Signal Processing 191 (2022) 108361

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Subset based error recovery

Ömer Ekmekcioğlu, Deniz Akkaya, Mustafa Ç. Pınar*

Bilkent University Ankara 06800, Turkey

ARTICLE INFO

Article history:
Received 26 July 2021
Revised 9 October 2021
Accepted 10 October 2021
Available online 12 October 2021

Keywords:
Robust Networks
Extreme Learning Machine
Sparse Recovery
Regularization
Hard Thresholding

ABSTRACT

We propose a data denoising method using Extreme Learning Machine (ELM) structure which allows us to use Johnson-Lindenstrauß Lemma (JL) for preserving Restricted Isometry Property (RIP) in order to give theoretical guarantees for recovery. Furthermore, we show that the method is equivalent to a robust two-layer ELM that implicitly benefits from the proposed denoising algorithm. Current robust ELM methods in the literature involve well-studied L1, L2 regularization techniques as well as the usage of the robust loss functions such as Huber Loss. We extend the recent analysis on the Robust Regression literature to be effectively used in more general, non-linear settings and to be compatible with any ML algorithm such as Neural Networks (NN). These methods are useful under the scenario where the observations suffer from the effect of heavy noise. We extend the usage of ELM as a general data denoising method independent of the ML algorithm. Tests for denoising and regularized ELM methods are conducted on both synthetic and real data. Our method performs better than its competitors for most of the scenarios, and successfully eliminates most of the noise.

1. Introduction

Foundations of the ELM are rooted in function approximation theory. ELM uses a randomly generated network layer to obtain successful approximations on continuous functions [1]. This random layer is shown to be effective and efficient in terms of both accuracy and the computation complexity [2]. Randomly generated weights are not only used in ELM's but also in dimensionality reduction and compressed sensing due to their performance on accuracy/computation complexity trade-off. Johnson-Lindenstrauß Lemma allows ELMs to reduce the dimension of the problem for efficiency in computations while preserving the structure of the data [3,4]. The second layer introduces non-linear interactions of the features to improve the prediction capabilities of the system. At this point, using ELMs as a sparse error recovery and data denoising tool becomes highly efficient. Therefore, a robust ELM application along with an extendable data denoising method applicable to different machine learning frameworks (especially NN's) is proposed in this paper. In Section 2, we describe the contribution of our approach to the literature, and in Section 3 we briefly explain the requisite background information on the problem structure, ELM and the theorems used in the analysis. In Sections 4 and 5 we describe our algorithm and give theoretical results, respectively. In Section 6 we compare our algorithm with multiple ELM methods and show the effectiveness of data denoising with the comparison of multiple learning algorithms on both synthetic and real data.

2. Our contribution

Under the non-linear CS framework, recovery guarantees of the proposed denoising algorithm are analyzed. We shall use these results to provide the denoising algorithm for non-linear ML problems by extending the robust regression analysis [5,6] into a general denoising method applicable for neural networks, ELM and other ML algorithms. The motivation for applying such a denoising technique originates from the fact that sparse recovery methods are highly disturbed under heavy corruption. Denoising methods effective to address this issue are also expected to be effective in non-linear optimization problems. Furthermore, in light of the recent convex NN interpretations and following the studies on activation regions, we propose to use randomized activation regions to effectively evaluate the quality of the data points.

First, we describe how multiple layers of ELMs can be used to formulate sparse recovery problems for non-linear machine learning problems to denoise data from highly corruptive noise. Second, we provide a hard threshold based subset selection algorithm for an ELM application that outperforms robust loss functions and regularization methods [7], and derive convergence guarantees. One of the main contributions to the analysis performed on the convergence guarantees involves the JL Lemma and its relation with the

* Corresponding author.
E-mail addresses: omer.ekmekcioglu@bilkent.edu.tr (Ö. Ekmekcioğlu), deniz.akkaya@bilkent.edu.tr (D. Akkaya), mustafap@bilkent.edu.tr (M.Ç. Pınar).

https://doi.org/10.1016/j.sigpro.2021.108361
0165-1684/

RIP property which allows the former proofs to be still valid. To the best of our knowledge, only robust functions and their combinations with regularization methods were previously studied in the robust ELM literature. Therefore, a method providing robustness with a theoretical background is deemed a welcome and timely contribution to the literature.

3. Background

3.1. Sparse recovery

The Literature on sparse recovery mainly focuses on the following problem

$$\begin{array}{ll}\min & \|y - Xw\|_2^2 \\ \text{s.t.} & \|w\|_0 \leq k,\end{array}$$

where $X \in \mathbb{R}^{n\times p}$ represents the data matrix, $y \in \mathbb{R}^n$ contain the observations, and $w \in \mathbb{R}^p$ are the unknown coefficients to be estimated. The notation $\|w\|_0$ denotes the ℓ_0-norm of w which counts the number of non-zero elements of w. The number of non-zero elements (the cardinality of the support of w) is restricted to be at most k.

Due to the ℓ_0 constraint, the problem is NP-Hard [8,9]. In the literature, various solution techniques have been proposed, ranging from convex relaxations of the problem to heuristic algorithms to handle the cardinality constraint. Some of the previously proposed and prominent solution methods are Fista [10] and Iterative Hard Thresholding (IHT) [11].

In the present study, the IHT algorithm is used as one of the main building blocks of our algorithm. However, to clean the data from corrupting errors, the problem will be cast as selecting sparse observations from the data instead of finding a sparse regression solution.

