

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Neural Network Principles and Applications

Amer Zayegh and Nizar Al Bassam

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.80416>

Abstract

Due to the recent trend of intelligent systems and their ability to adapt with varying conditions, deep learning becomes very attractive for many researchers. In general, neural network is used to implement different stages of processing systems based on learning algorithms by controlling their weights and biases. This chapter introduces the neural network concepts, with a description of major elements consisting of the network. It also describes different types of learning algorithms and activation functions with the examples. These concepts are detailed in standard applications. The chapter will be useful for undergraduate students and even for postgraduate students who have simple background on neural networks.

Keywords: neural network, neuron, digital signal processing, training, supervised learning, unsupervised learning, classification, time series

1. Introduction

The artificial neural network is a computing technique designed to simulate the human brain's method in problem-solving. In 1943, McCulloch, a neurobiologist, and Pitts, a statistician, published a seminal paper titled "A logical calculus of ideas immanent in nervous activity" in *Bulletin of Mathematical Biophysics* [1], where they explained the way how brain works and how simple processing units—neurons—work together in parallel to make a decision based on the input signals.

The similarity between artificial neural networks and the human brain is that both acquire the skills in processing data and finding solutions through training [1].

2. Neural network's architecture

To illustrate the structure of the artificial neural network, an anatomical and functional look must be taken on the human brain first.

The human brain consists of about 10^{11} computing units "neurons" working in parallel and exchanging information through their connectors "synapses"; these neurons sum up all information coming into them, and if the result is higher than the given potential called action potential, they send a pulse via axon to the next stage. Human neuron anatomy is shown in **Figure 1** [2].

In the same way, artificial neural network consists of simple computing units "artificial neurons," and each unit is connected to the other units via weight connectors; then, these units calculate the weighted sum of the coming inputs and find out the output using squashing function or activation function. **Figure 2** shows the block diagram of artificial neuron.

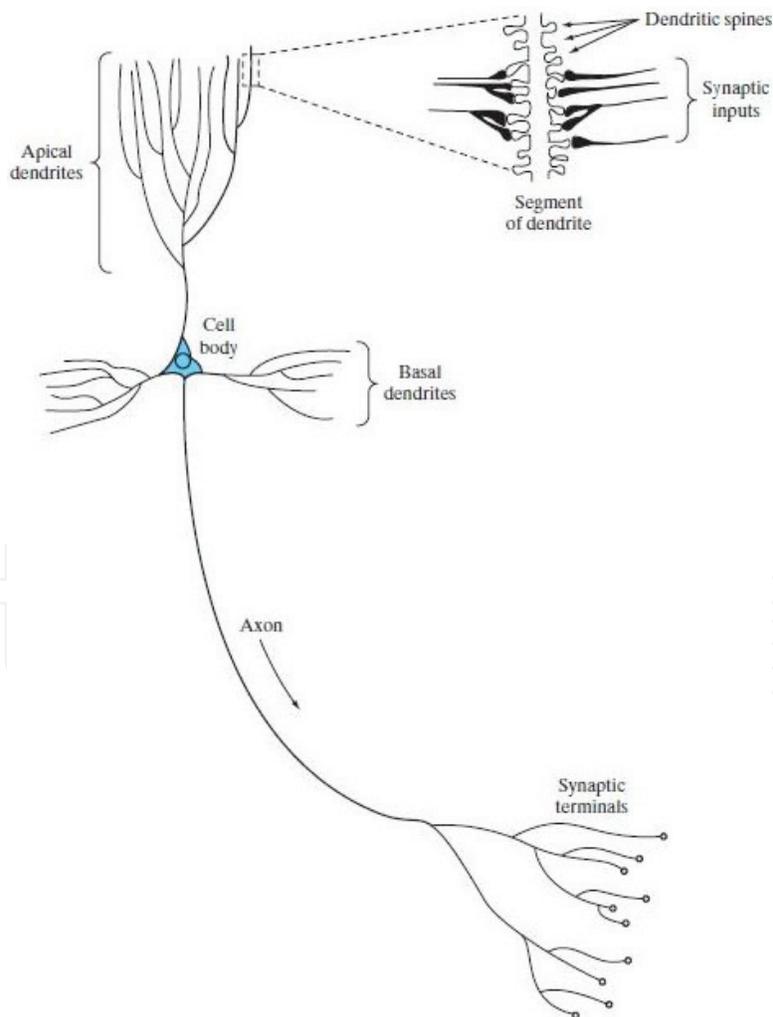


Figure 1. Human neuron anatomy.

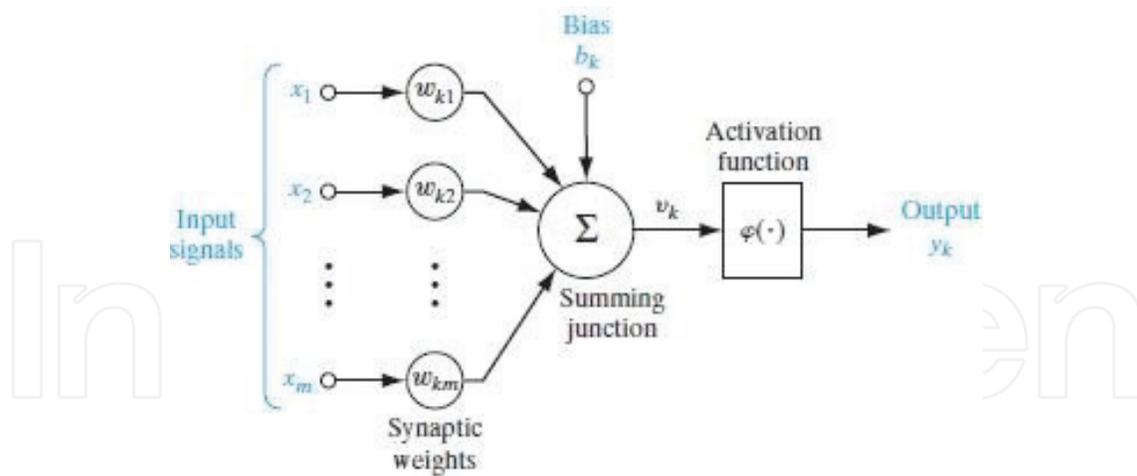


Figure 2. Block diagram of artificial neuron.

Based on the block diagram and function of the neural network, three basic elements of neural model can be identified:

1. Synapses, or connecting links, have a weight or strength where the input signal x_i connected to neuron k is multiplied by synaptic weight w_{ki} .
2. An adder for summing the weighted inputs.
3. An activation function to produce the output of a neuron. It is also referred to as a squashing function, in that it squashes (limits) the amplitude range of the output signal to a finite value.

