Understanding Geometry of Encoder-Decoder CNNs

Jong Chul Ye [1 2] **Woon Kyoung Sung** [2]

Abstract

Encoder-decoder networks using convolutional neural network (CNN) architecture have been extensively used in deep learning literatures thanks to its excellent performance for various inverse problems. However, it is still difficult to obtain coherent geometric view why such an architecture gives the desired performance. Inspired by recent theoretical understanding on generalizability, expressivity and optimization landscape of neural networks, as well as the theory of deep convolutional framelets, here we provide a unified theoretical framework that leads to a better understanding of geometry of encoder-decoder CNNs. Our unified framework shows that encoder-decoder CNN architecture is closely related to nonlinear frame representation using combinatorial convolution frames, whose expressivity increases exponentially with the depth. We also demonstrate the importance of skipped connection in terms of expressivity, and optimization landscape.

1. Introduction

For the last decade, we have witnessed the unprecedented success of deep neural networks (DNN) in various applications in computer vision, classification, medical imaging, etc. Aside from traditional applications such as classification (Krizhevsky et al., 2012), segmentation (Ronneberger et al., 2015), image denoising (Zhang et al., 2017), super-resolution (Kim et al., 2016), etc, deep learning approaches have already become the state-of-the-art technologies in various inverse problems in x-ray CT, MRI, etc (Kang et al., 2017; Jin et al., 2017; Hammernik et al., 2018)

However, the more we see the success of deep learning, the more mysterious the nature of deep neural networks becomes. In particular, the amazing aspects of *expressive power, generalization capability*, and *optimization landscape* of DNNs have become an intellectual challenge for machine learning community, leading to many new theoretical results with varying capacities to facilitate the understanding of deep neural networks (Ge & Ma, 2017; Hanin & Sellke, 2017; Yarotsky, 2017; Nguyen & Hein, 2017; Arora et al., 2016; Du et al., 2018; Raghu et al., 2017; Bartlett et al., 2017; Neyshabur et al., 2018; Nguyen & Hein, 2018; Rolnick & Tegmark, 2017; Shen, 2018).

In inverse problems, one of the most widely employed network architectures is so-called encoder-decoder CNN architectures (Ronneberger et al., 2015). In contrast to the simplified form of the neural networks that are often used in theoretical analysis, these encoder-decoder CNNs usually have more complicated network architectures such as symmetric network configuration, skipped connections, etc. Therefore, it is not clear how the aforementioned theory can be used to understand the geometry of encoder-decoder CNNs to examine the origin of their superior performance.

Recently, the authors in (Ye et al., 2018) proposed so-called deep convolutional framelets to explain the encoder-decoder CNN architecture from a signal processing perspective. The main idea is that a data-driven decomposition of Hankel matrix constructed from the input data provides encoder-decoder layers that have striking similarity to the encoder-decoder CNNs. However, one of the main weaknesses of the theory is that it is not clear where the exponential expressiveness comes from. Moreover, many theoretical issues of neural networks such as generalizability and the optimization landscape, which have been extensively studied in machine learning literature, have not been addressed.

Therefore, this work aims at filling the gap and finding the connections between machine learning and signal processing to provide a unified theoretical analysis that facilitates the geometric understanding of encoder-decoder CNNs. Accordingly, we have revealed the following geometric features of encoder-decoder CNNs:

- An encoder-decoder CNN with an over-parameterized feature layer approximates a map between two smooth manifolds that is decomposed as a high-dimensional embedding followed by a quotient map.

[1]Dept. of Bio/Brain Engineering, KAIST Daejeon 34141, Republic of Korea. [2]Dept. of Mathematical Sciences, KAIST, Daejeon 34141, Republic of Korea.. Correspondence to: Jong Chul Ye <jong.ye@kaist.ac.kr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

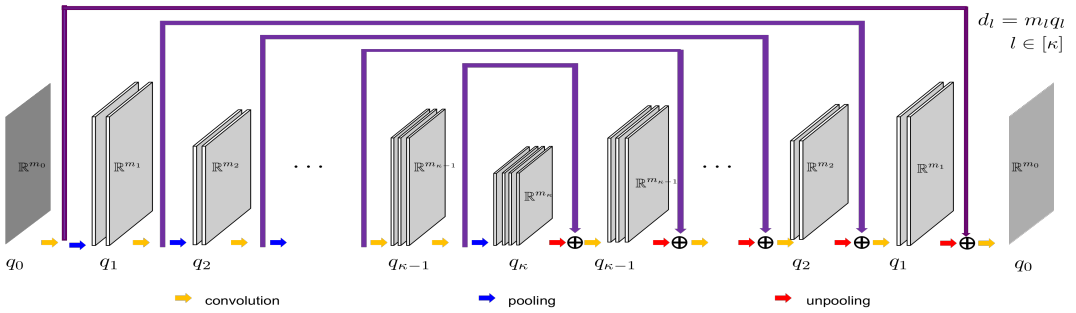

Figure 1. An architecture of κ-layer symmetric encoder-decoder CNN with skipped connections. Here, q_l denotes the number of channels at the l-th layer, whereas m_l refers to each channel dimension, and d_l represents the total dimension of the feature at the l-th layer.

- An encoder-decoder CNN with ReLU nonlinearity can be understood as deep convolutional framelets that use combinatorial frames of spatially varying convolutions. Accordingly, the number of linear representations increases exponentially with the network depth. This also suggests that the input space is divided into non-overlapping areas where each area shares the common linear representation.
- We derive an explicit form of the Lipschitz condition that determines the generalization capability of the encoder-decoder CNNs. The expression shows that the expressiveness of the network is not affected by the control of the Lipschitz constant.
- We provide explicit conditions under which the optimization landscape for encoder-decoder CNNs is benign. Specifically, we show that the skipped connection play important roles in smoothing out the optimization landscape.

All the proof of the theorems and lemmas in this paper are included in the Supplementary Material.

