Synthesis of Boolean Nets and Time Behavior of a General Mathematical Neuron

E. R. Caianiello and W. E. L. Grimson
Laboratorio di Cibernetica del C.N.R., Arco Felice, Naples, Italy

Received: August 6, 1974

Abstract

All neuronic equations proposed by Caianiello (1961) are completely linearized in tensor space, and all problems related to them are reduced to matricial relations, i.e. solved in principle in compact form. As a consequence, one obtains the general synthesis of an arbitrary boolean net, from which the complete treatment of the general equations for a single mathematical neuron is easily derived. This formalism appears to be the natural tool for obtaining concise solutions to the many problems previously posed concerning neuronic equations.

I.

Much of the attention of work in neural net theory in the past has been restricted to nets whose elements receive inputs only from the state of the net at the last time interval. In more precise terms, concern has been centered on nets which, at some time \( t + 1 \), achieve a new state solely as a function of the states of the elements of the net at time \( t \). Many relevant results have been obtained for such nets; as in Caianiello (1966), Caianiello et al. (1967), de Luca and Ricciardi (1969), Aiello et al. (1970), Accardi (1971), Cull (1971), de Luca (1971).

However, the general form of the neuronic equations for a net of \( N \) elements as proposed by Caianiello (1961) is usually taken as

\[
U_h(t + 1) = \sum_{k=1}^{N} \sum_{r=0}^{L(h)} \alpha_{hk, r} U_k(t - r\tau) - S_h
\]

where \( U_h(t) \) = state of element \( h \) at time \( t \) [in a \((1, 0)\) configuration],

\( \alpha_{hk, r} \) = Heaviside step function,

\( \alpha_{hk, r} \) = coupling coefficient from element \( k \) to element \( h \) at time \( t - r\tau \),

\( S_h \) = threshold,

with time being quantized in units of \( \tau \).

The above restriction to nets depending only on the previous state is obtained by requiring that \( L(h) \) be zero for all \( h \); \( 1 \leq h \leq N \). But clearly, nets of the general nature described by (1) should also be investigated.

We may show that nets which do extend backwards in time, i.e. receive inputs from states of the net several time units previously, can in fact be considered as nets depending only on the previous state, with particular restrictions. As a consequence, many of the techniques employed in the study of previous-state nets may be re-utilized in the study of general nets. To illustrate such a result, we consider the special case of the behavior of a single neuron in time. Indeed, the tensorial expansion formalism is particularly suited to such an investigation; we extend it to the case of boolean nets in the next section.

II.

Consider a net of \( N \) elements, the input of each of which is some boolean function \( \text{sgn}[f_i(x)] = \sigma[f_i(x)] \) of the outputs of all \( N \) elements; if time is quantified in discrete successive instants, the state of the net (i.e. the specific values \( \pm 1 \) of each of the \( N \) elements) can be represented as a vector \( \xi = (\xi_1, \ldots, \xi_N) \); it will change in time (we denote with \( m \) the discrete time label) according to the equations

\[
\xi_{m+1} = \sigma[f_i(\xi_1, \ldots, \xi_N)], \quad f = \{f_1, \ldots, f_N\}
\]

or, in tensor space:

\[
\xi_{h,m+1} = \sum_i f_{h,i} \eta_{h,m}.
\]

There are \( 2^N \) possible vector states, we form an \( N \times 2^N \) matrix \( \varphi_N \) with them; e.g., \( N = 3 \):

\[
\varphi_3 = 2^{-1} \begin{pmatrix}
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & -1 & -1
\end{pmatrix}
\]
The matrix \( \varphi_N \) can be argumented to the \( 2^N \times 2^N \) matrix that obtains when in the \( 2^N \)-component direct product
\[
\begin{pmatrix}
1 \\
\sqrt{2} \\
\xi_1 \\
\sqrt{2} \\
\vdots
\end{pmatrix}
\times \cdots
\times
\begin{pmatrix}
1 \\
\sqrt{2} \\
\xi_N \\
\sqrt{2} \\
\end{pmatrix}
\]
\( \xi_1, \ldots, \xi_N \) are all given \( \pm 1 \) values. Thus
\[
\Phi_N = \begin{pmatrix}
\sqrt{2} & 1 \\
1 & 1 \\
\sqrt{2} & \sqrt{2} \\
1 & \sqrt{2}
\end{pmatrix}
\times \cdots
\times
\begin{pmatrix}
1 & 1 \\
\sqrt{2} & 1 \\
1 & \sqrt{2}
\end{pmatrix}
\quad (N \text{ times}) \quad \text{(standard form)}.
\]

