Efficient Temporal Processing with Biologically Realistic Dynamic Synapses

Thomas Natschläger, Wolfgang Maass
Institute for Theoretical Computer Science
Technische Universität Graz
tnatschl,maass@igi.tu-graz.ac.at

Anthony Zador
Cold Spring Harbor Laboratory
zador@cshl.org

Abstract

Experimental data show that biological synapses behave quite differently from the symbolic synapses in common artificial neural network models. Biological synapses are dynamic, i.e., their "weight" changes on a short time scale by several hundred percent in dependence of the past input to the synapse. 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 demonstrate that with this approach the nonlinear filter considered in [1] can be approximated slightly better than by their model. Our numerical results are complemented by theoretical analysis which show that even with just a single hidden layer such networks can approximate a surprisingly large 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

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 [2] and spectrotemporal [3, 4] 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 [5, 6]. 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 time scale of learning. In real nervous systems, by contrast, synapses show dynamics on short time scales, from milliseconds to seconds [7, 8, 9, 10, 11]. 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 [12, 13, 14]. 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 [15] which is a generalized form of the backpropagation learning algorithm [16]. The architecture we propose represents a step toward understanding how neural circuits might process complex temporal patterns.

The article 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, 3-layer feed-forward network of units connected by such synapses can be trained to approximate a non-linear input-output system (Section 3). This training involves adjusting, by means of a conjugate gradient algorithm [15], 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 3-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).

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 [9, 17, 13, 18]. 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 j^{th} presynaptic unit on the i^{th} 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

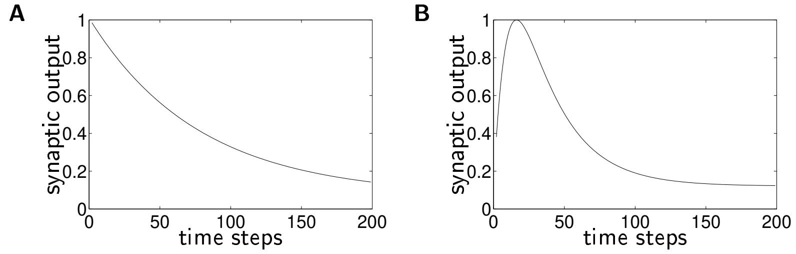

Figure 1: A single synapse can produce quite different outputs for the same input. The response of a single synapse to a step increase in input activity applied at time step 0 is compared for two different parameter settings. In **A**, the synapse responds with pure depression, while in **B** an initial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

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.

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 A.

The input-output behavior of this model synapse depends on four the synaptic parameters U_{ij}, F_{ij}, D_{ij} and W_{ij}, as described in A. The same input yields markedly different outputs for different values of these parameters. Fig. 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 Fig. 1A, the output begins at a maximal value and then, declines to nearly zero while in Fig. 1B the response increases to a maximum and then decreases. These examples illustrate just two of the range of input-output behaviors that a single synapse can achieve.

Note that the qualitative aspects of the results presented in this article do not critically depend on the particular model used for synaptic dynamics. In [19] a continuum version of the model proposed in [18] is considered, and the results are indeed very similar.

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 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 (Fig. 2A). To illustrate this, we consider a simple class of signals given by a quadratic filter $\mathscr{Q}$:[1]

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

where t is time, Δ is some time delay, $x(t)$ is the input, and the filter coefficients h_{kl} form a arbitrary $m \times m$ matrix H (we assume in this article that H is symmetric). An example of the input and output for one choice of quadratic parameters are shown in Figs. 2B and 2C, respectively. The filter $\mathscr{Q}$ 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 [4, 3] 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 $\mathscr{Q}$ 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 $\mathscr{Q}$? We tested the approximation capabilities of a rather small network with just 10 hidden units (5 excitatory and 5 inhibitory ones), and one output (Fig. 2A). The output $x_i(t)$ of the i^{th} unit is iven 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 *dynamic networks*. 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 $\mathscr{Q}$. 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

$$E[z, z_{\mathscr{Q}}] = \frac{1}{N} \sum_{t=0}^{T-1} (z(t) - z_{\mathscr{Q}}(t))^2 \tag{3}$$