2. Related Works

Choromanska et al (Choromanska et al., 2015) employed the spin glass model from statistical physics to analyze the representation power of deep neural networks. Telgarsky constructs interesting classes of functions that can be only computed efficiently by deep ReLU nets, but not by shallower networks with a similar number of parameters (Telgarsky, 2016). Arora et al (Arora et al., 2016) showed that for every natural number k there exists a ReLU network with k^2 hidden layers and total size of k^2, which can be represented by $\frac{1}{2}k^{k+1} - 1$ neurons with at most k-hidden layers. All these results agree that the expressive power of deep neural networks increases exponentially with the network depth.

The generalization capability have been addressed in terms of various complexity measures such as Rademacher complexity (Bartlett & Mendelson, 2002), VC bound (Anthony & Bartlett, 2009), Kolmorogov complexity (Schmidhuber, 1997), etc. However, a recent work (Zhang et al., 2016) showed intriguing results that these classical bounds are too pessimistic to explain the generalizability of deep neural networks. Moreover, it has been repeatedly shown that over-parameterized deep neural networks, which are trained with fewer samples than the number of neurons, generalize well rather than overfitting (Cohen et al., 2018; Wei et al., 2018; Brutzkus et al., 2017; Du & Lee, 2018), which phenomenon cannot be explained by the classical complexity results.

The optimization landscape of neural networks have been another important theoretical issue in neural networks. Originally observed in linear deep neural networks (Kawaguchi, 2016), the benign optimization landscape has been consistently observed in various neural networks (Du et al., 2018; Nguyen & Hein, 2018; Du et al., 2017; Nguyen & Hein, 2017).

However, these theoretical works mainly focus on simplified network architectures, and we are not aware of analysis for encoder-decoder CNNs.

3. Encoder-Decoder CNNs

3.1. Definition

In this section, we provide a formal definition of encoder-decoder CNNs (E-D CNNs) to facilitate the theoretical analysis. Although our definition is for 1-dimensional signals, its extension to 2-D images is straightforward.

3.1.1. Basic Architecture

Consider encoder-decoder networks in Fig. 1. Specifically, the encoder network maps a given input signal $x \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^{d_0}$ to a feature space $z \in \boldsymbol{\mathcal{Z}} \subset \mathbb{R}^{d_\kappa}$, whereas the decoder takes this feature map as an input, process it and produce an output $y \in \boldsymbol{\mathcal{Y}} \subset \mathbb{R}^{d_L}$. In this paper, symmetric configuration is considered so that both encoder and decoder have the same number of layers, say κ; the input and output dimensions for the encoder layer $\mathcal{E}^l$ and the decoder layer

$\mathcal{D}^l$ are symmetric:

$$\mathcal{E}^l : \mathbb{R}^{d_{l-1}} \mapsto \mathbb{R}^{d_l}, \quad \mathcal{D}^l : \mathbb{R}^{d_l} \mapsto \mathbb{R}^{d_{l-1}}$$

where $l \in [\kappa]$ with $[n]$ denoting the set $\{1, \cdots, n\}$; and both input and output dimension is d_0. More specifically, the l-th layer input signal for the encoder layer comes from q_{l-1} number of input channels:

$$\xi^{l-1} = \begin{bmatrix} \xi_1^{l-1\top} & \cdots & \xi_{q_{l-1}}^{l-1\top} \end{bmatrix}^\top \in \mathbb{R}^{d_{l-1}},$$

where $^\top$ denotes the transpose, and $\xi_j^{l-1} \in \mathbb{R}^{m_{l-1}}$ refers to the j-th channel input with the dimension m_{l-1}. Therefore, the overall input dimension is given by $d_{l-1} := m_{l-1} q_{l-1}$. Then, the l-th layer encoder generates q_l channel output using the convolution operation:

$$\xi_j^l = \sigma\left(\Phi^{l\top} \sum_{k=1}^{q_{l-1}} \left(\xi_k^{l-1} \circledast \overline{\psi}_{j,k}^l \right) \right), \; j \in [q_l] \tag{1}$$

where $\xi_j^l \in \mathbb{R}^{m_l}$ refers to the j-th channel output after the convolutional filtering with the r-tap filters $\overline{\psi}_{j,k}^l \in \mathbb{R}^r$ and pooling operation $\Phi^{l\top} \in \mathbb{R}^{m_l \times m_{l-1}}$, and $\sigma(\cdot)$ denotes the element wise rectified linear unit (ReLU). More specifically, $\overline{\psi}_{j,k}^l \in \mathbb{R}^r$ denotes the r-tap convolutional kernel that is convolved with the k-th input to contribute to the output of the j-th channel, $\circledast$ is the circular convolution via periodic boundary condition to avoid special treatment of the convolution at the boundary, and $\overline{v}$ refers to the flipped version of the vector v. For the formal definition of the convolution operation used in this paper, see Appendix A in Supplementary Material.

Moreover, as shown in Appendix B in Supplementary Material, an equivalent matrix representation of the encoder layer is then given by

$$\xi^l := \sigma(E^{l\top} \xi^{l-1}) = \begin{bmatrix} \xi_1^{l\top} & \cdots & \xi_{q_l}^{l\top} \end{bmatrix}^\top$$

where $E^l \in \mathbb{R}^{d_{l-1} \times d_l}$ is computed by[1]

$$E^l = \begin{bmatrix} \Phi^l \circledast \psi_{1,1}^l & \cdots & \Phi^l \circledast \psi_{q_l,1}^l \\ \vdots & \ddots & \vdots \\ \Phi^l \circledast \psi_{1,q_{l-1}}^l & \cdots & \Phi^l \circledast \psi_{q_l,q_{l-1}}^l \end{bmatrix} \tag{2}$$

with

$$\begin{bmatrix} \Phi^l \circledast \psi_{i,j}^l \end{bmatrix} := \begin{bmatrix} \phi_1^l \circledast \psi_{i,j}^l & \cdots & \phi_{m_l}^l \circledast \psi_{i,j}^l \end{bmatrix} \tag{3}$$

