Journal of Machine Learning Research 14 (2013) 3785-3812 Submitted 11/12; Revised 8/13; Published 12/13

Optimally Fuzzy Temporal Memory

Karthik H. Shankar SHANKARK@BU.EDU
Marc W. Howard MARC777@BU.EDU
Center for Memory and Brain
Boston University
Boston, MA 02215, USA

Editor: Peter Dayan

Abstract

Any learner with the ability to predict the future of a structured time-varying signal must maintain a memory of the recent past. If the signal has a characteristic timescale relevant to future prediction, the memory can be a simple shift register—a moving window extending into the past, requiring storage resources that linearly grows with the timescale to be represented. However, an independent general purpose learner cannot *a priori* know the characteristic prediction-relevant timescale of the signal. Moreover, many naturally occurring signals show scale-free long range correlations implying that the natural prediction-relevant timescale is essentially unbounded. Hence the learner should maintain information from the longest possible timescale allowed by resource availability. Here we construct a fuzzy memory system that optimally sacrifices the temporal accuracy of information in a scale-free fashion in order to represent prediction-relevant information from exponentially long timescales. Using several illustrative examples, we demonstrate the advantage of the fuzzy memory system over a shift register in time series forecasting of natural signals. When the available storage resources are limited, we suggest that a general purpose learner would be better off committing to such a fuzzy memory system.

Keywords: temporal information compression, forecasting long range correlated time series

1. Introduction

Natural learners face a severe computational problem in attempting to predict the future of time varying signals. Rather than being presented with a large training set of examples, they must compute on-line using a continuously evolving representation of the recent past. A basic question arises here—how much of the recent past is required to generate future predictions? Maintaining past information in memory comes with a metabolic cost; we would expect a strong evolutionary pressure to minimize the resources required. A shift register can accurately represent information from the recent past up to a chosen timescale, while consuming resources that grow linearly with that timescale. However, the prediction-relevant timescale of the signal is generally unknown prior to learning. Moreover there are many examples of naturally occurring signals with scale-free long range correlations (Voss and Clarke, 1975; Mandelbrot, 1982; Field, 1987; Torralba and Oliva, 2003; Linkenkaer-Hansen et al., 2001; Baillie, 1996; Gilden, 2001; Van Orden et al., 2003; Wagenmakers et al., 2004), commonly known as $1/f$ signals, making the natural prediction-relevant timescale essentially unbounded. Our focus is on the following question: If an independent general purpose learner is to forecast long range correlated natural signals, what is the optimal way to represent the past information in memory with limited resources?

©2013 Karthik H. Shankar and Marc W. Howard.

We argue that the solution is to construct a memory that reflects the natural scale-free temporal structure associated with the uncertainties of the world. For example, the timing of an event that happened 100 seconds ago does not have to be represented as accurately in memory as the timing of an event that happened 10 seconds ago. Sacrificing the temporal accuracy of information in memory leads to tremendous resource conservation, yielding the capacity to represent information from exponentially long timescales with linearly growing resources. Moreover, by sacrificing the temporal accuracy of information in a scale-free fashion, the learner can gather the relevant statistics from the signal in a way that is optimal if the signal contains scale-free fluctuations. To mechanistically construct such a memory system, it is imperative to keep in mind that the information represented in memory should *self sufficiently* evolve in real time without relying on any information other than the instantaneous input and what is already represented in memory; reliance on any external information would require additional storage resources. In this paper we describe a *Fuzzy* (meaning temporally inaccurate) memory system that (i) represents information from very long timescales under limited resources, (ii) optimally sacrifices temporal accuracy while maximally preserving the prediction-relevant information from the past, and (iii) evolves self sufficiently in real time.[1]

The layout of the paper is as follows. In Section 2, based on some general properties of long range correlated signals, we derive the criterion for optimally sacrificing the temporal accuracy so that the prediction relevant information from exponentially long time scales is maximally preserved in the memory with finite resources. However, it is non-trivial to construct such a memory in a self sufficient way. In Section 3, we describe a strategy to construct a self sufficient scale-free representation of the recent past. This strategy is based on a neuro-cognitive model of internal time, TILT (Shankar and Howard, 2012), and is mathematically equivalent to encoding the Laplace transform of the past and approximating its inverse to reconstruct a fuzzy representation of the past. With an optimal choice of a set of memory nodes, this representation naturally leads to a self-sufficient fuzzy memory system. In Section 4, we illustrate the utility of the fuzzy memory with some simple time series forecasting examples. We show that the fuzzy memory enhances the ability to predict the future in comparison to a shift register with equal number of nodes. Optimal representation of the recent past in memory does not by itself guarantee the ability to successfully predict the future, for it is crucial to learn the prediction-relevant statistics underlying the signal with an efficient learning algorithm. The choice of the learning algorithm is however largely modular to the choice of the memory system. Here we entirely sidestep the problem of learning, and only focus on the memory. As a place holder for a learning algorithm, we use linear regression in the demonstrations of time series forecasting.

2. Optimal Accuracy-Capacity Tradeoff

Suppose a learner needs to learn to predict a real valued time series with long range correlations. Let $V = \{v_n : n \in (1,2,3...\infty)\}$ represent all past values of the time series relative to the present moment at $n = 0$. Let V be a stationary series with zero mean and finite variance. Ignoring any higher order correlations, let the two point correlation be $\langle v_n v_m \rangle \simeq 1/|n-m|^{\alpha}$. When $\alpha \leq 1$, the series will be long range correlated (Beran, 1994). The goal of the learner is to successfully predict the current value v_o at $n = 0$. Figure 1 shows a sample time series leading up to the present moment. The y-axis corresponds to the present moment and to its right lies the unknown future values of the time series. The x-axis labeled as shift register denotes a memory buffer wherein each v_n is accurately stored in

1. Fuzzy temporal memory is not related to fuzzy logic.

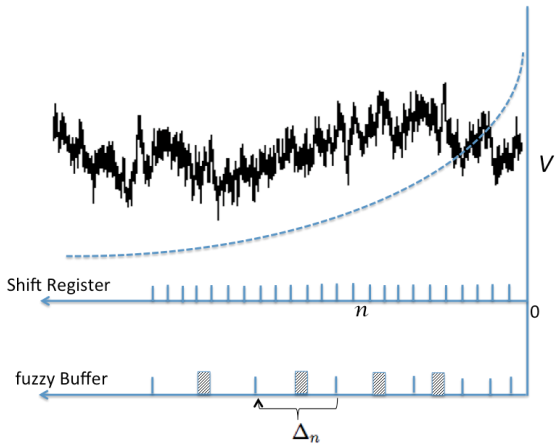

Figure 1: A sample time series V with power-law two-point correlation is plotted w.r.t. time(n) with the current time step taken to be $n = 0$. The figure contrasts the way in which the information is represented in a shift register and the fuzzy buffer. Each node of the fuzzy buffer linearly combines information from a bin containing multiple shift register nodes. The dashed curve shows the predictive information content from each time step that is relevant for predicting the future value of the time series.

a unique node. As we step forward in time, the information stored in each node will be transferred to its left-neighbor and the current value v_o will enter the first node. The shift register can thus self sufficiently evolve in real time. The longest time scale that can be represented with the shift register is linearly related to the number of nodes.

Given a limited number of memory nodes, what is the optimal way to represent V in the memory so that the information relevant to prediction of v_o is maximally preserved? We will show that this is achieved in the fuzzy buffer shown in Figure 1. Each node of the fuzzy buffer holds the average of v_ns over a bin. For the fuzzy buffer, the widths of the bins increase linearly with the center of the bin; the bins are chosen to tile the past time line. Clearly, the accuracy of representing V is sacrificed in the process of compressing the information in an entire bin into a real number, but note that we attain the capacity to represent information from exponentially long timescales. With some analysis, we will show that sacrificing the accuracy with bin widths chosen in this way leads to maximal preservation of prediction-relevant information.

It is clear that the fuzzy buffer shown in Figure 1 cannot evolve self sufficiently in real time; information lost during compression of a bin at any moment is required at the next moment to recompute the bin average. At each moment we would explicitly require all the v_ns to correctly update the values in the fuzzy buffer. Even though such a fuzzy buffer is not self sufficient, we shall analyze it to derive the optimal binning strategy that maximally preserves prediction-relevant information from the past . The reader who is willing to take the optimality of linearly-increasing bin widths on faith can skip ahead to Section 3 where we construct a self-sufficient memory system with that property.

2.1 Deriving the Optimal Binning Strategy

To quantify the prediction relevant information contained in V, let us first review some general properties of the long range correlated series. Since our aim here is only to represent V in memory and not to actually understand the generating mechanism underlying V, it is sufficient to consider V as being generated by a generic statistical algorithm, the ARFIMA model (Granger and Joyeux, 1980; Hosking, 1981; Wagenmakers et al., 2004). The basic idea behind the ARFIMA algorithm is that white noise at each time step can be fractionally integrated to generate a time series with long range correlations.[2] Hence the time series can be viewed as being generated by an infinite auto-regressive generating function. In other words, v_o can be generated from the past series V and an instantaneous Gaussian white noise input η_o.

$$v_o = \eta_o + \sum_{n=1}^{\infty} a(n) v_n. \tag{1}$$

The ARFIMA algorithm specifies the coefficients $a(n)$ in terms of the exponent α in the two point correlation function.

$$a(n) = \frac{(-1)^{n+1}\Gamma(d+1)}{\Gamma(n+1)\Gamma(d-n+1)}, \tag{2}$$

where d is the fractional integration power given by $d = (1-\alpha)/2$. The time series is stationary and long range correlated with finite variance only when $d \in (0, 1/2)$ or $\alpha \in (0,1)$ (Granger and Joyeux, 1980; Hosking, 1981). The asymptotic behavior of $a(n)$ for large n can be obtained by applying Euler's reflection formula and Stirling's formula to approximate the Gamma functions in Equation 2. It turns out that when either d is small or n is large,

$$a(n) \simeq \left[\frac{\Gamma(d+1)\sin(\pi d)}{\pi}\right] n^{-(1+d)}. \tag{3}$$

For the sake of analytic tractability, the following analysis will focus only on small values of d. From Equation 1, note that $a(n)$ is a measure of the relevance of v_n in predicting v_o, which we shall call the $\mathcal{PIC}$-Predictive Information Content of v_n. Taylor expansion of Equation 3 shows that each $a(n)$ is linear in d up to the leading order. Hence in small d limit, any v_n is a stochastic term η_n plus a history dependent term of order $O(d)$. Restricting to linear order in d, the total $\mathcal{PIC}$ of v_n and v_m is simply the sum of their individual $\mathcal{PIC}$s, namely $a(n)+a(m)$. Thus in the small d limit, $\mathcal{PIC}$ is a simple measure of predictive information that is also an extensive quantity.

When the entire V is accurately represented in an infinite shift register, the total $\mathcal{PIC}$ contained in the shift register is the sum of all $a(n)$. This is clearly the maximum attainable value of $\mathcal{PIC}$ in any memory buffer, and it turns out to be 1.

$$\mathcal{PIC}_{max} = \sum_{n=1}^{\infty} a(n) = 1.$$

In a shift register with $N_{\max}$ nodes, the total $\mathcal{PIC}$ is

$$\begin{aligned} \mathcal{PIC}^{\mathrm{SR}}_{\mathrm{tot}} = \sum_{n=1}^{N_{\max}} a(n) \quad &= \quad 1 - \sum_{N_{\max}}^{\infty} a(n) \;\simeq\; 1 - \frac{\sin(\pi d)\Gamma(d+1)}{\pi d} N_{\max}^{-d}, \\ &\xrightarrow{d\to 0} \quad d \ln N_{\max}. \end{aligned} \tag{4}$$

2. The most general model ARFIMA(p,d,q) can generate time series with both long and short range structures. Here we choose $p = q = 0$ to ignore the short range structures.

