ISSN: 2692-5079

A Compact Gradient Based Neural Network for Capon Spectral Estimation

Abderrazak Benchabane, and Fella Charif

Abstract—This paper describes the use of a novel gradient based recurrent neural network to perform Capon spectral estimation. Nowadays, in the fastest algorithm proposed by Marple et al., the computational burden still remains significant in the calculation of the autoregressive (AR) Parameters. In this paper we propose to use a gradient based neural network to compute the AR parameters by solving the Yule-Walker equations. Furthermore, to reduce the complexity of the neural network architecture, the weights matrix-inputs vector product is performed efficiently using the fast Fourier transform. Simulation results show that proposed neural network and its simplified architecture lead to the same results as the original method which prove the correctness of the proposed scheme.

Keywords—Gradient-based neural networks, Toeplitz systems,Fast Fourier Transform, Spectral estimation. AR model.

I. Introduction

SPECTRAL estimation has been widely used in many practical applications such as radar, speech and communication, to mention a few [1]. Over the last century, a great effort has been made to develop new techniques for high performance spectral estimation. Broadly, the developed techniques can be classified in two categories: nonparametric and parametric methods [2]. The non parametric spectral estimation approaches are relatively simple and easy to compute via the Fast Fourier Transform (FFT) algorithm. However, these methods require the availability of long data records in order to yield a a higher frequency resolution. For the parametric approaches, we first design a model for the process of interest which is described by a small number of parameters. Based on this model, the spectral density of the process can be obtained by substituting the estimated parameters of the model in the theoretical expression of the spectral density [2]. If the assumed model fits the data well, the parametric methods yield more accurate spectrum with higher resolution than the former. However, if the model does not satisfy the data which is mostly encountered in practice, the parametric methods will inevitably lead to an incorrect estimate. Minimum Variance Spectral Estimation (MVSE), referenced also as the Capon spectral estimation, is one of the nonparametric methods which have a great interest in recent years[3]. It can be interpreted as an adaptive filter-bank spectral estimation due to the fact that it uses data-dependent bandpass filters. To compute directly the capon spectrum, we need to invert the correlation matrix of the available data samples; consequently, a huge amount of computation is required when the dimension of the matrix is large. This may not be practical for real-time processing. Nowadays, the only implementation in practice employs the fast algorithm developed by Musicus exploiting the Toeplitz structure of the covariance matrix[4]. this algorithm proceeds in three steps; the computation of the autoregressive coefficients using the Levinson-Durbin algorithm, the computation of the correlation of these parameters and then the evaluation of the spectral density using the fast Fourier transform.

In an attempt to further reduce the computational complexity,a super-fast algorithm was proposed in [5]. In this algorithm the power spectral density is only done in two stages, the first is used to compute the AR coefficients as in Musicus algorithm. The second step allows to evaluate the power spectral density using the fast Fourier transform. The main advantage in this algorithm is that the computation of the correlation AR parameters is avoided. We note that in the two algorithms, the main computational burden is the computation of the AR parameters using the Levinson-Durbin algorithm which requires $O(p^2)$ flops for a model of order p.

Neural networks have been widely investigated in spectral estimation [6], [7], [8]. The major advantage of neural network over other methods resides in their capability to perform more complex calculation in real times due to their parallel-disturbed nature [9], [10], [11]. The neural network consists of a large number of simple devices; each one computes little more than weighted sums. Consequently the complexity of computation can be dramatically reduced and the total computation time is comparable to the response time of a single processor which can be very small [7]. By considering these advantages, we investigate to use a gradient based recurrent neural networks to compute the AR parameters from the Yule-Walker equations. As well known, the activation of neural networks is based on the computation of a full matrix by vector multiplication where the matrix contains the connection weights and the vector contains the activation values. Moreover, for the same connection matrix, the multiplication has to be done in every iteration with a new vector input. In such cases, one seeks to identify special properties of the connection matrix in order to reduce the complexity computation.

