INSTITUTE OF PHYSICS PUBLISHING NETWORK: COMPUTATION IN NEURAL SYSTEMS

Network: Comput. Neural Syst. **12** (2001) 75–87 www.iop.org/Journals/ne PII: S0954-898X(01)20361-3

Efficient temporal processing with biologically realistic dynamic synapses

Thomas Natschläger[1,3], **Wolfgang Maass**[1] **and Anthony Zador**[2]

[1] Institute for Theoretical Computer Science, Technische Universität Graz, Klosterwiesgasse 32/II, A-8010 Graz, Austria
[2] Cold Spring Harbor Laboratory, 1 Bungtown Road, Cold Spring Harbor, NY 11724, USA

E-mail: tnatschl@igi.tu-graz.ac.at, maass@igi.tu-graz.ac.at and zador@cshl.org

Received 15 June 2000

Abstract

Synapses play a central role in neural computation: the strengths of synaptic connections determine the function of a neural circuit. In conventional models of computation, synaptic strength is assumed to be a static quantity that changes only on the slow timescale of learning. In biological systems, however, synaptic strength undergoes dynamic modulation on rapid timescales through mechanisms such as short term facilitation and depression. Here we describe a general model of computation that exploits dynamic synapses, and use a backpropagation-like algorithm to adjust the synaptic parameters. We show that such gradient descent suffices to approximate a given quadratic filter by a rather small neural system with dynamic synapses. We also compare our network model to artificial neural networks designed for time series processing. Our numerical results are complemented by theoretical analyses which show that even with just a single hidden layer such networks can approximate a surprisingly large class of nonlinear filters: all filters that can be characterized by Volterra series. This result is robust with regard to various changes in the model for synaptic dynamics.

1. Introduction

The brain is able to solve hard computational problems that remain beyond the reach of the most powerful computers, but the key to its success remains unclear. One possibility is that the properties of neuronal wetware—as opposed, for example, to the digital hardware found in a computer—enforce a style of computation that is particularly well suited to solving the kinds of problems important to survival. If this is true, then we may gain insight into the strategies employed by neuronal wetware by studying computational models that capture the essence of neural circuitry. This strategy has motivated the development of artificial neural network models of computation. Like brains, neural networks are massively parallel networks

[3] To whom correspondence should be addressed.

0954-898X/01/010075+13$30.00

composed of many simple repeating units. Neural networks share a number of characteristics with brains, including fault tolerance, generalization, and the ability to learn (or adapt) to new inputs. Neural network models have been useful for understanding what kinds of algorithms are well suited for brain-style computation.

Neural networks have been widely applied to the processing of static stimuli. In recent years, however, there has been increasing focus on the dynamic aspects of cortical processing. Spatiotemporal (Reid *et al* 1997) and spectrotemporal (Kowalski *et al* 1996, deCharms and Merzenich 1998) receptive field analysis, for example, reveal that the cortical neurons are sensitive to the temporal structure of sensory inputs. Processing of real-world time-varying stimuli is a difficult problem, and represents an unsolved challenge for artificial models of brain function.

More than two decades of research on artificial neural networks has emphasized the central role of synapses in neural computation (Sejnowski 1977, Hopfield 1982). In a conventional artificial neural network, all units ('neurons') are assumed to be identical, so that the computation is completely specified by the synaptic 'weights', i.e. by the strengths of the connections between the units. The identity of a neural circuit—including the circuit connectivity, which can be specified by including null weights—is thereby determined entirely by the matrix of synaptic connections. The synapses in most artificial neural network models are static: synaptic strength is fully characterized by a single value that remains fixed except on the slow timescale of learning. In real nervous systems, by contrast, synapses show dynamics on short timescales, from milliseconds to seconds (Magleby 1987, Markram and Tsodyks 1996, Abbott *et al* 1997, Zador and Dobrunz 1997, Dobrunz and Stevens 1999). Activity-dependent forms of short-term plasticity such as facilitation and depression modulate synaptic strength over a wide range.

Here we propose that synaptic dynamics provide a natural substrate for the processing of dynamic stimuli, and describe a novel artificial neural network architecture that exploits synaptic dynamics (Little and Shaw 1975, Tsodyks *et al* 1998, Liaw and Berger 1996). As in conventional artificial neural networks, synaptic strength determines the computation. In our framework, however, synaptic strength changes on the short time scale of each computation, and it is the balance of facilitation and depression that determines the temporal dynamics at each synapse and thereby forms the basis of each computation. To achieve the appropriate synaptic dynamics, we have used a conjugate gradient algorithm (Press *et al* 1992) which is a generalized form of the backpropagation learning algorithm (Hertz *et al* 1991). The architecture we propose represents a step toward understanding how neural circuits might process complex temporal patterns.

This paper is organized as follows. First we describe the dynamics of the single synapse model upon which the architecture is based (section 2). Next we show how a small, three-layer feed-forward network of units connected by such synapses can be trained to approximate a nonlinear input–output system (section 3). This training involves adjusting, by means of a conjugate gradient algorithm (Press *et al* 1992), a subset of the parameters that govern the synaptic dynamics; these parameters might be subject to plasticity in biological systems through mechanisms such as long term potentiation and depression. We also show that a such a three-layer feed-forward network with biologically realistic synaptic dynamics yields performance comparable to that of artificial networks that were previously designed to yield good performance in the time series domain without any claims of biological realism (section 4). We then assess which parameters are essential to produce good network performance (section 5). Finally we demonstrate that it is the synaptic rather than neuronal dynamics that are playing the critical role in the computation by showing how the same computation can be achieved with only two neurons in the hidden layer, as long as the neurons are connected through multiple synapses (section 6).

