Reinforcement Learning Predicts the Site of Plasticity for Auditory Remapping in the Barn Owl

Alexandre Pouget† alex@salk.edu
Cedric Deffayet‡ cedric@salk.edu
Terrence J. Sejnowski† terry@salk.edu
†Howard Hughes Medical Institute
The Salk Institute
La Jolla, CA 92037
Department of Biology
University of California, San Diego
and
‡Ecole Normale Superieure
45 rue d'Ulm
75005 Paris, France

Abstract

The auditory system of the barn owl contains several spatial maps. In young barn owls raised with optical prisms over their eyes, these auditory maps are shifted to stay in register with the visual map, suggesting that the visual input imposes a frame of reference on the auditory maps. However, the optic tectum, the first site of convergence of visual with auditory information, is not the site of plasticity for the shift of the auditory maps; the plasticity occurs instead in the inferior colliculus, which contains an auditory map and projects into the optic tectum. We explored a model of the owl remapping in which a global reinforcement signal whose delivery is controlled by visual foveation. A hebb learning rule gated by reinforcement learned to appropriately adjust auditory maps. In addition, reinforcement learning preferentially adjusted the weights in the inferior colliculus, as in the owl brain, even though the weights were allowed to change throughout the auditory system. This observation raises the possibility that the site of learning does not have to be genetically specified, but could be determined by how the learning procedure interacts with the network architecture.

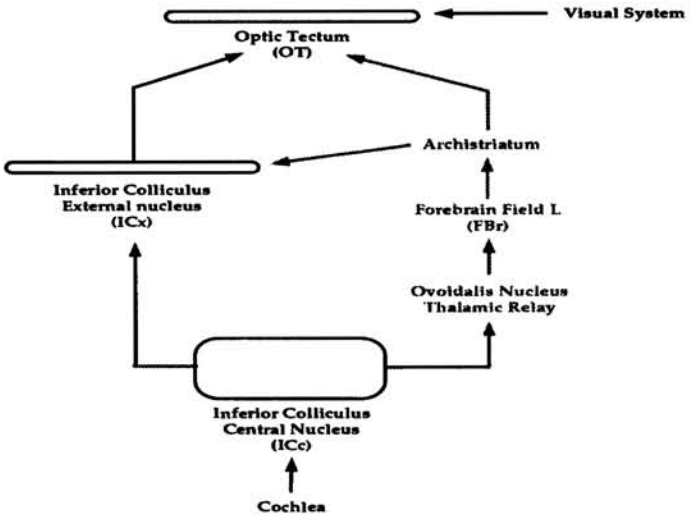

Figure 1: Schematic view of the auditory pathways in the barn owl.

1 Introduction

The barn owl relies primarily on sounds to localize prey [6] with an accuracy vastly superior to that of humans. Figure 1A illustrates some of the nuclei involved in processing auditory signals. The barn owl determines the location of sound sources by comparing the time and amplitude differences of the sound wave between the two ears. These two cues are combined together for the first time in the shell and core of the inferior colliculus (ICc) which is shown at the bottom of the diagram. Cells in the ICc are frequency tuned and subject to spatial aliasing. This prevents them from unambiguously encoding the position of objects. The first unambiguous auditory map is found at the next stage, in the external capsule of the inferior colliculus (ICx) which itself projects to the optic tectum (OT). The OT is the first subforebrain structure which contains a multimodal spatial map in which cells typically have spatially congruent visual and auditory receptive fields.

In addition, these subforebrain auditory pathways send one major collateral toward the forebrain via a thalamic relay. These collaterals originate in the ICc and are thought to convey the spatial location of objects to the forebrain [3]. Within the forebrain, two major structures have been involved in auditory processing: the archistriatum and field L. The archistriatum sends a projection to both the inferior colliculus and the optic tectum.

Knudsen and Knudsen (1985) have shown that these auditory maps can adapt to systematic changes in the sensory input. Furthermore, the adaptation appears to be under the control of visual input, which imposes a frame of reference on the incoming auditory signals. In owls raised with optical prisms, which introduce a systematic shift in part of the visual field, the visual map in the optic tectum was identical to that found in control animals, but the auditory map in the ICx was shifted by the amount of visual shift introduced by the prisms. This plasticity ensures that the visual and auditory maps stay in spatial register during growth

and other perturbations to sensory mismatch.

Since vision instructs audition, one might expect the auditory map to shift in the optic tectum, the first site of visual-auditory convergence. Surprisingly, Brainard and Knudsen (1993b) observed that the synaptic changes took place between the ICc and the ICx, one synapse before the site of convergence.

These observations raise two important questions: First, how does the animal knows how to adapt the weights in the ICx in the absence of a visual teaching signal? Second, why does the change take place at this particular location and not in the OT where a teaching signal would be readily available?

In a previous model [7], this shift was simulated using backpropagation to broadcast the error back through the layers and by constraining the weights changes to the projection from the ICc to ICx. There is, however, no evidence for a feedback projection between from the OT to the ICx that could transmit the error signal; nor is there evidence to exclude plasticity at other synapses in these pathways.

