Feed-forward networks composed by neurons with activation functions of different parity

1990 J. Phys. A: Math. Gen. 23 5045

(http://iopscience.iop.org/0305-4470/23/21/039)

J. Phys. A: Math. Gen. 23 (1990) 5045-5060. Printed in the UK

Feed-forward networks composed by neurons with activation functions of different parity

Edgardo A Ferrán and Roberto P J Perazzo

Departamento de Física (TANDAR), Comisión Nacional de Energía Atómica, Av Libertador 8250, 1429 Buenos Aires, Argentina

Received 12 February 1990

Abstract. We study feed-forward networks of formal neurons having even activation functions. We show that networks of this kind have different computational properties than the ones with neurons having odd processing functions. We show that networks containing mixtures of this two types of neurons have richer representability properties. We extend our results to cases of discrete processing. These properties have been checked in numerical simulations performed in small enough systems to allow for an explicit enumeration of all synaptic matrices and Boolean functions.

1. Introduction

In biological systems, each neuron produces a train of pulses on its outgoing axon that depends upon the polarization voltage established in the membrane of the soma. The maximum value of the frequency of emission of such pulses is limited by the refractory period τ_e, that is the minimum time interval allowed between two successive pulses. This frequency is related to the total intensity of the stimulus that the neuron receives from all the others. Such a relationship is represented by an *activation function* that is assumed to be embodied in each cell. Typically this function is nonlinear, bisaturating and bounded between the values 0 and $1/\tau_e$.

One model approaches this biological situation by introducing formal neurons that are bistable devices that switch from one state to the other depending on the sign of the total stimulus that is produced by all the neurons that are connected to it [8]. A second approach that is closer to biological systems assumes that the state of each neuron is represented by the time average of the emission frequencies of action potentials produced on its outgoing axon. The corresponding activation function is usually considered as a continuous odd function of the difference between the intensity of the total stimulus and an internal firing threshold of the neuron.

Large sets of formal neurons may be assumed to be organized as feed-forward networks with one or more layers of neurons. These are adaptable systems that can learn to represent a given Boolean function from a partial knowledge of it [1, 7, 9, 11, 12]. The learning procedure amounts to a change of the synaptic connections between the neurons and the firing thresholds following a prescribed algorithm.

In a preceding paper [3] we have studied special types of feed-forward networks, namely those composed by neurons having all the same activation function and the same firing threshold. We showed that this function determines to a large extent the representability properties of the whole system. In fact, if it is assumed to be odd and

0305-4470/90/215045+16$03.50

all the firing thresholds are set equal to zero, a rather limited class of Boolean functions may be represented. In this paper we deal with networks composed by neurons having other types of activation functions with emphasis for the case in which this is even. Neurons of this type have a firing regime in which, for the case of graded response, the maximum output signal is produced only for zero stimulus and the output signal gradually decreases for larger values of it regardless of its excitatory or inhibitory nature. Although this regime is not close to biological evidence the subject deserves attention since networks built with this type of neurons have radically different computational properties. In the present paper we investigate this point together with the properties of networks in which both kind of neurons are present.

2. Description of the network

In this section we briefly describe the structure of the networks that we are going to consider, following closely the notation of reference [3]. The nets process N_{in} external binary inputs to produce N_{out} external binary outputs. These are represented by the vectors:

$$\boldsymbol{E} = (E_1, E_2, \ldots, E_{N_{\mathrm{in}}}) \tag{1}$$

$$\boldsymbol{S} = (S_1, S_2, \ldots, S_{N_{\mathrm{out}}}) \tag{2}$$

with E_i and $S_i \in \{0, 1\}$. There are $N_e = 2^{N_{\mathrm{in}}}$ different binary input vectors $\boldsymbol{E}^{[j]}$, $(0 \leq j \leq N_e - 1)$. $\boldsymbol{S}^{[j]}$ is the output vector produced when the network is fed with $\boldsymbol{E}^{[j]}$. The network represents a particular Boolean function F, out of $N_{\mathrm{Bool}} = 2^{N_{\mathrm{out}} N_e}$. The truth table of F can be built with the arrays:

$$|E\rangle = \left| \begin{matrix} \boldsymbol{E}^{[0]} \\ \boldsymbol{E}^{[1]} \\ \vdots \\ \boldsymbol{E}^{[N_e - 1]} \end{matrix} \right\rangle = \left| \begin{matrix} 00\ldots0 \\ 00\ldots1 \\ \vdots \\ 11\ldots1 \end{matrix} \right\rangle \qquad \text{and} \qquad |S\rangle = \left| \begin{matrix} \boldsymbol{S}^{[0]} \\ \boldsymbol{S}^{[1]} \\ \vdots \\ \boldsymbol{S}^{[N_e - 1]} \end{matrix} \right\rangle \tag{3}$$

The network is assumed to have N formal neurons distributed in L layers, with n_ν neurons in the νth layer. We consider these neurons as processing elements with no limits imposed in their fan-in and in their fan-out. The ith neuron of the λth layer is assumed to produce an output signal V_i^λ that is a nonlinear, bounded function g_i^λ of the difference between the sum u_i^λ of all the weighted input signals that the neuron receives and its threshold θ_i^λ:

$$V_i^\lambda = g_i^\lambda (u_i^\lambda - \theta_i^\lambda) \tag{4}$$

$$u_i^1 = \sum_{j=1}^{N_{\mathrm{in}}} J_{ij}^{10} I_j \tag{5}$$

$$u_i^\lambda = \sum_{j=1}^{N_{\mathrm{in}}} J_{ij}^{\lambda 0} I_j + \sum_{\nu=1}^{\lambda-1} \sum_{j=1}^{n_\nu} J_{ij}^{\lambda\nu} V_j^\nu \qquad \forall \lambda = 2, \ldots, L. \tag{6}$$

In principle each of the neurons of the system can embody different functions g_i^λ. The models mostly studied in the literature assume however that all neurons of a given system have the same g_i^λ (homogeneous networks) and that this is an odd function of its argument. The reason for these assumptions lies principally on the biological experience that finds no essential difference among the elementary constituents of the central nervous system. In the present paper we chiefly consider cases in which g_i^λ is odd or even:

$$g_i^\lambda(x) = g_{i\lambda}^{[o]}(x) = -g_{i\lambda}^{[o]}(-x) \tag{7}$$

$$g_i^\lambda(x) = g_{i\lambda}^{[e]}(x) = g_{i\lambda}^{[e]}(-x). \tag{8}$$

Although in general $g_{i\lambda}^{[o]}$ and $g_{i\lambda}^{[e]}$ may involve a different bias for each neuron we consider this to be zero for all the elements of the system ($\theta_i^\lambda = 0, \forall i, \lambda$).

