Int. J. Communications, Network and System Sciences, 2015, 8, 374-398
Published Online September 2015 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2015.89036

Biological Inspiration—Theoretical Framework Mitosis Artificial Neural Networks Unsupervised Algorithm

Lácides Pinto Mindiola, Gelvis Melo Freile, Carlos Socarras Bertiz

Universidad de La Guajira, Riohacha, Colombia
Email: lpintom@uniguajira.edu.co, gelvismanuel@uniguajira.edu.co, csocarras@uniguajira.edu.co

Received 11 July 2015; accepted 27 September 2015; published 30 September 2015

Abstract

The modified approach to conventional Artificial Neural Networks (ANN) described in this paper represents an essential departure from the conventional techniques of structural analysis. It has four main distinguishing features: 1) it introduces a new simulation algorithm based on the biology; 2) it performs relatively simple arithmetic as massively parallel, during analysis of a structure; 3) it shows that it is possible to use the application of the modified approach to conventional ANN to solve problems of any complexity in the field of structural analysis; 4) the Neural Topologies for Structural Analysis (NTSA) system are recurrent networks and its outputs are connected to its inputs [1] and [2]. In NTSA system the DNA of the neuron mother and daughters would be defined by: 1) the same entry, from the corresponding neuron in the previous layer; 2) the same trend vector; 3) the same transfer function (purelin). The mother's neuron and her daughter's neuron differ only in the connection weight and its output signal.

Keywords

Mitosis, Artificial Neuron, Node, Structural Analysis, Neural Networks, Output, Layer, Simulation

1. Introduction

The ADALINE network is a fairly well known ANN and is very similar to the perceptron except that its transfer function is linear. Since its invention by Bernard Widrow and his graduate student Marcian Hoff in 1960, we have considered a lineal ANN of several layers with simple processors also called "neuron", "node", and "processing elements". Each layer has its own weight matrix $\boldsymbol{W}$, its own bias vector $\boldsymbol{b}$ (the bias is much like a weight except that it has a constant input of 1), a net input vector $\boldsymbol{n}$ and a output vector $\boldsymbol{a}$ $\left(\boldsymbol{a} = purelin\left(\boldsymbol{wp}+\boldsymbol{b}\right)\right)$. Note

How to cite this paper: Mindiola, L.P., Freile, G.M. and Bertiz, C.S. (2015) Biological Inspiration—Theoretical Framework Mitosis Artificial Neural Networks Unsupervised Algorithm. *Int. J. Communications, Network and System Sciences*, **8**, 374-398. http://dx.doi.org/10.4236/ijcns.2015.89036

that the scalar input p is multiplied by the scalar weight w to form wp.

A NTSA is an information processing system which operates on inputs to extract information, and produces outputs corresponding to the extracted information. NTSA means that the "distribution factors" in the nodes of the structures are the components of the input vector to the first layer. After a complete presentation of the training data, a new set of weights and biases are obtained, and new outputs are again evaluated in a feed-forward manner until a specific tolerance for error is obtained. Unsupervised training uses unlabeled training data and requires no external teaching.

A NTSA model is composed of simple processors, each having a local memory. Processing elements are connected by unidirectional links that carry discriminating data; the linear feed-forward net has been found to be a suitable one for training techniques. Outputs of neurons in one layer are transferred to their corresponding neuron in another layer through a link that amplifies or inhibits such outputs through weighting factors [3].

This paper evaluates a neural network approach on frame analysis using an unsupervised algorithm. The results are obtained programming the entire formulation of the algorithm using MATLAB. The aim of the study is to estimate the rotational end moment.

The modified neuronal structural analysis requires the addition of a set of new concepts but simple addition of traditional, also called classic. All of them eventually make themselves the nature of a new approach, whose theoretical and conceptual frameworks then present them in a concise form, but as clear as possible, so that it is accessible to all and everyone disposing of the basic conceptual tools tries to address that this new theoretical method is practical.

Artificial Neural Networks (**ANN**) is able to perform relatively simple arithmetic as massively parallel, during analysis of a structure. To this end, an urgent need is to define the terms: matrices and vectors weights trends, using the parameters derived from the physical and mechanical members.

We do not intend here to discuss the basic principles of analysis of statically indeterminate structures, nor the plausibility of traditional approaches. We simply want to show that it is possible to use the application of the modified approach or conventional artificial neural networks to solve problems of any complexity in the field of structural analysis.

Electronic neural networks generally exist as computer simulations. And that they are typically designed as very large. The ability to simulate is limited by the speed and storage capacity of digital computers available. Some researchers have developed hardware in order to increase processing speed, but conditions are not given to facilitate this event parallel [4]. However, there are two key similarities between biological and artificial neural networks. Firstly, the building blocks of the two kinds of networks are simple computational devices, although artificial neurons are much simpler, so that biological neurons are highly interconnected. Secondly, the connection among the processing elements determines the function of the network. The main objective of this paper is to determine the application of the modified approach or conventional ANN to solve problems of any complexity in the field of structural analysis. The work elaborated the building of the NTSA models as well as its features with the appropriate connections to solve particular problems [1].

Notation and Terminology

In this work, figures, mathematical equations and text discussing both figures and mathematical equations will use the following notation.

Basic Concepts

Scalars: Small italic letters a, b, c

Vectors: Small bold nonitalic letters ***a*, *b*, *c***

Matrices: Capital bold nonitalic letters ***A*, *B*, *C***

A. Structures

We append the number of the layer as a superscript to the rotational moments. Thus, the rotational for the first layer and node four on the right end of the beam is written as $M^1 4_{vd}$, and the rotational moment for the second layer and node five on the bottom column is written as $M^2 5_{ci}$. **Figure 1** and **Figure 2** show this notation.

End Moments

End Rotational Moments

$M^L n_{sub}$

L—layer, n—node, sub—viz, vd, cs, ci

Viz—left beam

Vd—right beam

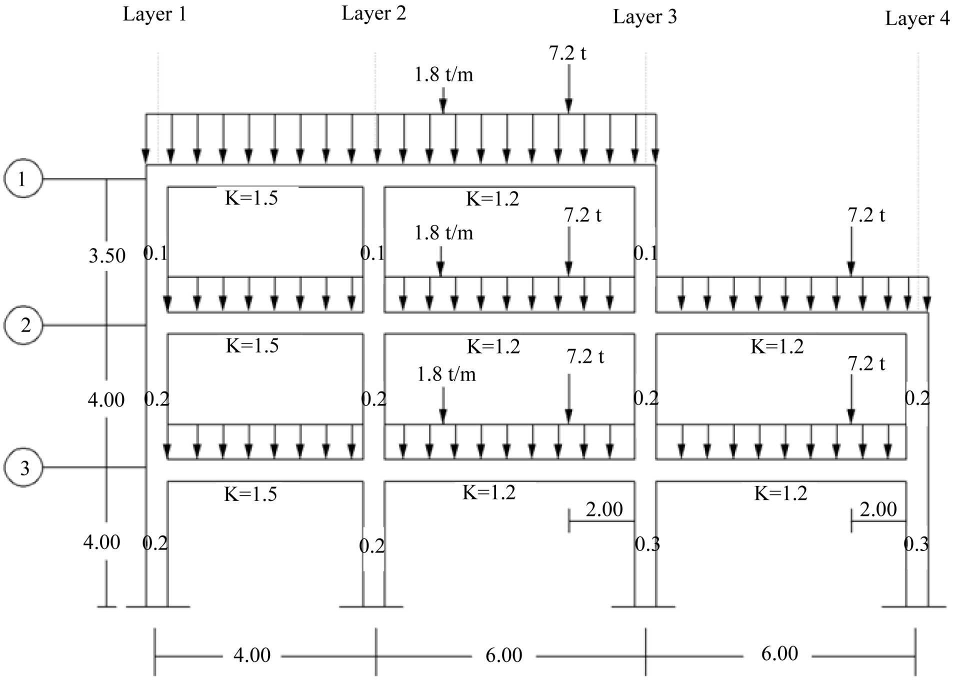


Figure 1. Primitive structure.

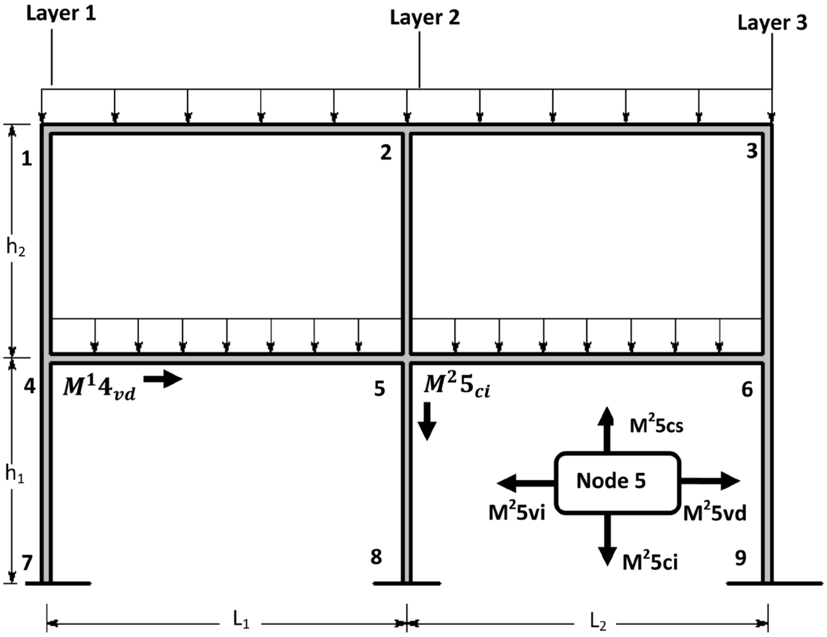


Figure 2. Structure—notation.

Cs—top column

Ci—bottom column

Horizontal Displacements

$M^{L}\Delta H_{i,j}$

ΔH —horizontal displacement, *i*—top column, *j*—*bottom* column

Moments of Perfect Embedding

$ME^{L}_{i,j}$

$ME^{L}_{j,i}$

i, j—Extremes

Sum of the Moments of Perfect Embedding

$\sum_{(n)} ME$

B. Neural Networks Model

We need to introduce some additional notation concerning the network architectures. See **Figure 3** and **Figure 4**.

Weight Matrices:

Scalar Element

$\boldsymbol{w}^{L}_{i,j}$

i—row, j—column, L—layer

Matrix

$\boldsymbol{W}^{L}$

Column Vector

$\boldsymbol{w}^{L}_{j}$

Row Vector

${}_{i}\boldsymbol{w}^{L}$

Bias Vector:

Scalar Element

b^{L}_{i}

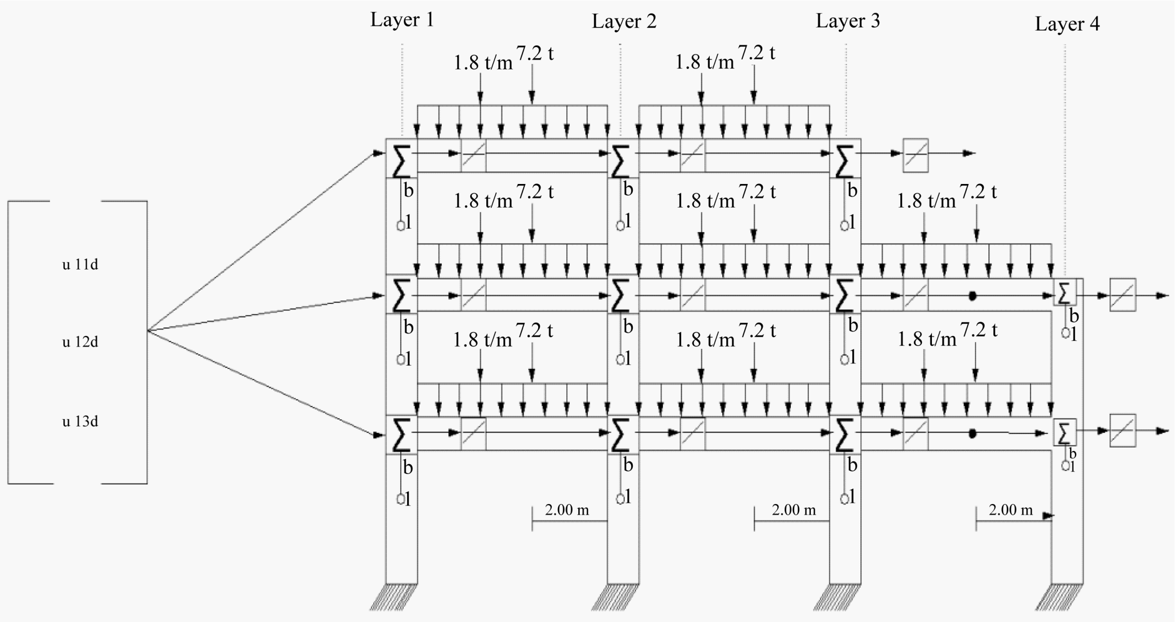


Figure 3. Biological inspiration NTSA system primitive.