The bias b_k has the effect of increasing or decreasing the net input of the activation function, depending on whether it is positive or negative, respectively.

Mathematically, the output on the neuron k can be described as

$$y_k = \varphi \left(\sum_{i=1}^m x_i \cdot w_{ki} + b_k \right) \quad (1)$$

where

$x_1, x_2, x_3, \dots, x_m$ are the input's signals.

$w_{k1}, w_{k2}, w_{k3}, \dots, w_{km}$ are the respective weights of neuron.

b_k is the bias.

φ is the activation function.

To clarify the effect of the bias on the performance of the neuron, the output given in Eq. (1) is processed in two stages, where the first stage includes the weighted inputs and the sum which is denoted as S_k :

$$S_k = \sum_{i=1}^m x_i \cdot w_{ki} \tag{2}$$

Then, the output of adder will be given in Eq. (3):

$$v_k = S_k + b_k \tag{3}$$

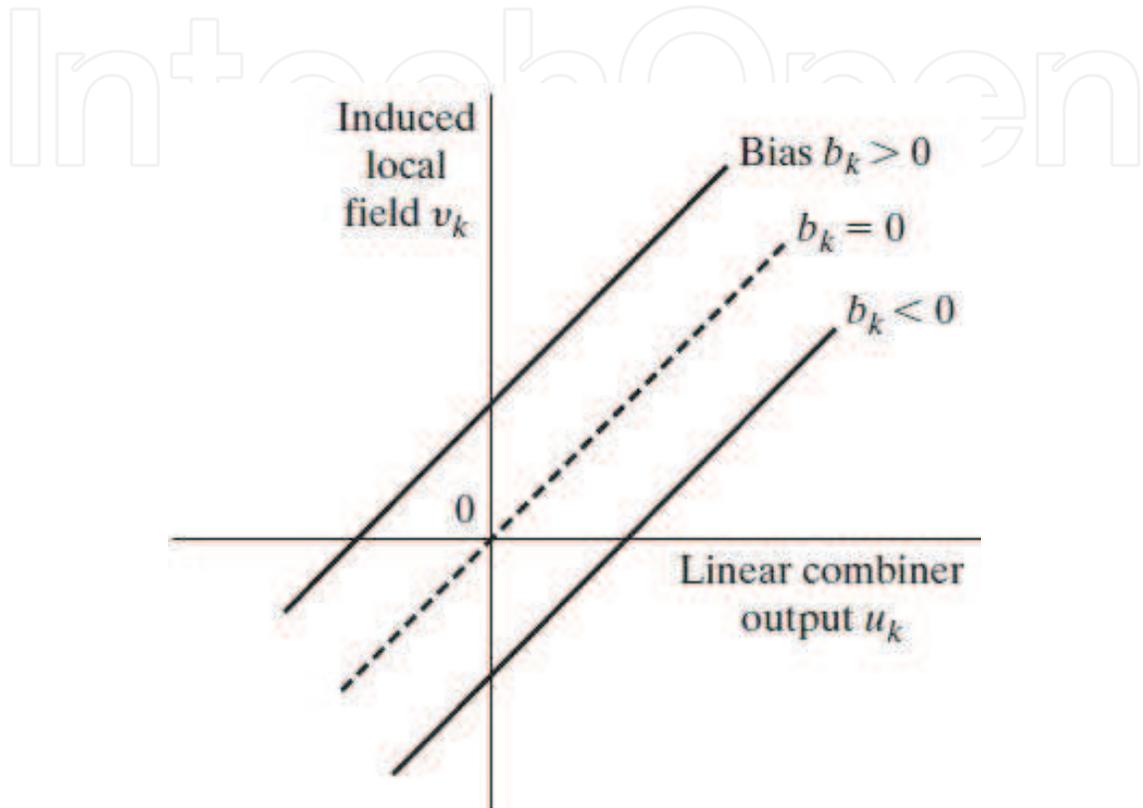


Figure 3. Effect of bias.

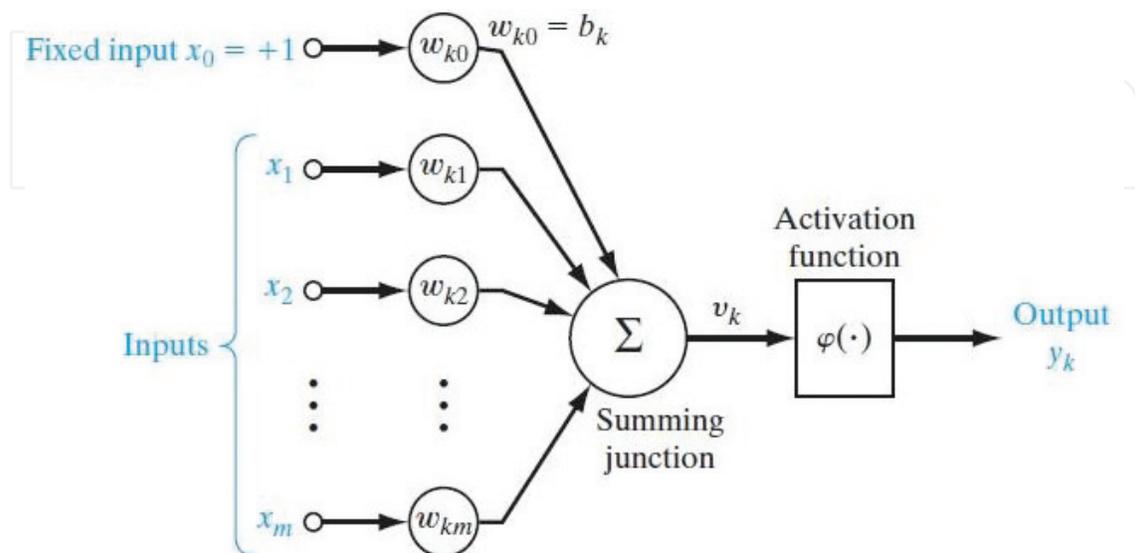


Figure 4. Neuron structure with considering bias as input [1].

where the output of neuron will be

$$y_k = \varphi(v_k) \quad (4)$$

Depending on the value of the bias, the relationship between the weighted input and adder output will be modified [3] as shown in **Figure 3**.

Bias could be considered as an input signal x_0 fixed at +1 with synaptic weight equal to the bias b_k as shown in **Figure 4** [3].

3. Types of activation function

Activation function defines the output of neuron as the function to the adder's output v_k . The following sections describe the different activation functions:

3.1. Linear function

Where neuron output is proportional to the input as shown in **Figure 5**.

