# PROCESSING OF TIME SERIES BY NEURAL CIRCUITS WITH BIOLOGICALLY REALISTIC SYNAPTIC DYNAMICS

**T. Natschläger [1], W. Maass [1], E. D. Sontag [2], and A. Zador [3]**

## 1 Introduction

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.

In this article we explore the consequences that this synaptic dynamics entails for the computational power of feedforward neural networks for computations on time series in the context of population coding.

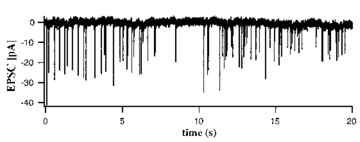
We present a rigorous theoretical result, which states that there are basically no a priori limits for the computational power of such feedforward neural networks, i.e. they can approximate arguably every filter that is potentially useful for a biological organism.

However, the theoretical analysis does not address the question how large such feedforward neural network has to be in order to approximate a given filter. We have investigated this problem empirically for the case of approximating given quadratic filters.

In addition we have studied the question of learning in the context of neural networks with dynamic synapses. In this case not only the synaptic weights known from artificial neural networks but also *parameters that govern the dynamics of a synapse are subject to adaptation*, which obviously has substantial impact on the design and performance of learning algorithms.

## The synapse model

*Biological synapses behave like small dynamical systems* (not like static weights).




Data from [Varela et al., 1997].

The dynamics of a synapse $\langle ij \rangle$ is usually described by a few *characteristic parameters*, e.g. the model of [Markram et al., 1998]:

$U_{ij} \approx$ initial release probability
$F_{ij} =$ time constant for recovery from facilitation
$D_{ij} =$ time constant for recovery from depression

Here we consider a *continuum model* which is related to several proposed previously [Varela et al., 1997, Markram et al., 1998, Tsodyks et al., 1998]. The synaptic strength $w_{ij}(t)$ at time $t$ time is modeled by

$$w_{ij}(t) = W_{ij} \cdot x_j(t) \cdot p_{ij}(t) \; .$$

$W_{ij}$ = static scale factor corresponding to the synaptic "potency"
$x_j(t)$ = activity of the $j^{th}$ presynaptic unit at time $t$
$p_{ij}(t) \approx$ current release probability at time $t$ modeled as the product of a facilitation and a depression term

The detailed equations read as follows:

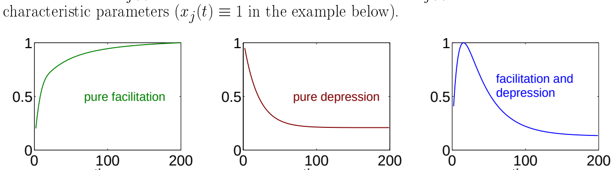
$$\begin{aligned} p_{ij}(t) &= f_{ij}(t) \cdot d_{ij}(t) \\ \frac{d\bar{f}_{ij}(t)}{dt} &= -\frac{\bar{f}_{ij}(t)}{F_{ij}} + U_{ij} \cdot (1 - \bar{f}_{ij}(t)) \cdot x_j(t) \\ \frac{d\, d_{ij}(t)}{dt} &= \frac{1 - d_{ij}(t)}{D_{ij}} - p_{ij}(t) \cdot x_j(t) \\ f_{ij}(t) &= \bar{f}_{ij}(t) \cdot (1 -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

For the numerical results we consider a *time discrete version*:

$$\begin{aligned} \bar{f}_{ij}(t+1) &= \bar{f}_{ij}(t) - \frac{\bar{f}_{ij}(t)}{F_{ij}} + U_{ij} \cdot (1 - \bar{f}_{ij}(t)) \cdot x_j(t) \\ d_{ij}(t+1) &= d_{ij}(t) + \frac{1 - d_{ij}(t)}{D_{ij}} - f^{+}_{ij}(t) \cdot d_{ij}(t) \cdot x_j(t) \end{aligned}$$

with $d_{ij}(0) = 1$ and $\bar{f}_{ij}(0) = 0$. $\bar{f}_{ij}(t)$ models facilitation (with time constant $F_{ij}$), whereas $d_{ij}(t)$ models the combined effects of synaptic depression (with time constant $D_{ij}$) and facilitation.

The *same input* $x_j(t)$ can yield *markedly different outputs* $w_{ij}(t)$ for different values of the characteristic parameters ($x_j(t) \equiv 1$ in the example below).