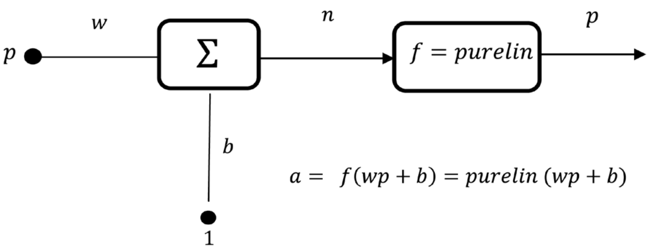


Figure 4. Single-input neuron.

Vector

$\boldsymbol{b}^L$

Input Vector:

Scalar Element

p_i

Net Input Vector:

Scalar Element

n_i^L

Vector

$\boldsymbol{n}^L$

Output Vector:

Scalar Element

a_i^L

Vector

$\boldsymbol{a}^L$

Transfer Function

Scalar Element

a_i^L

Vector

$\boldsymbol{a}^L = f^L\left(\boldsymbol{n}^L\right) = purelin^L\left(\boldsymbol{n}^L\right)$

Multi Layers of Neural networks

Layers Superscript

Input Vector: $\boldsymbol{p}$

Output Vector: $\boldsymbol{a}$

Bias Vector: $\boldsymbol{b}$

Net input Vector: $\boldsymbol{n}$

Weight Matrix: $\boldsymbol{W}$

Input Number: $\boldsymbol{R}$

Neuron Numbers for Layer: $\boldsymbol{S}$

2. General Architecture NTSA

A NTSA is an information processing system which operates on inputs to extract information, and produces outputs corresponding to the extracted information. NTSA model is composed of simple processors, each having a local memory. Processing elements are connected by unidirectional links that carry discriminating data the linear feed-forward net has been found to be a suitable one for training techniques. Outputs of neurons in one layer are transferred to their corresponding neuron in another layer through a link that amplifies or inhibits such outputs through weighting factors. Except for the processing elements of the input layer, the input of each neuron is the sum of the weighted outputs of the node in the prior layer and a bias. Each neuron is activated according to its input, transfer function, and threshold value [5].

There exist a variety of ANN models and learning procedures. Feed-forward networks are well known approaches for prediction and database processing applications. In this type (NTSA), the weighted and biases links feed activation functions from the input layer to the output layer in forward direction.

Figure 3 shows the general feed-forward multilayer networks model, including two hidden layers. The "distribution factors" beams [6], of the input layer constitutes the neurons inputs in layer (L_1), representing a set of variables.

$$M^1 4_{vd} \text{ Rotational Moment Node4 Layer1, Right Hand Beam}$$

$$M^1 5_{vd} \text{ Rotational Moment Node5 Layer2, Down Column}$$

A single-input neuron is shown in **Figure 2**. The scalar input p is multiplied by the scalar weight w to form wp, one of the terms that are sent to the summer. The neuron has a bias b, which is summed with the weighted input to form the net input n, goes into transfer function f (linear function), which produces the scalar neuron output a.

If we related this simple model of the neuron with biological neuron, then the weight *w* corresponds to the strength of a synapse, the neuron body is represented by the summation and the transfer function, and the neuron output *a* represents the signal on the axon [5].

$$\left(x_1, x_2, x_3\right)^{\mathrm{T}}.$$

The inputs and outputs for the i_{th} neuron are:

$$n_i^L = W_i^L a_i^{(L-1)} + b_i^L \tag{1}$$

$$a_i^L = f_i\left(n_i^L\right) = f_i^L\left(W_i^L a_i^{(L-1)} + b_i^L\right) \tag{2}$$

where f_i constitutes an activation function linear, its behavior is that of a threshold function, in which the output of the neuron is generated if a threshold level, is reached. The net input and output o the j_{th} neuron are similarity treated as in (1) and (2).

Typically, the "activation function" (purelin) is chosen by the designer and then the parameters *w* and *b* will be adjusted by some learning algorithm so that the neuron input/output relationship meets some specific goal.

2.1. Multiple Layers of Neuron

Generally, one neuron with many inputs may not be sufficient. We might need three, four or seven, operating in parallel, in what we will call a layer. Yet, can be considered a network with several layers. Each layer has its own weight matrix $\boldsymbol{W}$, its own bias vector $\boldsymbol{b}$, a net vector $\boldsymbol{n}$ and an output vector $\boldsymbol{a}$.

The first layer has $\boldsymbol{S}^1$ neurons, $\boldsymbol{S}^2$ neurons in the second layer, etc. As noted, different layers can have different numbers of neurons. Thus, the weight matrix for the first layer is written as $\boldsymbol{W}^1$, and the weight matrix for the second layer is written as $\boldsymbol{W}^2$. This notation is used in all network models of the NTSA system. The outputs of the layers one, two, and three are the inputs for layers two, three and four. Thus, layer three can be viewed as a one-layer network with $\boldsymbol{S}^3$ inputs.

There is, one correspondence between the S^1 neurons of the first layer, S^2 of the second layer neurons, the neurons S^3 of the third layer as long as $S^1 = S^2 = S^3 = \cdots$ and so on, so that the output of a neuron of the first layer is the input to the corresponding neuron in the second layer and so on, that is, the output of the L_{th} layer or output layer is $\boldsymbol{a}^L$ vector:

$$\begin{array}{l} a_1^1 \to a_1^2 \to a_1^3 \to \cdots \to a_1^L \\ a_2^1 \to a_2^2 \to a_2^3 \to \cdots \to a_2^L \\ a_3^1 \to a_3^2 \to a_3^3 \to \cdots \to a_3^L \quad , \quad \boldsymbol{a}^L = \left(a_1^L, a_2^L, a_3^L, \cdots, a_N^L\right)^{\mathrm{T}} \\ \qquad\qquad \vdots \\ a_N^1 \to a_N^2 \to a_N^3 \to \cdots \to a_N^L \end{array}$$

A layer whose output is the network output is called an output layer. The others layers are called "*hidden layers*". The network of **Figure 3** has an output (layer 4) and three hidden layers (layers 1, 2, and 3).

2.2. Recurrent Networks

The NTSA system is a recurrent network with feedback; some of its outputs are connected to the inputs subsequent network, put it in some way. This is quite different from the networks models that we have studied thus far, which were strictly feed-forward with no backward connection. The NTSA is of the forward-backward class by the system architecture.

3. Artificial Neural Mitosis (ANM)

In general, natural biological organisms are much more complicated than the automatic devices. However, some peculiarities that we observe in the organization of the first and the way they perform certain operations or behavior can serve as a reference to the inspiration of an approach required to solve a family of common problems in real life. The set of experiences and difficulties we face when we operate certain automatic devices can be the source of the interpretations of the physiological systems, mainly human and some vertebrate in particular.

The man has been inspired by the central nervous system of humans, which is actually the most difficult of all; which it has been the source of a biological inspiration that led to the creation of some automatic devices such as artificial neural networks. Such devices sometimes have certain limitations in its design and operation, when it comes to solve a certain class of problems. Consequently, we could ensure that such systems are still evolving. This means that the system is open to change, or equivalently; network models support new paradigms; without this, it's essential, such as learning and massive parallelism goals are violated.

The central nervous system of humans is of unlimited complexity. For their study, it is taken to biological neurons as independent units. Thus, it is isolated as the first stage of the problem; the structure and workings of such individual elemental processing units. The second part of the problem aims to understand how neurons are organized into a whole; and how that operation of all expressed, starting logically based on these individual elements of information processing.

Neuron is considered as automatic physiological devices as a "black box", which react to the presence of certain stimuli and issues a response as independent functional units. They have clearly defined characteristics, *i.e.*, neurons that receive signals can be of two types excitatory or inhibitory. To stimulate a neuron is necessary that this receive excitatory stimuli. After a certain time, the neuron will issue "one and only one" output pulse (signal).

What would you do if a neuron of the output layer of a network, like a node of a structure, had to pass over a different signal to other neurons or to the outside world? Another question, why, if the artificial neuron is a real operation of a biological neuron very large simplification, cannot inspire and create an artificial neuron that go further; and generate neurons daughters with the same DNA? The DNA of the neuron mother and daughters would be defined by: 1) the same entry, from the corresponding neuron in the previous layer; 2) the same trend vector; 3) the same transfer function. The mother and her daughter's neuron differ only in the connection weight and its output signal.

In this evolutionary scenario, it is another question. Is it possible to think of a neuronal structure and function, to incorporate the concept of neuronal multiplication? Yes, it's a positive response. Then, the artificial neural mitosis could be a feature attributable to the artificial neurons, though, does not share this property essentially biological equivalent.

There are a variety of kinds of designs and learning techniques that are enriched with its own peculiarities, which can generate or manufacture under its own power users and meeting their own needs. Let us not forget that the field of neural networks is in the throes of evolution and growth; and therefore, for their development, much remains to be done.

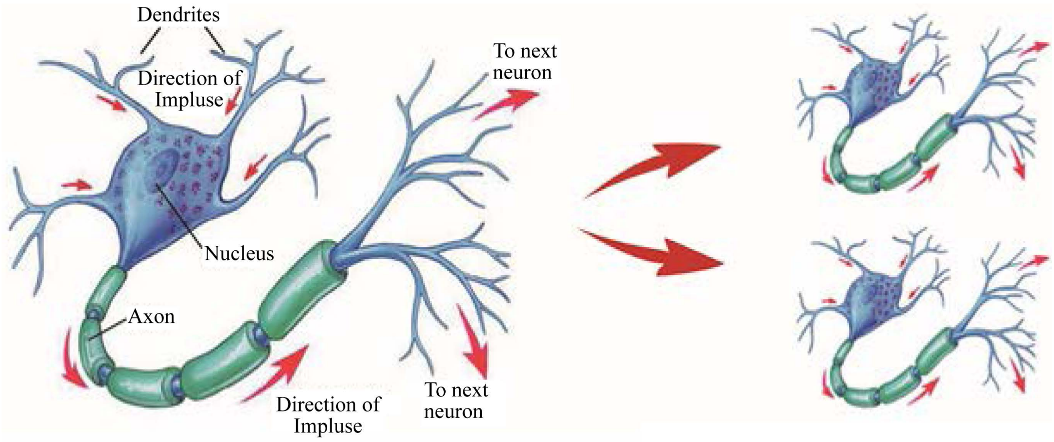
The synthesis of biological neurons is important and difficult because it is born the human brain. In it, we were inspired to produce electronic devices or artificial neural networks, which are the synthesis of artificial neurons. Artificial neural networks are characterized by their autonomy and logic flexibility. Running, at least part of the functions of the central nervous system [3].

Figure 5(a) and **Figure 5(b)** illustrate about the kinds of biological cells, which serve as inspiration to neurostructural analysis. The first is a biological neuron. The second is a cell of the class that forms the intestinal epithelium and the epidermis. The latter cells are able to multiply, to generate through a stem cell mitosis two or more identical to the mother daughters cells genetically. In other words, mitosis is a process of equitable sharing of genetic material, DNA [7].

Figures 6(c)-(e) show different versions of what could be the "mitosis of an artificial neuron" nodes in different frame structure. It can be seen at the bottom of the typical node scheme, a multilevel cross linked structure. Each node needs to transmit, two, three, four signals. Only after acceptance of the phenomenon of artificial neural mitosis, which replaces the neuron could meet the demand of the number of required signals.

If we accept the above approach, then we have met the expectations of the first part of the expansion of the framework of our theory, where the DNA of the neuron mother and her daughter's neurons is: DNA = Common entrance, from the corresponding neuron of the previous layer, as trend vector, same linear transfer function.

If a neuron of the output layer of a network, like a node of a structure had to transmit over a different signal to the external world. We say that is not possible, because, you're right is currently denied. However, in another scenario, as it, the available cells of the epidermis and corresponding to the intestinal epithelium, it would appear, through a biologically inspired by modifying the structure and neuronal functioning, so that the attribute Property neuronal multiplication (mitosis neuronal) is assigned to the artificial neuron. This would be debatable,



https://www.studyblue.com/notes/note/n/biological-foundations-neuron-communication-/deck/1025438 [8]

Figure 5. Biological cell mitosis.