The paper is organized as follows. In section 2, existing schemes for power spectral density estimation are presented . Section 3 presents the online spectral estimation using the discrete gradient neural network which is followed by a novel architecture of the DGNN in section 4. Section 5 is devoted to simulation studies. Finally , the conclusions are presented in Sect. 6.

A. Benchabane is with the Department of Electronics and Telecommunications, University of Kasdi Merbah, Ouargla, Algeria e-mail: ge.benchabane@gmail.com

F. Charif is with the Department of Electronics and Telecommunications, University of Kasdi Merbah, Ouargla, Algeria e-mail: fella.charif@gmail.com

II. Existing schemes

A. Standard Algorithm

We consider a data sequence of a stationary signal is available. Let R_p is the $p \times p$ autocorrelation matrix of the data samples. The MVSE in its standard form is as follow [3]:

$$p(f) = \frac{1}{e^H(f) R_p^{-1} e(f)} \tag{1}$$

where $e(f)$ is the frequency vector . in this standard formulation, the MVSE requires order p^3 computations.

B. Fast Algorithm

To avoid the direct computation of MVSE in its standard form which is not computationally efficient, Musicus has proposed a fast algorithm using the Gohberg-Semencul formula and the fast Fourier transform. The MVSE in its new formulation is given by [4]:

$$p(f) = \frac{1}{\sum_{k=-p}^{p} \mu(k) e^{-j2\pi fk}} \tag{2}$$

where

$$\mu(k) = \begin{cases} \frac{1}{\delta_p} \sum_{i=0}^{p-k} (p+1-k-2i) a_p(i) a_p^*(k+i), \\ k = 0, ..., p \\ \mu^*(-k), k = -p, ..., -1 \end{cases} \tag{3}$$

The terms $a_p(i)$ are the AR parameters computed with Levinson Recursion.

C. Super-Fast Algorithm

In [5], the authors have proposed a super-fast algorithm, in which the fast Fourier transform is fully exploited and the calculation of the autocorrelation of the AR parameters is avoided. The expression (2) in this case is rewritten as [5]:

$$p(f) = \frac{1}{(p+1)A(f)A^*(f) - A(f)B^*(f) - A^*(f)B(f)} \tag{4}$$

where

$$A(f) = \sum_{n=0}^{p} a_p(n) e^{-j2\pi fn} \tag{5}$$

$$B(f) = \sum_{n=0}^{p} n a_p(n) e^{-j2\pi fn} \tag{6}$$

The two terms $A(f)$, $B(f)$ are evaluated using the FFT.

III. Discrete based gradient neural network for spectral estimation

As mentioned in subsection 2.3, the spectral estimation using the super-fast algorithm is performed in two steps; firstly we compute the AR parameters using the Levinson-Durbin algorithm, and then we evaluate of the expression (4) using the FFT algorithm. we note here that the main computational burden of the super-fast algorithm is the step one, which requires order p^2 operations. in this paper, we propose to use the Gradient Neural Network (GNN) to perform the first step of the super-fast algorithm in just $O(plogp)$ operations.

A. AR parameters estimation using neural networks

Consider the parameter estimation problem of the noisy AR signal system [1], [6]:

$$x(n) = -\sum_{i=1}^{p} a_i x(n-i) + e(n) \tag{7}$$

where $a_i, i = 1, ..., p$ are the unknown AR parameters, $x(n-i), i = 1, ..., p$ are the p last data samples; $e(n)$ is white Gaussian process with variance σ_e^2 . The AR parameters to be estimated using the noisy observations are the solution of the Yule-Walker equations given by [1]:

$$Ra = r \tag{8}$$

where

$$R = \begin{bmatrix} r_x(0) & r_x(-1) & \cdots & r_x(-p+1) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p-1) & r_x(p-2) & \cdots & r_x(0) \end{bmatrix} \tag{9}$$

$$a = \begin{bmatrix} a_1 & a_2 & \cdots & a_p \end{bmatrix}^T \tag{10}$$

$$r = -\begin{bmatrix} r_x(1) & r_x(2) & \cdots & r_x(p) \end{bmatrix}^T \tag{11}$$

with:

$$r_x(k) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-k} x^*(n) x(n+k), k = 0, ..., p \\ r_x^*(-k), k = -p+1, ..., -1 \end{cases} \tag{12}$$

