eta} := \det(A)^{1/2}\Phi_{\det(A)\cdot E+|b|,\eta}(A\cdot -b)$ meets the desired properties. □

Next, we show that effective representability by neural networks is preserved under finite linear combinations of translates.

Proposition 4.7. *Let $d \in \mathbb{N}$ and $\rho : \mathbb{R} \to \mathbb{R}$. Assume that there exist constants $M, L \in \mathbb{N}$ such that for all $D, \epsilon > 0$ there is $\Phi_{D,\epsilon} \in \mathcal{NN}_{L,M,d,\rho}$ with*

$$\|f - \Phi_{D,\epsilon}\|_{L^2([-D,D]^d)} \le \epsilon. \tag{30}$$

Let $r \in \mathbb{N}$, $(c_i)_{i=1}^r \subset \mathbb{R}$ and $(d_i)_{i=1}^r \subset \mathbb{R}^d$. Then, there exists M' depending on M, d, and r only such that for all $E, \eta > 0$ there is $\Psi_{E,\eta} \in \mathcal{NN}_{L,M',d,\rho}$ with

$$\left\|\sum_{i=1}^r c_i f(\cdot - d_i) - \Psi_{E,\eta}\right\|_{L^2([-E,E]^d)} \le \eta. \tag{31}$$

If the weights of $\Phi_{D,\epsilon}$ are polynomially bounded in ϵ^{-1}, then the weights of $\Psi_{E,\eta}$ are polynomially bounded in $(\sum_{i=1}^r |c_i|, E, \max_{i=1}^r |d_i|, \eta^{-1})$.

[†] A matrix $A \in \mathbb{R}^{d\times d}$ is said to be expanding if all its eigenvalues are greater than or equal to 1.

Proof. We first note that for $\Phi \in \mathcal{NN}_{L,M,d,\rho}$ satisfying (30), the function

$$\Psi(\cdot) := \sum_{i=1}^{r} c_i \Phi(\cdot - d_i)$$

is in $\mathcal{NN}_{L,M',d,\rho}$ with M' depending on d, r, and M only. Moreover, by the triangle inequality, we have for all $D, \epsilon > 0$ that

$$\left\| \sum_{i=1}^{r} c_i f(\cdot - d_i) - \sum_{i=1}^{r} c_i \Phi_{D,\epsilon}(\cdot - d_i) \right\|_{L^2([-E,E]^d)} \leq \left(\sum_{i=1}^{r} |c_i| \right) \cdot \max_{i=1,\dots,r} \|f - \Phi_{D,\epsilon}\|_{L^2([-(E+|d_i|),(E+|d_i|)]^d)}.$$

Setting $D = E + \max_{i=1,\dots,r} |d_i|$ and $\epsilon = \eta / (\sum_{i=1}^{r} |c_i|)$, the network

$$\Psi_{E,\eta}(\cdot) := \sum_{i=1}^{r} c_i \Phi_{D,\epsilon}(\cdot - d_i)$$

exhibits the desired properties. □

The operations introduced in Propositions 4.6 and 4.7 can be used to construct neural networks which approximate functions with a given number of vanishing moments. This property will turn out crucial in demonstrating representability of affine systems.

Definition 4.8. *Let $R, d \in \mathbb{N}$, and $k \in \{1, \dots, d\}$. A function $g \in C(\mathbb{R}^d)$ is said to possess R directional vanishing moments in x_k-direction, if*

$$\int_{\mathbb{R}} x_k^{\ell} g(x_1, \dots, x_k, \dots, x_d) dx_k = 0 \quad \textit{for all } x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_d \in \mathbb{R},\ \ell \in \{0, \dots, R-1\}.$$

The next lemma establishes that an arbitrary number of vanishing moments in a given coordinate direction can be generated by suitable linear combinations of translates of a given function of compact support.

Lemma 4.9. *Let $R, d \in \mathbb{N}$, $B > 0$, $k \in \{1, \dots, d\}$, and $f \in C(\mathbb{R}^d)$ with compact support. Then, the function*

$$g(x_1, \dots, x_d) := \sum_{\ell=0}^{R-1} \binom{R-1}{\ell} (-1)^{\ell} f(x_1, \dots, x_k - \ell/B, \dots, x_d) \tag{32}$$

has R directional vanishing moments in x_k-direction.

Furthermore, if $\hat{f}(\xi) \neq 0$ for all $\xi \in \mathbb{R}^d$ with $|\xi| < B$, then

$$\hat{g}(\xi) \neq 0, \quad \textit{for all } \xi \in \mathbb{R}^d \textit{ with } 0 < |\xi| < B. \tag{33}$$

Proof. For simplicity of exposition, we consider the case $B = 1$ only. Taking the Fourier transform of (32) we see that

$$\hat{g}(\xi) = \sum_{\ell=0}^{R-1} \binom{R-1}{\ell} (-1)^{\ell} e^{2\pi i \ell \xi_k} \hat{f}(\xi) = \left(1 - e^{2\pi i \xi_k}\right)^{R-1} \cdot \hat{f}(\xi) \tag{34}$$

which implies

$$\frac{\partial^{\ell}}{\partial \xi_k^{\ell}} \hat{g}(\xi_1, \dots, \xi_{k-1}, 0, \xi_{k+1}, \dots, \xi_d) = 0, \quad \text{for all } \xi_1, \dots, \xi_{k-1}, \xi_{k+1}, \dots, \xi_d \in \mathbb{R},\ \ell \in \{0, \dots, R-1\}.$$

But looking at Definition 4.8, this property precisely says that g possesses the desired vanishing moments. Statement (33) follows by inspection of (34). □

Remark 4.10. *Lemma 4.9 and Definition 4.8 consider vanishing moments in one coordinate direction only. It is immediate that, along the same lines, vanishing moments in multiple coordinate directions according to*

$$\int_{\mathbb{R}^K} x_{k_1}^{\ell_1} \cdot \dots \cdot x_{k_K}^{\ell_K} g(x) dx_{k_1} \dots dx_{k_K} = 0$$

can be generated.

