Generic construction of scale-invariantly coarse grained memory

Karthik H. Shankar

1Center for Memory and Brain, Boston University

Encoding temporal information from the recent past as spatially distributed memory is essential to be able to instantly access the entire past. Any biological or synthetic agent that relies on the past to predict or plan the future, would be endowed with such a spatially distributed temporal memory. In natural settings where relevant signals show scale free temporal fluctuations, it is advantageous to have a memory system that scale invariently coarsely grains the past information while spatially distributing it. Here we examine the generic mechanism to encode the past in one spatial dimension with a specific emphasis on scale invariant coarse graining. We show that any mechanistic construction of such a memory system in the continuum limit is equivalent to encoding linear combinations of the Laplace transformed past and their approximated inverses. This mechanism is readily amenable to a discretized implementation in networks that maintain information from exponentially long past timescales within linearly increasing memory resources.

Representing the information from the recent past as transient activity spatially distributed over a network has been actively researched in biophysical as well as purely computational domains since the beginning of this century [1] [2]. It is understood that recurrent connections in the network can keep the information from distant past alive so that it can be recovered from the current state. The memory capacity of these networks are generally measured in terms of the accuracy of recovery of the past information [2] [4]. Although the memory capacity strongly depends on the network’s topology and sparsity [5] [8], it is known that it can be significantly increased by exploiting any prior knowledge of the underlying structure of the encoded signal, especially its temporal sparseness [9] [10].

Our approach to encoding the past as transient activity stems from a focus on the utility of memory in terms of its future relevance. The memory system should be evaluated not on the accuracy of recovery of the information from each past moment, rather on how well the represented past information can contribute to predicting the future. With this view, recent work [11] hints that for natural signals with long range temporal correlations and scale free fluctuations, coarse graining the past information in a scale invariant fashion would increase the predictively relevant information contained in a finite sized memory system. Such a memory system should represent the past as coarse grained over time windows that linearly scale with the past timescale. The accuracy with which the information from a particular past moment can be recovered will degrade as we go further into the past. But this feature can be viewed as essentially averaging over statistical fluctuations in the signal from the distant past whose accurate representation in memory will serve very little in predicting the future. In lay terms—it is not important to accurately remember whether an event occurred 101 seconds in the past or 110 seconds in the past, while it is important to remember whether the event occurred 1 second or 10 seconds in the past. Arguably, in the natural world ubiquitously filled with signals showing scale free fluctuations [12] [14], evolution would have pushed animals to adopt such a memory system conducive for future predictions. This is indeed evident from animal and human behavioral studies that show that our memory for time involves scale invariant errors which linearly scale with the target timescale [15] [16].

In this paper we shall assume the utility of a scale invariantly coarse grained memory system and only focus on analyzing the generic mechanism to construct it. It has previously been shown that one way to mechanistically construct such a memory system is to gradually encode information over real time as a Laplace transform of the past and approximately invert it [11] [17]. The primary result here is a generalization showing that any construction of such a memory system is essentially equivalent to encoding linear combinations of Laplace transformed past and their approximate inverses. Rather than considering a network with a discrete set of nodes, for analysis benefit, we consider a continuum limit where the information from the past time is smoothly projected on a spatial axis. The construction can however be discretized and implemented in a network with finite nodes representing coarse grained past information from timescales that exponentially scale with the network size.

Scale Invariant Coarse Graining

Consider a real valued function \( f(\tau) \) observed over time \( \tau \). The aim is to encode this time-varying function into a spatially distributed representation in one dimension parametrized by \( s \), such that at any moment \( \tau \) the entire past from \(-\infty\) to \( \tau \) is represented in a coarse grained fashion as \( T(\tau, s) \).

\[
T(\tau, s) = \int_{-\infty}^{\tau} f(\tau') W(\tau - \tau', s) \, d\tau',
\]  

where \( W(\ldots) \) is the coarse graining window function with normalized area for all \( s \), \( \int_{-\infty}^{\tau} W(\tau - \tau', s) \, d\tau' = 1 \). Different points on the spatial axis uniquely and monotonically represents coarse grained averages about different instants in the past. See figure 1.
For brevity, we denote the order of time derivative within $T$ to express the general space-time derivative of $t$.