In this paper, we suggest an alternative approach in which vision guides the remapping of auditory maps by controlling the delivery of a scalar reinforcement signal. This learning proceeds by generating random actions and increasing the probability of actions that are consistently reinforced [1, 5]. In addition, we show that reinforcement learning correctly predicts the site of learning in the barn owl, namely at the ICx-ICc synapse, whereas backpropagation [8] does not favor this location when plasticity is allowed at every synapse. This raises a general issue: the site of synaptic adjustment might be imposed by the combination of the architecture and learning rule, without having to restrict plasticity to a particular synapse.

2 Methods

2.1 Network Architecture

The network architecture of the model based on the barn owl auditory system, shown in figure 2A, contains two parallel pathways. The input layer was an 8x21 map corresponding to the ICc in which units responded to frequency and interaural phase differences. These responses were pooled together to create auditory spatial maps at subsequent stages in both pathways. The rest of the network contained a series of similar auditory maps, which were connected topographically by receptive fields 13 units wide. We did not distinguish between field L and the archistriatum in the forebrain pathways and simply used two auditory maps, both called FBr.

We used multiplicative (sigma-pi) units in the OT whose activities were determined according to:

$$y_i = \sum_j w_{ij}^{FBr} y_j^{FBr} w_{jk}^{FBr} y_j^{ICx} \tag{1}$$

The multiplicative interaction between ICx and FBr activities was an important assumption of our model. It forced the ICx and FBr to agree on a particular position before the OT was activated. As a result, if the ICx-OT synapses were modified during learning, the ICx-FBr synapses had to be changed accordingly.

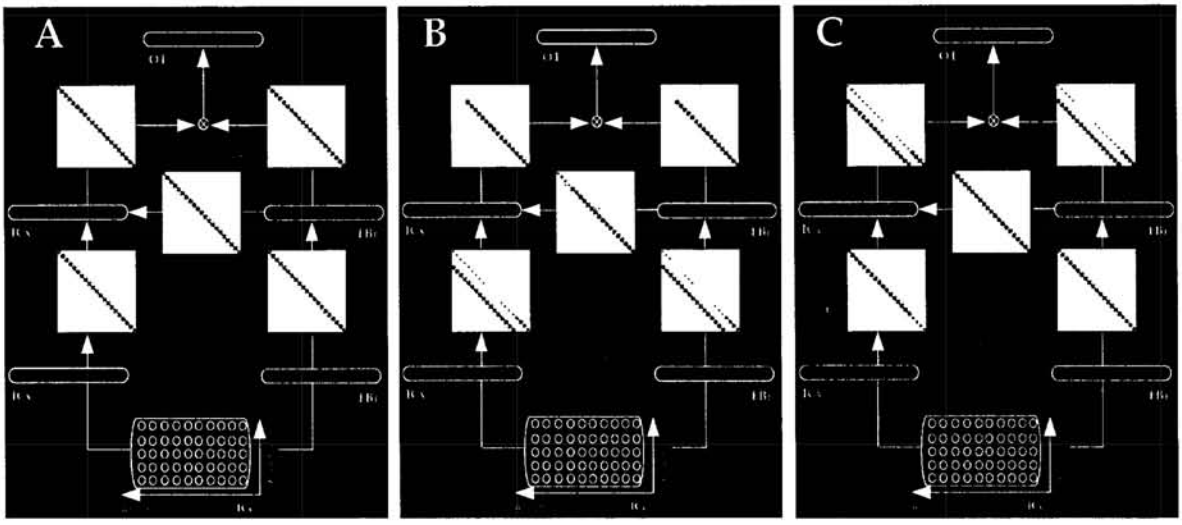

Figure 2: Schematic diagram of weights (white blocks) in the barn owl auditory system. A) Diagram of the initial weights in the network. B) Pattern of weights after training with reinforcement learning on a prism-induced shift of four units. The remapping took place within the ICx and FBr. C) Pattern of weights after training with backpropagation. This time the ICx-OT and FBr-OT weights changed.

Weights were clipped between 5.0 and 0.01, except for the FBr-ICx connections whose values were allowed to vary between 8.0 and 0.01. The minimum values were set to 0.01 instead of zero to prevent getting trapped in unstable local minima which are often associated with weights values of zero. The strong coupling between FBr and ICx was another important assumption of the model whose consequence will be discussed in the last section.

Examples were generated by simply activating one unit in the ICc while keeping the others to zero, thereby simulating the pattern of activity that would be triggered by a single localized auditory stimulus. In all simulations, we modeled a prism-induced shift of four units.