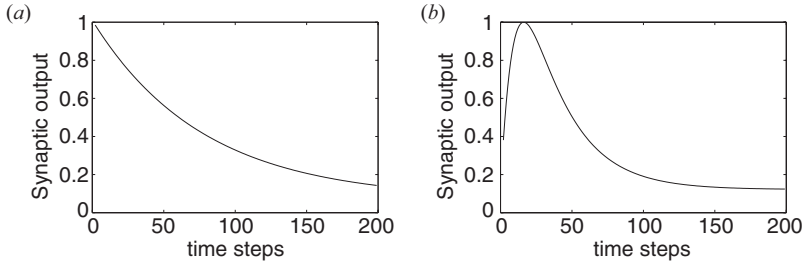

Figure 1. A single synapse can produce quite different outputs for the same input. The response of a single synapse to a step increase in input activity applied at time step 0 is compared for two different parameter settings. In (*a*), the synapse responds with pure depression, while in (*b*) an initial facilitation precedes the depression.

2. Single synapse

We begin by describing a formal model of a single synapse. The continuum model we consider is related to several proposed previously (Abbott *et al* 1997, Markram *et al* 1998, Tsodyks *et al* 1998, Maass and Zador 1999). It incorporates short term facilitation and depression.

Synaptic strength depends on three quantities: presynaptic activity $x_j(t)$; a use-dependent term $p_{ij}(t)$ which may loosely related to presynaptic release probability; and postsynaptic efficacy W_{ij}. Synaptic dynamics arise from the dependence of presynaptic release probability on the history of presynaptic activity. Specifically, the effect of activity $x_j(t)$ in the jth presynaptic unit on the ith postsynaptic unit is given by the product of the synaptic coupling between the two units and the instantaneous presynaptic activity, $x_j(t) \cdot p_{ij}(t) \cdot W_{ij}$. The presynaptic activity $x_j(t)$ is a continuous value (constrained to fall in the range [0, 1]) rather than a discrete spike train, and can be considered to represent an instantaneous firing rate. The coupling is in turn the product of a history-dependent 'release probability' $p_{ij}(t)$, and a static scale factor W_{ij} corresponding to the postsynaptic response or 'potency' at the synapse connecting j and i. Note that $W_{ji} \geqslant 0$ for excitatory synapses and $W_{ji} \leqslant 0$ for inhibitory synapses.

The history-dependent component $p_{ij}(t)$ is constrained to fall in the range [0, 1]. This component in turn depends on two auxiliary history-dependent functions $f_{ij}(t)$ and $d_{ij}(t)$. The quantity $d_{ij}(t)$ can be interpreted as the number of releasable synaptic vesicles; it decreases with activity and thereby instantiates a form of use-dependent depression. The quantity $f_{ij}(t)$ represents the propensity of each vesicle to be released; like (Ca^{2+}) in the presynaptic terminal, it increases with presynaptic activity $x_j(t)$ and thereby instantiates a form of facilitation. The details of the activity-dependence of $f_{ij}(t)$ and $d_{ij}(t)$ are given in appendix A.

The input–output behaviour of this model synapse depends on the four synaptic parameters U_{ij}, F_{ij}, D_{ij} and W_{ij}, as described in appendix A. The same input yields markedly different outputs for different values of these parameters. Figure 1 compares the output of a single synapse in response to a step input, i.e. $x_j(t) = 1$ for $t > 0$, for two sets of synaptic parameters. In figure 1(*a*), the output begins at a maximal value and then, declines to nearly zero while in figure 1(*b*) the response increases to a maximum and then decreases. These examples illustrate just two of the range of input–output behaviours that a single synapse can achieve.

Note that the qualitative aspects of the results presented in this paper do not critically depend on the particular model used for synaptic dynamics. In Natschläger (1999) a continuum version of the model proposed in Maass and Zador (1999) is considered, and the results are indeed very similar.

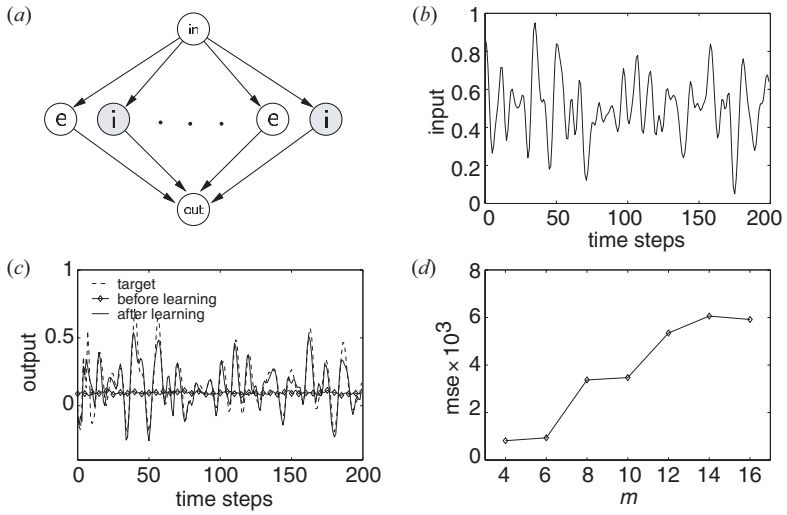

Figure 2. A network with units coupled by dynamic synapses can approximate randomly drawn quadratic filters. (*a*) Network architecture. The network had one input unit, ten hidden units (five excitatory, five inhibitory), and one output unit. The activation function at the hidden units was sigmoidal, but linear at the output unit. (*b*) One of the input patterns used in the training ensemble. For clarity, only a portion of the actual input is shown. (*c*) Output of the network prior to training, with random initialization of the parameters (—◇—). Output of the dynamic network (DN) *after* learning (——). The target (– – –) was the output of the quadratic filter given by equation (1), the coefficients h_{kl} ($1 \leqslant k, l \leqslant 10$) of which were generated randomly by subtracting $\mu/2$ from a random number generated from an exponential distribution with mean μ. (*d*) Performance after network training. For different sizes of $\boldsymbol{H}$ ($\boldsymbol{H}$ is a symmetric $m \times m$ matrix) we plotted the average performance (MSE measured on a test set) over 20 different filters $\mathcal{Q}$, i.e. 20 randomly generated matrices $\boldsymbol{H}$.

