Sparse Wavelet Networks

Amir Reza Sadri, Mehemmed Emre Celebi, *Senior Member, IEEE*, Nazanin Rahnavard, *Senior Member, IEEE*, and Satish E. Viswanath

***Abstract*—A wavelet network (WN) is a feed-forward neural network that uses wavelets as activation functions for the neurons in its hidden layer. By predetermining the wavelet positions and dilations, the WN can turn into a linear regression model. The common approach for the construction of these WN families is to use least-squares type algorithms. In this letter, we propose a novel approach by formulating a WN as a sparse linear regression problem, which we call a sparse wavelet network (SWN). In this WN, the problem of calculating the unknown inner parameters of the network becomes that of finding the sparse solution of an under-determined system of linear equations. Our sparse solution algorithm is a non-convex sparse relaxation approach inspired by smoothed L0 (SL0), a distinguished sparse recovery algorithm. The proposed SWN can be applied as a tool for the prediction and identification of dynamical systems.**

***Index Terms*—Wavelet network, sparse representation, non-convex regularization, system identification.**

I. Introduction

SPARSE modeling is a flourishing interdisciplinary field of research that bridges signal processing, machine learning, and statistics. It is particularly advantageous in selecting or constructing a small set of predictive variables in cases where the aim is to find the input and output of a system relationship [1]. Building on sparse modeling, in this letter, we propose a novel wavelet network (WN) that has the potential to be used in various areas, for example, in engineering disciplines [2]–[5].

The inherent time-frequency localization property of the wavelet basis makes them more effective than other basis functions. This insight inspired the concept of WNs by using wavelets as the basic components of a traditional neural network [6]–[9]. Depending on the types of wavelets and network training scheme used, there are different categories of WNs [10]. The adaptive wavelet network (AWN) is a primitive type of WN that takes advantage of the continuous wavelet transform for the formation of the network building blocks and a gradient type algorithm for model training [11]. Model initialization and training complications often limit AWNs to low dimensional applications [6].

A WN is called fixed grid wavelet network (FGWN) if it originates from the discrete wavelet transform with predefined network inner parameters (the wavelet shifts and scales) [11]. The FGWNs basically act by using various model structures or different parameter estimation algorithms [12]. Substantial techniques for modifying and improving the efficiency of FGWNs have been created in the literature. For example, in [6] multiscale wavelet decomposition was applied as the model construction and the orthogonal least-squares algorithm was applied for computing the network outer parameters. Li *et al.* extended the model structure based on multi-wavelet basis functions and refined the associated regression using a block least mean squares method [8] and an ultra-orthogonal forward regression algorithm aided by mutual information [12].

As a linear regression model, the output vector of an FGWN can be represented as the multiplication of a *wavelet matrix* and the *coefficient vector*. The common approach for finding the coefficient vector is based on greedy strategies such as forward selection which are highly suboptimal [13]. Since the construction of the wavelet matrix is based on the positions and dilations of wavelet coefficients, in order to simplify computational complexity, the FGWN regression problem may be considered as an optimization equation and equivalently as an under-determined systems of linear equations (USLE). Considering the sparse solution of a USLE taken from the corresponding FGWN, which is equivalent to the hidden layer weights, a network with low inner dimension is achieved. This procedure might be useful for high dimensional problems. In the current study, we take an FGWN as a sparse linear regression problem, which we refer to as a sparse wavelet network (SWN). Our proposed algorithm for finding the sparse solution is based on the graduated non-convexity (GNC) method and in particular, the smoothed ℓ_0 norm (SL0 algorithm) which is an effective and fast approach [14]. This letter is a major contribution to the literature on WN for at least two reasons: (i) sparse modeling of the WN which brings about a network with simple internal structure one that is easy to implement; (ii) analyzing the convergence of the SL0 method using ℓ_0 norm approximation with a non-convex but gradient-Lipschitz function.