The GNN system design is based on the set of linear equations $Ra - r = 0$. To do this, the parameters estimation problem must be transformed to a minimization problem suitable for dynamic neural networks processing [6], [7]. Conventional gradient-based neural networks have been developed and widely investigated for online solution of the linear system [12], [13], [14], [15].
Consider the scalar-valued norm-based energy function

$$E = \frac{1}{2} \|Ra - r\|_2^2 \tag{13}$$

with $\|.\|_2$ denoting the two-norm of a vector. The minimum of this cost function is the solution of the above linear system $Ra = r$. To reach this minimum let us taking the negative of the gradient of the energy function

$$-\frac{\partial E}{\partial a} = -R^T (Ra - r) \tag{14}$$

By using a typical continuous-time adaptation rule, equation (6) leads to the following differential equation (linear GNN)

$$\frac{da}{dt} = -\frac{\partial E}{\partial a} = -\gamma R^T (Ra(t) - r) \tag{15}$$

where $\gamma > 0$ is a design parameter used to scale the GNN convergence rate. We could obtain the general nonlinear GNN model by using a general nonlinear activation function $f(\cdot)$ as follows [12]:

$$\dot{a}(t) = -\gamma R^T f (Ra(t) - r) \tag{16}$$

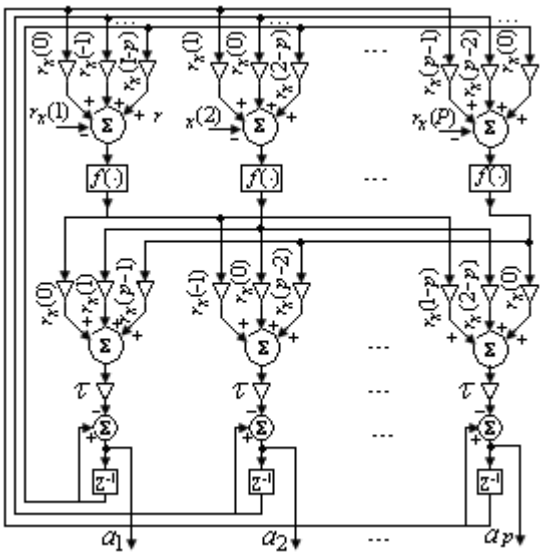

Fig. 1. Architecture of DGNN model

A discrete model of GNN can be obtained by the use of the forward-difference rule to compute $\dot{a}(t)$.

$$\dot{a}(t=kh) \approx (a((k+1)h) - a(kh))/h \tag{17}$$

where h denotes the sampling interval. For presentation convenience, we use $a_k = a(t=kh)$. Thus the presented DGNN model (14) can be reformulated as:

$$a_{k+1} = a_k - \tau R^T f(Ra_k - r) \tag{18}$$

with $\tau = \gamma h$. Figure (1) shows the architecture of the discrete neural network. As we can see, we have $2p^2$ weighting function, p adders of p elements, p adders of $p+1$ elements and p time-delays.

B. Spectrum estimation using DGNN model and the FFT algorithm

In this subsection, we describe the spectrum estimation procedure using both the DGNN model and the super-fast algorithm proposed in [5]. As mentioned below, in the exression (4), the terms $A(f)$ and $B(f)$ may be evaluated by the use of the FFT algorithm. Instead of using the Levinson-Durbin algorithm to compute the AR parameters, we see that the DGNN constitutes a good alternative to compute these parameters. So, the proposed spectrum estimation algorithm can be described as follow: ones the AR parameters are computed using the DGNN model, we compute the two terms $A(f)$ and $B(f)$ using the FFT algorithm and finally we need some algebric operators to compute the spectrum. The figure (2) shows the diagram of the proposed scheme.

