

Identification of the Connections In Biologically Inspired Neural Networks

H. Demuth, K. Leung and M. Beale
Department of Electrical Engineering
University of Idaho
Moscow, Idaho 83843
J. Hicklin
The Mathworks, Inc.
South Natick, Massachusetts 01760

Abstract - We developed an identification method to find the strength of the connections between neurons from their behavior in small biologically-inspired artificial neural networks. That is, given the network external inputs and the temporal firing pattern of the neurons, we can calculate a solution for the strengths of the connections between neurons and the initial neuron activations if a solution exists. The method determines directly if there is a solution to a particular neural network problem. No training of the network is required.

It should be noted that this is a first pass at the solution of a difficult problem. The neuron and network models chosen are related to biology but do not contain all of its complexities, some of which we hope to add to the model in future work.

A variety of new results have been obtained. First, the method has been tailored to produce connection weight matrix solutions for networks with important features of biological neural (bioneural) networks. Second, a computationally efficient method of finding a robust central solution has been developed. This later method also enables us to find the most consistent solution in the presence of noisy data.

Prospects of applying our method to identify bioneural network connections are exciting because such connections are almost impossible to measure in the laboratory. Knowledge of such connections would facilitate an understanding of bioneural networks and would allow the construction of the electronic counterparts of bioneural networks on very large scale integrated (VLSI) circuits.

Keywords: neural networks, biological models, neurons, identification

1 Introduction

Brains of living organisms are difficult to understand in part because they can contain billions of extremely complex neurons [6], [1]. However, there is an additional significant difficulty in that the effects of connections between the neurons cannot be measured

effectively or directly in the laboratory. Willows, for instance, reports some 20,000 measurements to determine which regions of a *Tritonia* would evoke responses of the mollusk's tens of neural cells, and goes on to point out that these experiments had to be conducted in reverse to find where the cells sent their messages [12]. His work, absolutely first class in every respect, illustrates the problem of determining the existence of neural connections, let alone evaluating their strengths.

Previous attempts to understand bioneural networks have involved direct measurement of physical properties or partial destruction of networks to observe the effects of such changes. Both approaches are long and tedious procedures. Our research suggests an alternative method of solving the problem of determining the strengths of connections between neurons. An understanding of the structure of bioneural networks would enhance the feasibility of neural repair, neural prosthesis design, and the construction of electronic equivalent circuits such as artificial visual or recognitions systems.

At present there is great interest in artificial neural networks that can recognize objects or patterns in complex data. The current approach of many investigators who wish to build a practical neural network to perform a specific task is to approximate a neuron, constrain the neural network connections to be feedforward, and train the networks by a method such as backpropagation [9]. Such networks bear little resemblance to bioneural systems, and it is not clear in each case whether a network of a specific size can be trained to do the desired job. However, networks of this kind have been trained to do interesting tasks, including some tasks that are not commonly done well by conventional computers or by any neurobiological system [8]. In contrast to the approach of many investigators, we have chosen neuron and network models that capture some of the most important aspects of the behavior of bioneural networks. Note that if one could understand bioneural networks, one could build them directly to solve some difficult problems.

The purpose of this paper is to present the neuron and network models that we have assumed, to show how our identification method proceeds and to tell of new results, including a way to find a robust central solution. The contributions of the paper lie in the novelty, power and simplicity of the method, which allows one to solve the "identification problem" for biologically inspired neural networks which are nonlinear systems with memory; and in various new results presented, including the development of an efficient method for finding a robust solution.

2 Neuron and Network Models

A recent DARPA Neural Network Study [3] defines a neural network as "a computational structure modeled on biological processes." We have chosen our neuron model and network model in that spirit. A schematic of the neuron model that we will adopt is shown in Figure 1.