II. Structure of SWN

Assume that the observations of input-output data pairs are as $\{(\mathbf{x}^{(p)}, y^{(p)}) : \mathbf{x}^{(p)} \in \mathbb{R}^n, y^{(p)} \in \mathbb{R}, p = 1, \ldots, P\}$. The pth output sample of the WN is given by [15]:

$$y^{(p)} = \sum_{i=1}^{m} \theta_i \left|\mathbf{D}_i^{1/2}\right| \psi(\mathbf{D}_i \mathbf{x}^{(p)} - \mathbf{B}\mathbf{t}_i) = \sum_{i=1}^{m} \theta_i \psi_i^{(p)} \tag{1}$$

where m is the number of wavelons (wavelet neurons) in the hidden layer, θ_i are the weights between the hidden layer and output, $\psi \in L^2(\mathbb{R}^n)$ is the mother wavelet function, $\mathbf{D}_i = \mathrm{diag}(\mathbf{d}_i)$, $\mathbf{d}_i \in \mathbb{R}^n$ is the scale parameter vector of the wavelets, $\mathbf{t}_i \in \mathbb{R}^n$ is the shift parameter vector of the wavelets, and $\mathbf{B} = \mathrm{diag}(\mathbf{b})$,

Manuscript received September 26, 2019; revised December 7, 2019; accepted December 8, 2019. Date of publication December 11, 2019; date of current version January 24, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Demetrio Labate.
(Corresponding author: Amir Reza Sadri.)

A. R. Sadri and S. E. Viswanath are with the Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: ars329@case.edu; sev21@case.edu).

M. E. Celebi is with the School of Computer Science, University of Central Arkansas, Conway, AR 72035 USA (e-mail: ecelebi@uca.edu).

N. Rahnavard is with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: nazanin@eecs.ucf.edu).

Digital Object Identifier 10.1109/LSP.2019.2959219

$\mathbf{b} \in \mathbb{R}^n$ is the discretization factor. Considering the total number of samples, the network output vector $\mathbf{y} \in \mathbb{R}^P$ given in matrix form as:

$$\mathbf{y} = \sum_{i=1}^{m} \theta_i \boldsymbol{\psi}_i = \mathbf{W}\boldsymbol{\theta} \tag{2}$$

where $\mathbf{W} = [\boldsymbol{\psi}_1 \dots \boldsymbol{\psi}_m]$ is called the *wavelet matrix* (dictionary). The vectors $\boldsymbol{\psi}_i = [\psi_i^{(1)} \dots \psi_i^{(P)}]^T$ are *regressors* (atoms) and $\boldsymbol{\theta} = [\theta_1 \dots \theta_m]^T$ is the *coefficient vector*.

A. The Wavelet Matrix

According to the multiscaling wavelet frame theorem [16], the wavelet matrix constitutes a multidimensional frame and has significant characteristics by the following elementary lemmalemm and corollary.

Lemma 1: If the columns of $\mathbf{W} = [\boldsymbol{\psi}_1 \dots \boldsymbol{\psi}_m]$ are frames, with frame bounds $A > 0, B < \infty$, then inequalities $A\mathbf{I} \preceq \mathbf{W}\mathbf{W}^T \preceq B\mathbf{I}$ hold. For a tight frame $A = B$ and thus $\mathbf{W}\mathbf{W}^T = A\mathbf{I}$.

Proof: See, for example, [17]. ■

Corollary 1: The wavelet matrix $\mathbf{W}$ is full row rank.

Proof: The matrix $\mathbf{W}$ is full row rank if and only if $\{\forall \mathbf{f} \in \mathbb{R}^P, \mathbf{f}\mathbf{W} = \mathbf{0} \Longrightarrow \mathbf{f} = \mathbf{0}\}$. The condition $\mathbf{f}\mathbf{W} = \mathbf{0}$ implies $\mathbf{f}\mathbf{W}\mathbf{W}^T = \mathbf{0}$, which in turn implies $\mathbf{f} = \mathbf{0}$ because the $\mathbf{W}\mathbf{W}^T$ in invertible according to Lemma 1. ■

B. The Coefficient Vector

Since the wavelet matrix is full row rank, the USLE extracted from the FGWN has infinitely many solutions. We are interested in seeking its sparsest solution of the coefficient vector. The sparsity of the coefficient vector affects the internal structure of the WN. The sparser the coefficient vector, the less the network's computational complexity. A WN with too many hidden layer nodes is slower, may cause training to diverge, or lead to overfitting, which would reduce the network generalizability [18]. On the other hand, having too few hidden units, results in large training and generalization errors due to underfitting and high statistical bias [6]. Therefore, we should look for a coefficient vector that has an acceptable error and as much sparsity as possible.

