Can Random Linear Networks Store Multiple Long Input Streams?

Adam S. Charles
School of ECE
Georgia Institute of Technology
Atlanta, Georgia, USA
acharles6@gatech.edu

Dong Yin
Department of Automation
Tsinghua University
Beijing, China, 100084
dongyin2010@hotmail.com

Christopher J. Rozell
School of ECE
Georgia Institute of Technology
Atlanta, Georgia, USA
crozell@gatech.edu

***Abstract*—The short term memory of randomly connected networks has been recently studied in order to better understand the computational and predictive power of such networks. In particular, random, linear, orthogonal networks have been explored extensively in the context a single input stream driving the network. The most recent results state that a stream of length N can be recovered from a network of size $O(S\log^5(N))$ assuming that the input is S-sparse in some basis. Little work, however, addresses more complex networks where multiple input streams feed into the same network. In this paper we extend the results for recovering sparse input streams the multiple input streams feeding into the same network. We find that we can recover L input streams of length N with a network that has $O(S\log^5(LN))$ nodes.**

***Index Terms*—Short-term memory, linear neural network, sparse signals, restricted isometry constant**

I. Introduction

In the past two decades, randomly connected networks have demonstrated surprising utility in predictive tasks [1]. In particular, the literature has considered networked systems with M nodes $\boldsymbol{x} \in \mathbb{R}^N$, where the nodes evolve with an input stream $s[n] \in \mathbb{R}$ as

$$\boldsymbol{x}[n] = f\left(\boldsymbol{W}\boldsymbol{x}[n-1] + \boldsymbol{z}s[n] + \boldsymbol{\epsilon}[n]\right) \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ is the connectivity matrix that describes how the nodes influence each other, $\boldsymbol{z} \in \mathbb{R}^M$ is the feed-forward vector that describes how the input stream drives the nodes, $\boldsymbol{\epsilon} \in \mathbb{R}^M$ is the system noise and $f(\cdot)$ is a potential nonlinearity (e.g. a sigmoidal saturation). The implication of these prediction results is that such networks implicitly store information about the input stream which makes the node values informative enough to predict future values of the data stream. Understanding this short-term memory (STM) of networked systems is therefore important to uncovering the extent of their computational utility.

Recent work has focused on analysis of STM for linear network dynamics where $f(\cdot)$ is the identity function, and the network dynamics are simply

$$\boldsymbol{x}[n] = \boldsymbol{W}\boldsymbol{x}[n-1] + \boldsymbol{z}s[n] + \boldsymbol{\epsilon}[n] \tag{2}$$

Precisely, this area of research has sought methods to quantify the information of the input signal's history stored in the network nodes [1]–[4]. Many methods have been used to these ends, including calculating the correlation of past inputs with the nodes, calculating the VC-dimension of the output states, and learning linear readout functions to recover the history.

Initial work assuming Gaussian input statistics resulted in bounds that require a network of at least $O(N)$ nodes to store input sequences of length N [3]. More recently, the literature on structured signal analysis has yielded a number of tools to exploit a fairly common structure: that many signals can be expressed using few elements of a known basis. This underlying low dimensional structure has been shown to be present in many signal classes (e.g. images [5]), and many novel applications such as compressive sensing [6] leverage this structure in inverting under-determined linear processes. In particular, in compressive sensing, this structure has been used to determine bounds on how many random samples of a signal are needed to recover the signal when the number of measurements is much less than the signal's ambient dimension. Since randomly connected networks can behave much like random sampling, it is reasonable to expect that such networks can also exploit low-dimensional structure to compress and store long input sequences in far fewer than N nodes. To this end, recent work has shown that for certain random network constructions a network with only $O(K\log^\gamma(N))$ nodes (where K is the sparsity of the signal in a known basis and γ is a bounded constant) can store input sequences of length N [7], [8].