3. Processing dynamic signals

Research on conventional artificial neural networks has emphasized tasks, such as the classification of static images, in which both the inputs and the outputs are devoid of temporal structure. However, ecologically relevant signals often have a rich temporal structure, and neural circuits must process these signals in real time. In many signal processing tasks, such as audition, almost all of the information is embedded in the temporal structure. In the visual domain, movement represents one of the fundamental features extracted by the nervous system. In the following we refer to systems which map a time varying signal onto another time varying signal as *filter*.

The dynamic synapses we have described are ideally suited to process signals with temporal structure (figure 2(*a*)). To illustrate this, we consider a simple class of signals given by a quadratic filter[4] $\mathcal{Q}$:

$$\mathcal{Q}x(t) = \sum_{l=1}^{m} \sum_{k=1}^{m} h_{kl}\, x(t-k)\, x(t-l) \tag{1}$$

where t is discrete time, and $x(t)$ is the input, and the filter coefficients h_{kl} form a arbitrary $m \times m$ matrix $\boldsymbol{H}$ (we assume in this paper that $\boldsymbol{H}$ is symmetric). An example of the input and output for one choice of quadratic parameters are shown in figures 2(*b*) and (*c*), respectively. The filter

[4] We adopt the common notation $\mathcal{F}x(t)$ to denote the output that the filter $\mathcal{F}$ gives at time t for the input function x.

$\mathcal{Q}$ is an idealization of the kinds of complex transformations that are important to an organism's survival, such as those required for motor control and the processing of time-varying sensory inputs. For example, the spectrotemporal receptive field of a neuron in the auditory cortex (deCharms and Merzenich 1998, Kowalski *et al* 1996) reflects some complex transformation of sound pressure to neuronal activity. The real transformations actually required for survival may be very complex, but the simple filter $\mathcal{Q}$ provides a useful starting point for assessing the capacity of this architecture to transform one time-varying signal into another.

Can a network of units coupled by dynamic synapses implement the filter $\mathcal{Q}$? We tested the approximation capabilities of a rather small network with just ten hidden units (five excitatory and five inhibitory ones), and one output (figure 2(*a*)). The output $x_i(t)$ of the ith unit is given by

$$x_i(t) = \sigma\left(\sum_j W_{ij} \cdot p_{ij}(t) \cdot x_j(t)\right) \tag{2}$$

where $x_j(t)$ is the input from the previous layers, $p_{ij}(t)$ corresponds to the activity-dependent release probability, W_{ij} to the static postsynaptic efficacy, and σ is either the sigmoid function $\sigma(u) = 1/(1 + \exp(-u))$ (in the hidden layers) or just the identity function $\sigma(u) = u$ (in the output layer). In the following we refer to such networks as dynammic networks (DNs). The dynamics of inhibitory synapses is described by the same model as that for excitatory synapses. For any particular temporal pattern applied at the input and any particular choice of the synaptic parameters, this network generates a temporal pattern as output. This output can be thought of, for example, as the activity of a particular population of neurons in the cortex, and the target function as the time series generated for the same input by some unknown quadratic filter $\mathcal{Q}$. The synaptic parameters W_{ij}, D_{ij}, F_{ij} and U_{ij} are chosen so that, for each input in the training set, the network minimized the mean-square error (MSE)

$$E[z, z_{\mathcal{Q}}] = \frac{1}{T}\sum_{t=1}^{T}(z(t) - z_{\mathcal{Q}}(t))^2 \tag{3}$$

between its output $z(t)$ and the desired output $z_{\mathcal{Q}}(t) = \mathcal{Q}x(t)$ specified by the filter $\mathcal{Q}$. To achieve this minimization, we used a conjugate gradient algorithm, see appendix B for details.

The training inputs were random signals, an example of which is shown in figure 2(*b*). The test inputs were drawn from the same random distribution as the training inputs, but were not actually used during training. This test of generalization ensured that the observed performance represented more than simple 'memorization' of the training set. To avoid overfitting, minimization of $E[z, z_{\mathcal{Q}}]$ was stopped when the error on a validation set (distinct from training and test set) reached its first minimum. Figure 2(*c*) compares the network performance before and after training. Prior to training, the output is nearly flat, while after training the network output tracks the filter output closely ($E[z, z_{\mathcal{Q}}] = 0.0032$).

Figure 2(*d*) shows the performance after training for different randomly chosen quadratic filters $\mathcal{Q}$ with different dimensions m of $\boldsymbol{H}$. Even for larger values of m the relatively small network with ten hidden units performs rather well. Note that a quadratic filter of dimension m has $m(m + 1)/2$ free parameters, whereas the DN has a constant number of 80 adjustable parameters. This shows clearly that dynamic synapses enable a small network to mimic a wide range of possible quadratic target filters.