4.3. **Affine Representation Systems.** In this section, we introduce a family of representation systems, coined *affine systems*, that includes all representation systems based on affine scaling and translations of a given "mother function". Special cases of affine systems are wavelets, ridgelets, curvelets, shearlets, α-shearlets, and more generally α-molecules, as well as tensor products thereof.

The formal definition of affine systems is as follows.

Definition 4.11. *Let $d \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ bounded, and $f \in L^2(\mathbb{R}^d)$ compactly supported. Let $\delta > 0$, $(c_i^s)_{i=1}^r \subset \mathbb{R}$, for $s = 1, \dots, S$, and $(d_i)_{i=1}^r \subset \mathbb{R}^d$. Moreover, let $(A_j)_{j\in\mathbb{N}} \subset \mathbb{R}^{d\times d}$ be expanding. Consider the compactly supported functions*

$$g_s := \sum_{i=1}^r c_i^s f(\cdot - d_i), \quad s = 1, \dots, S.$$

We define the corresponding affine system *as*

$$\mathcal{D} := \left\{ g_s^{j,b} := \det(A_j)^{1/2} g_s(A_j \cdot - \delta \cdot b) \cdot \chi_\Omega : \ s = 1, \dots, S, \ b \in \mathbb{Z}^d, \ j \in \mathbb{N}, \ \textit{and } g_s^{j,b} \neq 0 \right\},$$

where χ_Ω denotes the indicator function of Ω.

For a given affine system according to Definition 4.11, we can establish a natural order of its elements by concatenating the sub-systems

$$\mathcal{D}_{s,j} = \left\{ g_s^{j,b} : \ b \in \mathbb{Z}^d, g_s^{j,b} \neq 0 \right\} \tag{35}$$

as follows. First, we note that, as a consequence of the compact support of the functions g_s, all sub-systems $\mathcal{D}_{s,j}$ have their cardinality bounded by

$$|\mathcal{D}_{s,j}| = O(|\det(A_j)|), \tag{36}$$

where the implicit constant depends on δ and Ω only. Each sub-system $\mathcal{D}_{s,j}$ may be ordered arbitrarily. The overall representation system $\mathcal{D}$ is organized according to

$$\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} = (\mathcal{D}_{1,1}, \dots, \mathcal{D}_{S,1}, \mathcal{D}_{1,2}, \dots, \mathcal{D}_{S,2}, \dots). \tag{37}$$

This ordering is assumed in the remainder of the paper and will sometimes be referred to as *canonical ordering*.

The next result establishes that *all affine systems are (effectively) representable by neural networks* in the sense of Definition 3.1.

Theorem 4.12. *Suppose that $\Omega \subset \mathbb{R}^d$ is bounded and $\mathcal{D} = (\varphi_i)_{i\in\mathbb{N}} \subset L^2(\Omega)$ is an affine system according to Definition 4.11. Suppose further that for the rectifier function $\rho : \mathbb{R} \to \mathbb{R}$ there exists a constant M such that for all $D, \epsilon > 0$ there is $\Phi_{D,\epsilon} \in \mathcal{NN}_{L,M,d,\rho}$ with*

$$\|f - \Phi_{D,\epsilon}\|_{L^2([-D,D]^d)} \leq \epsilon, \tag{38}$$

where f is as in Definition 4.11. Then, $\mathcal{D}$ is representable by neural networks with rectifier function ρ in the sense of Definition 3.1.

If, in addition, the weights of $\Phi_{D,\epsilon}$ are polynomially bounded in (D, ϵ^{-1}), and if there exist $a > 0$ and $c > 0$ such that for all $j \in \mathbb{N}$

$$\sum_{k=1}^{j-1} \det(A_k) \geq c|\det(A_j)|^a, \tag{39}$$

then $\mathcal{D}$ is effectively representable by neural networks with rectifier function ρ.

Proof. Without loss of generality, we assume that $\Omega = [-D, D]^d$ for some fixed $D > 0$, otherwise we pick D such that $\Omega \subset [-D, D]^d$. We first show that (38) implies representability of $\mathcal{D}$ by neural networks with rectifier function ρ. To this end, we need to establish the existence of a constant $C > 0$ such that for all $i \in \mathbb{N}$ and all $\eta > 0$ there exist $\Phi_{i,\eta} \in \mathcal{NN}_{\infty,C,d,\rho}$ with

$$\|\varphi_i - \Phi_{i,\eta}\|_{L^2(\Omega)} \leq \eta. \tag{40}$$

The elements of $\mathcal{D}$ are constructed through dilations and translations of f according to

$$\varphi_i = \det(A_j)^{1/2} \sum_{k=1}^r c_k^s f(A_j \cdot - \delta b - d_k), \tag{41}$$

for some $s \in \{1, \ldots, S\}$, $j \in \mathbb{N}$, and $b \in \mathbb{R}^d$. One can now see that for all $\varphi_i \in \mathcal{D}$ we have $|b| = O(|\det(A_j)|)$ in (41). Thus (40) follows directly from Propositions 4.6 and 4.7.