3.2. Non-linear robust model description

The model of this paper is a non-linear one where the observations are heavily corrupted by a noise similar to those analysed in [5]:

$$y^* = \Phi(Xw^*) \tag{1}$$

$$y = y^* + b + \epsilon, \tag{2}$$

where X is a matrix of features, w^* is a vector of weights, $\Phi(\cdot)$ is a non-linear map, b denotes the corruptive noise in the observation due to the measurements, and ϵ denotes the regular Gaussian white noise. In the following sections, the model will be analyzed for the case $y = \hat{y} + b$ without the Gaussian white noise to be able to devise a simple yet efficient hard thresholding method. This relaxation allows the IHT approach to be viable during subset selection. In robust network literature, noise vector b is generally taken as a sparse vector such that $\|b\|_0 \leq 0.4n$ where n is the number of observations [7]. Due to this sparsity pattern in b which is induced from the ℓ_0-norm, one can reformulate the problem in the form of a compressed sensing problem [5]. Bhatia et al. [5] proves the convergence guarantees and compares performances for the subset based regression techniques TORRENT and ADACRR [12] using this sparse recovery reformulation to the robust regression methods. We shall follow a similar approach to analyze the recovery problem and its applications.

3.3. ELM Model description

Let $X = [x_1, x_2, \ldots, x_n]^T$ be a feature matrix of dimension $n \times p$, such that $x_i \in \mathbb{R}^p$. Let $y \in \mathbb{R}^n$ be the target vector for all n observations. Weight matrix $W_1 \in \mathbb{R}^{p\times l}$ represents the randomly generated layer and $w_2 \in \mathbb{R}^l$ is optimized in the second layer. The dimension of the first layer is denoted by l whereas ϕ denotes any transfer function such as ReLu, Leaky ReLU, tanh or sigmoid:

$$Z = \phi(XW_1) \tag{3}$$

$$\hat{y} = Zw_2. \tag{4}$$

In order to calculate the second layer weights, w_2, one can use various gradient descent algorithms in addition to the widely used ℓ_2 norm minimization formula. The widely used closed form solution of the second layer weight is shown in [1] as

$$w_2 = \begin{cases}(Z^TZ)^{-1}Z^Ty & n \geq l, \\ Z^T(ZZ^T)^{-1}y & n \leq l.\end{cases} \tag{5}$$

The above closed form expressions are derived from least squares minimization. Depending on the existence of the generalized inverse, one of the identities is used.

The transformation in the random layer is analogous to dimensionality reduction using random projections when $l < p$ and the related JL Lemma. This property will be useful to show that the data structure is preserved throughout the network regardless of the non-linear transforms.

In addition, there are algorithms involving Iterative Hard Thresholding in ELMs [13]. However, these algorithms are applied to the decision weights on the second layer to obtain sparse weights. The present paper is completely different in terms of the use of the iterative hard thresholding and sparsity sought in the variables. However, our theoretical study supports the foundation of the proposed algorithm in [13] implicitly where they lack theoretical results.

3.4. Robust methods and related loss functions

As an overview, one can summarize the most commonly used regularization methods in NN's as ℓ_1, ℓ_2 regularization and dropout. Regularization methods are studied in detail in many different areas including machine learning, compressed sensing and optimization.

Robust loss functions are selected from the functions which are less sensitive to the outliers to induce robustness in the system. Huber loss is one of the most commonly used robust loss functions in the literature. Intensive analysis on the function and its implementations for many ML studies are available, e.g., [14].

In the literature there exist methods to transform robust functions and regularization methods into a compact format to be used in robust networks [7]. These are efficient in terms of computation and implementation as the form of each loss can be written in terms of iteratively re-weighted least squares function. Furthermore, involved methods on parameter selection are known for online-sequential learning [15] with more specific implementation details that are not within the scope of this paper.

3.5. Convex neural networks

In very recent literature [16] the convexity of the two-layer Neural Networks is analyzed. Pilancı and Ergen [16] shows the equivalence of the classical two-layer relu neural network (3) to the following convex program

$$\begin{aligned}\min_{\{v_i,w_i\}_1^P} \frac{1}{2}\left\|\sum_{i=1}^P D_iX(v_i - w_i) - y\right\|_2 + \beta\sum_{i=1}^P(\|v_i\|_2 + \|w_i\|_2) \\ \text{s.t.} \quad (2D_i - I)Xv_i \geq 0 \quad \forall i \in \{1, ., P\} \\ (2D_i - I)Xw_i \geq 0, \quad \forall i \in \{1, ., P\}\end{aligned}$$

where the diagonal matrices D_i's correspond to a hyperplane arrangement and P is the number of all hyperplane arrangements. Since the analysis of the hyperplane arrangements and the equivalence of the two problems are out of the scope of this paper, we refer the readers to [16] for details. Furthermore, the original weights for the neural network can be obtained based on a result detailed in [16]. The most crucial point of this convex formulation is the hyperplane arrangements denoted by D_i in the formulation. This feature is introduced to aggregate the data with its small subsets so that the problem becomes a sparse recovery problem with the group sparsity regularization term.

There are robust Neural Network studies extending this approach in [17]. With this approach the robustness around a given perturbation ball can be implemented using convex optimization. However, from a practical viewpoint these implementations are not on a par with the classical neural networks, and the scaling performance to large data sizes is worse compared to that of neural networks.

The afore-mentioned convex approach is generally not practical but very insightful for theoretical analysis. We also find that our approach parallels those theoretical insights presented in [16] and [17].

4. Algorithm

The main reason for using ELM architecture in the data selection is to calculate the most important entries as fast as possible while capturing the possible non-linearities in the data. The preservation of the data after the random projections is a consequence of the JL Lemma:

Lemma 1 (JL). *Given $0 < \delta < 1$, a set X of n points in $\mathbb{R}^d$, and a number $k \geq \frac{3}{c\delta^2} \ln n$ for an appropriate positive constant c, there exists a random projection $f : \mathbb{R}^d \to \mathbb{R}^k$ which has the following property with probability at least $1 - \frac{3}{2}n$,*

$$\left| \left\| f(v_i) - f(v_j) \right\| - \sqrt{k} \left\| v_i - v_j \right\| \right| \leq \delta \sqrt{k} \left\| v_i - v_j \right\|$$

for all distinct pairs of points v_i and v_j in X.