4. Comparison with the model of Back and Tsoi

Our DN model is not the first to incorporate temporal dynamics via dynamic synapses. Perhaps the earliest suggestion for a role for synaptic dynamics in network computation was by Little

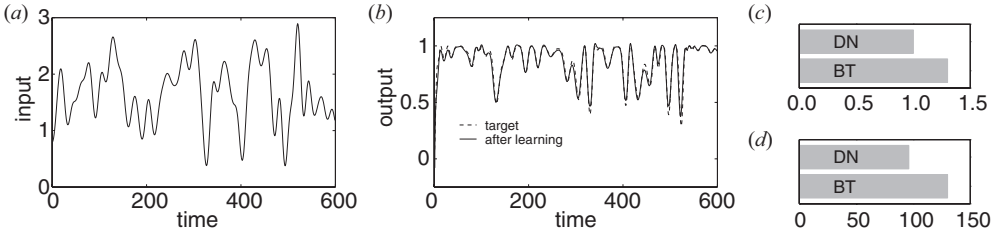

Figure 3. Performance of our model on the system identification task used in Back and Tsoi (1993). The network architecture is the same as in figure 2. (*a*) One of the input patterns used in the training ensemble. (*b*) Output of the network *after* learning (——). The target (– – –) is the output of the filter function given by equations (4) and (5). (*c*) Comparison of the MSE (in units of 10^{-3}) achieved on test data by the model of BT and by the DN. (*d*) Comparison of the number of adjustable parameters. The network model of BT utilizes slightly more adjustable parameters than the DN.

and Shaw (1975). More recently, a number of networks have been proposed in which synapses implemented linear filters; in particular see Back and Tsoi (1993).

To assess the performance of our network model in relation to the model proposed in Back and Tsoi (1993) we have analysed the performance of our DN model for the same system identification task that was employed as benchmark task in Back and Tsoi (1993). The goal of this task is to learn the filter $\mathcal{F}$

$$z_{\mathcal{F}}(t) = \mathcal{F}x(t) = \sin(u(t)) \tag{4}$$

where $u(t)$ is the solution to the difference equation

$$u(t) - 1.99u(t-1) + 1.572u(t-2) - 0.4583u(t-3) = 0.0154x(t) + 0.0462x(t-1) + 0.0462x(t-2) + 0.0154x(t-3). \tag{5}$$

Hence, $u(t)$ is the output of a linear filter applied to the input $x(t)$.

The result is summarized in figure 3. It can clearly be seen that our network model (see figure 2(*a*) for the network architecture) is able to learn this particular filter. The MSE on the test data is 0.0010, which is slightly smaller than the MSE of 0.0013 reported in Back and Tsoi (1993). Note that the network Back and Tsoi (BT) used to learn the task had 130 adjustable parameters (13 parameters per IIR synapse, ten hidden units) whereas our network model had only 80 adjustable parameters (all parameters U_{ij}, F_{ij}, D_{ij} and W_{ij} were adjusted during learning).

Hence simple feedforward networks with biologically realistic synaptic dynamics are comparable (no significant differences in performance and network size) to artificial networks that were previously designed to yield good performance in the time series domain without any claims of biological realism. This indicates that dynamic synapses entail feedforward networks with the same computational power as infinite impulse response (IIR) synapses (Back and Tsoi 1993)[5].

5. Which parameters matter?

It remains an open experimental question which synaptic parameters are subject to use-dependent plasticity, and under what conditions. For example, long term potentiation appears

[5] Note that empirical comparisons between different models are difficult due to the number of factors that can be involved.

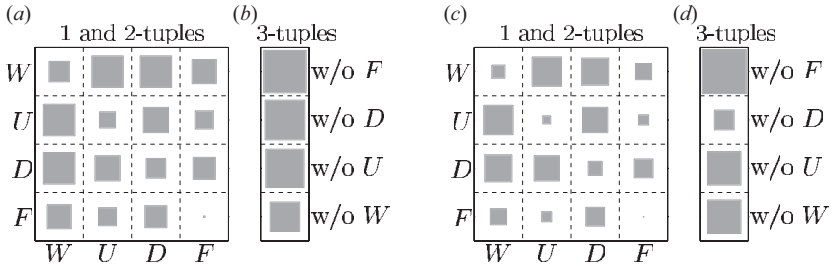

Figure 4. Impact of different synaptic parameters on the learning capabilities of a DN. The area of a square (the 'impact') is proportional to the inverse of the MSE averaged over N trials. (*a*) In each trial ($N = 100$) a different quadratic filter matrix $\boldsymbol{H}$ ($m = 6$) was randomly generated as described in figure 2. Along the diagonal one can see the impact of a single parameter, whereas the off-diagonal elements (which are symmetric) represent the impact of changing pairs of parameters. (*b*) The impact of subsets of size three is shown where the labels indicate which parameter is not included. (*c*) Same interpretation as for (*a*) but the results shown ($N = 20$) are for the filter used in Back and Tsoi (1993) (equations (4) and (5)). (*d*) Same interpretation as for (*b*) but the results shown ($N = 20$) are for the same filter as in (*c*).

to change synaptic dynamics between pairs of layer 5 cortical neurons (Markram and Tsodyks 1996) but not in the hippocampus (Selig *et al* 1999). We therefore wondered whether plasticity in the synaptic dynamics is essential for a DN to be able to learn a particular target filter. To address this question, we compared network performance when different parameter subsets were optimized using the conjugate gradient algorithm, while the other parameters were held fixed. In all experiments, the fixed parameters were chosen to ensure heterogeneity in presynaptic dynamics.

Figure 4 shows that changing only the postsynaptic parameter W has comparable impact to changing only the presynaptic parameters U or D, whereas changing only F has little impact on the dynamics of these networks (see diagonal of figures 4(*a*) and (*c*)). However, to achieve good performance one has to change at least two different types of parameters such as $\{W, U\}$ or $\{W, D\}$ (all other pairs yield worse performance). Hence, neither plasticity in the presynaptic dynamics (U, D, F) alone nor plasticity of the postsynaptic efficacy (W) alone was sufficient to achieve good performance in this model.