2.2 Reinforcement learning

We used stochastic units and trained the network using reinforcement learning [1]. The weighted sum of the inputs, net_i, passed through a sigmoid, f(x), is interpreted as the probability, p_i, that the unit will be active:

$$p_i = f(net_i) * 0.99 + 0.01 \tag{2}$$

were the output of the unit y_i was:

$$y_i = \begin{cases} 0 & \text{with probability } 1 - p_i \\ 1 & \text{with probability } p_i \end{cases} \tag{3}$$

Because of the form of the equation for p_i, all units in the network had a small probability (0.01) of being spontaneously active in the absence of any inputs. This is what allowed the network to perform a stochastic search in action space to find which actions were consistently associated with positive reinforcement.

We ensured that at most one unit was active per trial by using a winner-take-all competition in each layer.

Adjustable weights in the network were updated after each training examples with hebb-like rule gated by reinforcement:

$$\Delta w_{ij} = \epsilon y_i \frac{\partial net_i}{\partial w_{ij}} r \tag{4}$$

A trial consisted in choosing a random target location for auditory input (ICc) and the output of the OT was used to generate a head movement. The reinforcement, r, was then set to 1 for head movements resulting in the foveation of the stimulus and to -0.05 otherwise.

2.3 Backpropagation

For the backpropagation network, we used deterministic units with sigmoid activation functions in which the output of a unit was given by:

$$y_i = f(net_i) \tag{5}$$

where net_i is the weighted sum of the inputs as before.

The chain rule was used to compute the partial derivatives of the squared error, E, with respect to each weights and the weights were updated after each training example according to:

$$\Delta w_{ij} = -\epsilon \frac{\partial E}{\partial w_{ij}} \tag{6}$$

The target vectors were similar to the input vectors, namely only one OT units was required to be activated for a given pattern, but at a position displaced by 4 units compared to the input.

3 Results

3.1 Learning site with reinforcement

In a first set of simulation we kept the ICc-ICx and ICc-FBr weights fixed. Plasticity was allowed at these site in later simulations.

Figure 2A shows the initial set of weights before learning starts. The central diagonal lines in the weight diagrams illustrate the fact that each unit receives only one non-zero weight from the unit in the layer below at the same location.

There are two solutions to the remapping: either the weights change within the ICx and FBr, or from the ICx and the FBr to the OT. As shown in figure 2B, reinforcement learning converged to the first solution. In contrast, the weights between the other layers were unaltered, even though they were allowed to change.

To prove that the network could have actually learned the second solution, we trained a network in which the ICc-ICx weights were kept fixed. As we expected, the network shifted its maps simultaneously in both sets of weights converging onto the OT, and the resulting weights were similar to the ones illustrated in figure 2C. However, to reach this solution, three times as many training examples were needed.

The reason why learning in the ICx and FBr were favored can be attributed to probabilistic nature of reinforcement learning. If the probability of finding one solution is p, the probability of finding it twice independently is p^2. Learning in the ICx and FBR is not independent because of the strong connection from the FBr to the ICx. When the remapping is learned in the FBR this connection automatically remapped the activities in the ICx which in turn allows the ICx-ICx weights to remap appropriately. In the OT on the other hand, the multiplicative connection between the ICx and FBr weights prevent a cooperation between this two sets of weights. Consequently, they have to change independently, a process which took much more training.

3.2 Learning at the ICc-ICx and ICc-FBr synapses

The aliasing and sharp frequency tuning in the response of ICc neurons greatly slows down learning at the ICc-ICx and ICc-FBr synapses. We found that when these synapses were free to change, the remapping still took place within the ICx or FBr (figure 3).

3.3 Learning site with backpropagation

In contrast to reinforcement learning, backpropagation adjusted the weights in two locations: between the ICx and the OT and between the Fbr and OT (figure 2C). This is the consequence of the tendency of the backpropagation algorithm to first change the weights closest to where the error is injected.

3.4 Temporal evolution of weights

Whether we used reinforcement or supervised learning, the map shifted in a very similar way. There was a simultaneous decrease of the original set of weights with a simultaneous increase of the new weights, such that both sets of weights coexisted half way through learning. This indicates that the map shifted directly from the original setting to the new configuration without going through intermediate shifts.

This temporal evolution of the weights is consistent the findings of Brainard and Knudsen (1993a) who found that during the intermediate phase of the remapping, cells in the inferior colliculus typically have two receptive fields. More recent work however indicates that for some cells the remapping is more continuous(Brainard and Knudsen, personal communication), a behavior that was not reproduced by either of the learning rule.

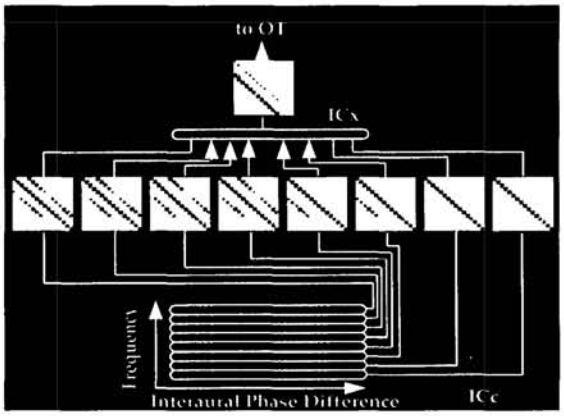

Figure 3: Even when the ICc-ICx weights are free to change, the network update the weights in the ICx first. A separate weight matrix is shown for each isofrequency map from the ICc to ICx. The final weight matrices were predominantly diagonal; in contrast, the weight matrix in ICx was shifted.