The external input signals E_i are transformed into the afferent currents I_i and fed into the neurons of the system as indicated in (5) and (6). A convention must be chosen to relate E_i with I_i. In order to take advantage of the whole dynamic range of the activation functions we consider:

$$I_j = \varphi_e(E_j) = 2E_j - 1. \tag{9}$$

The set of all the afferent currents I_j corresponds to a 0th layer.

The latin subindices i, j of the synaptic efficacies $J_{ij}^{\lambda\nu}$ in (5) and (6) label, respectively, the neurons receiving and producing the signal. The two greek indices λ, ν number the layers that respectively contain those neurons. The neurons that belong to a given layer can only feed signals to any neuron of the subsequent ones. The synaptic matrix J is therefore organized in an $L \times L$ block form, with each block containing the connections between two layers, and with blocks of vanishing matrix elements above the diagonal. The matrix J has L sets of columns and L sets of rows. The νth set of columns corresponds to signals coming from the $(\nu - 1)$th layer ($\nu = 1, \ldots, L$), and the λth set of rows corresponds to neurons belonging to the λth layer ($\lambda = 1, 2, \ldots, L$). We assume that the matrix elements of the non-zero blocks of J can have discrete and bounded positive or negative values as well as zero corresponding respectively to excitatory or inhibitory synaptic connections and to non-connected neurons.

Each layer of neurons may process not only the external input signals but also the output signals generated by neurons of any of the previous layers. We refer to this architecture as fully connected networks. In a more restricted case each layer can only feed its output signals to the subsequent one. We refer to this architecture as cascade networks.

For graded response neurons the binary values of the external signals S are obtained by filtering the outputs produced by the last layer of neurons. We consider two possible conventions for this filtering:

$$\varphi_S^-(V_i^L) = \begin{cases} 0 & \text{if } V_i^L \leqslant 0 \\ 1 & \text{if } V_i^L > 0 \end{cases} \tag{10}$$

$$\varphi_S^+(V_i^L) = \begin{cases} 0 & \text{if } V_i^L < 0 \\ 1 & \text{if } V_i^L \geqslant 0. \end{cases} \tag{11}$$

3. Representability properties

The networks such as the ones we are considering are adaptive systems that can be trained to reproduce a given Boolean function by adjusting the synaptic matrix. However, if only odd activation functions are considered not any function can be represented by the network. To check this it is enough to compare the outputs that can be obtained by feeding the same input to two networks of the same architecture but having synaptic matrices J and $\tilde{J}$ related to each other by:

$$\begin{cases} \tilde{J}_{ij}^{\lambda 0} = -J_{ij}^{\lambda 0} & \forall i, j, \lambda \\ \tilde{J}_{ij}^{\lambda \nu} = J_{ij}^{\lambda \nu} & \forall i, j, \nu > 0, \lambda > \nu. \end{cases} \tag{12}$$

It can be proved [3] that, if the processing function embodied in each neuron is odd and all the firing thresholds are equal to zero, then each component of the output vectors $\boldsymbol{S}$ and $\tilde{\boldsymbol{S}}$ must fulfil:

$$S_i \wedge \tilde{S}_i = 0 \qquad \text{for } \varphi_S = \varphi_S^-, \forall i \tag{13}$$

or:

$$S_i \vee \tilde{S}_i = 1 \qquad \text{for } \varphi_S = \varphi_S^+, \forall i \tag{14}$$

where $\wedge$ and $\vee$ denote the Boolean operators AND and OR, respectively. If (13) and (14) are not verified it is not possible to find a synaptic matrix J (and $\tilde{J}$) allowing the network to represent the corresponding function F (and $\tilde{F}$). The prescriptions (13) and (14) are necessary but not sufficient conditions for the representability of a given Boolean function.

The XOR cannot fulfil (13) or (14) and therefore cannot be represented by feed-forward networks such as the ones we are considering. This conclusion does not depend upon the number of layers or the number of neurons per layer in the network.

Within the assumption that the biases are kept all equal to zero, the limitations imposed by (13) and (14) depend critically upon the fact that the activation function is odd. We next study two cases in which this condition is relaxed, giving rise to networks with different representability properties.

3.1. Networks with discrete response neurons

Within the present framework we regard the processing of a discrete response neuron as the combination of a continuous internal activation function such as (7) or (8) with an output filtering. We denote these combinations as $g_{i\lambda}^{[e+]}$, $g_{i\lambda}^{[e-]}$, $g_{i\lambda}^{[o+]}$ and $g_{i\lambda}^{[o-]}$ (see figure 1).

Let us first consider the case of a single neuron with two binary inputs E_1 and E_2 and synaptic efficacies J_{1j}^{10} $(j = 1, 2)$ that can only take the values $+1$, -1 and 0. It is easy to check that for $g_{i\lambda}^{[o+]}$ and $g_{i\lambda}^{[o-]}$ each of the nine possible combinations of the pairs $(J_{11}^{10}, J_{12}^{10})$ is associated with a different Boolean function. The complete list is given in table 1 (see also (23) and (24) for how we denote each function).

The Boolean function F_7 in table 1 corresponds to the NAND operation between the inputs E_1 and E_2. It is well known that using elementary logic reduction formulae any Boolean function of two inputs can be expressed in terms of NAND gates. It therefore follows that any F can be represented by a feed-foward network composed by discrete response neurons†.

As shown in table 1, $g_{i\lambda}^{[o-]}$ and $g_{i\lambda}^{[o+]}$ are enough to represent fourteen of the sixteen Boolean functions of two inputs with only one neuron. The only missing functions are the XOR (F_6) and its logical negation (F_9). It can be checked that this situation does not change if one allows J_{ij}^{10} to take more values.

† Note that the universal Boolean representability achieved with discrete response neutrons is not in contradiction with (13) and (14) since $g_{i\lambda}^{[o+]}$ and $g_{i\lambda}^{[o-]}$ are not odd.

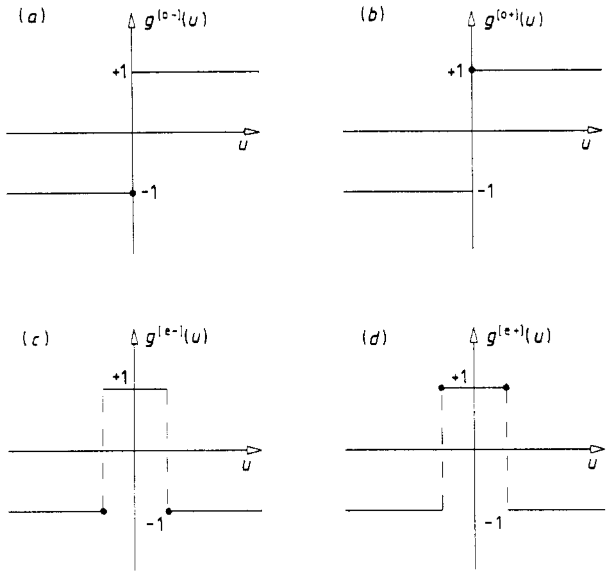

Figure 1. Discrete activation functions. (*a*) $g^{[o-]}(u)$, (*b*) $g^{[o+]}(u)$, (*c*) $g^{[e-]}(u)$ and (*d*) $g^{[e+]}(u)$.

