ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 67-72

A General Model for Higher Order Neurons

Francisco J. Lopez-Aligue, Miguel A. Jaramillo-Moran, Isabel Acevedo-Sotoca and Montserrat G. Valle.

Departamento de Electronica. Universidad de Extremadura.
Avda. de Elvas, s/n - 06071 - Badajoz (Spain)

Abstract. In this work we present a new methodology for describing the functioning of artificial neurons, including new, as yet untested, types of behaviour. It also provides the possibility of defining artificial neurons of any order, not only first and second, and a wide range of functions from which to choose. As illustration of the new formulation, a practical realization of the new formulation is analyzed, consisting of a multilayered neural network applied to the image processing of black and white scenes, making manifest the possibilities of this new type of neuron in the field of cellular logic but with new types of processing.
This is just an early stage in the development of the new neurons, so that many of their possible applications have yet to be initiated. Among them, one can already foresee those related to fuzzy models, analogue models, and many others. Now, for these applications, it will no longer be necessary to make any change in the network design, but just to make a choice from the proposed library of functions.

1. Introduction

With the evolution of neural networks, new, ever more complex, fields of applications have been opening up. Many problems are beginning to be approached by comparing how they are solved by means of neural networks with their resolution using classical methods (when they exist). For instance, in the partitioning of hypercubes, nets of linear threshold neurons have been compared with classical logical implementations (Andrée et al.).
In other cases, the solution has consisted in developing second and even higher order functions (Simpson; Xu and Tsai), which have received extensive theoretical treatement but present serious difficulties in their practical implementation (Lopez et al.). Nonetheless, despite their potential lack of ease of handling, models of higher order (mainly second) offer clear advantages over those of first order as to their learning capacity and practical interest.
In next sections of this paper, we therefore present a new mathematical approach to these higher order models remembering that they have already allowed image treatement processes to be performed in the first stages of practical application when they had previously seemed reserved for other methodologies such as cellular logic (Rosenfeld). The latter has currently been the object of attention because of its possible synthesis by means of neural networks (Chua & Yang; Chua), in which there has also been a powerful incidence of the use of higher order neural models (Schmidt & Davis; Seiler et al.), given the evident similarity between the two structures.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 67-72

2. A General Neural Model

In most problems which one attemps to solve by means of neural networks, the road to the solution is always by way of establishing the function describing the behaviour of the system or the structure of the connections between the different blocks of the system: external inputs to the layer, lateral interaction (intra-layer connectivity) and connections between layers (feedforward, feedback, ..., etc.) or, finally, the learning schemes used.
Here, we shall centre on the first aspect, presenting a general mathematical model of the neuron's behaviour. The said model can be molded to whatever practical realization one likes, so that each application will be the result of specifying the values that correspond to a series of coefficients. In addition, many new neural functions can be built with this model, so that these neurons present an interesting new set of properties reserved, until today, to other fields of data processing.
As a starting point, our model must enable one to set up equations of second order or higher in such a way that, having chosen the desired order, we can decide whether one wishes to include the factors of lower order than that chosen or not, and, in the affirmative case, to be able to include them with the most convenient mathematical expression so that no a priori restriction is established.
The conventional formulation of the mathematical models as a function of their order is in the following form:
First order:

$$y_i(t+1) = \Psi \{ C_{nrm} [\sum_{j=1}^{N} m^i_j x_j - \theta_i \} \quad (1)$$

Second order:

$$y_i(t+1) = \Psi \{ C_{nrm} [\sum_{j_1,j_2=1}^{N} m^i_{j_1 j_2} x_{j_1} x_{j_2} - \theta_i \} \quad (2)$$

where the function Ψ is usually chosen to be either nonlinear and monotonically increasing, such as the popular sigmoid function, or with a sharp change, such as the threshold function, depending on the type of inputs being dealt with (continuous, binary, etc.). The constant C_{nrm} is chosen to normalize the function, and can in many cases be taken equal to unity. In practical applications, this formulation is found to be excessively rigid, so that we have to define clearly both the type of interconnections and the learning algorithm chosen to modify the coefficients "m" .
This rigidity can be avoided if we use, for all orders, the following definition:

$$F_{ordP} = C_{ordP} [\sum_{j_1,j_2,\ldots,j_P=1}^{N} m^i_{j_1,j_2,\ldots,j_P} \varphi^P_{j_1}[x_{j_1}, \varphi^P_{j_2}[x_{j_2}, \ldots]] - \theta_P] \quad (3.1)$$