In biological terms, the soma on the left of Figure 1 is the body of the neuron, and its axon projects to the right. External inputs impinge on the soma. Input signals from presynaptic neurons impinge on the soma and dendrites. Output signals are transmitted

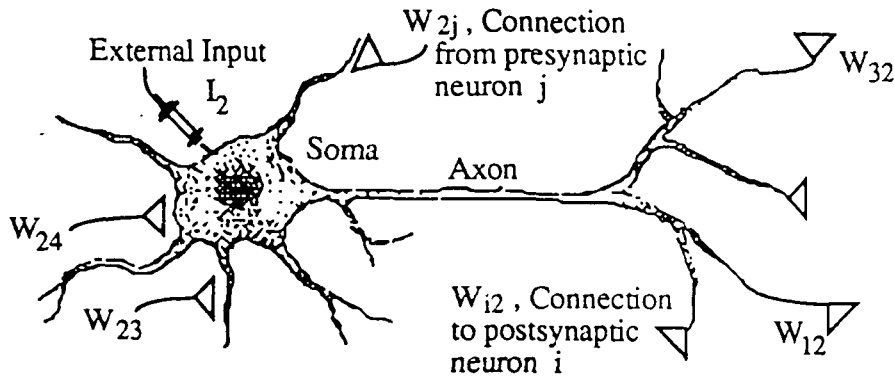


Figure 1: A Single Neuron, Neuron 2 (N2)

through the axon to the postsynaptic neurons on the right.

External inputs to the neuron and signals from other neuron's firings are integrated to form an "activation level." When a neuron's activation exceeds a "threshold" T , the neuron "fires", sending output signals to postsynaptic neurons and setting its own activation to the resting potential, taken here to be zero. The strengths of connections between neurons are given by the element w_{ij} . Thus, when neuron j (N_j) fires, it sends a signal of strength w_{ij} to N_i , and sets its own activation level to the resting potential. The set of all w_{ij} will be referred to as the weight matrix, \mathbf{W} , and the set of initial activations $a_i(1)$ will be referred to as the initial activation vector $\mathbf{A}(1)$. Our objective is to develop a method for finding a weight matrix \mathbf{W} and an initial activation vector \mathbf{A} for a particular neural network whose external inputs and temporal firing pattern are specified.

An equation for the activation level a_i of N_i in a neural network and definitions of the equation's terms follow.

$$a_i(t+1) = I_i(t) + (1 - F[a_i(t)])a_i(t) + \sum_{j=1}^N F[a_j(t)]w_{ij} \quad (1)$$

$$F[a_j(t)] = 1 \text{ if } a_j(t) \geq T \quad F[a_j(t)] = 0 \text{ if } a_j(t) < T \quad (2)$$

$a_i(t)$ is the activation of neuron i at time t ,
 w_{ij} is the strength of the connection from neuron j to neuron i ,
 $I_i(t)$ is the external input to neuron i at time t , and
 T is the threshold at which the neurons will fire.

An example of a two-neuron network with known connections and initial activations is shown in Figure 2.

As this figure shows in the diagram on the top and in the weight matrix \mathbf{W} , the strength of the connection from N_1 to itself is 11, the strength of the connection from N_2 to N_1 is

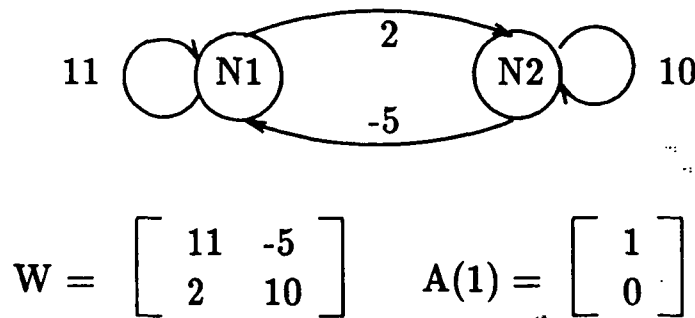


Figure 2: A Simple Two-Neuron Neural Network

-5, etc.. The expression for A shows that the initial activation of N1 is 1, which will cause N1 to fire at $t = 1$.

The networks need not be fully connected, and can be constrained in various ways if desired. For instance, a network can be made feed-forward by setting the upper right coefficients of W to be zero.

In our model, the weight matrix W is not a function of time. However, the method described here can be applied at various times to track changes in weight matrix connections with time.