We require that coarse graining about any past instant linearly scales with the past timescale. So, for any pair of points $s_1$ and $s_2$, there exists a scaling constant $\alpha_{12}$ such that $W(\tau - \tau', s_1) = \alpha_{12} W(\alpha_{12}(\tau - \tau'), s_2)$. For the window function to satisfy this scale-invariance property, there should exist a monotonic mapping $s(\alpha)$ from a scaling variable $\alpha$ to the spatial axis so that

$$W(\tau - \tau', s(\alpha)) = a W(\alpha(\tau - \tau'), s(1)). \quad (2)$$

Without loss of generality let’s pick $s(\alpha) = \alpha$ for it can be retransformed to any other monotonic $s(\alpha)$ mapping after the analysis. Hence with $0 < s < \infty$,

$$W(\tau - \tau', s) = s W(s(\tau - \tau'), 1) \equiv G(s(\tau - \tau')). \quad (3)$$

**Space-Time Local mechanism**

Equation 1 expresses the encoded memory as an integral over the entire past. However, the encoding mechanism can only have access to the instantaneous functional value of $f$ and its derivatives. The spatial pattern should self sufficiently evolve in real time to encode eq. 4. Since the spatial axis is organized monotonically to correspond to different past moments, only the local neighborhood of any point would affect its time evolution. So we postulate that the most general encoding mechanism that can yield eq. 1 is a space-time local mechanism given by some differential equation. To analyze this, let us first express the general space-time derivative of $T(\tau, s)$ as

$$T^{[n]}_{(m)}(\tau, s) = \sum_{j=0}^{n-1} f^{[n-j-1]}(\tau) W^{[j]}_{(m)}(0, s) + \int_{-\infty}^{\tau} f(\tau') W^{[n]}_{(m)}(\tau - \tau', s) \, dt'. \quad (4)$$

For brevity, we denote the order of time derivative within a square bracket in the superscript and the order of space derivative within a parenthesis in the subscript.

Since $f(\tau)$ is an arbitrary input, $T(\tau, s)$ should satisfy a time independent differential equation which can depend on instantaneous time derivatives of $f(\tau)$. The first term in the r.h.s of eq. 4 is time-local, while the second term involves the entire past. The necessary condition for $T(\tau, s)$ to satisfy a time independent differential equation is tantamount to the derivatives of the window function satisfying a linear equation.

$$\sum_{n,m} C_{nm}(s) W^{[n]}_{(m)}(\tau - \tau', s) = 0. \quad (5)$$

The aim here is to not extract the differential equation satisfied by $T(\tau, s)$, but just to impose its existence. To impose the above condition, note that the derivatives of the window function, namely $W^{[n]}_{(m)}(\tau - \tau', s)$, can be expressed as

$$\sum_{n,m} (n+1)! m! \frac{s^{n+1-m+r}}{r!(m-r)!^2}(\tau - \tau')^r G^{[n+r]}(s(\tau - \tau')),$$

where $r_a = \max[0, m-n-1]$ and the superscript on $G$ represents the order of the derivative w.r.t $s(\tau - \tau')$. Defining $z \equiv s(t-t')$, eq. 5 can be expressed as

$$\sum_{n,m} C_{nm}(s) s^{n+1-m} \sum_{r=r_a}^{m} \frac{(n+1)! m!}{r!(m-r)!^2} z^r G^{[n+r]}(z) = 0 \quad (6)$$

For multiple combinations of $n$ and $m$, $C_{nm}(s)$ could have the same functional form, based on which the above equation will be separable into a set of linear differential equations for $G(z)$ with coefficients given by integer powers of $z$. The general solution is of the form

$$G(z) = \sum_{i,k} a_{ik} z^k e^{-b_i z}. \quad (7)$$

The coefficients $a_{ik}$ and $b_i$ cannot be picked arbitrarily as they are severely constrained along with $C_{nm}(s)$ through eq. 4. The differential equation satisfied by $T(\tau, s)$ can then be obtained by iteratively substituting $W^{[n]}_{(m)}(\tau - \tau', s)$ in the second term of the r.h.s of eq. 4 in terms of lower derivatives and replacing the integral in terms of derivatives of $T(\tau, s)$. For our purposes it suffices to note that the general space-time-local mechanism to construct $T(\tau, s)$ as given in eq. 1 will require a window function satisfying eq. 3 and eq. 7.

Since by definition the window function at each $s$ coarse grains the input about some past moment, we expect it to be non-oscillatory and hence restrict our focus to real values of $b_i$. Further, the requirement of the window function to have normalized area at all $s$ restricts $b_i$ to be positive.