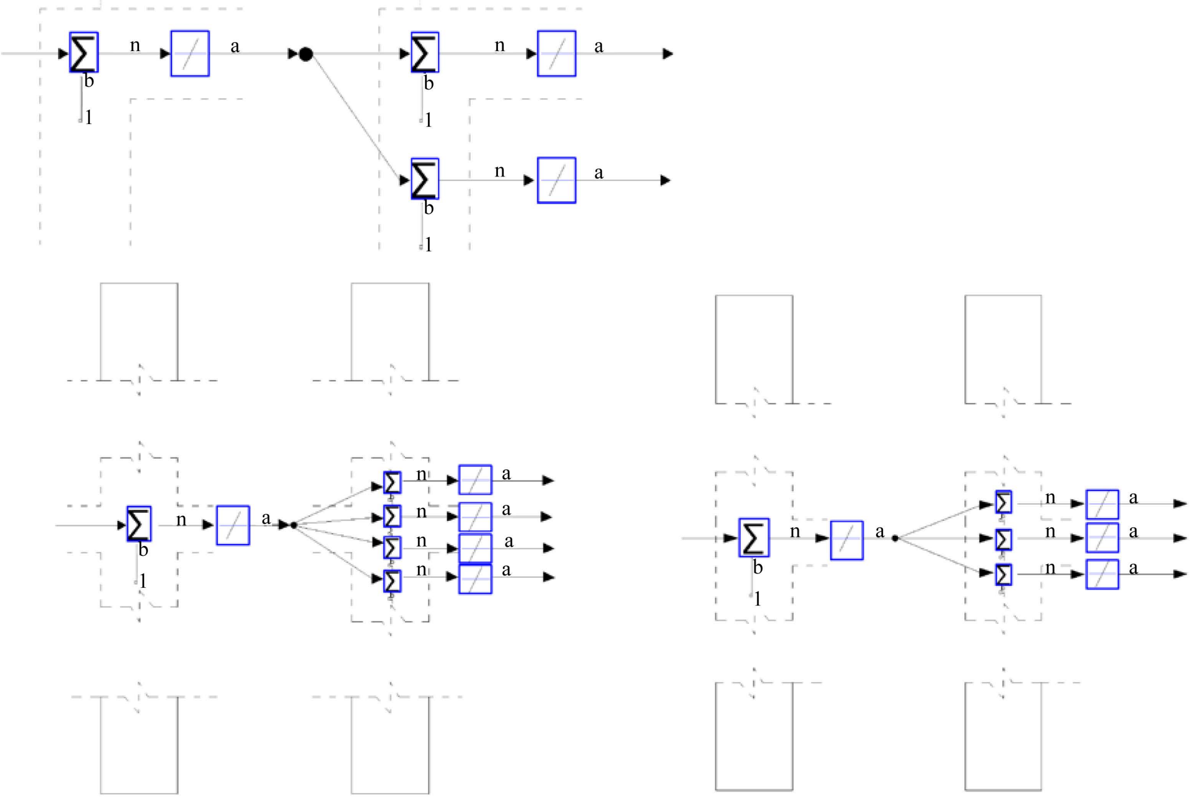


Figure 6. Artificial neural mitosis in the nodes of a structure.

but viable. Let's concede now, this transcendental to artificial neuron, as a result of biological inspiration derived from some special cells, such as the class of the above named property. Consequently, the Artificial Neural Mitosis (MNA), will become a feature attributable to the artificial neurons, though, this essentially **not share with their biological equivalent.**

4. General Architecture of Proposed Network

To bring about the first artificial neural network, a neuron is placed on each node of the network structure proposal. Thus, we have a one to one correspondence between nodes and gantry neurons of the various layers of the neural network thus formed [8]. To generate further models neural networks, neuro-structural model primitive neuro be divided into sub-models derived (sub-domains). This is achieved by eliminating successively, the last model obtained, and the layer of the right to illustrate the statement above with **Figure 7**.

The process of building the sub-neural network models starts with the artificial mitosis, the last layer in the first model (original model). The remaining sub-models of neuro-structural derivatives, network obtained after removal of the respective output layer network model precedent, then, produce mitosis of the output layer of the new sub-model.

All neuro-network models thus obtained operate in a massively parallel and have as output vector, the ends rotational moments associated with the nodes of the output layer. The calculation process is iterative and each network will be implemented through successive approximations; and the same, will be controlled by the learning algorithm. Thus, we can know the outputs of the hidden layers of the original multi-cross-linked network. It is important to note that each network model will shed their own results. Values are retained as data, to operate the following neuro-network model. Finally, the matrix of the results of total rotational moments at the ends of all members of the structure is obtained.

All neuro-derived models have the same input vector.

$\boldsymbol{p} = \left(\mu 11_{vd}\ \mu 12_{vd}\ \mu 13_{vd} \cdots \mu 1N_{vd}\right)^{\mathrm{T}}$. An exception, the derivative of neuro-layer model, whose components of the input vector are all factors of rotation of the ends of the rods, which contribute to the nodes of the first layer, that is,

$$\boldsymbol{p} = \left(\mu 11_{vd}\ \mu 11_{ci}\ \mu 12_{cs}\ \mu 12_{vd}\ \mu 12_{ci} \cdots \mu 1\left(3N-2\right)_{vd}\ \mu 1\left(3N-1\right)_{ci}\right)^{\mathrm{T}}$$

4.1. Division of Neuronal Structure Substructures—Mitosis 1

The "artificial neural network model primitive" or "original model" constitutes the basic network for the development of neural approach "structural analysis with artificial mitosis". This network represents the inspiring model, from which we will aim to get a set of "sub-derived models of neural networks". See the original structural model of **Figure 7**.

If you look closely, the output layer plane gantry N levels and L layers of **Figure 7**, we realize that the layer required on each node, as many outlets as members concur in it. **Figure 8** shows an overview of artificial neural network model derived mitosis 1, which has the same architecture ANN primitive, but differs from this only in the number of neurons in the output layer. Mitotic neurons are locally attached Model 1, with unidirectional connections except the output layer. The model is a back-forward whose learning algorithm is self-supervised; and therefore will not require a supervisor, and that by itself, it will run internal monitoring to monitor network performance, with a linear transfer function (*purelin*). This algorithm is also used by the other models derived, with operating in series. It has been proven that the network model is an effective and autonomous system that enables processing of the structural physical parameters, entering the network from an input vector whose components are the distribution factors, which belong to the beams of the first layer. These factors are clearly part of certain characteristics of the original structure [9] [10].

The outputs of the hidden layer neurons any, are transferred to corresponding processing elements of the back

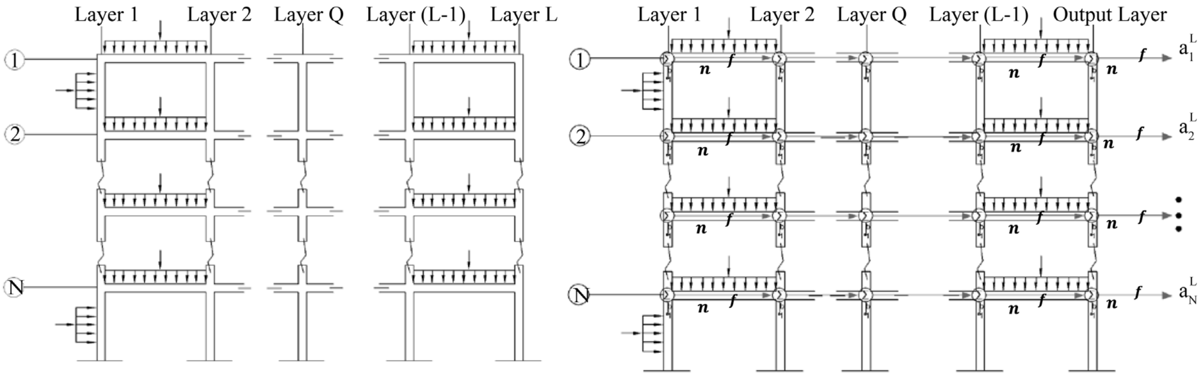


Figure 7. Primitive structure—artificial neural network primitive.

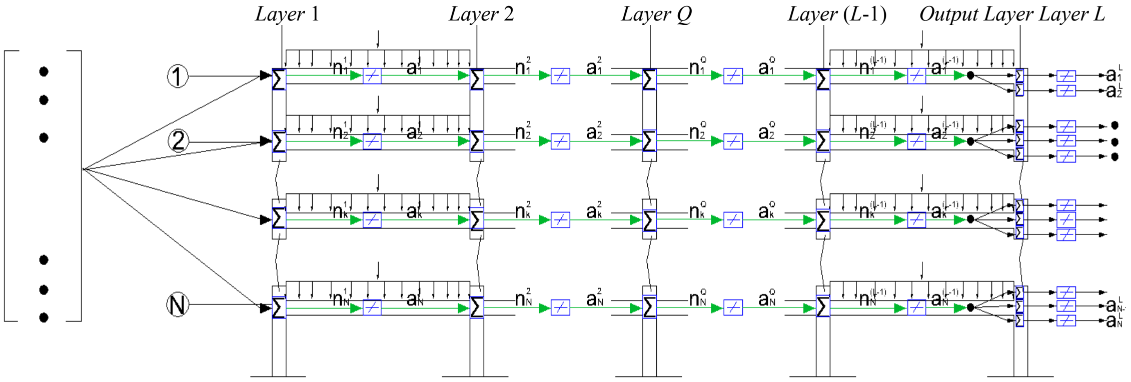


Figure 8. Model mitosis 1.

layer, through local connections, after balancing the weight factors. Except for the processing elements of the input layer neurons and output layer, the input of each neuron is the sum of the weighted outputs of the nodes (neurons) in the previous layer plus the corresponding trend.

Each neuron or processing element is activated in accordance with the input into the neuron activation function and the value node trend or threshold of the neuron. L_e input layer consists of an arrangement of $S^1 = N$ neurons, an output layer, containing $S^L = 3N-1$ processing elements. Between layers of input and output L_e and L_o are $(L-2)$ hidden layers. Each layer has its own matrix of weights $\boldsymbol{W}$ your own trend vector $\boldsymbol{b}$, a vector of net input $\boldsymbol{n}$ and a vector output $\boldsymbol{a}$.

4.2. Sub-Models Derived by Sectioning

For rotational moments at the ends of the members who attend nodes in the hidden layers, the neuro-structural model primitive neuro be divided into sub-models derived (sub-domains). This is achieved by removing successively each time the original model, a layer neuron from right to left. This process begins immediately after, the mitosis 1 generated by the first sub-model derived. The $(L-1)$ sub-models of neuro-derived remaining structural network are obtained after each respective switching and mitosis of the output layer of the new sub-model

4.3. Sectioning to Mitosis 2

For rotational moments at the ends of the members of the hidden layers of a structure, primitive neuro-structural model must be divided into sub-models derived neuro (sub-domains). This is achieved by removing successively each time the original model, a layer neuron from right to left. The key partitioning process of primitive neuro-structural model sequentially, to give rise to sub domains or neuro-substructures, lies in the fact that the nodes of the new output layer are predominantly high connectivity; and the same, are assigned to processors distributed memory which leads to an iterative application of massively parallel processing, with the support of the neural mitosis [11]. This process is solved the problem of determination sub-end rotational moments of hidden layers of the structure.

4.4. Neuronal Architecture Mitosis Model

The model derived mitosis 2 is a network fed forward (back-forward) whose learning algorithm is self-supervised; and therefore did not require a supervisor, and that by itself, it will run internal monitoring to monitor network performance, with a linear transfer function.

The outputs of the neurons of a layer are transferred to the next layer corresponding through local connections excite or inhibit such exits through weighting factors. Except for the processing elements of the input layer neurons and output layer.

The input of each neuron is the sum of the weighted outputs of the neurons in the previous layer plus the corresponding trend. Each neuron or processing element is activated in accordance with the input into the neuron activation function and knot trend value or threshold of the neuron. In **Figure 9** the mitosis neuronal model 2, which is composed of $L-1$ layers shown, namely, having a layer unless the original model and the ANN mito-

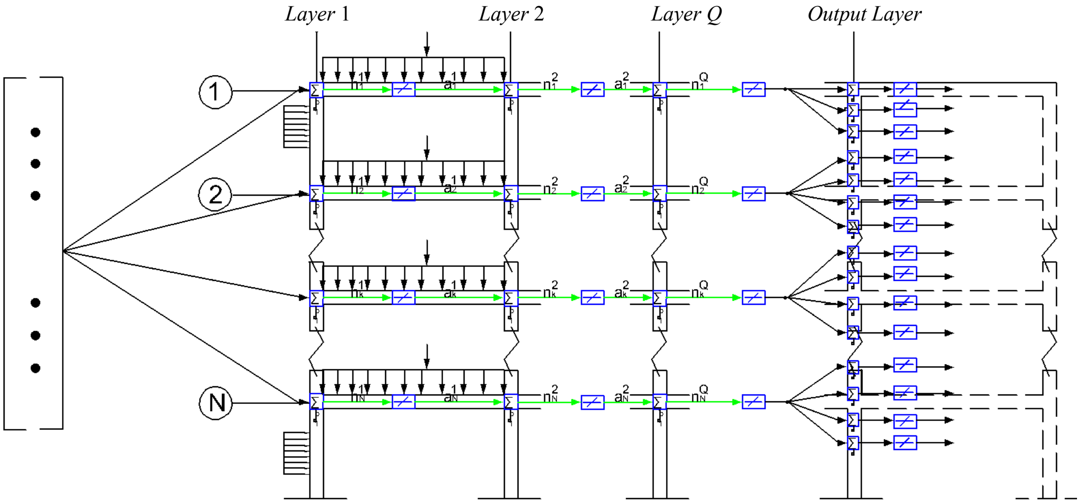


Figure 9. Mitosis artificial 2.

sis 1. The input layer L_e is an arrangement of $S^1 = N$ neurons; the output layer L_o contains $S^{(L-1)} = 4N-1$ processing elements. Between layers of input L_e and output L_o are $(L-2)$ hidden layers. Each layer has its own matrix of weights $\boldsymbol{W}$, its own trend vector $\boldsymbol{b}$, a vector of net input $\boldsymbol{n}$ and a vector output $\boldsymbol{a}$.

For rotational moments at the ends of the members attending the nodes of the *Q-th* hidden layer of the structure, primitive neuro-structural model will be severing all layers to the right of the *Q-th* layer sequentially one to one.

4.5. Neuronal Architecture Model *Q*-th Artificial Mitosis

Then in Figure 10 an overview, the artificial neural network model built with simple processing elements shown, which have a local memory. Neurons are connected with unidirectional connections, except the output layer. The model is a self-supervised back forward network, the algorithm is similar to the other models that operates in series and therefore will not require a supervisor and that by it will run internal monitoring, to monitor the performance network, with a linear transfer function.