3 Identification Method

The identification and design method pioneered by Hicklin [5] and used here will be illustrated through a simple example. Let us consider the following firing pattern of length $L = 6$ or a hypothetical two-neuron system with zero inputs:

Time t	1	2	3	4	5	6
Firing of N1	1	1	1	1	1	1
Firing of N2	1	0	1	0	1	0

Table 1: Two-neuron Firing Pattern

What can we deduce about the network producing this behavior? The fact that both neurons fired at $t = 1$ tells us that the initial activation of each neuron was above threshold and thus:

$$a_1(1) \geq T \quad (3)$$

$$a_2(1) \geq T \quad (4)$$

The fact that N1 fired on the second step tells us that its activation at $t = 2$ was also above threshold. What was its activation? From Equation 1 :

$$a_1(2) = I_1(1) + (1 - F[a_1(1)])a_1(1) + F[a_1(1)]w_{11} + F[a_2(1)]w_{12} \quad (5)$$

There were no inputs, so the first term is zero. Both neurons fired on the previous time step ($t=1$) so we know that $F[a_1(1)] = 1$ and $F[a_2(1)] = 1$. When we insert these values in the equation shown above, remembering that N1 fired on the second time step, we have:

$$a_1(2) = w_{11} + w_{12} \geq T \quad (6)$$

We have reached an important point in our development, so let us pause to consider what has been done. As shown in Equation 6 we have established the first weight matrix inequality. It constrains the sum of the two weight matrix elements, w_{11} and w_{12} . Any weight matrix that is to reproduce the specified firing pattern must satisfy this inequality. We can continue extracting inequalities from the firing pattern in this manner. When we have written $N * L = 12$ constraint inequalities we will seek a W and an A that will satisfy all of them. If a solution exists, the network using the W and A solution will produce the desired firing pattern. Now let us proceed with finding additional constraint equations. N2 did not fire on the second time step, so the mathematical process used to obtain the inequality of eq. 6 can be applied here to yield:

$$a_2(2) = w_{21} + w_{22} < T \quad (7)$$

As seen in eqs. 6 and 7, at $t = 2$ each neuron's behavior yielded an inequality relating the threshold and the strengths of the connections to that neuron from firing neurons. Note that the two weight matrix elements w_{11} and w_{12} of eq. 6 are in the first row of the weight matrix, while the two elements w_{21} and w_{22} of eq. 7 are in the second row of the weight matrix. Thus, when we seek a solution, we only need to solve two independent sets of inequalities (eqs. 6 and 7) each involving two elements of the weight matrix, rather than solving one set of inequalities involving all four weight matrix elements.

Proceeding in this way, six inequalities for N1 and six inequalities for N2 can be developed. The most common and well-known method of solving such a system of inequalities is the simplex method, which is described in most introductory linear programming texts [2]. When we apply the simplex method to the linear inequalities for this network we find that the following solution does exist. We have simulated a neural network with the

$$W = \begin{bmatrix} 2.00 & -1.00 \\ 1.00 & -1.00 \end{bmatrix} \quad A(1) = \begin{bmatrix} 1.00 \\ 1.00 \end{bmatrix}$$

initial conditions and weight matrix shown. The network does produce the firing pattern specified in Table 1.

4 Graphical Viewpoint

A weight matrix solution to each of two sets of equations such as those of eqs. 6 and 7 can be obtained graphically in two dimensions for the two-neuron case. This graphical technique can give some insight into methods for solving systems with more variables. In

two dimensions an inequality is a line that partitions a plane into a solution region and a non-solution region. A set of linear inequalities will create a more complicated situation. Two or more linear inequalities may result in either an infinite solution region, a finite solution region, or no solution region at all. Figure 3 shows the partitioning of the solution space for two inequalities associated with the first row elements of a weight matrix of a neural network that has a firing threshold of one. As shown above, the solution area is