On the other hand, the l-th layer input signal for the decoder layer comes from q_l channel inputs, i.e. $\tilde{\xi}^l = \begin{bmatrix} \tilde{\xi}_1^{l\top} & \cdots & \tilde{\xi}_{q_l}^{l\top} \end{bmatrix}^\top \in \mathbb{R}^{d_l}$, and the decoder layer convolution is given by

$$\tilde{\xi}_j^{l-1} = \sigma\left(\sum_{k=1}^{q_l} \left(\tilde{\Phi}^l \tilde{\xi}_k^l \circledast \tilde{\psi}_{j,k}^l \right) \right), \quad j \in [q_{l-1}] \tag{4}$$

where the unpooling layer is denoted by $\tilde{\Phi}^l \in \mathbb{R}^{m_{l-1} \times m_l}$. Note that (1) and (4) differ in their order of the pooling or unpooling layers. Specifically, a pooling operation is applied after the convolution at the encoder layer, whereas, at the decoder, an unpooling operation is performed before the convolution to maintain the symmetry of the networks. In matrix form, a decoder layer is given by

$$\tilde{\xi}^{l-1} := \sigma(D^l \tilde{\xi}^l) = \begin{bmatrix} \tilde{\xi}_1^{l-1\top} & \cdots & \tilde{\xi}_{q_{l-1}}^{l-1\top} \end{bmatrix}^\top$$

where $D^l \in \mathbb{R}^{d_l \times d_{l-1}}$ is computed by

$$D^l = \begin{bmatrix} \tilde{\Phi}^l \circledast \tilde{\psi}_{1,1}^l & \cdots & \tilde{\Phi}^l \circledast \tilde{\psi}_{1,q_l}^l \\ \vdots & \ddots & \vdots \\ \tilde{\Phi}^l \circledast \tilde{\psi}_{q_{l-1},1}^l & \cdots & \tilde{\Phi}^l \circledast \tilde{\psi}_{q_{l-1},q_l}^l \end{bmatrix} \tag{5}$$

3.1.2. E-D CNN WITH SKIPPED CONNECTION

As shown in Fig. 1, a skipped connection is often used to bypass an encoder layer output to a decoder layer. The corresponding filtering operation at the l-th layer encoder is described by

$$\begin{bmatrix} \xi_j^l \\ \chi_j^l \end{bmatrix} = \begin{bmatrix} \sigma\left(\Phi^{l\top} \sum_{k=1}^{q_{l-1}} \left(\xi_k^{l-1} \circledast \overline{\psi}_{j,k}^l \right) \right) \\ \sigma\left(\sum_{k=1}^{q_{l-1}} \left(\xi_k^{l-1} \circledast \overline{\psi}_{j,k}^l \right) \right) \end{bmatrix} \tag{6}$$

where χ_j^l and ξ_j^l denote the skipped output, and the pooled output via $\Phi^{l\top}$, respectively, after the filtering with $\overline{\psi}_{j,k}$. As shown in Fig. 1, the skipped branch is no more filtered at the subsequent layer, but is merged at the symmetric decoder layer:

$$\tilde{\xi}_j^{l-1} = \sigma\left(\sum_{k=1}^{q_l} \left((\tilde{\Phi}^l \tilde{\xi}_k^l + \chi_k^l) \circledast \tilde{\psi}_{j,k}^l \right) \right)$$

In matrix form, the encoder layer with the skipped connection can be represented by

$$\mathcal{E}^l : \xi^{l-1} \mapsto \begin{bmatrix} \xi^{l\top} & \chi^{l\top} \end{bmatrix}^\top$$

where

$$\xi^l := \sigma(E^{l\top} \xi^{l-1}) \quad , \quad \chi^l := \sigma(S^{l\top} \xi^{l-1}) \tag{7}$$

where E^l is given in (2) and the skipped branch filter matrix S^l is represented by

$$S^l = \begin{bmatrix} I_{m_{l-1}} \circledast \psi_{1,1}^l & \cdots & I_{m_{l-1}} \circledast \psi_{q_l,1}^l \\ \vdots & \ddots & \vdots \\ I_{m_{l-1}} \circledast \psi_{1,q_{l-1}}^l & \cdots & I_{m_{l-1}} \circledast \psi_{q_l,q_{l-1}}^l \end{bmatrix} \tag{8}$$

[1] Here, without loss of generality, bias term is not explicitly shown, since it can be incorporated into the matrix E^l and D^l as an additional column.

where $I_{m_{l-1}}$ denotes the $m_{l-1} \times m_{l-1}$ identity matrix. This implies that we can regard the skipped branch as the identity pooling $I_{m_{l-1}}$ applied to the filtered signals. Here, we denote the output dimension of the skipped connection as

$$s_l := m_{l-1} q_l \quad .$$

Then, the skipped branch at the l-th encoder layer is merged at the l-th decoder layer, which is defined as

$$\mathcal{D}^l : \begin{bmatrix} \tilde{\xi}^{l\top} & \chi^{l\top} \end{bmatrix}^\top \mapsto \tilde{\xi}^{l-1}$$

where

$$\tilde{\xi}^{l-1} := \sigma(D^l \tilde{\xi}^l + \tilde{S}^l \chi^l) \tag{9}$$

and D^l is defined in (5), and $\tilde{S}^l$ is given by

$$\tilde{S}^l = \begin{bmatrix} I_{m_{l-1}} \circledast \tilde{\psi}^l_{1,1} & \cdots & I_{m_{l-1}} \circledast \tilde{\psi}^l_{1,q_l} \\ \vdots & \ddots & \vdots \\ I_{m_{l-1}} \circledast \tilde{\psi}^l_{q_{l-1},1} & \cdots & I_{m_{l-1}} \circledast \tilde{\psi}^l_{q_{l-1},q_l} \end{bmatrix} \tag{10}$$

3.2. Parameterization of E-D CNNs

At the l-th encoder (resp. decoder) layer, there are $q_l q_{l-1}$ filter set that generates the q_l (resp. q_{l-1}) output channels from q_{l-1} (resp. q_l) input channels. In many CNNs, the filter lengths are set to equal across the layer. In our case, we set this as r, so the number of filter coefficients for the l-layer is

$$n_l := r q_l q_{l-1}, \quad l \in [\kappa]$$

These parameters should be estimated during the training phase. Specifically, by denoting the set of all parameter matrices $\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}_E \times \boldsymbol{\mathcal{W}}_D$ where $\boldsymbol{\mathcal{W}}_E := \mathbb{R}^{n_\kappa} \times \cdots \times \mathbb{R}^{n_1}$ and $\boldsymbol{\mathcal{W}}_D := \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_\kappa}$, we compose all layer-wise maps to define an encoder-decoder CNN as

$$z = F(\mathbf{W}, x). \tag{11}$$

Regardless of the existence of skipped connections, note that the same number of unknown parameters is used because the skipped connection uses the same set of filters.