**FIG. 1.** Coarse grained averages over different past times with scale invariant window function projected on to different points on the spatial axis.
Two step process

Consider the simplest scenario in eq. where only one of the coefficients in the set of \( a_{ik} \) and \( b_i \) are non-zero. With the nonzero \( b_i = b \) and \( a_{ik} = b^k/k! \), the window function takes the form

\[ W(\tau - \tau', s) = \frac{(bs)^{k+1}}{k!} (\tau - \tau')^k e^{bs(\tau - \tau')} .\]

To highlight the dependence of the window function on \( k \) and \( b \), we shall denote it by \( w\{k,b\} \). Note that \( W^{[j]}(0, s) = 0 \) for any \( k > j \). The corresponding differential equation satisfied by \( T(\tau, s) \) turns out to be first order in both space and time.

\[ bT^{[1]}(\tau, s) + bsT^{[0]}(\tau, s) - \frac{(k + 1)}{s} T^{[1]}(\tau, s) = 0. \quad (8)\]

The only value \( j \) can take in eq. is zero because the maximum value of \( n \) is 1. Hence the first term in the r.h.s of eq. vanishes while the second term is expressed as lower derivatives of \( T(\tau, s) \), leading to a differential equation consisting only derivatives of \( T(\tau, s) \) and requiring the boundary condition \( T(\tau, \infty) = f(\tau) \).

Interestingly, the time derivative and space derivative can be successively employed in a two step process. The first step is equivalent to encoding the Laplace transform of \( f(\tau) \) as \( F(\tau, s) \) and the second step is equivalent to approximately inverting the Laplace transform to construct \( T(\tau, s) \).

\[ f(\tau) \xrightarrow{Laplace} F(\tau, s) \xrightarrow{Inv.Laplace} T(\tau, s) \]

\[ F^{[1]}(\tau, s) = -bsF(\tau, s) + f(\tau), \quad (9) \]

\[ T(\tau, s) = \frac{(-1)^k}{k!} bs^{k+1} F^{[k]}(\tau, s). \quad (10) \]

Taking \( f(\tau) \) to be a function of bounded variation and \( F(-\infty, s) = 0 \), eq. can be integrated to see that \( F(\tau, s) = \int_{-\infty}^{\tau} f(\tau') e^{-bs(\tau - \tau')} d\tau' \). Based on Post’s inversion formula [15], \( T(\tau, s) \) in eq. is an approximation to the inverse Laplace transform of \( F(\tau, s) \) which grows accurate for large \( k \).

From eq. it is now clear that the general solution for the window function is a linear combination of \( w\{k,b\} \) for different values of \( k \) and \( b \). However, the differential equation satisfied by \( T(\tau, s) \) cannot be constrained to be first order in space and time as in eq. Nevertheless the mechanism for constructing \( T(\tau, s) \) is equivalent to taking linear combinations of the two step process given by equations and for different values of \( k \) and \( b \).

At any given \( s \), \( w\{k,b\} \) is a unimodal function with a peak at \( \tau - \tau' = kb/s \). Arbitrary combinations of \( w\{k,b\} \) will result in undesirable shapes of the window function, hence the values of \( k \) and \( b \) should be appropriately tuned. Figure shows the window functions constructed from four combinations of \( b \) and \( k \) given in table [II]. The combinations are chosen such that at the point \( s = 50 \), the window function coarse grains around a past time of \( \tau' - \tau \approx -1 \). The scale invariance property guarantees that its shape remains identical at any other value of \( s \) with a linear shift in the coarse graining timescale. Comparing combinations 1 and 3, note that the window function is narrower for larger \( k \) (100) than for a smaller \( k \) (8). Combination 2 has been chosen to illustrate a plateau shaped window function whose sides can be made arbitrarily vertical by fine tuning the combinations. Combination 4 presented as dotted curve in fig. illustrates that combining different values of \( k \) for the same \( b \) will generally lead to a multimodal window function which would be an undesirable feature.

**Discretized spatial axis**

A memory system represented on a continuous spatial axis may not be practical, so the spatial axis should be discretized to finite points (nodes). The two step process given by equations and is optimal for discretization particularly when the nodes are picked from a geometric progression in the values of \( s \) [11]. Eq. implies that the activity of each node evolves independently of the others to construct \( F(\tau, s) \) with real time input \( f(\tau) \). This is achieved with each node recurrently connected on to itself with an appropriate decay constant \( bs \). Eq. involves taking the spatial derivative of order \( k \) which can be approximated by the discretized derivative requiring
FIG. 3. Window function \(w\{8, 1\}\) at two points \(s_1 = 6.72\) and \(s_2 = 2.59\) computed on a discretized spatial axis with \(c = 0.1\). The dotted curves correspond to the window functions on the continuous spatial axis (\(c \to 0\)).

linear combinations of activities from \(k\) neighbors on either sides of any node \([11]\).