Between layers of input L_e and output L_s^Q, are $(Q-1)$ hidden layers, which generally own N neurons. Each layer has its own matrix of weights $\boldsymbol{W}$, its own trend vector $\boldsymbol{b}$, a vector of net input $\boldsymbol{n}$ and a vector output $\boldsymbol{a}$.

4.6. Sectioning of the *L*-th Artificial Mitosis

To bring the last sub-model derived corresponding to the *L-th* neural mitosis to run the *L-th* isolation of primitive neural model accompanied the final process of neural mitosis. Precisely, this sub-model is the only one with a different input vector with $R = (3N-1)$ components.

4.7. Layer Neural Architecture Model: *L*-th Artificial Mitosis

Figure 11 shows a general scheme, the *L-th* neural network model derived from a layer with $S^1 = (3N-1)$ neurons which have a small local memory. The input vector to the network has $R = (3N-1)$ components.

$$\boldsymbol{p} = \left(\mu 11_{vd}\, \mu 11_{ci}\, \mu 12_{cs}\, \mu 12_{vd}\, \mu 12_{ci} \cdots \mu 1(3N-3)_{cs}\, \mu 1(3N-2)_{vd}\, \mu 1(3N-1)_{ci}\right)^{\mathrm{T}}$$

It has been found that the *L-th* network model derivative, *L-th* mitosis is an effective and independent system, which enables processing of the structural physical parameters, entering the network from an input vector whose components distribution are factors that belong to the beams and columns that meet in the first layer of the structure. These factors are clearly part of the original structure certain physical characteristics.

In Figure 11 the neural model of a layer, is composed of neurons $S^1 = 3N-1$. The layer has its own array of weights $\boldsymbol{W}^1$, its own vector $\boldsymbol{b}^1$ trend; net input vector $\boldsymbol{n}^1$ a component input vector $\boldsymbol{p} = 3N-1$ component and a vector output $\boldsymbol{a}^1$.

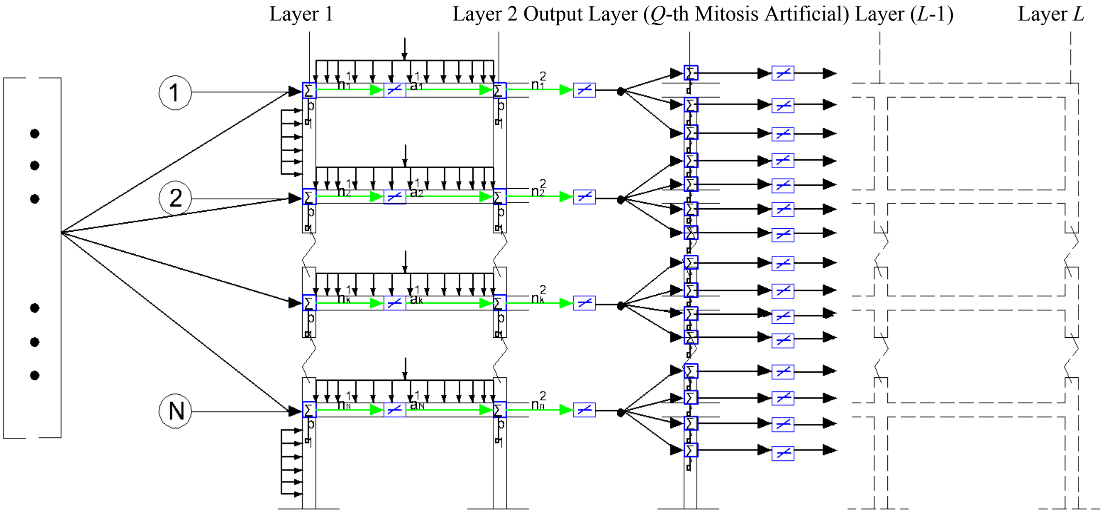

Layer 1
Layer 2 Output Layer (*Q*-th Mitosis Artificial) Layer (*L*-1)
Layer *L*
1
2
N

Figure 10. Mitosis artificial *Q*-th.

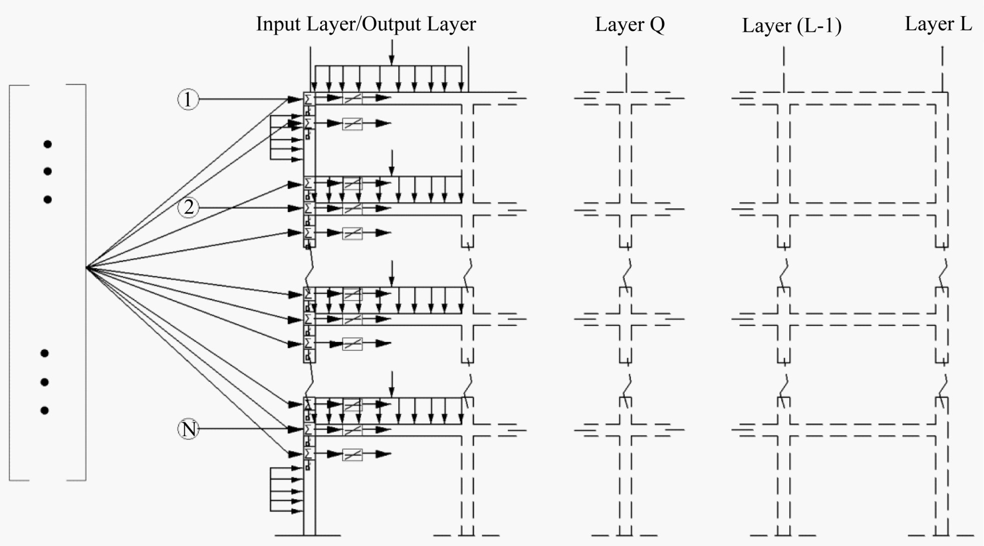


Figure 11. First layer mitosis artificial.

5. Sequence of Neuro-Structural Models

The sequential division of neuro-structural primitive, together model with artificial neural mitosis is a mechanism by which a system of hybrid neural sub-models, capable of giving rise to a registration process, input, processing, and storage is generated and output information from internal and external means of a flat lattice structure of several levels. **Figure 12** is shown in simplified form.

The first sub-model derived neuronal differs from sub-neuro-structural model primitive, just in the number of neurons in the output layer. From this differentiation, each model differs from the previous derivative in chronological order; and having a layer of neurons unless this; and possibly the number of neuron of the output layer.

All neural models like its primitive have the same input vector. An exception, the model derived from a layer, which has an input vector with a number of components equal to the number of neurons of a layer model.

System Networks

The neuro-structural analysis is a new alternative, which is based on two strategies: the division of the porch in

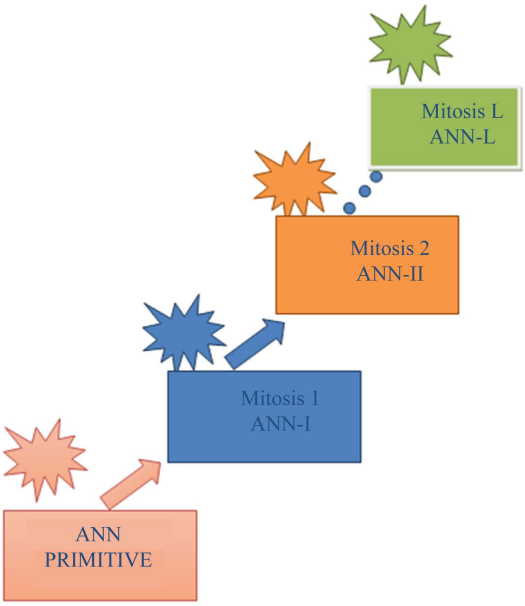


Figure 12. Neural system sub-models series.

sub-structures or sub-domains and artificial mitosis of the output layer, the primitive model.

The first sub-model derived is called NTSA-I (artificial mitosis 1) shown in **Figure 7**. NTSA-I calculated rotational moments at the ends of the members which contribute to the final layer of nodes of the output layer structure or network. The sub-model-II ANN (artificial mitosis 2) calculates the rotational moment at the ends, which contribute to the nodes of the penultimate layer, or output layer of the network; and so on, until the sub-model simulation of NTSA-L (sub-neural network model of a single layer), which calculates the rotational moments at the ends of the members who attend at the nodes of the first layer original structure

A macro program called mitosis generates artificial series of calls to all derivatives sub-models; and then the resulting NTSA-I sub-model calls the NTSA-II and so on, until the call of the NTSA-L, to complete the implementation of the series.

6. The Procedure

The procedure of the NTSA system depends on the solution of three problems for the determination of member constants on fixed end moments, the stiffness at each end of member, and of the over-carry factor distribution factors μ_{ik} for the rotational end moments and distribution factors ϑ_{ik} for the lateral displacement moments at each end for each member of the frame under consideration. The determination of these values is not a part of the presented approach.

The NTSA model has four layer, three inputs and five output values. The '*function network*' of the Toolbox of the Matlab creates the net, which generates the first layer weights matrices and biases vectors for the four linear layers required for this problem. These weights and biases can now be trained incrementally using the algorithm. The network must be trained in order to obtain first layer weights and biases. For the second, third and fourth layers, the weights and biases are modified in response to network's inputs and will lead to the correct output vector. There are not target outputs available. The linear network was able to adapt very quickly to the change in the outputs. The fact that it takes only seven iterations for the network to learn the input pattern is quite an impressive accomplishment [1] [2].

The scheme for entering the calculations systematically is shown in **Figure 13**. The procedure explained above, is best illustrated by solving the structure in **Figure 13**, which is loaded in a rather complex fashion. The distribution factors for nodes 1, 2, and 3 constitute the net's input vector.

The fixed end moments for the different loaded members are calculated by using the standard formula available in any structural handbook. Having completed these preliminary calculations, the training can be initiated. The network was set up with the three parameters (distribution factors of the beam) as the input, and the rota-

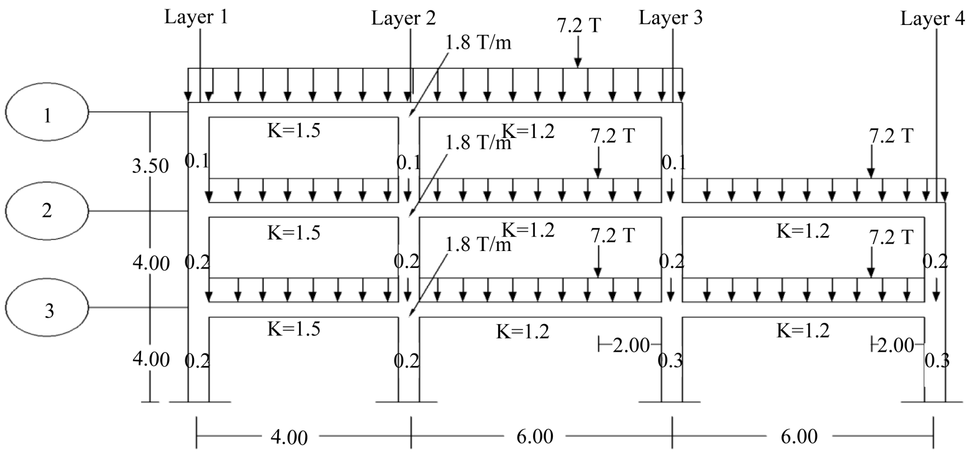


Figure 13. Example of application.

tional end moments due to rotation as the outputs determined by the first layer [3].

The calculation starts in the input layer and continues from one layer to the next. Such calculation is carried out quickly. After 6 or 7 iterations have been performed, as explained earlier, it will be noted that there is little or no change in the values of two consecutive sets of calculations. The calculations are now stopped and the values of the last iteration are taken as the correct ones, with the previous values being ignored. For the sake of clarity, these final values have been indicated separately [9].

The architecture of the ANN will be designed with the "Network Toolbox function of the Artificial Neural Networks" and simulations will be performed using the "Sim tool", both belonging to the Matlab environment. This algorithm, through multiple interactions with the application of the method of the successive approximations, achieves the learning object models "artificial mitosis" [11].

The first ANN model derived or "artificial mitosis 1", allows us to obtain the rotational moments at all ends of the members who concur in the nodes of the fourth layer and output layer, see **Figure 15**. The second network model "artificial mitosis 2" gives as results the rotational moments at all ends of the bars that come together in the nodes of the output layer (third layer). The third neural network model of the application is the "artificial mitosis 3", which calculates the rotational moments at the ends of rods in either concurrent nodes of the second layer counted from left to right these.

Finally, the fourth model derived or "artificial mitosis 4" consists of a single layer of neurons. This network allows us to calculate the rotational moments, partners at all ends of the members (beams and columns) that access to the nodes of the first layer, the artificial neural network model primitive (Ben Mark) [9].

The primitive neural network model consists of four layers of neurons, three hidden layers including the input layer and the output layer. We adopt the type of diagram shown in **Figure 14**, which allows us to write the successive values at the ends of each member. And also it facilitates the systematic income calculations. Show of the numerical calculations will to be performed later, including distribution factors and moments of perfect embedding at the ends of the members.