4. Theoretical Analysis of E-D CNNs

4.1. Differential Topology

First, we briefly revisit the work by Shen (Shen, 2018), which gives an topological insight on the E-D CNNs.

Proposition 1 (Extension of Theorem 3 in (Shen, 2018))**.** *Let $f : \boldsymbol{\mathcal{X}} \mapsto \boldsymbol{\mathcal{Y}} \subset \mathbb{R}^q$ be a continuous map of smooth manifolds such that $f = g \circ h$, where $g : \mathbb{R}^p \mapsto \mathbb{R}^q$ with $p \geq q$ is a Lipschitz continuous map. If $p > 2 \dim \boldsymbol{\mathcal{X}}$, then there exists a smooth embedding $\tilde{h} : \boldsymbol{\mathcal{X}} \mapsto \mathbb{R}^p$, so that the following inequality holds true for a chosen norm and all $x \in \boldsymbol{\mathcal{X}}$ and $\epsilon > 0$:*

$$\|f(x) - g \circ \tilde{h}(x)\| \leq \epsilon$$

Here, $p > 2 \dim \boldsymbol{\mathcal{X}}$ comes from the weak Whitney embedding theorem (Whitney, 1936; Tu, 2011). Note that Theorem 1 informs that a neural network, designed as a continuous map of smooth manifolds, can be considered as an approximation of a task map that is composed of a smooth embedding followed by an additional map. In fact, this decomposition is quite general for a map between smooth manifolds as shown in the following proposition:

Proposition 2. *(Shen, 2018) Let $f : \boldsymbol{\mathcal{X}} \mapsto \boldsymbol{\mathcal{Y}} \subset \mathbb{R}^q$ be a map of smooth manifolds, then the task f admits a decomposition of $f = g \circ h$, where $h : \boldsymbol{\mathcal{X}} \mapsto \boldsymbol{\mathcal{Z}} \subset \mathbb{R}^p$ with $p \geq 2 \dim \boldsymbol{\mathcal{X}}$ is a smooth embedding. Furthermore, the task map f is a quotient map, if and only if the map g is a quotient map.*

To understand the meaning of the last sentence in Proposition 2, we briefly review the concept of the quotient space and quotient map (Tu, 2011). Specifically, let $\sim$ be an equivalence relation on $\boldsymbol{\mathcal{X}}$. Then, the quotient space, $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} / \sim$ is defined to be the set of equivalence classes of elements of $\boldsymbol{\mathcal{X}}$. For example, we can declare images perturbed by noises as an equivalent class such that our quotient map is designed to map the noisy signals to its noiseless equivalent image.

It is remarkable that Proposition 1 and Proposition 2 give interpretable conditions for design parameters such as network width (i.e. no of channels), pooling layers, etc. For example, if there are no pooling layers, the dimensionality conditions in Proposition 1 and Proposition 2 can be easily met in practice by increasing the number of channels more than twice the input channels. With the pooling layers, one could calculate the number of channels in a similar way. In general, Proposition 1 and Proposition 2 strongly suggest an encoder-decoder architecture with the constraint $d_0 \leq d_1 \leq \cdots \leq d_\kappa$ with $d_\kappa > 2d_0$, where an encoder maps an input signal to higher dimensional feature space whose dimension is at least twice bigger than the input space. Then, the decoder determines the nature of the overall neural network.

4.2. Links to the frame representation

One of the important contributions of recent theory of deep convolutional framelets (Ye et al., 2018) is that encoder-decoder CNNs have an interesting link to multi-scale convolution framelet expansion. To see this, we first define filter matrices $\Psi^l \in \mathbb{R}^{r q_{l-1} \times q_l}$ and $\tilde{\Psi}^l \in \mathbb{R}^{r q_{l-1} \times q_l}$ for encoder and decoder:

$$\Psi^l := \begin{bmatrix} \psi^l_{1,1} & \cdots & \psi^l_{q_l,1} \\ \vdots & \ddots & \vdots \\ \psi^l_{1,q_{l-1}} & \cdots & \psi^l_{q_l,q_{l-1}} \end{bmatrix}$$

$$\tilde{\Psi}^l := \begin{bmatrix} \tilde{\psi}^l_{1,1} & \cdots & \tilde{\psi}^l_{1,q_l} \\ \vdots & \ddots & \vdots \\ \tilde{\psi}^l_{q_{l-1},1} & \cdots & \tilde{\psi}^l_{q_{l-1},q_l} \end{bmatrix}$$

Then, the following proposition, which is novel and significantly extended from (Ye et al., 2018), states the importance of the frame conditions for the pooling layers and filters to obtain convolution framelet expansion (Yin et al., 2017).

Proposition 3. *Consider an encoder-decoder CNN without ReLU nonlinearities. Let $\Phi^{l\top}$ and $\tilde{\Phi}^l$ denote the l-th encoder and decoder layer pooling layers, respectively, and Ψ^l and $\tilde{\Psi}^l$ refer to the encoder and decoder filter matrices. Then, the following statements are true.*

1) For the encoder-decoder CNN without skipped connection, if the following frame conditions are satisfied for all $l \in [\kappa]$

$$\tilde{\Phi}^l \Phi^{l\top} = \alpha I_{m_{l-1}}, \; \Psi^l \tilde{\Psi}^{l\top} = \frac{1}{r\alpha} I_{rq_{l-1}} \tag{12}$$

then we have

$$x \;=\; \sum_i \langle b_i, x \rangle \tilde{b}_i \tag{13}$$

where b_i and $\tilde{b}_i$ denote the i-th column of the following frame basis and its dual:

$$B \;=\; E^1 E^2 \cdots E^\kappa, \tag{14}$$
$$\tilde{B} \;=\; D^1 D^2 \cdots D^\kappa \tag{15}$$