It remains to show that a polynomial bound in $D \cdot \epsilon^{-1}$ on the weights of $\Phi_{D,\epsilon}$ in (38) implies that $\mathcal{D}$ is effectively representable by neural networks with rectifier function ρ. By Definition 3.1 this means that the coefficients of $\Phi_{i,\eta}$ grow at most polynomially in (i, η^{-1}). Consider φ_i as in (41). Propositions 4.6 and 4.7 state that the weights of $\Phi_{i,\eta}$ are polynomially bounded in

$$\left(|\det(A_j)|, D, |b|, \sum_{i=1}^{r} |c_i|, E, \max_{i=1,\ldots,r} |d_i|, \eta^{-1}\right).$$

Since $|b| = O(|\det(A_j)|)$ and the quantities D, $\sum_{i=1}^{r} |c_i|$, E, and $\max_{i=1,\ldots,r} |d_i|$ do not depend on i, we can absorb them into a fixed constant and conclude that the weights of $\Phi_{i,\eta}$ are polynomially bounded in

$$(|\det(A_j)|, \eta^{-1}). \tag{42}$$

It remains to show that the quantities $|\det(A_j)|$ are polynomially bounded in i. To see this, consider the canonical ordering (37) and observe that φ_i according to (41) satisfies $\varphi_i \in \mathcal{D}_{s,j}$ for some $s \in \{1, \ldots, S\}$, where $\mathcal{D}_{s,j}$ was defined in (35). Together with the bound (36) it follows that, thanks to the canonical ordering, there exists a constant $c > 0$ such that

$$i \geq c \cdot \sum_{k=1}^{j-1} |\det(A_k)|.$$

We finally appeal to (39) to conclude that $|\det(A_j)|$ is polynomially bounded in i which, together with (42), establishes the desired result. □

Note that condition (39) is very weak and we are not aware of an affine representation system that would violate it.

We now present the main technical result of this paper, namely that *neural networks provide optimal approximations for all signal classes that are optimally approximated by any affine system*!

Theorem 4.13. *Assume that $\Omega \subset \mathbb{R}^d$ is bounded and $\mathcal{D} = (\varphi_i)_{i \in \mathbb{N}} \subset L^2(\Omega)$ is an affine system according to Definition 4.11. Assume further that for a rectifier function $\rho : \mathbb{R} \to \mathbb{R}$ there exists a constant M such that for all $D, \epsilon > 0$ there is $\Phi_{D,\epsilon} \in \mathcal{NN}_{L,M,d,\rho}$ satisfying (38). Then, for all signal classes $\mathcal{C} \subset L^2(\Omega)$, we have*

$$\gamma^*_{\mathcal{NN}}(\mathcal{C}) \geq \gamma^*(\mathcal{C}, \mathcal{D}).$$

If, in addition, the weights of $\Phi_{D,\epsilon}$ in (38) are polynomially bounded in (D, ϵ^{-1}), and if there exist $a > 0$ and $c > 0$ such that for all $j \in \mathbb{N}$

$$\sum_{k=1}^{j-1} \det(A_k) \geq c|\det(A_j)|^a, \tag{43}$$

and, if $\mathcal{C} \subset L^2(\Omega)$ is effectively representable in $\mathcal{D}$ (according to Definition 2.2), then, for all $\gamma < \gamma^*(\mathcal{C}, \mathcal{D})$ there exist $c, C > 0$ and a mapping

$$\mathbf{Learn} : \left(0, \frac{1}{2}\right) \times L^2(\Omega) \to \mathcal{NN}_{\infty,\infty,d,\rho},$$

such that for every $f \in \mathcal{C}$

(i) each weight of the neural network $\mathbf{Learn}(\epsilon, f)$ can be represented with at most $-c\log_2(\epsilon)$ bits,

(ii) the estimate

$$\|f - \mathbf{Learn}(\epsilon, f)\|_{L^2(\Omega)} \leq \epsilon$$

holds, and

(iii) the neural network $\mathbf{Learn}(\epsilon, f)$ has at most $C \cdot \epsilon^{-\gamma}$ edges.

Specifically, optimality of $\mathcal{D}$ for the signal class $\mathcal{C}$ implies optimal representability of $\mathcal{C}$ by neural networks.

Proof. The proof follows directly by combining Theorem 4.12 with Theorems 3.4 and 3.5. □

Theorem 4.13 reveals a remarkable universality property of neural networks. In particular, it provides an answer to the question posed at the beginning of Section 3: As soon as a signal class can be optimally represented by an affine system, the bound in Theorem 2.8 is tight!

5. α-Shearlets and Cartoon-Like Functions

We next present a pair $(\mathcal{C}, \mathcal{D})$ of signal class and dictionary such that $\gamma^*_{\mathcal{NN}}(\mathcal{C}) = \gamma^*(\mathcal{C}, \mathcal{D})$. Specifically, we consider α-shearlets as representation systems in $L^2(\mathbb{R}^2)$ and $(1/\alpha)$-cartoon-like functions as signal class. We point out that the definition of α-shearlets in this paper differs slightly from that in [22]. Our definition replaces $1/\alpha$ by α such that α-shearlets are α-molecules, whereas in [22] α-shearlets are a special case of $1/\alpha$-molecules. We will need scaling and shearing matrices defined as

$$D_{\alpha,a} = \begin{pmatrix} a & 0 \\ 0 & a^\alpha \end{pmatrix}, \qquad J = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{and} \quad S_k = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}.$$

This leads us to the following definition which is a slightly modified version of the corresponding definition in [39].

Definition 5.1 ([39])**.** *For $\delta \in \mathbb{R}^+$, $\alpha \in [0,1]$, and $f, g \in L^2(\mathbb{R}^d)$, the* cone-adapted α-shearlet system $\mathcal{SH}_\alpha(f,g,\delta)$ *generated by $f, g \in L^2(\mathbb{R}^2)$ is defined as*

$$\mathcal{SH}_\alpha(f,g,\delta) := \mathcal{SH}^0_\alpha(f,g,\delta) \cup \mathcal{SH}^1_\alpha(f,g,\delta),$$

where

$$\mathcal{SH}^0_\alpha(f,g,\delta) := \left\{ f(\cdot - t), t \in \delta\mathbb{Z}^2 \right\},$$
$$\mathcal{SH}^1_\alpha(f,g,\delta) := \left\{ 2^{\ell(1+\alpha)/2}\psi(S_k D_{\alpha,2^\ell} J^\tau \cdot - t),\ \ell \in \mathbb{N}_0,\ |k| \le \lceil 2^{\ell(1-\alpha)} \rceil,\ t \in \delta\mathbb{Z}^2,\ k \in \mathbb{Z}^2,\ \tau \in \{0,1\} \right\}.$$

We mention that α-shearlets fall into the framework of α-molecules [22]. Our interest in α-shearlets stems from the fact that they yield optimal M-term approximation rates for β-cartoon-like functions defined as follows.