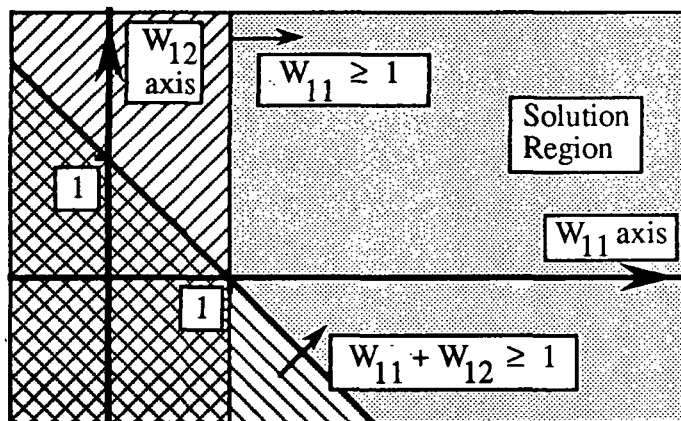


Figure 3: Example of Constraints on w_{11} and w_{12}

constrained to be to the right of the vertical line $w_{11} = 1$ and to the upper right of the sloped line $w_{11} + w_{12} = 1$. If a solution space remains after all of the inequalities have been extracted from a firing pattern, then any point in the solution space can be used as the weights in the matrix and the resulting network will exhibit the desired behavior. If there is no solution space, then no two-neuron network can perform as desired. As mentioned previously, if a network is to be realized then solutions must exist for all rows of W .

5 Identification and Design Applications

The methods that we have developed can be used to identify the connections of neural networks that have an arbitrary desired firing pattern and input pattern. Let us take an example that requires the design of a four neuron system that produces a specific nine bit temporal firing pattern. The network is to have a threshold of 1. The desired firing pattern and inputs are shown in Table 2.

Time Step	1	2	3	4	5	6	7	8	9
Input to neuron1	2	7	1	2	-2	8	1	-2	3
Input to neuron2	1	-4	1	4	2	1	-3	5	6
Input to neuron3	6	0	2	-2	1	6	9	2	3
Input to neuron4	2	1	-2	7	9	2	0	-3	1
Firing of neuron1	0	1	0	0	0	0	1	0	0
Firing of neuron2	0	1	0	1	0	1	0	0	1
Firing of neuron3	1	1	0	0	0	0	0	1	1
Firing of neuron4	1	1	1	0	1	1	1	0	0

Table 2: The Specified Behavior

Our computer program generated the inequalities and presented them to the simplex method. The solution shown below was found in about 5 seconds on an Apple Macintosh II computer.

$$W = \begin{bmatrix} -3.00 & -3.00 & -1.61 & -0.34 \\ 0.00 & -3.03 & 1.05 & 3.00 \\ -3.00 & -3.00 & -0.98 & 0.20 \\ 1.48 & -0.48 & -0.48 & -0.50 \end{bmatrix} \quad A(1) = \begin{bmatrix} 0.97 \\ -3.00 \\ 1.02 \\ 1.02 \end{bmatrix}$$

Table 3: The Designed Network

Next consider the probability of realization of firing patterns when there are no external inputs and when W and A are free variables. Our method was applied to determine the percentage of patterns realizable for up to ten neurons and ten time steps. The results in Fig. 4 show that the probability of realization increases as the number of neurons increase and as the number of time steps decrease.

6 Application To Biology

Obtaining data in the laboratory about bioneural network connections is extremely difficult. Many biologists, working for decades, have been able to establish some understanding of the neural connections in only a dozen or so crustacean "minibrains", each of which has ten to twenty neurons [10], [11]. Consequently, the prospect of applying our identification method to bioneural systems is exciting and we have begun to make our model more biologically realistic. For instance our method can be tailored so that it can calculate the weight matrix connections of networks having additional features of bioneural networks, as described below.

An individual bioneuron does not have direct connections from its axon back to its soma or dendrites. Thus, the diagonal terms w_{ii} of the weight matrix, W , must be constrained

