LETTER Communicated by Terrence Sejnowski

The Dynamic Neural Filter: A Binary Model of Spatiotemporal Coding

Brigitte Quenet
Brigitte.Quenet@espci.fr
Laboratoire d'Electronique, Ecole Superieure de Physique et Chimie Industrielles, Paris 75005, France

David Horn
horn@post.tau.ac.il
School of Physics and Astronomy, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel

We describe and discuss the properties of a binary neural network that can serve as a dynamic neural filter (DNF), which maps regions of input space into spatiotemporal sequences of neuronal activity. Both deterministic and stochastic dynamics are studied, allowing the investigation of the stability of spatiotemporal sequences under noisy conditions. We define a measure of the coding capacity of a DNF and develop an algorithm for constructing a DNF that can serve as a source of given codes. On the basis of this algorithm, we suggest using a minimal DNF capable of generating observed sequences as a measure of complexity of spatiotemporal data. This measure is applied to experimental observations in the locust olfactory system, whose reverberating local field potential provides a natural temporal scale allowing the use of a binary DNF. For random synaptic matrices, a DNF can generate very large cycles, thus becoming an efficient tool for producing spatiotemporal codes. The latter can be stabilized by applying to the parameters of the DNF a learning algorithm with suitable margins.

1 Introduction

Three types of neural network paradigms (Hertz, Krogh, & Palmer, 1991; Peretto, 1992) were developed two decades ago. Two of them were based on supervised learning, in order to construct feedback attractor networks (Hopfield, 1982) for associative memory and feedforward multilayer perceptrons (Rumelhart, Hinton, & Williams, 1986) for functional representation. The third used unsupervised learning to produce self-organized maps (Kohonen, 1982). These three conventional paradigms of neural computation were formulated in terms of mathematical neurons that are very different from biological reality, and it is still unclear how, if at all, biological neural

networks perform any of them. Over the past decade, interest in theoretical neuroscience (Dayan & Abbott, 2001) has shifted to analyzing neural networks of spiking neurons with dynamical synapses in the hope of getting closer to the natural roles and purposes of the relevant biological networks. We will argue that it is nevertheless timely to consider yet another model of binary neurons with discrete temporal dynamics and show its applicability to a biological system.

An interesting issue in neuroscience is the question of spatiotemporal coding. Its existence has been demonstrated (Wehr & Laurent, 1996) in the locust olfactory system, where the spatiotemporal behavior of projection neurons encodes the odor presented to its receptor neurons. This transformation from odor input to spatiotemporal activity occurs in the antennal lobe, that is, the first module of the olfactory system. This system may therefore be regarded as a dynamic neural filter that turns spatial information distributed over its many glomeruli, fed by the receptor neurons, to specific spatiotemporal outputs. Although this is a complicated biological system, it has an important simplifying feature that allows it to be represented by a simplified model of mathematical neurons: the fact that the activity of the projection neurons is limited to temporal bins defined by an oscillatory local field potential. Hence, a model with binary neurons obeying Hopfield-Little dynamics (Peretto, 1992) can provide a valid first-order approximation of the spatiotemporal behavior of that system. We study a recurrent model of this kind using an asymmetric coupling matrix that allows for the generation of large temporal sequences. The novelty of our approach is that we use this model as a dynamic neural filter (DNF), relating input space to spatiotemporal behavior of the recurrent network. As such, it does not correspond to any of the three paradigms noted above, it does not come to rest at fixed points, and it is not necessarily based on supervised learning.

We have presented this model in a previous work (Quenet, Horn, Dreyfus, & Dubois, 2001) and demonstrated how it can be used to generate the spatiotemporal behavior of projection neurons observed by Wehr and Laurent (1996). Here we discuss several fundamental issues concerning this model, such as its stability and coding capacity. Moreover, we provide an algorithm for constructing a DNF that can generate a given set of spatiotemporal data. This algorithm is used to define and characterize the complexity of the data set of Wehr and Laurent (1996). It should be emphasized that the latter is used only as a relevant example, and we do not claim that the DNF is a realistic representation of the biological reality. Nonetheless, as in the conventional three paradigms, it can be used as a schematic model that will subserve further detailed modeling of biological systems.

2 The Model

2.1 Short Biological Motivation. The antennal lobe of insects serves as the first stage of olfactory computation, using as input signals of chemical

receptors and transforming them into outputs of projection neurons (PN) to the next stage of the olfactory tract (the mushroom body). In the locust, there are about 800 excitatory PN cells (Laurent and Naraghi, 1994) and a slightly smaller number of inhibitory local interneurons (LN) that form dendrodendritic interactions with the PNs and with themselves, and receive receptor information as well. The output of this system is that of PNs only; hence, we will consider our model as a rough sketch of PNs, with interactions that should be regarded as effective interactions mediated by the LNs.

The antennal lobe exhibits a reverberating field potential (LFP) at a frequency of about 20 Hz. This LFP reverberates for a few cycles (up to tens of them) and then quiets down until it begins to repeat such behavior. (See Laurent et al., 2001, for details.) The PNs fire within the up-phase of the LFP; hence the latter may be regarded as defining an effective clock, with time bins of, for example, 30 ms of activity and 10 ms of rest. This motivates us to use a simplified model with discrete temporal dynamics.

2.2 The Recurrent Network. In our model, a binary neuron i, representing a PN, may either fire, $n_i = 1$, or be quiescent, $n_i = 0$, in a given temporal bin of the LFP. There are N neurons in the model obeying the following Hopfield-Little dynamics,

$$n_i(t+1) = H(h_i(t+1)) = H\left(\sum_j w_{ij} n_j(t) + R_i - \theta_i\right), \tag{2.1}$$

where w_{ij} is the synaptic coupling matrix, R_i is the external constant input (specifying odor activation), and θ_i is the threshold. H is the Heaviside step function taking the values 0 for nonpositive arguments and 1 for positive ones.

This model can be readily generalized to account for the presence of noise by replacing the deterministic rule by the stochastic one,

$$Prob[n_i(t+1) = 1] = \frac{1}{1 + e^{-h_i(t+1)/\epsilon}}. \tag{2.2}$$

where ϵ is the noise parameter. We will study the stability of results of the deterministic dynamics to the noise introduced by the stochastic dynamics.

2.3 Deterministic Dynamics. In order to analyze the one-step dynamics of equation 2.1, we define a Lyapunov function L that describes just this one-step, in the sense that, considering all dynamical states at time $t+1$, it will be minimal for the state that is a solution to these dynamics. Let us define the initial and final states,

$$n_i^I = n_i(t) \qquad n_i^F = n_i(t+1), \tag{2.3}$$

relevant to the one-step of equation 2.1. Clearly, I and F are just two of the 2^N states that the system of neuron possesses. We can prove that

$$L^{JI} = -\sum_{ij} w_{ij} n_i^J n_j^I - \sum_i n_i^J (R_i - \theta_i) \tag{2.4}$$

obtains its unique minimum on the state $J = F$.