Definition 5.2. *Let $\beta \in [1,2)$, and $\nu > 0$. Define*

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

where $f_0, f_1 \in C^\beta_0([0,1]^2)$, $B \subset [0,1]^2$, $\partial B \in C^\beta$, and $\|f_1\|_{C^\beta}, \|f_2\|_{C^\beta}, \|\partial B\|_{C^\beta} < \nu$. The elements of $\mathcal{E}^\beta(\mathbb{R}^2;\nu)$ are called β-cartoon-like functions.

This function class was originally introduced in [15] as a model class for functions governed by curvilinear discontinuities of prescribed regularity. In this sense, β-cartoon-like functions provide a convenient model for images governed by edges or for the solutions of transport equations which are often governed by singularity curves.

The optimal exponent $\gamma^*(\mathcal{E}^\beta(\mathbb{R}^2;\nu))$ was found in [15, 23]:

Theorem 5.3. *For $\beta \in [1,2)$, and $\nu > 0$, we have*

$$\gamma^*(\mathcal{E}^\beta(\mathbb{R}^2;\nu)) = \frac{2}{\beta}.$$

Proof. The proof of [15, Thm. 2] demonstrates that $\mathcal{C} \subset L^2(\mathbb{R}^2)$ has optimal exponent $\gamma^*(\mathcal{C}) = \frac{2p}{2-p}$ if $\mathcal{C}$ contains a copy of ℓ^p_0. The result now follows, since by [23], the function class $\mathcal{E}^\beta(\mathbb{R}^2;\nu)$ does, indeed, contain a copy of ℓ^p_0 for $p = \frac{2}{\beta+1}$. □

Using Theorem 2.8 this result allows us to conclude that, independently of the learning algorithm employed, and asymptotically in ϵ, there always exists a function $f \in \mathcal{E}^\beta(\mathbb{R}^2;\nu)$ such that the learned network approximating f with an accuracy of $\epsilon < M^{-\beta/2}$ must have more than $O(M)$ edges. It will turn out that a neural network exists which admits precisely this optimal number of edges.

The following theorem now shows that α-shearlets yield best M-term approximation rates for $1/\alpha$-cartoon-like functions.

Theorem 5.4 ([39], Thm. 6.3 and Rem. 6.4)**.** *Let $1/2 \le \alpha \le 1$, $\nu > 0$, $f \in C^{12}(\mathbb{R}^2)$, $g \in C^{32}(\mathbb{R}^2)$, both compactly supported such that*

(i) $\widehat{f}(\xi) \neq 0$, for all $|\xi| \le 1$,
(ii) $\widehat{g(\xi)} \neq 0$, for all $\xi = (\xi_1, \xi_2)^T \in \mathbb{R}^2$ such that $1/3 \le |\xi_1| \le 3$ and $|\xi_2| \le |\xi_1|$,
(iii) g has at least 7 vanishing moments in x_1-direction, i.e.,

$$\int_{\mathbb{R}} x_1^\ell g(x_1, x_2) dx_1 = 0, \quad \textit{for all } x_2 \in \mathbb{R},\ \ell \in \{0, \dots, 6\}.$$

Then, there exists a constant $\delta^* > 0$ such that, for all $\delta < \delta^*$, the system $\mathcal{SH}_\alpha(f, g, \delta)$ is optimal for the signal class $\mathcal{E}^{1/\alpha}(\mathbb{R}^2; \nu)$ in the sense of Definition 2.5.

Remark 5.5. *The assumptions on the smoothness and vanishing moments of f and g in Theorem 5.4 follow from* [39, Equation 4.9] *with $s_1 = 3/2, s_0 = 0, p_0 = q_0 = 2/3$, and $|\beta| \leq 4$. While these particular choices allow the statement of the theorem to be independent of α, it is possible to weaken the assumptions, if a fixed α is considered. For example, for $\alpha = 1/2$ the smoothness assumptions on f and g reduce to $f \in C^{11}, g \in C^{28}$.*

We proceed to the main statement of this section.

Theorem 5.6. *Suppose that $\rho : \mathbb{R} \to \mathbb{R}$ is either a strongly sigmoidal function of order $k \geq 2$ (see Definition 4.1) or an admissible smooth rectifier (see Definition 4.2). Then, for every $\beta \in [1, 2)$, the signal class $\mathcal{E}^\beta(\mathbb{R}^2; \nu)$ is optimally representable by a neural network with rectifier function ρ.*

Proof. In view of Theorem 5.4 all we have to show is that, for suitable f and g, $\mathcal{SH}_\alpha(f, g, \delta)$ is a subset of an affine system satisfying the assumptions of Theorem 4.13.

First, note that by Theorems 4.4 and 4.5 a function f satisfying the assumptions of Theorem 5.4 can be approximated arbitrarily well by a neural network that has its number of edges and its weights polynomially bounded in the inverse of the approximation error. Furthermore, by Lemma 4.9 it is possible to construct from this f a function g which satisfies the assumptions of Theorem 5.4. In view of Definition 4.11, we put $g_1 = f$ and $g_2 = g$ so that for suitable matrices $(A_j)_{j\in\mathbb{N}}$ the system $\mathcal{SH}_\alpha(f, g, \delta)$ is a subset of an affine system with generators g_1 and g_2. Specifically, these matrices A_j are of the form $S_k D_{\alpha,2^\ell} J^\tau$ for $\ell \in \mathbb{N}_0, |k| \leq \lceil 2^{\ell(1-\alpha)} \rceil, \tau \in \{0, 1\}$ and we order them such that their determinants increase monotonically. It is easily verified that (43) is satisfied and therefore we can apply Theorem 4.13 to infer the optimality of neural networks for the approximation of $1/\alpha$-cartoon functions. □

6. Generalization To Manifolds

Frequently, a target function f to be approximated by a neural network models functions that live on a (possibly low-dimensional) immersed submanifold $\Gamma \subset \mathbb{R}^d$ of dimension $m < d$. We next briefly describe how our main results can be extended to this situation. Since analogous results are already contained in [43] for the case of wavelets we will allow ourselves to be somewhat informal in certain places.

