On Deep Learning for Inverse Problems

Jaweria Amjad
Electronics & Electrical Engineering Dept
University College London
London, UK
jaweria.amjad.16@ucl.ac.uk

Jure Sokolić
Biomedical Engineering Dept
King's College London
London, UK
jure.sokolic@kcl.ac.uk

Miguel R.D. Rodrigues
Electronics & Electrical Engineering Dept
University College London
London, UK
m.rodrigues@ucl.ac.uk

***Abstract*—This paper analyses the generalization behaviour of a deep neural networks with a focus on their use in inverse problems. In particular, by leveraging the robustness framework by Xu and Mannor, we provide deep neural network based regression generalization bounds that are also specialized to sparse approximation problems. The proposed bounds show that the sparse approximation performance of deep neural networks can be potentially superior to that of classical sparse reconstruction algorithms, with reconstruction errors limited only by the noise level independently of the underlying data.**

I. Introduction

A large number of phenomena arising in science and engineering – including problems in medical imaging, remote sensing, chemometrics, and more – can be approximated using the linear observation model given by:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e} \tag{1}$$

where $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^{N_y}$ corresponds to a vector of observations, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{N_x}$ corresponds to a vector of underlying causes, $\mathbf{e} \in \mathbb{R}^{N_y}$ is a vector modelling noise or other perturbations, and $\mathbf{A} \in \mathbb{R}^{N_y \times N_x}$ is a usually known linear operator modelling the relationship between the observations and the causes.

A very common problem – known as inverse problem – then involves inferring the vector $\mathbf{x}$ from the vector $\mathbf{y}$ given knowledge of the linear operator $\mathbf{A}$. However, for $N_y < N_x$, this problem is severely ill-posed so – without resorting to additional assumptions – a unique solution does not exist.

A number of approaches to solve inverse problems have therefore been proposed over the past years leveraging the fact that many phenomena in nature admit some form of structure – such as sparsity, group sparsity, manifold structures, and more – that is key to restrict the space of possible solutions. In particular, the use of sparsity – exploiting the fact that the vector to be inferred from observations admits a sparse representation in some basis or frame – has led to a number of methods to approximate the solution of a linear inverse problem using greedy algorithms or convex optimization based algorithms [1]. For example, under the assumption that the desired vector contains at most $k \ll N_x$ non-zero entries, the well-known Basis Pursuit Denoise (BPDN) algorithm delivers an estimate of the desired vector $\mathbf{x}$ from the observation vector $\mathbf{y}$ given knowledge of the linear operator $\mathbf{A}$ as follows [1]:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq k \tag{2}$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ are the ℓ_2 and ℓ_1 norms of a vector. Moreover, the BPDN estimate of the desired vector can also be shown to approximate very well the true vector provided that the linear operator $\mathbf{A}$ obeys various conditions [2]. Other state-of-the-art approaches exploiting sparsity to solve this class of linear inverse problems – such as iteratively reweighted least squares and iterative soft-thresholding methods – are reported in [3]. However, these various approaches often require the linear operator to satisfy certain conditions to guarantee exact inference (in the absence of noise) or stable inference (in the presence of noise) of the desired vector from the observation vector [1], failing drastically otherwise.

Another class of approaches to solve linear inverse problems has also recently emerged in view of advances in deep learning. In particular, the use of deep learning approaches to solve inverse problems involves two phases: (i) in the *training phase*, a number of pairs of training vectors $\mathbf{x}$ and $\mathbf{y}$ corresponding to one another are used to tune the set of parameters of a deep neural network (DNN) architecture in order to implement a mapping from $\mathbf{y}$ to $\mathbf{x}$; [1] (ii) in the *testing phase*, a test vector $\mathbf{y}$ is mapped onto the vector $\mathbf{x}$ via the network. Interestingly, this procedure has been shown to perform exceedingly well in a wide variety of inverse problems such as compressed sensing, image denoising, image deblurring, image super-resolution, and many more [4], [5]. However, a justification for such outstanding performance is currently unknown, because recent frameworks attempting to provide a rationale for the efficacy of DNNs primarily focus on classification tasks rather than the regression tasks arising in inverse problems [6].

