VC Dimension of an Integrate-and-Fire Neuron Model

Anthony M. Zador* **Barak A. Pearlmutter†**

To appear (1996) in *Neural Computation* *8*(3)

Abstract

We compute the VC dimension of a leaky integrate-and-fire neuron model. The VC dimension quantifies the ability of a function class to partition an input pattern space, and can be considered a measure of computational capacity. In this case, the function class is the class of integrate-and-fire models generated by varying the integration time constant τ and the threshold θ, the input space they partition is the space of continuous-time signals, and the binary partition is specified by whether or not the model reaches threshold at some specified time. We show that the VC dimension diverges only logarithmically with the input signal bandwidth N. We also extend this approach to arbitrary passive dendritic trees. The main contributions of this work are (1) it offers a novel treatment of the computational capacity of this class of dynamic system; and (2) it provides a framework for analyzing the computational capabilities of the dynamical systems defined by networks of spiking neurons.

1 Introduction

A central concern in computational neuroscience is understanding the functional significance of single neuron complexity. On the one hand, the success of artificial neural network models, which begin with the notion that brain-like computation can be well described by large interconnected networks of very simple elements, argues that the computational capabilities of the individual elements can be neglected. On the other hand, a vast body of research (see *e.g.* McKenna *et al.* (1992)) supports the notion that single neurons are complex dynamical systems, able to perform a wide range of interesting computations. Brown *et al.* (1992) have argued for a synthesis of these positions: if individual neurons have computational significance, then perhaps each should be considered a *micronet* in its own right.

In order to assess the computational signficance of single neurons, it would be useful to have a quantitative measure of computational capacity. The *Vapnik-Chervonenkis*

*Salk Institute, 10010 N. Torrey Pines Rd., La Jolla, CA 92037, zador@salk.edu

†Siemens Corp. Res., 755 College Rd. E., Princeton, NJ 08540, barak.pearlmutter@alumni.cs.cmu.edu

dimension (1971) can be considered such a measure for static neural networks (or, more generally, for any boolean function class.) It is a measure of the richness of the mappings possible within a class of functions, and typically increases as the size of the network (*i.e.* number of free parameters) increases. Such measures have not been applied to models of real neurons, in part because real neurons are dynamical systems.

There is as yet no satisfactory general theory of computation in dynamical systems. As a step in that direction, we have extended the notion of the VC dimension to dynamical systems. We consider the class of noiseless leaky integrate-and-fire threshold models with time constant τ and threshold θ driven by continuous-time inputs; we then extend our analysis to noisy inputs. These models have been developed as simplified descriptions of the more complex dynamics of real neurons. We define the VC dimension in terms of the ability of this class to assign an arbitrary boolean "label" to each input signal; the largest number of signals to which every possible labeling can be assigned is its VC dimension. We show that the VC dimension diverges logarithmically with the input signal bandwidth N.

2 Review of VC dimension

The VC dimension Vapnik and Chervonenkis (1971) is a measure of the richness of a class of boolean functions. It gives an upper bound on the number of exemplars required to guarantee that a set of parameters fit to data will provide a good fit for new data (Blumer *et al.* 1989). It has been applied in the neural network literature to give a measure of the number of patterns needed to train a network of a given size. Here we present a brief overview of the VC dimension in the context of neural networks (see Abu-Mostafa (1989) for an introduction.)

Let $F \subset \Re^N \rightarrow \{0, 1\}$, and $f_w \in F$ be some member of that class. For example, F could be the class of all 3-layer feedforward threshold networks with N inputs, 12 hidden units, and one output, parameterized by $c = 12N + 25$ weights, and f_w would be some particular choice of the c-dimensional weight vector w. For every set $\mathbf{I} = (I^{(1)}, \ldots, I^{(M)})$ of inputs, any choice of $w = w'$ specifies an M-digit binary string $Y_{w'} = f_{w'}(\mathbf{I}) = (f_{w'}(I^{(1)}), \ldots, f_{w'}(I^{(M)}))$ in which the m^{th} digit corresponds to the output of f_w on the m^{th} input $I^{(m)}$; varying w will in general produce a different binary string. $Y_{w'}$, which is actually a function of the inputs, $Y_{w'}(I^{(1)}, \ldots, I^{(M)})$, can be thought of as the truth table for a particular choice of $w = w'$ on the inputs. Now in principle Y_w can take on 2^M possible values; but for large M there may not be choices of w that instantiate every possible binary number. When there exist 2^M values of w such that Y_w takes on all the 2^M possible values, the function class F is said to *shatter* the set of inputs $\mathbf{I}$. This leads to the VC dimension d_{VC} of F: the VC dimension is the largest number M for which there exists a set of inputs $(I^{(1)}, \ldots, I^{(M)})$ which is shattered by F.