For any fixed $N_{\max}$, when d is sufficiently small $\mathcal{PIC}_{\rm tot}^{\rm SR}$ will be very small. For example, when $N_{\max} = 100$ and $d = 0.1$, $\mathcal{PIC}_{\rm tot}^{\rm SR} \simeq 0.4$. For smaller values of d, observed in many natural signals, $\mathcal{PIC}_{\rm tot}^{\rm SR}$ would be even lower. When $d = 0.01$, for $N_{\max}$ as large as 10000, the $\mathcal{PIC}_{\rm tot}^{\rm SR}$ is only 0.08. A large portion of the predictive information lies in long timescales. So the shift register is ill-suited to represent information from a long range correlated series.

The $\mathcal{PIC}_{\rm tot}$ for a memory buffer can be increased if each of its nodes stored a linear combination of many v_ns rather than a single v_n as in the shift register. This can be substantiated through information theoretic considerations formulated by the Information Bottleneck ($I\mathcal{B}$) method (Tishby et al., 1999). A multi-dimensional Gaussian variable can be systematically compressed to lower dimensions by linear transformations while maximizing the relevant information content (Chechik et al., 2005). Although V is a unidimensional time series in our consideration, at any moment the entire V can be considered as an infinite dimensional Gaussian variable since only its two point correlations are irreducible. Hence it heuristically follows from $I\mathcal{B}$ that linearly combining the various v_ns into a given number of combinations and representing each of them in separate memory nodes should maximize the $\mathcal{PIC}_{\rm tot}$. By examining Equation 1, it is immediately obvious that if we knew the values of $a(n)$, the entire V could be linearly compressed into a single real number $\sum a(n)v_n$ conveying all of the prediction relevant information. However, such a single-node memory buffer is not self sufficient: at each moment we explicitly need the entire V to update the value in that node. As an unbiased choice that does not require *a priori* knowledge of the statistics of the time series, we simply consider uniform averaging over a bin. Uniform averaging over a bin discards separate information about the time of the values contributing to the bin. Given this consideration, how should we space the bins to maximize the $\mathcal{PIC}_{\rm tot}$?

Consider a bin ranging between n and $n+\Delta_n$. We shall examine the effect of averaging all the v_ms within this bin and representing it in a single memory node. If all the v_ms in the bin are individually represented, then the $\mathcal{PIC}$ of the bin is $\sum_m a(m)$. Compressing all the v_ms into their average would however lead to an error in prediction of v_o; from Equation 1 this error is directly related to the extent to which the $a(m)$s within the bin are different from each other. Hence there should be a reduction in the $\mathcal{PIC}$ of the bin. Given the monotonic functional form of $a(n)$, the maximum reduction can only be $\sum_m |a(m)-a(n)|$. The net $\mathcal{PIC}$ of the memory node representing the bin average is then

$$\begin{aligned}\mathcal{PIC} &= \sum_{m=n}^{n+\Delta_n} a(m) - \sum_{m=n}^{n+\Delta_n} |a(n)-a(m)|, \\ &\simeq \frac{n\,a(n)}{d}\left[2-2\left(1+\frac{\Delta_n}{n}\right)^{-d}-\frac{d\,\Delta_n}{n}\right].\end{aligned}$$

The series summation in the above equation is performed by approximating it as an integral in the large n limit. The bin size that maximizes the $\mathcal{PIC}$ can be computed by setting the derivative of $\mathcal{PIC}$ w.r.t. Δ_n equal to zero. The optimal bin size and the corresponding $\mathcal{PIC}$ turns out to be

$$\Delta_n^{opt} = \left[2^{1/(1+d)}-1\right] n, \qquad \mathcal{PIC}^{opt} \simeq \frac{n\,a(n)}{d}\left[2+d-(1+d)2^{1/(1+d)}\right]. \tag{5}$$

When the total number of nodes $N_{\max}$ is finite, and we want to represent information from the longest possible timescale, the straightforward choice is to pick successive bins such that they completely tile up the past time line as schematically shown by fuzzy buffer in Figure 1. If we label

the nodes of the fuzzy buffer by N, ranging from 1 to $N_{\max}$, and denote the starting point of each bin by n_N, then

$$n_{N+1} = (1+c)n_N \qquad \Longrightarrow \qquad n_N = n_1(1+c)^{(N-1)}, \tag{6}$$

where $1+c = 2^{1/(1+d)}$. Note that the fuzzy buffer can represent information from timescales of the order $n_{N_{\max}}$, which is exponentially large compared to the timescales represented by a shift register with $N_{\max}$ nodes. The total $\mathcal{PIC}$ of the fuzzy buffer, $\mathcal{PIC}^{\text{FB}}_{\text{tot}}$, can now be calculated by summing over the $\mathcal{PIC}$s of each of the bins. Focusing on small values of d so that $a(n)$ has the power law form for all n, applying Equations 3 and 5 yields

$$\mathcal{PIC}^{\text{FB}}_{\text{tot}} \simeq \sum_{N=1}^{N_{\max}} \frac{\sin(\pi d)\Gamma(d+1)}{\pi d} \left[2+d-(1+d)2^{1/(1+d)}\right] n_N^{-d}.$$

Taking $n_1 = 1$ and n_N given by Equation 6,

$$\begin{aligned} \mathcal{PIC}^{\text{FB}}_{\text{tot}} \quad &\simeq \quad \frac{\sin(\pi d)\Gamma(d+1)}{\pi d} \left[2+d-(1+d)2^{1/(1+d)}\right] \left[\frac{1-(1+c)^{-d\ N_{\max}}}{1-(1+c)^{-d}}\right], \\ &\xrightarrow{d\to 0} \quad [\ln 4 - 1] d\ N_{\max}. \end{aligned} \tag{7}$$

Comparing Equations 4 and 7, note that when d is small, the $\mathcal{PIC}^{\text{FB}}_{\text{tot}}$ of the fuzzy buffer grows linearly with $N_{\max}$ while the $\mathcal{PIC}^{\text{SR}}_{\text{tot}}$ of the shift register grows logarithmically with $N_{\max}$. For example, with $N_{\max} = 100$ and $d = 0.01$, the $\mathcal{PIC}^{\text{FB}}_{\text{tot}}$ of the fuzzy buffer is 0.28, while the $\mathcal{PIC}^{\text{SR}}_{\text{tot}}$ of the shift register is only 0.045. Hence when $N_{\max}$ is relatively small, the fuzzy buffer represents a lot more predictive information than a shift register.

The above description of the fuzzy buffer corresponds to the ideal case wherein the neighboring bins do not overlap and uniform averaging is performed within each bin. Its critical property of linearly increasing bin sizes ensures that the temporal accuracy of information is sacrificed optimally and in a scale-free fashion. However, this ideal fuzzy buffer cannot self sufficiently evolve in real time because at every moment all v_ns are explicitly needed for its construction. In the next section, we present a self sufficient memory system that possesses the critical property of the ideal fuzzy buffer, but differs from it by having overlapping bins and non-uniform weighted averaging within the bins. To the extent the self sufficient fuzzy memory system resembles the ideal fuzzy buffer, we can expect it to be useful in representing long range correlated signals in a resource-limited environment.

3. Constructing Self Sufficient Fuzzy Memory

In this section, we first describe a mathematical basis for representing the recent past in a scale-free fashion based on a neuro-cognitive model of internal time, TILT (Shankar and Howard, 2012). We then describe several critical considerations necessary to implement this representation of recent past into a discrete set of memory nodes. Like the ideal fuzzy buffer described in the Section 2, this memory representation will sacrifice temporal accuracy to represent prediction-relevant information over exponential time scales. But unlike the ideal fuzzy buffer, the resulting memory representation will be self sufficient, without requiring additional resources to construct the representation.

Let $\mathbf{f}(\tau)$ be a real valued function presented over real time τ. Our aim now is to construct a memory that represents the past values of $\mathbf{f}(\tau)$ as activity distributed over a set of nodes with

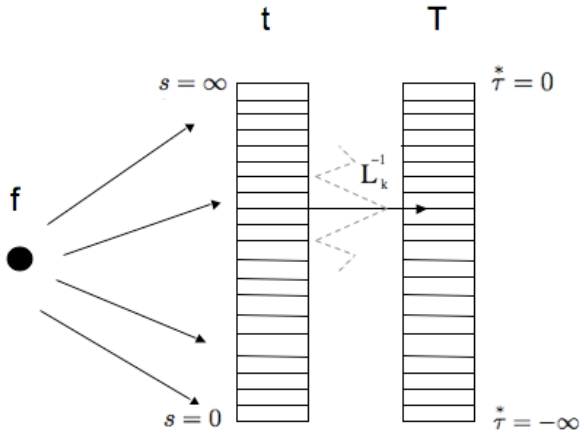

Figure 2: The scale-free fuzzy representation - Each node in the **t** column is a leaky integrator with a specific decay constant s that is driven by the functional value **f** at each moment. The activity of the **t** column is transcribed at each moment by the operator $\mathbf{L}_k^{-1}$ to represent the past functional values in a scale-free fuzzy fashion in the **T** column.

accuracy that falls off in a scale-free fashion. This is achieved using two columns of nodes **t** and **T** as shown in Figure 2. The **T** column estimates $\mathbf{f}(\tau)$ up to the present moment, while the **t** column is an intermediate step used to construct **T**. The nodes in the **t** column are leaky integrators with decay constants denoted by s. Each leaky integrator independently gets activated by the value of **f** at any instant and gradually decays according to

$$\frac{d\mathbf{t}(\tau,s)}{d\tau} = -s\mathbf{t}(\tau,s) + \mathbf{f}(\tau). \tag{8}$$

At every instant, the information in the **t** column is transcribed into the **T** column through a linear operator $\mathbf{L}_k^{-1}$.

$$\begin{aligned} \mathbf{T}(\tau,\overset{*}{\tau}) &= \frac{(-1)^k}{k!} s^{k+1}\mathbf{t}^{(k)}(\tau,s) \ : \ \text{where } s=-k/\overset{*}{\tau}. \\ \mathbf{T} &\equiv \mathbf{L}_k^{-1}\big[\mathbf{t}\big]. \end{aligned} \tag{9}$$

Here k is a positive integer and $\mathbf{t}^{(k)}(\tau,s)$ is the k-th derivative of $\mathbf{t}(\tau,s)$ with respect to s. The nodes of the **T** column are labeled by the parameter $\overset{*}{\tau}$ and are in one to one correspondence with the nodes of the **t** column labeled by s. The correspondence between s and $\overset{*}{\tau}$ is given by $s=-k/\overset{*}{\tau}$. We refer to $\overset{*}{\tau}$ as *internal time*; at any moment τ, a $\overset{*}{\tau}$ node estimates the value of **f** at a time $\tau+\overset{*}{\tau}$ in the past. The range of values of s and $\overset{*}{\tau}$ can be made as large as needed at the cost of resources, but for mathematical idealization we let them have an infinite range.