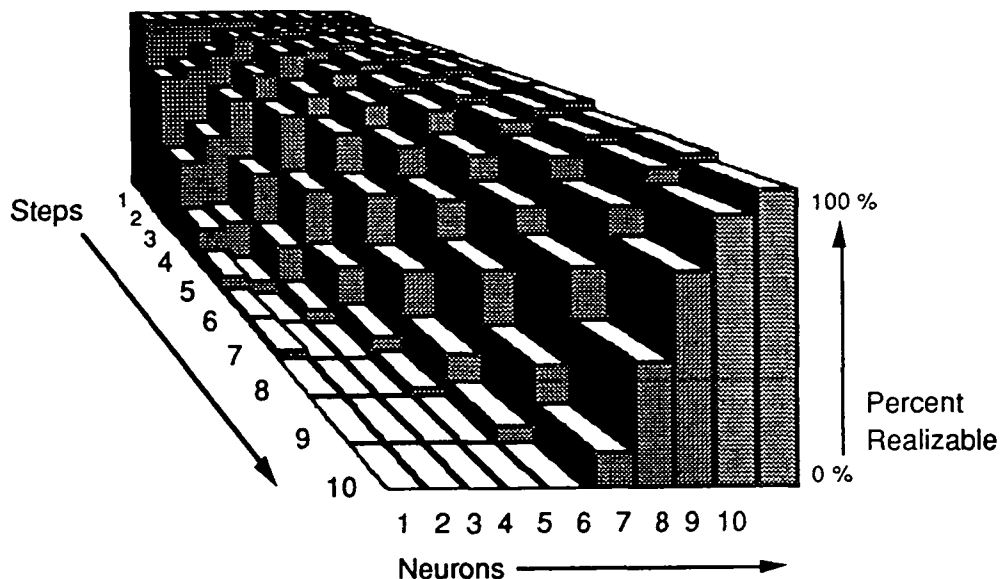


Figure 4: Percent of Patterns Realizable

to the value zero. Also, bioneurons commonly are either excitatory or inhibitory to any particular group of postsynaptic neurons so the columns of W will be either positive or negative.

The N diagonal terms of the weight matrix can be constrained to have the value zero by introducing N such constraints, one for each diagonal term. There are $2N$ possible sets of such column constraints and we can find a solution by using a tree search. The column constraints do not couple the N sets of inequalities associated with N neurons, so we are still dealing with L inequalities for each row. We have applied our method to design a variety of networks having zero diagonal and positive or negative column constraints. As additional constraints are introduced, the probability of realization of a network for a randomly generated temporal firing pattern decreases. We have run tens of thousands of cases on a Cray X-MP to determine the effects of the diagonal zero constraint and the column constraints and combinations of both. Each kind of constraint decreases the probability of realization. The decrease is especially significant in those cases where there are few neurons and/or many time steps. Details of these results will be reported in an upcoming M.S. thesis [7].

Thus, our method can be tailored to include additional biological features. This gives hope that bioneural connection strengths could be obtained from a manageable combination of experiments and calculations based on our method.

Now let us see how some of our viewpoints can be applied in a specific biological example. Consider a lobster's stomatogastric ganglion. This minibrain of some twenty neurons must produce a satisfactory firing pattern if the lobster is to survive. Suppose we were to examine the inputs to this minibrain, measure its firing pattern, apply our (extended and refined) method to identify a weight matrix solution, and map the associated solution space. The lobster could be using any solution in this space satisfactorily. The question is, "Where does he (or she?) live?". Presumably the lobster is not "living on the

edge", because the slightest indigestion might put him/her out of the solution space and thereby out of commission. The species has survived, so it must have solved the problems of establishing a solution space of sufficient size and of finding a robust solution, presumably in the central part of its solution space. We could apply our own inputs to the lobster's minibrain in a laboratory test to produce additional inequalities that would allow us to calculate the lobster's exact weight matrix solution. Knowledge of the lobster's solution space and his/her exact weight matrix solution would help us to predict and understand the effects on the lobster of neural damage, drugs, etc.

7 Robust Solution

Phase one of the simplex method as used above to identify networks produces a solution on the vertex of the solution space. Such a solution is not robust, for a slight change in connection strengths may result in the "network solution" no longer being in the solution space. In other words, the phase one simplex solution is not robust because it is at the edge of the solution space. We would prefer a solution which is in some sense "central" in the solution space.

A solution that is maximally distant from its nearest constraint can be obtained efficiently by using the second phase of the simplex method [4]. To do this, we define a special slack variable δ that is common to all constraints. This variable represents the minimum distance of a trial solution from each constraint. We direct phase two of the simplex method to maximize δ . It does so and produces the desired result directly and efficiently. Robust solutions for many networks have been found. All of the probabilities of realization for the cases involving diagonal and column constraints were obtained with this method. We look forward to using our methods to determine the solution space of a lobster, to finding if a particular lobster's actual solution is the same as our robust solution, and to comparing solution spaces and specific solutions for various lobsters.