While such results are highly encouraging, these approaches still only address networks with a single input stream. Many networks, such as biological networks, may have multiple inputs feeding into a single network. Thus understanding the interplay of many inputs being processed in a single network, and the abilities of the network to remember the inputs and which stream they came from is pivotal in expanding our ability to analyze networked systems. To this end we provide here a first step in this direction, expanding the sparsity based analysis presented in [8] to the multiple input case. We provide high probability guarantees for the recoverability of L input streams of length N into a random orthogonal network of size M. Moreover we show that the recoverability is robust to system noise, which allows standard ℓ_1 recovery techniques to recover the input values from the node values.

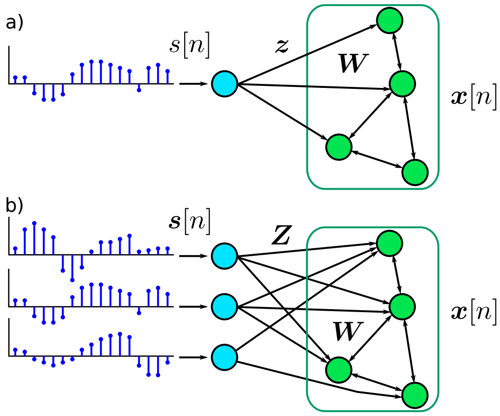

Fig. 1. a) A network with a single input projects the incoming signal through the feed-forward vector $\boldsymbol{z}$ into the network which evolves with the feedback connectivity matrix $\boldsymbol{W}$. The node values $\boldsymbol{x}[n]$ encode the input signal and can be used for prediction or classification tasks. b) A multi input network projects many input streams into the same evolving state. The node values then include influences from all input streams, however the streams must be decoupled to understand how each input influenced the nodes.

II. Background

A. Network memory

The literature on the STM of networked systems has thus far focused on understanding how a single input stream influences the node values at a later time. While STM of random distributed networks has been studied for many system architectures, (including continuous-time networks [4], [9] and spiking networks [10]–[13]), here we focus on discrete-time networks [2], [3], [14].

Initial approaches in understanding STM for discrete-time networks make standard Gaussian assumptions on the network inputs and analyze how informative the nodes are of the past inputs. Throughout the literature, different definitions have been used to quantify how the current nodes were related to the past inputs. For example, one method to test the information content of the nodes is to measure the correlation between a past input and the current network state [2]–[4], where a high correlation would indicate the information of the input was still present in the system. Another method used in [14] was based on the ability to train a linear readout of the network states to recover the past inputs. Throughout all these methods, the Gaussianity assumption has caused any theoretical bounds to limit the recoverable inputs to the number of nodes in the network. This bound is clearly pessimistic for many input sequences, as even early works have shown that for more structured inputs the empirically recoverable input history is longer than the theoretical $M > N$ limit [14].

B. Restricted isometry property

To formalize the observation of [14], recent work has looked at a specific class of structured input streams where the vector of inputs $\boldsymbol{s} = [s[1], \dots, s[N]]^T \in \mathbb{R}^N$ is sparse in a basis. Specifically, $\boldsymbol{s}$ can be written as $\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{a}$ where $\boldsymbol{a}$ is composed of only $K \ll N$ non-zero entries. Such signals have been extensively analyzed in the literature in the context of linear inverse problems, where a series of observations

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\epsilon}. \tag{3}$$

for a linear operator $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ and a noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^M$ must be inverted for $M \ll N$ (i.e. the problem is highly under-determined). By leveraging the sparsity of $\boldsymbol{s}$, certain conditions on $\boldsymbol{A}$ guarantee that this highly compressive operator can be inverted [6]. Specifically, if $\boldsymbol{A}$ satisfies the restricted isometry property (RIP) of order $2K$ with parameters δ, C, i.e. for every $2K$ sparse signal $\boldsymbol{s}$, the following condition holds,

$$C\left(1-\delta\right) \leq \left|\left|\boldsymbol{A}\boldsymbol{s}\right|\right|_2^2 / \left|\left|\boldsymbol{s}\right|\right|_2^2 \leq C\left(1+\delta\right) \tag{4}$$

then the sparse coefficients $\boldsymbol{a}$ can be recovered via the convex optimization program

$$\widehat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \left|\left|\boldsymbol{a}\right|\right|_1 \quad \text{such that} \quad \left|\left|\boldsymbol{x} - \boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{a}\right|\right|_2 \leq \left|\left|\boldsymbol{\epsilon}\right|\right|_2. \tag{5}$$