The mathematical inspiration of this approach comes from the fact that $\mathbf{t}(\tau,s)$ encodes the Laplace transform of the entire history of the function **f** up to time τ, and the operator $\mathbf{L}_k^{-1}$ approximately inverts the Laplace transform (Post, 1930). As $k\to\infty$, $\mathbf{T}(\tau,\overset{*}{\tau})$ becomes a faithful

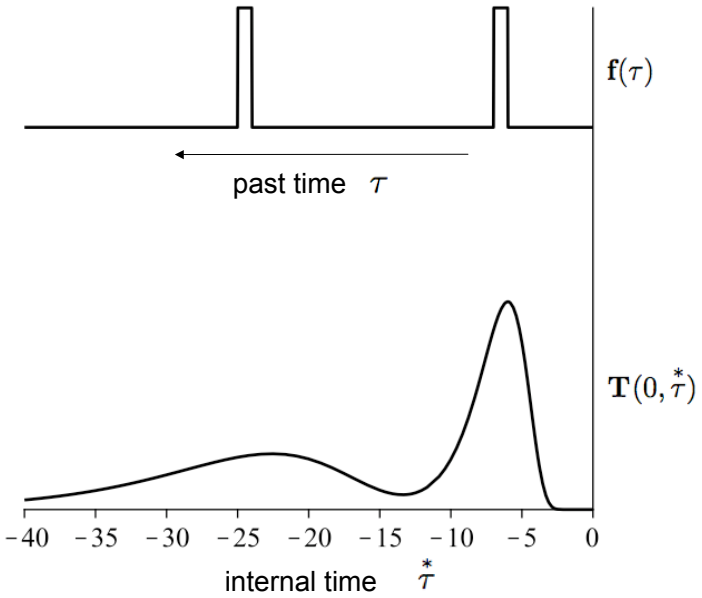

Figure 3: Taking the present moment to be $\tau = 0$, a sample $\mathbf{f}(\tau)$ is plotted in the top, and the momentary activity distributed across the $\mathbf{T}$ column nodes is plotted in the bottom.

reconstruction of the history of $\mathbf{f}$ from $-\infty$ to τ, that is $\mathbf{T}(\tau,\overset{*}{\tau}) \simeq \mathbf{f}(\tau+\overset{*}{\tau})$ for all values of $\overset{*}{\tau}$ from 0 to $-\infty$. When k is finite $\mathbf{T}(\tau,\overset{*}{\tau})$ is an inaccurate reconstruction of the history of $\mathbf{f}$. For example, taking the current moment to be $\tau = 0$, Figure 3 illustrates an $\mathbf{f}$ that is briefly non-zero around $\tau = -7$ and $\tau = -23$. The reconstructed history of $\mathbf{f}$ in the $\mathbf{T}$ column shows two peaks approximately around $\overset{*}{\tau} = -7$ and $\overset{*}{\tau} = -23$. The value of $\mathbf{f}$ at any particular moment in the past is thus smeared over a range of $\overset{*}{\tau}$ values, and this range of smearing increases as we go deeper into the past. Thus, the more distant past is reconstructed with a lesser temporal accuracy.

Furthermore, it turns out that the smear is precisely scale invariant. To illustrate this, consider $\mathbf{f}(\tau)$ to be a Dirac delta function at a moment τ_o in the past, $\mathbf{f}(\tau) = \delta(\tau - \tau_o)$, and let the present moment be $\tau = 0$. Applying Equations 8 and 9, we obtain

$$\mathbf{T}(0,\overset{*}{\tau}) = \frac{1}{|\tau_o|}\frac{k^{k+1}}{k!}\left(\frac{\tau_o}{\overset{*}{\tau}}\right)^{k+1} e^{-k(\tau_o/\overset{*}{\tau})}. \tag{10}$$

In the above equation both τ_o and $\overset{*}{\tau}$ are negative. $\mathbf{T}(0,\overset{*}{\tau})$ is the fuzzy reconstruction of the delta function input. $\mathbf{T}(0,\overset{*}{\tau})$ is a smooth peaked function whose height is proportional to $1/|\tau_o|$, width is proportional to $|\tau_o|$, and the area is equal to 1. Its dependence on the ratio $(\tau_o/\overset{*}{\tau})$ ensures scale invariance—for any τ_o we can linearly scale the $\overset{*}{\tau}$ values to hold the shape of the function fixed. In this sense, $\mathbf{T}$ represents the history of $\mathbf{f}$ with a scale invariant smear. To quantify how much smear is introduced, we can estimate the width of the peak as the standard deviation σ of $\mathbf{T}(0,\overset{*}{\tau})$ from the

above equation, which for $k > 2$ turns out to be

$$\sigma[\mathbf{T}(0,\overset{*}{\tau})] = \frac{|\tau_o|}{\sqrt{k-2}}\left[\frac{k}{k-1}\right]. \tag{11}$$

Note that k is the only free parameter that affects the smear; k indexes the smear in the representation. The larger the k the smaller the smear. In the limit $k \to \infty$, the smear vanishes and the delta function input propagates into the **T** column exactly as delta function without spreading, as if **T** were a shift register.

Though we took **f** to be a simple delta function input to illustrate the scale invariance of the **T** representation, we can easily evaluate the **T** representation of an arbitrary **f** from Equations 8 and 9.

$$\mathbf{T}(0,\overset{*}{\tau}) = \int_{-\infty}^{0}\left[\frac{1}{|\overset{*}{\tau}|}\frac{k^{k+1}}{k!}\left(\frac{\tau'}{\overset{*}{\tau}}\right)^{k} e^{-k(\tau'/\overset{*}{\tau})}\right]\mathbf{f}(\tau')\,d\tau'. \tag{12}$$

The **f** values from a range of past times are linearly combined and represented in each $\overset{*}{\tau}$ node. The term in the square brackets in the above equation is the weighting function of the linear combination. Note that it is not a constant function over a circumscribed past time bin, rather it is a smooth function peaked at $\tau' = \overset{*}{\tau}$, with a spread proportional to $|\overset{*}{\tau}|$. Except for the fact that this weighting function is not uniform, the activity of a $\overset{*}{\tau}$ node has the desired property mimicking a bin of the ideal fuzzy buffer described in Section 2.

3.1 Self Sufficient Discretized Implementation

Although it appears from Equation 12 that the **T** representation requires explicit information about the **f** values over the past time, recall that it can be constructed from instantaneous **t** representation. Since Equation 8 is a local differential equation, the activity of each **t** node will independently evolve in real time, depending only on the present value of the node and the input available at that moment. Hence any discrete set of **t** nodes also evolves self sufficiently in real time. To the extent the activity in a discrete set of **T** nodes can be constructed from a discrete set of **t** nodes, this memory system as a whole can self sufficiently evolve in real time. Since the activity of each $\overset{*}{\tau}$ node in the **T** column is constructed independently of the activity of other $\overset{*}{\tau}$ nodes, we can choose any discrete set of $\overset{*}{\tau}$ values to form our memory system. In accordance with our analysis of the ideal fuzzy buffer in Section 2, we shall pick the following nodes.

$$\overset{*}{\tau}_{min}\,,\ \overset{*}{\tau}_{min}(1+c)\,,\ \overset{*}{\tau}_{min}(1+c)^2\,,\ldots\ \overset{*}{\tau}_{min}(1+c)^{(N_{\max}-1)} = \overset{*}{\tau}_{max}. \tag{13}$$

Together, these nodes form the fuzzy memory representation with $N_{\max}$ nodes. The spacing in Equation 13 will yield several important properties.

Unlike the ideal fuzzy buffer described in Section 2, it is not possible to associate a circumscribed bin for a node because the weighting function (see Equation 12) does not have compact support. Since the weighting function associated with neighboring nodes overlap with each other, it is convenient to view the bins associated with neighboring nodes as partially overlapping. The overlap between neighboring bins implies that some information is redundantly represented in multiple nodes. We shall show in the forthcoming subsections that by appropriately tuning the parameters k and c, this information redundancy can be minimized and equally spread in a scale-free fashion.

3.1.1 DISCRETIZED DERIVATIVE

Although any set of $\overset{*}{\tau}$ nodes could be picked to form the memory buffer, their activity is ultimately constructed from the nodes in the $\mathbf{t}$ column whose s values are given by the one to one correspondence $s = -k/\overset{*}{\tau}$. Since the $\mathbf{L}_k^{-1}$ operator has to take the k-th derivative along the s axis, $\mathbf{L}_k^{-1}$ depends on the way the s-axis is discretized.

For any discrete set of s values, we can define a linear operator that implements a discretized derivative. For notational convenience, let us denote the activity at any moment $\mathbf{t}(\tau, s)$ as simply $\mathbf{t}(s)$. Since $\mathbf{t}$ is a column vector with the rows labeled by s, we can construct a derivative matrix [D] such that

$$\mathbf{t}^{(1)} = [\mathrm{D}]\mathbf{t} \qquad \Longrightarrow \qquad \mathbf{t}^{(k)} = [\mathrm{D}]^k \mathbf{t}.$$

The individual elements in the square matrix [D] depends on the set of s values. To compute these elements, consider any three successive nodes with s values s_{-1}, s_o, s_1. The discretized first derivative of $\mathbf{t}$ at s_o is given by

$$\mathbf{t}^{(1)}(s_o) = \frac{\mathbf{t}(s_1) - \mathbf{t}(s_o)}{s_1 - s_o}\left[\frac{s_o - s_{-1}}{s_1 - s_{-1}}\right] + \frac{\mathbf{t}(s_o) - \mathbf{t}(s_{-1})}{s_o - s_{-1}}\left[\frac{s_1 - s_o}{s_1 - s_{-1}}\right].$$

The row in [D] corresponding to s_o will have non-zero entries only in the columns corresponding to s_{-1}, s_o and s_1. These three entries can be read out as coefficients of $\mathbf{t}(s_{-1})$, $\mathbf{t}(s_o)$ and $\mathbf{t}(s_1)$ respectively in the r.h.s of the above equation. Thus the entire matrix [D] can be constructed from any chosen set of s values. By taking the k-th power of [D], the $\mathbf{L}_k^{-1}$ operator can be straightforwardly constructed and the activity of the chosen set of $\overset{*}{\tau}$ nodes can be calculated at each moment.[3] This memory system can thus self sufficiently evolve in real time.

When the spacing between the nodes (controlled by the parameter c) is small, the discretized k-th derivative will be accurate. Under uniform discretization of the s axis, it can be shown that the relative error in computation of the k-th derivative due to discretization is of the order $O(k\delta_s^2/24)$, where δ_s is the distance between neighboring s values (see appendix B of Shankar and Howard, 2012). Based on the s values corresponding to Equation 13, it turns out that the relative error in the construction of the activity of a $\overset{*}{\tau}$ node is $O(k^3c^2/96\overset{*}{\tau}^2)$. For large $\overset{*}{\tau}$ the error is quite small but for small $\overset{*}{\tau}$ the error can be significant. To curtail the discretization error, we need to hold $\overset{*}{\tau}_{min}$ sufficiently far from zero. The error can also be controlled by choosing small c for large k and vice versa. If for practical purposes we require a very small $\overset{*}{\tau}_{min}$, then ad hoc strategies can be adopted to control the error at low $\overset{*}{\tau}$ nodes. For example, by relaxing the requirement of one to one correspondence between the $\mathbf{t}$ and $\mathbf{T}$ nodes, we can choose a separate set of closely spaced s values to exclusively compute the activity of each of the small $\overset{*}{\tau}$ nodes.