6. Multiple neurons and multiple synapses

So far we have assumed that each axon makes only one synapse onto its postsynaptic target. While such connectivity is common in the hippocampus (Harris and Stevens 1989), in the neocortex and elsewhere the multiplicity is often higher, so that a single presynaptic axon makes several independent contact with the postsynaptic target (Markram *et al* 1997). We therefore tested a modified architecture in which each axon made several synapses. The parameters at each synapse were modified independently.

Figure 5 shows a limiting case an architecture with high synapse multiplicity. The number of plastic synapses is the same as in figure 5, but here instead of ten hidden units there are only two. The performance of the network is as good as the performance of the network considered in figure 2 (cf figures 2(*d*) and 5(*d*)), emphasizing that it is the synaptic and not the neuronal dynamics that are the key to this architecture. If, as these results suggest, synapses can under some conditions replace neurons with little loss of computational power, the strong pressures to maximise wiring economy (Chklovskii 1998) might favour synapse multiplicity.

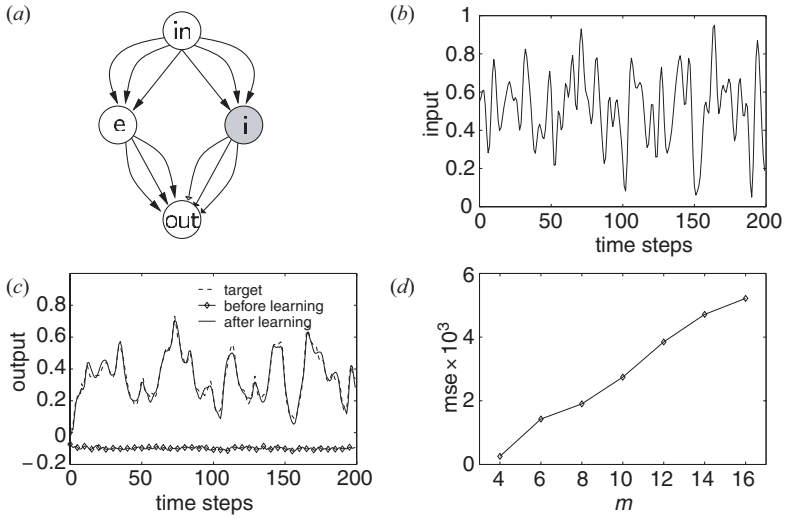

Figure 5. Dynamic synapses can substitute for neurons. (*a*) Network architecture. In contrast to the network in figure 2, this network had only two hidden units (one excitatory and one inhibitory), but with higher synapse multiplicity (five synapses/axon). (*b*) One of the input patterns used in the training ensemble. For clarity, only a portion of the actual input is shown. (*c*) Output of the network prior to training, with random initialization of the parameters (—◇—). Output of the DN *after* learning (——). The target (– – –) was the output of the quadratic filter given by equation (1), the coefficients h_{kl} ($1 \leqslant k, l \leqslant 10$) of which were generated randomly by subtracting $\mu/2$ from a random number generated from an exponential distribution with mean μ. (*d*) Performance after network training. For different sizes of $\boldsymbol{H}$ ($\boldsymbol{H}$ is a symmetric $m \times m$ matrix) we plotted the average performance (MSE measured on a test set) over 20 different filters $\mathcal{Q}$, i.e. 20 randomly generated matrices $\boldsymbol{H}$.

7. A universal approximation theorem for dynamic networks

In the preceding sections we have presented empirical evidence for the approximation capabilities of our DN model for computations in the time series domain. This gives rise to the question: what are the theoretical limits of their approximation capabilities? The rigorous theoretical result presented in this section shows that basically there are no significant *a priori* limits. Furthermore, in spite of the rather complicated system of equations that defines DNs, one can give a precise mathematical characterization of the class of filters that can be approximated by them. This characterization involves the following basic concepts.

An arbitrary filter $\mathcal{F}$ is called *time invariant* if a shift of the input functions by a constant t_0 just causes a shift of the output function by the same constant t_0.

Another essential property of filters is *fading memory*. A filter $\mathcal{F}$ has fading memory if and only if the value of $\mathcal{F}\underline{x}(0)$ can be approximated arbitrarily closely by the value of $\mathcal{F}\underline{\tilde{x}}(0)$ for functions $\underline{\tilde{x}}$ that approximate the functions $\underline{x}$ for sufficiently long bounded intervals $[-T, 0]$.

Interesting examples of linear and nonlinear time invariant filters with fading memory can be generated with the help of representations of the form

$$\mathcal{F}x(t) = \int_0^\infty \cdots \int_0^\infty x(t-\tau_1)\ldots x(t-\tau_k)h(\tau_1,\ldots,\tau_k)\,\mathrm{d}\tau_1,\ldots,\mathrm{d}\tau_k$$

for measurable and essentially bounded functions $x : \mathbb{R} \to \mathbb{R}$ (with $h \in L^1$). One refers to such an integral as a *Volterra term of order k*. Note that for $k = 1$ it yields the usual representation for a *linear* time invariant filter. The class of filters that can be represented by

Volterra series, i.e., by finite or infinite sums of Volterra terms of arbitrary order, has been investigated for quite some time in neurobiology (Rieke *et al* 1997) and engineering (Schetzen 1980).