Using this projection property, in a two-layer ELM, we can preserve the data structure in the first random layer, transform the data with a transfer function and create non-linearities in the second "calculated" layer which will be helpful to capture non-linearities. In the numerical tests in Section 6, the addition of multiple random layers is studied to analyze the effectiveness of the method in capturing highly dependent data structures.

Remark 1. The idea of random projections is similar to creating random activation patterns using randomized hyperplane arrangements in the convex neural network formulations. Using randomized activation patterns allow us to benefit from only a specific combination of fixed activation region from the data. Using this fixed activation region selected, we evaluate the performance of the data points. Finally, we select a useful subset of the data with respect to that evaluation.

The hard thresholding step is introduced to the robust regression literature in [6] and extensively studied in [5,12]. A similar idea can be extended to the proposed ELM architecture to obtain the best subset of the data which is not corrupted for non-linear setting:

$$\begin{array}{ll} \min_b & \|y - ZW_2 + b\|_2^2 \\ \text{s.t.} & Z = \phi(XW_1) \\ & \|b\|_0 \leq k \end{array}$$

The first constraint above varies in the problem formulation depending on the number of layers that will be used in the denoising algorithm. The number of layers is viewed as a hyper-parameter depending on the structure of the non-linearities within the data. The general denoising problem is formulated as follows

$$\begin{array}{ll} \min & \|y - ZW_n + b\|_2^2 \\ \text{s.t.} & Z = \phi(\ldots\phi(\phi(XW_1)W_2)\ldots W_{n-1}), \\ & \|b\|_0 \leq k. \end{array}$$

First, the algorithm considers a θ-layered neural network where the $\theta - 1$ hidden layers are fixed and randomly generated. The function ϕ is selected as Leaky ReLU for theoretical analysis. However, ReLU, Sigmoid, tanh or any other injective transfer function could be used. The second layer output is obtained using least squares loss. The weight calculation is performed under the assumption $n \geq l$, otherwise the generalized inverse should be used as explained in the background section. Furthermore, the proposed method will be used as a pre-processing method in most applications, therefore the layer dimension is kept smaller than the data dimension with the given bounds of JL-Lemma to make the system work as fast as possible while preserving RIP property [3].

Algorithm 1: Subset Based Error Recovery (SuBER).

Input: X
Result: $\hat{w_{2_t}}$
e : residual error
initialization:
$W_1 = \mathcal{N}(0, I)$;
$t = 0$;
k: hyperparameter for subset size;
compute first layer: $Z = \phi(XW_1)$;
$w_2 = (Z^TZ)^{-1}Z^Ty_{train}$;
while $t \leq$ *max iter* **do**
  calculate predictions: $\hat{y_t} = Zw_{2_t}$;
  select minimum k elements:
  $S_t := \text{mink}(\|y - \hat{y_t}\|)$**calculate** w_{2_t}**:** $w_{2_t} = (Z_{S_t}^T Z_{S_t})^{-1} Z_{S_t}^T y_{S_t}$;
end

Algorithm 1 relies on the idea that one could disregard the indices where the error is large using IHT. This can be interpreted as an IHT method applied on X^T instead of X after the w_2 weights are calculated using the closed-form solution of the least-squares regression. Iteratively calculating the final layer weights and the best subset of data points allows us to converge to a denoised subset of the data. A more detailed explanation is provided in Section 5.

In the algorithm, the hyperparameter for the subset size is required as a hyperparameter λ that would be used analogously in ℓ_1 or ℓ_2 regularization methods or the parameter γ that would be used in the Huber Loss. In addition, the number of random layers is adjusted as a hyperparameter.

The special case of the algorithm when $\theta = 2$ and $\phi =$ Leaky ReLU reduces the problem into a regression problem with a regular ELM architecture where the data subsets are selected dynamically. This special case is analyzed below as the ELM application and its performance on the existing ELM methods in the literature will be presented.

5. Theoretical analysis

In order to provide the convergence guarantees, we use an approach similar to the convergence proof of the Robust Regression algorithm [5]. First, we recall the following definitions in order to use the RIP results. We use $\mathcal{B}_0(k)$ to denote the "ball" consisting of k-sparse vectors.

Definition 1 (RSC and RSS Properties, [18]). *A matrix $X \in R^{n \times p}$ is said to satisfy the α restricted strong convexity (RSC) property and the*

β restricted smoothness (RSS) property of order k if for all $w \in \mathcal{B}_0(k)$, we have

$$\alpha_k \|w\|_2^2 \leq \frac{1}{n}\|Xw\|_2^2 \leq \beta_k \|w\|_2^2.$$

The definition above is a more stringent version of the definition used in similar settings. Definition 3 below implies the same properties for any subset $K \subseteq X$.

It is important to note that the first layer of random weights in this study is a matrix instead of a vector as it usually is in the Compressed Sensing framework. However, one can assume to have the collection of vectors w to form W_1 in the NN and ELM cases.

Definition 2 (NSC and NSS Properties)**.** *A non-linear transformation of a matrix $X \in R^{n\times p}$ is said to satisfy the α non-linear strong convexity (NSC) property and the β non-linear smoothness (NSS) property of order k if for all $w \in \mathcal{B}_0(k)$, we have*

$$\alpha_k \|w\|_2^2 \leq \frac{1}{n}\|\phi(Xw)\|_2^2 \leq \beta_k \|w\|_2^2.$$

It was shown in [19] that if the function ϕ is injective, the necessary and sufficient conditions for NSC and NSS properties are satisfied. Similar to the subset version of the RSC and RSS, NSC and NSS imply the same properties for the subsets $K \subseteq X$.