Consider

$$k_i^{JI} = -(2n_i^J - 1)h_i^I = -(2n_i^J - 1)\left(\sum_j w_{ij} n_j^I + R_i - \theta_i\right). \tag{2.5}$$

For $J = F$, this quantity is negative for every i, following the deterministic dynamics. For any other J, there will be elements i for which the sign will flip. Hence, $\sum_i k_i^{JI}$ will be minimal for $J = F$. We note in equation 2.5 that k_i^{JI} contains terms that are dependent on n_i^J, to be denoted by $2l_i^{JI}$ where

$$l_i^{JI} = -n_i^J \left(\sum_j w_{ij} n_j^I + R_i - \theta_i\right), \tag{2.6}$$

and terms that are independent of n_i^J. Since the latter are common for all J, the minimum of $\sum_i k_i^{JI}$ will be reached for the same J as the minimum of $L^{JI} = \sum_i l_i^{JI}$, the Lyapunov function of equation 2.4.

2.4 Stochastic Dynamics. The one-step Lyapunov function, equation 2.4, plays an important role in the stochastic dynamics.[1] As we will see, the probability of obtaining a state J after starting from a state I can be written in terms of this function:

$$P(J \mid I) = \frac{e^{-L^{JI}/\epsilon}}{\sum_K e^{-L^{KI}/\epsilon}}. \tag{2.7}$$

To prove this result, we start from the probability of obtaining the state F, that is, the one following from the deterministic dynamics. It is straightforward to show that

$$P(n_i^F \mid I) = \frac{1}{1 + e^{k_i^{FI}/\epsilon}}; \tag{2.8}$$

[1] It was applied to a model with symmetric synaptic couplings by Quenet, Cerny, Dreyfus, and Lutz (1997).

hence,

$$P(F \mid I) = \Pi_i P(n_i^F \mid I). \tag{2.9}$$

Any other state J differs from the state F by some flips of neuronal states, for example, $n_m^J = 1 - n_m^F$. In such a case, $P(J \mid I)$ will be the probability of obtaining the corrupted state F, having the wrong digit in its mth place, which is $P(F \mid I)e^{k_m^{FI}/\epsilon}$. But at this location m,

$$k_m^{FI} = \frac{1}{2}(k_m^{FI} - k_m^{JI}) = l_m^{FI} - l_m^{JI}. \tag{2.10}$$

Since for all other neurons $l_i^{FI} = l_i^{JI}$, it follows that

$$P(J \mid I) = P(F \mid I)e^{(L^{FI} - L^{JI})/\epsilon}. \tag{2.11}$$

This will be true for any state regardless of how many flips occur because of their independent nature. Hence, equation 2.7 is proved.

The stochastic dynamics of equations 2.1 and 2.2 lead to a homogeneous and irreducible Markov process; the transition matrix is time invariant, and all elements of the transition matrix,

$$T_{JI} = P(J \mid I), \tag{2.12}$$

are strictly positive. Such a Markov process has a stationary probability distribution, $p(I)$, that is, an eigenvector of the transition matrix with the highest eigenvalue, 1:

$$\sum_I T_{JI} p(I) = p(J). \tag{2.13}$$

From this stationary probability distribution, obtained asymptotically in the evolution of the system, one can calculate the asymptotic probabilities of the transitions $p(I \to J)$:

$$p(I \to J) = P(J \mid I)p(I). \tag{2.14}$$

The stationary Markov process has an entropy rate (Cover & Thomas, 1991),

$$\mathcal{H} = -\sum_{IJ} p(I) T_{JI} \log_2 T_{JI}, \tag{2.15}$$

that serves as a lower bound on the expected dimension of any binary code of the Markov process. Since in our case all states are described by vectors of length N, it follows that

$$\mathcal{H} \leq N. \tag{2.16}$$

The upper limit is approached when the noise parameter ϵ is high and all transition matrix elements tend to be equal.

3 Numerical Examples

We propose to view the model described in the previous section as a filter-relating input space R_i to spatiotemporal sequences of the recurrent network. In this section, we will demonstrate numerically how the DNF acts. For simplicity, we choose both w_{ij} and R_i to have positive and negative integer values, while fixing $\theta_i = \frac{1}{2}$. This means that R_i will now effectively represent both input and threshold of neuron i.

3.1 Mapping of Input Space. Once the matrix $\mathbf{w}$ is given, there is a restricted range of interest for R:

$$-\sum_j w_{ij}H(w_{ij}) \leq R_i \leq -\sum_j w_{ij}H(-w_{ij}) + 1. \tag{3.1}$$

Outside this range the dynamics becomes trivial because the input determines the neuronal values directly.

As a first example of deterministic dynamics, we study an $N = 2$ system because it can be easily mapped in an exhaustive manner. We choose the synaptic matrix

$$\mathbf{w} = \begin{pmatrix} 1 & 2 \\ -2 & -1 \end{pmatrix}$$

and use an initial null state $n_i(0) = 0$. The dynamics generate 14 different sequences, including the four (2^N) fixed points, several two-cycles, and one four-cycle. In Figure 1, we display the coding zones, that is, the ranges of R corresponding to a single temporal sequence. The overall range of R space is chosen to be somewhat larger than the range of equation 3.1. In its center, we find that many different sequences are produced (small coding zones), while in the periphery, a single sequence dominates. Figure 2 shows the length of each sequence on the same R plane. We define it as the length of the sequence the model generates before one of its states is being repeated. This definition of length takes into account both the order of the cycle and the transition time it takes to reach it. The interesting (i.e., long) sequences appear near the center of R space. The periphery is dominated by fixed points and two-cycles.

Building on this experience, we next study an $N = 5$ system with the synaptic matrix

$$\mathbf{w} = \begin{pmatrix} 0 & -2 & -5 & -3 & 0 \\ 6 & 2 & 8 & -14 & 0 \\ 1 & 1 & 0 & -2 & 1 \\ -4 & 6 & 1 & 1 & 3 \\ 4 & -1 & 2 & -4 & 0 \end{pmatrix}.$$