Suppose that $f : \Gamma \to \mathbb{R}$ is compactly supported. Let $(U_i)_{i\in\mathbb{N}} \subset \Gamma$ be an open cover of Γ such that for each $i \in \mathbb{N}$ the manifold patch U_i can be parametrized as the graph of a function over a subset of the Euclidean coordinates, e.g. there exist coordinates $x_{d_1}, \ldots, x_{d_m}$, open sets $V_i \subset \mathbb{R}^m$, and smooth mappings

$$\gamma_\ell : \mathbb{R}^m \to \mathbb{R}, \quad \ell \in \{1, \ldots, d\} \setminus \{d_1, \ldots, d_m\}$$

such that

$$U_i = \{\Xi_i(x_{d_1}, \ldots, x_{d_m}) := (\gamma_1(x_{d_1}, \ldots, x_{d_m}), \ldots, x_{d_1}, \ldots, \gamma_d(x_{d_1}, \ldots, x_{d_m})) : (x_{d_1}, \ldots, x_{d_m}) \in V_i\}.$$

Pick a smooth partition of unity $(h_i)_{i\in\mathbb{N}}$, where $h_i : \Gamma \to \mathbb{R}$ is smooth with $\operatorname{supp}(h_i) \subset \overline{U_i}$ and $\sum_{i\in\mathbb{N}} h_i = 1$. Define the localization of f to U_i by $f_i := f \cdot h_i$ such that

$$f = \sum_{i\in\mathbb{N}} f_i. \tag{44}$$

Every $f_i : U_i \to \mathbb{R}$ can be reparametrized to

$$\tilde{f}_i : \begin{cases} \mathbb{R}^m & \to & \mathbb{R} \\ (x_{d_1}, \ldots, x_{d_m}) & \mapsto & f_i \circ \Xi_i(x_{d_1}, \ldots, x_{d_m}). \end{cases}$$

Suppose that there exists a neural network $\tilde{\Phi}_i \in \mathcal{NN}_{L,M,m,\rho}$ with

$$\|\tilde{f}_i - \tilde{\Phi}_i\|_{L^2(V_i)} \leq \epsilon. \tag{45}$$

Then, we can construct a neural network $\Phi_i \in \mathcal{NN}_{L,M+m\cdot d,d,\rho}$ according to

$$\Phi_i(x) := \tilde{\Phi}_i(P_i x),$$

where P_i denotes the orthogonal projection of x onto the coordinates $(x_{d_1}, \ldots, x_{d_m})$. Since P is linear, Φ_i is a neural network. Moreover, since P_i is the inverse of the diffeomorphism Ξ_i, we get

$$\|\Phi_i - f_i\|_{L^2(U_i)} \leq C \cdot \epsilon,$$

with $C > 0$ depending on the curvature of $\Gamma|_{U_i}$ only. Now we may build a neural network Φ by setting $\Phi := \sum_{i\in\mathbb{N}} \Phi_i$. Combining (45) with the observation that, owing to the compact support of f, only a finite number of summands appears in the definition of f, we have constructed a neural network Φ which approximates f on Γ. We thus observe the following.

Whenever a signal class $\mathcal{C}$ is invariant w.r.t. diffeomorphisms (in our construction the functions Ξ_i) and multiplication by smooth functions (in our construction the functions h_i), then approximation results on $\mathbb{R}^m$ can be leveraged to approximation results on m-dimensional submanifolds $\Gamma \subset \mathbb{R}^d$.

Such invariances are, in particular, satisfied for all signal classes characterized by a particular smoothness behavior; for example, the class of cartoon-functions as studied in Section 5. Somewhat informally we thus summarize that *neural networks are optimal for the approximation of piecewise smooth functions on manifolds*.

7. Numerical Results

Our theoretical results in Section 2 show that, for every function class $\mathcal{C} \subseteq L^2(\mathbb{R}^d)$, the number of edges of a neural network stemming from a fixed learning algorithm must obey a fundamental lower bound. One of the most widely used learning algorithms is stochastic gradient descent with the gradient computed via backpropagation [41]. The purpose of this section is to investigate how the stochastic gradient descent algorithm fares relative to our lower bound.

Surprisingly, our numerical experiments indicate that for a fixed sparsely connected network topology inspired by the construction of bump functions in Equations (26) and (27) and with the ReLU as rectifier, the stochastic gradient descent algorithm generates neural networks that achieve high M-edge approximation rates. In particular, we identify two classes where these rates are optimal.

The topology that we prescribe is depicted in Figure 3. Let us briefly explain why we chose this network topology. As mentioned before, admissible smooth rectifiers consist of smooth functions which equal a ReLU outside a compact interval. For this class of rectifiers, the appropriate shearlet generators were constructed from a function g as specified in (27). Choosing $p_1 = p_2 = 1$ and $p_3 = 2$ in (26) yields hat functions t. This construction implies that we only require six nodes in the first layer in each subnetwork. Additionally, the derivative of the resulting functions t does not vanish on the support of t. Both properties are beneficial in terms of the learning speed of the stochastic gradient descent algorithm.