4 Discussion

Our simulations suggest a biologically plausible mechanism by which vision can guide the remapping of auditory spatial maps in the owl's brain. Unlike previous approaches, which relied on visual signals as an explicit teacher in the optic tectum [7], our model uses a global reinforcement signal whose delivery is controlled by the foveal representation of the visual system. Other global reinforcement signals would work as well. For example, a part of the forebrain might compare auditory and visual patterns and report spatial mismatch between the two. This signal could be easily incorporated in our network and would also remap the auditory map in the inferior colliculus.

Our model demonstrates that the site of synaptic plasticity can be constrained by the interaction between reinforcement learning and the network architecture. Reinforcement learning converged to the most probably solution through stochastic search. In the network, the strong lateral coupling between ICx and FBr and the multiplicative interaction in the OT favored a solution in which the remapping took place simultaneously in the ICx and FBr. A similar mechanism may be at work in the barn owl's brain. Colaterals from FBr to ICx are known to exist, but the multiplicative interaction has not been reported in the barn owl optic tectum.

Learning mechanisms may also limit synaptic plasticity. NMDA receptors have been reported in the ICx, but they might not be expressed at other synapses. There may, however, be other mechanisms for plasticity.

The site of remapping in our model was somewhat different from the existing observations. We found that the change took place *within* the ICx whereas Brainard and Knudsen [3] report that it is *between* the ICc and the ICx. A close examination of their data (figure 11 in [3]) reveals that cells at the bottom of ICx were not

remapped, as predicted by our model, but at the same time, there is little anatomical or physiological evidence for a functional and hierarchical organization within the ICx. Additional recordings are need to resolve this issue. We conclude that for the barn owl's brain, as well as for our model, synaptic plasticity within ICx was favored over changes between ICc and ICx. This supports the hypothesis that reinforcement learning is used for remapping in the barn owl auditory system.

Acknowledgments

We thank Eric Knudsen and Michael Brainard for helpful discussions on plasticity in the barn owl auditory system and the results of unpublished experiments. Peter Dayan and P. Read Montague helped with useful insights on the biological basis of reinforcement learning in the early stages of this project.

References

[1] A.G. Barto and M.I. Jordan. Gradient following without backpropagation in layered networks. *Proc. IEEE Int. Conf. Neural Networks*, 2:629–636, 1987.

[2] M.S. Brainard and E.I. Knudsen. Dynamics of the visual calibration of the map of interaural time difference in the barn owl's optic tectum. In *Society For Neuroscience Abstracts*, volume 19, page 369.8, 1993.

[3] M.S. Brainard and E.I. Knudsen. Experience-dependent plasticity in the inferior colliculus: a site for visual calibration of the neural representation of auditory space in the barn owl. *The journal of Neuroscience*, 13:4589–4608, 1993.

[4] E. Knudsen and P. Knudsen. Vision guides the adjustment of auditory localization in the young barn owls. *Science*, 230:545–548, 1985.

[5] P.R. Montague, P. Dayan, S.J. Nowlan, A. Pouget, and T.J. Sejnowski. Using aperiodic reinforcement for directed self-organization during development. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, San Mateo, CA, 1993.

[6] R.S. Payne. Acoustic location of prey by barn owls (tyto alba). *Journal of Experimental Biology*, 54:535–573, 1970.

[7] D.J. Rosen, D.E. Rumelhart, and E.I. Knudsen. A connectionist model of the owl's sound localization system. In *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, San Mateo, CA, 1994.

[8] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing*, volume 1, chapter 8, pages 318–362. MIT Press, Cambridge, MA, 1986.

Dynamic Modelling of Chaotic Time Series with Neural Networks

Jose C. Principe, Jyh-Ming Kuo
Computational NeuroEngineering Laboratory
University of Florida, Gainesville, FL32611
principe@synapse.ee.ufl.edu

Abstract

This paper discusses the use of artificial neural networks for dynamic modelling of time series. We argue that multistep prediction is more appropriate to capture the dynamics of the underlying dynamical system, because it constrains the iterated model. We show how this method can be implemented by a recurrent ANN trained with trajectory learning. We also show how to select the trajectory length to train the iterated predictor for the case of chaotic time series. Experimental results corroborate the proposed method.

1.0 Introduction

The search for a model of an experimental time series has been an important problem in science. For a long time the linear model was almost exclusively used to describe the system that produced the time series [1], but recently nonlinear models have also been proposed to replace the linear ones [2]. Lapedes and Farber [3] showed how artificial neural networks (ANNs) can be used to identify the dynamics of the unknown system that produced the time series. He simply used a multilayer perceptron to predict the next point in state space, and trained this topology with backpropagation. This paper explores more complex neural topologies and training methods with the goal of improving the quality of the identification of the dynamical system, and to understand better the issues of dynamic modelling with neural networks which are far from being totally understood.