2) For the encoder-decoder CNN with skipped connection, if the following frame conditions are satisfied for all $l \in [\kappa]$:

$$\tilde{\Phi}^l \Phi^{l\top} = \alpha I_{m_{l-1}}, \; \Psi^l \tilde{\Psi}^{l\top} = \frac{1}{r(\alpha+1)} I_{rq_{l-1}} \tag{16}$$

then (13) holds, where b_i and $\tilde{b}_i$ denote the i-th column of the following frame and its duals:

$$B^{skp} \quad (\in \mathbb{R}^{d_0 \times (d_\kappa + \sum_{l=1}^\kappa s_l)}) \tag{17}$$
$$:= \begin{bmatrix} E^1 \cdots E^\kappa & E^1 \cdots E^{\kappa-1} S^\kappa & \cdots & E^1 S^2 & S^1 \end{bmatrix}$$

$$\tilde{B}^{skp} \quad (\in \mathbb{R}^{d_0 \times (d_\kappa + \sum_{l=1}^\kappa s_l)}) \tag{18}$$
$$:= \begin{bmatrix} D^1 \cdots D^\kappa & D^1 \cdots D^{\kappa-1} \tilde{S}^\kappa & \cdots & D^1 \tilde{S}^2 & \tilde{S}^1 \end{bmatrix}$$

Furthermore, the following corollary shows that the total basis and its dual indeed come from multiple convolutional operations across layers:

Corollary 4. *If there exist no pooling layers, then the t-th block of the frame basis matrix for $t \in [q_l]$ is given by*

$$\begin{bmatrix} E^1 \cdots E^l \end{bmatrix}_t = \begin{bmatrix} E^1 \cdots E^{l-1} S^l \end{bmatrix}_t$$

$$= I_m \circledast \left(\sum_{j_{l-1}, \cdots, j_1 = 1}^{q_{l-1}, \cdots, q_1} \psi^l_{j_1,1} \circledast \cdots \circledast \psi^l_{t, j_{l-1}} \right)$$

Similarly,

$$\begin{bmatrix} D^1 \cdots D^l \end{bmatrix}_t = \begin{bmatrix} D^1 \cdots D^{l-1} \tilde{S}^l \end{bmatrix}_t$$

$$= I_m \circledast \left(\sum_{j_{l-1}, \cdots, j_1 = 1}^{q_{l-1}, \cdots, q_1} \tilde{\psi}^l_{j_1,1} \circledast \cdots \circledast \tilde{\psi}^l_{t, j_{l-1}} \right)$$

This suggests that the length of the convolutional filters increases with the depth by cascading multiple convolution operations across the layers. While Proposition 3 informs that the skipped connection increases the dimension of the feature space from d_κ to $d_\kappa + \sum_{l=1}^\kappa s_l$, Corollary 4 suggest that the cascaded expression of the filters becomes more diverse for the case of encoder-decoder CNNs with skipped connection. Specifically, instead of convolving all κ layers of filters, the skipped connection allows the combination of subset of filters. All these make the frame representation from skipped connection more expressive.

4.3. Expressiveness

However, to satisfy the frame conditions (12) or (16), we need $q_l \geq r q_{l-1}$ so that the number of output filter channel q_l should increase exponentially. While this condition can be relaxed when the underlying signal has low-rank Hankel matrix structure (Ye et al., 2018), the explicit use of the frame condition is still rarely observed. Moreover, in contrast to the classical wavelet analysis, the perfect reconstruction condition itself is not interesting in neural networks, since the output of the network should be different from the input due to the task dependent processing.

Here, we claim that one of the important roles of using ReLU is that it allows combinatorial basis selection such that exponentially large number of basis expansion is feasible once the network is trained. This is in contrast with the standard framelet basis estimation. For example, for a given target data $Y = \begin{bmatrix} y^{(1)} & \cdots & y^{(T)} \end{bmatrix}$ and the input data $X = \begin{bmatrix} x^{(1)} & \cdots & x^{(T)} \end{bmatrix}$, the estimation problem of the frame basis and its dual in Proposition 3 is optimal for the given training data, but the network is not expressive and does not generalize well when the different type of input data is given. Thus, one of the important requirements is to allow large number of expressions that are adaptive to the different inputs.

Indeed, ReLU nonlinearity makes the network more expressive. For example, consider a trained two layer encoder-decoder CNN:

$$y = \tilde{B} \Lambda(x) B^\top x \tag{19}$$

where $\tilde{B} \in \mathbb{R}^{d_0 \times d_1}$ and $B \in \mathbb{R}^{d_0 \times d_1}$ and $\Lambda(x)$ is a diagonal matrix with 0, 1 elements that are determined by the ReLU output. Now, the matrix can be equivalently represented by

$$\tilde{B}\Lambda(x)B^\top = \sum_{i=1}^{d_1} \sigma_i(x)\tilde{b}_i b_i^\top \tag{20}$$

where $\sigma_i(x)$ refers to the (i,i)-th diagonal element of $\Lambda(x)$. Therefore, depending on the input data $x \in \mathbb{R}^{d_0}$, $\sigma_i(x)$ is either 0 or 1 so that a maximum 2^{d_1} distinct configurations of the matrix can be represented using (20), which is significantly more expressive than using the single representation with the frame and its dual. This observation can be generalized as shown in Theorem 5.