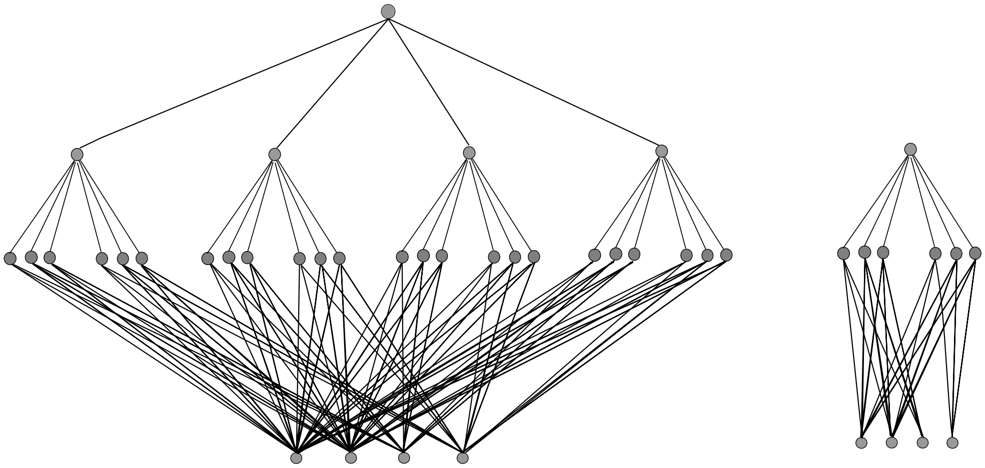
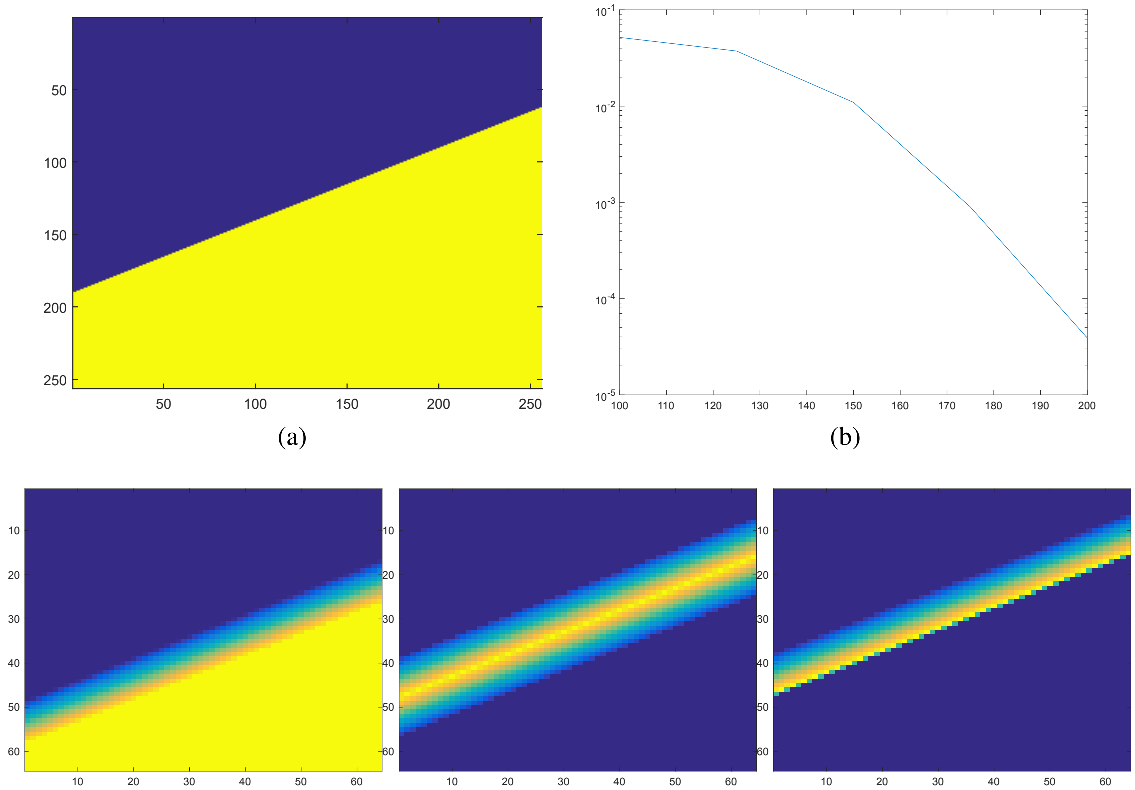


FIGURE 3. **Left:** Topology of the neural network trained using stochastic gradient descent. The network consists of a weighted sum of four subnetworks. The four inputs are $x_1, x_2 \in [-1, 1]$ plus two inputs set to 1. **Right:** A single subnetwork.

In Figure 3, we see four network realizations of g in parallel. The input layer consists of four inputs. The first two inputs represent x_1, x_2, whereas the third and fourth inputs are constant. The first four neurons in the first layer correspond to the construction (26). The first neuron in the second layer corresponds to (27). In the output layer, all incoming signals are added to yield a sum of the subnetworks.

We now train the network using the backpropagation algorithm. Following (27) the weights associated with the edges between the first and the second layer remain fixed, and only the weights of the edges between the input and the first layer as well as the edges between the second layer and the output layer are trained. Training is performed for two different functions, where one is a function with a line singularity (Figure 4(a)), and the other one is a cartoon-like function (Figure 5(a)). We train the network by drawing samples $(x_1, x_2) \in [-1, 1]^2$ from an equispaced grid in $[-1, 1]^2$ to which we then apply the network. The error determined by comparing to the original function is backpropagated through the network. We repeat this procedure for different network sizes, i.e., for different numbers of subnetworks.



(a)

(b)

(c)

(d)

(e)