According to Takens' embedding theorem, a map $F: R^{2m+1} \rightarrow R^{2m+1}$ exists that transforms the current reconstructed state $\hat{y}(t)$ to the next state $\hat{y}(t+1)$, i.e.

$$\hat{y}(t+1) = F(\hat{y}(t)) \qquad (1)$$

or

$$\begin{bmatrix} x(t+1) \\ \cdots \\ x(t+1-2m) \end{bmatrix} = F\left(\begin{bmatrix} x(t) \\ \cdots \\ x(t-2m) \end{bmatrix}\right)$$

where m is the estimated dimension of the unknown dynamical system Φ. Note that the map contains several trivial (nonlinear) filters and a predictor. The predictive mapping $F^{\perp}: R^{2m+1} \rightarrow R$ can be expressed as

$$x(t+1) = F^{\perp}(\hat{x}(t)) \tag{2}$$

where $\hat{x}(t) = [x(t-2m) \ldots x(t-1)\, x(t)]^T$. This is actually the estimated nonlinear autoregressive model of the input time series. *The existence of this predictive model lays a theoretical basis for dynamic modelling in the sense that we can build from a vector time series a model to approximate the mapping* $F^{\perp}$. If the conditions of Takens embedding theorem are met, this mapping captures some of the properties of the unknown dynamical system Φ that produced the time series [7].

Presently one still does not have a capable theory to guarantee if the predictor has successfully identified the original model Φ. The simple point by point comparison between the original and predicted time series used as goodness of fit for non-chaotic time series breaks down for chaotic ones [5]. Two chaotic time series can be very different pointwise but be produced by the same dynamical system (two trajectories around the same attractor). The dynamic invariants (correlation dimension, Lyapunov exponents) measure global properties of the attractor, so they should be used as the rule to decide about the success of dynamic modelling. Hence, *a pragmatic approach in dynamic modelling is to seed the predictor with a point in state space, feed the output to its input as an autonomous system, and create a new time series.* If the dynamic invariants computed from this time series match the ones from the original time series, then we say that dynamic modelling was successful [5]. The long term behavior of the autonomous predictive model seems to be the key factor to find out if the predictor identified the original model. This is the distinguishing factor between prediction of chaotic time series and dynamic modelling. The former only addresses the instantaneous prediction error, while the latter is interested in long term behavior.

In order to use this theory, one needs to address the choices of predictor implementation. Due to the universal mapping characteristics of multilayer perceptrons (MLPs) and the existence of well established learning rules to adapt the MLP coefficients, this type of network appears as an appropriate choice [3]. However, one must realize that the MLP is a static mapper, and in dynamic modelling we are dealing with time varying signals, where the past of the signal contains vital information to describe the mapping. The design considerations to select the neural network topology are presented elsewhere [4]. We just would like to say that the MLP has to be enhanced with short term memory mechanisms, and that the *estimation of the correlation dimension should be used to set the size of the memory layer.* The main goal of the paper is to establish the methodology to efficiently train neural networks for dynamic modelling.

2. Iterated versus Single Step Prediction.

From eqn. 2 it seems that the resulting dynamic model F can be obtained through single step prediction. This has been the conventional way to handle dynamic modelling [2],[3]. The predictor is adapted by minimizing the error

$$E = \sum_{i=2m+1}^{L} dist(x(i+1) - \tilde{F}^{\perp}(\hat{x}(i))) \tag{3}$$

where L is the length of the time series, x(i) is the i^{th} data sample, $\tilde{F}^{\perp}$ is the map developed by the predictor and *dist()* is a distance measure (normally the L2 norm). Notice that the training to obtain the mapping is done independently from sample to sample, i.e.

$$\begin{aligned} x(i+1) &= \tilde{F}^{\perp}(\hat{x}(i)) + \delta_1 \\ &\cdots\cdots \\ x(i+j) &= \tilde{F}^{\perp}(\hat{x}(i+j-1)) + \delta_j \end{aligned}$$

where δ_j are the instantaneous prediction errors, which are minimized during training. Notice that the predictor is being optimized under the assumption that the previous point in state space is known without error.

The problem with this approach can be observed when we iterate the predictor as an autonomous system to generate the time series samples. If one wants to produce two samples in the future from sample i the predicted sample i+1 needs to be utilized to generate sample i+2. The predictor was not optimized to do this job, because during training the true i+1 sample was assumed known. As long as δ_1 is nonzero (as will be always the case for nontrivial problems), errors will accumulate rapidly. Single step prediction is more associated with extrapolation than with dynamic modelling, which requires the identification of the unique mapping that produces the time series.

When the autonomous system generates samples, past values are used as inputs to generate the following samples, which means that the training should constrain also the iterates of the predictive mapping. Putting it in a simple way, we should train the predictor in the same way we are going to use it for testing (i.e. as an autonomous system).