The model of multi-gantry plane lattice structure will be used as a source of direct inspiration for the design of primitive neural network model. The analogy between models allows the following procedure: Repeat, *i.e.* double the structural network model, putting in place of each node of a neuron structural model. During the transformation of the structural model in neural model, columns connections between nodes are removed. Thus, the two models differ in the connectivity leading columns. But we should not worry about this, because, associated with each of the layers of primitive neural model and all possible models derived from it, the vectors trends of each layer, taking into account actions that produce connections columns that disappear in the primitive model and its derivatives. In this way, the equivalence between models remains.

The primitive neural network model is shown in **Figure 15**, this network is back forward. The network consists of simple processing elements, each of which has a local memory. These neurons are connected by unidirectional connections that transfer data from one neuron to the next corresponding layer. The model is a network whose self-supervised algorithm does not require a supervisor, and that by itself, it will run internal monitoring

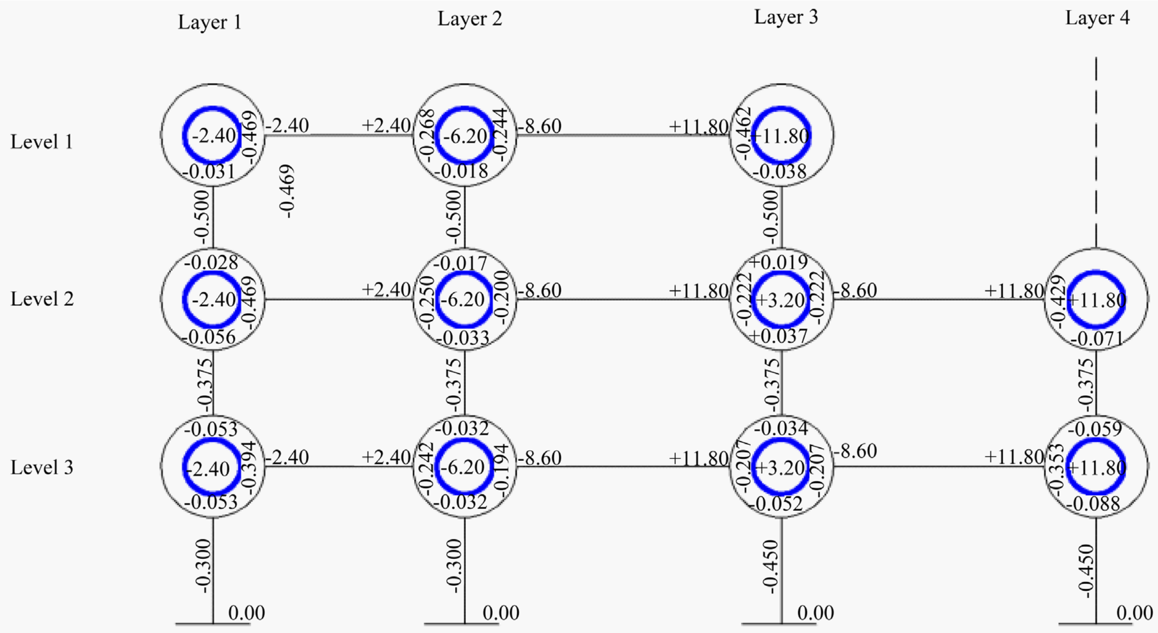


Figure 14. Distribution factors and fixed-end moments.

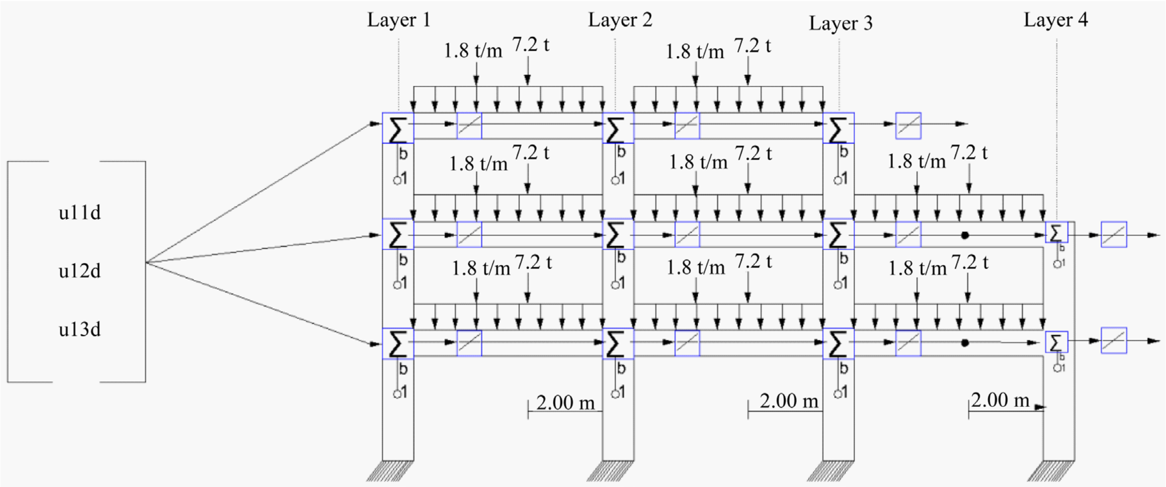


Figure 15. Neuronal model primitive.

to monitor network performance, using a linear transfer function (purelin). The input vector components are factors corresponding to the beam ends that meet in the input layer to the distribution network [12].

Here are the original network model inspired by the structural model, keeping one correspondence between nodes in the structural model and the model of primitive neurons network.

6.1. Primitive Neural Network

The model consists of four layers of neurons. Each layer has its own matrix of weights $\boldsymbol{W}$, its own trend vector $\boldsymbol{b}$, a vector of net inflow $\boldsymbol{n}$ and a vector output $\boldsymbol{a}$. Following established notation the first, second, third and fourth layer respectively as will matrices $\boldsymbol{W}^1\boldsymbol{W}^2\boldsymbol{W}^3\boldsymbol{W}^4$.

Artificial neural networks are intelligent tools which are extremely useful in situations, for which the rules are not clear enough or are difficult to establish artificial neural networks are robust and fault tolerant. You can calculate a function without a mathematical description of how the output function is operatively associated with the input. She learns to approximate functions even when its form cannot be specified accurately.

6.2. Artificial Mitosis 1 Model

It is an artificial neural network with four layers. The first three layers of "mitosis Model 1" are similar to the original model. The input layer is $S^1 = 3$ neurons. Each neuron is connected locally to its corresponding vector component $\boldsymbol{p}$ input components $R = 3$. The input vector components are key values, called "distribution factors" [2]. The matrix of the weights of the second layer (hidden) $\boldsymbol{W}^2$ is a diagonal matrix and its elements are the distribution factors of the left ends of the beams are located between the second and third layer of nodes. While the trend vector $\boldsymbol{b}^2$ of the second layer has as fastening components moments belonging to each node of the layer, the moments due to horizontal displacement of the ends of columns and the rotational moments of the opposite ends of the beams and columns which contribute to the knot or processing unit [1].

The second layer is $S^1 = 3$ neurons. Each neuron is connected locally to the unit of processing of the previous layer. The outputs of the layers one and two are the inputs to the layers two and three. Thus, layer two can be seen as a network of a layer with $R = S^1 = 3$ inputs, $S = S^1 = 3$ neurons and an array of weights $\boldsymbol{W}^2$ diagonal $S^1 \times S^3$. The input to the layer two is the vector $\boldsymbol{a}^1$, and $\boldsymbol{a}^2$ is output vector [10].

The matrix of the weights of the third layer (hidden) W^3 is a diagonal matrix whose elements are also the (non-adjustable) "distribution factors" left ends of the beams are between the third and fourth layer of neurons. While the bias vector associated $\boldsymbol{b}^3$ third layer has components similar to the other bias nature. This layer has $\boldsymbol{S}^3$ neurons. As with the second layer, this is locally connected through the vector $\boldsymbol{a}^2$ to the next layer. The output layer is also the output of the network; this layer has an array of weights $\boldsymbol{W}^4$ and $S^4 = 5$ neurons. The matrix of weights has the elements to "factors of distribution" of all the ends of the members attending the nodes of the output layer [1]

Figure 16 shows the network model after suffering the first mitosis the output layer of the original model, taking into account the number of required outputs node double edge and one edge node.

Mitosis Neuronal Architecture Model 1

```
net =
Neural Network object:          numInputs: 1
numLayers: 4                biasConnect: [1; 1; 1; 1]
inputConnect: [1; 0; 0; 0]      layerConnect: [4 × 4 boolean]
```

Figures 17-20 show codes (mitosis 1, mitosis 2, mitosis 3 and mitosis 4) and explained as the neural network of the program Matlab toolbox organises the simulation of networks. MATLAB writes and reads many file during a typical NTSA system analysis.

6.3. Artificial Mitosis 2 Model

It is a neural network model composed of three layers of neurons. The first two layers of Mitosis model 2 are

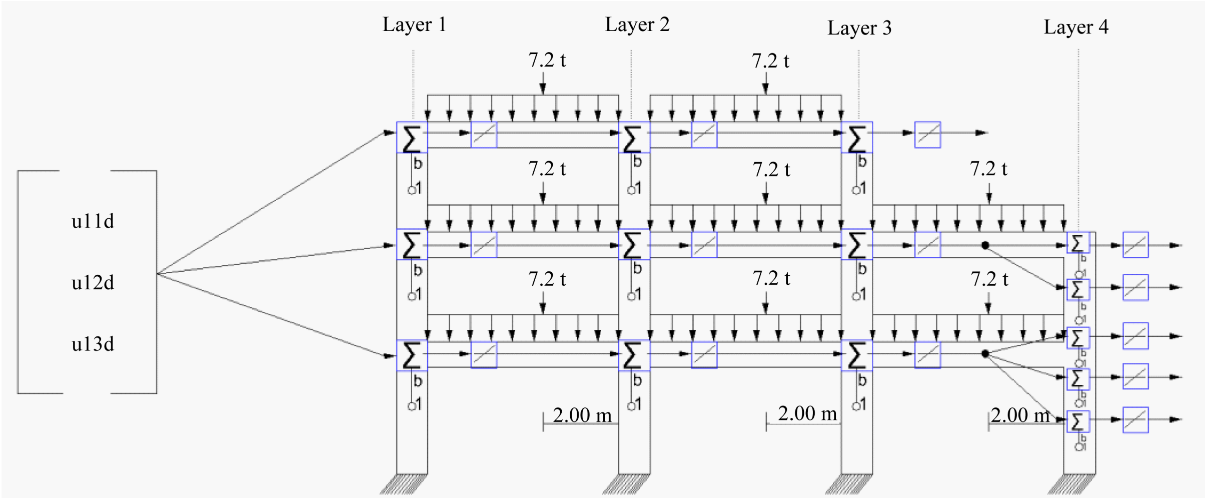
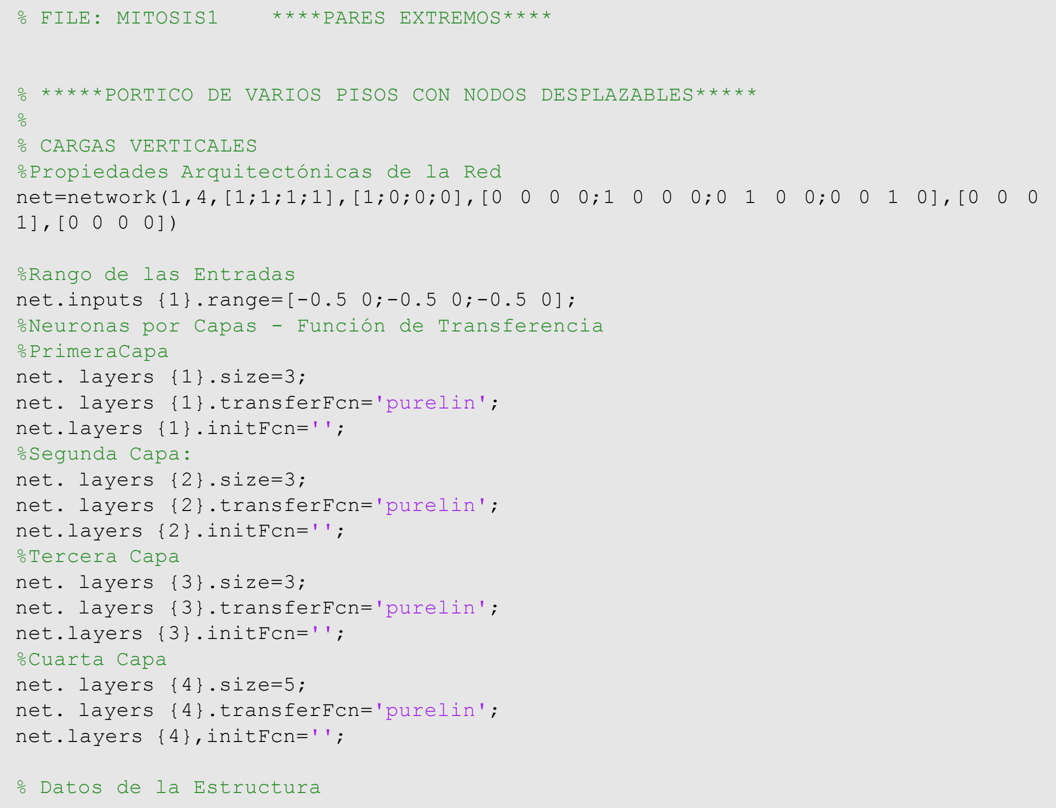


Figure 16. Mitosis model 1—fourth layer output.