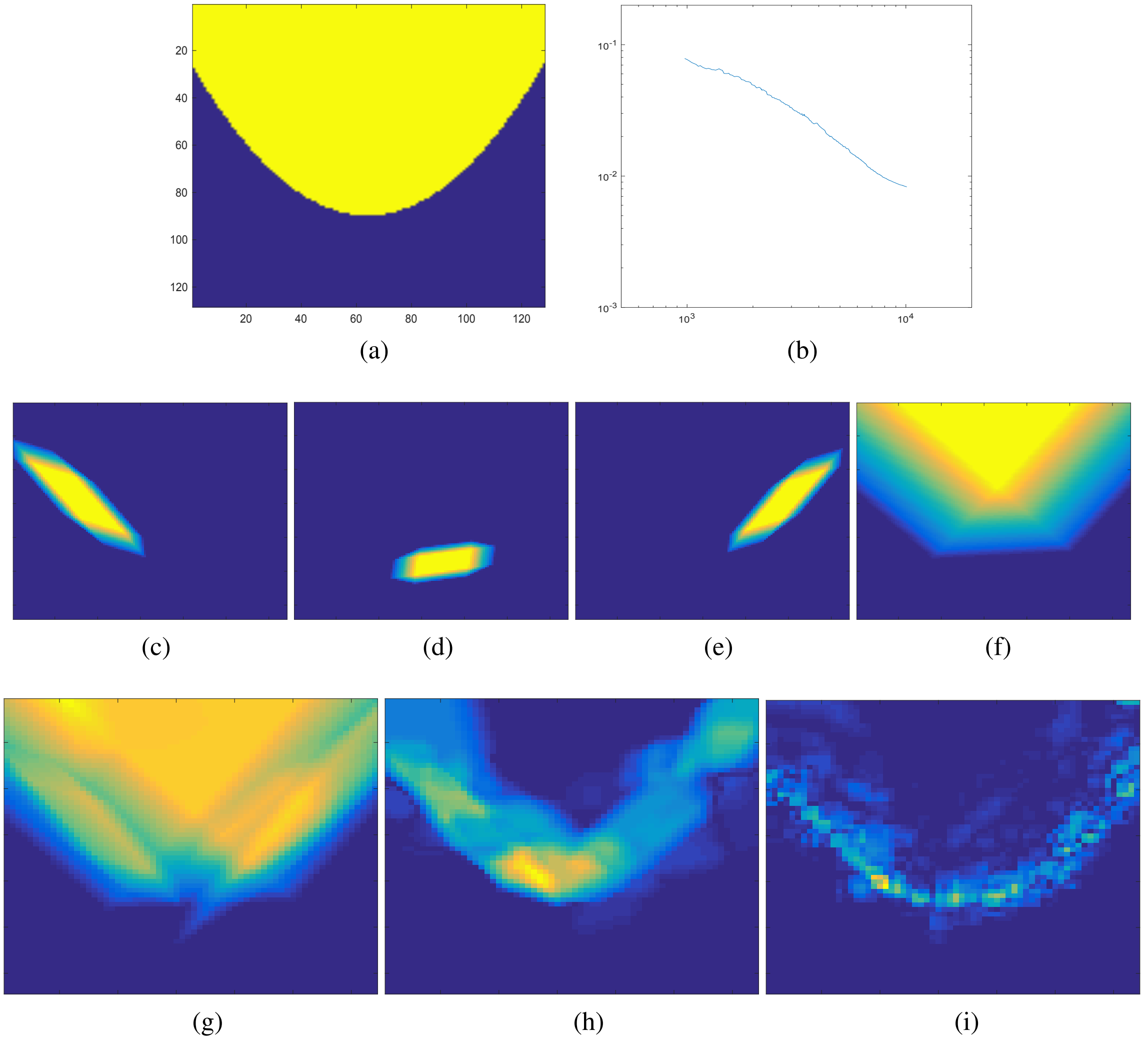
FIGURE 4. (a): Function with a line singularity. (b): Approximation error as a function of the number of edges. (c)-(e): Depiction of the functions obtained by restricting to the subnetworks with the largest weights in modulus in the final layer.

We start by discussing the results for the function with a line singularity in Figure 4(a). The approximation error corresponding to the trained neural network is shown in Figure 4(b). The super-linear decay of the error in the semi-logarithmic scale indicates super-exponential decay of the error with respect to the number of edges.

Also, interestingly, the trained subnetworks yield α-molecules for $\alpha = 0$ (see Figures 4(c)-(e)). These functions are constant along one direction and vary along another, hence can be considered part of a ridgelet system, which is, in fact, an optimally sparsifying system for line singularities. Moreover, the orientation of the three learned ridge functions matches that of the original function.

In the second experiment, we draw samples from the function depicted in Figure 5(a) below, which exhibits a curvilinear singularity. Figures 5(c)-(e) show that the corresponding trained subnetworks resemble anisotropic molecules with different scales and of different orientations. Concerning the approximation error, we report that the expected decay rate of $O(N^{-1})$ was not visible when simply training with different network sizes. However, with a slight adaptation one obtains the result of Figure 5(b), which demonstrates a decay of roughly N^{-1}. The specifics of this adaptation are as follows: We first train a large network with $\sim$ 10000 edges, again by stochastic gradient descent.

Then, the weights in the last layer are optimized using the Lasso [45] to obtain a sparse weight vector c^*. Considering networks built from a combination of the subnetworks associated with the N largest coefficients of c^* yields a sequence of networks with corresponding errors according to Figure 5(b). Finally, we investigate whether the approximation delivered by this procedure is similar to what would be obtained by best N-term approximation with standard shearlet systems, where it is known [31, 25] that shearlet elements on high scales tend to cluster around singularities. We analyze this behavior in Figures 5(g)-(i). Here, we depict in Figure 5(g) the weighted sum of those subnetworks that have the largest support. In Figure 5(h) we show weighted sums of subnetworks with medium-sized support, and in Figure 5(i) we sum up only the subnetworks with the smallest supports. We observe that, indeed, subnetworks of large support approximate the smooth part of the underlying function, whereas the small-sized subnetworks resolve the jump singularity.



(a) (b)

(c) (d) (e) (f)

(g) (h) (i)

FIGURE 5. (a): Function with curvilinear singularity to be approximated by the network. (b): Approximation error in terms of edges. (c)-(f): Shearlet-like subnetworks. (g): Reconstruction using only the 10 subnetworks with the largest supports. (h): Reconstruction using only subnetworks with medium-sized support. (i): Reconstruction using only subnetworks with very small support.