In this study, injective transfer functions such as Leaky ReLu, Sigmoid, tanh are used to satisfy this property. However, it is observed that the non-injective function ReLU performs well in practice.

Definition 3 (SSC, SSS, [5])**.** A matrix $X \in \mathbb{R}^{n\times p}$ is α strong convex and β strong smooth of order k for $S \subseteq \{1,\dots,n\}$ with $|S| \leq k$ iff

$$\alpha_k \leq \lambda_{\min}(X_S^T X_S) \leq \|X_S\|_2^2 \leq \lambda_{\max}(X_S^T X_S) \leq \beta_k,$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and the maximum eigenvalues for the given matrix and X_S is a matrix consisting rows of X corresponding to indices chosen from S.

Definition 4. For any $w \in \mathbb{R}^l$ and $c_0 > 0$ the random variable $\|XW_1 w\|_2^2$ is *strongly concentrated* about its expected value if

$$P\left(\left|\|XW_1 w\|_2^2 - \|w\|_2^2\right| \geq \epsilon \|w\|_2^2\right) \leq 2e^{-nc_0}$$

for $0 < \epsilon < 1$.

Lemma 2. *[20] ReLU and Leaky ReLU functions can be characterized as*

$$\phi(Xw) = D_w U \Sigma V^T w$$

where SVD of X is expressed as $X = U\Sigma V^T$ and D_w is a diagonal matrix with ReLU/Leaky ReLU coefficients on the diagonals.

For the Leaky ReLU activation function, the matrix D_w is invertible. This is not possible for ReLU when there are 0 entries on the diagonal.

Theorem 1 (L. ReLU Preserves SSC,SSS)**.** *Let ϕ be the Leaky ReLU function and assume $\|XW_1 w\|_2^2$ is strongly concentrated about its expected value. Then for all $w \in \mathcal{B}_0(k)$ and any $0 < \delta < 1$ we have*

$$\frac{1-\delta}{10}\|w\|_2 \leq \|Zw\|_2 \leq (1+\delta)\|w\|_2$$

with probability at least $1 - 2(12/\delta)^k e^{-c_0(\delta/2)n}$.

Proof. Since $\|XW_1 w\|_2^2$ is strongly concentrated we have

$$(1-\delta)^2\|w\|_2^2 \leq \|XW_1 w\|_2^2 \leq (1+\delta)^2\|w\|_2^2$$

for all $\delta \in (0,1)$ and $w \in \mathcal{B}_0(k)$ with the given probability [3]. Then, for the Leaky ReLU function ϕ, we have

$$\sigma_{\max}^2(D_w)\|XW_1 w\|_2^2 \geq \|D_w XW_1 w\|_2^2 \geq \sigma_{\min}^2(D_w)\|XW_1 w\|_2^2,$$

where $\sigma_{\max}^2$ and $\sigma_{\min}^2$ are the maximum and minimum singular values for a given matrix. Combining these two results, we obtain

$$(1-\delta)^2\sigma_{\min}^2(D_w)\|w\|_2^2 \leq \|Zw\|_2^2 \leq (1+\delta)^2\sigma_{\max}^2(D_w)\|w\|_2^2.$$

For any w, D_w is a diagonal matrix having entries 0.1 and1's. Thus one can find global upper and lower bounds as desired. □

Remark 2. For any piecewise linear transfer function with $\sigma_{\max} = \sigma_{\min} = 1$ at all pieces, SSC and SSS bounds are equivalent after the transformation

$$\alpha \leq \sigma_{\min}^2(XW_1) \leq \|\phi(XW_1)\|_2^2 \leq \sigma_{\max}^2(XW_1) \leq \beta.$$

We note that the above bound is equivalent to the bound in Definition 3.

Hard Thresholding Step: The reduced formulation without the Gaussian noise, i.e, $y = \hat{y} + b$, is used to transform the problem into a hard thresholding problem properly. The hard thresholding step consists of the following optimization problem

$$\text{(HTS)} \min_b \quad \left\|(I - Z(Z^TZ)^{-1}Z^T)(y+b)\right\|_2^2 \quad \text{s.t.} \quad \|b\|_0 \leq k,$$

where $Z = \phi(XW_1)$ or in the more convoluted form of multiple ϕ functions. After the forward propagation, the equivalence of the residuals and the b value can be seen by the definition that $y - \hat{y} = b$. As a result the formulation above is simply reduced to selecting the observation indices with the largest b values.

Convergence Guarantees: For the proof we will combine the following relations: (Exactly the same as in [5] using the nonlinear case Definition 2 instead of their Definition 1). We show that essentially the same convergence guarantees hold.

Theorem 2. *Let $Z \in \mathbb{R}^{n\times l}$ satisfy the SSC property at order k with parameter α_k and the SSS property at order $l-k$ with parameter β_{l-k} such that $\frac{\beta_{l-k}}{\alpha_k} < \frac{1}{1+\sqrt{2}}$. Let $W_2 \in \mathbb{R}^l$ be an arbitrary vector and $y = ZW_2 + b^*$ where $\|b^*\| \leq l-k$ is a sparse vector of possibly unbounded corruptions. Then Subset Based Regularization yields an ϵ-accurate solution $\|W_2 - W_{2_t}\| \leq \epsilon$.*

It is important to note that the convergence guarantee is exactly the same as the one required for the robust regression problem in [5]. In view of the proof provided in [5], we can use our Theorem 1 to obtain the convergence proof. Therefore, we have omitted the details.