In the context of learning theory the VC dimension is useful because of a relation between the number of labeled exemplars in a training set and the probability of generating the correct output on a new exemplar (Vapnik and Chervonenkis 1971). If the number of exemplars is greater than the VC dimension, then the probability of producing an incorrect

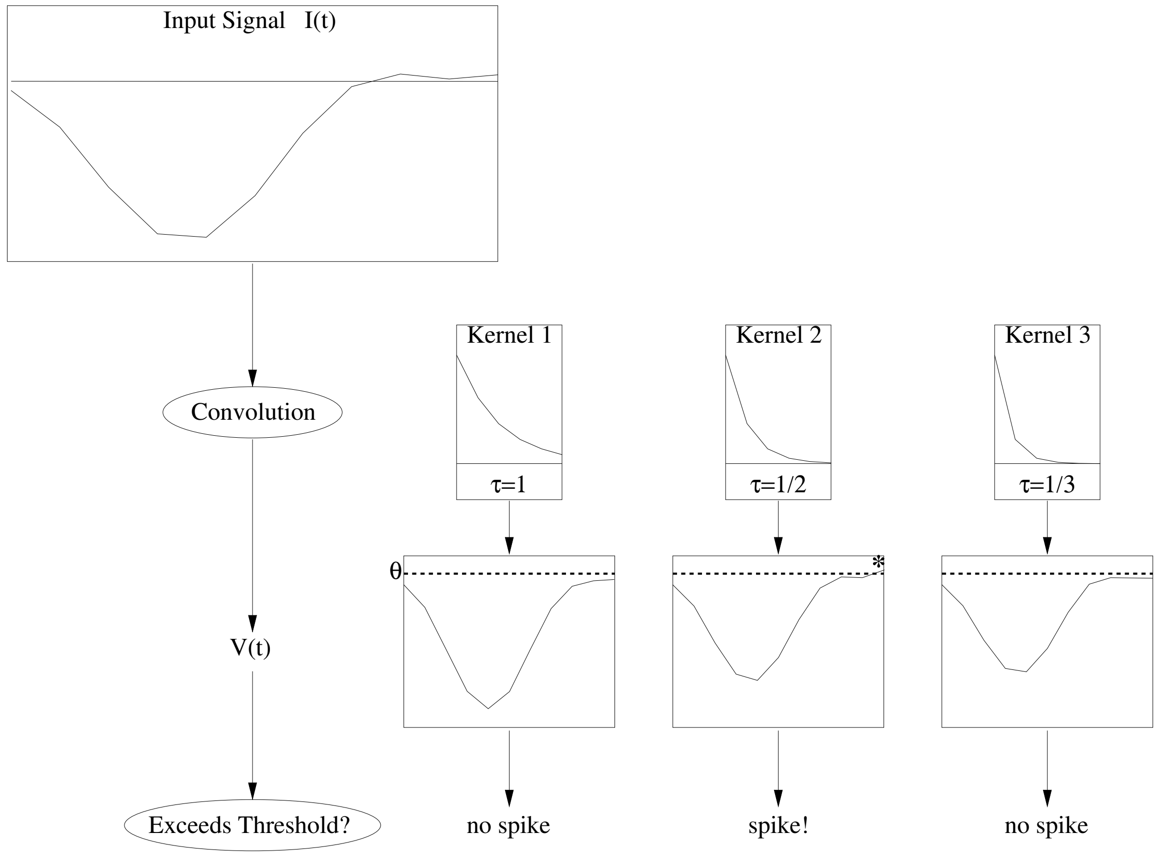

Figure 1: The model. An input signal $I(t)$ is convolved with a kernel and passed through a threshold to produce a binary output. The output of three distinct kernels, differing only by the time constant τ, to the same input is shown. The input has been constructed so that for low ($\tau = 1/3$) and high ($\tau = 1$) values, $V(t)$ remains below the threshold θ. For an intermediate value ($\tau = 1/2$) $V(t)$ exceeds θ at the $*$ and emits a spike. Note that the fluctations around the threshold are very small, indicating a high sensitivity of the system to noise.

response decreases exponentially with the number of exemplars. Much work has gone into computing the VC dimension of certain classes of neural networks (Baum and Haussler 1989; Anthony 1994; Maass 1995).

3 An Integrate-and-fire Classifier

The classifier we consider is a leaky integrate-and-fire unit with two free parameters: a single time constant τ and a threshold θ, as shown in figure 1. The inputs are continuous time signals, and the output is a binary variable determined by whether the voltage exceeds the threshold at any time t.

The voltage $V(t)$ of the unit at time t is given by the convolution of the input $I(t)$ with a single exponential kernel $w(t) = e^{-t/\tau}$,

$$V(t) = \int_0^t I(t-\xi)\, e^{-\xi/\tau}\, d\xi. \tag{1}$$

The convolution kernel has only a single exponential; this corresponds to the output of a single RC integrator.