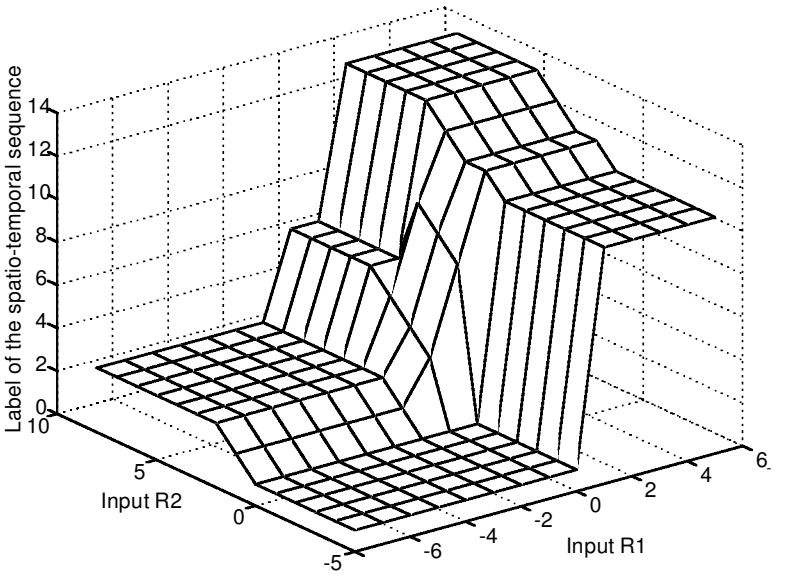

Figure 1: Ranges of input space in a two-neuron problem that lead to identical spatiotemporal behavior, defined as coding zones. The label of each code, or spatiotemporal behavior, varies from 1 to 14 in this problem.

Now much larger sequences can be generated, and we would like to see the ensuing mapping of input space and test the stability of the sequences to noise. In order to be able to view the results, we choose R_3 R_4 R_5 to lie in the center of their expected range, as specified by equation 3.1, and study the system in the R_1 R_2 plane, limited to their relevant ranges.

First, let us view, in Figure 3, the coding zones of different sequences in R space. Their average size is 10; that is, there is an order of magnitude of information compression from R space to the spatiotemporal sequences. The lengths of the latter are displayed in Figure 4. They include many cycles of lengths 4, 5, and 6.

The coding zones are the analogs of basins of attraction, which signify domains of initial conditions that lead to the same attractor. Here, we stay with a fixed initial condition (the null state) and search for domains in input space that lead to the same temporal sequence (transition into a cycle). Note the large number of different sequences obtained in this problem. It is exponentially larger than the number of fixed points that we are accustomed to in attractor neural networks (describable by the same dynamics with symmetric synaptic matrices). We estimate, on the basis of simulations, that the total number of sequences in the five-dimensional R space is of the order of 3800. In the two-dimensional section of R space shown in Figure 3, we find 38 different sequences, whose lengths are displayed in Figure 4.

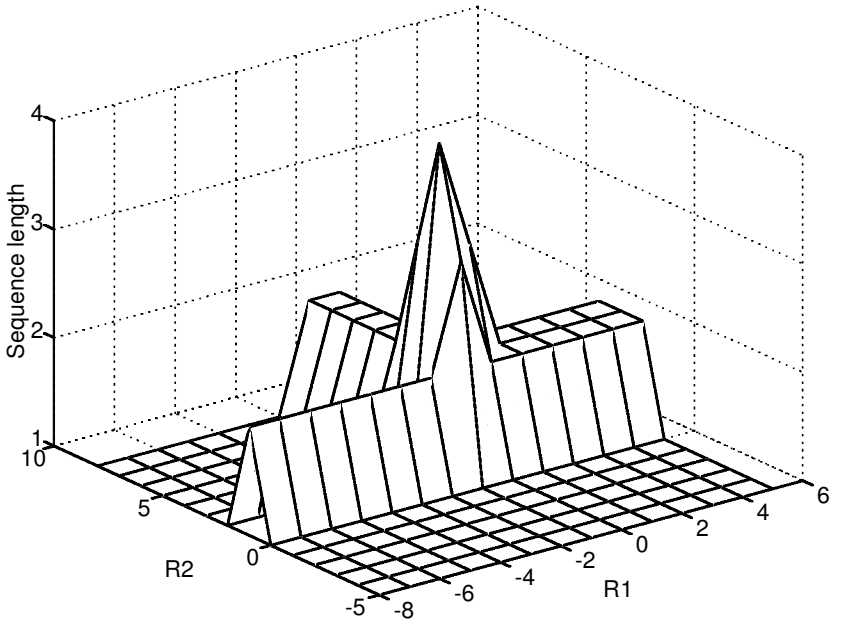

Figure 2: Sequence length of the $N = 2$ problem mapped on the two-dimensional input space.

Table 1: Six Neighboring Sequences in a System with Five Neurons, Displayed over Seven Time Steps.

R_2	D_E	D_{H7}	$t = 1$	2	3	4	5	6	7
−15	0	0	17	22	6	8	3	17	22
−12	2	1	17	22	14	8	3	17	22
−8	4	2	17	22	14	16	3	17	22
−3	5	10	17	30	16	3	17	30	16
2	6	11	25	30	16	3	17	30	16
8	7	7	25	30	16	11	3	17	30

3.2 Distances Between Codes. The mapping of R space into spatiotemporal codes can also serve as defining a distance between different codes. It is thus interesting to compare different sequences generated over a neighborhood of R space, as seen in Table 1. All of these sequences can be read off as we move along the R_2 axis at fixed $R_1 = 4$. The sequences are ordered according to increasing value of R_2, where we indicate the central value of the coding zone, and the states for each time bin are written in the binary representation $1 + \sum_{i=1}^{5} n_i 2^{5-i}$. The columns D_E and D_{H7} describe two distances, defined below, from the first sequence.

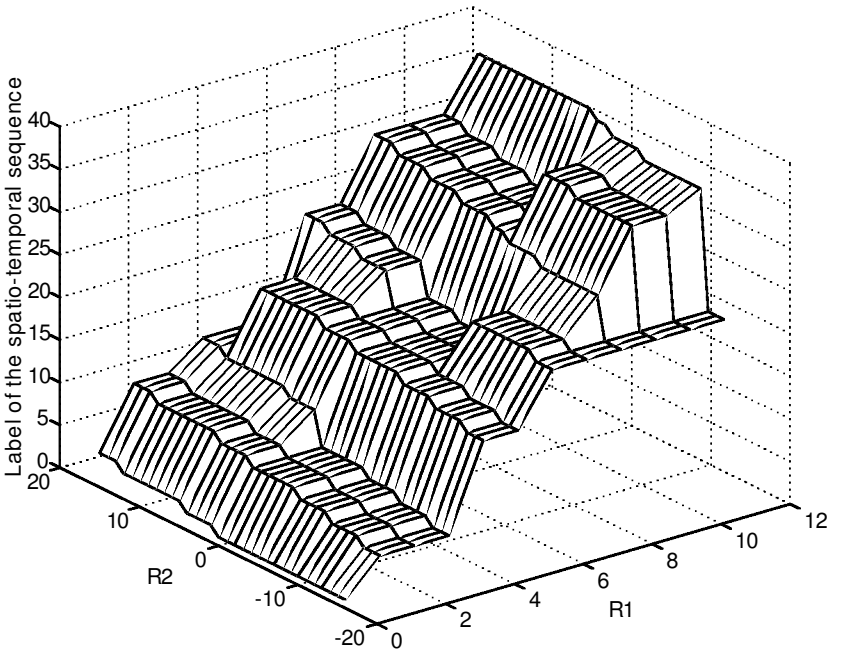

Figure 3: Coding zones of spatiotemporal sequences of an $N = 5$ problem, plotted on a plane defined by two of the input variables. Other inputs are held constant at their central values.