And, it can be described by

$$y_k = v_k \quad (5)$$

3.2. Threshold (step) function

This activation function is described in **Figure 6** where the output of neuron is given by

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases} \quad (6)$$

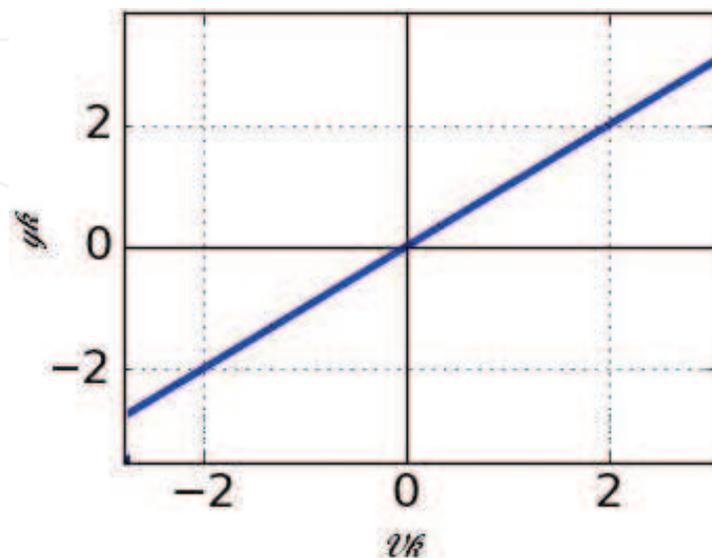


Figure 5. Linear activation function.

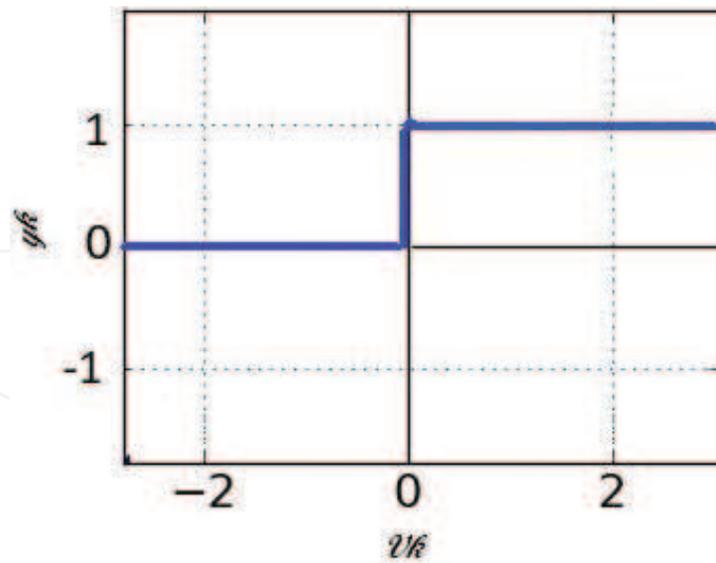


Figure 6. Threshold activation function.

In neural computation, such a neuron is referred to as the *McCulloch-Pitts model* in recognition of the pioneering work done by McCulloch and Pitts (1943); the output of the neuron takes on the value of 1 if the induced local field of that neuron is nonnegative and 0 otherwise. This statement describes the *all-or-none property* of the McCulloch-Pitts model [4].

3.3. Sigmoid function

The most common type of activation functions in neural network is described by

$$y_k = \frac{1}{1 + e^{-v_k}} \quad (7)$$

Figure 7 shows the sigmoid activation function, it is clearly observed that this function has nonlinear nature and it can produce analogue output unlike threshold functions which produce output in discrete range $[0, 1]$.

Also, we can note that sigmoid activation function is limited between 0 and 1 and gives an advantage over linear activation function which produces output form $-\infty$ to $+\infty$ [5].

3.4. Tanh activation function

This activation function has the advantages of sigmoid function, while it is characterized by output range between -1 and 1 as shown in **Figure 8**.

The output is described by

$$y_k = \frac{2}{1 + e^{-2v_k}} - 1 \quad (8)$$

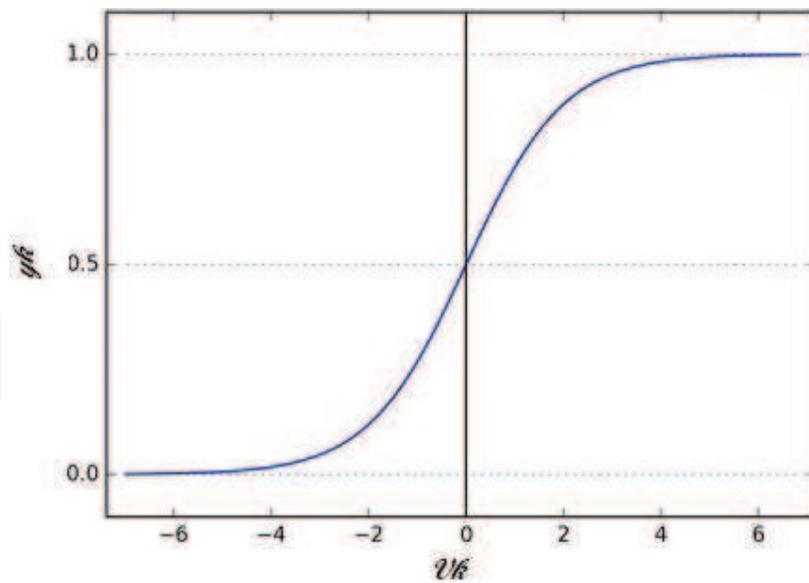


Figure 7. Sigmoid activation function.

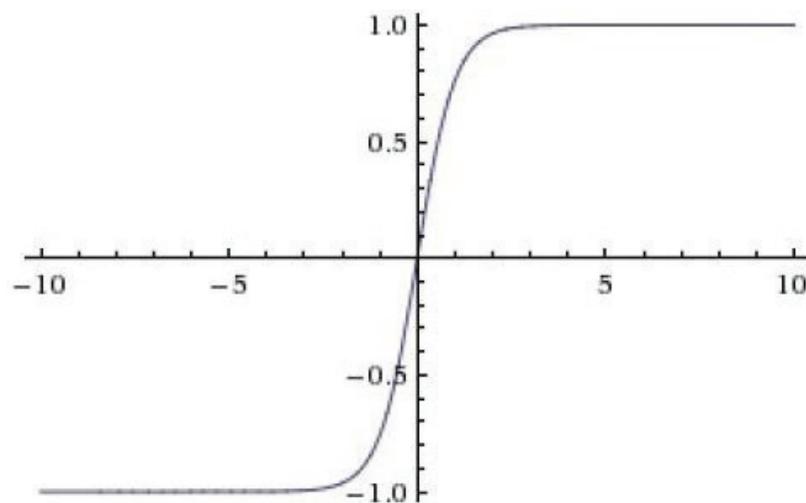


Figure 8. Tanh activation function.