We now define $\tilde{V}$ as the voltage at the end of the interval $[0, t_f]$,

$$\tilde{V} = V(t_f).$$

The output Y of the unit over this interval is a binary variable, obtained by applying a threshold θ to $\tilde{V}$,

$$Y = \mathrm{sgn}(\tilde{V} - \theta). \tag{2}$$

Notice that the voltage $V(t)$, $t < t_f$ does not involve thresholding; the threshold is imposed only at $t = t_f$, so the present model is an integrate-and-fire model without reset. Only when $V(t)$ remains subthreshold over the interval does it give the the same output as standard integrate-and-fire models, which reset after each threshold-crossing. If we would like our results to carry over to resetting models, we must be careful to consider inputs that do not cause $V(t)$ to exceed threshold prematurely.

4 Convolution as a product

We now consider the temporal discretization into evenly spaced intervals, $t_0, \ldots, t_{N-1}$,

$$\tilde{V} = \sum_{i=0}^{N-1} I_i w_i, \tag{3}$$

where $I_i = I(t_{N-1} - t_i)$ and N is the signal bandwidth, with

$$w_i = e^{-t_i/\tau}. \tag{4}$$

This is simply the discrete convolution of the input I_i with a kernel w_i. Note that this equation can be interpreted as a one-output perceptron with an N-dimensional input vector $\mathbf{I}$ and a weight vector w. We observe that due to the physical constraints of positive time constants $\tau > 0$, and $t_i \geq 0$, we find a constraint on w_i,

$$0 \leq w_i \leq 1. \tag{5}$$

In eq. (3), we used the conventional represention of the discrete convolution as a sum. In assessing the VC dimension it will be convenient to work with an equivalent representation as a product. We observe that the convolution eq. (3) is polynomial in w_1, since

$$w_i = e^{-i\Delta t/\tau} = (w_1)^i, \tag{6}$$

where $\Delta t = t_{i+1} - t_i$. We can therefore write

$$\tilde{V} = \sum_{i=0}^{N} I_i (w_1)^i = I_N \prod_{i=0}^{N-1} (w_1 - r_i) \tag{7}$$

where r_i are the $N = t_f/\Delta t + 1$ roots of the polynomial.

Eq. (7) expresses the output $\tilde{V}$ of the integrate-and-fire unit as a polynomial of degree N in the weight kernel w_i, specified by the parameter w_1. The output is a function of τ, since the weight kernel is related to τ by eq. (6). The coefficients w_i of the sum, and therefore the locations r_i of the roots, are determined by the inputs. Different integrate-and-fire units may assign different outputs to a given input as the parameter w_1 is varied. The advantage of the product representation is that it allows us to see explicitly the critical values of w_1 at which the output in response to a given input changes. Specifically, the critical values are the roots r_i of the polynomial. Since the roots are determined by the input signal itself, the critical values of w_1 depend on the input itself, and will in general be different for different inputs.

4.1 Constructing a shatterable set of inputs

The key construction of this section (illustrated in figure 2) is a procedure for "inverting" the integrate-and-fire neuron, by constructing an input signal $I(t)$ given a list of w_1 values and corresponding responses (spike *vs.* no spike.)[1] For now we consider only the zero-threshold case.

Before proceeding, let us specify the elements of the construction. We will form a set of input vectors $\mathbf{I}^{(1)}, \ldots, \mathbf{I}^{(M)}$. Each N-dimensional input vector is obtained by sampling a continuous waveform $I(t)$ at N uniformly spaced points. For any given value of w_1, eqs. (2) and (3) determine the binary value of the output $Y^{(m)}$ in response to input $\mathbf{I}^{(m)}$. Thus each value of w_1 specifies an M-digit binary number, in which the m^{th} digit is the output $Y^{(m)}$ in response to input $\mathbf{I}^{(m)}$. We call $Y^{(m)}$ the *label* associated to the input $\mathbf{I}^{(m)}$ by a given value of w_1, and Y is the M-digit label associated by a given value of w_1 to the set of M inputs. There are 2^M possible such labels associated with any set of M inputs. Recall that if a set of 2^M values of w_1 can be specified, such that this set associates all possible labels Y to the input set, then this set is said to shatter the inputs. The VC dimension is the largest value of M for which a shattering set of w_1s can be found.

Our task is therefore to construct set of M inputs and specify a corresponding set of 2^M values of w_1 such that the input set is shattered. We begin by considering how the labeling of a given input varies with w_1. That is, what is the m^{th} digit, considered as a function

[1] If a finer temporal discretization is desired, adding extraneous roots gives the input waveform more sample points without introducing extra sign changes. If explicitly continuous-time inputs are to be constructed, one can consider the Laplace transform of the input, and note that the desired outputs correspond to simple sign constraints in the Laplace domain. A function in the Laplace domain that meets the constraints can be concocted, and an inverse Laplace transform gives the corresponding time-domain input. There is a great deal of freedom in this concocting, but one natural class of inputs resulting from the inverse Laplace transform is a series of modulated delta pulses. It is interesting to note that the inputs neurons typically receive consist of a series of action potentials.