Theorem 1. *Assume that X is the class of functions from $\mathbb{R}$ into $[B_0, B_1]$ which satisfy $|x(t) - x(s)| \leqslant B_2 \cdot |t - s|$ for all $t, s \in \mathbb{R}$, where B_0, B_1, B_2 are arbitrary real-valued constants with $0 < B_0 < B_1$ and $0 < B_2$. Let $\mathcal{F}$ be an arbitrary filter that maps vectors of functions $\underline{x} = \langle x_1, \ldots, x_n \rangle \in X^n$ into functions from $\mathbb{R}$ into $\mathbb{R}$.*

Then the following are equivalent:

(a) $\mathcal{F}$ can be approximated by DNs $\mathcal{N}$ defined by equations (2) and (A.1) (i.e., for any $\varepsilon > 0$ there exists such network $\mathcal{N}$ such that $|\mathcal{F}\underline{x}(t) - \mathcal{N}\underline{x}(t)| < \varepsilon$ for all $\underline{x} \in X^n$ and all $t \in \mathbb{R}$).
(b) $\mathcal{F}$ can be approximated by DNs according to equations (2) and (A.1) with just a single layer of sigmoidal neurons.
(c) $\mathcal{F}$ is time invariant and has fading memory.
(d) $\mathcal{F}$ can be approximated by a sequence of (finite or infinite) Volterra series.

The *proof* of theorem 1 relies on the Stone–Weierstrass theorem, and is contained as the proof of theorem 3.4 in Maass and Sontag (2000).

The *universal approximation result* contained in theorem 1 turns out to be rather robust with regard to changes in the definition of a DN. DNs with just one layer of dynamic synapses and one subsequent layer of sigmoidal gates can approximate the same class of filters as DNs with an arbitrary number of layers of dynamic synapses and sigmoidal neurons. For details we refer the interested reader to Maass and Sontag (2000).

This theoretical analysis does not address the question how large such DNs have to be in order to approximate a given filter. However, the computer experiments reported in sections 3–6 of this paper provide first (empirical) data regarding these complexity issues.

8. Discussion

Our central hypothesis is that rapid changes in synaptic strength, mediated by mechanisms such as facilitation and depression, are an integral part of neural processing. We have proposed a general computational model in which such rapid changes endow a neural circuit with the capacity to process temporal patterns. This model differs from most conventional models of neural computation, based on static synapses, in which synaptic strength changes during learning but not during performance. The architecture we propose provides a framework for studying how neural circuits compute in real time.

We have used a simple task—a quadratic filter—to illustrate the potential of this architecture. This task allows us to focus on temporal dynamics, an essential aspect of cortical computation that is absent from many artificial neural network formulations. In this task, the goal is to transform a time-varying input into the appropriate time-varying output; our results thereby complement Buonomano and Merzenich (1995), where synaptic dynamics are used to transform temporal patterns into spatial patterns. Such a transformation from one time-varying signal to another must be performed, for example, to generate the motor commands used involved in reaching, or in the real-time recognition of speech sounds.

Our very general framework differs from the more specific computational roles, such as gain control (Abbott *et al* 1997), that have been proposed for synaptic dynamics. Gain control is a mechanism that allows the input–output transformation to remain invariant over a wide range of input intensities. To achieve gain control, synaptic efficacy rapidly adapts to compensate for changes in the neuronal firing rate. Gain control thus represents an important

special case of the larger role we are proposing for synaptic dynamics. Indeed, the conjugate gradient algorithm we have used enables the present architecture to implement nearly arbitrary transformations of one time-varying signal into another.

In the supervised learning paradigm we have explored here, a neural circuit is trained to approximate a fully specified input–output system, where both the inputs and the outputs are time-varying functions. We have focused on this paradigm not because we believe it is necessarily the best model for learning in neural circuits—we are not proposing that synapses in cortical circuits are subject to modification by the kind of learning algorithm we have used—but rather because it is the best understood paradigm. Our results represent part of the larger program of incorporating the key features of neural circuits into simple and tractable mathematical formulations. Since our formalism is a natural extension of artificial neural networks, it should be possible to derive comparable results from other paradigms, including unsupervised and reinforcement learning.

Analytical results show that the class of nonlinear filters that can be approximated by DNs, even with just a single hidden layer of sigmoidal neurons, is remarkably rich. It contains every time invariant filter with fading memory, hence arguable every filter that is potentially useful for a biological organism.

The computer simulations we performed show that rather small DNs are not only able to perform interesting computations on time series, but their performance is comparable to that of previously considered artificial neural networks that were designed for the purpose of yielding efficient processing of temporal signals. We have tested DNs on tasks such as the learning of a randomly chosen quadratic filter, as well as on the system identification task used in Back and Tsoi (1993), to illustrate the potential of this architecture. The results are very encouraging.

Acknowledgments

We would like to thank Lynn Dobrunz, Virginia de Sa and Zachary Mainen for comments, and TZ would like to thank Chuck Stevens for his generous support. This paper was supported by the Salk Sloan Foundation for Theoretical Neuroscience, project no P12153 of the Fonds zur Förderung wissenschaftlicher Forschung, and the NeuroCOLT project of the EC.

Appendix A. Single synapse model

The model is described in detail in Tsodyks *et al* (1998). For convenience we restate the equations in our notation, which read as follows

$$\begin{aligned} p_{ij}(t) &= f_{ij}(t) \cdot d_{ij}(t) \\ \frac{\mathrm{d}\bar{f}_{ij}(t)}{\mathrm{d}t} &= -\frac{\bar{f}_{ij}(t)}{F_{ij}} + U_{ij} \cdot (1 - \bar{f}_{ij}(t)) \cdot x_j(t) \\ \frac{\mathrm{d}d_{ij}(t)}{\mathrm{d}t} &= \frac{1 - d_{ij}(t)}{D_{ij}} - p_{ij}(t) \cdot x_j(t) \\ f_{ij}(t) &= \bar{f}_{ij}(t) \cdot (1 - U_{ij}) + U_{ij} \end{aligned} \tag{A.1}$$