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

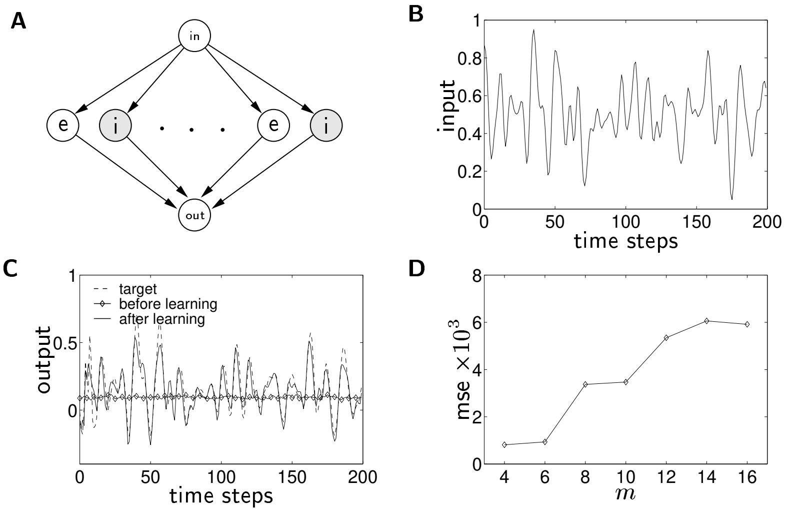

Figure 2: A network with units coupled by dynamic synapses can approximate randomly drawn quadratic filters. **A** Network architecture. The network had one input unit, 10 hidden units (5 excitatory, 5 inhibitory), and one output unit. The activation function at the hidden units was sigmoidal, but linear 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 *after* learning (——). The target (– – –) was the output of the quadratic filter given by Eq. (1), the coefficients h_{kl} ($1 \leq k, l \leq 10$) 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 H (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 H.

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 [15].[2]

The training inputs were random signals, an example of which is shown in Fig. 2B. 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)

[2]In order to apply such a conjugate gradient algorithm ones has to calculate the partial derivatives $\frac{\delta E[z,z_{\mathcal{Q}}]}{\delta U_{ij}}$, $\frac{\delta E[z,z_{\mathcal{Q}}]}{\delta D_{ij}}$, $\frac{\delta E[z,z_{\mathcal{Q}}]}{\delta F_{ij}}$ and $\frac{\delta E[z,z_{\mathcal{Q}}]}{\delta W_{ij}}$ for all synapses $\langle ij \rangle$ in the network. When one performs these rather straightforward calculations one gets equations which relates the derivatives $\frac{\delta f(t)}{\delta U}$ (indices omitted for clarity) at time t to the derivatives $\frac{\delta f(t-1)}{\delta U}$ at time $t-1$ similar as in real-time recurrent learning [16, Section 7.3]. The same holds for $d(t)$ and the other parameters D, F and W. Hence one can calculate the derivatives $\frac{\delta f(t)}{\delta U}$ in one "sweep" from $t = 0$ to $t = N$.

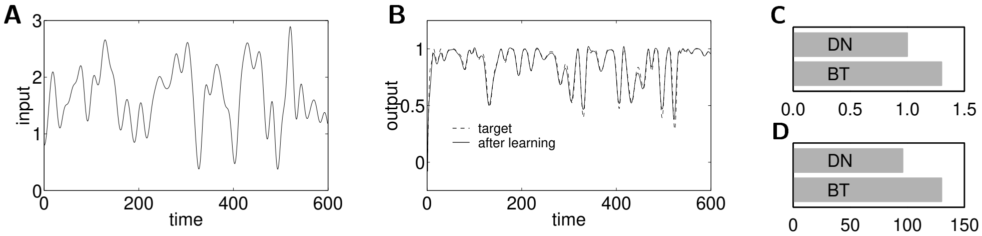

Figure 3: Performance of our model on the system identification task used in [1]. The network architecture is the same as in Fig. 2. **A** One of the input patterns used in the training ensemble. **B** Output of the network *after* learning (—). The target (− − −) is the output of the filter function given by Eq. (4) and Eq. (5). **C** Comparison of the mean square error (in units of 10^{-3}) achieved on test data by the model of Back and Tsoi (BT) and by the dynamic network (DN). **D** Comparison of the number of adjustable parameters. The network model of Back and Tsoi (BT) utilizes slightly more adjustable parameters than the dynamic network (DN).

reached its first minimum. Fig. 2C 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 ($E[z, z_{\mathscr{Q}}] = 0.0032$).