IV. Compact architecture of the DGNN

By settig $z_1 = Ra_k$, $z_2 = f(z_1 - r)$ and $z_3 = R^T z_2$, the dynamic of the neural network can be rewritten in a compact form as:

$$a_{k+1} = a_k - \gamma h z_3 \tag{19}$$

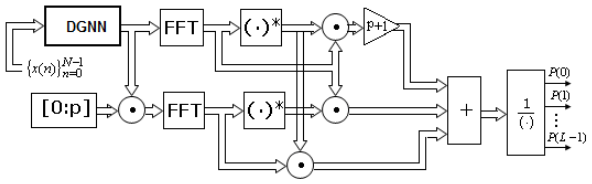

Fig. 2. Diagramm of the proposed model

This equation consists of two Toeplitz matrix-vector products $z_1 = Ra_k$ and $z_3 = R^T z_2$ which can be computed efficiently using the following algorithm.

A. Fast matrix-vector product computation

Since R is a Toeplitz matrix which is fully defined by its first column and first row, thus it depends only on $2p-1$ parameters rather than p^2 . To compute the product z_1 , the Toeplitz matrix R can be first embedded into a circulant matrix $C \in \mathcal{R}^{2p \times 2p}$ as [16], [17]:

$$C = \begin{bmatrix} R & S \\ S & R \end{bmatrix} \tag{20}$$

where

$$S = \begin{bmatrix} 0 & r_x(p-1) & \cdots & r_x(1) \\ r_x(1-p) & 0 & \cdots & r_x(2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(-1) & r_x(-2) & \cdots & 0 \end{bmatrix} \tag{21}$$

The matrix S never needs to be formed explicitly as C is simply a Toeplitz matrix with columns described by a circular-shift of the first column of the matrix given by:

$$c_1 = [r_x(0) r_x(1) \cdots r_x(p-1) 0 r_x(1-p) r_x(2-p) \cdots r_x(-1)] \tag{22}$$

Now we form a new matrix-vector product as:

$$C \cdot \begin{bmatrix} a_k \\ 0_n \end{bmatrix} = \begin{bmatrix} R & S \\ S & R \end{bmatrix} \cdot \begin{bmatrix} a_k \\ 0_n \end{bmatrix} = \begin{bmatrix} Ra_k \\ Sa_k \end{bmatrix} \tag{23}$$

Note that the vector $\begin{bmatrix} a_k & 0_n \end{bmatrix}^T$ is simply the vector a_k zero padded to the length of c_1 and will be noted x_p . Then the equation (21) will be rewritten as:

$$C \cdot x_p = \begin{bmatrix} Ra_k \\ Sa_k \end{bmatrix} \tag{24}$$

The product Ra_k can be computed efficiently using the following algorithm [16]:
- Compute $X_p = FFT(x_p)$.
- Compute $w = FFT(c_1)$.
- Compute the element wise vector-vector product $H = X_p \odot w$.
- Compute $z = IFFT(H)$.

The p first elements of the vector z constitute the product Ra_k. since the FFT algorithm can be done in $O(plogp)$ operations, the product Ra_k can be also obtained in $O(plogp)$ operations [16], [17]. Figure (3) shows the block diagram illustrating the fast matrix-vector multiplication.