Thus, we can take as neuron function

$$y_i(t+1) = \Psi \{ \varphi_{ord1} [F_{ord1}, \varphi_{ord2} [F_{ord2}, \ldots]] \} \quad (3.2)$$

where the elements "φ" in the two expressions correspond to a function that can be

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 67-72

chosen from among a wide group of the same, corresponding to the functions that are usually employed in neural networks, as listed in Table I.

Unit :	If $x_i , x_j \in [0,1]$,	$\varphi_{UNIT} [x_i , x_j] = x_i$
Binary AND :	If $x_i , x_j \in \{0,1\}$,	$\varphi_{BAND} [x_i , x_j] = x_i \cdot x_j$
Binary OR :	If $x_i , x_j \in \{0,1\}$,	$\varphi_{BOR} [x_i , x_j] = x_i + x_j$
Binary EXOR :	If $x_i , x_j \in \{0,1\}$,	$\varphi_{BXOR} [x_i , x_j] = x_i \oplus x_j$
Analog MULT:	If $x_i , x_j \in [0,1]$,	$\varphi_{AMUL} [x_i , x_j] = x_i * x_j$
Analog ADD:	If $x_i , x_j \in [0,1]$,	$\varphi_{AADD} [x_i , x_j] = x_i + x_j$
Fuzzy AND:	If $x_i , x_j \in [0,1]$,	$\varphi_{FAND} [x_i , x_j] = \min (x_i , x_j)$
Fuzzy OR :	If $x_i , x_j \in [0,1]$,	$\varphi_{FOR} [x_i , x_j] = \max (x_i , x_j)$
Fuzzy EXOR:	If $x_i , x_j \in [0,1]$,	$\varphi_{FXOR} [x_i , x_j] = \lvert x_i - x_j \rvert$

Table I. The set of different functions φ.

In almost all cases developed up to the present, the second order neural model formulations have been performed by using exclusively the analogue multiplication (AMUL), setting the whole burden of the specific implementation to be simulated on either the calculation of the coefficients (weights) "m" by means of different adaptive processes or on the structure of the networks. This is, of course, a case that the present formulation also includes, so that a conventional first and second order model is obtained by simply choosing the following selection :

$$\varphi_{ord1} = \varphi_{AADD}; \varphi_{ord2} = \varphi_{UNIT}; \varphi_{ordP} \text{ arbitrary } \forall P>2$$

$$\varphi^1_1 = \varphi_{UNIT} ; \varphi^1_P = \text{arb. } \forall P>1; \varphi^2_1 = \varphi_{AMUL}; \varphi^2_2 = \varphi_{UNIT}; \varphi^2_P = \text{arb. } \forall P>2$$

$$\Psi = \text{sigmoid function } \sigma$$

$$y_i (t+1) = \sigma \{ C_{ord1} [\sum_{j=1}^{N} m^i_j x_j - \theta_1] + C_{ord2} [\sum_{j,k=1}^{N} m^i_{jk} x_j(t) x_k(t) - \theta_2] \} \quad (4)$$

As to the coefficients "m" and connectivity diagram that are used, they will have to be chosen in a form that is suited to the case at hand.
Equations 3.1. and 3.2 together with the "library" of functions "φ", permit the design of neural networks which function in a wholly different way from those used up to now. Thus, even while keeping a given connectivity scheme and having a learning algorithm for the "m" coefficients, it will be enough to change some of the φ's for the network to execute totally different functions. This situation is particularly interesting when one is dealing with a software implementation of high complexity networks. The end program is unique, and the functions are assigned arbitrarily to the different parts into which the network is to be subdivided. Consequently, an indefinitely repeated neural structure can perform quite different functions with no more than choosing the φ's from among those described above.