Fig. 2D shows the performance after training for different randomly chosen quadratic filters $\mathscr{Q}$ with different dimensions m of H. Even for larger values of m the relatively small network with 10 hidden units performs rather well. Note that a quadratic filter of dimension m has $m(m+1)/2$ free parameters, whereas the dynamic network 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 dynamic network 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 [12]. More recently, a number of networks have been proposed in which synapses implemented linear filters; in particular [1].

To assess the performance of our network model in relation to the model proposed in [1] we have analyzed the performance of our dynamic network model for the same system identification task that was employed as benchmark task in [1]. The goal of this task is to learn the filter $\mathscr{F}$

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

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

$$\begin{aligned} u(t) - 1.99u(t-\Delta) + 1.572u(t-2\Delta) - 0.4583u(t-3\Delta) = \\ = 0.0154x(t) + 0.0462x(t-\Delta) + 0.0462x(t-2\Delta) + 0.0154x(t-3\Delta) \;, \end{aligned} \tag{5}$$

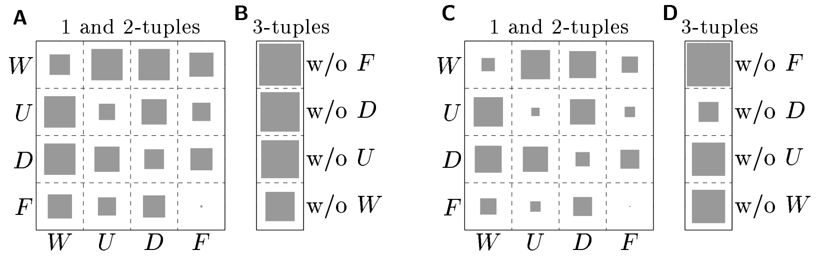

Figure 4: Impact of different synaptic parameters on the learning capabilities of a dynamic network. The size of a square (the "impact") is proportional to the inverse of the mean squared error averaged over N trials. **A** In each trial ($N = 100$) a different quadratic filter matrix H ($m = 6$) was randomly generated as described in Fig. 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 panel A but the results shown ($N = 20$) are for the filter used in [1] (Eq. (4) and Eq. (5)). **D** Same interpretation as for panel B but the results shown ($N = 20$) are for the same filter as in panel C.

for some time delay Δ . Hence, $u(t)$ is the output of a linear filter applied to the input $x(t)$.

The result is summarized in Fig. 3. It can clearly be seen that our network model (see Fig. 2A for the network architecture) is able to learn this particular filter. The mean square error (mse) on the test data is 0.0010, which is slightly smaller than the mse of 0.0013 reported in [1]. Note that the network Back and Tsoi used to learn the task had 130 adjustable parameters (13 parameters per IIR synapse, 10 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).

This shows that a very simple feedforward 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.

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 to change synaptic dynamics between pairs of layer 5 cortical neurons [8] but not in the hippocampus [20]. We therefore wondered whether plasticity in the synaptic dynamics is essential for a dynamic network 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.

Fig. 4 shows that changing only the postsynaptic parameter W has comparable impact to

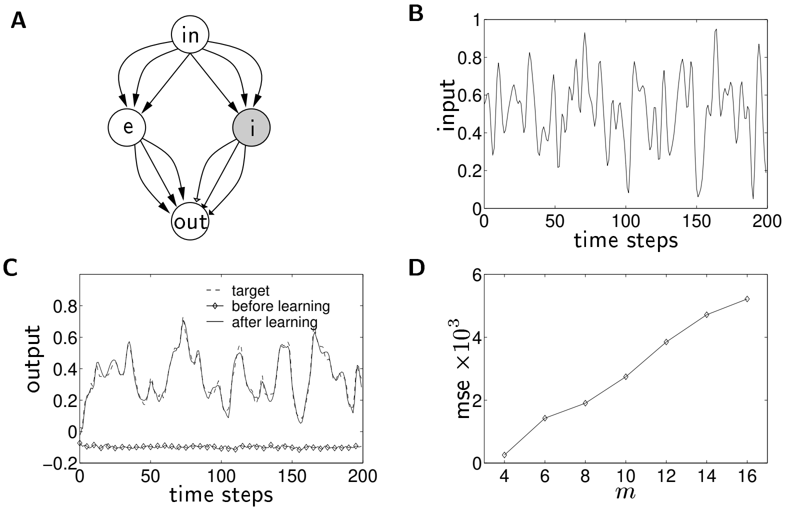

Figure 5: Dynamic synapses can substitute for neurons. **A** Network architecture. In contrast to the network in Fig. 2, this network had only 2 hidden units (one excitatory and one inhibitory), but with higher synapse multiplicity (5 synapses/axon). **B** One of the input patterns used in the training 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 dynamic network *after* learning (——). The target (− − −) was the output of the quadratic filter given by Eq. (1), the coefficients h_{kl} $(1 \leq k, l \leq 10)$ 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 H (H is a symmetric $m \times m$ matrix) we plotted the average performance (mse measured on a test set) over 20 different filters $\mathscr{Q}$, i.e. 20 randomly generated matrices H.