As one moves along from the first sequence to the second, and then to the third, only one state changes over the observed range of $T = 7$. Moreover, this change corresponds to the flipping of just one neuron (n_2 at $t = 3$ between the first and second sequence). Proximity in R may therefore be related to proximity in the identity of states appearing in two sequences, or proximity in the sense of Hamming distance between all neurons at all time steps. Thus, we define three distances between sequences: D_E is the "edit distance," defined as the number of insertions and deletions needed to change one sequence into another, limiting oneself to the natural length of each sequence, that is, until one of the states reappears. D_H is the Hamming difference between the two spatiotemporal patterns of neural activities. Clearly this grows with the length of the sequence. In the example of Table 1, where we limit ourselves to seven time steps, we list the corresponding seven-step Hamming distance, D_{H7}. Finally, we may define D_R, the Euclidean distance between the centers of the coding zones in R space. Table 1 lists the values of D_E and D_{H7} distances from the first sequence in this table. Using 18 sequences of this problem, including the six shown in Table 1, we find a high correlation (0.87) between D_E and D_R and a lower cor-

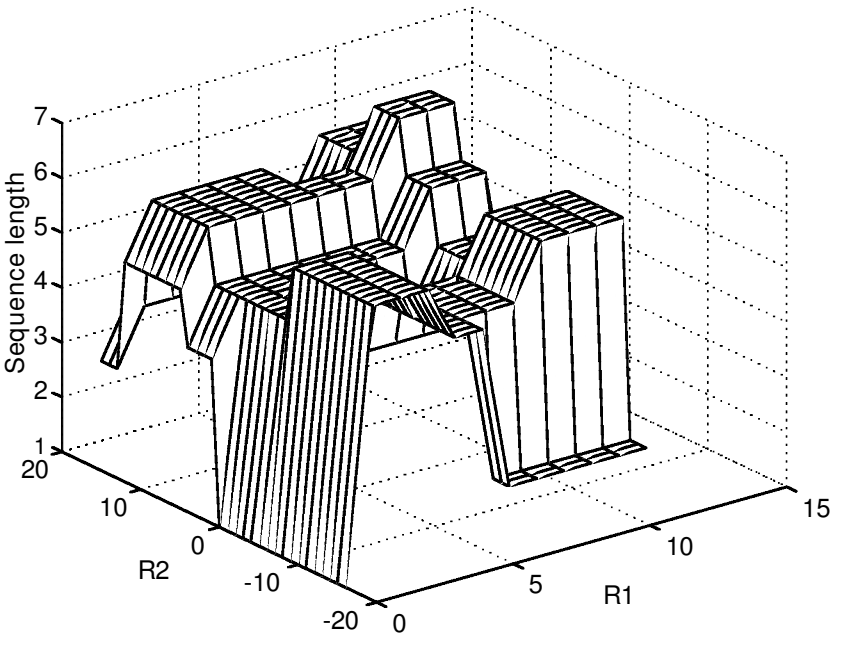

Figure 4: Sequence length of the 38 sequences mapped out in Figure 3.

relation (0.56) between D_{H7} and D_R. The latter signifies the fact that whereas the Hamming distance correlates well with R distance for some sequences, there are cases where several neurons flip as one moves from one code to the next. In Table 1, this occurs between the third and the fourth sequences and between the fifth and the sixth ones. Note, however, that in the fourth sequence, the state 30 replaces two states, 22 and 14, of the third sequence. Therefore, D_E between the two will be only 3. This explains in what sense the two sequences should be regarded as close to one another, as also borne out by the short distance in R.

These results can also be studied from the viewpoint of the individual neurons. Since the first line has the lowest value of R_2, none of its states involves an active neuron number 2, that is, all have $n_2 = 0$. Moving up to $R_2 = -12$, n_2 is activated in state 14. At the next stage, it is activated in state 16. Further increases of R_2 lead to activation of neuron number 2 in the states 30 and 25. Thus, we observe a gradual increase in the number of states that contain an active neuron number 2. Since the volume of state-space is relatively moderate, 32, we find that in spite of the change in the identity of the state, for example, a change of 6 to 14 in step $t = 3$ between the first and the second row, the next state remains 8, and the rest of the sequence is reiterated. This leads to the good correlation with D_E. We believe that this

property may be useful in DNF applications to the study of problems such as gene sequences, where low N models may apply.

The correlation between R distance and edit distance may be lost at large N. The reason is that the volume of state-space grows like 2^N. Hence, once a change occurs in one state, the next transition may point to another state in this large space, and the ensuing sequence may change completely. Numerical trials in an $N = 50$ model have shown this to be the case. Thus, the correlation of nearby sequences, as defined by proximity in R space, may be lost in large neural networks.

3.3 Stability Under Stochastic Dynamics. Next, let us ask how robust these sequences are against noise. We choose $\epsilon = 0.5$, which is where considerable effects may be expected, since this is the lowest absolute value that the potentials h_i can have. As an example, let us look at the probability of a correct four-step sequence,

$$P_4 = P(F_4 \mid F_3)P(F_3 \mid F_2)P(F_2 \mid F_1)P(F_1 \mid 1), \tag{3.2}$$

where 1 designates the initial null state. This is the product of the probabilities of obtaining the correct first four states of the appropriate deterministic sequence at a given point in R space. The results for the problem at hand are plotted in Figure 5 on the same grid of R space as in the two previous figures. We see that a few sequences are relatively stable, but others have a low probability of being correctly recovered during the first four steps of this Markov process.

To exemplify the origins of instability we compare two six-cycles, at locations (10, −10) and (10, 15) of Figures 4 and 5. In Figure 6 we plot the histograms of the h_i values (over all neurons for the first four steps of the dynamics) for these two cases. Clearly, at $\epsilon = \frac{1}{2}$, the occupancy of the bins of $h = \pm\frac{1}{2}$ will determine the instability following from equation 2.2. Since the sequence at (10, −10) has three elements in these bins, it is reduced to $P_4 = 0.36$. All other sequences in Figure 5 have three elements in these bins or more. The sequence at (10, 15) has five elements in these bins, as can be seen in Figure 6; hence, its retrieval probability is reduced to $P_4 = 0.18$.