Theorem 5 (Expressiveness of encoder-decoder networks)**.** *Let*

$$\tilde{\Upsilon}^l = \tilde{\Upsilon}^l(x) := \tilde{\Upsilon}^{l-1}\tilde{\Lambda}^l(x)D^l, \tag{21}$$
$$\Upsilon^l = \Upsilon^l(x) := \Upsilon^{l-1}E^l\Lambda^l(x), \tag{22}$$

with $\tilde{\Upsilon}^0(x) = I_{d_0}$ *and* $\Upsilon^0(x) = I_{d_0}$*, and*

$$M^l = M^l(x) := S^l\Lambda^l_S(x) \tag{23}$$
$$\tilde{M}^l = \tilde{M}^l(x) := \tilde{\Lambda}^l(x)\tilde{S}^l \tag{24}$$

where $\Lambda^l(x)$ *and* $\tilde{\Lambda}^l(x)$ *refer to the diagonal matrices from ReLU at the* l*-th layer encoder and decoder, respectively, which have 1 or 0 values;* $\Lambda^l_S(x)$ *refers to a similarly defined diagonal matrices from ReLU at the* l*-th skipped branch of encoder. Then, the following statements are true.*

1) Under ReLUs, an encoder-decoder CNN without skipped connection can be represented by

$$y = \tilde{\mathcal{B}}(x)\mathcal{B}^\top(x)x = \sum_i \langle x, b_i(x)\rangle \tilde{b}_i(x) \tag{25}$$

where

$$\mathcal{B}(x) = \Upsilon^\kappa(x) \quad , \quad \tilde{\mathcal{B}}(x) = \tilde{\Upsilon}^\kappa(x) \tag{26}$$

Furthermore, the maximum number of available linear representation is given by

$$N_{rep} = 2^{\sum_{i=1}^{\kappa} d_i - d_\kappa}, \tag{27}$$

2) An encoder-decoder CNN with skipped connection under ReLUs is given by

$$y = \tilde{\mathcal{B}}^{skp}(x)\mathcal{B}^{skp\top}(x)x = \sum_i \langle x, b_i^{skp}(x)\rangle \tilde{b}_i^{skp}(x) \tag{28}$$

where

$$\mathcal{B}^{skp}(x) := \begin{bmatrix} \Upsilon^\kappa & \Upsilon^{\kappa-1}M^\kappa & \Upsilon^{\kappa-2}M^{\kappa-1} & \cdots & M^1 \end{bmatrix} \tag{29}$$

$$\tilde{\mathcal{B}}^{skp}(x) := \begin{bmatrix} \tilde{\Upsilon}^\kappa & \tilde{\Upsilon}^{\kappa-1}\tilde{M}^\kappa & \tilde{\Upsilon}^{\kappa-2}\tilde{M}^{\kappa-1} & \cdots & \tilde{M}^1 \end{bmatrix} \tag{30}$$

Furthermore, the maximum number of available linear representation is given by

$$N_{rep} = 2^{\sum_{i=1}^{\kappa} d_i - d_\kappa} \times 2^{\sum_{i=1}^{\kappa} s_k} \tag{31}$$

This implies that the number of representation increase exponentially with the network depth, which again confirm the expressive power of the neural network. Moreover, the skipped connection also significantly increases the expressive power of the encoder-decoder CNN. Another important consequence of Theorem 5 is that the input space $\boldsymbol{\mathcal{X}}$ is partitioned into the maximum N_{rep} non-overlapping regions so that inputs for each region shares the same linear representation.

Due to the ReLU, one may wonder whether the cascaded convolutional interpretation of the frame basis in Corollary 4 still holds. A close look of the proof of Corollary 4 reveals that this is still the case. Under ReLUs, note that $(I_m \circledast \psi^l_{j,s})(I_m \circledast \psi^{l+1}_{t,j}) = I_m \circledast (\psi^l_{j,s} \circledast \psi^{l+1}_{t,j})$ in Lemma 11 in Supplementary Material should be replaced with $(I_m \circledast \psi^l_{j,s})\Lambda^l_j(x)(I_m \circledast \psi^{l+1}_{t,j})$ where $\Lambda^l_j(x)$ is a diagonal matrix with 0 and 1 values due to the ReLU. This means that the $\Lambda^l_j(x)$ provides spatially varying mask to the convolution filter $\psi^{l+1}_{t,j}$ so that the net effect is a convolution with the the spatially varying filters originated from masked version of $\psi^{l+1}_{t,j}$. This results in a spatially variant cascaded convolution, and only change in the interpretation of Corollary 4 is that the basis and its dual are composed of *spatial variant* cascaded convolution filters. Furthermore, the ReLU works to diversify the convolution filters by masking out the various filter coefficients. It is believed that this is another source of expressiveness from the same set of convolutional filters.

4.4. Generalizability

To understand the generalization capability of DNNs, recent research efforts have been focused on reducing the gap by suggesting different ways of measuring the network capacity (Bartlett et al., 2017; Neyshabur et al., 2018). These works consistently showed the importance of Lipschitz condition for the encoder and decoder parts of the networks.

More specifically, we have shown that the neural network representation varies in exponentially many different forms depending on inputs, so one may be concerned that the output might vary drastically with small perturbation of the inputs. However, Lipschitz continuity of the neural network prevents such drastic changes. Specifically, a neural network

$F(\mathbf{W}, x)$ is Lipschitz continuous, if there exists a constant $K > 0$ such that

$$\|F(\mathbf{W}, x^{(1)}) - F(\mathbf{W}, x^{(2)})\|_2 \leq K\|x^{(1)} - x^{(2)}\|_2 \,.$$

where the Lipschitz constant K can be obtained by

$$K = \sup_{x \in \boldsymbol{\mathcal{X}}} \|D_2 F(\mathbf{W}, x)\|_2 \tag{32}$$

where $D_2 F(\mathbf{W}, x)$ is the Jacobian with respect to the second variable. The following proposition shows that the Lipschitz constant of encoder-decoder CNNs is closely related to the frame basis and its duals.

Proposition 6. *The Lipschitz constant for encoder-decoder CNN without skipped connection is given by*

$$K = \sup_{x \in \boldsymbol{\mathcal{X}}} \|\tilde{\mathcal{B}}(x)\mathcal{B}(x)^\top\|_2 \tag{33}$$

whereas Lipschitz constant for encoder-decoder CNN with skipped connection is given by

$$K = \sup_{x \in \boldsymbol{\mathcal{X}}} \|\tilde{\mathcal{B}}^{skp}(x)\mathcal{B}^{skp\top}(x)\|_2 \tag{34}$$

where $\mathcal{B}(x), \tilde{\mathcal{B}}(x), \mathcal{B}^{skp}(x)$ and $\tilde{\mathcal{B}}^{skp}(x)$ are defined in (26), (29) and (30).