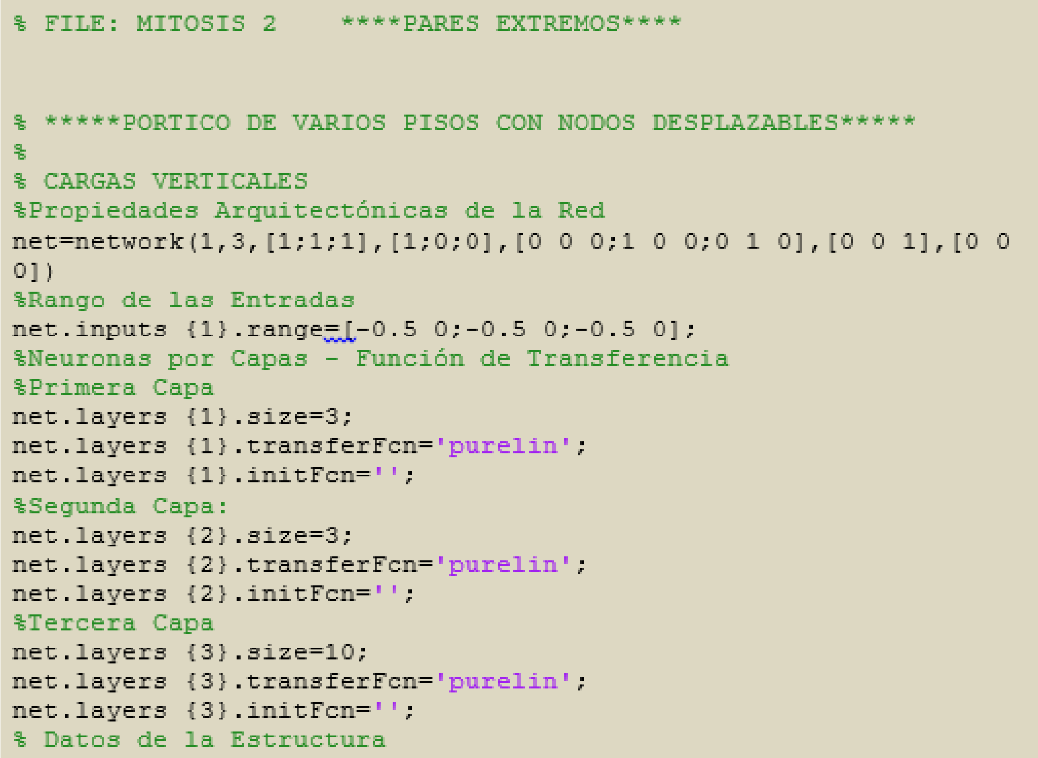
```
% FILE: MITOSIS1    ****PARES EXTREMOS****


% *****PORTICO DE VARIOS PISOS CON NODOS DESPLAZABLES*****
%
% CARGAS VERTICALES
%Propiedades Arquitectónicas de la Red
net=network(1,4,[1;1;1;1],[1;0;0;0],[0  0  0  0;1  0  0  0;0  1  0  0;0  0  1  0],[0  0  0
1],[0 0 0 0])

%Rango de las Entradas
net.inputs {1}.range=[-0.5 0;-0.5 0;-0.5 0];
%Neuronas por Capas - Función de Transferencia
%PrimeraCapa
net. layers {1}.size=3;
net. layers {1}.transferFcn='purelin';
net.layers {1}.initFcn='';
%Segunda Capa:
net. layers {2}.size=3;
net. layers {2}.transferFcn='purelin';
net.layers {2}.initFcn='';
%Tercera Capa
net. layers {3}.size=3;
net. layers {3}.transferFcn='purelin';
net.layers {3}.initFcn='';
%Cuarta Capa
net. layers {4}.size=5;
net. layers {4}.transferFcn='purelin';
net.layers {4},initFcn='';

% Datos de la Estructura
```

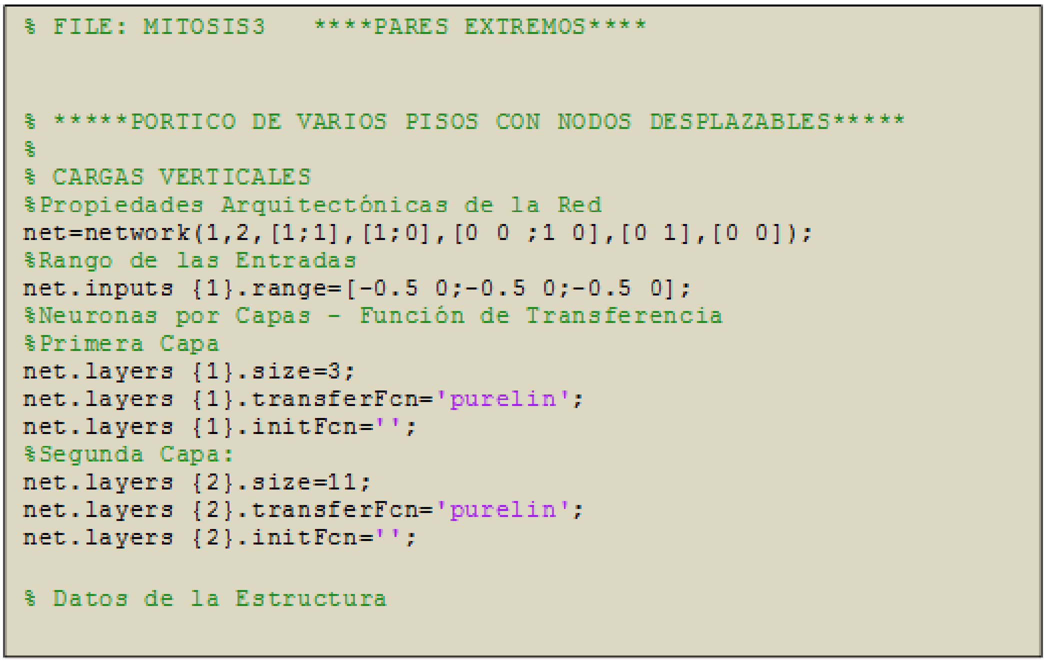
Figure 17. Mitosis 1 code-1.



```
% FILE: MITOSIS 2    ****PARES EXTREMOS****


% *****PORTICO DE VARIOS PISOS CON NODOS DESPLAZABLES*****
%
% CARGAS VERTICALES
%Propiedades Arquitectónicas de la Red
net=network(1,3,[1;1;1],[1;0;0],[0 0 0;1 0 0;0 1 0],[0 0 1],[0 0
0])
%Rango de las Entradas
net.inputs {1}.range=[-0.5 0;-0.5 0;-0.5 0];
%Neuronas por Capas - Función de Transferencia
%Primera Capa
net.layers {1}.size=3;
net.layers {1}.transferFcn='purelin';
net.layers {1}.initFcn='';
%Segunda Capa:
net.layers {2}.size=3;
net.layers {2}.transferFcn='purelin';
net.layers {2}.initFcn='';
%Tercera Capa
net.layers {3}.size=10;
net.layers {3}.transferFcn='purelin';
net.layers {3}.initFcn='';
% Datos de la Estructura
```

Figure 18. Mitosis 2 code-2.

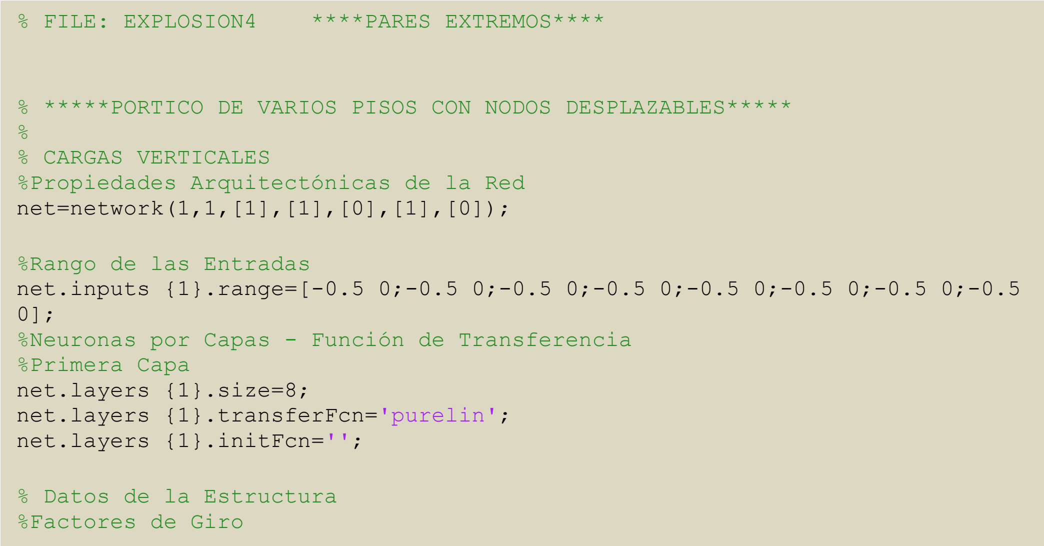


```
% FILE: MITOSIS3    ****PARES EXTREMOS****


% *****PORTICO DE VARIOS PISOS CON NODOS DESPLAZABLES*****
%
% CARGAS VERTICALES
%Propiedades Arquitectónicas de la Red
net=network(1,2,[1;1],[1;0],[0 0 ;1 0],[0 1],[0 0]);
%Rango de las Entradas
net.inputs {1}.range=[-0.5 0;-0.5 0;-0.5 0];
%Neuronas por Capas - Función de Transferencia
%Primera Capa
net.layers {1}.size=3;
net.layers {1}.transferFcn='purelin';
net.layers {1}.initFcn='';
%Segunda Capa:
net.layers {2}.size=11;
net.layers {2}.transferFcn='purelin';
net.layers {2}.initFcn='';

% Datos de la Estructura
```

Figure 19. Mitosis 3 code-3.



```
% FILE: EXPLOSION4    ****PARES EXTREMOS****


% *****PORTICO DE VARIOS PISOS CON NODOS DESPLAZABLES*****
%
% CARGAS VERTICALES
%Propiedades Arquitectónicas de la Red
net=network(1,1,[1],[1],[0],[1],[0]);

%Rango de las Entradas
net.inputs {1}.range=[-0.5 0;-0.5 0;-0.5 0;-0.5 0;-0.5 0;-0.5 0;-0.5 0;-0.5
0];
%Neuronas por Capas - Función de Transferencia
%Primera Capa
net.layers {1}.size=8;
net.layers {1}.transferFcn='purelin';
net.layers {1}.initFcn='';

% Datos de la Estructura
%Factores de Giro
```

Figure 20. Mitosis 4 code-4.

equal to the first two layers of the original model and the two of "mitosis Model 1". Each layer of the model has its own matrix respectively weights: $\boldsymbol{W}^1$, $\boldsymbol{W}^2$, $\boldsymbol{W}^3$ and their delivery trend: $\boldsymbol{b}^1$, $\boldsymbol{b}^2$, $\boldsymbol{b}^3$. The parameters that characterize matrices and vectors weights trends continue thus in the primitive model, as in each of the models subject to mitosis except for its output layer. In **Figure 21**, the mitosis model 2 wherein the penultimate layer of primitive object model is a mitosis, considering the inherent connections to each of the nodes in the proposed structure is presented.

6.4. Artificial Mitosis 3 Model

Derived network model two layer is sectioning product and artificial mitosis of the second layer of primitive neural network model. Weights matrices and vectors associated trends to the two layers of the neural network

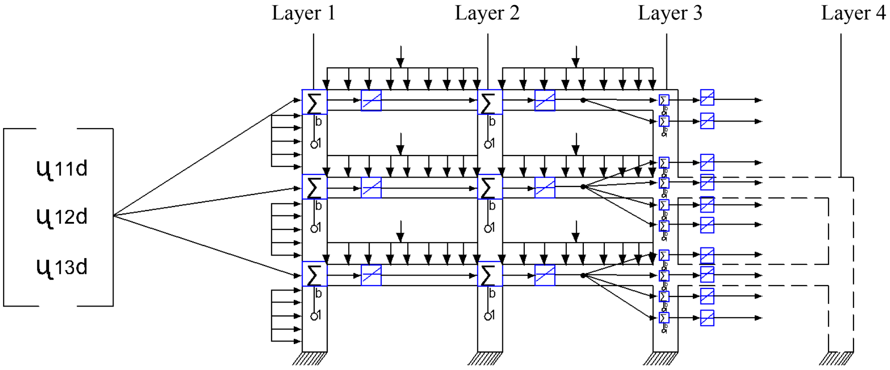


Figure 21. Mitosis model 2—third layer output.

model are derived respectively: $\boldsymbol{W}^1$, $\boldsymbol{W}^2$ and $\boldsymbol{b}^1$, $\boldsymbol{b}^2$. Weights matrix and vector trend corresponding to the first layer of the neural network model remain primitive. See Figure 22:

6.5. Artificial Mitosis 4 Model

This is a neural network model of a layer with eight (8) neurons and an input vector p of eight (8) components. In general, the number of components of the input vector to the layer is different from the number of neurons, ie, $R \neq S$. But in this particular case, the number of components of the input vector ($R = 8$) is equal and corresponds to $S = 8$ number of neurons of the input layer. Each component of the input vector p is connected to the corresponding neuron through the matrix of weights W, which is a diagonal matrix whose elements are the rightmost rotational moments of the beams are between the first and second layer columns or nodes. See Figure 23. Each *i-th* neuron has a tendency $\boldsymbol{b}_i$, an adder block, linear transfer function and output $\boldsymbol{a}_i$. Artificial neurons belonging to artificial neural mitosis 4 model are directly connected with the outside world; and in no case with other neurons [12]. Input vector:

$$\boldsymbol{p} = \left(\mu 11_{vd}\, \mu 11_{ci}\, \mu 12_{cs}\, \mu 12_{vd}\, \mu 12_{ci} \cdots \mu 1(3N2)_{vd}\, \mu 1(3N1)_{ci}\right)^{\mathrm{T}}$$

It has components distribution factors of the rotational moments of the members present in the first layer. This model has many components as vector has the layer neurons.