We propose multistep prediction (or trajectory learning) as the way to constrain the iterates of the mapping developed by the predictor. Let us define

$$E = \sum_{i=2m+1}^{k} dist(x(i+1) - \tilde{x}(i+1)) \tag{4}$$

where k is the number of prediction steps (length of the trajectory) and $\tilde{x}(i+1)$ is an estimate of the predictive map

$$\tilde{x}(i+1) = \tilde{F}^{\perp}(\hat{x}(i-2m),\ldots,\hat{x}(i)) \tag{5}$$

with

$$\hat{x}(i) = \begin{cases} x(i) & 0 \le i \le 2m \\ \tilde{F}^{\perp}(x(i-2m-1),\ldots,x(i-1)) & i > 2m \end{cases}$$

Equation (5) states that $\tilde{x}(i)$ is the i-2m iterate of the predictive part of the map (for i>2m), i.e.

$$\tilde{x}(i+1) = (\tilde{F}^{\perp}(\tilde{F}^{\perp}(\ldots\tilde{F}^{\perp}(\hat{x}(2m))))) = (\tilde{F}^{\perp}(\hat{x}(2m)))^{i-2m} \qquad (6)$$

Hence, minimizing the criterion expressed by equation (4) an optimal multistep predictor is obtained. The number of constraints that are imposed during learning is associated with k, the number of prediction steps, which corresponds to the number of iterations of the map. The more iterations, the less likely a sub-optimal solution is found, but note that the training time is being proportionally increased. In a chaotic time series there is a more important consideration that must be brought into the picture, the divergence of nearby trajectories, as we are going to see in a following section.

3. Multistep prediction with neural networks

Figure 1 shows the topology proposed in [4] to identify the nonlinear mapping. Notice that the proposed topology is a *recurrent neural network, with a global feedback loop*. This topology was selected to allow the training of the predictor in the same way as it will be used in testing, i.e. using the previous network outputs to predict the next point. This recurrent architecture should be trained with a mechanism that will constrain the iterates of the map as was discussed above. Single step prediction does not fit this requirement.

With multistep prediction, the model system can be trained in the same way as it is used in testing. We seed the dynamic net with a set of input samples, disconnect the input and feed back the predicted sample to the input for k steps. The mean square error between the predicted and true sample at each step is used as the cost function (equation (4)). If the network topology was feedforward, batch learning could be used to train the network, and static backpropagation applied to train the net. However, as a recurrent topology is utilized, a learning paradigm such as backpropagation through time (BPTT) or real time recurrent learning (RTRL) must be utilized [6]. The use of these training methods should not come as a surprise since we are in fact fitting a trajectory over time, so the gradients are time varying. This learning method is sometimes called "trajectory learning" in the recurrent learning literature [6]. A criterion to select the length of the trajectory k will be presented below.

The procedure described above must be repeated for several different segments of the time series. For each new training segment, 2m+1 samples of the original time series are used to seed the predictor. To ease the training we suggest that successive training sequences of length k overlap by q samples (q<k). For chaotic time series we also suggest that the error be weighted according to the largest Lyapunov exponent. Hence

the cost function becomes

$$E = \sum_{j=0}^{r} \sum_{i=2m+1}^{k} h(i) \cdot dist(x(i+jq+1) - \tilde{x}(i+jq+1)) \tag{7}$$

where r is the number of training sequences, and

$$h(i) = (e^{\lambda_{max}\Delta t})^{-(i-2m-1)} \tag{8}$$

In this equation λ_{max} is the largest Lyapunov exponent and Δt the sampling interval. With this weighting the errors for later iteration are given less credit, as they should since due to the divergence of trajectories a small error is magnified proportionally to the largest Lyapunov exponent [7].

4. Finding the length of the trajectory

From the point of view of dynamic modelling, each training sequences should preferably contain enough information to model the attractor. This means that each sequence should be no shorter than the orbital length around the attractor. We proposed to estimate the orbital length as the reciprocal of the median frequency of the spectrum of the time series [8]. Basically this quantity is the average time required for a point to return to the same neighborhood in the attractor.

The length of the trajectory is also equivalent to the number of constraints we impose on the iterative map describing the dynamical model. However, in a chaotic time series there is another fundamental limitation imposed on the trajectory length - the natural divergence of trajectories which is controlled by λ_{max}, the largest Lyapunov exponent. If the trajectory length is too long, then instabilities in the training can be expected. A full discussion of this topic is beyond the scope of this paper, and is presented elsewhere [8]. We just want to say that when λ_{max} is positive there is an uncertainty region around each predicted point that is a function of the number of prediction steps (due to cummulative error). If the trajectory length is too long the uncertainty regions from two neighboring trajectories will overlap, creating conflicting requirements for training (the model is requried to develop a map to follow both segments A and B- Figure 2).