Finally, it has to be noted that the discretization error induced in this process should not be considered as an error in the conventional sense of numerical solutions to differential equations. While numerically evolving differential equations with time-varying boundary conditions, the discretization error in the derivatives will propagate leading to large errors as we move farther away from the boundary at late times. But in the situation at hand, since the activity of each $\overset{*}{\tau}$ node is computed independently of others, the discretization error does not propagate. Moreover, it should be noted

3. Note that we need k extra nodes in the top and bottom of the $\mathbf{t}$ column in addition to those that come from one to one correspondence with the chosen $\overset{*}{\tau}$ values.

that the effect of discretization can be better viewed as coarse-graining the k-th derivative rather than as inducing an error in computing the k-th derivative. The fact that each $\overset{*}{\tau}$ node ultimately holds a scale-free coarse-grained value of the input function (see Equation 12), suggests that we wouldn't need the exact k-th derivative to construct its activity. To the extent the discretization is scale-free as in Equation 13, the $\mathbf{T}$ representation constructed from the coarse-grained k-th derivative will represent some scale free coarse grained value of the input; however the weighting function would not exactly match that in Equation 12. In other words, even if the discretized implementation does not accurately match the continuum limit, it still accurately satisfies the basic properties we require from a fuzzy memory system.

3.1.2 SIGNAL-TO-NOISE RATIO WITH OPTIMALLY-SPACED NODES

The linearity of Equations 8 and 9 implies that any noise in the input function $\mathbf{f}(\tau)$ will exactly be represented in $\mathbf{T}$ without any amplification. However when there is random uncorrelated noise in the $\mathbf{t}$ nodes, the discretized $\mathbf{L}_k^{-1}$ operator can amplify that noise, more so if c is very small. It turns out that choice of nodes according to Equation 13 results in a constant signal-to-noise ratio across time scales.

If uncorrelated noise with standard deviation η is added to the activation of each of the $\mathbf{t}$ nodes, then the $\mathbf{L}_k^{-1}$ operator combines the noise from $2k$ neighboring nodes leading to a noisy $\mathbf{T}$ representation. If the spacing between the nodes neighboring a s node is δ_s, then the standard deviation of the noise generated by the $\mathbf{L}_k^{-1}$ operator is approximately $\eta\sqrt{2^k}s^{k+1}/\delta_s^k k!$. To view this noise in an appropriate context, we can compare it to the magnitude of the representation of a delta function signal at a past time $\tau_o = -k/s$ (see Equation 10). The magnitude of the $\mathbf{T}$ representation for a delta function signal is approximately $k^k e^{-k} s/k!$. The signal to noise ratio (SNR) for a delta function signal is then

$$\text{SNR} = \eta^{-1}\left(\frac{k[\delta_s/s]}{\sqrt{2}e}\right)^k. \tag{14}$$

If the spacing between neighboring nodes changes such that $[\delta_s/s]$ remains a constant for all s, then the signal to noise ratio will remain constant over all timescales. This would however require that the nodes are picked according to Equation 13, making SNR = $\eta^{-1}(kc/\sqrt{2}e)^k$. Any other set of nodes would make the signal to noise ratio zero either at large or small timescales.

This calculation however does not represent the most realistic situation. Because the $\mathbf{t}$ nodes are leaky integrators (Equation 8), the white noise present across time will accumulate and hence nodes with long time constants should have a higher value of η. In fact, the standard deviation of white noise in the $\mathbf{t}$ nodes should go down with s according to $\eta \propto 1/\sqrt{s}$. From Equation 14, we can then conclude that the SNR of large $\overset{*}{\tau}$ nodes should drop down to zero as $1/\sqrt{|\overset{*}{\tau}|}$. However, because each $\overset{*}{\tau}$ node represents a weighted temporal average of the past signal, it is not appropriate to use an isolated delta function signal to estimate the signal to noise ratio. It is more appropriate to compare temporally averaged noise to a temporally spread signal. We consider two such signals. (i) Suppose $\mathbf{f}(\tau)$ itself is a temporally uncorrelated white noise like signal. The standard deviation in the activity of a $\overset{*}{\tau}$ node in response to this signal is proportional to $1/\sqrt{|\overset{*}{\tau}|}$ (see Equation 16 in the appendix). The SNR for this temporally-extended signal is a constant over all $\overset{*}{\tau}$ nodes. (ii) Consider a purely positive signal where $\mathbf{f}(\tau)$ is a sequence of delta function spikes generated by a Poisson process.

The total expected number of spikes that would be generated in the timescale of integration of a $\overset{*}{\tau}$ node is simply $\sqrt{|\overset{*}{\tau}|}$. Consequently, the expectation value of the activity of a $\overset{*}{\tau}$ node in response to this signal would be $\sqrt{|\overset{*}{\tau}|}$ multiplied by the magnitude of representation of a single delta function. The SNR for this signal is again a constant over all $\overset{*}{\tau}$ nodes. So, we conclude that for any realistic stationary signal spread out in time, the SNR will be a constant over all timescales as long as the nodes are chosen according to Equation 13.

3.2 Information Redundancy

The fact that a delta function input in $\mathbf{f}$ is smeared over many $\overset{*}{\tau}$ nodes implies that there is a redundancy in information representation in the $\mathbf{T}$ column. In a truly scale-free memory buffer, the redundancy in information representation should be equally spread over all time scales represented in the buffer. The information redundancy can be quantified in terms of the mutual information shared between neighboring nodes in the buffer. In the appendix, it is shown that in the presence of scale free input signals, the mutual information shared by any two neighboring buffer nodes can be a constant only if the $\overset{*}{\tau}$ nodes are distributed according to Equation 13. Consequently information redundancy is uniformly spread only when the $\overset{*}{\tau}$ nodes are given by Equation 13.

The uniform spread of information redundancy can be intuitively understood by analyzing how a delta function input spreads through the buffer nodes as time progresses. Figure 4 shows the activity of the buffer nodes at three points in time following the input. In Figure 4a where the $\overset{*}{\tau}$ values of the buffer nodes are chosen to be equidistant, the activity is smeared over more and more number of nodes as time progresses. This implies that the information redundancy is large in the nodes representing long timescales. In Figure 4b where the $\overset{*}{\tau}$ values of the buffer nodes are chosen according to Equation 13, the activity pattern does not smear, instead the activity as a whole gets translated with an overall reduction in size. The translational invariance of the activity pattern as it passes through the buffer nodes explains why the information redundancy between any two neighboring nodes is a constant. The translational invariance of the activity pattern can be analytically established as follows.

Consider two different values of τ_o in Equation 10, say τ_1 and τ_2. Let the corresponding $\mathbf{T}$ activities be $\mathbf{T}_1(0,\overset{*}{\tau})$ and $\mathbf{T}_2(0,\overset{*}{\tau})$ respectively. If the $\overset{*}{\tau}$ value of the N-th node in the buffer is given by $\overset{*}{\tau}_N$, then the pattern of activity across the nodes is translationally invariant if and only if $\mathbf{T}_1(0,\overset{*}{\tau}_N) \propto \mathbf{T}_2(0,\overset{*}{\tau}_{N+m})$ for some constant integer m. For this to hold true, we need the quantity

$$\frac{\mathbf{T}_1(0,\overset{*}{\tau}_N)}{\mathbf{T}_2(0,\overset{*}{\tau}_{N+m})} = \left(\frac{\tau_1}{\tau_2}\right)^k \left[\frac{\overset{*}{\tau}_{N+m}}{\overset{*}{\tau}_N}\right]^{k+1} e^{k\left[\frac{\tau_1}{\overset{*}{\tau}_N} - \frac{\tau_2}{\overset{*}{\tau}_{N+m}}\right]}$$

to be independent of N. This is possible only when the quantity inside the power law form and the exponential form are separately independent of N. The power law form can be independent of N only if $\overset{*}{\tau}_N \propto (1+c)^N$, which implies the buffer nodes have $\overset{*}{\tau}$ values given by Equation 13. The exponential form is generally dependent on N except when its argument is zero, which happens whenever $(1+c)^m = \tau_2/\tau_1$ for some integer m. For any given τ_1, there are infinitely many τ_2 values for which the condition holds. Moreover when c is small, the condition will approximately hold for

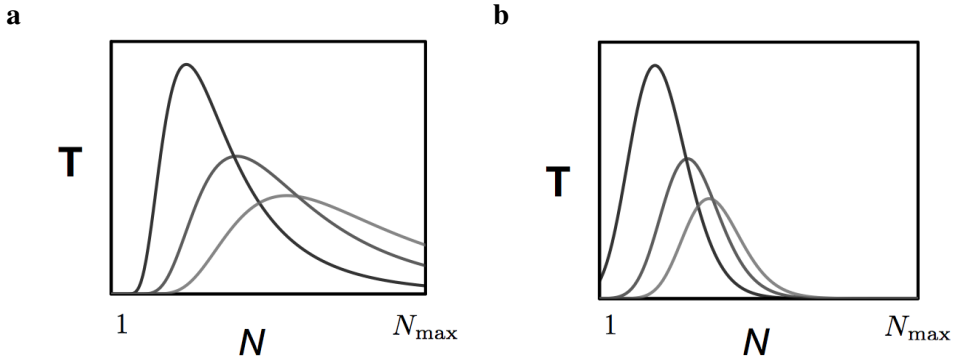

Figure 4: Activity of the fuzzy memory nodes in response to a delta function input at three different times with (**a**) uniformly spaced nodes and (**b**) nodes chosen in accordance with Equation 13.

any τ_2. Hence, the translational invariance of the activity pattern holds only when the $\overset{*}{\tau}$ values of the buffer nodes conform to Equation 13.

3.3 Balancing Information Redundancy and Information Loss

We have seen that the choice of $\overset{*}{\tau}$ nodes in accordance with Equation 13 ensures that the information redundancy is equally distributed over the buffer nodes. However, equal distribution of information redundancy is not sufficient; we would also like to minimize information redundancy. It turns out that we cannot arbitrarily reduce the information redundancy without creating information loss. The parameters k and c have to be tuned in order to balance information redundancy and information loss. If k is too small for a given c, then many nodes in the buffer will respond to input from any given moment in the past, resulting in information redundancy. On the other hand, if k is too large for a given c, the information from many moments in the past will be left unrepresented in any of the buffer nodes, resulting in information loss. So we need to match c with k to simultaneously minimize both information redundancy and information loss.

This can be achieved if the information from any given moment in the past is not distributed over more than two neighboring buffer nodes. To formalize this, consider a delta function input at a time τ_o in the past and let the current moment be $\tau = 0$. Let us look at the activity induced by this input (Equation 10) in four successive buffer nodes, $N-1$, N and $N+1$ and $N+2$. The $\overset{*}{\tau}$ values of these nodes are given by Equation 13, for instance $\overset{*}{\tau}_N = \overset{*}{\tau}_{min}(1+c)^{N-1}$ and $\overset{*}{\tau}_{N+1} = \overset{*}{\tau}_{min}(1+c)^{N}$. From Equation 10, it can be seen that the N-th node attains its maximum activity when $\tau_o = \overset{*}{\tau}_N$ and the $(N+1)$-th node attains its maximum activity when $\tau_o = \overset{*}{\tau}_{N+1}$, and for all the intervening times of τ_o between $\overset{*}{\tau}_N$ and $\overset{*}{\tau}_{N+1}$, the information about the delta function input will be spread over both N-th and the $(N+1)$-th nodes. To minimize the information redundancy, we simply require that when τ_o is in between $\overset{*}{\tau}_N$ and $\overset{*}{\tau}_{N+1}$, all the nodes other than the N-th and the $(N+1)$-th nodes should have almost zero activity.