6.6. Results

The results of the simulation are presented in Figure 24.

The results matrix "all" showed above presents the values of the rotational moments at the ends of the various members who access the different layers of flat porch under discussion. The system of neuronal sub-types: mitosis1, mitosis 2, mitosis 3 and mitosis 4 result in a serial process that ends with the mitosis4, during which he lead to the recording, input, processing, storage and output of information from the media internal and external to the portal frame cross-linked multilevel flat. The neuro-structural analysis relies on two strategies: the neural mitosis and sectioning of the portal frame composed of layers.

Finally, a program called mitosis macro generates a series of sequential calls to the respective sub-models derivatives mitosis 3 mitosis 1…mitosis 4 mitosis 2; and thus, the execution of the series is completed. Figure 25 shows the final results of total extreme rotational moments of the portal frame members.

6.6.1. Verification of Results

Refer to example. One way of checking for the validity of our findings is to arbitrarily cut a node and apply the static equilibrium conditions, for node *i*, as in the Accompanying illustration, Figure 22. In all, the end moment can be written as follow:

$$M_{ik} = ME_{ik} + 2M'_{ik} + M'_{ki} + M''_{ik}$$

$$\text{and } M_{ki} = ME_{ki} + 2M'_{ki} + M'_{ik} + M''_{ik}$$

where,

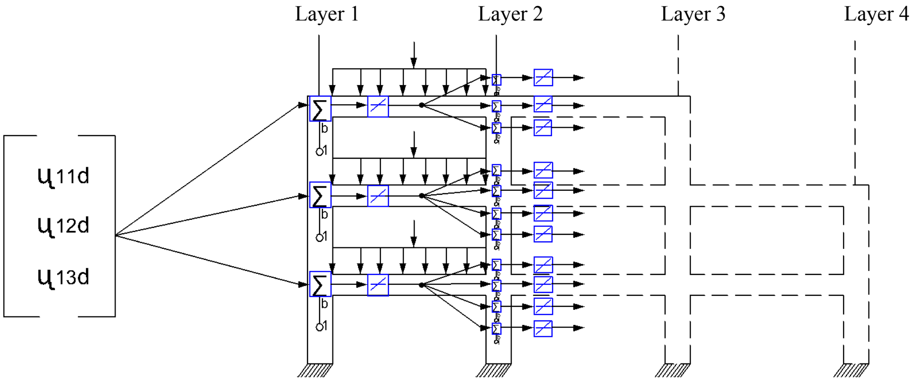
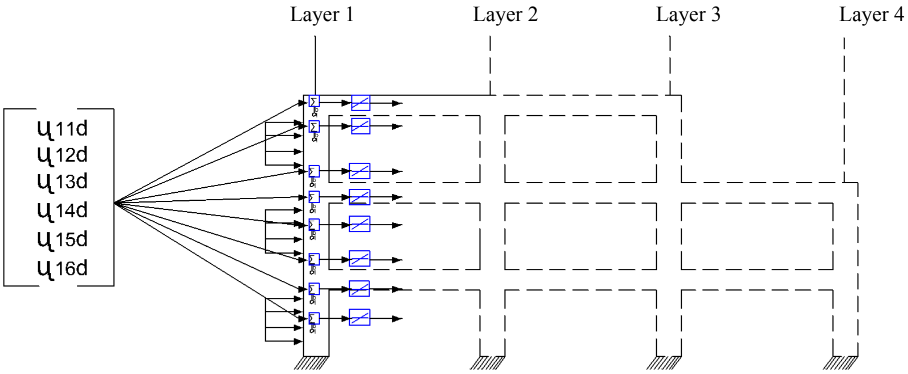
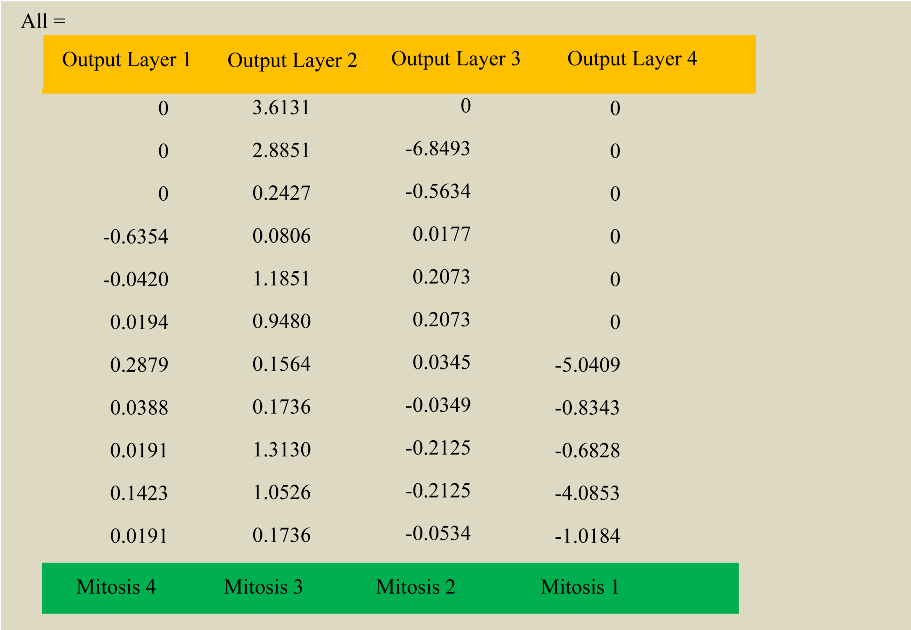


Figure 22. Mitosis model 3—out of the second layer.





All =

Output Layer 1	Output Layer 2	Output Layer 3	Output Layer 4
0	3.6131	0	0
0	2.8851	-6.8493	0
0	0.2427	-0.5634	0
-0.6354	0.0806	0.0177	0
-0.0420	1.1851	0.2073	0
0.0194	0.9480	0.2073	0
0.2879	0.1564	0.0345	-5.0409
0.0388	0.1736	-0.0349	-0.8343
0.0191	1.3130	-0.2125	-0.6828
0.1423	1.0526	-0.2125	-4.0853
0.0191	0.1736	-0.0534	-1.0184
Mitosis 4	Mitosis 3	Mitosis 2	Mitosis 1

Figure 24. Results—rotational moments at the ends of the members.

M_{ik}, M_{ki}: are the end known as the end moment and exerted by the node *i*, and *k* on the corresponding ends of the element, ME_{ik}, ME_{ki}: are fixed end moments induced at the ends, M'_{ik}: is termed as the near end rota-

Figure 23. Mitosis model 4—output first layer.

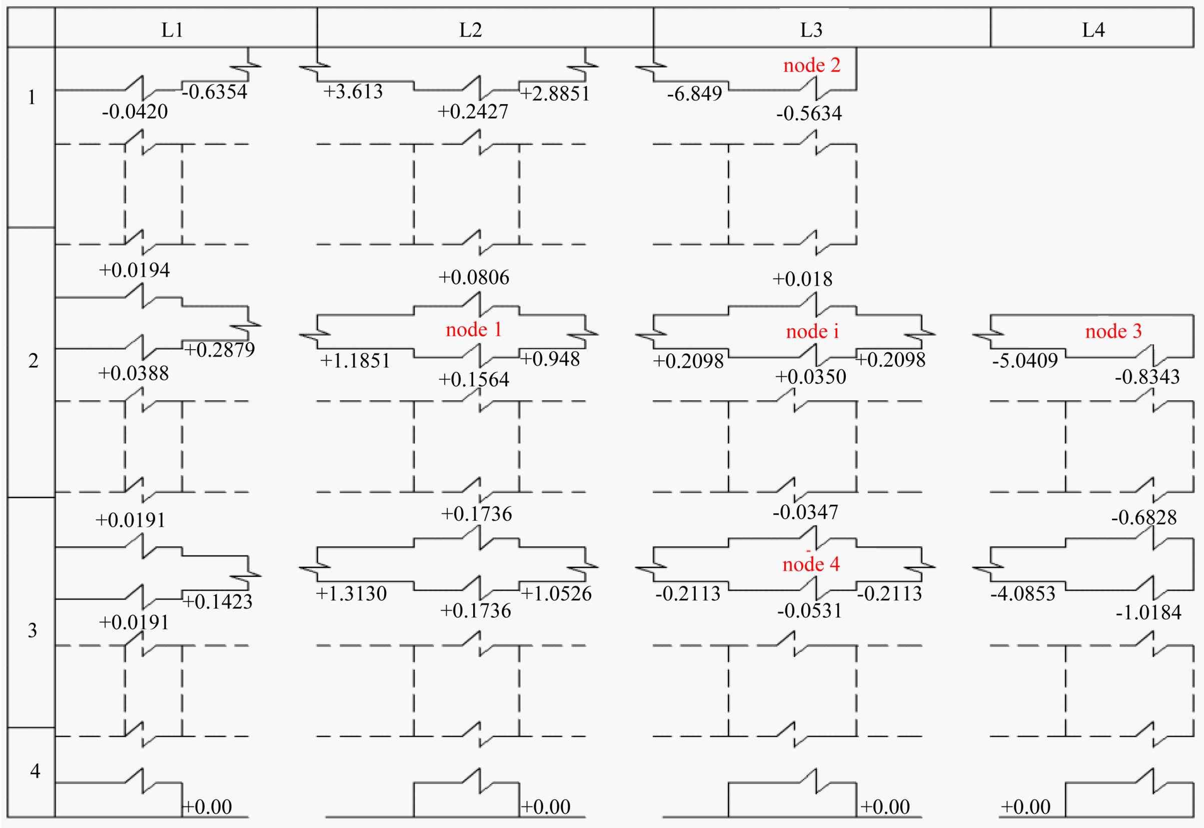


Figure 25. Rotational moments ends in the porch.

tional moment, and M'_{ki} : as the far end rotational end moment.

Applying the static equilibrium equations we have: $\sum_{j=1}^{4} M_{ij} = 0$

Considering the node *i* of the of the structure, **Figure 26**, of the structure, located between the three layer and level two, shown in **Figure 19**, where the members: $i-1, i-2, i-3, i-4$:

$$M_{i1} = +11.80 + 2\times 0.2098 + 0.948 = +13.1667t - m$$

$$M_{i2} = 2\times 0.018 - 0.5634 + 0.12\ (\text{is the total displacement moment}) = -0.4074t - m$$

$$M_{i3} = -8.60 + 2\times 0.2098 - 5.0409 = -13.2213t - m$$

$$M_{i4} = 2\times 0.0350 - 0.0347 + 0.42\ (\text{total displacement moment}) = +0.4553t - m$$

$$\sum_{j=1}^{4} M_{ij} = (+13.1667 - 0.4074 - 13.2213 + 0.4553) \approx 0$$

The validity of the computed the static equilibrium conditions, for node *i* is verified.

6.6.2. Comparison with Kani's Method

A comparison between the presented NTSA model and Kani's Method was performed on the same example, and it can be shown in **Figure 25** and **Figure 26**. A discrepancy ratio $D_r \frac{C_N}{C_K}$ was used for comparison, where C_N the rotational moment (output network) or total end moment and C_K is Kani's result. The mean value was performed e comparison is shown in **Table 1** and **Table 2**. From looking at the table, one may conclude that the presented model gives a better agreement with Kani's Method. A group of 15 nodes was used for verification. **Figure 27** and **Figure 28** show the nodes of the structure under study, and also the members who access the

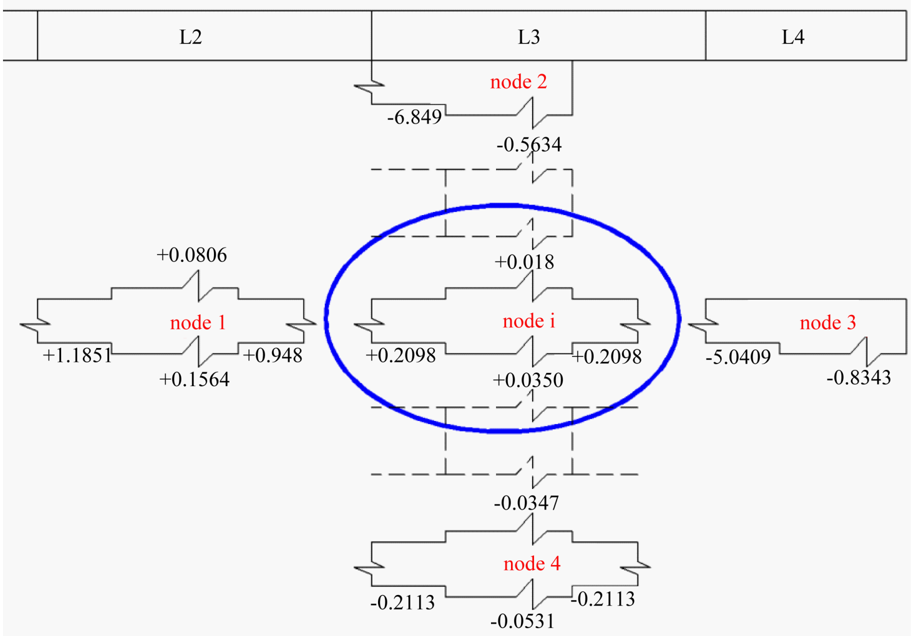


Figure 26. Node *i* and 1, 2, 3 and 4 equilibrium conditions.