Recall that the input space $\boldsymbol{\mathcal{X}}$ is partitioned into regions that share the same linear representation. Therefore, the local Lipschitz constant within the p-th partition is given by

$$\begin{aligned} K_p &= \sup_{z \in \boldsymbol{\mathcal{X}}_p} \|\tilde{\mathcal{B}}(z)\mathcal{B}^\top(z)\|_2 \\ &= \|\tilde{\mathcal{B}}(z_p)\mathcal{B}^\top(z_p)\|_2, \quad \forall z_p \in \boldsymbol{\mathcal{X}}_p \end{aligned} \tag{35}$$

for the case of E-D CNN without skipped connections. Here, $\boldsymbol{\mathcal{X}}_p$ denotes the p-th input space partition, and the last equality in (35) comes from the fact that every point in $\boldsymbol{\mathcal{X}}_p$ shares the same linear representation. Thus, it is easy to see that the global Lipschitz constant can be given by

$$K = \sup_{x \in \boldsymbol{\mathcal{X}}} \|\tilde{\mathcal{B}}(x)\mathcal{B}(x)^\top\|_2 = \sup_p K_p \tag{36}$$

Furthermore, Theorem 5 informs that the number of partition is bonded by N_{rep}. Therefore, (36) suggests that by bounding the local Lipschitz constant within each linear region, one could control the global Lipschitz constant of the neural network. Similar observation holds for E-D CNNs with skipped connection.

One of the most important implications of (36) is that the expressiveness of the network is not affected by the control of the Lipschitz constant. This in turn is due to the combinatorial nature of the ReLU nonlinearities, which allows for an exponentially large number of linear representations.

4.5. Optimization landscape

For a given ground truth *task map* $f^* : \boldsymbol{\mathcal{X}} \mapsto \boldsymbol{\mathcal{Y}}$ and given training data set $\{(x^{(i)}, y^{(i)})\}_{i=1}^T$ such that $y^{(i)} = f^*(x^{(i)})$, an encoder-decoder CNN training problem can be formulated to find a neural network parameter weight $\mathbf{W}$ by minimizing a specific loss function. Then, for the case of l_2 loss:

$$C(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{T} \|F(\mathbf{W}, x^{(i)}) - y^{(i)}\|^2 \,, \tag{37}$$

Nguyen et al (Nguyen & Hein, 2018) showed that over-parameterized CNNs can produce zero training errors. Their results are based on the following key lemma.

Lemma 7. *(Nguyen & Hein, 2018) Consider an encoder-decoder CNN without skipped connection. Then, the Jacobian of the cost function in (37) with respect to E^κ is bounded as*

$$\|\nabla_{E^\kappa} C\|_F$$

$$\geq \sigma_{\min}(\Xi^\kappa) \min_{i \in [T]} \sigma_{\min}\left(\Lambda^\kappa(x^{(i)})\left(\tilde{\Upsilon}^\kappa(x^{(i)})\right)^\top\right)\sqrt{2C(\mathbf{W})}$$

and

$$\|\nabla_{E^\kappa} C\|_F$$

$$\leq \sigma_{\max}(\Xi^\kappa) \max_{i \in [T]} \sigma_{\max}\left(\Lambda^\kappa(x^{(i)})\left(\tilde{\Upsilon}^\kappa(x^{(i)})\right)^\top\right)\sqrt{2C(\mathbf{W})}$$

where $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ denote the minimum and maximum singular value for a matrix $A \in \mathbb{R}^{n \times m}$ with $n \geq m$, respectively; $\tilde{\Upsilon}^\kappa$ is defined in (21), and Ξ^κ denotes the feature matrix for the training data

$$\Xi^\kappa = \begin{bmatrix} \xi^{\kappa(1)} & \cdots & \xi^{\kappa(T)} \end{bmatrix} \quad \in \mathbb{R}^{d_\kappa \times T}$$

and $C(\mathbf{W})$ is the cost in (37).

The authors in (Nguyen & Hein, 2018) further showed that if every shifted r-segment of training samples is not identical to each other and $d_\kappa \geq T$, then Ξ^κ has full column rank. Additionally, if the nonlinearity at the decoder layer is analytic, then they showed that $\tilde{\Upsilon}^\kappa(x)\Lambda^\kappa(x)$ has almost always full row rank. This implies that both $\sigma_{\min}(\Xi^\kappa)$ and $\sigma_{\min}(\Lambda^\kappa(\tilde{\Upsilon}^\kappa)^\top)$ are non-zero so that $\nabla_{E^\kappa} C|_{\mathbf{W}} = 0$ if and only if $y^{(i)} = F(\mathbf{W}, x^{(i)})$ for all $i \in [T]$ (that is, the loss becomes zero, i.e. $C(\mathbf{W}) = 0$).

Unfortunately, this almost always guarantee cannot be used for the ReLU nonlinearities at the decoder layers, since the ReLU nonlinearity is not analytic. In this paper, we extend the result of (Nguyen & Hein, 2018) for the encoder-decoder CNN with skipped connection when ReLU nonlinearities are used. In addition to Lemma 7, the following lemma, which is original, does hold for this case.

Lemma 8. *Consider an encoder-decoder CNN with skipped connection. Then, the Jacobian of the cost function in (37) with respect to $\tilde{S}^l$ for $l \in [\kappa]$ is bounded as*

$$\|\nabla_{\tilde{S}^l} C\|_F$$

$$\geq \sigma_{\min}(\Gamma^l) \min_{i\in[T]} \sigma_{\min}\left(\tilde{\Lambda}^l(x^{(i)})\left(\tilde{\Upsilon}^{l-1}(x^{(i)})\right)^\top\right)\sqrt{2C(\mathbf{W})}$$

and

$$\|\nabla_{\tilde{S}^l} C\|_F$$

$$\leq \sigma_{\max}(\Gamma^l) \max_{i\in[T]} \sigma_{\max}\left(\tilde{\Lambda}^l(x^{(i)})\left(\tilde{\Upsilon}^{l-1}(x^{(i)})\right)^\top\right)\sqrt{2C(\mathbf{W})}$$

where Γ^l denotes the feature matrix from the skipped branch

$$\Gamma^l = \begin{bmatrix}\chi^{l(1)} & \cdots & \chi^{l(T)}\end{bmatrix} \quad \in \mathbb{R}^{s_l \times T}$$

and $C(\mathbf{W})$ is the cost in (37).