changing only the presynaptic parameters U or D, whereas changing only F has little impact on the dynamics of these networks (see diagonal of Fig. 4A and Fig. 4C). 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 [21], in the neocortex and elsewhere the multiplicity is often higher, so that a single presynaptic axon makes several independent contact with the postsynaptic target [22]. We therefore tested a modified architecture in which each axon made several synapses. The parameters at each synapse were modified independently.

Fig. 5 shows a limiting case an architecture with high synapse multiplicity. The number of plastic synapses is the same as in Fig. 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 Fig. 2 (compare Fig. 2D and Fig. 5D), 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 maximize wiring economy [23] might favor synapse multiplicity.

7 A Universal Approximation Theorem for Dynamic Networks

In the preceding sections we had presented empirical evidence for the approximation capabilities of our dynamic network model for computations in the time series domain. This gives rise to the question, what the theoretical limits of their approximation capabilities are. 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 dynamic networks, 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 $\mathscr{F}$ 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 $\mathscr{F}$ has fading memory if and only if the value of $\mathscr{F}\underline{x}(0)$ can be approximated arbitrarily closely by the value of $\mathscr{F}\underline{\tilde{x}}(0)$ 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

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

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 [24] and engineering [25].

Theorem 1 *Assume that X is the class of functions from $\mathbb{R}$ into $[B_0, B_1]$ which satisfy $|x(t) - x(s)| \leq B_2 \cdot |t-s|$ 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 $\mathscr{F}$ 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) $\mathscr{F}$ can be approximated by dynamic networks $\mathscr{N}$ defined by Eq. (2) and (6) to (9) (i.e., for any $\varepsilon > 0$ there exists such network $\mathscr{N}$ such that $|\mathscr{F}\underline{x}(t) - \mathscr{N}\underline{x}(t)| < \varepsilon$ for all $\underline{x} \in X^n$ and all $t \in \mathbb{R}$)

(b) $\mathscr{F}$ can be approximated by dynamic networks according to Eq. (2) and (6) to (9) with just a single layer of sigmoidal neurons

(c) $\mathscr{F}$ is time invariant and has fading memory

(d) $\mathscr{F}$ 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 [26].

The *universal approximation result* contained in Theorem 1 turns out to be rather robust with regard to changes in the definition of a dynamic network. Dynamic networks with just one layer of dynamic synapses and one subsequent layer of sigmoidal gates can approximate the same class of filters as dynamic networks with an arbitrary number of layers of dynamic synapses and sigmoidal neurons.

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 [27], 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 [9], 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 dynamic networks, 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 dynamic networks 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 dynamic networks on tasks such as the learning of a randomly chosen quadratic filter, as well as on the system identification task used in [1], to illustrate the potential of this architecture. The results are very encouraging.

Acknowledgements

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 work was supported by the Salk Sloan Foundation for Theoretical Neuroscience, project P12153 of the Fonds zur Förderung wissenschaftlicher Forschung, and the NeuroCOLT project of the EC.

References

[1] A. D. Back and A. C. Tsoi. A simplified gradient algorithm for IIR synapse multilayer perceptrons. *Neural Computation*, 5:456–462, 1993.

[2] R.C. Reid, J.D. Victor, and R.M. Shapley. The use of m-sequences in the analysis of visual neurons: linear receptive field properties. *Visual Neuroscience.*, 14:1015–27, 1997.

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

[4] R.C. deCharms and M.M. Merzenich. Optimizing sound features for cortical neurons. *Science*, 280:1439–43, 1998.

[5] T. J. Sejnowski. Statistical constraints on synaptic plasticity. *J. Theor. Biol.*, 69:385–389, 1977.

[6] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, 79:2554–2558, April 1982.