4. Neural network models

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network [1]. Three main models can be identified for the neural network.

4.1. Single-layer feedforward neural network

In a layered neural network, the neurons are organized in the form of layers [1]. The simplest structure is the single-layer feedforward network that consists of input nodes connected

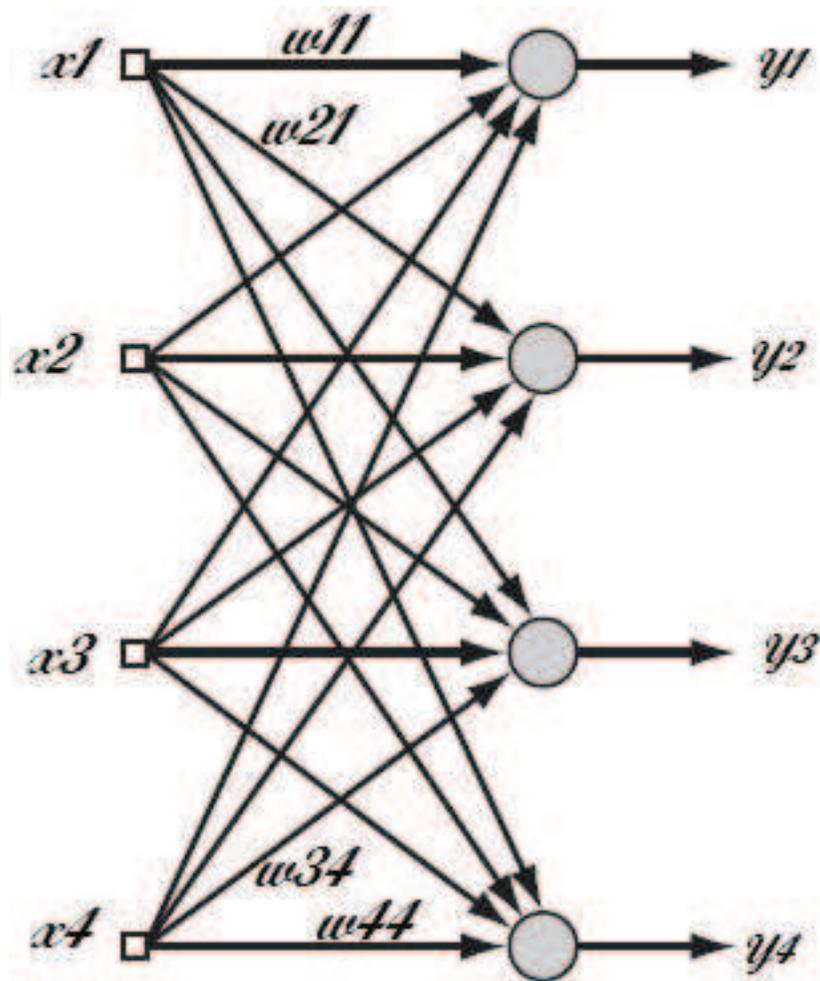


Figure 9. Single-layer neural network.

directly to the single layer of neurons. The node outputs are based on the activation function as shown in Figure 9.

Mathematically, the inputs will be presented as vectors with dimensions of $1 \times i$, while the weights will be presented as a matrix with dimensions of $i \times k$, and outputs will be presented as a vector with dimensions of $1 \times k$ as given in Eq. (9):

$$[y_1, y_2, \dots, y_k] = [x_1, x_2, \dots, x_i] \begin{bmatrix} w_{11} & w_{21} & \dots & w_{k1} \\ \vdots & \ddots & \ddots & \vdots \\ w_{1k} & w_{2k} & \dots & w_{ik} \end{bmatrix} \quad (9)$$

4.2. Multilayer feedforward neural network

The second class of a feedforward neural network distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons as shown in Figure 10.