Table 1. Boolean functions represented by a single neuron having two inputs and different discrete activation functions g. The values $J_{1j}^{10} = \pm 1$ are indicated only by the sign. The functions are denoted following the convention of (23) and (24).

J_{11}^{10}	−	−	+	+	0	0	−	+	0
J_{12}^{10}	−	+	−	+	−	+	0	0	0
$g^{[o-]}$	F_1	F_2	F_4	F_8	F_5	F_{10}	F_3	F_{12}	F_0
$g^{[o+]}$	F_7	F_{11}	F_{13}	F_{14}	F_5	F_{10}	F_3	F_{12}	F_{15}
$g^{[e-]}$	F_6	F_9	F_9	F_6	F_0	F_0	F_0	F_0	F_{15}
$g^{[e+]}$	F_6	F_9	F_9	F_6	F_0	F_0	F_0	F_0	F_{15}

The XOR function can be represented without a drastic increase† in the number of neurons if we introduce a third type of neuron with an even internal activation function $g_{i\lambda}^{[e+]}$ (or $g_{i\lambda}^{[e-]}$). We can see that, in this case, the filtering does not introduce an additional variety. With the conventions chosen hitherto these neurons can represent the functions F_6 and F_9 (see table 1). Therefore a network consisting of a single neuron of at most three different types of activation functions ($g_{i\lambda}^{o+]}$, $g_{i\lambda}^{[o-]}$ and $g_{i\lambda}^{[e+]}$), can represent all the sixteen Boolean functions of two inputs.

The set of the inverted activation functions $\{-g_{i\lambda}^{[o+]}, -g_{i\lambda}^{[o-]}, -g_{i\lambda}^{[e+]}\}$ are also an equally acceptable set for this complete representation.

By inspection of table 1 it is easy to check that any further constraint on the possible values of the synaptic matrix elements (such as only considering ± 1 or 0, 1) prevents to represent the complete set of Boolean functions.

† Note that to represent F_6 five NAND neurons distributed in three layers are needed.

3.2. Cascade architectures of even neurons

We now turn to the case of cascade networks composed by neurons having even continuous activation functions $g_{i\lambda}^{[e]}$. The corresponding synaptic matrix fulfils $J_{ij}^{\lambda\nu}=0$ if $\lambda \neq \nu+1$. The filtering is applied only to the outputs of the last layer of neurons.

We again consider, as in the beginning of this section, two networks with synaptic matrices related to each other by (12) and fed with the same input vector. We obtain

$$\tilde{V}_i^1 = g_{i1}^{[e]}(\Sigma_{j=1}^{N_{in}} \tilde{J}_{ij}^{10} I_j) = g_{i1}^{[e]}(-\Sigma_{j=1}^{N_{in}} J_{ij}^{10} I_j) = g_{i1}^{[e]}(\Sigma_{j=1}^{N_{in}} J_{ij}^{10} I_j) = V_i^1 \tag{15}$$

hence

$$\tilde{V}_i^2 = g_{i2}^{[e]}(\Sigma_{j=1}^{n_1} \tilde{J}_{ij}^{21} \tilde{V}_j^1) = g_{i2}^{[e]}(\Sigma_{j=1}^{n_1} J_{ij}^{21} V_j^1) = V_i^2 \tag{16}$$

and therefore

$$\tilde{V}_i^\lambda = V_i^\lambda \qquad \forall i, \lambda. \tag{17}$$

This is enough to prove that all components of the output vectors $\boldsymbol{S}$ and $\tilde{\boldsymbol{S}}$ must fulfil the condition:

$$S_i = \tilde{S}_i \qquad \forall i \tag{18}$$

no matter the filtering function that has been chosen for the network.

The property (18), as opposed to (13) and (14), means that cascade networks of identical 'even neurons' can only represent functions that are invariant under the change of ones and zeros in the input signals. The XOR and its logical inverted are precisely functions of this type.

As long as we only consider cascade architectures of homogeneous networks, the use of odd or even neurons do not exhaust the whole space $\mathcal{F}$ of Boolean functions. The fractions of $\mathcal{F}$ that fulfil respectively (13), (14) and (18) are:

$$r^{[o]} = 2\cdot(\tfrac{3}{4})^\eta - (\tfrac{1}{2})^\eta \tag{19}$$

$$r^{[e]} = (\tfrac{1}{2})^\eta \tag{20}$$

with

$$\eta = N_{out}\, 2^{N_{in}-1}. \tag{21}$$

These two sets intersect each other with the two functions that have as output all ones and all zeros. Therefore, the fraction of $\mathcal{F}$ that cannot be represented with either of the preceding type of nets tends to 1 for $\eta \to \infty$.

Mixed networks in which odd and even neurons coexist can represent some Boolean functions that are impossible to obtain with homogenous ones. For instance, if we take the case of a single layer with two neurons, fed with two external signals, it is simple to check that the function whose output array is

$$|S\rangle = \left|\begin{matrix} 01 \\ 11 \\ 10 \\ 00 \end{matrix}\right\rangle \tag{22}$$

cannot be represented if the activation functions of the two neurons are of the same kind ($g_{i\lambda}^{[o]}$ or $g_{i\lambda}^{[e]}$), but it may be obtained using an even neuron for S_1 and an odd neuron for S_2.

In the next section we present some numerical simulations to explore the representability properties of networks composed by even neurons in which is relaxed the restriction of having a cascade architecture. We also consider mixed networks in which odd and even neurons coexist.

4. Numerical simulations

In the present section we perform numerical simulations using networks that have few enough elements to allow for an exhaustive enumeration of all possible synaptic matrices and functions. We only consider $N_{\rm in} = N_{\rm out} = 2$. We thus have 256 possible functions F. The space $\mathscr{J}$ of all the synaptic matrices has been limited by requiring that the matrix elements $J_{ij}^{\lambda\nu}$ can only take the values -1, 0 or $+1$.

The output arrays $|S\rangle$ that denote any of the 256 possible Boolean functions F_k are written as:

$$|S_k\rangle = F_k|E\rangle = \left| \begin{matrix} b_1 b_2 \\ b_3 b_4 \\ b_5 b_6 \\ b_7 b_8 \end{matrix} \right\rangle \tag{23}$$

where

$$|E\rangle = \left| \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \right\rangle$$

and

$$k = \sum_{i=1}^{8} 2^{i-1} b_i. \tag{24}$$

These networks have symmetries under four possible elementary transformations (see the appendix for a brief review of the symmetry properties of feed-forward neural networks): (i) the permutation of the two external inputs, (ii) the permutation of the two output signals and (iii) the interchange of ones and zeros in each of the incoming signals.