with $d_{ij}(0) = 1$ and $\bar{f}_{ij}(0) = 0$. $f_{ij}(t)$ models facilitation (with time constant $F_{ij} > 0$ and the 'initial release probability' $U_{ij} \in [0, 1]$), whereas $d_{ij}(t)$ models the combined effects of synaptic depression (with time constant $D_{ij} > 0$) and facilitation. Hence, the dynamics of a synaptic connection is characterized by the three parameters $U_{ij} \in [0, 1]$, $D_{ij} > 0$, $F_{ij} > 0$. For the numerical results presented in this paper we consider a discrete time ($t = 1, \ldots, T$)

version of the model defined by equation (A.1). In this setting we consider the dynamics

$$\begin{aligned} p_{ij}(t) &= f_{ij}(t) \cdot d_{ij}(t) \\ \bar{f}_{ij}(t+1) &= \bar{f}_{ij}(t) - \frac{\bar{f}_{ij}(t)}{F_{ij}} + U_{ij} \cdot (1 - \bar{f}_{ij}(t)) \cdot x_j(t) \\ d_{ij}(t+1) &= d_{ij}(t) + \frac{1 - d_{ij}(t)}{D_{ij}} - f_{ij}(t) \cdot d_{ij}(t) \cdot x_j(t) \\ f_{ij}(t) &= \bar{f}_{ij}(t) \cdot (1 - U_{ij}) + U_{ij} \end{aligned} \tag{A.2}$$

with the initial conditions $\bar{f}_{ij}(1) = 0$ (i.e. $f_{ij}(1) = U_{ij}$) and $d_{ij}(1) = 1$. Note that in this case the time constants F_{ij} and D_{ij} have to be $\geqslant 1$.

Appendix B. The learning algorithm

In this section we describe the basics of the conjugate gradient learning algorithm—a generalized version of simple gradient descent—which we used to minimize the MSE

$$E[z, z_{\mathcal{F}}] = \frac{1}{T} \sum_{t=0}^{T-1} (z(t) - z_{\mathcal{F}}(t))^2 \tag{B.1}$$

between the network output $z(t)$ in response to the input time series $x(t)$ and the target output $z_{\mathcal{F}}(t) = (\mathcal{F}x)(t)$ provided by the target filter $\mathcal{F}$. The mathematical background of such algorithms can be found, for example, in Hertz *et al* (1991), and an implementation in C is provided in Press *et al* (1992). In order to apply a conjugate gradient algorithm ones has to calculate the partial derivatives $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta U_{ij}}$, $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta D_{ij}}$, $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta F_{ij}}$ and $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta W_{ij}}$ for all synapses $\langle ij \rangle$ in the network.

As an example we state the equations for the partial derivatives $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta U_{ij}}$ for the network architecture shown in figure 2 (i.e. one-dimensional network input and output and a single synapse between a pair of neurons) where $\langle ij \rangle$ is a synapse between a hidden neuron and an output neuron. To simplify further notation we denote synapses from the input to the hidden layer by $\langle kj \rangle$ and synapses from the hidden layer to the output simply by the index k. Using this notation the output $z(t)$ is given as

$$z(t) = \sum_{k \in E} W_k \cdot p_k(t) \cdot y_k(t) - \sum_{k \in I} W_k \cdot p_k(t) \cdot y_k(t) \tag{B.2}$$

where $y_k(t) = \sigma(W_{kj} \cdot p_{kj} \cdot x(t))$ is the output of the kth hidden unit (there is only the single input $x(t)$). The kth hidden unit is either excitatory ($k \in E$) or inhibitory ($k \in I$). Hence $W_k \geqslant 0$ for all $k \in (E \cup I)$. For the partial derivatives $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta U_k}$ we get the rather lengthy expressions

$$\begin{aligned} \frac{\delta E[z, z_{\mathcal{F}}]}{\delta U_k} &= \frac{2}{T} \sum_{t=0}^{T-1} (z(t) - z_{\mathcal{F}}(t)) \cdot \frac{\delta z(t)}{\delta U_k} \\ \frac{\delta z(t)}{\delta U_k} &= W_k \cdot y_k(t) \cdot \frac{\delta p_k(t)}{\delta U_k} = W_k \cdot y_k(t) \cdot \left(d_k(t) \cdot \frac{\delta f_k(t)}{\delta U_k} + \frac{\delta d_k(t)}{\delta U_k} \cdot f_k(t) \right) \\ \frac{\delta f_k(t)}{\delta U_k} &= (1 - U_k) \cdot \frac{\delta \bar{f}_k(t)}{\delta U_k} \\ \frac{\delta \bar{f}_k(t)}{\delta U_k} &= \left(1 - \frac{1}{F_k}\right) \cdot \frac{\delta \bar{f}_k(t-1)}{\delta U_k} + y_k(t) \cdot \left(1 - \bar{f}_k(t-1) - U_k \cdot \frac{\delta \bar{f}_k(t-1)}{\delta U_k}\right) \\ \frac{\delta d_k(t)}{\delta U_k} &= \left(1 - \frac{1}{D_k}\right) \cdot \frac{\delta d_k(t-1)}{\delta U_k} \\ &\quad - y_k(t) \cdot \left(d_k(t-1) \cdot \frac{\delta f_k(t-1)}{\delta U_k} + \frac{\delta d_k(t-1)}{\delta U_k} \cdot f_k(t-1) \right) \end{aligned}$$

and the initial conditions $\frac{\delta d_k(1)}{\delta U_k} = 0$ and $\frac{\delta f_k(1)}{\delta U_k} = 1$.