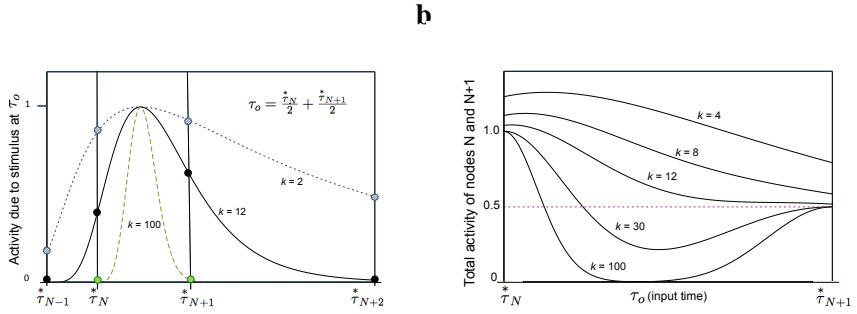

Figure 5: **a**. The activity of four successive fuzzy memory nodes $N-1$, N, $N+1$, and $N+2$ in response to a delta function input at a past moment τ_o that falls right in between the timescales of the N-th and the $(N+1)$-th nodes. The nodes are chosen according to the distribution given by Equation 13 with $c = 1$. **b**. The sum of activity of the N-th and $(N+1)$-th nodes in response to a delta function input at various times τ_o ranging between the timescales of N-th and $(N+1)$-th nodes. For each k, the activities are normalized to have values in the range of 0 to 1.

Figure 5a plots the activity of the four successive nodes with $c = 1$, when τ_o is exactly in the middle of $\overset{*}{\tau}_N$ and $\overset{*}{\tau}_{N+1}$. For each value of k, the activity is normalized so that it lies between 0 and 1. The four vertical lines represent the 4 nodes and the dots represent the activity of the corresponding nodes. Note that for $k = 2$ the activity of all 4 nodes is substantially different from zero, implying a significant information redundancy. At the other extreme, $k = 100$, the activity of all the nodes are almost zero, implying that the information about the delta function input at time $\tau_o = (\overset{*}{\tau}_N + \overset{*}{\tau}_{N+1})/2$ has been lost. To minimize both the information loss and the information redundancy, the value of k should be neither too large nor too small. Note that for $k = 12$, the activities of the $(N-1)$-th and the $(N+2)$-th nodes are almost zero, but activities of the N-th and $(N+1)$-th nodes are non-zero.

For any given c, a rough estimate of the appropriate k can be obtained by matching the difference in the $\overset{*}{\tau}$ values of the neighboring nodes to the smear σ from Equation 11.

$$\sigma = \frac{|\overset{*}{\tau}_{N+1}|}{\sqrt{k-2}} \left[\frac{k}{k-1}\right] \simeq |\overset{*}{\tau}_{N+1} - \overset{*}{\tau}_N| \qquad \Rightarrow \qquad \frac{k}{(k-1)\sqrt{k-2}} \simeq \frac{c}{1+c}. \tag{15}$$

This condition implies that a large value of k will be required when c is small and a small value of k will be required when c is large. In particular, Equation 15 suggests that $k \simeq 8$ when $c = 1$, which will be the parameters we pick for the demonstrations in Section 4.

To further illustrate the information loss at high values of k, Figure 5b shows the sum of activity of the N-th and the $(N+1)$-th nodes for all values of τ_o between $\overset{*}{\tau}_N$ and $\overset{*}{\tau}_{N+1}$. For each k, the activities are normalized so that the N-th node attains 1 when $\tau_o = \overset{*}{\tau}_N$. Focusing on the case of $k = 100$ in Figure 5b, there is a range of τ_o values for which the total activity of the two nodes is very close to zero. The input is represented by the N-th node when τ_o is close to $\overset{*}{\tau}_N$, and is represented by the $(N+1)$-th node when τ_o is close to $\overset{*}{\tau}_{N+1}$, but at intermediate values of τ_o the input is not represented by any node. One way to avoid such information loss is to require that the

total activity of the two nodes not have a local minimum—in other words the minimum should be at the boundary, at $\tau_o = \overset{*}{\tau}_{N+1}$. This is apparent in Figure 5b for k =4, 8 and 12. For $c = 1$, it turns out that there exists a local minimum in the summed activity of the two nodes only for values of k greater than 12. For any given c, the appropriate value of k that simultaneously minimizes the information redundancy and information loss is the maximum value of k for which a plot similar to Figure 5b will not have a local minimum.

In summary, the fuzzy memory system is the set of **T** column nodes with $\overset{*}{\tau}$ values given by Equation 13, with the value of k appropriately matched with c to minimize information redundancy and information loss.

4. Time Series Forecasting

We compare the performance of the self-sufficient fuzzy memory to a shift register in time series forecasting with a few simple illustrations. Our goal here is to illustrate the differences between a simple shift register and the self-sufficient fuzzy memory. Because our interest is in representation of the time series and not in the sophistication of the learning algorithm, we use simple linear regression algorithm to learn and forecast these time series.

We consider three time series with different properties. The first was generated by fractionally integrating white noise (Wagenmakers et al., 2004) in a manner similar to that described in Section 2. The second and third time series were obtained from the online library at http://datamarket.com. The second time series is the mean annual temperature of the Earth from the year 1781 to 1988. The third time series is the monthly average number of sunspots from the year 1749 to 1983 measured from Zurich, Switzerland. These three time series are plotted in the top row of Figure 6. The corresponding two point correlation function of each series is plotted in the middle row of Figure 6. Examination of the two point correlation functions reveal differences between the series. The fractionally-integrated noise series shows long-range correlations falling off like a power law. The temperature series shows correlations near zero (but modestly positive) over short ranges and weak negative correlation over longer times. The sunspots data has both strong positive short-range autocorrelation and a longer range negative correlation, balanced by a periodicity of 130 months corresponding to the 11 year solar cycle.

4.1 Learning and Forecasting Methodology

Let $N_{\max}$ denote the total number of nodes in the memory representation and let N be an index corresponding to each node ranging from 1 to $N_{\max}$. We shall denote the value contained in the nodes at any time step i by $B_i[N]$. The time series was sequentially fed into both the shift register and the self-sufficient fuzzy memory and the representations were evolved appropriately at each time step. The values in the shift register nodes were shifted downstream at each time step as discussed Section 2. At any instant the shift register held information from exactly $N_{\max}$ time steps in the past. The values in the self-sufficient fuzzy memory were evolved as described in Section 3, with $\overset{*}{\tau}$ values taken to be 1, 2, 4, 8, 16, 32,...$2^{(N_{\max}-1)}$, conforming to Equation 13 with $\overset{*}{\tau}_{min} = 1$, $c = 1$ and $k = 8$.

At each time step i, the value from each of the nodes $B_i[N]$ was recorded along with the value of the time series at that time step, denoted by V_i. We used a simple linear regression algorithm to extract the intercept I and the regression coefficients R_N so that the predicted value of the time series

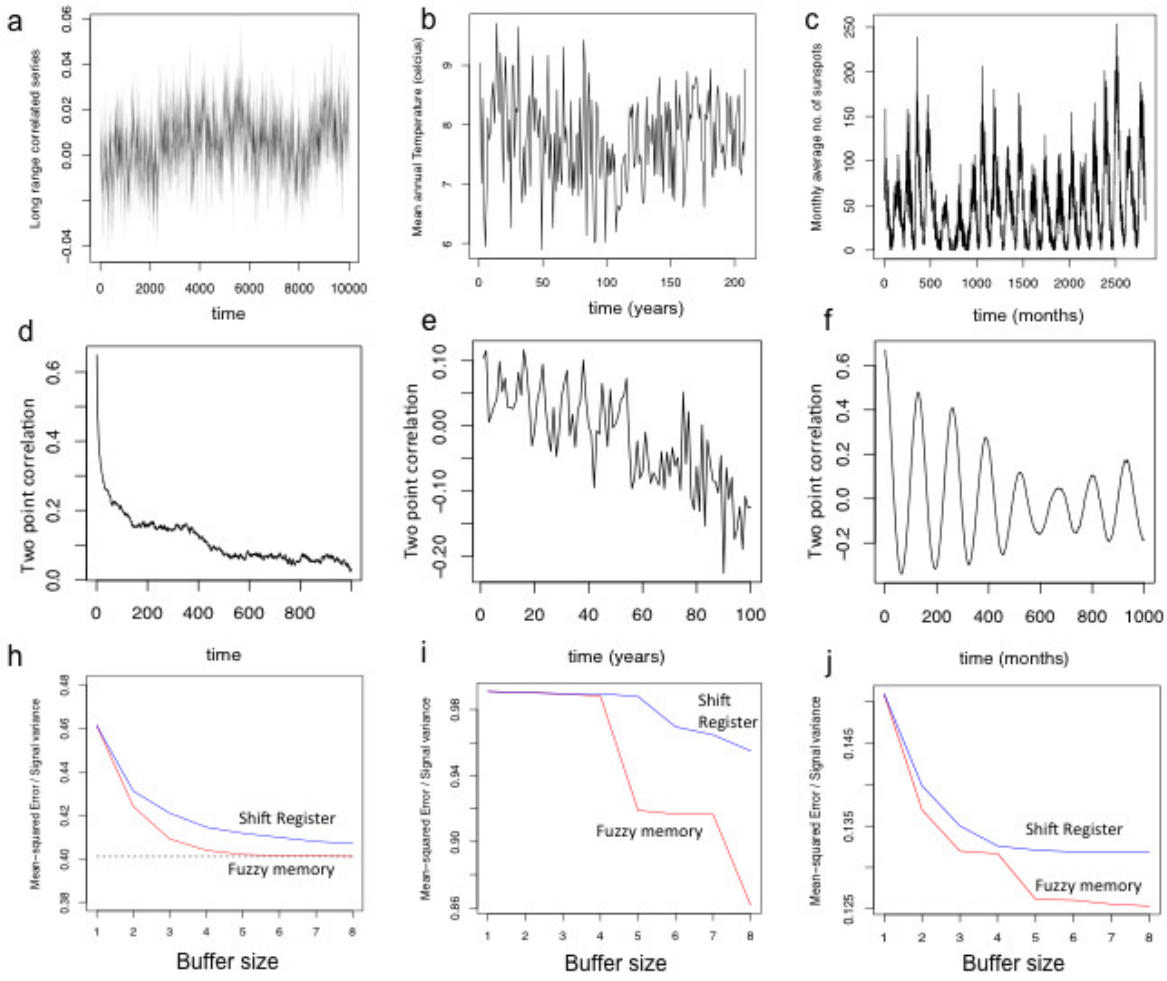

Figure 6: (**a**) Simulated time series with long range correlations based on ARFIMA model with $d = 0.4$, and white noise of standard deviation 0.01. (**b**) Average annual temperature of the Earth from the year 1781 to 1988. (**c**) Monthly average number of sunspots from the year 1749 to 1983. (**d**,**e**,**f**) Two point correlations of the series in **a**, **b** and **c**. (**h**,**i**,**j**) Error in forecasting the series **a**, **b** and **c** using either the fuzzy memory or the the shift register.

at each time step P_i and the squared error in prediction E_i are

$$P_i = I + \sum_{N=1}^{N_{\max}} R_N B_i[N], \qquad E_i = [P_i - V_i]^2.$$

The regression coefficients were extracted by minimizing the total squared error $E = \sum_i E_i$. For this purpose, we used a standard procedure `lm()` in the open source software R.

The accuracy of forecast is inversely related to the total squared error E. To get an absolute measure of accuracy we have to factor out the intrinsic variability of the time series. In the bottom row of Figure 6, we plot the mean of the squared error divided by the intrinsic variance in the time series $\mathrm{Var}[V_i]$. This quantity would range from 0 to 1; the closer it is to zero, the more accurate the prediction.