[7] K. Magleby. Short term synaptic plasticity. In G. M. Edelman, W. E. Gall, and W. M. Cowan, editors, *Synaptic function*. Wiley, New York, 1987.

[8] H. Markram and M. Tsodyks. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 382:807–10, 1996.

[9] L.F. Abbott, J.A. Varela, K.A. Sen, and Nelson S.B. Synaptic depression and cortical gain control. *Science*, 275:220–4, 1997.

[10] A. Zador and L.E. Dobrunz. Dynamic synapses in the cortex. *Neuron*, 19:1–4, 1997.

[11] L. Dobrunz and C. F. Stevens. Response of hippocampal synapses to natural stimulation patterns. *Neuron*, 22:157–66, 1999.

[12] W.A. Little and G.L. Shaw. A statistical theory of short and long term memory. *Behavioural Biology*, 14:115–33, 1975.

[13] M.V. Tsodyks, K. Pawelzik, and H. Markram. Neural networks with dynamic synapses. *Neural Computation*, 10:821–35, 1998.

[14] J-S. Liaw and T.W. Berger. Dynamic synapse: A new concept of neural representation and computation. *Hippocampus*, 6:591–600, 1996.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, editors. *Numerical Recipes in C*, chapter 10, pages 394–455. Cambridge University Press, 1992.

[16] John Hertz, Anders Krogh, and Richard Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.

[17] H. Markram, Y. Wang, and M. Tsodyks. Differential signaling via the same axon of neocortical pyramidal neurons. *PNAS*, 95:5323–8, 1998.

[18] W. Maass and A. Zador. Dynamic stochastic synapses as computational units. *Neural Computation*, 11:903–917, 1999.

[19] T. Natschläger. *Efficient Computation in Networks of Spiking Neurons – Simulations and Theory*. PhD thesis, Graz University of Technology, 1999. Accessible via http at `www.neurocolt.com` as NeuroCOLT2 Technical Report 1999-050.

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

[21] K.M. Harris and J.K. Stevens. Dendritic spines of CA1 pyramical cells in the rat hippocampus: Serial electron microscopy with reference to their biophyscial characteristics. *J. Neurosci.*, 9:2982–2997, 1989.

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

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

[24] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek. *Spikes: Exploring the neural code*. MIT Press, 1997.

[25] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. Wiley, New York, 1980.

[26] WE. Maass and E. D. Sontag. Neural systems as nonlinear filters. in press. Online available as 107 from http://www.tu-graz.ac.at/igi/maass, 2000.

[27] D.V. Buonomano and M.M. Merzenich. Temporal information transformed into a spatial code by a neural network with realistic properties. *Science*, 267:1028–30, 1995.

A Single Synapse Model

The model is described in detail in [13]. For convenience we restate the equations in our notation, which read as follows:

$$p_{ij}(t) = f_{ij}(t) \cdot d_{ij}(t) \tag{6}$$

$$\frac{d\bar{f}_{ij}(t)}{dt} = -\frac{\bar{f}_{ij}(t)}{F_{ij}} + U_{ij} \cdot (1 - \bar{f}_{ij}(t)) \cdot x_j(t) \tag{7}$$

$$\frac{dd_{ij}(t)}{dt} = \frac{1 - d_{ij}(t)}{D_{ij}} - p_{ij}(t) \cdot x_j(t) \tag{8}$$

$$f_{ij}(t) = \bar{f}_{ij}(t) \cdot (1 - U_{ij}) + U_{ij} \tag{9}$$

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 above. In this setting we consider the dynamics

$$p_{ij}(t) = f_{ij}(t) \cdot d_{ij}(t) \tag{10}$$

$$\bar{f}_{ij}(t+1) = \bar{f}_{ij}(t) - \frac{\bar{f}_{ij}(t)}{F_{ij}} + U_{ij} \cdot (1 - \bar{f}_{ij}(t)) \cdot x_j(t) \tag{11}$$

$$d_{ij}(t+1) = d_{ij}(t) + \frac{1 - d_{ij}(t)}{D_{ij}} - f_{ij}(t) \cdot d_{ij}(t) \cdot x_j(t) \tag{12}$$

$$f_{ij}(t) = \bar{f}_{ij}(t) \cdot (1 - U_{ij}) + U_{ij} \tag{13}$$

with the initial conditions $\bar{f}_{ij}(1) = 0$ (i.e. $f_{ij}(1) = U_{ij}$) and $d_{ij}(1) = 1$.