Finding the proper solution of vector $\boldsymbol{\theta}$ can be cast as a constrained optimization problem as follows:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\|_2 \le \epsilon \tag{3}$$

where ϵ is a predefined error tolerance.

Our strategy for solving (3) is based on the non-convex sparse regularization technique. These methods are part of the GNC family and are often significantly slower than greedy algorithms [19]. A fast GNC technique which is based on smoothed ℓ_0 norm (SL0) with reasonable computing time is proposed in [14]. Inspired by the SL0 method, we propose a mathematical framework for finding the sparsest solution of the USLE (2) as the coefficient vector of the SWN.

III. Finding a Sparse Solution

A. Non-Convex Regularization

The strategy of the SL0 algorithm is based on the definition of a smoothing parameter $\sigma \ge 0$ and approximates the smoothed ℓ_0 norm with a non-convex function $\|\cdot\|_\sigma$ as $\|\boldsymbol{\theta}\|_0 = \lim_{\sigma \to 0} \|\boldsymbol{\theta}\|_\sigma$. In this way, the sparsity is induced gradually by decreasing the smoothing parameter, so the nonconvexity of the smooth function increases without getting trapped in local minima [20].

The function $\|\boldsymbol{\theta}\|_\sigma : \mathbb{R}^n \to \mathbb{R}$ parameterized by $\sigma \ge 0$ is defined as

$$f(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_\sigma = \sum_{i=1}^{m} (1 - f_\sigma(\theta_i)) \tag{4}$$

where the one variable function $f_\sigma(\cdot)$ has the following properties:

P1) $\lim_{\sigma \to 0} f_\sigma(\theta_i) = \begin{cases} 1\,; & \text{if} \quad \theta_i = 0 \\ 0\,; & \text{if} \quad \theta_i \neq 0 \end{cases}$

P2) $f'_\sigma(\theta_i)$ is gradient-Lipschitz with constant M/σ^2, where M is a positive constant. Hence the second derivative of $f_\sigma(\theta_i)$ is bounded (i.e. $\forall \theta_i \in \mathbb{R} : |f''(\theta_i)| \le M/\sigma^2$).

With the definition of $\|\boldsymbol{\theta}\|_\sigma$, the sparsest solution of (3) is in the form

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in C_\epsilon} \left\{ f(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_\sigma \right\} \tag{5}$$

where $C_\epsilon = \{\boldsymbol{\theta} : \|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\| \le \epsilon\}$.

B. The Final Algorithm

The function f in the form of (4) is gradient-Lipschitz through the following Lemma.

Lemma 2: If $f_\sigma(\theta_i)$ is gradient-Lipschitz with constant L then $\|\boldsymbol{\theta}\|_\sigma$ in the form of (4) is gradient-Lipschitz with constant L.

Proof: See [20]. ■

A gradient-Lipschitz function has an elementary but important property which is expressed through the descent lemma as follows:

Lemma 3 (descent lemma [21]): Assume that $f : \mathrm{dom} f \to \mathbb{R}$ is gradient-Lipscitz function with constant $L > 0$. Then for any two vectors $\boldsymbol{\theta}, \boldsymbol{\theta}_k \in \mathrm{dom} f$

$$f(\boldsymbol{\theta}) \le f(\boldsymbol{\theta}_k) + \nabla^T f(\boldsymbol{\theta}_k)(\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2\gamma} \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|_2^2 \tag{6}$$

where $\gamma \in (0, 1/L]$ and $\mathrm{dom} f$ express the domain of the function f. The right hand side of (6) is called the *upper-bound* of $f(\boldsymbol{\theta})$ at the point $\boldsymbol{\theta}_k$ and it is shown by $\bar{f}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$. The minimum upper-bound is attained when $\gamma = 1/L$.