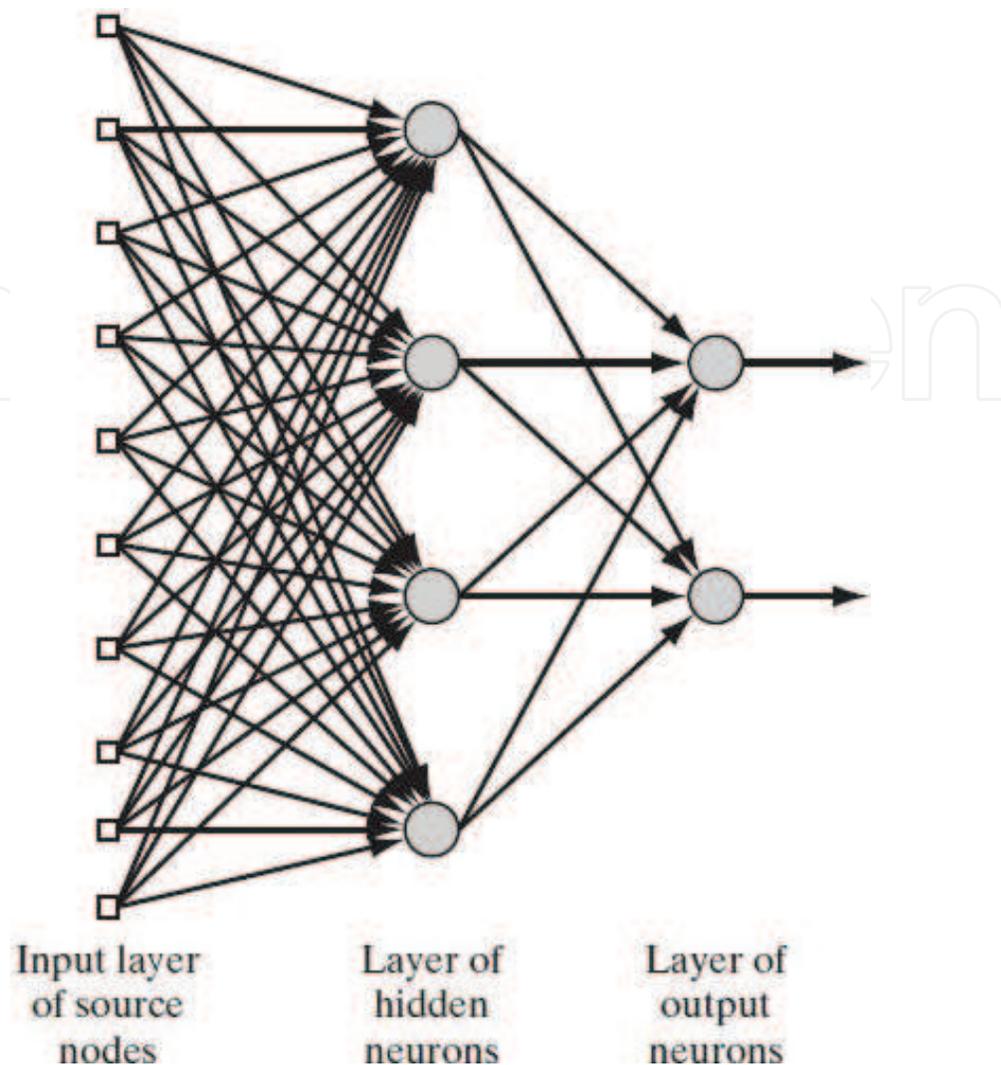


Figure 10. Multilayer feedforward neural network.

By adding one or more hidden layers, the network is enabled to extract higher-order statistics from its input [1].

5. Neural network training

The process of calibrating the values of weights and biases of the network is called training of neural network to perform the desired function correctly [2].

Learning methods or algorithms can be classified into:

5.1. Supervised learning

In supervised learning, the data will be presented in a form of couples (input, desired output), and then the learning algorithm will adapt the weights and biases depending on the error signal between the real output of network and the desired output as shown in **Figure 11**.

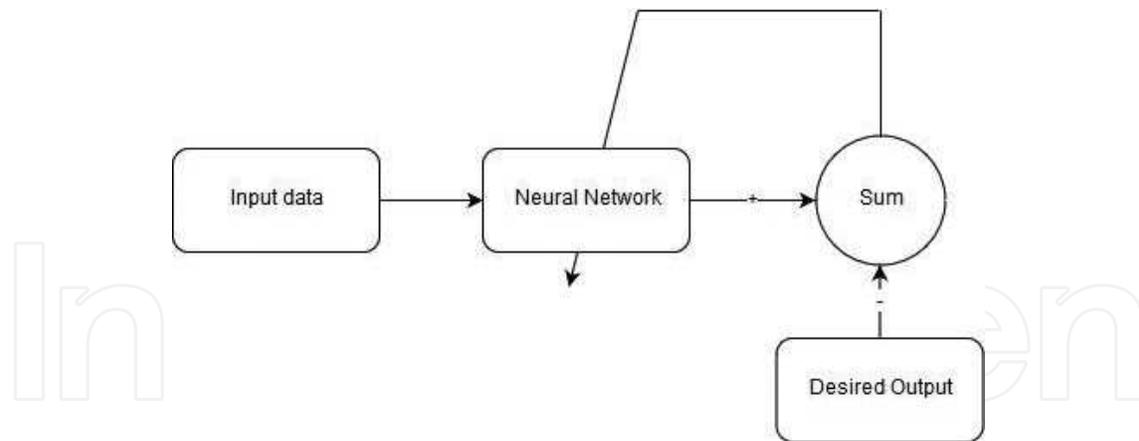


Figure 11. Supervised learning.

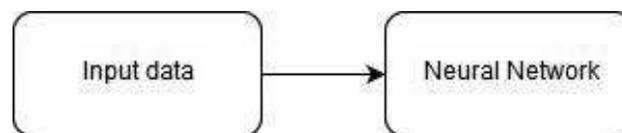


Figure 12. Unsupervised learning.

As a performance measure for the system, we may think in terms of the mean squared error or the sum of squared errors over the training sample defined as a function of the free parameters (i.e., synaptic weights) of the system [1].

5.2. Unsupervised learning

To perform unsupervised learning, a competitive learning rule is used. For example, we may use a neural network that consists of two layers—an input layer and a competitive layer. The input layer receives the available data. The competitive layer consists of neurons that compete with each other (in accordance with a learning rule) for the “opportunity” to respond to features contained in the input data (Figure 12) [1].

6. Neural networks’ applications in digital signal processing

Digital signal processing could be defined using field of interest statement of the IEEE Signal Processing Society as follows:

Signal processing is the enabling technology for the generation, transformation, extraction, and interpretation of information. It comprises the theory, algorithms with associated architectures and implementations, and applications related to processing information contained in many different formats broadly designated as signals. Signal processing uses mathematical, statistical,

computational, heuristic, and/or linguistic representations, formalisms, modelling techniques and algorithms for generating, transforming, transmitting, and learning from signals. [6].

Based on this definition, many neural network structures could be developed to achieve the different processes mentioned in the definition.

6.1. Classification

One of the most important applications of an artificial neural network is classification, which can be used in different digital signal processing applications such as speech recognition, signal separation, and handwriting recognition and detection [7].

The objects of interest can be classified according to their features, and classification process could be considered as probability process, since the classification of any object under a given class depends on the likelihood that the object belongs to the class more than the probability of belonging to the other classes [8].

Assume that X is the vector of features for the objects of interest which could be classified into classes $c \in \psi$ where ψ is the pool of classes. Then, classification will be applied as follows:

$$X \text{ belongs to the class } c_i \text{ if } P(c_i|X) > P(C_j|X) \text{ when } i \neq j \quad (10)$$

To decrease the difficulty of solving probability equations in Eq. (10), discriminant function is used, and then Eq. (10) will be.

$$Q_i(X) > Q_j(X) \quad \text{if } (c_i|X) > P(C_j|X) \text{ when } i \neq j \quad (11)$$

Classification process will be described using Eq. (12)

$$X \text{ belongs to the class } c_i \text{ if } Q_i(X) > Q_j(X) \quad (12)$$

One of the examples of classification is QPSK modulator output detection, where detection is considered as a special case of classification.

Assume that the received signal is X :

$$X = s + n \quad (13)$$

where n is normally the distributed noise signal and s is the transmitted signal.

The output of QPSK modulator is shown in **Figure 13**, where the samples are arranged in four classes.

By adding white Gaussian noise, the received signal will be as shown in **Figure 14**.

The neural network shown in **Figure 15** is used to detect and demodulate the received signal, where the network consists of one hidden layer with five neurons and an output layer with two neurons.