It turns out that one can approximately find the number of iterations i_s that will guarantee no overlap of uncertainty regions [8]. The length of the principal axis of the uncertainty region around a signal trajectory at iteration i can be estimated as

$$\varepsilon_i = \varepsilon_0 e^{\lambda_{max} i\Delta t} \tag{9}$$

where ε_0 is the initial separation. Now assuming that the two principal axis of nearby trajectories are colinear, we should choose the number of iterations i_s such that the distance d_i between trajectories is larger than the uncertainty region, i.e. $d_{i_s} \geq 2\varepsilon_{i_s}$. The estimate of i_s should be averaged over a number of neighboring training sequences (~50 depending on the signal dynamics).

Hence, to apply this method three quantities must be estimated: the largest Lyapunov

exponent, using one of the accepted algorithms. The initial separation can be estimated from the one-step predictor. And i_s by averaging local divergence. The computation time required to estimate these quantities is usually much less than setting by trial and error the length of the trajectory until a reasonable learning curve is achieved.

We also developed a method to train predictors for chaotic signals with large λ_{max}, but it will not be covered in this paper [8].

5. Results

We used this methodology to model the Mackey-Glass system (d=30, sampled at 1/6 Hz). A signal of 500 samples was obtained by 4th order Runge-Kutta integration and normalized between -1,1. The largest Lyapunov exponent for this signal is 0.0071 nats/sec. We selected a time delay neural network (TDNN) with topology 8-14-1. The output unit is linear, and the hidden layer has sigmoid nonlinearities. The number of taps in the delay line is 8.

We trained a one-step predictor and the multistep predictor with the methodology developed in this paper to compare results. The single step predictor was trained with static backpropagation with no momentum and step size of 0.001. Trained was stopped after 500 iterations. The final MSE was 0.000288. After training, the predictor was seeded with the first 8 points of the time series and iterated for 3,000 times. Figure 3a shows the corresponding output. Notice that the waveform produced by the model is much more regular that the Mackey-Glass signal, showing that some fine detail of the attractor has not been captured.

Next we trained the same TDNN with a global feedback loop (TDNNGF). The estimate of the i_s over the neighboring orbits provided an estimate of 14, and it is taken as the length of the trajectory. We displaced each training sequence by 3 samples (q=3 in eqn 7). BPTT was used to train the TDNNGF for 500 iterations over the same signal. The final MSE was 0.000648, higher than for the TDNN case. We could think that the resulting predictor was worse. The TDNNGF predictor was initialized with the same 8 samples of the time series and iterated for 3,000 times. Figure 3b shows the resulting waveform. It "looks" much closer to the original Mackey-Glass time series. We computed the average prediction error as a function of iteration for both predictors and also the theoretical rate of divergence of trajectories assuming an initial error ε_0 (Casdagli conjecture, which is the square of eqn 9) [7]. As can be seen in Figure 4 the TDNNGF is much closer to the theoretical limit, which means a much better model. We also computed the correlation dimension and the Lyapunov exponent estimated from the generated time series, and the figures obtained from TDNNGF are closer to the original time series.

Figure 5 shows the instability present in the training when the trajectory length is above the estimated value of 14. For this case the trajectory length is 20. As can be seen the MSE decreases but then fluctuates showing instability in the training.

6. Conclusions

This paper addresses dynamic modelling with artificial neural networks. We showed

that the network topology should be recurrent such that the iterative map is constrained during learning. This is a necessity since dynamic modelling seeks to capture the long term behavior of the dynamical system. These models can also be used as a sample by sample predictors. Since the network topology is recurrent, backpropagation through time or real time recurrent learning should be used in training. In this paper we showed how to select the length of the trajectory to avoid instability during training.

A lot more work needs to be done to reliably capture dynamical properties of time series and encapsulate them in artificial models. But we believe that the careful analysis of the dynamic characteristics and the study of its impact on the predictive model performance is much more promising than guess work. According to this (and others) studies, modelling of chaotic time series of low λ_{max} seems a reality. We have extended some of this work for time series with larger λ_{max}, and successfully captured the dynamics of the Lorenz system [8]. But there, the parameters for learning have to be much more carefully selected, and some of the choices are still arbitrary. The main issue is that the trajectories diverge so rapidly that predictors have a hard time to capture information regarding the global system dynamics. It is interesting to study the limit of predictability of this type of approach for high dimensional and high λ_{max} chaos.