This paper – which aims to fill-in this gap – is motivated by two overarching questions:

- *How can we quantify the performance of DNN approaches in solving inverse problems?*
- *How does the performance of DNN approaches compare to the performance of other classical approaches for solving inverse problems?*

In particular, in our attempt to answer these questions, we build upon the robustness framework introduced by Xu and Mannor in [7]: (i) we then introduce new DNN based regres-

[1] Note that the operational principle associated with deep learning networks is different from that of classical approaches. Classical approaches to solve inverse problems attempt to directly invert the mapping from $\mathbf{x}$ to $\mathbf{y}$. In contrast, deep learning approaches attempt to learn a mapping from $\mathbf{y}$ to $\mathbf{x}$.

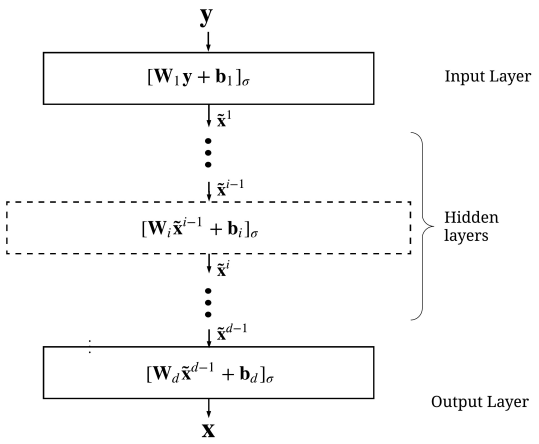

Fig. 1. A d-layer deep neural network.

sion generalization bounds; (ii) we show how these bounds can be used to quantify the performance of DNNs in solving a particular inverse problem involving sparse approximation; and (iii) we also show how the performance of a DNN compares with the performance of other classical approaches, notably BPDN, for solving such sparse approximation problems.

The remainder of the paper is organized as follows: We start by introducing our problem set-up in Section II. We then provide DNN generalization bounds applicable to general regression problems in Section III. We also provide specializations of these generalization bounds applicable to typical inverse problems in Section IV. This opens up the possibility of comparing DNN based approaches to classical approaches to solving inverse problems. Finally, concluding remarks are drawn in Section V.

Due to space limitations, the proofs are appearing in an upcoming preprint [8]

II. Setup

We concentrate on a supervised learning setup. In particular, we consider the problem of estimating a vector $\mathbf{x} \in \mathcal{X}$ from another vector $\mathbf{y} \in \mathcal{Y}$, where the pair of vectors $\mathbf{s} = (\mathbf{x}, \mathbf{y})$ is drawn from the sample space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ according to some unknown distribution μ. We also consider we have access to a set of m training samples $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i \leq m}$, drawn independently and identically distributed (i.i.d.) according to μ. These training samples are used to learn a regressor

$$\Xi_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X} \tag{3}$$

that can be used to deliver an estimate of the vector $\mathbf{x} \in \mathcal{X}$ given the vector $\mathbf{y} \in \mathcal{Y}$. We will be assuming for technical reasons that the input space $\mathcal{X}$ and the output space $\mathcal{Y}$ are compact with respect to the ℓ_2-metric and that the sample space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ is compact with respect to the sup-metric.

We are interested in characterising the quality of this learnt regressor, by capturing the deviation between the regressor estimate of the desired vector and the actual desired vector. In particular, this will be done via the *generalization error* (GE)

TABLE I
A List of Point-Wise Activation Functions $[\mathbf{z}]_\sigma = \{\sigma(z_i)\}_{i \leq N_i}$.