Convergence Guarantees For Multiple Layers: The idea for wider networks follows a similar pattern using the previous result.

Remark 3. If we work with θ layers defining

$$Z = \phi(\phi(\dots\phi(XW_1)W_2)\dots W_\theta),$$

then we may apply Theorem 2 after assuming SSC and SSS properties for Z. Also, one can see that if we apply the steps in the proof of Theorem 1 we may obtain an SSS-SSC guarantee for such Z under mild conditions.

With each additional non-random layer, the minimum and the maximum singular values have an impact on the convergence guarantees on top of the structure of the original covariance matrix.

Remark 4. In [20], the analysis shows that the magnitude of the eigenvalues diminishes with each successive layer. This suggests that the proposed algorithm converges for θ-Layers with very high probability if the 2-Layer ELM convergence condition holds, which is similar to the condition of the convergence of a robust regression algorithm [5].

6. Results

In this section we first present the performance of our algorithm when it is used as a denoising tool. Second, we deploy our algorithm as a stand-alone ELM algorithm, and compare it with other robust ELM's architectures in the literature. In both sections, the synthetic data $X \in \mathbb{R}^{n\times p}$ where $n > p$ is generated similarly to the tests conducted in [5] and [7]. For corruptions, we set $\|b\|_0 = 0.2n$ and randomly apply corruption to randomly selected indices with the randomly selected magnitudes of $\pm 5\|y\|_\infty$. More specifically, initial tests were made on randomly generated observations $x_i \in \mathbb{R}^{1000}$ where $i \in \{1, \ldots, 2000\}$. The error size is selected as 400 and the entries are corrupted such that observation instances are selected at random and corrupted with additive corruption $b \sim Unif(-5\|y\|_\infty, 5\|y\|_\infty)$. The original outputs, y, are produced such that $y = Xw + b + \epsilon$ for linear case and $y = X^T Xw + Xw + b + \epsilon$, where $w \sim \mathcal{N}(0, 1)$ denotes the randomly generated weights. To be able to demonstrate the flexibility of the algorithm there is no additional sparsity pattern requirement enforced on the weight vector w in contrast to other studies. In the output function, Huber loss has been adopted for all of the models.

6.1. Data denoising

After the original data is generated, two-layer, three-layer and four-layer denoising methods are used to select the noiseless subset candidates to be used in the network. These models are trained and tested using Python Keras Library. Feed-forward networks with two hidden layers are used with neuron sizes equal to 64 in each layer for the results. Tables 1 and 2 show the performance of the denoising algorithm where the data had low non-linearity and high non-linearity, respectively.

The performance of the neural networks in Tables 1 and 2 indicates that our denoising algorithm introduces significant amount of robustness.

The performance of the multi-layer denoising does not appear to be affected by the number of layers in terms of the MSE. Synthetic data may not always be very suitable for deep learning, therefore the following tests were conducted on real data. *Boston Housing Prices* and *Diabetes* datasets are used for this purpose. The original dataset is corrupted using heavy noises as explained previously using the sparse noise vector $\|b\|_0 < 0.4n \approx 160$. In parallel with the previous tests, we take, $b \sim Unif(-5\|y\|_\infty, 5\|y\|_\infty)$.

In the tests, the models compared are benefiting from robust loss functions and regularization methods. The denoising method alone was able to surpass the competing methods. It was also observed that 2-Layer Denoise gives a better performance than the competitor robust methods. The differences in the layers create different initializations of the NN activation patterns. The results show that using the 2-Layer approach is also highly effective. Moreover, the same robust loss functions and regularization methods are applicable to denoised data theoretically, and better results could have been obtained if dropout was included in our algorithm tests. The main goal here is not to find the best possible fit for the real data, but to demonstrate the power of data-denoising even compared to relatively complex models. The results in Tables 3 and 4 and the MSE results in Tables 7 and 8 point out to similar outcomes obtained both from denoising and the proposed ELM algorithm.

Table 1
NN Result Part 1.

Denoising Results for Low Non-Linearity		
Data	Loss	MSE
Original	45.947	1141.648
2-Layer Denoise	**13.202**	**72.372**
3-Layer Denoise	13.765	**72.372**
4-Layer Denoise	13.558	78.429
Original+Dropout	30.166	543.952

Table 2
NN Results Part 2.

Denoising Results for High Non-linearity		
Data	Loss	MSE
Original	112.575	5211.718
2-Layer Denoise	93.261	3455.253
3-Layer Denoise	92.295	3455.25
4-Layer Denoise	**92.193**	**3349.651**
Original+Dropout	100.328	4090.479

Table 3
Boston Price Dataset Results.

Denoising Results for Boston Pricing Dataset		
Data	Loss	MSE
Original	5.779	66.788
2-Layer Denoise	4.195	12.838
3-Layer Denoise	4.199	12.838
4-Layer Denoise	**4.131**	**12.562**
Original+Dropout	10.667	64.376

Table 4
Diabetes Dataset Results.

Denoising Results for Diabetes Dataset		
Data	Loss	MSE
Original	38.335	126.748
2-Layer Denoise	37.239	103.539
3-Layer Denoise	37.236	103.539
4-Layer Denoise	**36.509**	**93.557**
Original+Huber+Dropout	36.609	112.05

Table 5
ELM Results.