These transformations are the generators of a finite, non-Abelian group G_S of sixteen transformations T_i, $(i = 1, 2, \ldots, 16)$. It is therefore possible to group the 256 elements of the space $\mathscr{F}$ into 34 different 'families' or symmetry classes $\mathscr{D}_\alpha$ of at most 16 elements each. When some of these symmetries are broken the number of functions in each class diminishes and the total number of classes becomes larger.

The existence of these symmetries allows us to perform a complete study of the space $\mathscr{F}$ by only concentrating on a few special cases. This also holds for the learning curves $\Pi^{(n)}(F)$ that we define as the average probability to represent correctly one given function F for all the N_e possible input values, provided it is properly reproduced for only n input values:

$$\Pi^{(n)}(F) = \binom{N_e}{n}^{-1} \sum_{\{(j_1, j_2, \ldots, j_n)\}} \frac{H[\mathscr{L}(F)]}{H[\mathscr{M}^{j_1, j_2, \ldots, j_n}(F)]}. \tag{25}$$

In (25) $\mathscr{L}(F)$ is the set of all synaptic matrices that represent the Boolean function F and $\mathscr{M}^{j_1,j_2,\ldots,j_n}(F)$ is the set of all synaptic matrices that represent a Boolean function that coincides with F for the inputs $E^{[j_k]}$ of the training set ($0 \leqslant j_k \leqslant N_e - 1$, $1 \leqslant k \leqslant n$). In (25) $H[\mathscr{P}]$ denotes the number of elements of the set $\mathscr{P} \subset \mathscr{J}$. An average is performed in (25) over all possible training sets of the same size to obtain a probability that depends only upon the number of inputs and not upon a particular composition of the training set. $\Pi^{(n)}(F)$ is usually called the probability of generalization and the plot of $\Pi^{(n)}(F)$ against the amount of training n is known as the learning curve of F [2, 3].

We have considered neurons having the following internal processing functions:

(*a*) discrete:

$$g_{i\lambda}^{[\mathrm{o}-]} = \begin{cases} -1 & \text{if } u_i^\lambda \leqslant 0 \\ 1 & \text{if } u_i^\lambda > 0 \end{cases} \tag{26}$$

$$g_{i\lambda}^{[\mathrm{e}-]} = \begin{cases} -1 & \text{if } |u_i^\lambda| \geqslant 0.5 \\ 1 & \text{if } |u_i^\lambda| < 0.5 \end{cases} \tag{27}$$

(*b*) graded:

$$g_{i\lambda}^{[\mathrm{o}]} = \tanh(u_i^\lambda), \tag{28}$$

$$g_{i\lambda}^{[\mathrm{e}]} = 2\mathrm{e}^{-(u_i^\lambda)^2/2} - 1. \tag{29}$$

In all the cases we use the filtering φ_S^-.

We describe first the simulations performed with graded neurons. We denote $R_i^{n_1,n_2,\ldots,n_L}$ the fully connected architecture of the network and $\tilde{R}_i^{n_1,n_2,\ldots,n_L}$ the cascade one. The subindex i is defined by:

$$i = \sum_{j=1}^{N} 2^{j-1} p_j \tag{30}$$

where $p_j = 1$ for even and $p_j = 0$ for odd activation functions, respectively. The neurons are numbered beginning with the first layer. For example, the fully connected architecture of the network composed by two odd neurons in a first layer and two even neurons in a second and last layer is denoted by $R_{12}^{2,2}$.

Each architecture is simulated with all possible synaptic patterns to obtain the numbers $H[\mathscr{L}(F)]$. The complete list is given in tables 2 and 3.

The cascade architectures $\tilde{R}_3^2$, $\tilde{R}_{15}^{2,2}$ and $\tilde{R}_{63}^{2,2,2}$ are homogeneous networks composed only by neurons having even activation functions. As expected they always fulfil the

Table 2. Number of Boolean functions (N_f) and classes (N_c) that are represented by the architectures R_i^2. Both graded and discrete activation functions are considered. The architectures of the last line of the table have neurons of different kinds. In that case, the symmetry associated with $\mathscr{T}_S(P_S^{1,2})$ is broken thus increasing the total number of classes from 34 to 55.

	Graded		Discrete	
i	N_c	N_f	N_c	N_f
0	11	81	11	8
3	4	9	7	16
1, 2	7	27	10	36

Table 3. Number of Boolean functions (N_f) and classes (N_c) that are represented by various architectures. Both graded and discrete activation functions are considered. The architectures of the last three rows of the table have neurons of different kinds in the last layer.

	Graded				Discrete			
	$\tilde{R}_i^{2,2}$		$R_i^{2,2}$		$\tilde{R}_i^{2,2}$		$R_i^{2,2}$	
i	N_c	N_f	N_c	N_f	N_c	N_f	N_c	N_f
0	11	81	11	81	29	218	34	256
1, 2	29	212	33	252	28	188	34	256
3	7	16	34	256	7	16	34	256
12	6	12	7	16	18	88	34	256
13, 14	7	33	34	256	18	88	34	256
15	6	12	34	256	7	16	31	224
4, 8	9	32	10	36	44	206	55	256
7, 11	10	16	55	256	10	16	55	256
5, 6, 9, 10	24	112	55	256	40	152	55	256

prescription (18) (see table 4). The learning curves are shown in figures 2(a), (b) and (c). The prescription (18) is a necessary but not a sufficient condition. In table 4 we see that not all the sixteen functions (belonging to seven symmetry classes) that fulfil (18) are represented by the architectures that we have considered.

A remarkable change occurs for networks composed only by even neurons when the restriction of having a cascade connection pattern is relaxed. The sixth row of table 3 shows that all the 256 possible Boolean functions are represented by the $R_{15}^{2,2}$ architecture. Figure 2(d) shows the corresponding learning curves.

The fully connected architectures $R_i^{2,2}$ combining neural processing of different parities show, in some cases, a striking richness. As can be seen in table 3, except for the cases $R_0^{2,2}$, $R_1^{2,2}$, $R_2^{2,2}$, $R_4^{2,2}$, $R_8^{2,2}$ and $R_{12}^{2,2}$, any Boolean function can be represented by the same architecture. It is worth remarking that this effect occurs when the even neural processing is used in the first layer, as in $R_3^{2,2}$, $R_7^{2,2}$ and $R_{11}^{2,2}$ (notice that in $R_3^{2,2}$ is only used in that layer).