Furthermore, the recovered coefficients can be shown to be recoverable up to an estimation error given by

$$\left|\left|\boldsymbol{s} - \widehat{\boldsymbol{s}}\right|\right|_2 \leq \alpha \left|\left|\boldsymbol{\epsilon}\right|\right|_2 + \beta \frac{\left|\left|\boldsymbol{\Psi}^T\left(\boldsymbol{s} - \boldsymbol{s}_K\right)\right|\right|_1}{\sqrt{K}}, \tag{6}$$

where the constants α and β depend on the RIP constant δ of $\boldsymbol{A}$, and $\boldsymbol{s}_K$ represents the best K-term approximation to $\boldsymbol{s}$.

C. RIP for STM

While a number of methods have been attempted to understand how inputs to an evolving network influence the state at a later time, one method in particular is based on showing that the states at any time are the result of a unique series of inputs [10], [12], [14], [15]. This echo state property (ESP) of the network ensures that since different inputs cause different network states, the inputs should then be recoverable from the node values, essentially allowing the inversion of a highly compressive function. The ESP is very similar to the RIP with an important difference being that the ESP requires only uniqueness while the RIP implies robustness of the inversion process. In our initial work in [8] we showed that linear distributed networks with random orthogonal connectivity matrices can satisfy the RIP by proving the following theorem:

Theorem II.1. *(Theorem 4.1 from [8]) Suppose $N \geq M$, $N \geq K$ and $N \geq O(1)$. Let $\boldsymbol{U}$ be any unitary matrix of eigenvectors (containing complex conjugate pairs) and the entries of $\boldsymbol{z}$ be i.i.d. zero-mean Gaussian random variables with variance $\frac{1}{M}$. For M an even integer, denote the eigenvalues of $\boldsymbol{W}$ by $\{e^{jw_m}\}_{m=1}^M$. Let the first $M/2$ eigenvalues ($\{e^{jw_m}\}_{m=1}^{M/2}$) be chosen uniformly at random on the complex unit circle (i.e., we chose $\{w_m\}_{m=1}^{M/2}$ uniformly at random from $[0, 2\pi)$) and the other $M/2$ eigenvalues as the complex conjugates of these values. Then, for a given RIP conditioning δ and failure probability $N^{-\log^4 N} \leq \eta \leq \frac{1}{e}$, if*

$$M \geq C\frac{K}{\delta^2}\mu^2\left(\boldsymbol{\Psi}\right)\log^5\left(N\right)\log(\eta^{-1}),$$

the network dynamics satisfy RIP-(K,δ) with probability exceeding $1-\eta$ for a universal constant C, and the coherence parameter defined in terms of the sparsity basis as

$$\mu(\boldsymbol{\Psi}) = \max_{n=0,\ldots,N-1} \sup_{t\in[0,2\pi]} \left| \sum_{m=0}^{N-1} \boldsymbol{\Psi}_{m,n} e^{-jtm} \right|.$$

III. Multiple input model

While previous work has primarily dealt with single input streams as in Equation (2), we can extend these results to more general architectures where multiple input streams feed into the same network. Multiple input models are more complex since all input streams feed in simultaneously and the same dynamics affects all streams equally. Thus any RIP condition needs to show that despite the severe mixing, the node values can still be used to differentiate between different input streams.

Mathematically, we can express the new input model by expanding our network dynamics to consider the input at each time step to be a vector rather than a scalar

$$\begin{aligned} \boldsymbol{x}[n] &= \boldsymbol{W}\boldsymbol{x}[n-1] + \sum_{l=1}^{L} \boldsymbol{z}_l s_l[n] + \widetilde{\boldsymbol{\epsilon}}[n] \\ &= \boldsymbol{W}\boldsymbol{x}[n-1] + \boldsymbol{Z}\boldsymbol{s}[n] + \widetilde{\boldsymbol{\epsilon}}[n], \end{aligned} \quad (7)$$