Note that these equations relate the derivatives $\frac{\delta f_k(t)}{\delta U_k}$ and $\frac{\delta d_k(t)}{\delta U_k}$ at time t to the derivatives $\frac{\delta f_k(t-1)}{\delta U_k}$ and $\frac{\delta d_k(t-1)}{\delta U_k}$ at time $t-1$ similar as in real-time recurrent learning; see e.g. Hertz *et al* (1991, section 7.3). Hence one can calculate all the derivatives in one 'sweep' from $t = 1$ to T. Similar equations are obtained for the partial derivatives $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta D_k}$, $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta F_k}$ and $\frac{\delta E[z,z_{\mathcal{F}}]}{\delta W_k}$ and for synapses $\langle kj \rangle$ connecting the input neuron to the hidden neurons (in this case the equations get even more complex).

To ensure that the parameters $0 \leqslant U \leqslant 1$, $D \geqslant 1$, $F \geqslant 1$ and $W \geqslant 0$ (indices skipped for clarity) stay within their allowed range we introduce the 'unbounded' parameters $\tilde{U}$, $\tilde{D}$, $\tilde{F}$, $\tilde{W} \in \mathbb{R}$ with the following relationships to the original parameters: $U = 1/(1+\exp(-\tilde{U}))$, $D = 1+\exp(\tilde{D})$, $F = 1+\exp(\tilde{F})$, $W = \exp(\tilde{W})$. The conjugate gradient algorithm was then used to adjust these unbounded parameters which are allowed to have any value in $\mathbb{R}$. The partial derivatives of $E[z, z_{\mathcal{F}}]$ with respect to the unbounded parameters can easily be obtained by applying the chain rule, e.g.

$$\frac{\delta E[z, z_{\mathcal{F}}]}{\delta \tilde{U}} = \frac{\delta E[z, z_{\mathcal{F}}]}{\delta U} \cdot \frac{\delta U}{\delta \tilde{U}} = \frac{\delta E[z, z_{\mathcal{F}}]}{\delta U} \cdot \frac{\exp \tilde{U}}{(1+\exp(-\tilde{U}))^2}.$$

References

Abbott L, Varela J, Sen K and Nelson S 1997 Synaptic depression and cortical gain control *Science* **275** 220–4

Back A D and Tsoi A C 1993 A simplified gradient algorithm for IIR synapse multilayer perceptrons *Neural Comput.* **5** 456–62

Buonomano D and Merzenich M 1995 Temporal information transformed into a spatial code by a neural network with realistic properties *Science* **267** 1028–30

Chklovskii D B 1998 Binocular disparity and the pattern of ocular dominance stripes in primates *Soc. Neurosci. Ab.* **24** 645

deCharms R and Merzenich M 1998 Optimizing sound features for cortical neurons *Science* **280** 1439–43

Dobrunz L and Stevens C F 1999 Response of hippocampal synapses to natural stimulation patterns *Neuron* **22** 157–66

Harris K and Stevens J 1989 Dendritic spines of CA1 pyramical cells in the rat hippocampus: serial electron microscopy with reference to their biophyscial characteristics *J. Neurosci.* **9** 2982–97

Hertz J, Krogh A and Palmer R 1991 *Introduction to the Theory of Neural Computation* (New York: Addison-Wesley)

Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc. Natl Acad. Sci. USA* **79** 2554–8

Kowalski N, Depireux D and Shamma S 1996 Analysis of dynamic spectra in ferret primary auditory cortex. i. characteristics of single-unit responses to moving ripple spectra *J. Neurophys.* **76** 3503–23

Liaw J-S and Berger T 1996 Dynamic synapse: a new concept of neural representation and computation *Hippocampus* **6** 591–600

Little W and Shaw G 1975 A statistical theory of short and long term memory *Behav. Biol.* **14** 115–33

Maass W and Sontag E D 2000 Neural systems as nonlinear filters *Neural Comput.* **12** 1743–72

Maass W and Zador A 1999 Dynamic stochastic synapses as computational units *Neural Comput.* **11** 903–17

Magleby K 1987 Short term synaptic plasticity *Synaptic Function* ed G M Edelman, W E Gall and W M Cowan (New York: Wiley)

Markram H, Lubke J, Frotscher M, Roth A and B S 1997 Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex *J. Physiol.* **500** 409–40

Markram H and Tsodyks M 1996 Redistribution of synaptic efficacy between neocortical pyramidal neurons *Nature* **382** 807–10

Markram H, Wang Y and Tsodyks M 1998 Differential signaling via the same axon of neocortical pyramidal neurons *Proc. Natl Acad. Sci. USA* **95** 5323–8

Natschläger T 1999 Efficient computation in networks of spiking neurons—simulations and theory *PhD Thesis* Graz University of Technology

NeuroCOLT2 *Technical Report 1999-050* webpage www.neurocolt.com

Press W H, Teukolsky S A, Vetterling W T and Flannery B P (ed) 1992 *Numerical Recipes in C* (Cambridge: Cambridge University Press) pp 394–455 chapter 10

Reid R, Victor J and Shapley R 1997 The use of m-sequences in the analysis of visual neurons: linear receptive field properties *Vis. Neurosci.* **14** 1015–27

Rieke F, Warland D, de Ruyter van Steveninck R and Bialek W 1997 *Spikes: Exploring the Neural Code* (Cambridge, MA: MIT Press)

Schetzen M 1980 *The Volterra and Wiener Theories of Nonlinear Systems* (New York: Wiley)

Sejnowski T J 1977 Statistical constraints on synaptic plasticity *J. Theor. Biol.* **69** 385–9

Selig D, Nicoll R and Malenka R 1999 Hippocampal long-term potentiation preserves the fidelity of postsynaptic responses to presynaptic bursts *J. Neurosci.* **19** 1236–46

Tsodyks M, Pawelzik K and Markram H 1998 Neural networks with dynamic synapses *Neural Comput.* **10** 821–35

Zador A and Dobrunz L 1997 Dynamic synapses in the cortex *Neuron* **19** 1–4