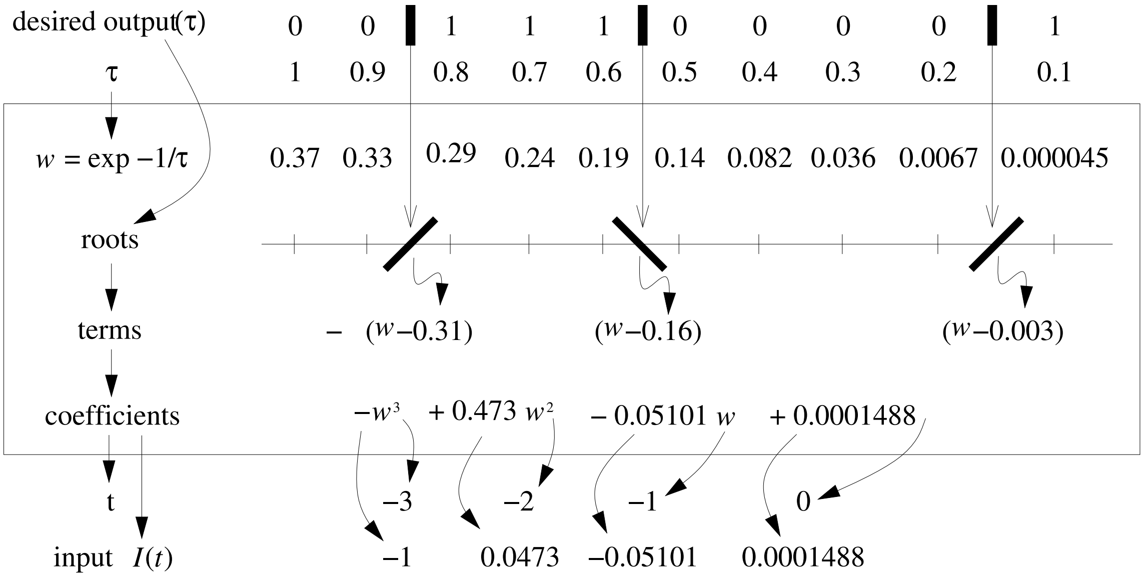

Figure 2: Diagrammatic representation of the construction of an input which results in output spikes at particular desired neuronal time constants τ_i. Given a set of time constants τ_i and associated binary desired outputs, a single temporal input is constructed which has the property that, when the neuron's time constant is set to τ_i, the associated desired output is produced. The construction proceeds in stages: the time constants are passed through a function, the transitions in the desired outputs are marked and arbitrary points in the corresponding intervals are chosen, a polynomial with these points as roots is constructed, and the coefficients of this polynomial form the desired temporal input. To construct a set of 2^M shatterable inputs, this construction is used 2^M times.

of w_1, of the label Y associated with the input $\mathbf{I}^{(m)}$? Using the product representation of the convolution from the previous section, we observe that the label changes whenever w_1 passes through a root. Within the interval between two roots, $r_i < w_1 < r_{i+1}$, the label remains unchanged. We can therefore conveniently manipulate the labeling associated with a given input by judicious placement of the roots. In fact, once the roots are specified, the input is obtained simply by multiplying out the product in eq. (7) to obtain the coefficients I_i.

Now we turn to the M-digit label Y associated with a specified value of w_1. For this we hold w_1 fixed, and consider the label associated with each input $\mathbf{I}^{(1)}, \ldots, \mathbf{I}^{(M)}$ in turn; these are the digits of Y. But since we have already shown how to obtain the desired label for any particular input—by placing the roots appropriately—obtaining the desired label Y for a given w_1 merely requires choosing the roots associated with each input in turn. Thus we have a procedure for constructing an input set that associates a specified label Y with the input set for a particular value of w_1.

4.2 VC dimension depends on signal bandwidth

So far we have shown how to construct an input set labeled by a specified Y for a given value of w_1. The final step requires constructing an input set that is shatterable—a set for which Y assumes all 2^M possible values, at 2^M values of w_1, $0 < w_1 < 1$. That is, we must partition the w_1-axis into 2^M regions. The boundaries between the regions are determined by the roots: the presence of a root at some w_1 for the m^{th} input means that the m^{th} digit of Y changes at that value of w_1. The number of digit changes is NM: N roots/input times M inputs.