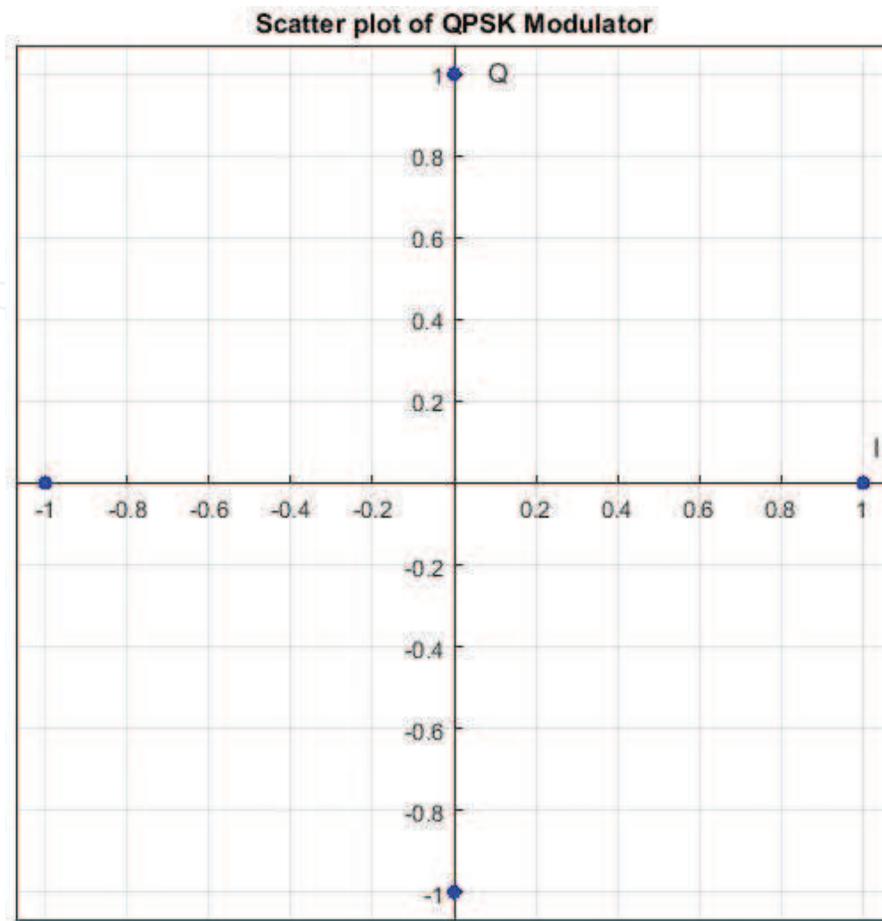


Figure 13. QPSK modulator output.

Figures 16 and 17 show the performance of neural network evaluated using mean squared error (MSE) criteria.

6.2. Time series prediction

A series is a sequence of values as a function of parameter; in the case of time series, the values will be as a function of the time. So, many applications use time series to express their data, for example, metrology, where the temperature is described as time series [7].

The interesting problem in time series is the future prediction of the series values; neural networks can be used to predict the future results in series in three ways [9]:

- Predict the future values based on the past values of the same series; this way can be described by

$$\hat{y}(t) = E\{y(t)|y(t-1), y(t-2), \dots\} \quad (14)$$

- Predict the future values based on the values of relevant time series, where

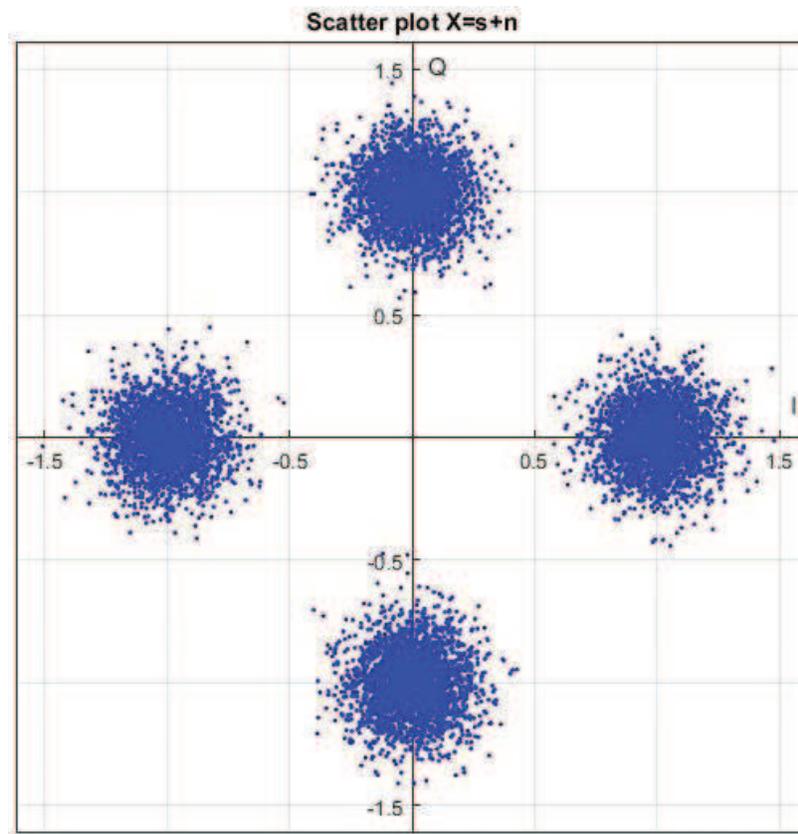


Figure 14. QPSK output with noise.

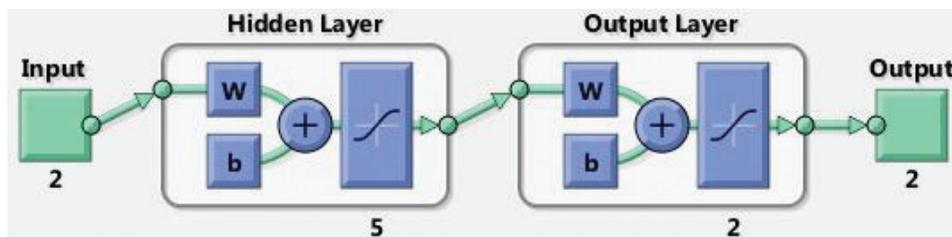


Figure 15. Structure of neural network.

$$\hat{y}(t) = E\{y(t)|x(t), x(t-1), x(t-2), \dots\} \quad (15)$$

- Predict the future values based on both previous cases, where

$$\hat{y}(t) = E\{y(t)|x(t), x(t-1), x(T-2), \dots, y(t-1), y(t-2), \dots\} \quad (16)$$

Figure 18 shows the predicted series by neural network based on the first way, where the samples of original series are given of determined period, and then the neural network predicts the future values of the series based on series behavior.