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 67-72

3. Cellular Neural Networks

Using the notation proposed by Chua and Yang, it is possible to construct cellular neural networks with the formulation that we present here with no more than choosing the functions suitably. In principle, we employ the same concept of neighborhood of any cell for the conventional type of rectangular pixel arrangement. We choose the functions φ from Table I to work with the pixel values. For the case of the data of Table II,

$$\varphi_{ord1} = \varphi_{AADD};\ \varphi_{ord2} = \varphi_{UNIT};\ \varphi_{ordP} \text{ arbitrary } \forall\ P>2$$

$$\varphi_1^1 = \varphi_{UNIT};\ \varphi_P^1 = \text{arb. } \forall\ P>1;\ \varphi_1^2 = \varphi_{BXOR};\ \varphi_2^2 = \varphi_{UNIT};\ \varphi_P^2 = \text{arb. } \forall\ P>2$$

$$\Psi = \text{sigmoid function } \sigma$$

Table II. Functions choice for the imlementation of cellular neural logic

to obtain the function

$$y_i(t+1) = \Psi \{ C_{ord1} [\sum_{j=1}^{N} m_j^i\ x_j - \theta_1] + C_{ord2} [\sum_{j,k=1}^{N} m_{jk}^i [x_j(t) \oplus x_k(t)] - \theta_2 \} \quad (5)$$

Observe that, in this equation, since $x_j \in \{0,1\}$, $\varphi_{BXOR} = \varphi_{FXOR}$, which will be of great importance when it comes to dealing with fuzzy functions.
Applying this function to the neighborhood of radius 1 (nearest surrounding cells) so that inputs to the neuron are taken to be the values of all the pixels of the neighborhood, and using the coefficients of Table III,

$$\theta_1 = 2,\ \theta_2 = 4,\ C_{ord1} = 0.5, C_{ord2} = 5,\ \Psi = \text{Threshold function},\ m_{jk}^i = m_j^i = 1$$

Table III. Selected values for image processing

one obtains a true cellular image processing system with noise filter capabilities and, also, by modifying the connections structure trough "m", spatial filtering.

4. Stability of the Network

Grouping θ_1 and θ_2 in a unique θ_i term in (5) and makig $C_{ord1} = C_{ord1} = 1$, we get, in the case of second order, the following as the digital neuron function

$$y_i(t+1) = \Psi \{ \sum_{j=1}^{r} m_j^i\ y_j(t) + \sum_{j=1}^{r} m_{jk}^i\ \varphi(y_j(t), y_k(t)) - \theta_i \} \quad (6)$$

Which, with the following restrictions, defines a convergent system:

ESANN'1994 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 20-21-22 April 1994, D-Facto public., ISBN 2-9600049-1-4, pp. 67-72

- The coefficients m_j^i and m_{jk}^i are constants with the suffix "k" representing the next neuron, proceeding in a clockwise direction, within a determined radius. The connection scheme is the same for all the neurons.
- $m_j^i = m_i^j$ and $m_i^i = 0$. The coefficients m_{jk}^i must be such that:

$$\text{sign}(m_{jk}^i) = \text{sign}(m_k^i) = \text{sign}(m_j^i) \quad \text{and} \quad m_{jk}^i = 0 \text{ if } \text{sign}(m_j^i) \neq \text{sign}(m_k^i) \tag{7}$$

- "φ" is an increasing logic function, i.e.,

$$\varphi(y_j(t+1),y_k(t)) \geq \varphi(y_j(t),y_k(t)) \quad \text{if} \quad \Delta y_j > 0$$
$$\varphi(y_j(t),y_k(t+1)) \geq \varphi(y_j(t),y_k(t)) \quad \text{if} \quad \Delta y_k > 0$$

The energy function will be of the form:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{r} m_j^i\, y_j(t)y_i(t) - \sum_{i=1}^{N}\sum_{j=1}^{r} m_{jk}^i\, \varphi(y_j(t),y_k(t))y_i(t) + \sum_{i=1}^{N} \theta_i\, y_i(t) \tag{8}$$