ELM Results for Linear Case			
Methods	MSE	Rel. Err.	Corr. Rate
ELM	1.5788	2.5679	3.2650
SuBER	**0.2550**	**1.0323**	0.6491
ELM+$\ell 2$	1.5870	2.5746	3.2745
RP+Bisquare	0.2846	1.0912	**0.5666**
IRLS+Huber	0.3192	1.1547	0.9080

6.2. ELM Method

In this section, the goal is to show that the ELM inheriting the denoising method similar to the robust linear regression [5] approaches remains prevalent compared to similar methods in the literature. For the tests, layer sizes are selected equal to 500 in order to benefit from the fast denoising due to the JL Lemma. MSE and Relative error results are displayed in Table 5 where Relative error is defined as $\frac{\|y_{test} - \hat{y}\|_2}{\|y_{train}\|}$ and MSE values are normalized with the observation number n. As an alternative measure, the corruption effect of the corrupted observations on the weights and the original weights are presented as the *corruption rate* below

Table 6
ELM Results.

ELM Results for Non-Linear Case			
Methods	MSE	Rel. Err.	Corr. Rate
ELM	7.0031	5.3384	5.6655
SuBER	1.0626	2.0804	0.9662
ELM+$\ell 2$	7.0534	5.3575	5.6405
RP+Bisquare	**0.7061**	**1.6966**	**0.8745**
IRLS+Huber	1.2196	2.2286	1.4429

Table 7
ELM Results.

ELM Results for Boston Price Dataset			
Methods	MSE	Rel. Err.	Corr. Rate
ELM	34.1896	0.4582	2.4161
SuBER	**14.8351**	**0.3076**	0.5088
ELM+$\ell 2$	56.8503	0.5936	4.0187
RP+Bisquare	1.4507e+03	3.0544	**0.2306**
IRLS+Huber	15.2007	0.3111	0.6351

Table 8
ELM Results.

ELM Results for Diabetes Dataset			
Methods	MSE	Rel. Err.	Corr. Rate
ELM	615.0192	0.7787	1.1799
SuBER	**277.9764**	**0.5237**	**0.3165**
ELM+$\ell 2$	603.1095	0.7700	1.0743
RP+Bisquare	412.9272	0.6479	0.6624
IRLS+Huber	294.3530	0.5389	0.5048

i.e. $corruptionrate = \frac{\|w_o - w\|_2}{\|w_o\|_2}$, where w_o denotes the least squares solution obtained through the y values before the corruption occurs. In the tests, 100 simulations were made for each method, and the average of these results is reported. The corruptions are set such that $\|b\|_0 = 0.2n$ for the Tables 5 and 6 and $\|b\|_0 = 0.4n$ for Tables 7 and 8 as [7] and [5] perform tests up to this level of corruption.

The method *RP+Bisquare* is simply the random projections followed by robust regression library in MATLAB as the models are equivalent. From this analysis, it is apparent that our method is at least on-par with the competing methods, and even better under some of the categories. The linear model performance of the proposed model is slightly worse than the regular regression problem [5]. However, it is quite difficult to observe such linear data in real datasets. Even the Boston Price dataset is not completely linear even though it is one of the simplest datasets. Also, the increasing