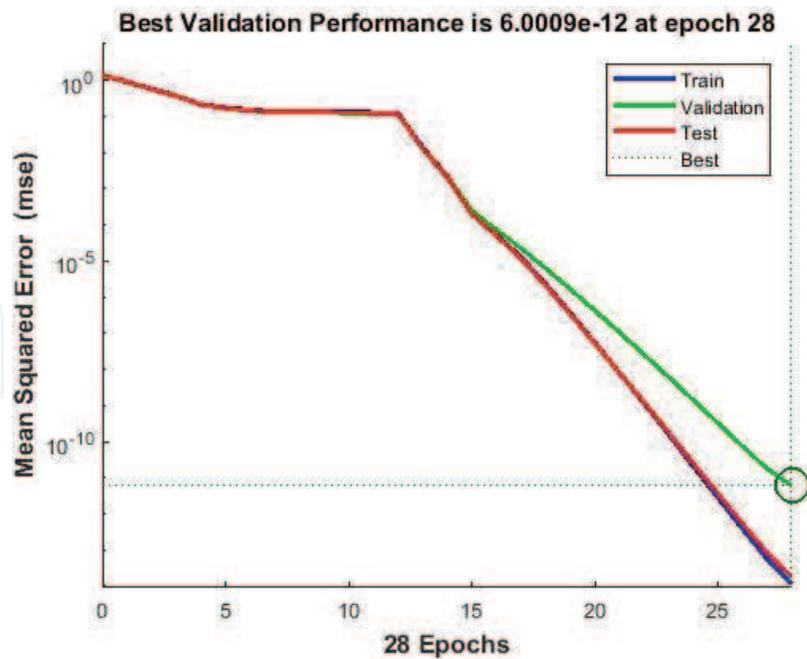


Figure 16. MSE of training, validation, and test vs no. of epochs.

6.3. Independent component analysis

The goal of the independent component analysis (ICA) is to separate the linearly mixed signals. ICA is a type of blind source separation when the separation is performed without the pre-information about the source of signals or the signal-mixing coefficients. Although the problem of separating the blind source, in general, is not specified, the solution of use can be obtained under some assumptions [10].

ICA model assumes that n independent signals $s_i(t)$ where $i = 1, 2, 3, \dots, n$ are mixed using matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (17)$$

Then, mixed signal $x_i(t)$ could be expressed as

$$x_i(t) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} s_j(t) \quad (18)$$

As the separation process is blind, that is, both a_{ij} and $s_j(t)$ are unknown; thus, ICA assumes that the mixed signals are statistically independent and have non-Gaussian distribution [11].

Neural network shown in **Figure 19** is used to estimate the unmixing matrix W .

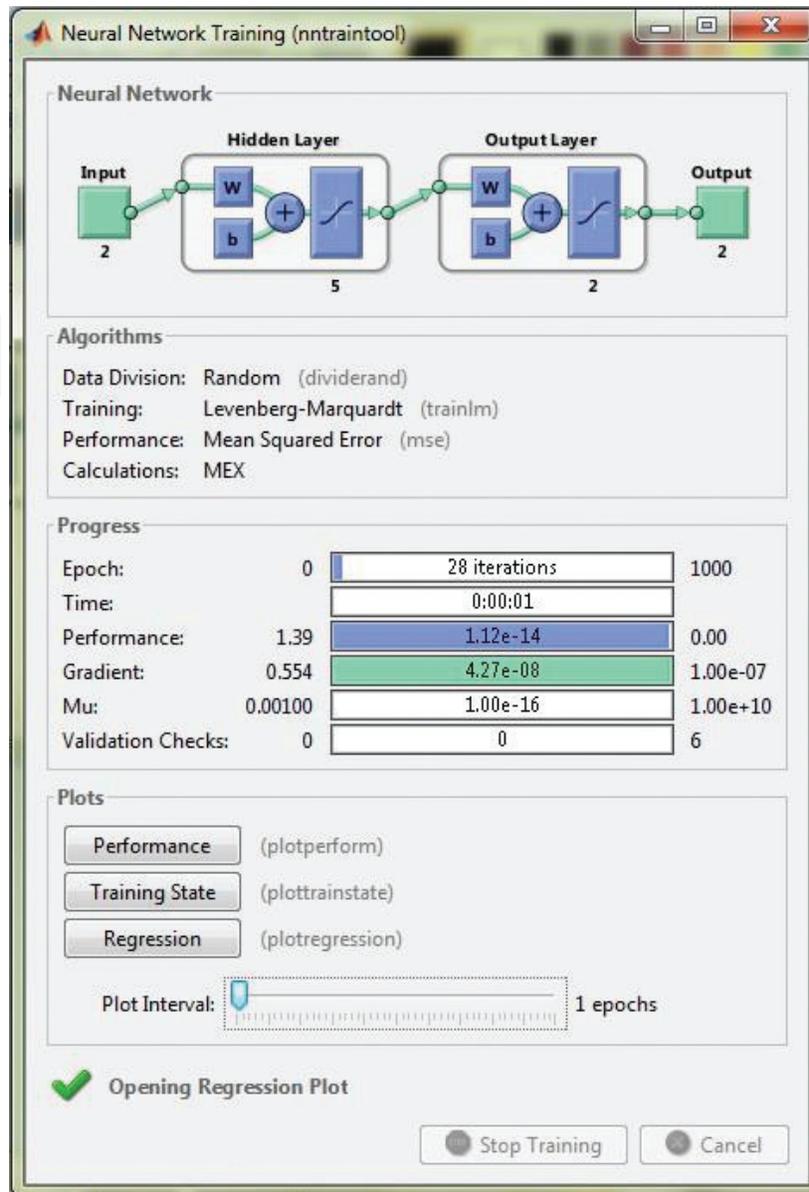


Figure 17. Training parameters and results.

The separated signals $y_i(t)$ are given as

$$y = Wx \tag{19}$$

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & \ddots \end{bmatrix} \tag{20}$$

Different methods could be applied to find W , for example, natural gradient based define W as

$$\frac{dW}{dt} = \eta(t)[1 - f(y(t))g^T(y(t))]W \tag{21}$$

where $\eta(t)$ is the training factor and both f and g are the odd functions.