Since Y is an M-digit binary string that we require to assume all 2^M possible values, we can regard this as counting in binary. Now counting from 0 to $2^M - 1$ in standard binary involves $2^M \log M$ digit changes. For example, the transition from $0111_2 = 7$ to $1000_2 = 8$ involves 4 digit changes. In order to make best use of the NM roots, we therefore adopt a different counting scheme, a Gray's code,[2] so that only NM digit changes are required. Figure 3 shows how to construct a shatterable set of $M = 2$ inputs using this scheme. Here the requisite bandwidth is $N = 2$. The roots of the first input $\mathbf{I}^{(0)}$ are placed at $r^{(0)} = (1/8, 3/8, 6/8)$. Expanding as in eq. 7 gives the actual sampled values of $\mathbf{I}^{(0)}$.

The number of roots of each polynomial is determined by the temporal discretization N. For a set of M bandlimited signals, there are at most NM distinct roots, which can be used to divide the w-axis into $NM + 1$ regions

$$\text{number of labels} = NM + 1.$$

Thus the VC dimension is determined by the sampling rate. To achieve $d_{VC} = M$, we choose a sufficiently large N given by

$$N = \left\lceil \frac{2^M - 1}{M} \right\rceil, \tag{8}$$

where $\lceil \cdot \rceil$ indicates rounding up to the largest integer. This shows that with a sufficiently high sampling rate an arbitrarily high VC dimension can be achieved. Since N is determined by the sampling rate of a continuous signal, the VC dimension of a signal of infinite bandwidth is unbounded. It is important to note, however, that the dependence of the VC dimension on the signal bandwith is only logarithmic, and therefore the divergence is weak.

4.3 Threshold: Preventing Premature Discharge

The model we have been considering (eqs. 1 and 2) has no reset; $\tilde{V}$ does not depend on whether $V(t)$ exceeds threshold at any time within the interval $0 \le t \le t_f$. This of course is not the expected behavior from an integrate-and-fire model. Typically, integrate-and-fire models reset $V(t) \to 0$ after discharging (sometimes imposing a refractory period as well.)

The inputs we have constructed will not typically be shattered by an integrate-and-fire model with reset. However, by using a non-zero threshold, we can construct a new

[2] A Gray's code is an ordering of the binary numbers $0, \ldots, 2^M - 1$ such that adjacent numbers differ in only one digit. For our purposes, we choose a Gray's code in which all digits changes state the same number of times, namely $2^M/M$ times.

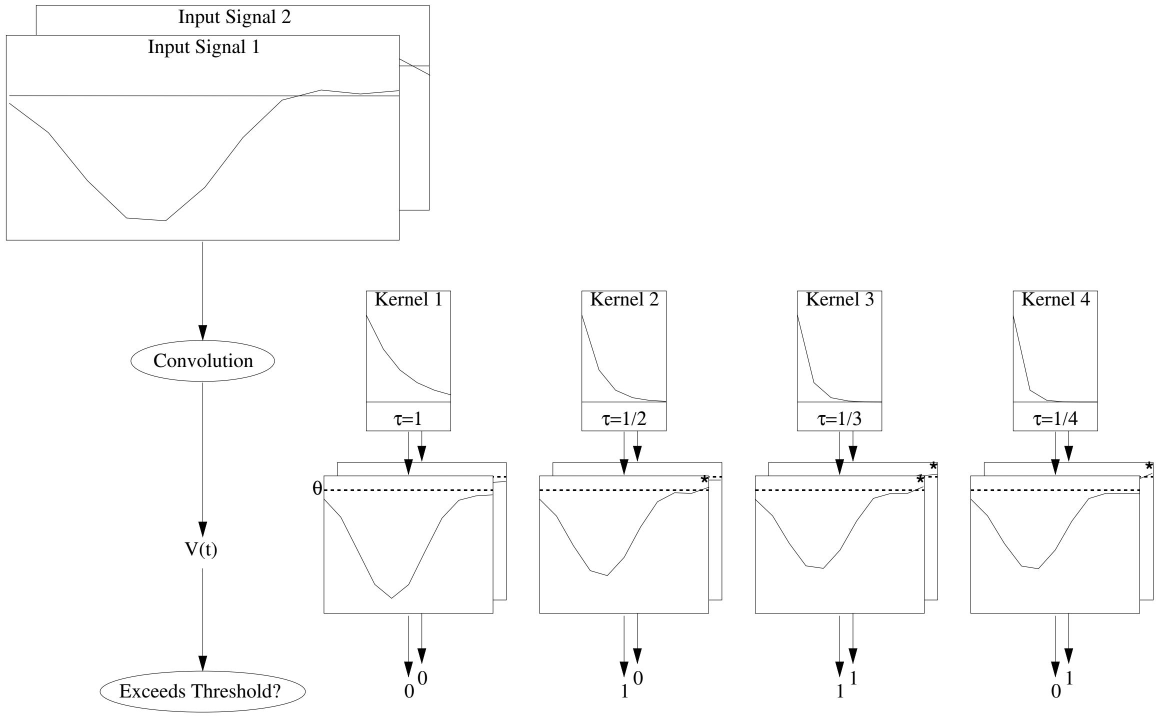

Figure 3: A set of shattered inputs. $M = 2$ input signals are constructed such that there exist neuronal time constants $\tau_1, \tau_2, \tau_3, \tau_4$ that induce all $2^M = 4$ possible labelings.