ACKNOWLEDGMENTS

The authors would like to thank J. Bruna, E. Candès, M. Genzel, S. Güntürk, Y. LeCun, K.-R. Müller, H. Rauhut, and F. Voigtländer for interesting discussions. G.K. and P.P. are grateful to the Faculty of Mathematics at the Universität Wien for the hospitality and support during their visits. Moreover, G.K. thanks the Department of Mathematics at Stanford University whose support allowed for completion of a portion of this work. G.K. acknowledges partial support by the Einstein Foundation Berlin, the Einstein Center for Mathematics Berlin (ECMath), the European Commission-Project DEDALE (contract no. 665044) within the H2020 Framework Program, DFG Grant KU 1446/18, DFG-SPP 1798 Grants KU 1446/21 and KU 1446/23, and by the DFG Research Center MATHEON "Mathematics for Key Technologies". G.K. and P.P acknowledge support by the DFG Collaborative Research Center TRR 109 "Discretization in Geometry and Dynamics".

REFERENCES

[1] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39(3):930–945, 1993.

[2] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Mach. Learn.*, 14(1):115–133, 1994.

[3] E. J. Candès. Ridgelets: Theory and Applications, 1998. Ph.D. thesis, Stanford University.

[4] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise-C2 singularities. *Comm. Pure Appl. Math.*, 57:219–266, 2002.

[5] C. K. Chui, X. Li, and H. N. Mhaskar. Neural networks for localized approximation. *Math. Comp.*, 63(208):607–623, 1994.

[6] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore. Tree approximation and optimal encoding. *Appl. Comput. Harmon. Anal.*, 11(2):192–226, 2001.

[7] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signal*, 2(4):303–314, 1989.

[8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[9] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[10] M. Dauge and R. Stevenson. Sparse tensor product wavelet approximation of singular functions. *SIAM J. Math. Anal.*, 42(5):2203–2228, 2010.

[11] L. Demanet and L. Ying. Wave atoms and sparsity of oscillatory patterns. *Appl. Comput. Harmon. Anal.*, 23(3):368–387, 2007.

[12] R. DeVore, K. Oskolkov, and P. Petrushev. Approximation by feed-forward neural networks. *Ann. Numer. Math.*, 4:261–287, 1996.

[13] R. A. DeVore and G. G. Lorentz. *Constructive Approximation*. Springer, 1993.

[14] D. L. Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. *Appl. Comput. Harmon. Anal.*, 1(1):100 – 115, 1993.

[15] D. L. Donoho. Sparse components of images and optimal atomic decompositions. *Constr. Approx.*, 17(3):353–382, 2001.

[16] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA*, pages 907–940, 2016.

[17] S. Ellacott. Aspects of the numerical analysis of neural networks. *Acta Numer.*, 3:145–202, 1994.

[18] K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

[19] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[20] K. Gröchenig. *Foundations of time-frequency analysis*. Springer Science & Business Media, 2013.

[21] P. Grohs. Optimally sparse data representations. In *Harmonic and Applied Analysis*, pages 199–248. Springer, 2015.

[22] P. Grohs, S. Keiper, G. Kutyniok, and M. Schäfer. α-molecules. *Appl. Comput. Harmon. Anal.*, 41(1):297–336, 2016.

[23] P. Grohs, S. Keiper, G. Kutyniok, and M. Schäfer. Cartoon approximation with α -curvelets. *J. Fourier Anal. Appl.*, 22(6):1235–1293, 2016.

[24] P. Grohs and G. Kutyniok. Parabolic molecules. *Found. Comput. Math.*, 14:299–337, 2014.

[25] K. Guo, G. Kutyniok, and D. Labate. Sparse multidimensional representations using anisotropic dilation and shear operators. *Wavelets and Splines (Athens, GA, 2005)*, pages 189–201, 2006.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[27] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.*, 29(6):82–97, 2012.

[28] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

[29] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[31] G. Kutyniok and W.-Q. Lim. Compactly supported shearlets are optimally sparse. *Journal of Approximation Theory*, 163(11):1564 – 1589, 2011.

[32] Y. LeCun. *Modèles connexionnistes de l'apprentissage*. PhD thesis, These de Doctorat, Université Paris 6, 1987.

[33] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[34] W. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5:115–133, 1943.

[35] H. Mhaskar and C. Micchelli. Degree of approximation by neural and translation networks with a single hidden layer. *Adv. Appl. Math.*, 16(2):151–183, 1995.

[36] H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.*, 8(1):164–177, 1996.

[37] H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

[38] T. Nguyen-Thien and T. Tran-Cong. Approximation of functions and their derivatives: A neural network implementation with applications. *Appl. Math. Model.*, 23(9):687–704, 1999.

[39] A. Pein and F. Voigtländer. Analysis sparsity versus synthesis sparsity for α-shearlets. 2017. arXiv:1702.03559.

[40] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numer.*, 8:143–195, 1999.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Neurocomputing: Foundations of research: Learning representations by back-propagating errors. pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[42] M. Schmitt. Lower bounds on the complexity of approximating continuous functions by sigmoidal neural networks. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pages 328–334, Cambridge, MA, USA, 1999. MIT Press.

[43] U. Shaham, A. Cloninger, and R. R. Coifman. Provable approximation properties for deep neural networks. *Appl. Comput. Harmon. Anal.* to appear.

[44] I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer New York, 2008.

[45] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Series B Stat. Methodol.*, 58(1):267–288, 1996.

DEPARTMENT OF INFORMATION TECHNOLOGY AND ELECTICAL ENGINEERING, ETH ZÜRICH, 8092 ZÜRICH, SWITZERLAND
E-mail address: `boelcskei@nari.ee.ethz.ch`

FACULTY OF MATHEMATICS, UNIVERSITY OF VIENNA, 1090 VIENNA, AUSTRIA
E-mail address: `philipp.grohs@univie.ac.at`

INSTITUT FÜR MATHEMATIK, TECHNISCHE UNIVERSITÄT BERLIN, 10623 BERLIN, GERMANY
E-mail address: `kutyniok@math.tu-berlin.de`

INSTITUT FÜR MATHEMATIK, TECHNISCHE UNIVERSITÄT BERLIN, 10623 BERLIN, GERMANY
E-mail address: `petersen@math.tu-berlin.de`