Proof: See, for example, [21]. ■

As it stands, $\bar{f}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k) = f(\boldsymbol{\theta}_k)$. Therefore, instead of minimizing f, we can minimize its upper-bound. Thus, the iterative solution algorithm for (5) is $\boldsymbol{\theta}_{k+1} = \mathrm{argmin}_{\boldsymbol{\theta} \in C_\epsilon} \bar{f}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$. Considering (4), we have

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta} \in C_\epsilon} \Big\{ \|\boldsymbol{\theta}_k\|_\sigma + \nabla^T \|\boldsymbol{\theta}_k\|_\sigma (\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2\gamma} \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|_2^2 \Big\} \tag{7}$$

equivalently

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta} \in C_\epsilon} \frac{1}{2} \left\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_k\right\|_2^2 \tag{8}$$

where $\bar{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k - \gamma \nabla \|\boldsymbol{\theta}_k\|_\sigma$. So, the final solution to find the sparse solution of USLE (2), which is summarized in Algorithm 1, can be obtained.

Algorithm 1: The Sparse Solution of Coefficient Vector.

Input: $\mathbf{y}$, $\mathbf{W}$, M, σ_0, $\sigma_{\min}$, $0 < c < 1$, K, γ, ϵ
initialization : $\boldsymbol{\theta} = \mathbf{0}$, $\sigma = \sigma_0$
while $\sigma > \sigma_{\min}$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} - \gamma \nabla \|\boldsymbol{\theta}\|_\sigma$
    $\boldsymbol{\theta} = \operatorname{argmin}_{\boldsymbol{\theta} \in C_\epsilon} \frac{1}{2} \|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|_2^2$
  end for
  $\sigma = c\sigma$
  $\gamma = (\sigma^2/M)\gamma$
end while
Output: $\boldsymbol{\theta}$

Remark 1: In Algorithm 1, σ_0, $\sigma_{\min}$, and c are the initial value, the final value, and the reduction factor for σ, respectively, K is the number of inner-loop iterations, and γ is the learning rate.

Remark 2: Since $\bar{f} : C_\epsilon \times C_\epsilon \to \mathbb{R}$ satisfies $\bar{f}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \geq f(\boldsymbol{\theta})$, $\bar{f}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k) = f(\boldsymbol{\theta}_k)$ for $\boldsymbol{\theta}, \boldsymbol{\theta}_k \in C_\epsilon$, $\bar{f}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ is so-called *majorization* function of $f(\boldsymbol{\theta})$ [22]. Therefore, our algorithm is a type of majorization-minimization algorithms [21].

Remark 3: According to the *proximal operator* definition, (8) can be rewritten as $\boldsymbol{\theta}_{k+1} = \mathrm{prox}_{\gamma \mathrm{g}}(\overline{\boldsymbol{\theta}}_{\mathrm{k}})$, where g is an indicator function. So, our method can be considered as a proximal method for non-convex optimization [20].

C. Convergence Analysis

We will now assess the bound of the parameter γ to guarantee convergence of the iterations in (7) through the following theorem.

Theorem 1: Let $f(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_\sigma$. Then, the sequence $\{\boldsymbol{\theta}_k\}$ in (8) converges to a stationary point of f. To guarantee convergence, parameter γ should satisfy

$$0 < \gamma \leq \frac{\sigma^2}{M}. \tag{9}$$

Proof: According to (7), the iterations $\boldsymbol{\theta}_{k+1}$ can be written as the associated algorithm

$$\boldsymbol{\theta}_{k+1} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \nabla^T \|\boldsymbol{\theta}_k\|_\sigma (\boldsymbol{\theta} - \boldsymbol{\theta}_k) + \frac{1}{2\gamma} \|\boldsymbol{\theta} - \boldsymbol{\theta}_k\|_2^2 \right\}. \tag{10}$$

Since $\boldsymbol{\theta}_{k+1}$ is the minimizer of (10)

$$\nabla^T \|\boldsymbol{\theta}_k\|_\sigma (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) + \frac{1}{2\gamma} \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2^2 \leq \mathbf{0}. \tag{11}$$

On the other hand, by (6) for minimum upper-bound of $f(\boldsymbol{\theta})$ at the point $\boldsymbol{\theta}_k$, we have

$$\begin{aligned} \|\boldsymbol{\theta}_{k+1}\|_\sigma &\leq \|\boldsymbol{\theta}_k\|_\sigma + \nabla^T \|\boldsymbol{\theta}_k\|_\sigma (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) \\ &\quad + \frac{L}{2} \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2^2 \end{aligned} \tag{12}$$

where L is the Lipschitz constant of the $\nabla \|\boldsymbol{\theta}\|_\sigma$ and according to Lemma 2, $L = M/\sigma^2$.