The increment in energy produced as the i-th neuron goes from a state at time "t" to another at time "t+1" will be given by

$$\Delta E = -\Big[\sum_{j=1}^{r} m_j^i\, y_j(t) + \sum_{j=1}^{r} m_{jk}^i\, \varphi(y_j(t),y_k(t)) - \theta_i\Big]\Delta y_i - \sum_{j=1}^{r} m_{ik}^j\, [\varphi(y_i(t+1),y_k(t)) -$$

$$\varphi(y_i(t),y_k(t))]y_j(t) - \sum_{j=1}^{r} m_{hi}^j\, [\varphi(y_h(t),y_i(t+1)) - \varphi(y_h(t),y_i(t))]y_j(t) \tag{9}$$

where the suffix "h" represents the neuron before the i-th, counting clockwise as before. The term $\sum_{j=1}^{j} m_j^i\, y_j(t)$ is obtained by taking $m_j^i = m_i^j$ and $m_i^i = 0$. The expression in brackets is the value of $y_i(t+1)$ which will be positive when Δy_i is positive ($y_i(t) = 0$ and $y_i(t+1) = 1$) and negative when Δy_i is negative ($y_i(t) = 1$ and $y_i(t+1) = 0$). Therefore, the term in brackets multiplied by Δy_i will always be positive and its contribution to the energy increment negative.
With respect to the other two terms, one must bear in mind that:

1.- Due to (7), the second-order term of $y_i(t+1)$ will have the same behaviour as the first-order term, i.e., the two will increase or decrease together.
2.- The sign of m_{jk}^i, together with the fact that $m_j^i = m_i^j$ and $m_i^i = 0$, means that an

increase in the value of the y_i is transmitted to the other neurons to which it is connected as an increase in the activity of those it excites and a decrease of those it inhibits. This behaviour comes back onto y_i as an increase in its activity. Likewise, a decrease in the activity of the neuron being considered comes back on that neuron from the others that it affects as a decrease in activity. Therefore, given that "φ" has been defined as an increasing function, it can be stated that the aforementioned terms in ΔE also contribute negatively. Since the function E is bounded (as can easily be deduced from its definition) and decreasing, the system described by $y_i(t+1)$ will thus evolve towards stable states.

Acknowledgement

The authors wish to express their gratitude to the Spanish C.I.C.Y.T. for its support to this work through Grant TIC - 92 - 054.

References

1. Andree, H.M.A., Barkema, G.T., Lourens, W., & Taal, A. A comparison study of binary Feedforward neural networks and digital circuits. *Neural Networks*, **6**, 785-790 (1993).
2. Chua, L.O. & Yang, L. Cellular neural networks: Theory.*IEEE Transactions on CAS*. **35**, 10, 1257 - 1272. (1988)
3. Chua, L.O. Cellular neural networks: Applications. *IEEE Transactions on CAS*. **35**, 10, 1273-1290. (1988).
4. Dembo, A., Farotini, O. & Kailath, T. High order absolutely stable neural networks. *IEEE Transactions on CAS*, **38**, 1, 57-65. (1991).
5. Lopez-Aligue, F.J., Acevedo-Sotoca, M.I., & Jaramillo-Moran, M.A. *A high order neural network*. Lecture Notes on Computer Science, **686**, 108-113, Berlin: Springer-Verlag. (1993).
6. Psaltis, D., Park, C.H., & Hang, J. (1988). Higher order associative memories and their optical implementation. *Neural Networks*, **1**, 149-163.
7. Rosenfeld, A., & Kak, A.K. *Digital picture processing*. New York : Academic Press.(1982).
8. Schmidt, W.A.C., & Davis, J.P. Pattern recognition properties of various feature spaces for higher order neural networks. *IEEE Trans. on PAMI*, **15**,8,795-801. (1993).
9. Seiler, G., Schuler, A.J. & Nossek, J.A. Design of robust cellular neural networks. *IEEE Trans. on CAS*, **40**, 5, 358-364. (1993).
10. Shawe-Taylor, J., Anthony, M.H.G., & Kern, W. Classes of feedforward neural networks and their circuit complexity. *Neural networks*, **5**, 971-977.(1992).
11. Simpson, P. (1990). Higher-ordered and intraconnected bidirectional associative memories.*IEEE Trans. on System, Man and Cybernetic*, **20**, 3, 637-653.
12. Xu, X., & Tsai, T.Constructing associative memories using neural networks. *Neural Networks*, **3**, 301-309. (1990).