Inhomogeneous architectures allow for neurons of different parity in the last layer. In these cases we find a change in the number of families as a consequence of the breaking of the permutational symmetry of the two outputs (see the appendix). When this occurs the number of symmetry classes grows from 34 to 55 and the maximum

Table 4. Families of Boolean functions that may be represented by cascade networks composed only by even neurons and the corresponding values of $H[\mathcal{L}_\alpha]$.

α	Functions F_k belonging to the family α	$\tilde{R}_3^2$	$\tilde{R}_{15}^{2,2}$	$\tilde{R}_{63}^{2,2,2}$
0	0	0	100	12 420
20	20, 40, 65, 130	0	0	0
60	60, 195	4	96	3 072
85	85, 170	0	350	43 470
105	105, 150	4	32	256
125	125, 190, 215, 235	10	304	10 496
255	255	25	4289	383 441

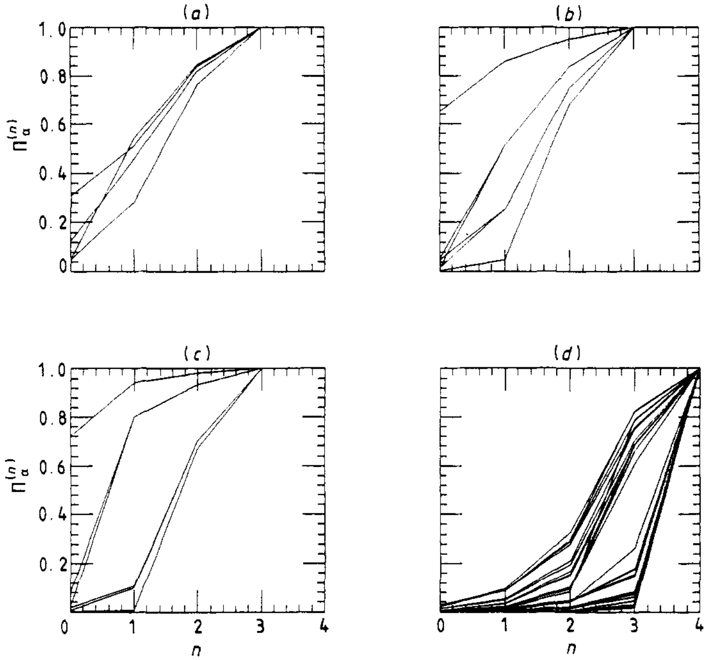

Figure 2. Probability of generalization as a function of the amount of training n, for networks composed by neurons having even activation functions. (a) R_3^2: at $n=1$, from top to bottom, the curves correspond respectively to the famlies $\alpha=60$, 255, 125 and 105. (b) $\tilde{R}_{15}^{2,2}$: at $n=0$, from top to bottom, the curves correspond respectively to the famlies $\alpha=255$, ($85 \approx 125$), ($0 \approx 60$) and 105. At $n=1$, from top to bottom, the curves correspond respectively to the families $\alpha=255$, ($85=0$), ($60 \approx 125$) and 105. (c) $\tilde{R}_{63}^{2,2,2}$: at $n=0$, from top to bottom, the curves correspond respectively to the families $\alpha=255$, 85, 0, 125, 60 and 105. At $n=1$, from top to bottom, the curves correspond respectively to the families $\alpha=255$, ($85=0$), 125, 60 and 105. (d) $R_{15}^{2,2}$: all the 34 symmetry classes are represented.

number of elements of each class becomes 8 instead of 16. Within each of the new, smaller families it still holds that all the Boolean functions have the same learning curves. This feature is illustrated in figure 3(a) in which are shown $\Pi_{85}^{(n)}$ and $\Pi_{170}^{(n)}$ for the homogeneous and inhomogeneous fully connected architectures $R_3^{2,2}$ and $R_7^{2,2}$. The same learning curves are represented in figure 3(b) for $\tilde{R}_3^{2,2}$ and $\tilde{R}_7^{2,2}$. In some cases one of the resulting subfamilies may not be realized as shown in figures 3(c) and (d).

In tables 2 and 3 we present also the results of the same numerical simulations for discrete response neurons. In this case the notation for the architectures is a natural extension of the one used for the graded response case. The noticeable result that follows from tables 2 and 3 is the systematic increase in the representability of networks composed by discrete neurons as compared with similar architecture with graded response ones. This can be understood in terms of the discussions made in subsection 3.1.

All the above described results suggest that there are general-purpose and special-purpose architectures. Figure 4 shows the different set of values $H[\mathscr{L}_\alpha]$ obtained for the cascade architectures $\tilde{R}_0^{2,2}$ and $\tilde{R}_{15}^{2,2}$ (figures 4(a) and (b)) and for the fully connected cases $R_0^{2,2}$ and $R_{15}^{2,2}$ (figures 4(c) and (d)). $\tilde{R}_0^{2,2}$, $\tilde{R}_{15}^{2,2}$ and $R_0^{2,2}$ should be considered as special-purpose networks while $R_{15}^{2,2}$ as a general-purpose one. It is possible to provide a quantiative measure of the degree of specialization through the inferential entropy

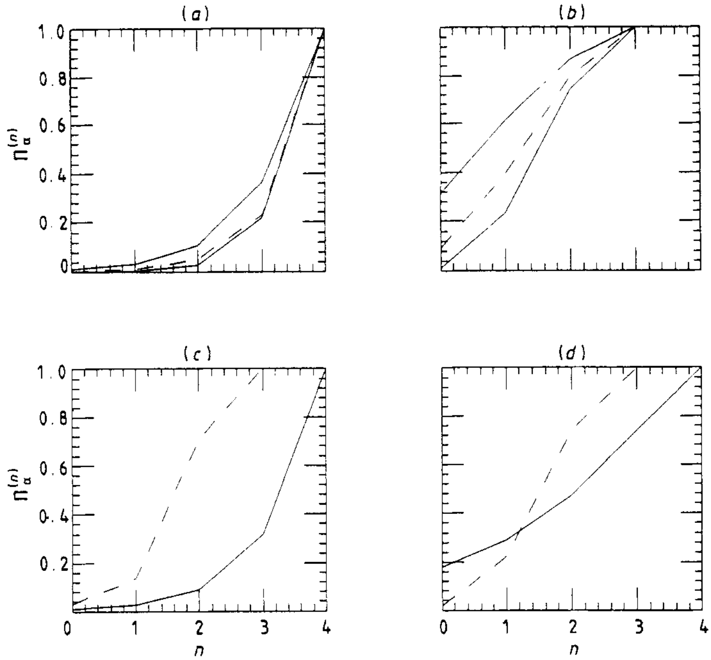

Figure 3. Effect of the symmetry breaking associated to the transformation $\mathcal{T}_S(P_S^{1,2})$, shown in the splitting of the learning curve corresponding to the functions F_{85} and F_{170}. The broken (full) lines correspond to the learning curves of a homogeneous (inhomogeneous) network. (*a*) $R_3^{2,2}$ (broken), $R_7^{2,2}$ (full); (*b*) $\tilde{R}_3^{2,2}$ (broken), $\tilde{R}_7^{2,2}$ (full); (*c*) $R_{12}^{2,2}$ (broken), $R_4^{2,2}$ (full) and (*d*) $\tilde{R}_{12}^{2,2}$ (broken), $\tilde{R}_4^{2,2}$ (full). In the last two cases F_{170} cannot be represented by the inhomogeneous networks.