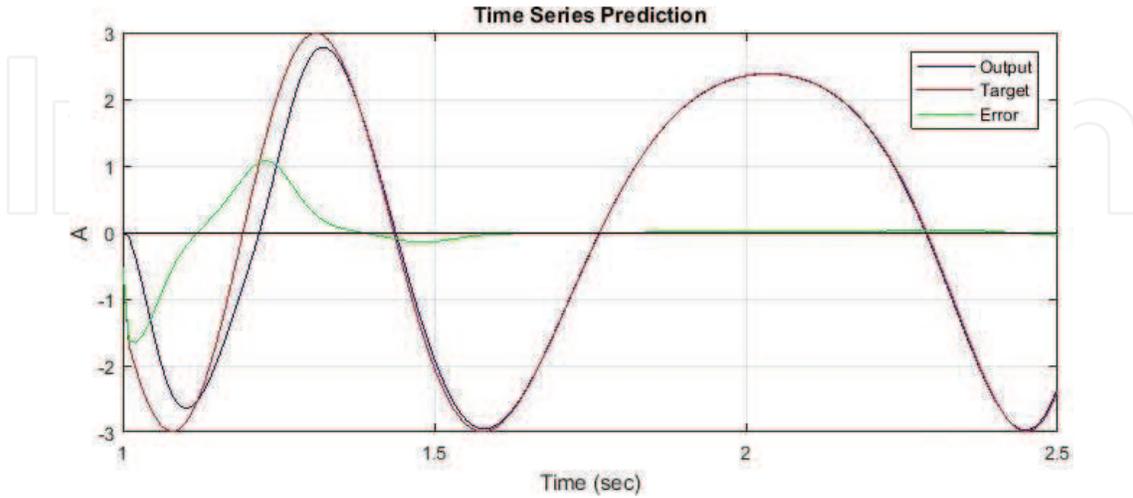


Figure 18. Time series prediction.

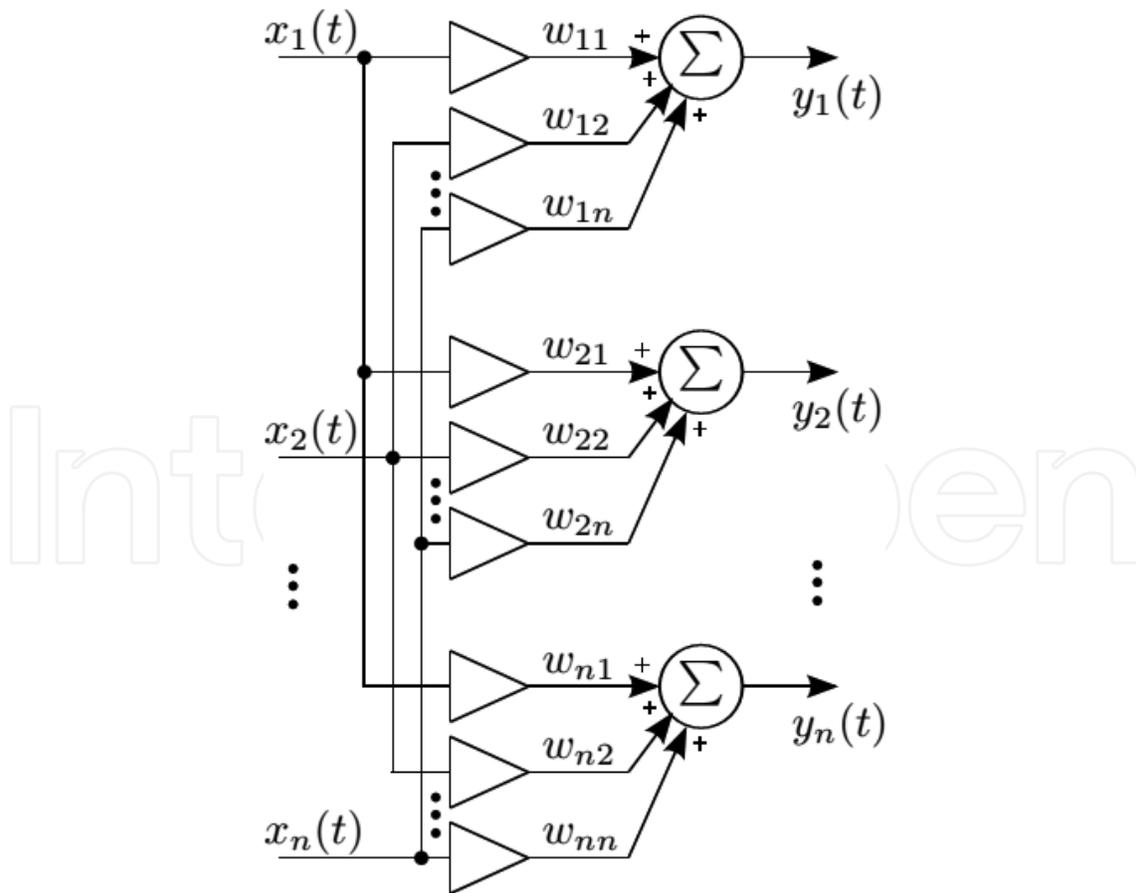


Figure 19. ICA neural network [12].

Author details

Amer Zayegh and Nizar Al Bassam*

*Address all correspondence to: nizar@mec.edu.om

Middle East College, Muscat, Oman

References

- [1] Haykin S. *Neural Networks and Learning Machines*. 3rd ed. Hamilton, Ontario, Canada: Pearson Education, Inc.; 2009. 938 p
- [2] Smith S. *The Scientist and Engineer's Guide to Digital Signal Processing*. 2011
- [3] Hu Y, Hwang J. *Handbook of Neural Network Signal Processing*. Boca Raton: CRC Press; 2002
- [4] Milad MAMRAN. Neural Network Demodulator For. *International Journal of Advanced Studies*. 2016;5(7):10-14
- [5] Jan Michalík. *Applied Neural Networks for Digital Signal Processing with DSC TMS320 F28335*. Technical University of Ostrava; 2009
- [6] Constitution. IEEE Signal Processing Society [Online]. 2018. Available from: <https://signal-processing.org/volunteers/constitution> [Accessed: March 03, 2018]
- [7] Kriesel D. *A Brief Introduction to Neural Network*. Bonn: University of Bonn in Germany; 2005
- [8] Gurney K. *An Introduction to Neural Networks*. London: CRC Press; 1997
- [9] Maier H, Dandy G. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environmental Modelling & Software*. 2000;15(1):101-124
- [10] Hansen LK, Larsen J, Kolenda T. On Independent Component Analysis for Multimedia Signals, Multimedia Image and Video Processing. CRC Press; 2000. pp. 175-199
- [11] Mørup M, Schmidt MN. Transformation invariant sparse coding, Machine Learning for Signal Processing, IEEE International Workshop on (MLSP), Informatics and Mathematical Modelling. Technical University of Denmark, DTU; 2011
- [12] Pedersen MS, Wang D, Larsen J, Kjems U. Separating Underdetermined Convolutional Speech Mixtures, ICA2006. 2006