Name	Function $\sigma(z_i)$
Hyperbolic tangent	$\tan(z_i)$
ReLU	$\max(z_i, 0)$
Sigmoid	$1/1+\exp(-z_i)$
Softmax	$\exp(z_i)/\sum_j \exp(z_j)$

associated with the regressor given by:

$$GE(\Xi_{\mathcal{S}}) = |l_{\text{exp}}(\Xi_{\mathcal{S}}) - l_{\text{emp}}(\Xi_{\mathcal{S}})| \tag{4}$$

corresponding to the difference between the expected loss and empirical losses given by:

$$l_{\text{exp}}(\Xi_{\mathcal{S}}) = \mathbb{E}[l(\Xi_{\mathcal{S}}(\mathbf{y}), \mathbf{x})]$$

$$l_{\text{emp}}(\Xi_{\mathcal{S}}) = \frac{1}{m}\sum_i l(\Xi_{\mathcal{S}}(\mathbf{y}_i), \mathbf{x}_i)$$

where the loss function $l(\cdot, \cdot)$ is taken to be the ℓ_2-loss.

We concentrate exclusively on deep neural networks based regression. A deep neural network is a multi-layered architecture consisting of a series of linear and non-linear transformations as shown in Fig. 1 [9]. In particular, we can express the i-th layer output $\tilde{\mathbf{x}}^i \in \mathbb{R}^{N_i}$ in terms of the i-th layer input $\tilde{\mathbf{x}}^{i-1} \in \mathbb{R}^{N_{i-1}}$ as follows:

$$\tilde{\mathbf{x}}^i = \left[\mathbf{W}_i\tilde{\mathbf{x}}^{i-1} + \mathbf{b}_i\right]_\sigma$$

where $\mathbf{W}_i \in \mathbb{R}^{N_i \times N_{i-1}}$ is the i-th layer weight matrix, $\mathbf{b}_i \in \mathbb{R}^{N_i}$ is the i-th layer bias vector, and $[.]_\sigma$ represents an element-wise nonlinear activation function such as hyperbolic tangent, rectified linear units (ReLU), or sigmoid (see Table I). We also denote the network input by $\tilde{\mathbf{x}}^1 = \mathbf{y}$ and we denote the network output by $\Xi_{\mathcal{S}}(\mathbf{y}) = \tilde{\mathbf{x}}$.

The various hyper-parameters associated with a deep neural network can be learnt using optimization techniques based on training data [10]. State-of-the-art approaches include [11], [12].

In view of the fact that the generalization ability of deep neural network regressors is poorly understood, our goal in the sequel is to provide general generalization bounds for DNN based regression – applicable to a wide range of settings – as well as specialized generalization bounds for DNN based regression applicable to inverse problems.

III. Generalization Error Bounds: General Case

We now derive performance guarantees for DNN based regression by capitalizing on the robustness framework [7].

A very important element of the robustness framework is the notion of algorithmic robustness.

Definition 1 (*Algorithmic Robustness* [7])**.** Let $\mathcal{S}$ denote the training set and $\mathcal{D}$ denote the sample space. A learning algorithm is said to be $(K, \epsilon(\mathcal{S}))$-robust if the sample space

$\mathcal{D}$ can be partitioned into K disjoint sets $\mathcal{K}_k$, $k = 1, \ldots, K$, such that for all $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}$ and all $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$

$$\begin{aligned} (\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}, \mathbf{y}) \in \mathcal{K}_k \implies \\ |l(\boldsymbol{\Xi}_{\mathcal{S}}(\mathbf{y}_i), \mathbf{x}_i) - l(\boldsymbol{\Xi}_{\mathcal{S}}(\mathbf{y}), \mathbf{x})| \leq \epsilon(\mathcal{S}) \end{aligned} \tag{5}$$

In other words, a learning algorithm is robust provided that the losses of a training sample and a test sample belonging to the same partition are close.

The relevance of this definition is associated with the fact that it provides a route to study the generalization ability of various learning algorithms, including deep neural networks [6]. However, Sokolic et al. [6] have provided generalization bounds for DNN based classifiers in *lieu* of DNN based regressors, so the results cannot be used to cast insight on the performance of deep neural networks in solving inverse problems. We will therefore generalize the results in [6] from the classification to the regression setting.

