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 prior 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

The dynamics of a synapse \( w_{ij}(t) \) is usually described by a few characteristic parameters, e.g., the model of [Munro et al., 1993]:

- \( U_{ij} \) = initial release probability
- \( F_{ij} \) = time constant for recovery from facilitation
- \( D_{ij} \) = time constant for recovery from depression

Here we consider a continuous model which is related to several proposed previously [Norde et al., 1997, Munro et al., 1998, Tsuchiya et al., 1998]. The synaptic strength \( w_{ij}(t) \) at time \( t \) is time modeled by:

\[
\frac{dw_{ij}(t)}{dt} = W_{ij} \cdot x_j(t) \cdot p_{ij}(t) - W_{ij} \cdot x_j(t) \cdot p_{ij}(t) \cdot \frac{1}{1 + e^{-t/T}}
\]

\( W_{ij} \) = static scale factor corresponding to the synaptic “potency"
\( x_j(t) \) = activity of the \( j \)-th presynaptic unit at time \( t \)
\( p_{ij}(t) \) = release probability at time \( t \)

modeled as the product of a facilitation and a depression term.

The model equations may be written as:

\[
\begin{align*}
\frac{dx_j(t)}{dt} &= \frac{1}{T_j} \left[ x_j(t) - F_j(t) \right] + \frac{1}{T_j} \left[ d_j(t) - x_j(t) \right] \\
\frac{dx_j(t)}{dt} &= -\frac{1}{T_j} \left[ x_j(t) - F_j(t) \right] + \frac{1}{T_j} \left[ d_j(t) - x_j(t) \right]
\end{align*}
\]

where \( d_j(t) = 1 - \frac{x_j(t)}{F_j(t)} \) models facilitation with time constant \( F_j \), whereas \( d_j(t) \) models the combined effects of synaptic depression with time constant \( D_j \) and facilitation.

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

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 model network

The output \( x_j(t) \) of the \( j \)-th unit is given by

\[
x_j(t) = \frac{1}{N} \sum_{k=1}^{N} \left( w_{kj} \cdot p_{kj}(t) \cdot x_j(t) \right)
\]

between the network output \( z(t) \) and the desired output \( x_j(t) \) specified by the target filter \( F \).

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 time series in the training set, the network minimizes the mean-square error

\[
E[z(t) - z_j(t)]^2 = \frac{1}{N} \sum_{k=1}^{N} (z_k(t) - z_k(t))^2
\]

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 one has to calculate the partial derivatives \( \frac{dE}{dW_{ij}}, \frac{dE}{dU_{ij}}, \frac{dE}{dD_{ij}}, \frac{dE}{dF_{ij}} \) for all synapses \( (i,j) \) in the network.

The output time series

The input time series

Before learning

Target output time series

After learning

Network output

Network output

Target output

Network output

Network output

Network output

Network output

Network output
3 Learning Experiments

Learning filters from a class of quadratic filters

We consider the class of quadratic filters $Q$ whose output $Q(x)$ is defined as a quadratic function of the input time series $x(t)$, i.e., $Q(x) = [h_0, h_1] x(t - h_1) + [h_2, \ldots, h_m] x(t - h_m)$. The coefficients $h_k$ are chosen randomly by subtracting $1/2$ from a random number generated from a normal distribution with mean $\mu = 0$.

\[ Q(x) = \sum_{k=0}^{m} h_k x(t - k) \cdot x(t - k) \]

A small network with 16 hidden units (8 excitatory, 8 inhibitory) can learn all parameters ($W$, $U$, $D$, and $F$) to approximate a randomly-chosen filter $Q \in Q_n$ ($n = 2, \ldots, 16$). The coefficients $h_k$ were generated randomly by subtracting $1/2$ from a random number generated from a normal distribution with mean $\mu = 0$.

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 the task is to learn a filter $F$ with $(P(x)) = \sin(x)$, where $x$ is the output of a linear filter applied to the input time series $z(t)$.

\[ y(t) = \int_{t-\tau}^{t} z(t) \cdot x(t - \tau) \cdot x(t - 2\tau) \cdot \ldots \cdot x(t - m\tau) \cdot \ldots \cdot d\tau \]

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 synaptic dynamics.

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 (a) two hidden units in which each neuron made several synapses (see “Learning filters from a class of quadratic filters” for details).

4 A universal approximation theorem

Theorem Assume that $Q \in Q_n$ is a function of $\mathbb{R}$ into $\mathbb{R}$ that satisfies $|Q(x)| \leq M$ on $\mathbb{R}$, for all $x \in \mathbb{R}$, where $M, B_1, B_2, \ldots, B_n$ are arbitrary real-valued constants and $0 < \epsilon < B_n$. Let $P$ be an arbitrary filter that maps vectors of functions $x(t) = [x_1(t), \ldots, x_n(t)]$ into functions from $\mathbb{R}^n$ into $\mathbb{R}$. Then the following are equivalent:

(a) $P$ can be approximated by dynamic networks
(b) $P$ can be approximated by dynamic networks with just a single layer of sigmoidal neurons
(c) $P$ is time invariant and has fading memory
(d) $P$ can be approximated by a sequence of (finite or infinite) Volterra series.

The proof of the Theorem relies on the Stone-Weierstrass Theorem, and is contained in the proof of Theorem 1.4 in Marcus and Singer, 2000.

An arbitrary filter $P$ is called time invariant if a shift of the input functions by a constant $\tau$ does not cause a shift of the output function by the same constant $\tau$.

Informally speaking, a filter $P$ has fading memory if the output at time $t$ primarily depends on inputs within a certain interval $[t - \tau, t]$, 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 (Bieße et al., 1996). A Volterra series of order $K$ is given by

\[ y(t) = \sum_{\ell=0}^{K} \int_{t-\tau}^{t} x(t) \cdot x(t - 1) \cdot x(t - 2) \cdot \ldots \cdot x(t - \ell\tau) \cdot \ldots \cdot d\tau \]

5 Summary

- We 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.ig:70000/ig/lsat11/psf1/1ps/dyncomp0908ps.ps.gz

Online version of the corresponding paper:
http://www.ig:70000/ig/lsat11/psf1/1ps/dyncomp1000-ps.ps.gz

References