Adding (11) and (12) results in

$$f(\boldsymbol{\theta}_{k+1}) \leq f(\boldsymbol{\theta}_k) - \left(\frac{1}{2\gamma} - \frac{M}{2\sigma^2} \right) \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2^2 \tag{13}$$

which implies that the sequence $\{f(\boldsymbol{\theta}_k)\}_0^\infty$ is decreasing if $0 < \gamma \leq \sigma^2/M$. Since f is bounded from below ($\|\boldsymbol{\theta}\|_\sigma \approx \|\boldsymbol{\theta}\|_0$) and decreasing, we conclude that $\{f(\boldsymbol{\theta}_k)\}_0^\infty$ converges.

Summing (13) over all $k \geq 0$ leads to

$$\sum_{k=0}^{\infty} \left\{ \left(\frac{1}{2\gamma} - \frac{M}{2\sigma^2} \right) \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|_2^2 \right\} \leq f(\boldsymbol{\theta}_0) - f(\boldsymbol{\theta}_\infty). \tag{14}$$

It is clear that right-hand side of (14) is finite and non-negative. Necessarily, $\boldsymbol{\theta}_{k+1} \to \boldsymbol{\theta}_k$ and therefore, $\{\boldsymbol{\theta}_k\}_0^\infty$ converges.

Furthermore, since $\boldsymbol{\theta}_{k+1}$ is the minimizer of (10), we have

$$\nabla \|\boldsymbol{\theta}_k\|_\sigma + \frac{1}{\gamma}(\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) = \mathbf{0}. \tag{15}$$

Since $\boldsymbol{\theta}_{k+1} \to \boldsymbol{\theta}_k$, so $\nabla \|\boldsymbol{\theta}_k\|_\sigma \to \mathbf{0}$. This means that as $k \to \infty$, $\boldsymbol{\theta}_k \to \boldsymbol{\theta}^*$ where $\boldsymbol{\theta}^*$ is a stationary point of f. ■

D. Tight Wavelet Frame

At each iteration of the algorithm, the following constrained minimization problem needs to be solved:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \left\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\right\|_2^2 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\|_2 \leq \epsilon \tag{16}$$

where $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} - \gamma \nabla \|\boldsymbol{\theta}\|_\sigma$ and ϵ denotes the error tolerance. To solve (16), we derive the Lagrangian with multiplier λ in the form

$$L(\boldsymbol{\theta}, \lambda) = \frac{1}{2} \left\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\right\|_2^2 + \lambda(\|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\|_2^2 - \epsilon^2). \tag{17}$$

Karush-Kuhn-Tucker conditions imply the following optimality conditions:

$$\begin{cases} \boldsymbol{\theta}^* = (\mathbf{I} + 2\lambda^* \mathbf{W}^T \mathbf{W})^{-1} (\bar{\boldsymbol{\theta}} + 2\lambda^* \mathbf{W}^T \mathbf{y}) \\ \|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\|_2^2 = \epsilon^2 \\ \lambda^* \geq 0 \end{cases} \tag{18}$$

and after substitutions, we obtain the following equation

$$\left\|\mathbf{y} - \mathbf{W}(\mathbf{I} + 2\lambda^* \mathbf{W}^T \mathbf{W})^{-1} (\bar{\boldsymbol{\theta}} + 2\lambda^* \mathbf{W}^T \mathbf{y})\right\|_2^2 = \epsilon^2. \tag{19}$$

Generally there is no closed-form solution for this nonlinear equation, unless $\mathbf{W}$ is a tight frame.

According to Lemma 1, if a wavelet matrix is a tight frame then $\mathbf{W}\mathbf{W}^T = A\mathbf{I}$. By applying the matrix inversion lemma, we obtain: $(\mathbf{I} + 2\lambda^* \mathbf{W}^T \mathbf{W})^{-1} = \mathbf{I} - (2\lambda^*/(1 + 2\lambda^* A))\mathbf{W}^T \mathbf{W}$, which by combining (19) and (18), leads to

$$\begin{cases} \lambda^* = \dfrac{1}{2A} \max \left\{ \dfrac{\left\|\mathbf{y} - \mathbf{W}\bar{\boldsymbol{\theta}}\right\|_2}{\epsilon} - 1, 0 \right\} \\ \boldsymbol{\theta}^* = \bar{\boldsymbol{\theta}} + \dfrac{2\lambda^*}{1 + 2\lambda^* A} \mathbf{W}^T (\mathbf{y} - \mathbf{W}\bar{\boldsymbol{\theta}}) \end{cases}. \tag{20}$$