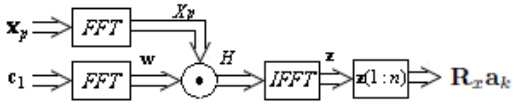

Fig. 3. Block diagram illustrating the fast matrix-vector multiplication

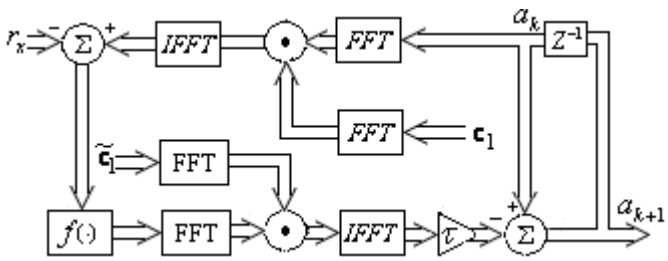

Fig. 4. Block diagrams of the CDGNN model

B. Proposed architecture of CDGNN

To compute the product $z_3 = R^T z_2$, we just replace R by R^T and a_k by z_2 . We note here that the matrix R^T is Toeplitz, then it can be generated only by its first row and first column. Furthermore the Teoplitz matrix R^T can be embedded into a circulant matrix C^T. Let $\tilde{c}_1$ be the first column of the matrix C^T and $\tilde{w}$ its Fourier transform. The block diagram realizing and the detailed architecture of the proposed neural network are shown in the figures (4) and (5) respectively. As we can see, the FFTs of the column c_1 and $\tilde{c}_1$ constitutes the connection weighting of the neural network. So we have just $4p$ weighting function instead of $2p^2$ in the original DGNN. The entire circuit contains 4 blocks FFT/IFFTs, $2p$ adders of 2 elements, p time-delays, and $4p$ weighted connections.

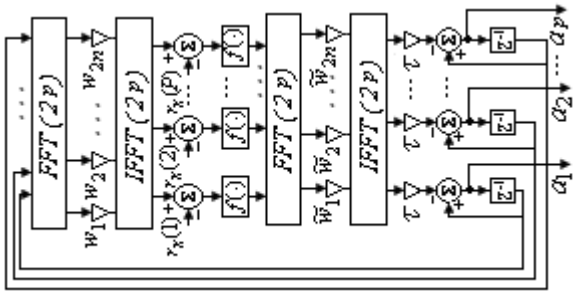

Fig. 5. Architecture of the proposed neural network

C. Complexity and Comparison

The complexity of a neural network is defined as the total number of multiplications per iteration. Since the FFT/IFFT algorithm using p points requires $0.5plogp$ multiplications, then it can be seen that the proposed neural network model requires $5p + 4plog2p$ multiplications per iteration instead of $2p^2 + p$ for the original DGNN model. As result, the computational complexity of the proposed neural network is reduced to $O(plogp)$ multiplications.
Concerning the memory storage, in addition to $O(plogp)$ memory required for the FFT/IFFT blocks, we need to store the $6n$ elements of the vectors c_1,$\tilde{c}_1$,b , and the outputs, thus only $O(plogp)$ elements need storage in the proposed CDGNN model instead of $O(p^2)$ elements in the original model.

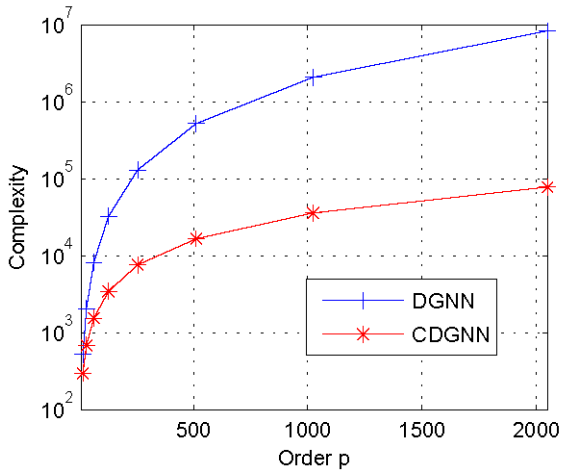

Fig. 6. Computational complexity of the two networks

V. Computer Simulations

Computer simulations have been performed to assess the performance of the proposed method in term of accuracy and computational complexity. In the first experiment,we will show the effectiveness of the CDGNN model for the AR parameters estimation. While in the second experiment,we will consider the estimation of a stationary process consisting of a mixture of two complex sinusoids corrupted by an additive zero-mean Gaussian noise.

A. AR parameters estimation

Let $x(n)$ an AR(4) process given by

$$x(n) = 2.0371x((n-1) - 2.4332x(n-2) + 1.7832x(n-3) - 0.7019x(n-4) + e(n) \quad (25)$$

where the input process $e(n)$ is a white Gaussian process with zero mean and variance σ_e^2. In this example, we let the number of process samples $N = 64$, and $h = 10^{-4}$ and $\gamma = 195000$. Figures (7)and (8) show the convergence behaviour of the original network and its simplified version. As we can see from figure (7), starting from a random initial state a_0 , the two networks converge in the same manner to the true parameters which implies the correctness of the proposed scheme. In the figure (8), we show the convergence error of the two networks, the residual error for the two networks is about 10^{-14} after convergence.