4.1.1 LONG RANGE CORRELATED SERIES

The long range correlated series (Figure 6a) is by definition constructed to yield a two point correlation that decays as a power law. This is evident from its two point correlation in Figure 6d that is decaying, but always positive. Since the value of the series at any time step is highly correlated with its value at the previous time step, we can expect to generate a reasonable forecast using a single node that holds the value from the previous time step. This can be seen from Figure 6h, where the error in forecast is only 0.46 with a single node. Adding more nodes reduces the error for both the shift register and the self-sufficient fuzzy memory. But for a given number of nodes, the fuzzy memory always has a lower error than the shift register. This can be seen from Figure 6h where the curve corresponding to the fuzzy memory falls below that of the shift register.

Since this series is generated by fractionally integrating white noise, the mean squared error cannot in principle be lower than the variance of the white noise used for construction. That is, there is a lower bound for the error that can be achieved in Figure 6h. The dotted line in Figure 6h indicates this bound. Note that the fuzzy memory approaches this bound with a smaller number of nodes than the shift register.

4.1.2 TEMPERATURE SERIES

The temperature series (Figure 6b) is much more noisy than the long range correlated series, and seems structureless. This can be seen from the small values of its two point correlations in Figure 6e. This is also reflected in the fact that with a small number of nodes, the error is very high. Hence it can be concluded that no reliable short range correlation exist in this series. That is, knowing the average temperature during a given year does not help much in predicting the average temperature of the subsequent year. However, there seems to be a weak negative correlation at longer scales that could be exploited in forecasting. Note from Figure 6i that with additional nodes the fuzzy memory performs better at forecasting and has a lower error in forecasting than a shift register. This is because the fuzzy memory can represent much longer timescales than the shift register of equal size, and thereby exploit the long range correlations that exist.

4.1.3 SUNSPOTS SERIES

The sunspot series (Figure 6c) is less noisy than the other two series considered, and it has an oscillatory structure of about 130 month periodicity. It has high short range correlations, and hence

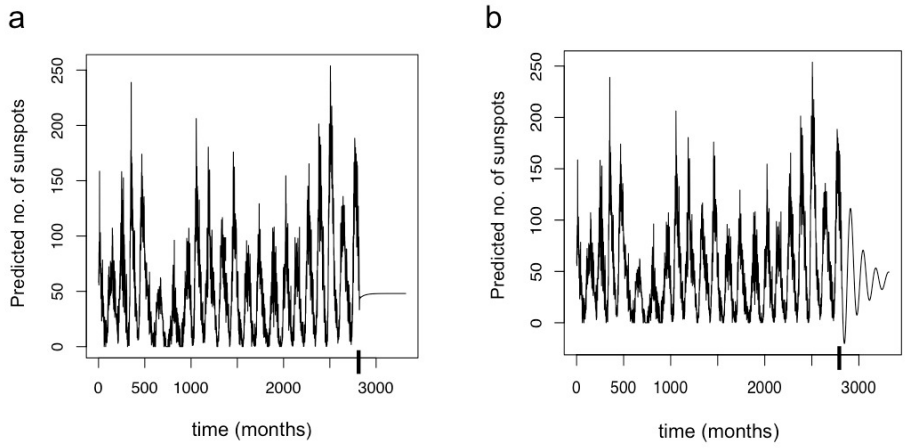

Figure 7: Forecasting the distant future. The sunspots time series of length 2820 is extrapolated for 500 time steps in the future using **(a)** shift register with 8 nodes, and **(b)** fuzzy memory with 8 nodes. The solid tick mark on the x-axis at 2820 corresponds to the point where the original series ends and the predicted future series begins.

even a single node that holds the value from the previous time step is sufficient to forecast with an error of 0.15, as seen in Figure 6j. As before, with more nodes, the fuzzy memory consistently has a lower error in forecasting than the shift register with equal number of nodes. Note that when the number of nodes is increased from 4 to 8, the shift register does not improve in accuracy while the fuzzy memory continues to improve in accuracy.

With a single node, both fuzzy memory and shift register essentially just store the information from the previous time step. Because most of the variance in the series can be captured by the information in the first node, the difference between the fuzzy memory and the shift register with additional nodes is not numerically overwhelming when viewed in Figure 6j. However, there is a qualitative difference in the properties of the signal extracted by the two memory systems. In order to successfully learn the 130 month periodicity, the information about high positive short range correlations is not sufficient, it is essential to also learn the information about the negative correlations at longer time scales. From Figure 6f, note that the negative correlations exist at a timescale of 50 to 100 months. Hence in order to learn this information, these timescales have to be represented. A shift register with 8 nodes cannot represent these timescales but the fuzzy memory with 8 nodes can.

To illustrate that it is possible to learn the periodicity using the fuzzy memory, we forecast the distant future values of the series. In Figure 7, we extend the sunspots series by predicting it for a future of 500 months. The regression coefficients R_N and the intercept I are extracted from the original series of length 2820. For the next 500 time steps, the predictions P_i are treated as actual values V_i, and the memory representations are evolved. Figure 7a shows the series generated using shift register with 8 nodes. The solid tick mark on the x-axis at 2820 represents the point at which

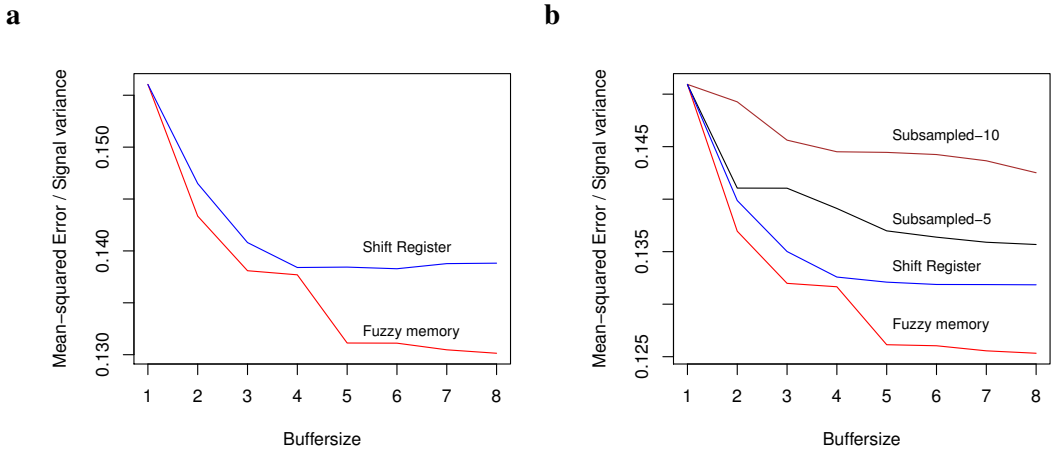

Figure 8: **a.** Testing error in forecasting. The regression coefficients were extracted from the first half of the sunspot series and testing was performed on the second half of the series. **b.** Forecasting error of the fuzzy memory, shift register and subsampled shift register with a node spacing of 5 and 10.

the original series ends and the predicted future series begins. Note that the series forecasted by the shift register immediately settles on the mean value without oscillation. This is because the time scale at which the oscillations are manifest is not represented by the shift register with 8 nodes. Figure 7b shows the series generated by the fuzzy memory with 8 nodes. Note that the series predicted by the fuzzy memory continues in an oscillating fashion with decreasing amplitude for several cycles eventually settling at the mean value. This is possible because the fuzzy memory represents the signal at a sufficiently long time scale to capture the negative correlations in the two-point correlation function.

Of course, a shift register with many more nodes can capture the long-range correlations and predict the periodic oscillations in the signal. However the number of nodes necessary to describe the oscillatory nature of the signal needs to be of the order of the periodicity of the oscillation, about 130 in this case. This would lead to overfitting the data. At least in the case of the simple linear regression algorithm, the number of regression coefficients to be extracted from the data increases with the number of nodes, and extracting a large number of regression coefficients from a finite data set will unquestionably lead to overfitting the data. Hence it would be ideal to use the least number of nodes required to span the relevant time scale.

In order to ensure that the extracted regression coefficients has not overfitted the data, we split the sunspots time series into two halves. We extracted the regression coefficients by using only the first half for training and used the second half for testing the predictions generated by those coefficients. Figure 8a plots this testing error, and should be compared to the training error plotted in Figure 6j. Other than the noticeable fact that the testing error is slightly higher than the training error, the shape of the two plots are very similar for both fuzzy memory and the shift register.

If our goal was to only capture the oscillatory structure of the sunspot series within a small number of regression coefficients, then we could subsample from a lengthy shift register so that

information from both positive and negative correlations can be obtained. Although the subsampled shift register contains relatively few nodes, it cannot self sufficiently evolve in real time; we would need the resources associated with the complete shift register in order to evolve the memory at each moment. By subsampling 8 equidistantly spaced nodes of the shift register 1,11,21,31,...81, and extracting the corresponding regression coefficients, it is possible to extend the series to have an oscillatory structure analogous to Figure 7b. However it turns out that the forecasting error for the subsampled shift register is significantly higher than the forecasting error from the fuzzy memory. Figure 8b shows the forecasting error for subsampled shift register with equidistant node spacing of 5 and 10. Even though the subsampled shift register with a node spacing of 10 extends over a similar temporal range as the fuzzy memory, and captures the oscillatory structure in the data, the fuzzy memory outperforms it with a lower error. The advantage of the fuzzy memory over the subsampled shift register comes from the property of averaging over many previous values at long time scales rather than picking a single noisy value and using that for prediction. This property helps to suppress unreliable fluctuations that could lead to overfitting the data.

5. Discussion

The fuzzy memory holds more predictively relevant information than a shift register with the same number of nodes for long-range correlated signals, and hence performs better in time series forecasting such signals. However, learning the relevant statistics from a lengthy time series is not the same as learning from very few learning trials. To learn from very few learning trials, a learner must necessarily make some generalizations based on some built-in assumptions about the environment. Since the fuzzy memory discards information about the precise time of a stimulus presentation, the temporal inaccuracy in memory can help the learner make such a generalization. Suppose it is useful for a learner to learn the temporal relationship between two events, say A and B. Let the statistics of the world be such that B consistently follows A after a delay period, which on each learning trial is chosen from an unknown distribution. After many learning trials, a learner relying on a shift register memory would be able to sample the entire distribution of delays and learn it precisely. But real world learners may have to learn much faster. Because the fuzzy memory system represents the past information in a smeared fashion, a single training sample from the distribution will naturally let the learner make a scale-free temporal generalization about the distribution of delays between A and B. The temporal profile of this generalization will not in general match the true distribution that could be learned after many learning trials, however the fact that it is available after a single learning trial provides a tremendous advantage for natural learners.

It then seems natural to wonder if human and animal memory resembles the fuzzy memory system. After all, animals have evolved in the natural world where predicting the imminent future is crucial for survival. Numerous behavioral findings on animals and humans are consistent with them having a memory system with scale-free representation of past events (Balsam and Gallistel, 2009; Gallistel and Gibbon, 2000). In human memory studies, the forgetting curve is usually observed to follow a scale invariant power law function (Donkin and Nosofsky, 2012). When humans are asked to reproduce or discriminate time intervals, they exhibit a characteristic scale-invariance in the errors they produce (Rakitin et al., 1998; Wearden and Lejeune, 2008). This is not just a characteristic feature in humans, but in a wide variety of animals like rats, rabbits and pigeons (Roberts, 1981; Smith, 1968). These findings across behavioral tasks and species suggest that a scale-free memory is an adaptive response to a world with structure at many temporal scales.