By choosing the nodes along the \(s\)-axis from a geometric progression, the error from the discretized spatial derivative will be uniformly spread over all timescales, hence such a discretization is ideal to preserve scale-invariance in coarse graining the past. Let us choose the \(s\)-values of successive nodes to have a ratio \((1 + c)\), where \(c < 1\). Figure 3 shows the window function \(w\{k, b\}\) with \(k = 8\) and \(b = 1\) constructed from the discretized approximation of eq. 10 with \(c = 0.1\) at two points on the discretized space, \(s_1 = 6.72\) and \(s_2 = 2.59\). As a comparison, the dotted curves are plotted to show the corresponding window function constructed in the continuous \(s\)-axis (limit \(c \to 0\)). The discretized window functions peak at a later past time and are wider than the window function on the continuous spatial axis. As \(c \to 0\), for any \(k\), the discretized window functions approximate those constructed from a continuous spatial axis, while for larger values of \(c\) the discrepancy grows larger. However, the discretized window function always stays scale-invariant as illustrated at \(s_1\) and \(s_2\) in figure 3. Analyzing eq. 10 it can be inferred that the only way to preserve scale invariance in the discretized window function is to choose the nodes from a geometric progression along the \(s\)-axis. Clearly, scale-invariant window functions of any shape can be constructed from linear combinations of the windowed \(w\{k, b\}\) analogous to the construction in figure 2.

An important consequence of such a construction is that with a total of \(N\) nodes in the memory system, it can represent the coarse grained past from timescales proportional to \((1 + c)^N\).

Alternatively, we could envision storing the past information in an accurate way using a shift register and applying any window function on the stored past. However, with \(N\) nodes in the shift register, only a maximum past timescale proportional to \(N\) can be stored. The accuracy of recovering past information from a shift register sharply drops to zero for timescales beyond its capacity, while in recurrent and feedforward networks it smoothly decays to zero \([3, 9]\). The two step process discussed here involves both recurrent and feedforward connectivities with tailored connection strengths. In the context of constructing memory networks with random recurrent connectivities, imposition of scale-invariance will place significant constraints on their construction due to their equivalence to linear combinations of the described two step process.

Hence, to the extent scale invariantly coarse graining the past has computational utility for memory, linear combinations of the two step process involving Laplace transform and its inversion is both generic and exponentially resource conserving.

**Acknowledgements**: The work was partly funded by NSF BCS-1058937 and AFOSR FA9550-12-1-0369.

---

[1] W. Maass, T. Natschläger, and H. Markram, Neural Computation **14**, 2531 (2002).
[2] H. Jaeger, *The echo state approach to analyzing and training recurrent networks*, GMD-Report 148 (GMD - German National Research Institute for Information Technology, 2001).
[3] O. L. White, D. D. Lee, and H. Sompolinsky, Physical Review Letters **92**, 148102 (2004).
[4] M. Hermans and B. Schrauwen, Neural Networks **23**, 341 (2010).
[5] S. Ganguli, D. Huh, and H. Sompolinsky, Proceedings of the National Academy of Sciences of the United States of America **105**, 18970 (2008).
[6] T. Strauss, W. Wustlisch, and R. Labahn, Neural Computation **24**, 3246 (2012).
[7] R. Legenstein and W. Maass, Neural Networks **20**, 323 (2007).
[8] E. Wallace, H. R. Maei, and P. E. Latham, Neural Computation **25**, 1408 (2013).
[9] S. Ganguli and H. Sompolinsky, Annual Review of Neuroscience **35**, 485 (2012).
[10] A. S. Charles, H. L. Yap, and C. J. Rozell, Neural Computation **26**, 1198 (2014).
[11] K. H. Shankar and M. W. Howard, Journal of Machine Learning Research **14**, 3785 (2013).
[12] W. H. Pres, Comments on Astrophysics **7**, 103 (1978).
[13] B. J. West and M. F. Shlesinger, American Scientist **78**, 40 (1990).
[14] R. F. Voss and J. Clarke, Nature **258**, 317 (1975).
[15] J. Gibbon, Psychological Review **84**, 279 (1977).
[16] B. C. Rakitin, J. Gibbon, T. B. Penny, C. Malapani, S. C. Hinton, and W. H. Meck, Journal of Experimental Psychology: Animal Behavior Processes **24**, 15 (1998).
[17] K. H. Shankar and M. W. Howard, Neural Computation **24**, 134 (2012).
[18] E. Post, Transactions of the American Mathematical Society **32**, 723 (1930).