Since this approach simplifies the computations, in this study, we consider only tight wavelet frames for the construction of the WN wavelet matrix. It is worth mentioning that, if $\epsilon = 0$ the final solution of the algorithm is given by $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} - \gamma \nabla \|\boldsymbol{\theta}\|_\sigma$. In this situation, the algorithm is indeed a gradient descent with step-size γ (similar to SL0).

IV. Simulation Results

In WN equation (1), $\mathbf{d}_i = [a^{i_1}, \ldots, a^{i_n}]^T, \mathbf{b} = [b, \ldots, b]^T_{1\times n}$ with $a > 1$ and $b > 0$ are considered [16]. One dimensional mother wavelet admissibility theorem for tight wavelet frame [17] states that the frame bound is equal to $A = \frac{2\pi}{b \ln a} \int_0^\infty \omega^{-1} |\widehat{\psi}(\omega)|^2 \mathrm{d}\omega$, where $\widehat{\psi}(\omega)$ is the Fourier transform of $\psi(x)$. Since in the multidimensional case, the wavelet function is the tensor product of one-dimensional mother wavelets [16], the tight frame bound is nA, where n is the WN input dimensionality. As is customary in the WN literature, we use the Mexican hat as the mother wavelet function in the construction of SWN. To take advantage of the Mexican hat tight wavelet frame, we are confined to choose $\{1 < a \leq 2^{0.25}\}$ [17, Ch. 3, p. 71] or $\{a = 2, 0 < b \leq 0.75\}$ [17, Ch. 3, p. 76]. In our experiments, we used MATLAB 9.4 on a PC with Intel(R) Core(TM) i7 CPU 930 (2.80 GHz) and 12 GB RAM on a 64 bit Windows 10 operating system.

As an example, suppose the nonlinear two inputs, two outputs system is given by

$$\begin{cases} y_1^{(p)} = \dfrac{1}{1 + \left(y_1^{(p-1)}\right)^2} \left(0.1 y_1^{(p-1)} + 0.9 u_1^{(p-2)} + 0.1 u_2^{(p-3)}\right) \\ y_2^{(p)} = \dfrac{1}{1 + \left(y_2^{(p-1)}\right)^2} \left(0.5 y_2^{(p-1)} + 0.3 u_1^{(p)} + u_2^{(p-1)}\right) \end{cases} \tag{21}$$

where, the pairs $(u_1^{(p)}, u_2^{(p)})$ and $(y_1^{(p)}, y_2^{(p)})$ are the input and output samples, respectively. An additive independent and identically distributed (i.i.d.) noise is also considered for both system outputs where the noise term is uniformly distributed in $[-\epsilon, \epsilon]$. For identifying this system, two SWN with $n = 3$ inputs and one output is formed. The inputs of the first SWN are $\mathbf{x}_1^{(p)} = [y_1^{(p-1)}, u_1^{(p-2)}, u_2^{(p-3)}]^T$ and the inputs of the second SWN are $\mathbf{x}_2^{(p)} = [y_2^{(p-1)}, u_1^{(p)}, u_2^{(p-1)}]^T$. 900 points are used for training the network. Half of them are uniformly distributed on $[-1, 1]$ and the remaining are sinusoids of the form $1.05 \sin(\pi k/45)$.

$$u_1^{(p)} = \begin{cases} \sin\left(\frac{\pi p}{25}\right) & p < 250 \\ 0.5 & 250 \leq p < 500 \\ -0.5 & 500 \leq p < 750 \\ 0.1(\sin\left(\frac{\pi p}{25}\right) + \sin\left(\frac{\pi p}{32}\right) & \\ \quad + 2\sin\left(\frac{\pi p}{10}\right)) & 750 \leq p < 1000 \end{cases} \tag{22}$$

$$u_2^{(p)} = \begin{cases} 0.6\sin\left(\frac{\pi p}{25}\right) & p < 250 \\ 0.3 & 250 \leq p < 500 \\ -0.3 & 500 \leq p < 750 \\ 0.01(\sin\left(\frac{\pi p}{25}\right) + \sin\left(\frac{\pi p}{32}\right) & \\ \quad + 20\sin\left(\frac{\pi p}{10}\right)) & 750 \leq p < 1000 \end{cases} \tag{23}$$