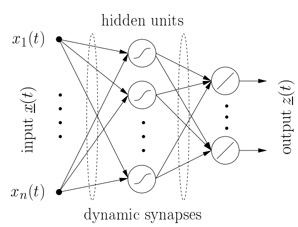

## 2 Our work

- We use a supervised learning algorithm (based on conjugate gradient methods) to train a small neural circuit to approximate a fully specified *filter*, i.e. a system which maps input time series to output time series.
- We consider feed-forward networks of sigmoidal units coupled by dynamic synapses, called *dynamic networks* in the following.
- Within that framework we address several questions:

a) *Can a dynamic network learn to approximate any filter from a given class of filters?*
b) *How large must a dynamic network be to approximate any filter from a given class of filters?*
c) *How does a dynamic network relate to artificial neural networks?*
d) *Which synaptic parameters matter?*

- Furthermore we give a precise mathematical characterization of the class of filters that can be approximated by dynamic networks.

## The dynamic network model




The output $x_i(t)$ of the $i^{th}$ unit is given by

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

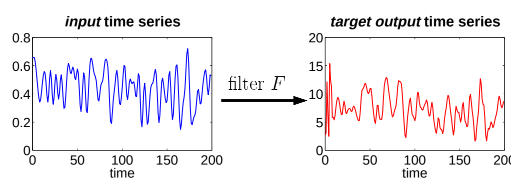
where $\sigma$ is either the sigmoid function $\sigma(u) = 1/(1 + \exp(-u))$ (in the hidden layers) or just the identity function $\sigma(u) = u$ (in the output layer).

## The learning algorithm

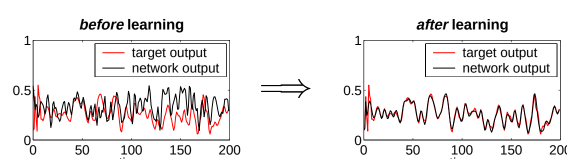
The synaptic parameters $W_{ij}$, $D_{ij}$, $F_{ij}$ and $U_{ij}$ are chosen so that, for each pair of input/output times series in the training set, the network minimized the mean-square error

$$E[z, z_F] = \frac{1}{N} \sum_{t=0}^{N-1} (z(t) - z_F(t))^2$$

between the network output $z(t)$ and the desired output $z_F(t) = (Fx)(t)$ specified by the target filter $F$.




To achieve this minimization, *we use a conjugate gradient algorithm* (see e.g. [Hertz et al., 1991]);




In order to apply such a conjugate gradient algorithm ones has to calculate the partial derivatives $\frac{\delta E[z,z_F]}{\delta U_{ij}}$, $\frac{\delta E[z,z_F]}{\delta D_{ij}}$, $\frac{\delta E[z,z_F]}{\delta F_{ij}}$ and $\frac{\delta E[z,z_F]}{\delta W_{ij}}$ for all synapses $\langle ij \rangle$ in the network.

{tnatschl, maass}@igi.tu-graz.ac.at

[2] Department of Mathematics, Rutgers University
sontag@hilbert.rutgers.edu

[3] Cold Spring Harbor Laboratory
zador@cshl.org

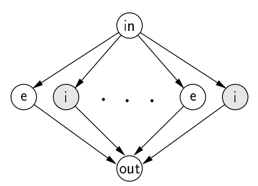
# 3 Learning Experiments

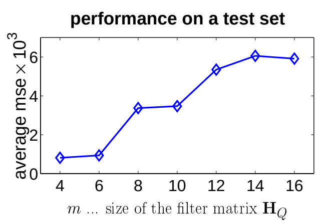
### Learning filters from a class of quadratic filters

We consider the class $\mathcal{Q}_m$ of quadratic filters $Q$ whose output $(Qx)(t)$ in response to the input time series $x(t)$ is defined by some symmetric $m \times m$ matrix $\mathbf{H}_Q = [h_{kl}]$ of filter coefficients $h_{kl} \in \mathbb{R}$, $k = 1 \ldots m$, $l = 1 \ldots m$.

$$(Qx)(t) = \sum_{l=1}^{m} \sum_{k=1}^{m} h_{kl}\, x(t-k)\, x(t-l) \; .$$

A small network with 10 hidden units (5 excitatory, 5 inhibitory) can learn (all parameters $W_{ij}$, $U_{ij}$, $D_{ij}$, and $F_{i}j$ adapted) to approximate a randomly chosen filter $Q \in \mathcal{Q}_m$ ($m = 2 \ldots 16$). The coefficients $h_{kl}$ were generated randomly by subtracting $\mu/2$ from a random number generated from an exponential distribution with mean $\mu = 3$.