Let us remark at this point on the Markov chain properties discussed in section 2.4 and, in particular, the stationary probability distribution $p(I)$ of equation 2.13. Using the R-space point (10, −10) as an example, we find that the following states are the most probable ones: 17, 22, 30, 32. Their respective probabilities are 0.106, 0.175, 0.173, 0.169. In the problem at hand, we start with state 1 as the initial condition, looking for the sequence $1 \to 17 \to 22 \to 30 \to 32$. The relevant transition probabilities for the terms in equation 3.2 are $0.98 \times 0.95 \times 0.73 \times 0.53$, leading to $P_4 = 0.36$. Obviously, the major reduction of the probability occurs at the first two steps, from state 1 to 17 and from 17 to 22. This is also where the three appearances of

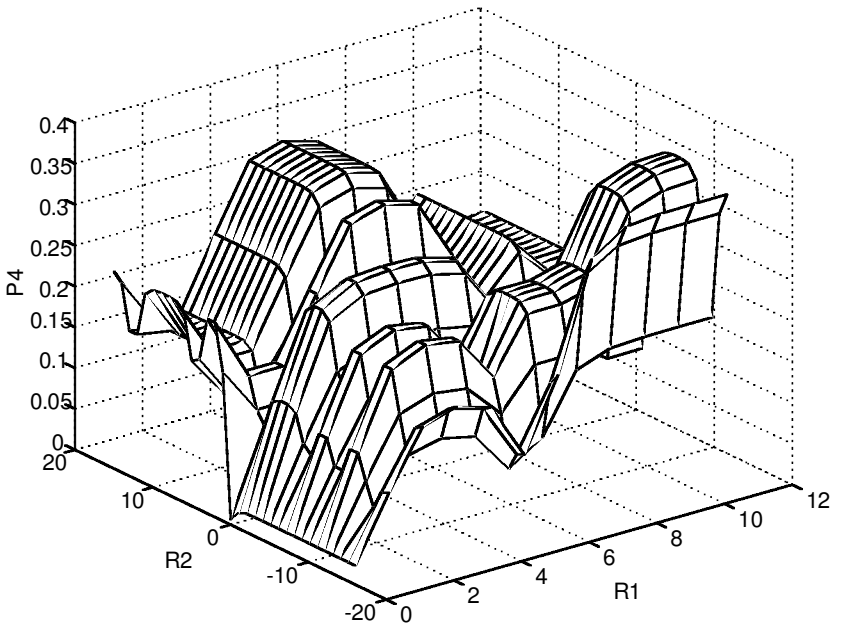

Figure 5: Probability of recovering correctly the first four time steps in the sequences of Figures 3 and 4 when stochastic dynamics is being used with $\epsilon = 0.5$.

$h = \pm\frac{1}{2}$ occur: two of them affect the transition of 1 to 17, and one occurs in the transition from 17 to 22.

3.4 Entropy of the Markov Chains. Finally, we show in Figure 7 an analysis of the entropy of the different Markov chains of this system. This is done for $\epsilon = 0.5$ and can be compared with the P_4 results in Figure 5. The Markov chains with low entropy are the ordered ones, leading to a relatively high probability of correct retrieval of the first four states. The chains with high entropy are the disordered ones, leading to a low probability of correct retrieval.

4 The Inverse Problem

The previous section exhibited examples of what may be called the direct DNF problem: given a matrix $\mathbf{w}$ and a set of R values, solve the dynamics of equation 2.1, and generate the resulting sequence. We now wish to pose an inverse problem: given a set of several sequences, find a corresponding DNF, that is, values of $\mathbf{w}$ and R that can produce this set. There are two parts to this question. First, does a solution exist? Second,

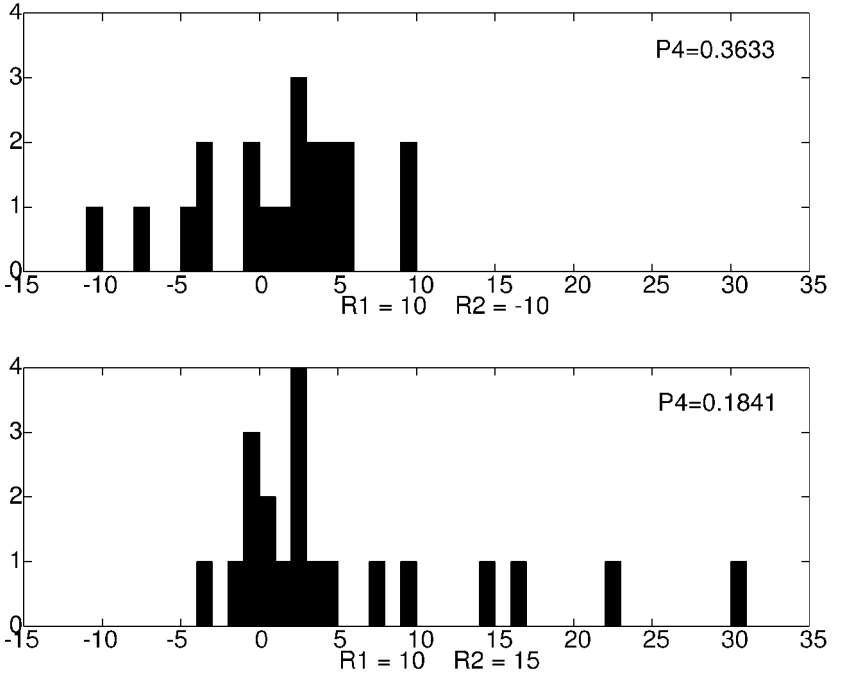

Figure 6: Histograms of h_i values of two six-cycles in the $N = 5$ problem.

Table 2: Six Spatiotemporal Sequences Defined for Four Time Steps in a System with Four Neurons.

t/k	1	2	3	4	5	6
1	1100	1000	1110	1000	1011	1000
2	1110	1100	1111	1010	1000	1110
3	1101	1101	0111	0110	1110	0111
4	0001	0001	0011	0111	1111	0001

if the problem is soluble, find an example. This may be expanded into a search for all possible solutions, which lies outside the scope of this work.

4.1 The Existence Problem. Suppose a given data set comprises K different spatiotemporal sequences of length T each, presented as activities of N neurons. An example is shown in Table 2 for $K = 6$, $T = 4$ in a system with $N = 4$ neurons. We wish to find out if there exists a DNF such that this set of sequences is elicited by K different inputs R_i^k, $k = 1, \ldots, K$.