The Mexican hat wavelet $\psi(x) = (1 - x^2)\exp(-0.5x^2)$ for each dimension is computed for all input samples by choosing $a = 2^{0.25}, b = 2$ and the scale levels over the interval $[-20, 20]$. Since the Mexican hat is a compactly supported wavelet by the support $[-4, 4]$, it can be shown that [6] the variation range for the shift parameter is $t_i \in [-5, 36]$. So, the number of wavelet frame bases is $\text{range}[-20, 20] \times \text{range}[-5, 36] = 1722$ and the frame bound is $A = 6\pi/\ln 2$. For running Algorithm 1, we considered $f_\sigma(\theta_i) = \exp(-(\theta_i/\sigma)^2)$, $M = 2$,

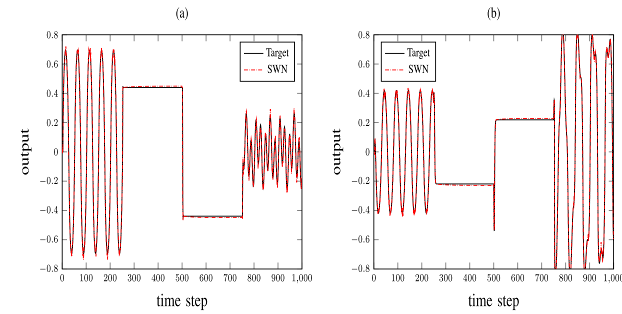

Fig. 1. Test results of the proposed SWN for the actual and approximated signals: (a) the output $\mathbf{y}_1$ and (b) the output $\mathbf{y}_2$.

TABLE I
RMSE Comparison of Some WNs Over the Testing Data

	Noise Level ($\epsilon = 0.1$)				Noise Level ($\epsilon = 0.25$)			
	Output 1		Output 2		Output 1		Output 2	
WN Type	wavelons	RMSE	wavelons	RMSE	wavelons	RMSE	wavelons	RMSE
AWN [15]	52	0.094456	56	0.097132	49	0.098001	51	0.099985
FGWN [18]	40	0.022512	43	0.030467	35	0.031605	37	0.041413
FGWN [11]	23	0.022193	26	0.029117	19	0.027884	22	0.040955
FGWN [24]	22	0.021033	26	0.028111	19	0.026981	21	0.039826
SWN	15	0.015521	19	0.021266	12	0.021703	14	0.033379

$\sigma_0 = 0.5$, $\sigma_{\min} = 0.05$, $c = 0.8$, and $\gamma = 0.1$. The algorithm is terminated when it reaches $K = 15$ or when a given noise level threshold ($\epsilon = 0.1, 0.25$) is met.

After SWN construction and determination of the coefficient vector, (22) and (23) test signals are used for testing the performance of the SWN models. The performance of the SWN outputs for the test signals are presented in Fig. 1. The SWN performance was evaluated through simulations and compared against several WN models using the same training and testing procedure. The results in terms of the number of network wavelons and root mean square error (RMSE) between the actual and predicted output signals for two different noise levels are given in Table I.

The AWN [15] is trained using the backpropagation algorithm in the publicly available wavenet MATLAB Toolbox [23]. In the FGWNs, instead of using the proposed algorithm, the orthogonal least-squares method [11] and [18] or the D-optimality orthogonal matching pursuit algorithm [24] is applied. It can be seen that the number of the proposed SWN wavelons are much lower than the other methods, while at the same time our models result in considerably smaller RMSE for both system outputs. Here the sparsity concept is directly fed to the model construction, which provides a parsimonious model with good generalization performance.