We first show that a d-layer DNN based regressor satisfies a Lipschitz continuity condition.

Theorem 1. *(Adapted from Theorem 2 and Lemma 1 in [6]) Consider a d-layer DNN based regressor $\boldsymbol{\Xi}_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$. Then, for any $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{Y}$, it follows that*

$$\|\boldsymbol{\Xi}_{\mathcal{S}}(\mathbf{y}_1) - \boldsymbol{\Xi}_{\mathcal{S}}(\mathbf{y}_2)\|_2 \leq \prod_{i=1}^{d} \|\mathbf{W}_i\|_F \|\mathbf{y}_1 - \mathbf{y}_2\|_2$$

where $\|.\|_F$ denotes the Frobenious norm of a matrix.

Proof: We only outline the proof. The results follows from Theorem 2 in [6] which proves that the ratio between the Euclidean distance at the input and the output of a d-layer DNN is less than the ℓ_2-norm of the Jacobian matrix which is upper bounded by the product of the frobenious norm of the weight matrices [6]. ∎

We can now show the main results. The following theorem characterizes the robustness of a d-layer neural network in terms of the covering number of the metric space $(\mathcal{D}, \rho)$ [13].

Theorem 2. *(Robustness) Consider that $\mathcal{X}$ and $\mathcal{Y}$ are compact spaces with respect to the ℓ_2 metric. Consider also the sample space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ equipped with a sup metric ρ. It follows that a d-layer DNN based regressor $\boldsymbol{\Xi}_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ trained on the training set $\mathcal{S}$ is*

$$\left(\mathcal{N}\left(\frac{\psi}{2}; \mathcal{D}, \rho\right), \left(1 + \prod_{i=1}^{d} \|\mathbf{W}_i\|_F\right) \psi\right) - robust$$

for any $\psi > 0$, where $\mathcal{N}\left(\frac{\psi}{2}; \mathcal{D}, \rho\right) < \infty$ represents the covering number of the metric space $(\mathcal{D}, \rho)$ using metric balls of radius $\psi/2$.

Proof: We provide a sketch of the proof only. A full version will appear in an upcoming manuscript [8]. The loss function of a Lipschitz continuous DNN can be Lipschitz continuous too. Thus the difference of the losses between two sample points is upper bounded by the product of the Lipschitz constant $(1+\prod_{i=1}^{d} \|\mathbf{W}_i\|_F)$ and distance ψ, between the samples. The claim then follows. ∎

The following theorem – building upon the previous one – now characterizes a bound to the generalization error of a d-layer neural network.

Theorem 3. *(GE Bound) Consider again that $\mathcal{X}$ and $\mathcal{Y}$ are compact spaces with respect to the ℓ_2 metric. Consider also the sample space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ equipped with a sup metric ρ. It follows that a d-layer DNN based regressor $\boldsymbol{\Xi}_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ trained on a training set $\mathcal{S}$ consisting of m i.i.d. training samples obeys with probability $1 - \zeta$, for any $\zeta > 0$, the generalization error bound given by:*

$$\begin{aligned} GE(\boldsymbol{\Xi}_{\mathcal{S}}) \leq \left(1 + \prod_{i=1}^{d} \|\mathbf{W}_i\|_F\right) \psi \\ + M(\mathcal{S}) \sqrt{\frac{2\mathcal{N}\left(\frac{\psi}{2}; \mathcal{D}, \rho\right) \log(2) + 2\log\left(\frac{1}{\zeta}\right)}{m}} \end{aligned} \tag{6}$$

for any $\psi > 0$, where $M(\mathcal{S}) < \infty$ is a constant that depends only on $\mathcal{S}$.