Clearly:
\[
\Phi_N = \varphi_N^2; \quad \varphi_N^2 = 1; \quad \Phi_N = \varphi_N^{-1}; \quad \det \Phi_N = (-1)^N.
\]

The matrix \( \varphi_N \) corresponds to the rows of \( \Phi_N \) which are generated by \( \xi_1, \xi_2, \ldots, \xi_N \) alone. Thus, we can write the equation of the net exhibiting all the states
\[
\varphi_{N,m+1} = f_N \varphi_{N,m} = \varphi_{N,m} P_N
\]
where \( f_N \) is an \( N \times 2^N \) matrix with components \( f_{h,s} \) and \( P_N \) is clearly a permutation matrix (which is projective if transient states exist) that from the right, permutes the columns of \( \varphi_{N,m} \) so as to bring it into \( \varphi_{N,m+1} \).

We can profitably enlarge the last equation by defining a \( 2^N \times 2^N \) matrix \( F_N \) as follows
\[
\Phi_{N,m+1} = F_N \Phi_{N,m} \quad \Phi_{N,m} = \Phi_{N,m} P_N
\]
this is done by replacing in the previous equation \( \varphi_N \) with \( \Phi_N \) and defining \( F_N \) from the last one written. We can start with \( m = 0 \):
\[
\Phi_{N,1} = F_N \Phi_{N,0} = \Phi_{N,0} P_N
\]

hence
\[
\Phi_{N,m} = F_N^m \Phi_{N,0} = F_{N,0}^m P_N
\]
By taking \( \Phi_{N,0} \) as the standard form (3)
\[
\Phi_{N,0} = \Phi_N
\]
we find immediately
\[
F_N = \Phi_N P_N \Phi_N
\]

hence the \( 2^N \times 2^N \) matrix \( F_N \) is immediately constructed when \( P_N \) is assigned, i.e. when the wanted sequence of temporal states of the net is prescribed. Since \( f_N \) is trivially extracted from \( F_N \):
\[
f_N = \varphi_N P_N \varphi_N
\]
we see that we have fully solved the general problem of the synthesis of a boolean net that performs an arbitrarily prescribed sequence of states.

To obtain similar results for the \((1,0)\) representation of boolean functions, we may just replace the matrix \( \Phi_1 \), with the matrix
\[
\Theta_1 = \begin{pmatrix}
1 & 1 \\
1 & 0
\end{pmatrix}
\]
so that, correspondingly,
\[
\Phi_N = \Theta_1 \times \cdots \times \Theta_1
\]

III.

If we are considering a net of the form described by (1) where \( L(h) \) is non-zero, we wish to construct a new net which retains the essential features and behaviors of the first, but has \( L(h) \) equal to zero for all \( h \).

Let us relabel the elements in the following manner where \( L = \max L(h) \)
\[
\begin{align*}
w_1(t) &= u_1(t) \\
w_2(t) &= u_1(t - \tau) \\
w_3(t) &= u_1(t - 2\tau) \\
w_{L+1}(t) &= u_1(t - L\tau) \\
w_{L+2}(t) &= u_2(t)
\end{align*}
\]
and in general
\[
w_q(t) = u_j(t - r\tau)
\]
where
\[
q = (j - 1)(L + 1) + r + 1.
\]

Then, with a proper relabelling of the coefficients \( a_{hk} \) we have:
\[
W_i(t + \tau) = 1 \left[ \sum_{m=1}^{R} a_{im} W_m(t) - S_i \right] \quad \text{for}
\]
\[
i = 1, \quad L + 2, 2L + 3, \ldots, \quad (N - 1) \quad L + N
\]
and
\[
w_i(t + \tau) = 1 \left[ w_{i-1}(t) \right] = w_{i-1}(t) \quad \text{for all other } i.
\]

The dimension of the net is given by \( R \):
\[
R = N(L + 1)
\]

A similar construction may be effected in the \((1, -1)\) configuration used when the Heaviside function is replaced by a signum function whose argument is non-zero.