set of inputs that is shattered. First we set the threshold to exceed the maximum over the interval, $\theta > \max_{t<t_f} V(t)$. We now add the threshold to the final term of each input signal (corresponding to the constant term of the associated polynomial) to create a new set of inputs I'. These new inputs differ only at I_0,

$$I'_0 = I_0 + \theta.$$

(Note that because of the definition of $\mathbf{I}$ in eq. (3), I_0 corresponds to $I(t_f)$, *i.e.* it is the last point of the sampled waveform.) Since $w_0 = 1$, this shift guarantees that digit changes, which previously occured when $\tilde{V}$ crossed zero for different values of w_1, now occur when $\tilde{V}$ crosses θ.

5 Special cases and extensions

5.1 VC dimension for purely positive inputs

The VC dimension of a system with purely positive inputs is 1. This is of interest when considering inputs generated by purely excitatory synaptic inputs. To show this, we note first that by construction, the shattering inputs oscillate around 0. That is, for each input, subsequent points I_i and I_{i+1} have opposite sign. This follows from eq. 7: the n^{th} order coefficient is generated by the sum of the products of $N - n$ negative terms (since $r_i > 0$), which is positive if $N - n$ is even and negative otherwise. Conversely, if I is purely positive, then the roots are all negative and imaginary. They are therefore physically unrealizable under our assumptions (eq. 5.) Thus the VC dimension is 1. Adding a threshold creates only at most one new root.

5.2 Passive dendritic trees

In the integrate-and-fire model we have been considering, the integrating kernel consists of a single exponential time constant, corresponding to a single RC circuit. One generalization of this model is to passive dendritic trees. Using the classic result that the convolution kernel (*i.e.* the Green's function) can be approximated as the sum of z exponentials,

$$W(t) = c_1 e^{-t/\tau_1} + \cdots + c_z e^{-t/\tau_z}.$$

Then the voltage can be represented by

$$V(t) = \int_0^t I(t - \xi)\, W(\xi)\, d\xi.$$

Discretizing as before, we have

$$\tilde{V} = \sum_{i=0}^{N-1} I_i W_i,$$

where

$$W_i = c_1(w_1)^i + \cdots + c_z(w_z)^i.$$

The effect of the dendritic tree is therefore to increase the number of roots for a given bandwidth from NM to NzM, since now for each of the M inputs there are now zN rather than N roots. The requisite bandwidth N to shatter M inputs is now

$$N = \left\lceil \frac{2^M - 1}{zM} \right\rceil, \tag{9}$$

where as before $\lceil\ \rceil$ indicates rounding up to the largest integer. This is less by a factor of z than in the case of a single exponential.

5.3 The effect of input noise

Finding that a concept class has unbounded VC dimension should be taken as a sign that issues of noise, precision, and physical realizability are the only bounds on PAC generalization. For instance, convex polygons in the plane have unbounded VC dimension. This is in contrast to a finite VC dimension, which means that even with unlimited precision and zero noise, there is a PAC generalization bound.

Here we consider the effect of noise added to the inputs. In general, noise in a system with signal power constraints determines a maximum resolvable frequency, which in the present context determines N, the signal bandwidth. The VC dimension depends only logarithmically on N, so although eq. 8 is formally a divergence of the VC dimension, actually this divergence is only logarithmic, and therefore weak. In practice, for any physically realizable system, the VC dimension given by eq. 8 will be quite small.

Another way to think about this is to suppose that

$$\mathbf{I}' = \mathbf{I} + \mathbf{n} \tag{10}$$

where $\mathbf{n}$ is gaussian white noise. From eq. (7), we have

$$\begin{aligned}
\tilde{V}' &= \sum_{i=0}^{N-1} I_i'(w_1)^i & (11)\\
&= \sum_{i=0}^{N-1} n_i(w_1)^i + \sum_{i=0}^{N-1} I_i(w_1)^i & (12)\\
&= z + \tilde{V} & (13)
\end{aligned}$$

where n_i and I_i are the i^{th} components of the input and noise, respectively. The first term (the random variable z) is the weighted sum of N iid gaussian variables, so it is also a gaussian, with variance σ_z^2; the second term is just $\tilde{V}$ in the noise-free case.

So how does this effect the VC dimension? Noise in this sense does not fall into the classical VC framework (but see Bartlett, Long, and Williamson (1994).) Nevertheless, the effect is clear: there is some probability P_m of misclassifying each input. This probability depends on $\tilde{V}$ and on z: it is the probability that $\mathrm{sgn}(\tilde{V} + z) \neq \mathrm{sgn}(\tilde{V})$. If both z and $\tilde{V}$ are 0-mean, then this is just $\mathrm{Prob}(|\tilde{V}| - |z| < 0)/2$ (we divide by 2 because half the time z has the same sign as $\tilde{V}$ and so doesn't change its sign.)