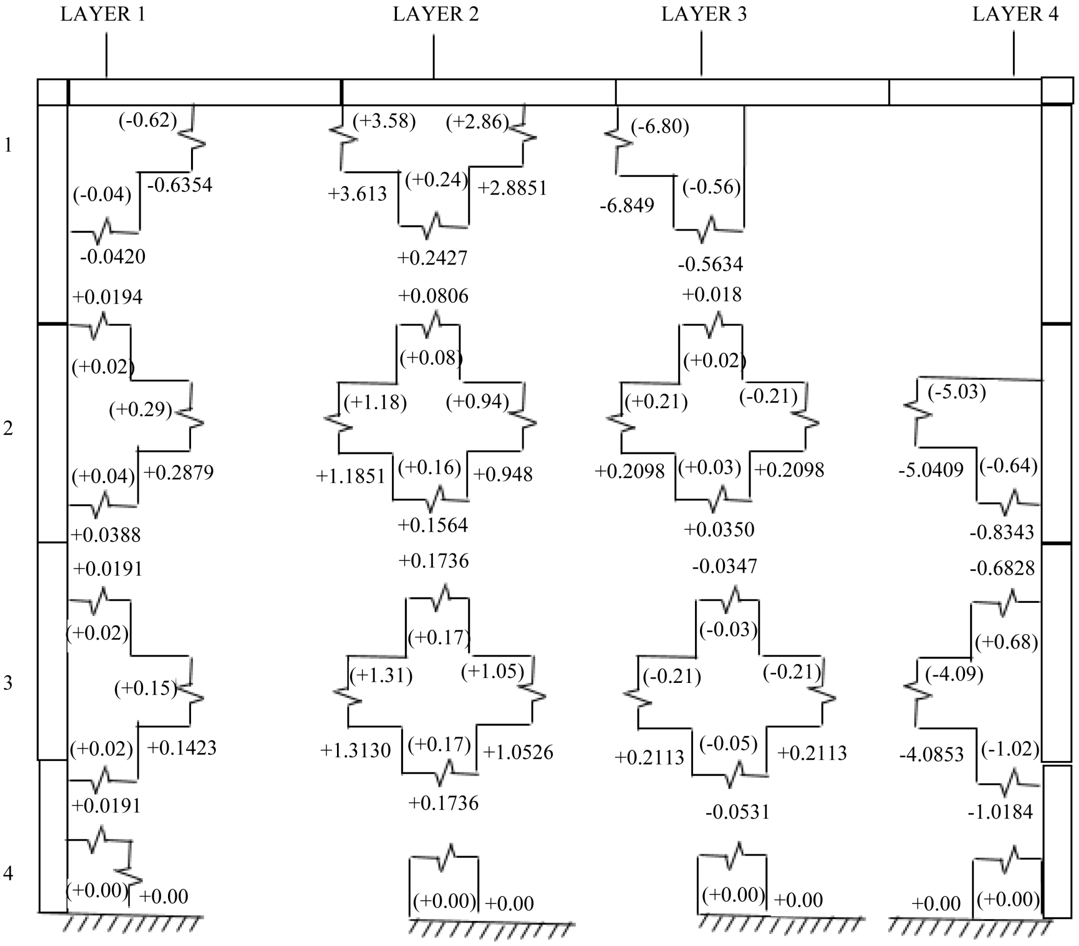
Table 1. Accuracy of formulas for rotational end moments.

Nodes	Layers	Statiscal Mean Value	Statiscal Standard Deviation
Nodes 1s	1	1.0375	0.0178
	2	1.0097	0.0013
	3	1.0067	0.0008
Nodes 2s	1	0.9776	0.0131
	2	0.9995	0.0147
	3	1.0016	0.1107
	4	0.9976	0.0036
Nodes 3s	1	0.9529	0.0036
	2	1.0118	0.0108
	3	1.0578	0.0710
	4	1.0005	0.0032
Nodes 4s	1	1.0000	0.0000
	2	1.0000	0.0000
	3	1.0000	0.0000
	4	1.0000	0.0000

aforementioned nodes. In these figures we present the comparison of some of the results: rotational moments at the extremes of members obtained with the application of G. Kani Method (distribution moments: these are the values shown in parentheses), and the results of the rotational moments calculated at the ends on the bars of the structure through the analysis executed by NTSA (whose values are outside the parentheses) that can improve the accuracy and speed of the results (Ben Mark) [9].

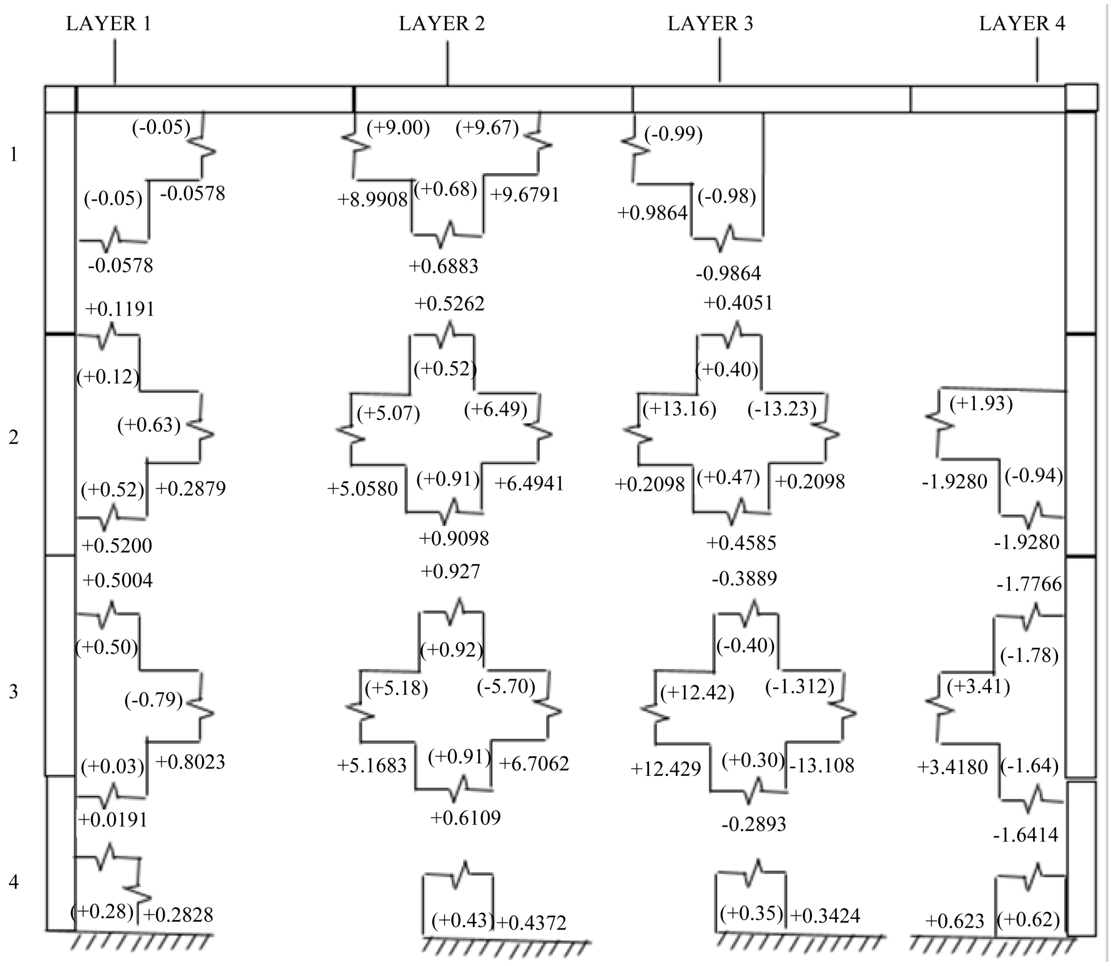
Table 2. Accuracy of formulas for total end moments.

Nodes	Layers	Statiscal Mean Value	Statiscal Standard Deviation
Nodes 1s	1	0.9633	0.0000
	2	1.0289	0.0412
	3	1.0015	0.0054
Nodes 2s	1	1.0023	0.0112
	2	1.0025	0.0064
	3	0.9970	0.0155
	4	0.9964	0.0037
Nodes 3s	1	1.0075	0.0076
	2	1.0061	0.0091
	3	0.9841	0.0186
	4	1.0040	0.0023
Nodes 4s	1	1.0100	0.0000
	2	1.0167	0.0000
	3	0.9783	0.0000
	4	1.0048	0.0000



Kani's Method: ( ): NTSA System: --------

Figure 27. Rotational end moments whit horizontal displacement.



Kani's Method: ( ): NTSA System: --------

Figure 28. Total end moment's whit horizontal displacement.

7. Conclusions

1) There is a knack for designing primitive network inspired in a simple lattice model of the proposed structure, since this automatically suggests the neural network topology, connectivity type and even the number of hidden layers and the number of constituent neurons.

2) The design of the various models of artificial neural networks is highly didactic, and from the primitive model the topology design of each NTSA is obtained, including the number of hidden layers, number of neurons per layer, and the way that these are connected.

3) In general, the algorithm is self-supervised learning NTSA common to all models. They differ in the judgments that define the individual architecture.

4) The NTSA models are connected in series allowing interaction between artificial neural network architecture with different connectivity (feed-forward-backward).

5) All NTSA models are linear, which significantly reduces the complexity calculations.

6) In conventional neural networks results hidden layers are not known; through the NSTA system we can know the values or results of the hidden layers of the network.

7) NTSA system introduces a new algorithm based on biology.

Acknowledgements

L.R.P.M. thanks to Prof. Alfonzo G. Cerezo for his advice developing this work.

References

[1] Pinto, L.R. and Zambrano, A.R. (2014) Unsupervised Neural Network Approach to Frame Analysis of Conventional Buildings. *Int. J. Communications*, *Network and Systems Sciences*, **7**, 203-211. http://dx.doi.org/10.4236/ijcns.2014.77022

[2] Rivero-Angeles, F.J., Gomez-Ramirez, E., Gomez-Gonzalez, B. and Garrido, R. (2005) Fault Detection in Shear Buildings Subject to Earthquakes Using a Neural Network. *Proceeding of the Eighth International Conference on the Application of Artificial Intelligence to Civil*, *Structural and Environmental Engineering*, Edited by B. H. V. Topping, 107, ISBN 1-905088-03-05. http://dx.doi.org/10.4203/ccp.82

[3] Pinto, L. (2008) Tesis Doctoral: AETN Analisis de Estructuras Mediante Topología Neuronal.

[4] DARPA (1987-1988) Neural Network Study (U.S.). Published by AFCEA International Press, a Division of the Armed Forces Communications and Electronics Association 4406 Fair Lakes Court Fairfax Virginia 22033-3899 USA.

[5] Beale, M., Hagan, M.T. and Demuth, H.B. (1995) Neural Network Design. Thomson Learning, Boston.

[6] Eaton, L.K. (2001) Hardy Cross and the "Moment Distribution Method". *Nexus Network Journal*, **3**, 15-24. http://dx.doi.org/10.1007/s00004-001-0020-y

[7] Hoffmann, F. Biological Therapies and Cancer. Produced through an educational grant from La Roche Ltd.

[8] Biological Foundations—Neuron Communication. www.studyblue.com/notes/note

[9] Kani, G. (1955) Cálculo de Pórticos de Varios Pisos. In: Reverte, S.A., Ed., 1978-1979, Printed in Spain, ISBN-84-291-2051-6, 19-20-21-22.

[10] Boso, D., Lefik, M. and Schnefler, B. (2005) Joint Finite Element: Artificial Neural Network Numerical Analysis of Multilevel Composites Artificial Intelligence to Civil, Structural and Environmental Engineering. Edited by B. H. V. Topping, Civil Comp. Ltd., 101, ISBN 1-905088-03-05.

[11] Lu, Y., Roychowdhury, V. and Vanderberghe, L. (2007) Distributed Parallel Support Vector Machines in Strongly Connected Networks. Neural Networks. A Publication of the IEEE Computational Intelligence Society.

[12] Bebbahani, S. and Nasrabadi, A.M. (2009) Application of Som Neural Network in Clustering.