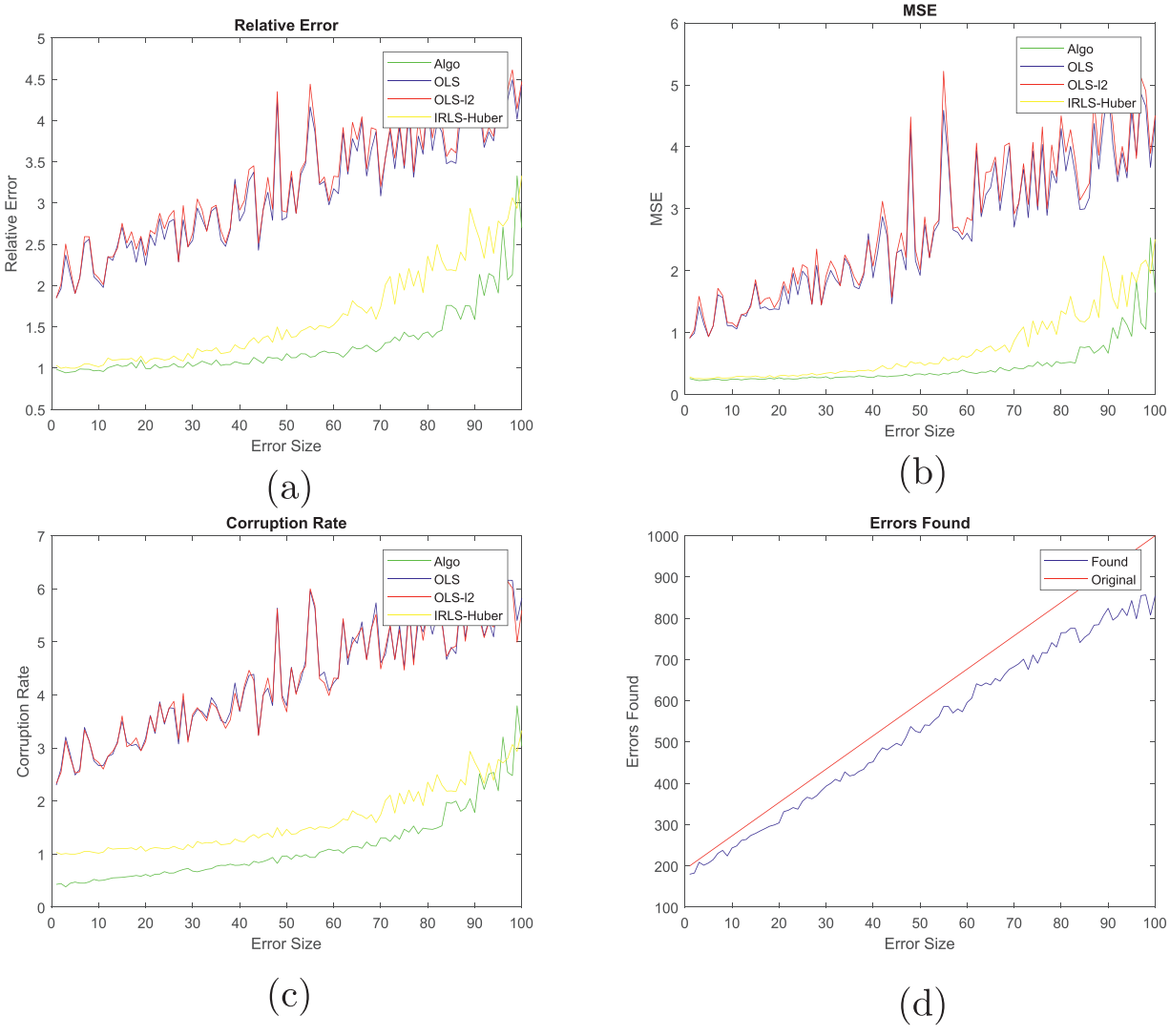

Fig. 1. Comparison of Results on Linear Case for Increasing Corruption Size.

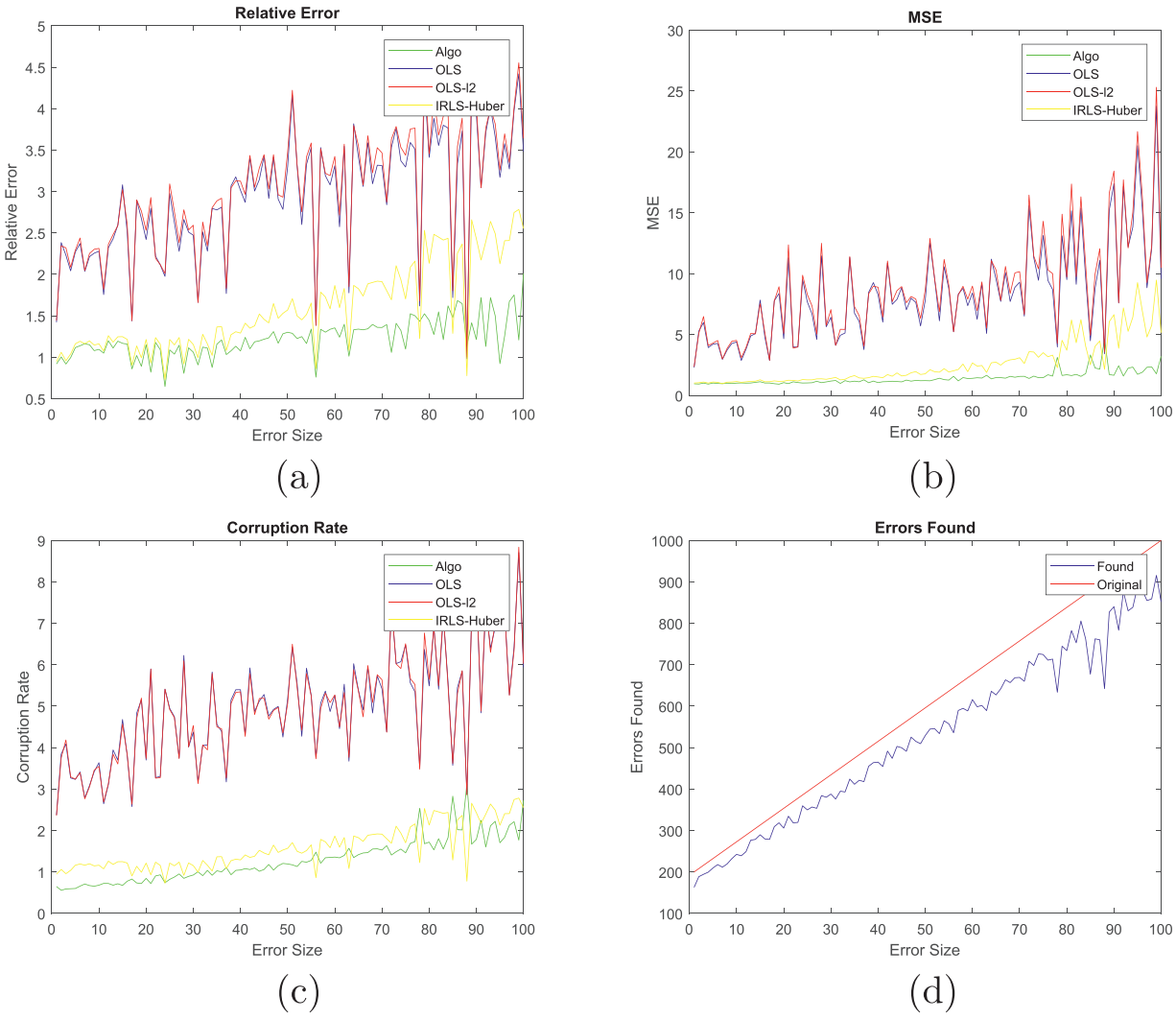

Fig. 2. Comparison of Results on Non-Linear Case for Increasing Corruption Size.

rate of corruption makes the convergence problematic for the robust regression libraries due to the corrupted entries. The proposed algorithm and the IRLS algorithm in [7] give on-par performances on the real dataset. As our method is originally proposed for denoising the data for different algorithms, a performance matching that of one of the most established robust ELM algorithms can be considered an encouraging result.