B. Spectrum estimation

The data used in this example consists of two sinusoids embedded in noise.

$$x(n) = \sum_{k=1}^{2} A_k \cos(2\pi f_k n) + e(n), n = 1, 2, \cdots, N \quad (26)$$

ISSN: 2692-5079

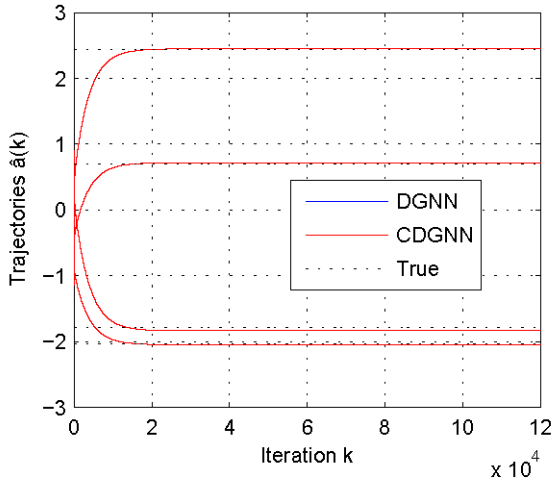

Fig. 7. Convergence behaviour of the state trajectory.

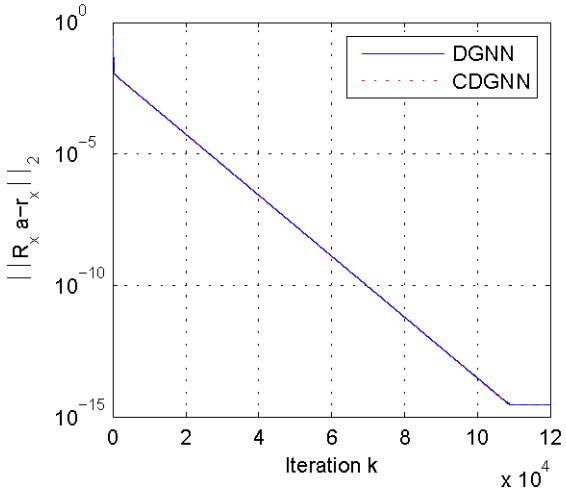

Fig. 8. Convergence error of the two neural networks.

The added noise $e(n)$ is white Gauusian. The amplitudes and frequencies of sinusoids A_1, A_2, $f_1$andf_2 were chosen as 1,1.5,0.1 and 0.3 respectively. The data samples we used in this simulation was 256. The neural network parameters were chosen as $h = 10^{-4}$,$\gamma = 500$, the model order $p = 8$ and the number of the FFT points is 512. Figure (9) shows the power spectral densities obtained by the use of the original method cited in [5] and the proposed one. As we can see, the two Methods lead to a similar results.

VI. Conclusion

Recurrent neural networks are very useful as computational models for solving computationally intensive problems due to their inherent nature of parallel and distributed information processing. The Toeplitz structure of the correlation matrix allows us to design a compact implementation of the gradient based neural network for solving Toeplitz systems using the fast Fourier transform to compute the neural network activation. The proposed reduced neural network is very suitable for estimating the AR parameters of models with large order.

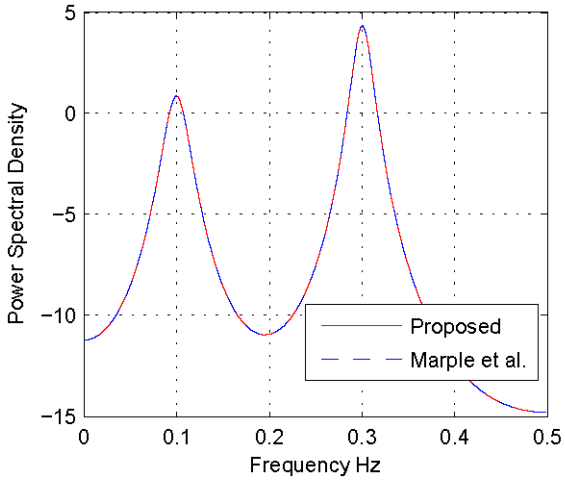

Fig. 9. Spectral density estimation using the proposed scheme

Computer simulations show that the proposed algorithm is very suitable for AR parameters and spectral density estimation.