Proof: This result follows from the generalization error bound provided in [7]. ∎

Theorems 2 and 3 provide various insights that are also aligned with previous results in the literature. In particular, these theorems suggest that the robustness and generalization properties of a d-layer neural network are not associated with the number of network parameters per layer but rather with appropriate norms of the weight matrices. Bartlett [14] had also shown the size of the network has no effect on the generalization error of a neural network by bounding the fat shattering dimension as a function of the ℓ_1 norm of the weights, so implying independence of the number of hidden units. Xu and Mannor [7] have also shown that robustness of a neural network does not depend on its size. Similarly, in [15], it is argued that norm based regularization can improve the generalization ability of a deep neural network.

These theorems also suggest that a deeper network may generalize better than a shallower one, by guaranteeing the Forbenius norm of the weight matrices is less than one. This result is aligned with similar claims by Neyshabur[15] resulting from matrix factorization approaches. In fact, it is possible to explicitly bound the norm of weight matrices via reprojection using gradient decent [10], and regularization of weight matrices has been empirically shown to result in better generalization [16].

Finally, Theorem 3 also suggests that – beyond the dependence on the number of training samples – the generalization ability of a d-layer neural network also depends directly on the complexity of the data space $\mathcal{D}$ captured via its covering number. In particular, the generalization error of more complex

data spaces will tend to be higher than the generalization error of a simpler data space.

IV. Generalization Error Bounds for Inverse Problems

We now specialize the performance guarantees from general regression problems to inverse problems, with a focus on sparse approximation tasks.

We consider specifically the linear observation model in (1), with some additional assumptions:

- First, the space $\mathcal{X}$ consists of k-sparse vectors, taken from an ℓ_2 ball of unit radius i.e.

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^{N_x} : \|\mathbf{x}\|_0 \leq k, \|\mathbf{x}\|_2 \leq 1\} \tag{7}$$

- Second, the space $\mathcal{Y}$ consists to a linear projection of the input space induced by observation matrix plus a perturbation associated with bounded ℓ_2-norm noise, i.e.

$$\mathcal{Y} := \{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e} \in \mathbb{R}^{N_y} : \mathbf{x} \in \mathcal{X}, \|\mathbf{e}\|_2 \leq \eta\} \tag{8}$$

- Third, we assume that the linear mapping represented by the matrix $\mathbf{A}$ is Lipschitz continuous with Lipschitz contant L, i.e.

$$\|\mathbf{A}\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2\|_2 \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{9}$$

  for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$. Note that this condition is in practice obeyed by linear mappings that conform to the Restriced Isometry Property (RIP) [17].

We also consider that an appropriately trained d-layer network – using a training set $\mathcal{S}$ – is employed to deliver an estimate of the sparse vector $\mathbf{x}$ given the measurement vector $\mathbf{y}$.

We can now immediately specialize the results appearing in Theorems 2 and 3 to this particular setting. The following upper bound on the covering number of the input space will be very useful [18]:

$$\mathcal{N}(\delta/2; \mathcal{X}, \|.\|_2) \leq \left(\frac{N_x e}{k}\right)^k \left(1 + \frac{4}{\delta}\right)^k \tag{10}$$

Corollary 1. *Consider the spaces $\mathcal{X}$ and $\mathcal{Y}$ in* (7) *and* (8) *equipped with a ℓ_2 metric, the space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ equipped with the sup-metric ρ, and the Lipschitz continuous mapping in* (9)*. It follows that a d-layer DNN based regressor $\mathbf{\Xi}_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ trained on the training set $\mathcal{S}$ is*

$$\left(\left(\frac{N_x e}{k}\right)^k \left(1 + \frac{4}{\delta}\right)^k, \left(1 + \prod_{i=1}^{d} \|\mathbf{W}_i\|_F\right)(L\delta + 2\eta)\right)$$

robust.