What is the misclassification error associated with z? This depends on the ratio $\tilde{V}/z$, which looks like a kind of signal to noise ratio. However, the natural measure of the signal strength is $\sum I_i^2$, and it is this quantity that should participate in the signal to noise ratio. Because of the manner in which they are constructed, for typical signals I_i is largest around $N/2$. If the roots are uniformly distributed between 0 and 1, we can actually estimate the typical signal strength as a function of N just by multiplying out the polynomial.

Numerical simulations suggest that very large signal to noise ratios are required to keep the error reasonable for even moderate values of N, even larger than those called for by the bandwidth requirements, which after all constitutes only an upper bound.

6 Discussion

This is to our knowledge the first application of the VC dimension to a dynamical system. We consider the thresholded output of an integrate-and-fire model to impose a binary partition on a set of continuous-time input signals. We have shown that the VC dimension of this model diverges logarithmically with the input signal bandwidth N.

Because our analysis is stronger than the usual VC dimension calculation, the consequences for generalization are slightly more robust to prior knowlege than the generic

PAC bound.[3]

6.1 Implications for single neuron computation

There is an extensive literature demonstrating the computational potential of single neurons and networks. Koch *et al.* (1982) showed how an AND-NOT of two inputs could be performed in the passive dendritic tree of a retinal ganglion cell, and suggested that this might play a role in the computation of directional selectivity. Shepherd and Brayton (1987) implemented a complete set of logic operations at single spines using Hodgkin-Huxley channels and inhibitory inputs for NOT. Zador *et al.* (1992) showed how calcium- and voltage-dependent channels could implement a kind of temporal XOR in the dendritic tree, without additional inhibitory inputs. Maass (1996) has shown that networks of simple spiking neurons possess rich computational properties, in the sense of complexity theory.

None of these demonstrations attempted a quantification of the overall computational capacity of a single neuron. To our knowledge, the only attempt to quantify the ability of a single neuron to partition an input space is Mel (1992). He implemented a model of a cortical neuron with nonlinear NMDA conductances in the dendritic tree, and with a biologically-motivated Hebb rule trained it to partition 100 high-dimensional patterns into two classes. The error rate on this set using various measures was about 10%. Note that the model class we consider—purely passive dendritic trees with integrate-and-fire nonlinearities—is more restricted than the NMDA-based nonlinearities considered by Mel (1992).

We have described a more formal approach to the analysis of single neuron computation. This approach takes into account the temporal structure of inputs. It puts a bound on the ability of a simple model to partition an input space. Because of the exponential dependence of the signal bandwidth required to achieve a given VC dimension (eq. 8), under reasonable physical assumptions the VC dimension must be quite small. The exquisite sensitivity to noise in the inputs further limits the number of inputs that could be shattered by any physically realizable system. This number can be considered to be less than 10.

The model we consider is of course a caricature of a real neuron—the dynamics of real neurons are much more complex (see *e.g.* McKenna *et al.* (1992).) The leaky integrate-and-fire model with reset is nevertheless a standard starting point for considering dynamical aspects of neuron behavior. A recent careful examination of its validity (Koch, Bernander, and Douglas 1995) supports the notion that for rapidly varying input signals of the kind considered here it offers a good first approximation.

[3] In order to show that the VC dimension of a concept class is at least M, one must show that there exists *some* set of 2^M concepts which shatters some set of M inputs. Here we have shown something more general, since our construction proceeds for *any* set of 2^M concepts from our concept class. Given any set of 2^M concepts, we can find a set of M inputs that these concepts shatter. This has consequences in the application to PAC learning (Valiant 1984), where it corresponds to generalizing one of the two worst-case assumptions of the PAC criterion. So the PAC lower bound on generalization here requires the usual worst-case assumption over distributions of inputs, but remains true for any reasonable distribution over concepts.

6.2 The VC dimension and dynamical systems

A useful rule of thumb is that the VC dimension often turns out to be roughly proportional to the number of free parameters. This is true, for example, in feedforward linear threshold networkss, where the VC dimension is equal to the number of free weights, up to a logarithmic factor (Baum and Haussler 1989). In our case, we expected the VC dimension to be about two, since there were two free parameters (θ and τ.) Furthermore, a small VC dimension for the integrate-and-fire model conforms to our intuitive notion of the simplicity of this model. In fact, eq. (3) shows that the integrate-and-fire model can be considered as a kind of perceptron, and thus can only impose linear partitions on the input space.