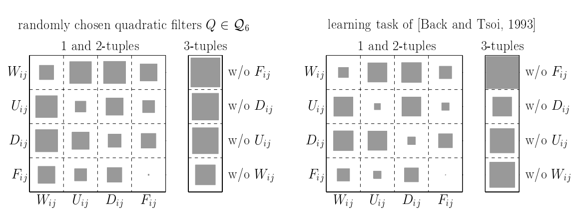




### Comparison with the model of Back and Tsoi (BT)

We have analyzed the performance of our dynamic network model (DN) for the same learning task as in [Back and Tsoi, 1993]. The goal of this task is to learn a filter $F$ with $(Fx)(t) = \sin(u(t))$ where $u(t)$ is the output of a linear filter applied to the input time series $x(t)$.


### Which parameters matter?

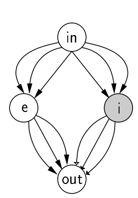
We compared network performance when different parameter subsets were optimized using the conjugate gradient algorithm, while the other parameters were held fixed. In all experiments, the fixed parameters were chosen to ensure heterogeneity in presynaptic dynamics.

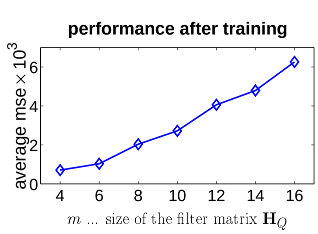



The area of a square (the "learning impact") is proportional to the inverse of the mean squared error averaged over $N = 100$ trials.

### Multiple neurons and multiple synapses

To address the question whether more synapses can replace neurons with little loss of computational power we tested a modified architecture with just two hidden units in which each axon made several (5) synapses (see "Learning filters from a class of quadratic filters" for details).




# 4 A universal approximation theorem

**Theorem** *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 $F$ 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) $F$ can be approximated by dynamic networks*

*(b) $F$ can be approximated by dynamic networks with just a single layer of sigmoidal neurons*

*(c) $F$ is time invariant and has fading memory*

*(d) $F$ can be approximated by a sequence of (finite or infinite) Volterra series.*

The *proof of this Theorem* relies on the Stone-Weierstrass Theorem, and is contained as the proof of Theorem 3.4 in [Maass and Sontag, 2000].

An arbitrary filter $F$ 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 $t_0$.

Informally speaking *a filter $F$ has fading memory* if the output at time $t$ primarily depends on inputs within a certain time interval $[t - T, t]$, i.e. it has essentially finite memory.

The class of filters that can be represented by *Volterra series* has been investigated for quite some time in neurobiology [Rieke et al., 1996]. A Volterra term of order $k$ is given by

$$y(t) = \int_0^\infty \ldots \int_0^\infty x(t - \tau_1) \cdot \ldots \cdot x(t - \tau_k) h(\tau_1, \ldots, \tau_k) d\tau_1 \ldots d\tau_k \; .$$

# 5 Summary

- **We have analyzed the computational power of *dynamic networks* (see "The dynamic network model"), which represent a new paradigm for neural computation on time series that is based on biologically realistic models for synaptic dynamics (see also [Zador, 2000]).**
- **Our analytical results show that the class of nonlinear filters that can be approximated by dynamic networks is remarkably rich. It contains every time invariant filter with fading memory, i.e. any filter that can be approximated by Volterra series.**
- **Our computer simulations show that rather small dynamic networks are able to perform interesting computations on time series.**
- **The performance of *dynamic networks* is comparable to that of previously considered artificial neural networks that were designed for the purpose of yielding efficient processing of temporal signals.**

# Online versions

**Online version of this poster:**
`http://www.igi.TUGraz.at/igi/tnatschl/psfiles/dynsyn-poster.ps.gz`

**Online version of the corresponding paper:**
`http://www.igi.TUGraz.at/igi/tnatschl/psfiles/dynsyn-nips00.ps.gz`

# References

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

[Hertz et al., 1991] Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley.

[Maass and Sontag, 2000] Maass, W. and Sontag, E. D. (2000). Neural systems as nonlinear filters. *Neural Computation*, 12(8):1743-1772.

[Markram et al., 1998] Markram, H., Wang, Y., and Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *PNAS*, 95:5323–5328.

[Rieke et al., 1997] Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W. (1997). *Spikes: Exploring the Neural Code*. MIT Press.

[Tsodyks et al., 1998] Tsodyks, M., Pawelzik, K., and Markram, H. (1998). Neural networks with dynamic synapses. *Neural Computation*, 10:821–835.

[Varela et al., 1997] Varela, J. A., Sen, K., Gibson, J., Fost, J., Abbott, L. F., and Nelson, S. B. (1997). A quantitative description of short-term plasticity at excitatory synapses in layer 2/3 of rat primary visual cortex. *J. Neurosci*, 17(20):7926-7940.

[Zador, 2000] Zador, A. M. (2000). The basic unit of computation. *Nature Neuroscience*, 3(Supp):1167.