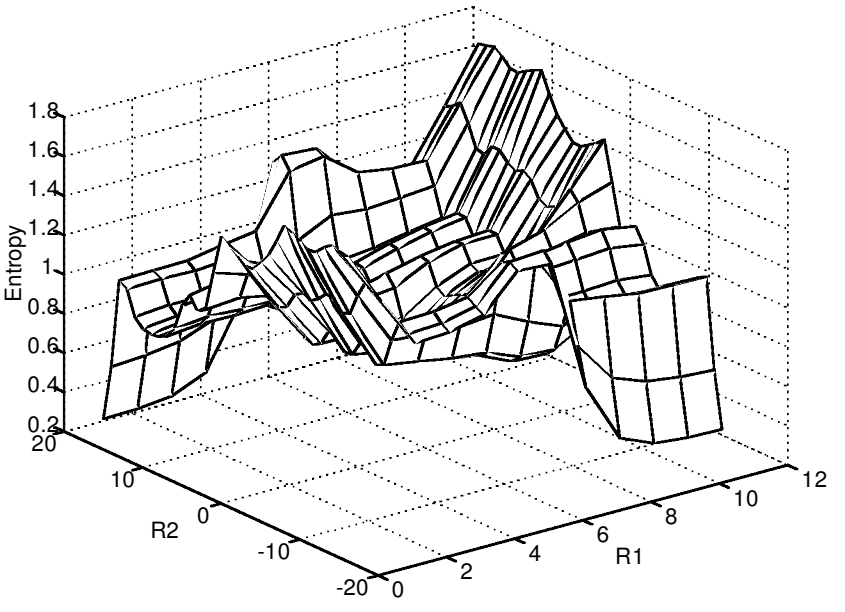

Figure 7: Entropy of the Markov chains corresponding to the sequences of Figures 3 and 4 when stochastic dynamics is being used with $\epsilon = 0.5$.

The neuronal values may be labeled $n_i^k(t),\ t = 1, \ldots, T$. Each time step, for each sequence, defines a state of the system

$$|k, t\rangle = \{n_i^k(t)\}_{i=1,\ldots,N}. \tag{4.1}$$

Twenty-four such states can be seen in Table 2, some of them identical, yet no identical states appear in a single sequence. Each state $|k, t\rangle$ follows from the previous state $|k, t-1\rangle$ through deterministic dynamics. The initial state $|k, 0\rangle$ is chosen as the null state $n_i^k = 0$ for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$.

Consider a single neuron i in this system. We may then reduce the question posed here into a perceptron problem for this neuron. The neuron is given KT initial vectors (counting states from $t = 0$ to $t = T - 1$) and its required output values $n_i^k(t)$ (from $t = 1$ to $t = T$ for all $k = 1, \ldots, K$). It has to perform calculation 2.1 given a different input R_i^k for each of the K sequences. To take account of the K different inputs (biases), we extend the vector states of length N to new vectors of length $N + K$ through the concatenation

$$|k, t) = |k, t\rangle \times \{\delta_{a,k}\}_{a=1,\ldots,K}, \tag{4.2}$$

where $\delta_{a,k}$ is the Kronecker delta function obtaining the value 1 for $a = k$ and 0 otherwise. The extension by K bias axes allows us to represent the

deterministic dynamics, for each neuron i, as KT perceptron inequalities in an $N + K$ dimensional system. These can be strictly met (Hertz et al., 1991) for N obeying

$$\mathcal{A}: \quad KT \leq N + K \quad \text{or} \quad K(T-1) \leq N. \tag{4.3}$$

In the large N limit, one can apply the Cover result (Cover, 1965), saying that a solution may be found for

$$\mathcal{B}: \quad KT \leq 2(N+K) \quad \text{or} \quad \frac{1}{2}K(T-2) \leq N. \tag{4.4}$$

For the example of Table 1, the strict condition $\mathcal{A}$ implies $N \geq 18$, and condition $\mathcal{B}$ leads to $N \geq 6$. This could mean one should extend the system shown in Table 2 by at least two, if not more, neurons. Nonetheless, one can obtain a solution for $N = 4$. This is due to the fact that the states in this table were not chosen in a random and independent fashion.[2] Note also the example discussed in section 3.1. Thirty-three of the 38 sequences in Figure 4 have length 4 or more. Counting only the first four states, we find 22 different sequences, many more than conditions $\mathcal{A}$ and $\mathcal{B}$ would suggest. This is so because we did not ask in section 3.1 for the matrix that can produce an arbitrary choice of states appearing in K sequences of length 4, but counted how many different sequences were produced by a given matrix. These sequences turned out to be quite similar to one another, as described in section 3.2.

Thus, conditions $\mathcal{A}$ and $\mathcal{B}$ should be regarded as sufficiency conditions, assuring us $\mathcal{A}$ that a solution must exist and $\mathcal{B}$ that it may be found for lower N. With more effort, one can try to solve the inverse problem, that is, find an explicit matrix $\mathbf{w}$, for yet smaller N.

4.2 Perceptron Learning. From our approach to the existence problem, it should be quite evident that if a solution exists, it may be obtained using a straightforward extension of the Rosenblatt algorithm (Rosenblatt, 1962; Hertz et al., 1991). We describe it in this section, realizing explictly the abstract vectors in $N + K$ dimensions described in the previous section.

Let us start by defining, for each neuron i, an $N + K$–dimensional vector of perceptron weights $\vec{w}^i$,

$$(\vec{w}^i)_j = w_{ij} \quad \text{for } j = 1, \ldots, N \qquad (\vec{w}^i)_{N+k} = R_i^k - \theta_i \quad \text{for } k = 1, \ldots, K. \tag{4.5}$$

The sequence states of interest will be represented by vectors $\vec{x}^{i,k}(t)$, for neuron i, living in the same $N + K$–dimensional space but carrying further

[2] The rationale for the choice of states in Table 2 is discussed in section 5.3.

indices of k and t as follows:

$$(\vec{x}^{i,k}(t))_j = n_j^k(t)(2n_i^k(t+1) - 1) \quad \text{for } j = 1, \ldots, N,\ t = 0, \ldots, T-1 \tag{4.6}$$

$$(\vec{x}^{i,k}(t))_{N+a} = \delta_{ak}(2n_i^k(t+1) - 1) \quad \text{for } a = 1, \ldots, K.$$

In other words, the vectors $\vec{x}^{i,k}(t)$ represent the states weighted by a positive or negative sign depending on whether the target neuron i at the next time step will be 1 or 0, respectively. With these definitions, all constraints of the inverse problem can be written as KT perceptron inequalities:

$$\vec{w}^i \cdot \vec{x}^{i,k}(t) > 0 \quad \text{for all } k = 1, \ldots, K,\ t = 0, 1, \ldots, T-1. \tag{4.7}$$

We propose to use the perceptron learning rule (Hertz et al., 1991),

$$\Delta\vec{w}^i = \eta\vec{x}^{i,k}(t)H(-\vec{w}_i \cdot \vec{x}^{i,k}(t)), \tag{4.8}$$

while iterating the system, time and again, over all KT states. The Heaviside function guarantees that $\vec{w}^i$ gets modified at a given iteration by the vectors $\vec{x}^{i,k}(t)$ that do not satisfy the inequality. This algorithm converges (Hertz et al., 1991) if the system of inequalities is soluble.