We were therefore surprised to find that for noiseless inputs, the VC dimension was unbounded. However, the apparent power of the integrate-and-fire unit arises not from a nonlinear partitioning, but rather from a linear partitioning in a space of unbounded dimension.[4] A similar dilemma arises when the discrete formulation of Shannon entropy is applied to continous variables: the information content of a noiseless random variable is infinite (since, for example, any message can be encoded in its decimal expansion.) Any finite noise, of course, renders finite the discrete entropy of the continuous variable.

Just as the discrete entropy of a continuous variable becomes finite in the presense of noise, so the unbounded VC dimension collapses when any notion of noise is included. We considered two ways noise could limit the VC dimension. First, the bandwidth of the signal is implicitly due to noise, and the VC dimension diverges logarithmically with the input signal bandwidth. Second, we considered the effect of noise added explicitly to the signal, and found that the probability of misclassification was a very steep function of the VC dimension. In both cases, the apparent VC dimension in the presense of noise conformed much more closely with our intuitive notion that it should be rather small.

It will be interesting to see whether related notions of computational capacity, such as those derived from work on average generalization (Haussler, Kearns, Seung, and Tishby 1994), can be extended to dynamical systems in a similar way.

Acknowledgments

We are grateful to the reviewers for their constructive and thought-provoking comments, and Paul Zador for many useful discussions. We also thank Christof Koch for support in this work, which was initiated while the authors were in Tom Brown's lab at Yale University. This research was funded by grants from the ONR and the NIMH Center for Neuroscience to Christof Koch.

References

Abu-Mostafa, Y. S. (1989). The Vapnik-Chervonenkis Dimension: Information versus Complexity in Learning. *Neural Computation*, *1*(3), 312–317.

[4] We speculate that local dendritic processing gives rise to a real increase in computational power—one that arises from a nonlinear partitioning of a space of fixed dimension.

Anthony, M. (1994). Probabilistic Analysis of Learning in Artificial Neural Networks: The PAC Model and its Variants. Mathematics preprint series LSE-MPS-67, Dept. of Mathematics, London School of Economics, Houghton St., London WC2A 2AE, UK. Also available as NeuroCOLT technical report NC-TR-94-3, ftp://cscx.cs.rhbnc.ac.uk/pub/neurocolt/tech_reports/.

Bartlett, P. L., Long, P. M., and Williamson, R. C. (1994). FAT-Shattering and the Learnability of Real-Valued Functions.. In COLT (1994), pp. 299–310.

Baum, E. and Haussler, D. (1989). What Size Net Gives Valid Generalization?. *Neural Computation*, *1*(1), 151–160.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM*, *36*, 929–965.

Brown, T. H., Zador, A. M., Mainen, Z. F., and Claiborne, B. J. (1992). Hebbian Computations in Hippocampal Dendrites and Spines. In McKenna *et al.*(1992), pp. 81–116.

COLT (1994). *Seventh Annual ACM Workshop on Computational Learning Theory*, New Brunswick, NJ.

Haussler, D., Kearns, M., Seung, H. S., and Tishby, N. (1994). Rigorous Learning Curve Bounds from Statistical Mechanics.. In COLT (1994), pp. 67–75.

Koch, C., Bernander, Ö., and Douglas, R. J. (1995). Do neurons have a voltage or a current threshold for action potential initiation?. *Journal of Computational Neuroscience*, *2*, 63–82.

Koch, C., Poggio, T., and Torre, V. (1982). Retinal Ganglion Cells: a Functional Interpretation of Dendritic Morphology. *Proc. of the Royal Soc. of London B*, *298*, 227–264.

Maass, W. (1995). Vapnik-Chervonenkis Dimension of Neural Networks. In Arbib, M. A. (Ed.), *Handbook of Brain Theory and Neural Networks*, pp. 1000–1002. MIT Press.

Maass, W. (1996). Lower bounds for the computational power of networks of spiking neurons. *Neural Computation*, *8*(1). In press.

McKenna, T., Davis, J., and Zornetzer, S. F. (Eds.). (1992). *Single Neuron Computation*. Academic Press.

Mel, B. W. (1992). NMDA-Based Pattern Discrimination in a Modeled Cortical Neuron. *Neural Computation*, *4*(4), 502–516.

Shepherd, G. and Brayton, R. (1987). Logic operations are properties of computer-simulated interactions between excitable dendritic spines. *Journal of Neuroscience*, *21*, 151–166.

Valiant, L. G. (1984). A Theory of the Learnable. *Communications of the ACM*, *27*(11), 1134–1142.

Vapnik, V. and Chervonenkis, A. (1971). On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and Its Applications*, *16*, 264–280.

Zador, A. M., Claiborne, B. J., and Brown, T. H. (1992). Nonlinear Pattern Separation in Single Hippocampal Neurons with Active Dendritic Membrane. In Moody, J. E., Hanson, S. J., and Lippmann, R. P. (Eds.), *Advances in Neural Information Processing Systems 4*, pp. 51–58. Morgan Kaufmann.