Phase two of the simplex method can also be used to deal with bioneural systems having inconsistent firing patterns for which there are no realizable solutions. Such cases may occur due to changing connection weights arising from biological noises. To find a solution in these inconsistent cases, we again use the common slack variable δ , but on this occasion we find the solution associated with the least negative value for δ . Basically we have pushed each constraint boundary back a minimum amount until a point solution is obtained. A solution obtained in this way may be viewed as a central solution in a noisy space, with the value of δ being a measure of the noise inherent to the system under study. The model with such a solution will act much like the original system with the noise removed.

8 Summary of Results

We have found an identification method to determine the connections between artificial neurons of a network from the behavior of the network. We applied the method to identify

neural networks. We found that the probability of realization of firing patterns increases as the number of neurons increases and as the number of time steps decreases.

We identified network connection constraints associated with biological structures and found ways to incorporate them in our design methods. We applied our biologically tailored identification method to identify neural networks having biological network constraints such as zero diagonals.

We have developed a method for finding a central robust solution for neural networks and an analogous procedure for dealing with networks having inconsistent temporal firing patterns.

9 Conclusions

We have developed a new tool for identifying the connections of biologically-inspired neural networks. This tool is important because it is extremely difficult, if not impossible, to measure neural connections in bioneural networks. The availability of such a tool gives hope that a combination of experiments and calculations can be applied to determine the detailed structure of biological neural networks such as visual systems, and that the knowledge obtained would enable us to build their electronic counterparts on VLSI circuit chips. The method that has been developed also might be used to guide the repair of damaged biological neural networks, to develop neural prostheses and to build electronic recognition systems based directly on biological systems.

We recognize that our work is only a beginning and that we need to include more biological features of neurons in our model before it can reasonably approximate living systems. After improvements we hope to interpret firing patterns taken by neuroscience researchers.

10 Acknowledgements

We wish to thank Professor Gary Maki, Director of the University of Idaho's NASA Space Engineering Research Center, Professor Martin Hagan of Oklahoma State University, Professors Mark DeSantis and James Calvert of the University of Idaho, Professors Katherine Demuth and Mark Johnson of Brown University, and Margi Jenks for their suggestions, criticism, and support.

This work was funded originally by a research seed grant from the University of Idaho, and is currently funded by NASA Space Engineering Research Center grant NGW-1406 to the University of Idaho.

References

- [1] T. Bullock, R. Orkland and A. Grinell, *Introduction to Nervous Systems*, CA: W. H. Freeman and Co., San Francisco, 1974

- [2] J. Calvert and W. Voxman, *Linear Programming*, Harcourt Brace Jovanovich, New York, NY, 1989
- [3] "DARPA Neural Network Study, Executive Summary," MIT Lincoln Laboratory, Lexington, MA, 1988
- [4] Personal communication from Martin Hagan, March 19, 1990
- [5] J. Hicklin, "Neural Network Design," University of Idaho Ph.D. Dissertation, November 1988
- [6] E. Kandel and J. Schwartz, "Principles of Neuroscience," Elsevier, New York, NY, 1985
- [7] K. Leung, "Identification of Biologically-Inspired Neural Networks," M.S. Thesis, Computer Engineering, University of Idaho, 1990
- [8] D. Nguyen and B. Widrow, "The Truck Backer- Upper," Proceedings of the International Neural Network Conference (INNC-90- Paris), 1990, pp.399-407
- [9] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge MA, 1986
- [10] A. Selverston and B. Mulloney, "Synaptic and Structural Analysis of a Small Neural System," in F. O. Schmitt and F. G. Warden (eds,) *The Neurosciences, Third Study Program*, pp 389-395, MIT Press, Cambridge MA, 1974
- [11] I. Selverston, "Neural Mechanisms for the Production of Cyclic Motor Patterns," *IEEE Trans on Sys, Man, and Cyb*, Vol smc-5, 1983, pp. 749-757
- [12] A. Willows, "Giant Brain Cells in Mollusks," *Scientific American*, Vol 224, No. 2, 1971, pp. 69-75