5.1 Neural Networks with Temporal Memory

Let us now consider the fuzzy memory system in the context of neural networks with temporal memory. It has been realized that neural networks with generic recurrent connectivity can have sufficiently rich dynamics to hold temporal memory of the past. Analogous to how the ripple patterns on a liquid surface contains information about the past perturbations, the instantaneous state of the recurrent network holds the memory of the past which can be simply extracted by training a linear readout layer. Such networks can be implemented either with analog neurons-echo state networks (Jaeger, 2001), or with spiking neurons-liquid state machines (Maass et al., 2002). They are known to be non-chaotic and dynamically stable as long as their spectral radius or the largest eigenvalue of the connectivity matrix has a magnitude less than one. Abstractly, such networks with fixed recurrent connectivity can be viewed as a reservoir of nodes and can be efficiently used for computational tasks involving time varying inputs, including time series prediction (Wyffels and Schrauwen, 2010).

The timescale of these reservoirs can be tuned up by introducing leaky integrator neurons in them (Jaeger et al., 2007). However, a reservoir with finite nodes cannot have memory from infinite past. In fact the criterion for dynamical stability of the reservoir is equivalent to requiring a fading memory (Jaeger, 2002). If we define a memory function of the reservoir to be the precision with which inputs from each past moment can be reconstructed, it turns out that the net memory, or the area under the memory function over all past times, is bounded by the number of nodes in the reservoir $N_{\max}$. The exact shape of the memory function will however depend on the connectivity within reservoir. For a simple shift register connectivity, the memory function is a step function which is 1 up to $N_{\max}$ time steps in the past and zero beyond $N_{\max}$. But for a generic random connectivity the memory function decays smoothly, sometimes exponentially and sometimes as a power law depending on the spectral radius (Ganguli et al., 2008). For linear recurrent networks, it turns out that the memory function is analytically tractable at least in some special cases (White et al., 2004; Hermans and Schrauwen, 2010), while the presence of any nonlinearity seems to reduce the net memory (Ganguli et al., 2008). By increasing the number of nodes in the reservoir, the net memory can be increased. But unless the network is well tailored, as in orthogonal networks (White et al., 2004) or a divergent feed-forward networks (Ganguli et al., 2008), the net memory grows very slowly and sub-linearly with the number of nodes. Moreover, analysis of trajectories in the state-space of a randomly connected network suggests that the net memory will be very low when the connectivity is dense (Wallace et al., 2013).

The self-sufficient fuzzy memory can be viewed as a special case of a linear reservoir with a specific, tailored connectivity. The $\mathbf{t}$ nodes effectively have a diagonal connectivity matrix making them leaky integrators, and the $\mathbf{L}_k^{-1}$ is the linear readout weights that approximately extracts the past inputs. For a white noise input signal, the memory function decays as a power law with exponent -1, and the net memory grows linearly with the number of nodes. However, as described above, the accuracy of reconstruction of the past is not the relevant quantity of interest here, it is the predictive information from the past that is of interest. Scale-free fluctuations in natural world imply that it isn't necessary to be accurate; in fact sacrificing accuracy in a scale-free fashion lets us represent predictive information from exponentially long timescales.

5.2 Unaddressed Issues

Two basic issues essential to real-world machine learning applications have been ignored in this work for the sake of theoretical simplicity. First is that we have simply focused on a scalar time varying signal, while any serious learning problem would involve multidimensional signals. When unlimited storage resources are available, each dimension can be separately represented in a lengthy shift register. To conserve storage resources associated with the time dimension, we could replace the shift register with the self-sufficient fuzzy memory. This work however does not address the issue of conserving storage resources by compressing the intrinsic dimensionality of the signal. In general, most of the information relevant for future prediction is encoded in few combinations of the various dimensions of the signal, called features. Techniques based on information bottleneck method (Creutzig and Sprekeler, 2008; Creutzig et al., 2009) and slow feature analysis (Wiskott and Sejnowski, 2002) can efficiently extract these features. The strategy of representing the time series in a scale invariantly fuzzy fashion could be seen as complementary to these techniques. For instance, slow feature analysis (Wiskott and Sejnowski, 2002) imposes the slowness principle where low-dimensional features that change most slowly are of interest. If the components of a time varying high dimensional signal is represented in a temporally fuzzy fashion rather than in a shift register, then we could potentially extract the slowly varying parts in an online fashion by examining differences in the activities of the largest two $\overset{*}{\tau}$ nodes.

The second issue is that we have ignored the learning and prediction mechanisms while simply focusing on the memory representation. For simplicity we used the linear regression predictor in Section 4. Any serious application should involve the ability to learn nonlinearities. Support vector machines (SVM) adopt an elegant strategy of using nonlinear kernel functions to map the input data to a high dimensional space where linear methods can be used (Vapnik, 1998; Müller et al., 1997). The standard method for training SVMs on time series prediction requires feeding in the data from a sliding time window, in other words providing shift registers as input. It has recently been suggested that rather than using standard SVM kernels on sliding time window, if we used recurrent kernel functions corresponding to infinite recurrent networks, performance can be improved on certain tasks (Hermans and Schrauwen, 2012). This suggests that the gradually fading temporal memory of the recurrent kernel functions is more effective than the step-function memory of shift register used in standard SVMs for time series prediction. Training SVMs with standard kernel functions along with fuzzy memory inputs rather than shift register inputs is an alternative strategy for approaching problems involving signals with long range temporal correlations. Moreover, since $\mathbf{t}$ nodes contain all the temporal information needed to construct the fuzzy memory, directly training the SVMs with inputs from $\mathbf{t}$ nodes could also be very fruitful.

Finally, it should be noted that if our aim is to build an autonomous agent we need both learning and prediction to happen in an online fashion. Many widely-used machine learning algorithms like SVM (Vapnik, 1998) and deep learning networks (Hinton et al., 2006), rely on batch processing which requires the availability of the entire data set prior to learning. Autonomous agents with limited memory resources cannot adopt such learning strategies. The learning mechanism cannot rely on information other than what is instantaneously available in the memory. An online learning algorithm tailored to act on the fuzzy memory representation could potentially be very useful for autonomous agents with finite memory resources.

6. Conclusion

Signals with long-range temporal correlations are found throughout the natural world. Such signals present a distinct challenge to machine learners that rely on a shift-register representation of the time series. Here we have described a method for constructing a self-sufficient scale-free representation of temporal history. The nodes are chosen in a way that minimizes information redundancy and information loss while equally distributing them over all time scales. Although the temporal accuracy of the signal is sacrificed, predictively relevant information from exponentially long timescales is available in the fuzzy memory system when compared to a shift register with the same number of nodes. This could be an extremely useful way to represent time series with long-range correlations for use in machine learning applications.

Acknowledgments

The authors acknowledge support from National Science Foundation grant, NSF BCS-1058937, and Air Force Office of Scientific Research grant AFOSR FA9550-12-1-0369.

Appendix A. Information Redundancy Across Nodes

The information redundancy in the memory representation can be quantified by deriving expressions for mutual information shared between neighboring nodes. When the input signal is uncorrelated or has scale-free long range correlations, it will be shown that the information redundancy is equally spread over all nodes only when the $\overset{*}{\tau}$ values of the nodes are given by Equation 13.

Taking $\mathbf{f}(\tau)$ to be a stochastic signal and the current moment to be $\tau = 0$, the activity of a $\overset{*}{\tau}$ node in the $\mathbf{T}$ column is (see Equation 12)

$$\mathbf{T}(0,\overset{*}{\tau}) = \frac{k^{k+1}}{k!}\int_{-\infty}^{0}\frac{1}{|\overset{*}{\tau}|}\left(\frac{\tau'}{\overset{*}{\tau}}\right)^{k} e^{-k\left(\frac{\tau'}{\overset{*}{\tau}}\right)}\mathbf{f}(\tau')d\tau'.$$

The expectation value of this node can be calculated by simply averaging over $\mathbf{f}(\tau')$ inside the integral, which should be a constant if it is generated by a stationary process. By defining $z = \tau'/\overset{*}{\tau}$, we find that the expectation of $\mathbf{T}$ is proportional to the expectation of $\mathbf{f}$.

$$\left\langle \mathbf{T}(0,\overset{*}{\tau})\right\rangle = \left\langle \mathbf{f}\right\rangle \frac{k^{k+1}}{k!}\int_{0}^{\infty} z^{k}e^{-kz}dz.$$

To understand the information redundancy in terms of correlations among the nodes, we calculate the correlations among the $\mathbf{T}$ nodes when $\mathbf{f}(\tau)$ is either a white noise or a long-range correlated signal.

A.1 White Noise Input

Let $\mathbf{f}(\tau)$ to be white noise, that is $\langle \mathbf{f} \rangle = 0$ and $\langle \mathbf{f}(\tau)\mathbf{f}(\tau') \rangle \sim \delta(\tau - \tau')$. The variance in the activity of each $\overset{*}{\tau}$ node is then given by

$$\begin{aligned}
\langle \mathbf{T}^2(0,\overset{*}{\tau}) \rangle &= \left(\frac{k^{k+1}}{k!}\right)^2 \int_{-\infty}^{0}\int_{-\infty}^{0} \frac{1}{|\overset{*}{\tau}|^2} \left(\frac{\tau}{\overset{*}{\tau}}\right)^k e^{-k\left(\frac{\tau}{\overset{*}{\tau}}\right)} \left(\frac{\tau'}{\overset{*}{\tau}}\right)^k e^{-k\left(\frac{\tau'}{\overset{*}{\tau}}\right)} \langle \mathbf{f}(\tau)\mathbf{f}(\tau') \rangle d\tau d\tau', \\
&= \frac{1}{|\overset{*}{\tau}|} \left(\frac{k^{k+1}}{k!}\right)^2 \int_0^\infty z^{2k} e^{-2kz} dz. \qquad (16)
\end{aligned}$$

As expected, the variance of a large $|\overset{*}{\tau}|$ node is small because the activity in this node is constructed by integrating the input function over a large timescale. This induces an artificial temporal correlation in that node's activity which does not exist in the input function. To see this more clearly, we calculate the correlation across time in the activity of one node, at time τ and τ'. With the definition $\delta = |\tau - \tau'|/|\overset{*}{\tau}|$, it turns out that

$$\langle \mathbf{T}(\tau,\overset{*}{\tau})\ \mathbf{T}(\tau',\overset{*}{\tau}) \rangle = |\overset{*}{\tau}|^{-1} \left(\frac{k^{k+1}}{k!}\right)^2 e^{-k\delta} \sum_{r=0}^{k} \delta^{k-r} \frac{k!}{r!(k-r)!} \int_0^\infty z^{k+r} e^{-2kz} dz. \qquad (17)$$

Note that this correlation is nonzero for any $\delta > 0$, and it decays exponentially for large δ. Hence even a temporally uncorrelated white noise input leads to short range temporal correlations in a $\overset{*}{\tau}$ node. It is important to emphasize here that such temporal correlations will not be introduced in a shift register. This is because, in a shift register the functional value of $\mathbf{f}$ at each moment is just passed on to the downstream nodes without being integrated, and the temporal autocorrelation in the activity of any node will simply reflect the temporal correlation in the input function.