Lemma 8 leads to the following key results on the optimization landscape for the encoder-decoder network with skipped connections.

Theorem 9. *Suppose that there exists a layer $l \in [\kappa]$ such that*

- *skipped features $\chi^{l(1)}, \cdots, \chi^{l(T)}$ are linear independent.*
- *$\tilde{\Upsilon}^{l-1}(x)\tilde{\Lambda}^l(x)$ has full row rank for all training data $x \in [x^{(1)}, \cdots, x^{(T)}]$.*

Then, $\nabla_{\tilde{S}^l} C|_{\mathbf{W}} = 0$ if and only if $y^{(i)} = F(\mathbf{W}, x^{(i)})$ for all $i \in [T]$ (that is, the loss becomes zero, i.e. $C(\mathbf{W}) = 0$).

Proof. Under the assumptions, both $\sigma_{\min}(\Gamma^l)$ and $\sigma_{\min}(\tilde{\Lambda}^l(\tilde{\Upsilon}^{l-1})^\top)$ are non-zero. Therefore, Lemma 8 leads to the conclusion. □

Note that the proof for the full column rank condition for Ξ^κ in (Nguyen & Hein, 2018) is based on the constructive proof using independency of intermediate features $\chi^{l(1)}, \cdots, \chi^{l(T)}$ for all $l \in [\kappa]$. Furthermore, for the case of ReLU nonlinearities, even when $\tilde{\Upsilon}^\kappa(x)\Lambda^\kappa(x)$ does not have full row rank, there are chances that $\tilde{\Upsilon}^{l-1}(x)\tilde{\Lambda}^l(x)$ has full row rank at least one $l \in [\kappa]$. Therefore, our result has more relaxed assumptions than the optimization landscape results in (Nguyen & Hein, 2018) that relies on Lemma 7. This again confirms the advantages of the skipped connection in encoder-decoder networks.

5. Discussion and Conclusion

In this paper, we investigate the geometry of encoder-decoder CNN from various theoretical aspects such as differential topological view, expressiveness, generalization capability and optimization landscape. The analysis was feasible thanks to the explicit construction of encoder-decoder CNNs using the deep convolutional framelet expansions. Our analysis showed that the advantages of the encoder-decoder CNNs comes from the expressiveness of the encoder and decoder layers, which are originated from the combinatorial nature of ReLU for decomposition and reconstruction frame basis selection. Moreover, the expressiveness of the network is not affected by controlling Lipschitz constant to improve the generalization capability of the network. In addition, we showed that the optimization landscape can be enhanced by the skipped connection.

This analysis coincides with our empirical verification using deep neural networks for various inverse problems. For example, in a recent work of k-space deep learning (Han & Ye, 2018), we showed that a neural network for compressed sensing MRI can be more effectively designed in the k-space domain, since the frame representation is more concise in the Fourier domain. Similar observation was made in sub-sampled ultrasound (US) imaging (Yoon et al., 2018), where we show that the frame representation in raw data domain is more effective in US so that the deep network is designed in the raw-date domain rather than image domain. These empirical examples clearly showed that the unified view between signal processing and machine learning as suggested in this paper can help to improve design and understanding of deep models.

Acknowledgements

The authors thank to reviewers who gave useful comments. This work was supported by the National Research Foundation (NRF) of Korea grant NRF-2016R1A2B3008104 and KI Science Technology Leading Primary Research.

References

Anthony, M. and Bartlett, P. L. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

Bartlett, P. L. and Mendelson, S. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-

normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. SGD learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. *Journal of Machine Learning Research*, 38:192–204, 2015.

Cohen, G., Giryes, R., and Sapiro, G. DNN or k-NN: That is the generalize vs. memorize question. *arXiv preprint arXiv:1805.06822*, 2018.

Du, S. S. and Lee, J. D. On the power of over-parametrization in neural networks with quadratic activation. *arXiv preprint arXiv:1803.01206*, 2018.

Du, S. S., Lee, J. D., Tian, Y., Póczos, B., and Singh, A. Gradient descent learns one-hidden-layer CNN: don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*, 2017.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Ge, R. and Ma, T. On the optimization landscape of tensor decompositions. In *Advances in Neural Information Processing Systems*, pp. 3656–3666, 2017.

Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., and Knoll, F. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

Han, Y. and Ye, J. k-space deep learning for accelerated MRI. *arXiv 2018(1805.03779)*, 2018.

Hanin, B. and Sellke, M. Approximating continuous functions by ReLU nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26 (9):4509–4522, 2017.

Kang, E., Min, J., and Ye, J. C. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Medical Physics*, 44(10):e360–e375, 2017.

Kawaguchi, K. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pp. 586–594, 2016.

Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1646–1654, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. *arXiv preprint arXiv:1704.08045*, 2017.

Nguyen, Q. and Hein, M. Optimization landscape and expressivity of deep CNNs. In *International Conference on Machine Learning*, pp. 3727–3736, 2018.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, pp. 2847–2854, 2017.

Rolnick, D. and Tegmark, M. The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*, 2017.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.

Schmidhuber, J. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

Shen, H. A differential topological view of challenges in learning with feedforward neural networks. *arXiv preprint arXiv:1811.10304*, 2018.

Telgarsky, M. Benefits of depth in neural networks. In *Conference on Learning Theory*, pp. 1517–1539, June 2016.

Tu, L. W. *An introduction to manifolds*. Springer,, 2011.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. On the margin theory of feedforward neural networks. *arXiv preprint arXiv:1810.05369*, 2018.

Whitney, H. Differentiable manifolds. *Annals of Mathematics*, pp. 645–680, 1936.

Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

Ye, J. C., Han, Y., and Cha, E. Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM Journal on Imaging Sciences*, 11(2):991–1048, 2018.

Yin, R., Gao, T., Lu, Y. M., and Daubechies, I. A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets. *SIAM Journal on Imaging Sciences*, 10(2):711–750, 2017.

Yoon, Y. H., Khan, S., Huh, J., Ye, J. C., et al. Efficient b-mode ultrasound image reconstruction from sub-sampled rf data using deep learning. *IEEE transactions on medical imaging*, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 2017.