V. Conclusion

In this letter, we made a novel contribution by looking at wavelet networks from a sparse linear regression point of view and proposed a sparse wavelet network (SWN). In an SWN, the sparsity concept is equivalent to the number of hidden layer neurons which are specified from the sparse solution of a linear regression model. Our sparse solution algorithm is based on ℓ_0 norm approximation with a non-convex gradient-Lipschitz function. The function non-convexity can be controlled by varying the smoothing parameter in each algorithm iteration [14]. The proposed SWN has a solid mathematical foundation with low complexity which can be utilized in practical implementations.

REFERENCES

[1] W. Pan, Y. Yuan, J. Gonalves, and G. B. Stan, "A sparse Bayesian approach to the identification of nonlinear state-space systems," *IEEE Trans. Autom. Control*, vol. 61, no. 1, pp. 182–187, Jan. 2016.

[2] P. Ong and Z. Zainuddin, "Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction," *Appl. Soft Comput. J.*, vol. 80, pp. 374–386, Jul. 2019.

[3] L. Lei, W. Chen, Y. Xue, and W. Liu, "A comprehensive evaluation method for indoor air quality of buildings based on rough sets and a wavelet neural network," *Build. Environ.*, vol. 162, Sep. 2019, Art. no. 106296.

[4] M. El-Diasty, S. Al-Harbi, and S. Pagiatakis, "Hybrid harmonic analysis and wavelet network model for sea water level prediction," *Appl. Ocean Res.*, vol. 70, pp. 14–21, Jan. 2018.

[5] H. Wei, S. Billings, Y. Zhao, and L. Guo, "An adaptive wavelet neural network for spatio-temporal system identification," *Neural Netw.*, vol. 23, no. 10, pp. 1286–1299, 2010.

[6] S. Billings and H.-L. Wei, "A new class of wavelet networks for nonlinear system identification," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 862–874, Jul. 2005.

[7] Y. Li, M. Lei, W. Cui, Y. Guo, and H. Wei, "A parametric time-frequency conditional Granger causality method using ultra-regularized orthogonal least squares and multiwavelets for dynamic connectivity analysis in EEGs," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 12, pp. 3509–3525, Dec. 2019.

[8] Y. Li, H. Wei, and S. A. Billings, "Identification of time-varying systems using multi-wavelet basis functions," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 3, pp. 656–663, May 2011.

[9] H. Wei, S. A. Billings, Y. Zhao, and L. Guo, "Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 181–185, Jan. 2009.

[10] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Hoboken, NJ, USA: Wiley, 2013.

[11] A. R. Sadri, M. Zekri, S. Sadri, N. Gheissari, M. Mokhtari, and F. Kolahdouzan, "Segmentation of dermoscopy images using wavelet networks," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 1134–1141, Apr. 2013.

[12] Y. Li, W. Cui, Y. Guo, T. Huang, X. Yang, and H. Wei, "Time-varying system identification using an ultra-orthogonal forward regression and multiwavelet basis functions with applications to EEG," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2960–2972, Jul. 2018.

[13] L. Zhang and K. Li, "Forward and backward least angle regression for nonlinear system identification," *Automatica*, vol. 53, pp. 94–102, Mar. 2015.

[14] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009.

[15] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 889–898, Nov. 1992.

[16] T. Kugarajah and Q. Zhang, "Multidimensional wavelet frames," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1552–1556, Nov. 1995.

[17] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA, USA: SIAM, 1992.

[18] Q. Zhang, "Using wavelet network in nonparametric estimation," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, pp. 227–236, Mar. 1997.

[19] I. W. Selesnick and I. Bayram, "Enhanced sparsity by non-separable regularization," *IEEE Trans. Signal Process.*, vol. 64, no. 9, pp. 2298–2313, May 2016.

[20] M. Sadeghi and M. Babaie-Zadeh, "Iterative sparsification-projection: Fast and robust sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5536–5548, Jun. 2016.

[21] H. H. Sohrab, *Basic Real Analysis*. Cambridge, MA, USA: Birkhauser Basel, 2014.

[22] T. Chen, M. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 2933–2945, Nov. 2014.

[23] "Matlab and wavenet toolbox release 2007a," The MathWorks, Inc., Natick, MA, USA.

[24] A. R. Sadri, S. Azarianpour, M. Zekri, M. E. Celebi, and S. Sadri, "WN-based approach to melanoma diagnosis from dermoscopy images," *IET Image Process.*, vol. 11, no. 7, pp. 475–482, Jul. 2017.