Sketch of Proof: For the model given by eqs. (7), (8) and (9), the $(L\delta+2\eta)/2$-covering number of metric space $(\mathcal{D}, \rho)$ is upper bounded by the $\delta/2$-covering number of $\mathcal{X}$. This result together with Theorem 2 proves the corollary. ∎

Corollary 2. *Consider again the spaces $\mathcal{X}$ and $\mathcal{Y}$ in* (7) *and* (8) *equipped with a ℓ_2 metric, the space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ equipped with the sup-metric ρ, and the Lipschitz continuous mapping in* (9)*. It follows that a d-layer DNN based regressor $\mathbf{\Xi}_{\mathcal{S}}(\cdot) : \mathcal{Y} \to \mathcal{X}$ trained on a training set $\mathcal{S}$ consisting of m i.i.d. training samples obeys with probability $1 - \zeta$, for any $\zeta > 0$, the generalization error bound given by:*

$$GE(\mathbf{\Xi}_{\mathcal{S}}) \leq \left(1 + \prod_{i=1}^{d} \|\mathbf{W}_i\|_F\right)(L\delta + 2\eta) + M(\mathcal{S})\sqrt{\frac{2\left(\frac{N_x e}{k}\right)^k \left(1 + \frac{4}{\delta}\right)^k \log(2) + 2\log\left(\frac{1}{\zeta}\right)}{m}} \tag{11}$$

for any $\delta > 0$, for some $M(\mathcal{S}) < \infty$.

Proof: The result follows directly from Theorem 3 and Corollary 1. ∎

The results embodied in these two corollaries can be used to illuminate further the performance of sparse approximation based on deep learning networks. In particular, let us assume we employ a regularization strategy during the training phase constraining the Frobenius norm of the weight matrices to be less than one, such as reprojection using gradient descent [10].

This leads immediately to another generalization error bound holding with probability $1 - \zeta$

$$GE(\mathbf{\Xi}_{\mathcal{S}}) \leq 2(L\delta + 2\eta) + M(\mathcal{S})\sqrt{\frac{2\left(\frac{N_x e}{k}\right)^k \left(1 + \frac{4}{\delta}\right)^k \log(2) + 2\log\left(\frac{1}{\zeta}\right)}{m}} \tag{12}$$

for any $\zeta > 0$ and any $\delta > 0$, and by setting $\delta = o\left(m^{-\frac{1}{k}}\right)$ and by setting trivially ζ to be a function of m such that $\log(1/\zeta)/m = o(1)$, to another generalization bound behaving as follows

$$GE(\mathbf{\Xi}_{\mathcal{S}}) \leq 4 \cdot \eta + o(1) \tag{13}$$

This suggests that – with the increase of the number of training samples m – the generalization ability of a deep neural network is limited only by the level of the noise independently of the parameter values of the linear observation model, namely N_y, N_x, k, and L. Instead, these parameters mainly influence the speed at which the generalization error asymptotics kick-in.

In turn, in view of the fact that the generalization error is upper bounded by the sum of the expected and empirical error, it is also possible to upper bound the expected sparse approximation error associated with a deep neural network as follows:

$$l_{\text{exp}}(\mathbf{\Xi}_{\mathcal{S}}) \leq l_{\text{emp}}(\mathbf{\Xi}_{\mathcal{S}}) + GE(\mathbf{\Xi}_{\mathcal{S}}) \leq l_{\text{emp}}(\mathbf{\Xi}_{\mathcal{S}}) + 4 \cdot \eta + o(1) \tag{14}$$

Recent results suggest that deep neural networks – with a sufficient number of parameters – tend to memorize the training dataset [19] suggesting that

$$l_{\exp}(\mathbf{\Xi}_{\mathcal{S}}) \leq GE(\mathbf{\Xi}_{\mathcal{S}}) \leq 4 \cdot \eta + o(1) \tag{15}$$

We conclude by comparing how the performance of a deep neural network compares to the performance of a well-known algorithm – BPDN – in sparse approximation problems.