that, for an untrained network, is defined by [4, 13]:

$$\mathfrak{S}_F^0 = -\sum_F \frac{H[\mathcal{L}(F)]}{H[\mathcal{J}]} \log_2 \frac{H[\mathcal{L}(F)]}{H[\mathcal{J}]}. \tag{31}$$

In (31) $H[\mathcal{J}]$ stands for the total number of synaptic matrices. In table 5 we give the values $\mathfrak{S}_F^0$ for the cases listed in table 4. Special-purpose architectures have lower entropy than the general-purpose ones. This can also be visualized comparing the values of $\mathfrak{S}_F^0$ with $\log_2 r$, r being the number of representable functions (see figure 5). This is the value that would correspond to the entropy of a system in which all the representable functions have equal values of $H[\mathcal{L}(F)]$.

It is also possible to define an entropy associated with the grouping of functions into symmetry classes by a given architecture [4]. This is a natural extension of (31):

$$\mathfrak{S}_{\mathcal{D}}^0 = -\sum_{\mathcal{D}} \frac{H[\mathcal{K}(\mathcal{D})]}{H[\mathcal{J}]} \log_2 \frac{H[\mathcal{K}(\mathcal{D})]}{H[\mathcal{J}]}. \tag{32}$$

In (32) $\mathcal{K}(\mathcal{D})$ is the set of matrices that represents any of the Boolean functions belonging to the class $\mathcal{D}$. $\mathfrak{S}_{\mathcal{D}}^0$ admits a similar interpretation than $\mathfrak{S}_F^0$. Large (small) values of $\mathfrak{S}_{\mathcal{D}}^0$ correspond to architectures that have an even (dissimilar) partition of the space $\mathcal{J}$ with respect to the whole set of symmetry classes. The values of $\mathfrak{S}_{\mathcal{D}}^0$ are listed in table 5. It can be checked that $\mathfrak{S}_F^0 > \mathfrak{S}_{\mathcal{D}}^0$ [4]. The 'general-purpose' against

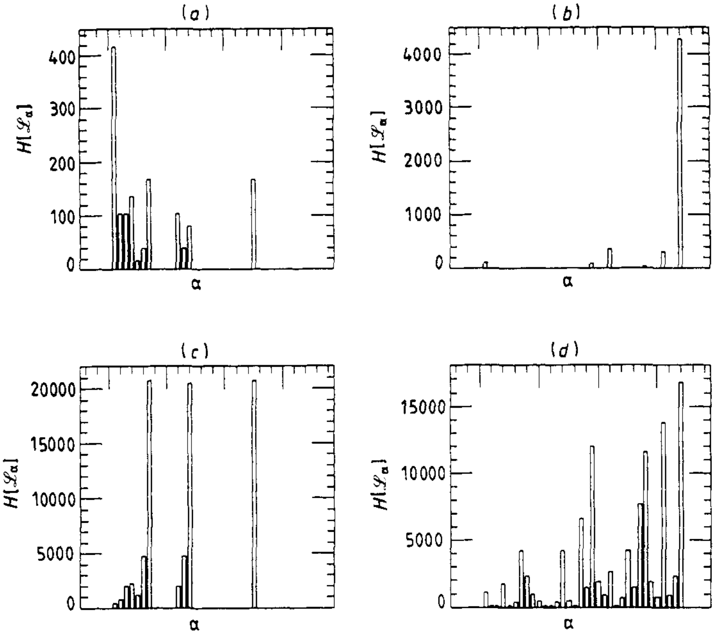

Figure 4. Historgrams of $H[\mathscr{L}_\alpha]$ as a function of the index α labelling the symmetry classes. The plots show how the space $\mathscr{I}$ is partitioned among the various Boolean functions (or classes) and consequently display different degrees of specialization. (a) $\tilde{R}_0^{2,2}$, (b) $\tilde{R}_{15}^{2,2}$, (c) $R_0^{2,2}$ and (d) $R_{15}^{2,2}$.

Table 5. Value sof the entropies $\mathfrak{S}_F^0$ and $\mathfrak{S}_{\mathscr{Q}}^0$ for the arachitectures $\tilde{R}_i^{2,2}$ and $R_i^{2,2}$. Both graded and discrete activation funitions are considered.

	Graded				Discrete			
	$\tilde{R}_i^{2,2}$		$R_i^{2,2}$		$\tilde{R}_i^{2,2}$		$R_i^{2,2}$	
i	$\mathfrak{S}_F^0$	$\mathfrak{S}_{\mathscr{Q}}^0$	$\mathfrak{S}_F^0$	$\mathfrak{S}_{\mathscr{Q}}^0$	$\mathfrak{S}_F^0$	$\mathfrak{S}_{\mathscr{Q}}^0$	$\mathfrak{S}_F^0$	$\mathfrak{S}_{\mathscr{Q}}^0$
0	6.005	3.332	5.632	2.692	6.669	4.013	7.102	3.960
1, 2	6.781	4.585	7.386	4.168	6.427	4.285	7.536	4.444
3	3.771	2.697	7.847	4.617	3.540	2.506	7.551	4.586
12	1.795	0.362	3.750	2.403	5.246	3.570	5.227	3.486
3, 14	2.586	1.660	6.935	4.318	5.395	3.504	5.569	3.488
15	2.019	1.502	6.961	4.145	3.244	2.385	5.156	3.251
4, 8	3.970	2.290	4.692	2.299	6.239	4.481	6.179	4.105
7, 11	2.971	2.484	7.411	5.055	3.449	2.976	6.368	4.461
5, 6, 9, 10	5.000	3.582	7.165	4.719	6.094	4.643	6.562	4.534

'special-purpose' concepts can thus be extended providing the location of a given architecture in the $\mathfrak{S}_F^0$ against $\mathfrak{S}_{\mathscr{Q}}^0$ plane (see figure 6). The points displayed in figure 6 can be grouped into, for instance, four sets according to the connection pattern (cascade or full) and to the activation function of the neurons (G: graded or D: discrete). This data can thus be summarized in the position of four centroids that provide an overall view of the effect of these two parameters. The values are (as usual