References

[1] S. M. Kay, editors, *Modern Spectral Estimation: Theory ad Application*, NJ: Prentice-Hall,Englewood Cliffs, 1988.
[2] S.L. Marple, editors, *Digital Spectral Analysis with Applications*, NJ: Prentice-Hall,Englewood Cliffs, 1987.
[3] J. Capon, High-resolution frequency-wavenumber spectrum analysis, *Proc. IEEE*, 57, 1408–1418, 1969.
[4] B. R. Musicus, Fast MLM Power Spectrum Estimation from Uniformly Spaced Correlations, *IEEE Trans. Acoustics, Speech and Signal Processing* , 33(4), 1333-1335, 1985.
[5] S. L. Marple, M. Adeli and H. Liu, Super-Fast Algorithm for Minimum Variance (Capon) Spectral Estimation, *Conf. Signals, Systems and Computers* , 1832-1836, 2010.
[6] S.K. Park, Hopfield Neural Network for AR Spectral Estimator, *in Proc. IEEE* , 487-490, 1990.
[7] Y.Xia and M.S. Kamel, A Cooperative Recurrent Neural Network Algorithm For Parameter Estimation of Autoregressive Signals,*International Joint Conference on Neural Networks*,2516- 2522, 2006.
[8] - A. Benchabane, A. Bennia, F. Charif and A. Ahmed-Taleb, Multi-Dimensional Capon Spectral Estimation Using Zhang Neural Networks,*Multidimensional systems and signals processing*, 24,583-598, 2013.
[9] P.S.Stanimirović ,M. D.Petković,Gradient neural dynamics for solving matrix equations and their applications, *Neurocomputing*, 306, 200-212, 2018.
[10] P.S.Stanimirović ,M.D.Petković and D. Gerontitis ,Gradient neural network with nonlinear activation for computing inner inverses and the Drazin inverse, *Neural Process. Lett.* , 48, 109–133, 2018.
[11] P.S.Stanimirović ,M. D.Petković,Improved GNN Models for Constant Matrix Inversion, *Neural Process. Lett.*, doi.org/10.1007/s11063-019-10025-9, 2019.
[12] Y. Zhang, K. Chen and W. Ma, MATLAB simulation and comparison of Zhang neural network and gradient neural network for online solution of linear time-varying equations,*proceedings of International Conference on Life System Modeling and Simulation*, 450-454, 2007.
[13] C. Yi and Y. Zhang, Analogue recurrent neural network for linear algebraic equation solving, *Electronics Letters*, 44(18), 1078-1079, 2008.
[14] L.Xia, W.Meng,R.Lu and L. Ding, Advances in Neural Networks, *Springer International Publishing Switzerland* ,444-451, 2015
[15] Y. Zhang, Z.H Chen and K. Chen, Convergence properties aalysis of gradient neural network for solving online liear equations, *Acta Automatica Sinica*, 35(8), 1136-1139,2009.
[16] V. Pan, editors, *Structured matrices and polynomials: Unified superfast algorithms*, Birkhauser, Boston ,2001.

ISSN: 2692-5079

[17] T. Kailath and A. H. Sayed, editors, *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, PA, 1999.

Abderrazak Benchabane received the B.Sc. degree in automatics from Annaba University, Algeria, in 1993, the M.Sc. degree in electronics from the University of Constantine, Algeria, in 2003. and the Ph.D. degree on the spectral estimation using neural networks at the University of Constantine in 2015. His research interests include signal processing and neural networks.

Fella Charif received the B.Sc. , M.Sc. and the Ph.D. degrees in electronics from the University of Biskra, Algeria, in 1994, 2006, and 2015 respectively. Her current research interests include signal and image and neural networks.