Predictor	Corr. Dim.	Lyapunov
MG30	2.70+/-0.05	0.0073+/-0.0001
TDNNGF	2.65+/-0.03	0.0074+/-0.0001
TDNN	1.60+/-0.10	0.0063+/-0.0001

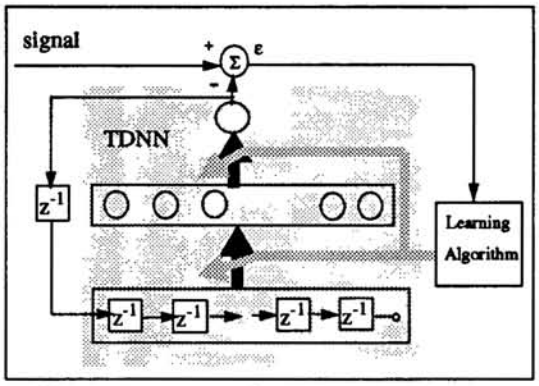

Figure 1. Proposed recurrent architecture (TDNNGF)

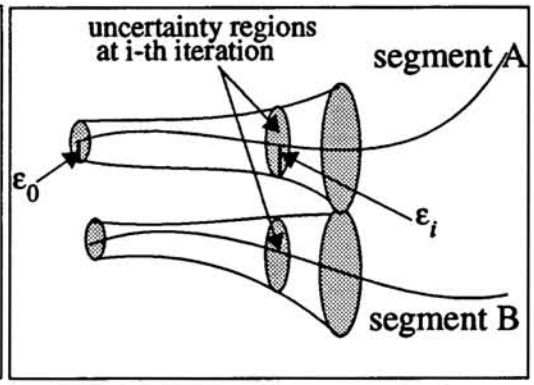

Figure 2. State space representation in training a model

7. Acknowledgments

This work was partially supported by NSF grant #ECS-9208789, and ONR #1494-94-1-0858.

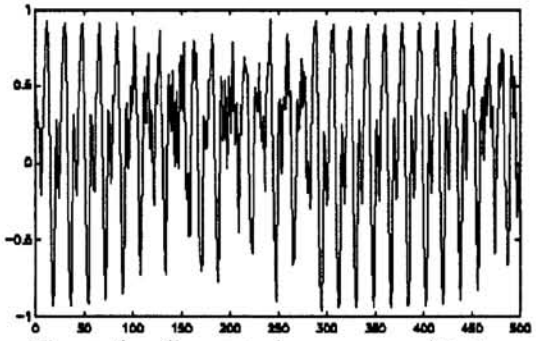

Figure 3a. Generated sequence with the TDNN

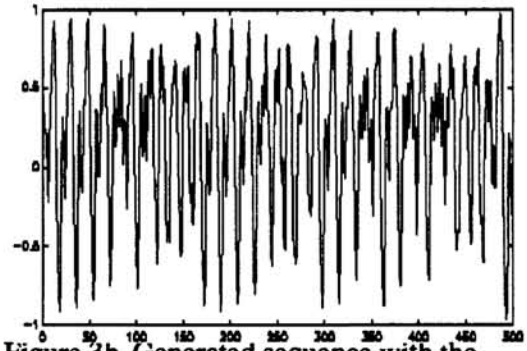

Figure 3b. Generated sequence with the TDNNGF

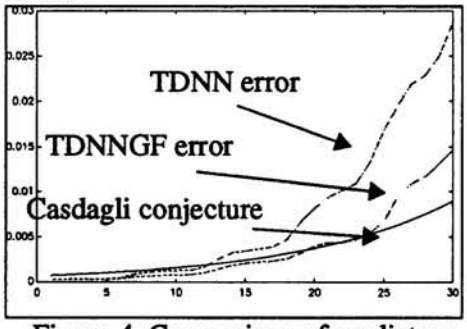

Figure 4. Comparison of predictors

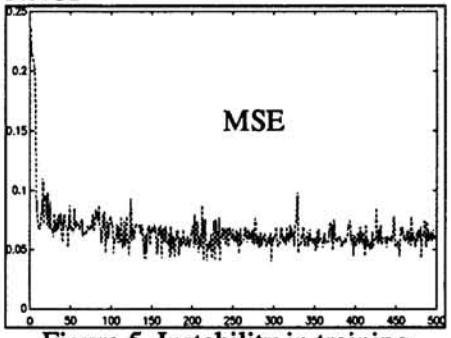

Figure 5. Instability in training

8. References

[1] Box, G. E., and G. M. Jenkins, Time Series Analysis, Forecasting and Control, Holden Day, San Francisco, 1970.

[2] Weigend, A. S., B. A. Huberman, and D. E. Rumelhart, "Predicting the future: a connectionist approach," International Journal of Neural Systems, vol. 1, pp. 193-209, 1990.

[3] Lapedes, R., and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modelling," Technical Report LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico, 1987.

[4] Kuo J-M., Principe J.C., "A systematic approach to chaotic time series modeling with neural networks", in IEEE Workshop on Neural Nets for Signal Processing, Ermioni, Greece, 1994.

[5] Principe, J. C., A. Rathie, and J. M. Kuo, "Prediction of chaotic time series with neural networks and the issue of dynamic modeling," International Journal of Biburcation and Chaos, vol. 2, no. 4, pp. 989-996, 1992.

[6] Hertz, J, A. Krogh, and R. G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Redwood City, CA, 1991.

[7] Casdagli, M., "Nonlinear prediction of chaotic time series," Physica D 35, pp.335-356, 1989.

[8] Kuo, J.M., "Nonlinear Dynamic Modelling with Artificial neural networks", Ph.D. dissertation, University of FLorida, 1993.