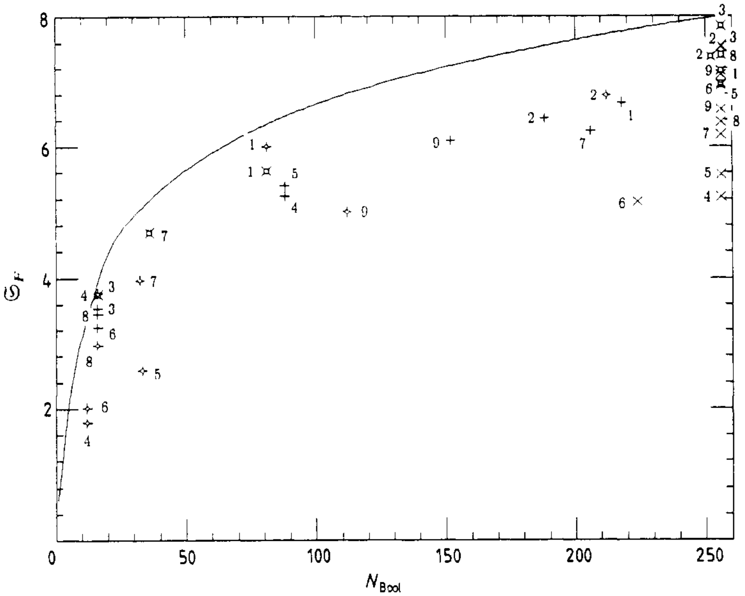

Figure 5. Values of the entropy $\mathfrak{S}^0_F$ against the number r of functions that the network is able to represent. The dots corresponds to the values listed in table 5. ✧: $\tilde{R}_i^{2,2}$, graded; ⌑: $R_i^{2,2}$, graded; +: $\tilde{R}_i^{2,2}$, discrete; ×: $R_i^{2,2}$, discrete. (1) $i=0$, (2) $i=1, 2$, (3) $i=3$, (4) $i=12$, (5) $i=13$, 14, (6) $i=15$, (7) $i=4, 8$, (8) $i=7, 11$, (9) $i=5, 6, 9, 10$. The continuous line represents the function $\log_2 r$.

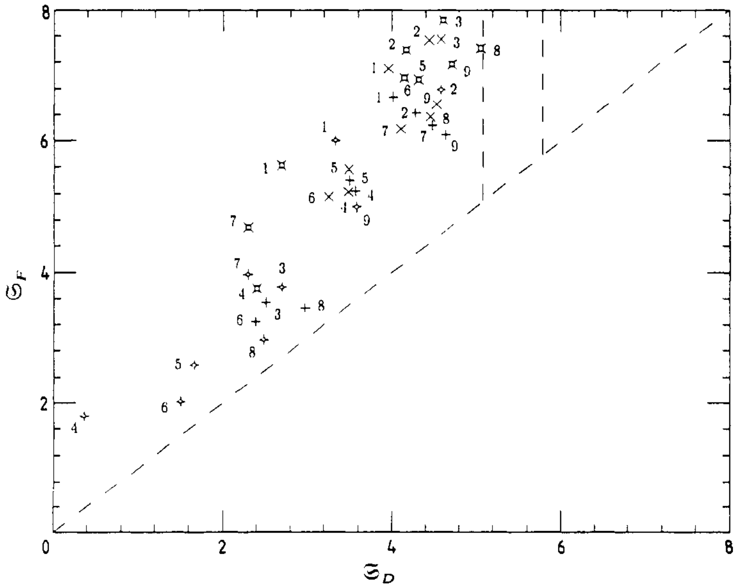

Figure 6. Pairs of values $(\mathfrak{S}^0_{\mathscr{D}}, \mathfrak{S}^0_F)$ for the architectures listed in table 5. ✧: $\tilde{R}_i^{2,2}$, grad; ⌑: $R_i^{2,2}$, graded; +: $\tilde{R}_i^{2,2}$, discrete; ×: $R_i^{2,2}$, discrete. (1) $i=0$, (2) $i=1$, 2, (3) $i=3$, (4) $i=12$, (5) $i=13$, 14, (6) $i=15$, (7) $i=4, 8$, (8) $i=7, 11$, (9) $i=5, 6, 9, 10$. The allowed region is above the line $\mathfrak{S}^0_F = \mathfrak{S}^0_{\mathscr{D}}$ and at the left of $\log_2 N_c$ (with $N_c = 34$ or 55). The borders are marked with broken lines.

a tilde denotes cascade architectures):

$$\begin{aligned} &(\mathfrak{S}^0_{\mathscr{D}}, \mathfrak{S}^0_F)_{\tilde{G}} = (2.766, 4.138) \qquad &(\mathfrak{S}^0_{\mathscr{D}}, \mathfrak{S}^0_F)_{\tilde{D}} = (3.846, 5.381) \\ &(\mathfrak{S}^0_{\mathscr{D}}, \mathfrak{S}^0_F)_G = (4.026, 6.606) \qquad &(\mathfrak{S}^0_{\mathscr{D}}, \mathfrak{S}^0_F)_D = (4.151, 6.412). \end{aligned} \tag{33}$$

It is seen that the change from graded to discrete response provides no significant increase in the representability for fully connected architectures while the opposite is true for cascade connection patterns.

5. Conclusions

In the present paper we have considered networks of neurons with different activation functions. The type of inhomogeneity that we have considered is not entirely based in the biological evidence but proves to be useful for artificial intelligence devices since it allows for a remarkable increase in the representability properties of the system. The fact that such drastic change occurs when the even processing is located in the input layer opens a suggestive field in the search of a biological counterpart in primary sensory neurons (or groups of them) and in receptors cells.

The models that we have considered can also be compared with networks built by Boolean gates [2, 10]. The comparison can be made taking into account: (*a*) the richness of the representability, (*b*) the degree of complexity of the processing elements and (*c*) the possible values of the connections between two processing elements.

The weight of the connections (analogous to the synaptic efficacies) of a network composed by Boolean gates that only admit two inputs, can only be zero (the gates are not connected) or one (the gates are connected). On the other hand there are sixteen possible different choices for each gate of the network.

From the present simulations we can conclude that networks with neurons having graded activation functions of different parity and networks constructed with Boolean gates have similar performances. A naive comparison of these two kinds of networks suggests that most of the complications that one type has in the architecture and in the synaptic connection pattern, the other has it in the internal processing function of the elementary building blocks. A neural network has a more complicated synaptic pattern because each neuron may be connected with more than two input signals. In addition each synaptic efficacy can take in principle any real value. On the other hand the possible internal activation functions of the analogue devices can be limited to be of only two types, by choosing the same odd or even functions for all the neurons of the system.

The results of our calculations therefore suggest that there exists some kind of balance between the simplicity of the neural processing and the complexity of the synaptic matrix. Less internal degrees of freedom in the processing element appears to be complemented by a more complex synaptic matrix.