Let us now consider the instantaneous correlation in the activity of two different nodes. At any instant, the activity of two different nodes in a shift register will be uncorrelated in response to a white noise input. The different nodes in a shift register carry completely different information, making their mutual information zero. But in the $\mathbf{T}$ column, since the information is smeared across different $\overset{*}{\tau}$ nodes, the mutual information shared by different nodes is non-zero. The instantaneous correlation between two different nodes $\overset{*}{\tau}_1$ and $\overset{*}{\tau}_2$ can be calculated to be

$$\begin{aligned}
\langle \mathbf{T}(0,\overset{*}{\tau}_1)\ \mathbf{T}(0,\overset{*}{\tau}_2) \rangle &= |\overset{*}{\tau}_2|^{-1} \left(\frac{k^{k+1}}{k!}\right)^2 \int_0^\infty z^{2k} (\overset{*}{\tau}_1/\overset{*}{\tau}_2)^k e^{-kz(1+\overset{*}{\tau}_1/\overset{*}{\tau}_2)} dz, \\
&\propto \frac{(\overset{*}{\tau}_1 \overset{*}{\tau}_2)^k}{(|\overset{*}{\tau}_1| + |\overset{*}{\tau}_2|)^{2k+1}}.
\end{aligned}$$

The instantaneous correlation in the activity of the two nodes $\overset{*}{\tau}_1$ and $\overset{*}{\tau}_2$ is a measure of the mutual information represented by them. Factoring out the individual variances of the two nodes, we have the following measure for the mutual information.

$$I(\overset{*}{\tau}_1,\overset{*}{\tau}_2) = \frac{\langle \mathbf{T}(0,\overset{*}{\tau}_1)\ \mathbf{T}(0,\overset{*}{\tau}_2) \rangle}{\sqrt{\langle \mathbf{T}^2(0,\overset{*}{\tau}_1) \rangle \langle \mathbf{T}^2(0,\overset{*}{\tau}_2) \rangle}} \propto \left[\frac{\sqrt{\overset{*}{\tau}_1/\overset{*}{\tau}_2}}{(1+\overset{*}{\tau}_1/\overset{*}{\tau}_2)}\right]^{2k+1}.$$

This quantity is high when $\overset{*}{\tau}_1/\overset{*}{\tau}_2$ is close to 1. That is, the mutual information shared between neighboring nodes will be high when their $\overset{*}{\tau}$ values are very close.

The fact that the mutual information shared by neighboring nodes is non-vanishing implies that there is redundancy in the representation of the information. If we require the information redundancy to be equally distributed over all the nodes, then we need the mutual information between any two neighboring nodes to be a constant. If $\overset{*}{\tau}_1$ and $\overset{*}{\tau}_2$ are any two neighboring nodes, then in order for $I(\overset{*}{\tau}_1,\overset{*}{\tau}_2)$ to be a constant, $\overset{*}{\tau}_1/\overset{*}{\tau}_2$ should be a constant. This can happen only if the $\overset{*}{\tau}$ values of the nodes are arranged in the form given by Equation 13.

A.2 Long Range Correlated Input

Now consider $\mathbf{f}(\tau)$ such that $\left\langle \mathbf{f}(\tau)\mathbf{f}(\tau')\right\rangle \sim 1/|\tau-\tau'|^{\alpha}$ for large values of $|\tau-\tau'|$. Reworking the calculations analogous to those leading to Equation 17, we find that the temporal correlation is

$$\left\langle \mathbf{T}(\tau,\overset{*}{\tau})\ \mathbf{T}(\tau',\overset{*}{\tau})\right\rangle = \frac{|\overset{*}{\tau}|^{-\alpha}}{2.4^k}\left(\frac{k^{k+1}}{k!}\right)^2 \sum_{r=0}^{k} C_r \int_{-\infty}^{\infty} \frac{|v|^{k-r}}{|v+\delta|^{\alpha}} e^{-k|v|}\, dv.$$

Here $\delta = |\tau-\tau'|/|\overset{*}{\tau}|$ and $C_r = \frac{k!k+r)!}{r!(k-r)!}\frac{2^{k-r}}{(k)^{k+r+1}}$. The exact value of C_r is unimportant and we only need to note that it is a positive number.

For $\alpha > 1$, the above integral diverges at $v = -\delta$, however we are only interested in the case $\alpha < 1$. When δ is very large, the entire contribution to the integral comes from the region $|v| \ll \delta$ and the denominator of the integrand can be approximated as δ^{α}. In effect,

$$\left\langle \mathbf{T}(\tau,\overset{*}{\tau})\ \mathbf{T}(\tau',\overset{*}{\tau})\right\rangle \sim |\overset{*}{\tau}|^{-\alpha}\delta^{-\alpha} = |\tau-\tau'|^{-\alpha}$$

for large $|\tau-\tau'|$. The temporal autocorrelation of the activity of any node should exactly reflect the temporal correlations in the input when $|\tau-\tau'|$ is much larger than the time scale of integration of that node ($\overset{*}{\tau}$). As a point of comparison, it is useful to note that any node in a shift register will also exactly reflect the correlations in the input.

Now consider the instantaneous correlations across different nodes. The instantaneous correlation between two nodes $\overset{*}{\tau}_1$ and $\overset{*}{\tau}_2$ turns out to be

$$\left\langle \mathbf{T}(0,\overset{*}{\tau}_1)\ \mathbf{T}(0,\overset{*}{\tau}_2)\right\rangle = |\overset{*}{\tau}_2|^{-\alpha}\left(\frac{k^{k+1}}{k!}\right)^2 \sum_{r=0}^{k} X_r \frac{\beta^{k-r}(1+\beta^{r-\alpha+1})}{(1+\beta)^{2k-r+1}}. \tag{18}$$

Here $\beta = |\overset{*}{\tau}_1|/|\overset{*}{\tau}_2|$ and each X_r is a positive coefficient. By always choosing $|\overset{*}{\tau}_2| \geq |\overset{*}{\tau}_1|$, we note two limiting cases of interest, $\beta \ll 1$ and $\beta \simeq 1$. When $\beta \ll 1$, the $r = k$ term in the summation of the above equation yields the leading term, and the correlation is simply proportional to $|\overset{*}{\tau}_2|^{-\alpha}$, which is approximately equal to $|\overset{*}{\tau}_2 - \overset{*}{\tau}_1|^{-\alpha}$. In this limit where $|\overset{*}{\tau}_2| \gg |\overset{*}{\tau}_1|$, the correlation between the two nodes behaves like the correlation between two shift register nodes. When $\beta \simeq 1$, note from Equation 18 that the correlation will still be proportional to $|\overset{*}{\tau}_2|^{-\alpha}$. Now if $\overset{*}{\tau}_1$ and $\overset{*}{\tau}_2$ are neighboring nodes with close enough values, we can evaluate the mutual information between them to be

$$I(\overset{*}{\tau}_1,\overset{*}{\tau}_2) = \frac{\left\langle \mathbf{T}(0,\overset{*}{\tau}_1)\ \mathbf{T}(0,\overset{*}{\tau}_2)\right\rangle}{\sqrt{\left\langle \mathbf{T}^2(0,\overset{*}{\tau}_1)\right\rangle\left\langle \mathbf{T}^2(0,\overset{*}{\tau}_2)\right\rangle}} \propto |\overset{*}{\tau}_2/\overset{*}{\tau}_1|^{-\alpha/2}.$$

Reiterating our requirement from before that the mutual information shared by neighboring nodes at all scales should be the same, we are once again led to choose $\overset{*}{\tau}_2/\overset{*}{\tau}_1$ to be a constant which is possible only when the $\overset{*}{\tau}$ values of the nodes are given by Equation 13.

References

R. T. Baillie. Long memory processes and fractional integration in econometrics. *Journal of Econometrics*, 73:5–59, 1996.

P. D. Balsam and C. R. Gallistel. Temporal maps and informativeness in associative learning. *Trends in Neuroscience*, 32(2):73–78, 2009.

J. Beran. *Statistics for Long-Memory Processes*. Chapman & Hall, New York, 1994.

G. Chechik, A. Globerson, N. Tishby, and Y. Weiss. Information bottleneck for gaussian variables. *Journal of Machine Learning Research*, 6:165–188, 2005.

F. Creutzig and H. Sprekeler. Predictive coding and the slowness principle: An information-theoretic approach. *Neural Computation*, 20(4):1026–1041, 2008.

F. Creutzig, A. Globerson, and N. Tishby. Past-future information bottleneck in dynamical systems. *Physical Review E*, 79:041925, 2009.

C. Donkin and R. M. Nosofsky. A power-law model of psychological memory strength in short- and long-term recognition. *Psychological Science*, 23:625–634, 2012.

D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of Optical Society of America A*, 4:2379–2394, 1987.

C. R. Gallistel and J. Gibbon. Time, rate, and conditioning. *Psychological Review*, 107(2):289–344, 2000.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 105(48):18970–18975, 2008.

D. L. Gilden. Cognitive emissions of $1/f$ noise. *Psychological Review*, 108:33–56, 2001.

C. W. J. Granger and R. Joyeux. An introduction to long-memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1:15–29, 1980.

M. Hermans and B. Schrauwen. Memory in linear recurrent neural networks in continuous time. *Neural Networks*, 23(3):341–355, 2010.

M. Hermans and B. Schrauwen. Recurrent kernel machines: Computing with infinite echo state networks. *Neural Computation*, 24(1):104–133, 2012.

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

H. Jaeger. The echo state approach to analyzing and training recurrent networks. GMD-Report 148, GMD - German National Research Institute for Information Technology, 2001.

H. Jaeger. Short term memory in echo state networks. GMD-Report 152, GMD - German National Research Institute for Information Technology, 2002.

H. Jaeger, M. Lukosevicius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky integrator neurons. *Neural Networks*, 20:335–352, 2007.

K. Linkenkaer-Hansen, V.V. Nikouline, J. M. Palva, and R. J. Ilmoniemi. Long-range temporal correlations and scaling behavior in human brain oscillations. *Journal of Neuroscience*, 21:1370–1377, 2001.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, San Fransisco, CA, 1982.

K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In *Proceedings of the International Conference on Analog Neural Networks*, 1997.

E. Post. Generalized differentiation. *Transactions of the American Mathematical Society*, 32:723–781, 1930.

B. C. Rakitin, J. Gibbon, T. B. Penny, C. Malapani, S. C. Hinton, and W. H. Meck. Scalar expectancy theory and peak-interval timing in humans. *Journal of Experimental Psychology: Animal Behavior Processes*, 24:15–33, 1998.

S. Roberts. Isolation of an internal clock. *Journal of Experimental Psychology: Animal Behavior Processes*, 7:242–268, 1981.

K. H. Shankar and M. W. Howard. A scale-invariant internal representation of time. *Neural Computation*, 24:134–193, 2012.

M. C. Smith. CS-US interval and US intensity in classical conditioning of rabbit's nictitating membrane response. *Journal of Comparative and Physiological Psychology*, 66(3):679–687, 1968.

N. Tishby, F. C. Pereira, and W. Bialek. Information bottleneck method. In *Proceedings of 37th Allerton Conference on Communication and Computation*, Monticello, IL, 1999.

A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391–412, 2003.

G. C. Van Orden, J. G. Holden, and M. T. Turvey. Self organization of cognitive performance. *Journal of Experimental Psychology: General*, 132:331–350, 2003.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

R. F. Voss and J. Clarke. $1/f$ noise in music and speech. *Nature*, 258:317–318, 1975.

E. J. Wagenmakers, S. Farrell, and R. Ratcliff. Estimation and interpretation of $1/f^{\alpha}$ noise in human cognition. *Psychonomic Bulletin & Review*, 11(4):579–615, 2004.

E. Wallace, H. R. Maei, and P. E. Latham. Randomly connected networks have short temporal memory. *Neural Computation*, 25:1408–1439, 2013.

J. H. Wearden and H. Lejeune. Scalar properties in human timing: conformity and violations. *Quarterly Journal of Experimental Psychology*, 61:569–587, 2008.

O. L. White, D. D. Lee, and H. Sompolinsky. Short-term memory in orthogonal neural networks. *Physical Review Letters*, 92(14):148102, 2004.

L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

F. Wyffels and B. Schrauwen. A comparative study of reservoir computing strategies for monthly time series prediction. *Neurocomputing*, 73:1958–1964, 2010.