Moreover, one can use the same algorithm to require stability under stochastic dynamics by insisting on a margin M in the perceptron inequalities:

$$\vec{w}^i \cdot \vec{x}^{i,k}(t) > M. \tag{4.9}$$

This leads to the learning rule:

$$\Delta\vec{w}^i = \eta\vec{x}^{i,k}(t)H(-\vec{w}^i \cdot \vec{x}^{i,k}(t) + M). \tag{4.10}$$

In section 3, where we used integer values of $\mathbf{w}$ and R and kept all $\theta = \frac{1}{2}$, we worked with an implicit margin of size $\frac{1}{2}$. This is the reason that testing the system with stochastic dynamics of $\epsilon = \frac{1}{2}$, we found considerable effects. Insisting on larger M, if the data allow it, will guarantee stability for larger ranges of ϵ. Given a situation of the type encountered in Figure 5, one may want to improve the robustness of a possible code, such as the one located at (10,15) in the R plane. This can be tried by applying the algorithm presented by the previous equation for a suitably chosen M. The size of M is, of course, limited by the data. (For a general discussion of optimal margin selection, see Vapnik, 1995.)

If the algorithms are implemented with $\eta = 1$, they lead to integer values of $\mathbf{w}$ and $R - \theta$. An arbitrary choice of $\eta < 1$ can lead to continuous h values. It is then advisable to keep a finite M to ensure some robustness to noise.

Finally, we wish to comment that one may apply in a similar manner the Ho-Kashyap procedure (Duda, Hart, & Stork, 2001), indicating, if it does not converge, that a given set of data is not linearly separable.

5 Capacity and Complexity

5.1 Capacity of Codes of Given Length. The DNF produces sets of spatiotemporal codes. Its capacity could be measured by the number of codes it can produce. This is a very large number and may not be very meaningful if we consider an application with stochastic dynamics. Moreover, what would be really interesting is a measure of capacity of random codes, expected to be very distant from one another. Clearly, two codes that are close will not constrain the system. If one is learned, some other close-by codes will be automatically produced for nearby R values.

Suppose we look for the number of codes of length T such that the same state does not appear twice, either within the same code or among the different codes. This would be the case if the states are chosen randomly from all 2^N possibilities of an N-neuron system. Following the reasoning of section 4.1, this number can be expected to reach

$$C_T \approx \frac{2N}{T-2} \tag{5.1}$$

if criterion $\mathcal{B}$ is applied. C_T may be viewed as a capacity measure of a DNF.

Note that the last equation leads to $C_T = 1$ for $T = 2N + 2$. This means that a DNF of order N may be expected to accommodate a cycle of length $2N + 2$. The latter may then be used to construct the DNF.

5.2 Random Synaptic Weights. There exist synaptic matrices that do not have the ability to generate large cycles. It is well known that symmetric matrices $\mathbf{w}$ lead to fixed points or two-cycles and antisymmetric matrices may lead up to four-cycles only (Peretto, 1992). In general, however, a system of N neurons may generate large cycles, limited by the total number of states, 2^N. Gutfreund, Reger, and Young (1988) have shown that for $R_i = 0$, large cycles are generated when values of w_{ij} are chosen randomly from a gaussian distribution centered at zero and thresholds $\theta_i = \frac{1}{2}\sum_j w_{ij} \approx 0$.[3] This was further studied numerically by Nützel (1991), who pointed out that large cycles can be obtained for some range of the asymmetry $\alpha = \dfrac{\sum_{ij} w_{ij} w_{ji}}{\sum_{ij} w_{ij} w_{ij}}$ around the point 0 of random asymmetry.[4] The average size of the cycles grows as an exponential in N, with an exponent that increases as the asymmetry

[3] Related questions within the framework of Ising spin systems have been studied by Crisanti and Sompolinsky (1988).

[4] This definition of asymmetry differs from the parameter used by Gutfreund et al. (1988) and Nützel (1991), but it shares the relevant characteristics of varying between 1 and -1, with the extremes characterizing the symmetric and antisymmetric cases. The completely asymmetric case corresponds to $\alpha = 0$. The example of the five-dimensional matrix in section 3 has $\alpha = -0.4067$.

Table 3: Binary Odor Coding.

t/k	1	2	3	4	5	6
1	1 1	1 0	1 1	1 0	1 0	1 0
2	1 1	1 1	1 1	1 0	1 0	1 1
3	1 1	1 1	0 1	0 1	1 1	0 1
4	0 0	0 0	0 0	0 1	1 1	0 0

approaches zero. A system with a similar structure but slightly different dynamics was studied by McGuire, Littlewort, and Rafelski (1991).

Working with nonzero R values in our DNF, we noted that the interesting large cycles occur around the center of the relevant range of R. This is where the average effect of the inputs cancels the average effect of the synaptic weights, in agreement with the observations of Gutfreund et al. (1988). Thus, we expect, for any given N, that matrices $\mathbf{w}$ that have large capacities C_T for large T have a random-like distribution of synaptic values. We are, however, content even with cycles of order N, as is the case of our examples in section 3. Note that in the locust antennal lobe (see section 2.1), hundreds of neurons are participating in the real biological system. Hence, it is quite reasonable for it not to display cyclic behavior over the few LFP bins over which it is measured.

5.3 An Example: The Wehr-Laurent Experiment. We return now to the example that motivated our investigation. Wehr and Laurent (1996) display in Figure 3 of their article the response of two specific neurons to nine mixtures of odors. Significant results were obtained during the first four reverberations of the local field potential. They can be presented as six different binary codings, shown in Table 3, three of which appeared twice in the nine odor mixtures. Each of the six columns specifies the state of activity of the observed pair of neurons for one of the six spatiotemporal codes.

Looking at the first column, it is clear that it cannot be produced by a DNF, where states at time t are determined by states at $t-1$, unless it possesses at least two hidden neurons. Hence, we conclude that this should be treated as a $T = 4, K = 6$ problem. Looking for a system with $C_4 \approx 6$, we know we can do it with $N = 18$ (using condition $\mathcal{A}$), but may expect to be able to do it with $N = 6$ (using condition $\mathcal{B}$) or less. It turns out we can do it with $N = 4$. A set of states of four neurons, which accommodate the two observed neurons of Table 3, is given in Table 2. The latter was constructed by choosing states of hidden neurons by trial and error until all perceptron inequalities were satisfied, ensuring the existence of DNFs with matrices $\mathbf{w}$ and inputs R^k that can generate this table. Thus, we are able to implement the binary odor coding of Wehr and Laurent (1996) in an $N = 4$ model.