In what refers to the adaptability properties of inhomogeneous cascade architectures one should notice that the whole set $\tilde{R}_i^{2,2}$, when jointly considered, leaves no Boolean functions without being represented. This implies that an adaptative learning algorithm may be worked out in which the search is performed within the limited set of cascade connectivities but allowing to change not only the synaptic efficacies but also the parity of the neural processing in each site of the network. This search can be performed with a fixed choice of the filtering devices because the representability in inhomogeneous networks turns out to be independent of the output filtering. A learning protocol based on simulated annealing [6] using this fact will be reported elsewhere [5].

The simultaneous consideration of processing elements with two kinds of graded activation functions (odd and even) appears as a valid and perhaps simpler alternative to networks with neurons having all different thresholds.

Appendix. Symmetries and symmetry breaking

In a previous paper [3] we have studied the symmetry properties of homogeneous feed-forward networks. The processing of the net can be regarded as the action of an operator F_J (parametrized by the synaptic matrix J) acting on the input array $|E\rangle$:

$$|S\rangle = F_J|E\rangle. \tag{34}$$

Some transformations $\mathcal{T}_e$ operating over $|E\rangle$ can be compensated by acting simultaneously on the synaptic matrix J with a transformation $\mathcal{T}_J^{-1}$:

$$|S\rangle = F_{\mathcal{T}_J^{-1}J}|\mathcal{T}_e E\rangle. \tag{35}$$

Since the output array $|S\rangle$ does not change, we can regard the simultaneous action of $\mathcal{T}_e$ and $\mathcal{T}_J^{-1}$ as a symmetry of the network.

Two kinds of transformations $\mathcal{T}_e$ have been considered in [3]: (i) the permutation of any pair of input signals E_k and E_l ($k \neq l$) [$\mathcal{T}_e(P_e^{l,k})$]; (ii) the logic negation of any of the input signals E_j [$\mathcal{T}_e(N_e^j)$]. The transformation $\mathcal{T}_e(P_e^{l,k})$ can be balanced interchanging the lth and kth columns associated with the 0th layer in the synaptic matrix J. Likewise, $|S\rangle$ does not change if the operation $\mathcal{T}_e(N_e^j)$ is accompanied by changing the signs of the jth column associated with the 0th layer of J. This last operation on J can actually be performed if both a and $-a$ are acceptable values for $J_{ij}^{\lambda 0}$. Allowing, for instance, only positive values for the synaptic efficacies would break the symmetries associated with the transformations $\mathcal{T}_e(N_e^j)$.

Other transformations $\mathcal{T}_S$ that operate over $|S\rangle$ can also be compensated acting simultaneously on J in such a way that:

$$|S\rangle = \mathcal{T}_S F_{\mathcal{T}_J^{-1}J}|E\rangle. \tag{36}$$

For instance, the permutation of the pair (l, k) of output signals [$\mathcal{T}_S(P_S^{l,k})$] can be cancelled by interchanging the lth and kth rows associated with the Lth layer in the matrix J.

The transformations $\mathcal{T}_e(P_e^{l,k})$ and $\mathcal{T}_e(N_e^j)$ act on the input array $|E\rangle$ but $\mathcal{T}_S(P_S^{l,k})$ operates on $|S\rangle$. To deal only with operators acting on the same space, instead of considering $\mathcal{T}_e(P_e^{l,k})$ and $\mathcal{T}_e(N_e^j)$ we use $\mathcal{T}_S(P_e^{l,k})$ and $\mathcal{T}_S(N_e^j)$ defined by:

$$\mathcal{T}_S(P_e^{l,k})|S\rangle = F_J|\mathcal{T}_e(P_e^{l,k})E\rangle \tag{37}$$

$$\mathcal{T}_S(N_e^j)|S\rangle = F_J|\mathcal{T}_e(N_e^j)E\rangle. \tag{38}$$

The transformations $\mathcal{T}_S(P_S^{l,k})$, $\mathcal{T}_S(P_e^{l,k})$ and $\mathcal{T}_S(N_e^j)$ are generators of a finite group G_S of transformations that act on the space $\mathcal{F}$ of all output arrays $|S\rangle$. The application of all the transformations of the group G_S to an output array $|S\rangle$ generates the symmetry class $\mathcal{D}(|S\rangle)$ (or $\mathcal{D}(F)$) associated with it. The whole space $\mathcal{F}$ can therefore be partitioned into disjoint symmetry classes or families $\mathcal{D}_\alpha$. All the functions belonging to the same class are represented by the same number of synaptic matrices [3]. When we consider networks with mixed processing some of the above mentioned symmetries may break. For example, in the case in which the last layer is composed with two different types of neurons the network is no longer invariant under the permutations of two output signals and the families $\mathscr{D}_\alpha$ that are not invariant under the transformation $\mathscr{T}_S(P_S^{1,2})$, split into subfamilies. This symmetry also breaks in homogeneous networks when the thresholds θ_i^λ of the neurons of the last layer are allowed to take different values.

Within the above considerations it can also be proved [3] that any two functions belonging to the same set $\mathscr{D}_\alpha$ have the same values of $\Pi^{(n)}(F)$, $\forall n$.

References

[1] Ackley D H, Hinton G E and Sejnowski T J 1985 A learning algorithm for Boltzmann machines *Cognitive Sci.* **9** 147-69

[2] Carnevali P and Patarnello S 1987 Exhaustive thermodynamical analysis of Boolean learning networks *Europhys. Lett.* **4** 1199-204

[3] Ferrán E A and Perazzo R P J 1989 Symmetry properties of feed-forward neural networks *Preprint* CNEA Buenos Aires, Argentina

[4] Ferrán E A and Perazzo R P J 1990 Inferential entropies of feed-forward neural networks *Phys. Rev.* A in press

[5] Ferrán E A and Perazzo R P J 1990 *Preprint* CNEA Buenos Aires, Argentina

[6] Kirkpatrick S, Gelatt Jr C D and Vecchi M P 1983. Optimization by simulated annealing *Science*, **220** 671-80

[7] Le Cun Y 1985 A learning scheme for asymmetric threshold network *Proc. of Cognitiva 85, Paris* vol 220 pp 599-604

[8] McCulloch W S and Pitts W H 1943 A logical calculus of the ideas immanent in neurons activity *Bull. Math. Biophys.* **5** 115-13

[9] Parker D B 1985 Learning logic *Technical Report R-47* (Cambridge, MA: Sloan School of Management, MIT)

[10] Patarnello S and Carnevali P 1987. Learning networks of neurons with Boolean logic. *Europhys. Lett.* **4** 503-8

[11] Rosenblatt F 1962 *Principles of Neurodynamics* (New York: Spartan)

[12] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors. *Nature*, **323**, 533-6

[13] Solla S 1989 Learning and generalization in layered neural networks: the contiguity problem *Neural networks from models to applications* ed L Personnaz and G Dreyfus (Paris: IDSET) p 168