Theorem 4 ([20])**.** *Consider the linear observation model in* (1) *where* $\mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^{N_x} : \|\mathbf{x}\|_0 \leq k\}$ *and* $\mathbf{y} \in \mathcal{Y} = \{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e} \in \mathbb{R}^{N_y} : \|\mathbf{x}\|_0 \leq k, \|\mathbf{e}\|_2 \leq \eta\}$*. Consider also the sparse approximation algorithm delivering an estimate of* $\mathbf{x}$ *from* $\mathbf{y}$ *given knowledge of* $\mathbf{A}$*:*

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^{N_x}} \|\mathbf{x}\|_1 \quad \textit{subject to,} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \varepsilon$$

where $\epsilon \geq \eta$*. It follows – under the assumption that* $k \leq (1+\mu)/4\mu$ *– the error of the approximation delivered by this algorithm can be bounded as follows:*

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq \frac{\eta + \varepsilon}{\sqrt{1 - \mu(4k-1)}}$$

where μ *corresponds to the mutual coherence of the matrix* $\mathbf{A}$*.*

This sparse approximation algorithm – along with other sparse approximation algorithms based on convex optimization approaches or greedy approaches (see [1] and references within) – are known to exhibit a phase transition. Here, when the data sparsity $k \leq (1+\mu)/4\mu$, the algorithm provides a reconstruction that scales with the amount of noise η; this is akin to the behaviour of the sparse approximation delivered by a deep neural network.

On the other hand, when the data sparsity $k > (1+\mu)/4\mu$ the algorithm does not give any reconstruction guarantees but the deep neural network may still be able to deliver an appropriate reconstruction of the sparse vector given its under-sampled linear observation. In fact, reference [21] has empirically demonstrated that the performance of a DNN degrades gradually as N_y decreases in relation to N_x and k.

V. Conclusions

This paper, by drawing on the robustness framework introduced by Xu and Mannor, puts forth a generalization bound for deep neural network based reconstruction that can be specialized for a wide range of settings.

The specialization of this bound to sparse approximation problems – occurring in various signal and image processing tasks – has shown that deep neural networks can lead to generalization errors that depend on the noise level only. This – together with the fact that recently established results suggest that deep neural networks can potentially memorize datasets – also suggests that the sparse approximation error incurred via the use of deep neural networks also depends on the noise level only. This behaviour can be in sharp contrast with the behaviour of classical sparse approximation algorithms.

Acknowledgements

This research is supported by the Commonwealth Scholarship Commission in UK.

References

[1] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[2] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, 2011.

[3] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*. IEEE, 2008, pp. 3869–3872.

[4] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging," *IEEE SIgnal ProcESSIng MagazInE*, vol. 1053, no. 5888/18, 2018.

[5] D. M. Nguyen, E. Tsiligianni, and N. Deligiannis, "Deep learning sparse ternary projections for compressed sensing of images," *arXiv preprint arXiv:1708.08311*, 2017.

[6] J. Sokolic, R. Giryes, G. Sapiro, and M. R. Rodrigues, "Robust large margin deep neural networks," *IEEE Transactions on Signal Processing*, 2017.

[7] H. Xu and S. Mannor, "Robustness and generalization," *Machine learning*, vol. 86, no. 3, pp. 391–423, 2012.

[8] J. Amjad, J. Sokolic, and M. R. D. Rodrigues, "On deep learning for inverse problems," *In Preparation*.

[9] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.

[10] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[12] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 1998, pp. 9–50.

[13] A. W. Van Der Vaart and J. A. Wellner, "Weak convergence," in *Weak Convergence and Empirical Processes*. Springer, 1996, pp. 16–28.

[14] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[15] B. Neyshabur, "Implicit regularization in deep learning," *arXiv preprint arXiv:1709.01953*, 2017.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes rendus mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[18] R. Giryes, G. Sapiro, and A. M. Bronstein, "Deep neural networks with random gaussian weights: a universal classification strategy?" *IEEE Trans. Signal Processing*, vol. 64, no. 13, pp. 3444–3457, 2016.

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

[20] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on information theory*, vol. 52, no. 1, pp. 6–18, 2006.

[21] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*. IEEE, 2015, pp. 1336–1343.