5.4 Minimal DNF as Measure of Complexity. The example of the Wehr-Laurent experiment raises the possibility of using the DNF model to define the complexity of a spatiotemporal data set. This degree of complexity can be characterized by the minimal number of neurons N needed to accommodate the data within a DNF. From the discussion in the previous paragraph we conclude that the Wehr-Laurent problem of Table 3 has DNF complexity of degree 4.[5]

6 Discussion

The problem of odor spatiotemporal encoding was recently reviewed in detail by Laurent et al. (2001). As a theoretical model, these authors suggest "winnerless competition," a concept that was expanded in Rabinovich et al. (2000, 2001). Basing their intuition on Lotka-Volterra equations they point out that for suitably chosen parameters, there exist heteroclinic orbits connecting all N attractors of the N-dimensional system, which are very sensitive to external inputs. They studied systems of FitzHugh-Nagumo neurons that exhibit spatiotemporal sensitivity to external inputs as a neural implementation, but made no attempt to fit a particular data set such as the one of Wehr and Laurent (1996).

In comparison, we use a binary model with discrete dynamics, but nevertheless we claim that it is relevant to the observed data. The reason is that the periodic LFP in the antennal lobe provides a temporal scale that accounts for specific time bins, and allows discrete coding, as observed by Wehr and Laurent (1996). With a DNF model that fits these data, one can envisage a model of spiking neurons capable of mimicking the experimental results (Quenet, Dubois, Sirapian, Dreyfus, & Horn, in press). This can be done by imposing overall periodic inhibition that allows for action potentials during periodic time windows and by fitting synaptic delay parameters to match the same windows.

Recently Friedrich & Laurent (2001) have observed spatiotemporal odor representations in the olfactory bulb of zebrafish. This system also has a reverberating local field potential and mitral cells that fire in coincidence with it. The authors have studied the response of this system to similar odors, characterized by small changes in molecular structures of the relevant chemicals. One of their interesting results is that the correlation between temporal patterns of similar odors over the mitral cells reduces with time (between the first and second 500 ms after odor presentation). This contradicts the expectation that short distances in R space lead to high correlations between the spatiotemporal sequences. The reason is the large value of N in

[5] Obviously this should be regarded as a mathematical statement and may have nothing to do with the biological reality. In the antennal lobe, one finds excitation of hundreds of PNs during each odor presentation.

this system. Since the volume of state-space is 2^N, once a change in a sequence occurs, it may lead to a new sequence uncorrelated with the old one.

Our model can be compared with recent work by Maass, Natschläger, and Markram (in press), who propose a recurrent neural network that performs what they call "liquid computation." Their network is composed of spiking neurons and is structured in two stages: (1) a filter transforming spatiotemporal input into spatiotemporal behavior of the network and (2) the activation of a readout map. Thus, this network projects spatiotemporal input into an output representation that is spatial in nature. Although these authors put special emphasis on real-time learning based on perturbations, the common feature of their approach and ours is the use of a filter projecting from an input to an output space. Whereas in our model the input space is spatial and the output space is spatiotemporal, their model goes in the opposite direction. However, in general, both models can connect spatiotemporal spaces to one another, and both use neural networks as filters, performing computations that are different from the conventional major paradigms. Our model is based on binary mathematical neurons, whereas Maass et al. (in press) deal with more realistic neural behavior. Nevertheless, the advantage of our DNF is that its structure can be made explicit, thus allowing full understanding of its operation and capabilities. Among other things, we can use this understanding to solve an inverse problem: construct a DNF that produces a given repertoire of spatiotemporal data.

Acknowledgments

We thank G. Dror for advice and assistance.

References

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Elect. Comp., 14*, 326–334.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Crisanti, A., & Sompolinsky, H. (1988). Dynamics of spin systems with randomly asymmetric bonds: Ising systems and Glauber dynamics. *Phys. Rev. A, 37*, 4865–4874.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley.

Friedrich, R. W., & Laurent, G. (2001). Dynamic optimization of odor representations by slow temporal patterning of mitral cell activity. *Science, 291*, 889–894.

Gutfreund, H., Reger, D. J., & Young, A. P. (1988). The nature of attractors in an asymmetric spin glass with deterministic dynamics. *J. Phys. A: Math. Gen., 21,* 2775–2797.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation.* Reading, MA: Addison-Wesley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA, 79,* 2554–2558.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cyb., 43,* 59–69.

Laurent, G., & Naraghi, M. (1994). Odorant-induced oscillations in the mushroom bodies of the locust. *J. Neurosciences, 14,* 2993–3004.

Laurent, G., Stopfer, M., Friedrich, R. W., Rabinovich, M. I., Volkovskii, A., & Abarbanel, H. D. I. (2001). Odor encoding as an active dynamical process: Experiments, computation and theory. *Ann. Rev. Neurosci., 24,* 263–297.

Maass, W., Natschläger, T., & Markram, H. (in press). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation.*

McGuire, P. C., Littlewort, G. C., & Rafelski, J. (1991). Brainwashing random asymmetric "neural" networks. *Physics Letters A, 160,* 255–260.

Nützel, K. (1991). The length of attractors in asymmetric random neural networks with deterministic dynamics. *J. Phys. A: Math. Gen., 24,* L151–L157.

Peretto, P. (1992). *An introduction to the modeling of neural networks.* Cambridge: Cambridge University Press.

Quenet, B., Cerny, V., Dreyfus, G., & Lutz, A. (1997). *A dynamic model of key feature extraction: The example of olfaction.* Unpublished manuscript, ESPCI.

Quenet, B., Dubois, R., Sirapian, S., Dreyfus, G., & Horn, D. (in press). Two step modeling of spatiotemporal olfactory data: From binary to Hodgkin-Huxley neurons. *Biosystems.*

Quenet, B., Horn, D., Dreyfus, G., & Dubois, R. (2001). Temporal coding in an olfactory oscillatory model. *Neurocomputing, 38–40,* 831–836.

Rabinovich, M. I., Huerta, R., Volkovskii, A., Abarbanel, H. D. I., Stopfer, M., & Laurent, G. (2000). Dynamical coding of sensory information with competitive networks. *J. Physiol. (Paris), 94,* 465–471.

Rabinovich, M. I., Volkovskii, A., Lecanda, P., Huerta, R., Abarbanel, H. D. I., & Laurent, G. (2001). Dynamical encoding by networks of competing neuron groups: Winnerless competition. *Phys. Rev. Lett., 87,* 068102.

Rosenblatt, F. (1962). *Principles of neurodynamics.* New York: Spartan.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representation by back-propagating errors. *Nature, 323,* 533–536.

Vapnik, V. (1995). *The nature of statistical learning theory.* New York: Springer-Verlag.

Wehr, M., & Laurent, G. (1996). Odor encoding by temporal sequences of firing in oscillating neural assemblies. *Nature, 384,* 162–166.

Received November 20, 2001; accepted July 30, 2002.