Fig. 1(a)–(c) are plotted with respect to the increasing corruption size for the linear model, and the rest of the plots in Fig. 2 concern the non-linear model. The corrupted index number was increased by 8 in each iteration. An average of 10 different runs per method is taken to smooth the effect of the random layer. In each figure, the dominance of our algorithm is visible. The regular OLS and ℓ_2 regularized OLS methods fail to adapt to the corruptions as expected. Commonly used Huber loss respectively performs better than the OLS. However, our algorithm performs better compared to the results of Huber loss as well. The Huber Loss used in these tests is borrowed from Chen et al. [7] IRLS-ℓ_2-Huber algorithm as it is one of the most competitive algorithms in the literature. In practice, Huber is the most common loss among the ML tools and libraries. Therefore it is the most meaningful loss selection for comparison.

The computational complexity varies with respect to the convergence of the inner step. In our algorithm, the inner step enjoys the property of "quick" steps as discussed in [5]. Since in each update weights are calculated with respect to the least squares solution without the need of a gradient method, the convergence of the weights occurs in very few iterations. In other robustness studies [7], proposed algorithms involving iteratively re-weighted least squares methods have a similar inner step. As a result, the time complexity of the proposed algorithm is comparable to the available methods in the literature. The advantages of our method can be summarised as follows:

1. Effective under heavy corruptions in terms of magnitude
2. Scales well with the corruption percentage
3. Hard-Threshold is simple to implement
4. Theoretically compatible with all injective activation functions
5. Time complexity increases with respect to the inner loop. Update method converges in 5–10 iterations
6. Fast in large scale data due to random projections (JL Lemma) with respect to the regular regression variant [5,6].

7. Conclusion

We have proposed an ELM architecture that can be used for data denoising and robust ELM regression problems. In the light of recent developments of convex neural networks, we have advocated that creating randomized activation patterns using ELM's would be a practical approach to evaluate the performance of the data points. To evaluate the data points, we cast the denoising

problem as a sparse recovery problem over the data points. This allows us to give theoretical guarantees for our algorithm, a feature which is rarely encountered in the literature. Furthermore, the denoised data obtained from our method can be fed into any NN architecture in order to benefit from the robustness properties of certain NN's. Therefore, the results we have obtained using our preprocessing step can be further improved when paired with proper NN architectures. In the second part of the study, we have shown that the proposed method can also be used as a standalone robust ELM architecture. Our numerical results indicated that both the denoising and standalone ELM methods achieve better performance compared to their competitors.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ömer Ekmekcioğlu: Conceptualization, Methodology, Software, Validation. **Deniz Akkaya:** Conceptualization, Methodology, Investigation. **Mustafa Ç. Pınar:** Methodology, Supervision, Writing – review & editing.

References

[1] G. Huang, Q. Zhu, C. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501, doi:10.1016/j.neucom.2005.12.126.

[2] G. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern.Part B 42 (2) (2012) 513–529.

[3] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin, A simple proof of the restricted isometry property for random matrices, Constr. Approx. 28 (2008) 253–263, doi:10.1007/s00365-007-9003-x.

[4] T. Cheng, Restricted conformal property of compressive sensing, CoRR (2014) arXiv preprint arXiv:1408.5543.

[5] K. Bhatia, P. Jain, P. Kar, Robust regression via hard thresholding, CoRR (2015) arXiv preprint arXiv:1506.02428.

[6] Y. Chen, C. Caramanis, S. Mannor, Robust sparse regression under adversarial corruption, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 28, PMLR, Atlanta, Georgia, USA, 2013, pp. 774–782.

[7] K. Chen, Q. Lv, Y. Lu, Y. Dou, Robust regularized extreme learning machine for regression using iteratively reweighted least squares, Neurocomputing 230 (2017) 345–358, doi:10.1016/j.neucom.2016.12.029.

[8] B. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comp. 24 (2) (1995) 227–234.

[9] B. Moghaddam, Y. Weiss, S. Avidan, Generalized Spectral Bounds for Sparse LDA, Technical Report TR2006-046, MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, 2006.

[10] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.

[11] T. Blumensath, M.E. Davies, Iterative hard thresholding for compressed sensing, Appl. Comput. Harmon. Anal. 27 (3) (2009) 265–274, doi:10.1016/j.acha.2009.04.002.

[12] A.S. Suggala, K. Bhatia, P. Ravikumar, P. Jain, Adaptive hard thresholding for near-optimal consistent robust regression, CoRR (2019) arXiv preprint arXiv:1903.08192.

[13] O.F. Alcin, A. Sengur, S. Ghofrani, M.C. Ince, GA-SELM: greedy algorithms for sparse extreme learning machine, Measurement 55 (2014) 126–132, doi:10.1016/j.measurement.2014.04.012.

[14] E. Tsakonas, J. Jaldén, N.D. Sidiropoulos, B. Ottersten, Convergence of the huber regression m-estimate in the presence of dense outliers, IEEE Signal Process. Lett. 21 (10) (2014) 1211–1214.

[15] Z. Shao, M. Er, An online sequential learning algorithm for regularized extreme learning machine, Neurocomputing 173 (2015), doi:10.1016/j.neucom.2015.08.029.

[16] M. Pilanci, T. Ergen, Neural networks are convex regularizers: exact polynomial-time convex optimization formulations for two-layer networks, ICML, 2020.

[17] Y. Bai, T. Gautam, Y. Gai, S. Sojoudi, Practical convex formulation of robust one-hidden-layer neural network training, 2021.

[18] P. Jain, P. Kar, Non-convex optimization for machine learning, Found. Trends Mach. Learn. 10 (3–4) (2017) 142–336, doi:10.1561/2200000058.

[19] J. Yi, G. Tan, Nonlinear compressed sensing based on composite mappings and its pointwise linearization, CoRR (2015) arXiv preprint arXiv:1506.02212.

[20] S. Dittmer, E.J. King, P. Maass, Singular values for ReLU layers, CoRR (2018) arXiv preprint arXiv:1812.02566.