where now $\boldsymbol{s}[n] \in \mathbb{R}^L$ for all n and $\boldsymbol{Z} \in \mathbb{R}^{M\times L}$ is now a feed-forward matrix, which is composed of concatenating all the individual feed-forward vectors $\boldsymbol{z}_l$. We can analyze this network by taking similar steps as were used to analyze the single input case in [8]. First we express the influence of the input history on the current nodes by assuming $\boldsymbol{x}[0] = \boldsymbol{0}$[1] and iterating Equation (7):

$$\boldsymbol{x}[N] = \sum_{k=1}^{N} \boldsymbol{W}^{N-k}\boldsymbol{Z}\boldsymbol{s}[k] + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} = \sum_{k=1}^{N} \boldsymbol{W}^{N-k}\widetilde{\boldsymbol{\epsilon}}[k]$ is the cumulative system error. This expression allows to formulate the dynamics as a matrix-vector multiplication that expresses the input history-to-node relationship,

$$\boldsymbol{x}[N] = \left[\boldsymbol{Z}, \boldsymbol{W}\boldsymbol{Z}, \cdots, \boldsymbol{W}^{N-1}\boldsymbol{Z}\right] \left[\boldsymbol{s}^T[N], \cdots, \boldsymbol{s}^T[1]\right]^T + \boldsymbol{\epsilon}.$$

From this step we can then use the eigenvalue decomposition of the connectivity matrix $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{-1}$ to express the first matrix in this product as

$$\boldsymbol{x}[N] = \boldsymbol{U}\left[\boldsymbol{D}^0, \cdots, \boldsymbol{D}^{N-1}\right]\boldsymbol{U}^{-1}\boldsymbol{Z}\left[\boldsymbol{s}^T[N], \cdots, \boldsymbol{s}^T[1]\right]^T + \boldsymbol{\epsilon}.$$

Now, by reorganizing the columns, we can obtain

$$\begin{aligned} \boldsymbol{x}[N] &= \boldsymbol{U}\left[\widetilde{\boldsymbol{Z}}_1\boldsymbol{F}, \cdots, \widetilde{\boldsymbol{Z}}_L\boldsymbol{F}\right]\left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \cdots, \boldsymbol{s}_L^T\right]^T + \boldsymbol{\epsilon} \\ &= \boldsymbol{A}\widetilde{\boldsymbol{s}} + \boldsymbol{\epsilon}, \end{aligned} \quad (8)$$

where $\boldsymbol{s}_l \in \mathbb{R}^N$ is the l^{th} input stream ($\boldsymbol{s}_l = [s_l[N], \cdots, s_l[1]]^T$), $\boldsymbol{F}_{i,k} = \boldsymbol{D}_{i,i}^k$ is a projection onto the Fourier coefficients determined by the eigenvalues of $\boldsymbol{W}$, and $\widetilde{\boldsymbol{Z}}_l = \mathrm{diag}(\boldsymbol{U}^{-1}\boldsymbol{z}_l)$ modulates the Fourier measurements for each block. From Equation (8) we can see that the current state is simply the sum of L compressed input streams, where the compression for each block essentially performs the same compression as a single stream network would.

While the similarity between the multiple and single input networks seem to indicate that most aspects of the analysis for the single stream case could be used in the new analysis, the nature of how the streams are mixed through the feed-forward matrix $\boldsymbol{Z}$ precludes some of the network choices available for the single input case. For example, in the single input case, the feed-forward vector $\boldsymbol{z}$ can be chosen to project maximally into the eigenspace of $\boldsymbol{W}$, and choosing a random $\boldsymbol{z}$ incurs additional logarithmic penalties on the nodes needed to store long input sequences. Thus, while it may be tempting to similarly define $\boldsymbol{Z}$ based on the eigenvectors of $\boldsymbol{W}$, we can quickly see that such a strategy would provide poor results. Specifically, if we choose each $\boldsymbol{z}_l$ such that every $\widetilde{\boldsymbol{Z}}_l = \boldsymbol{I}$, then we can see that Equation (8) reduces to

$$\begin{aligned} \boldsymbol{x}[N] &= \boldsymbol{U}\left[\boldsymbol{F}, \boldsymbol{F}, \cdots, \boldsymbol{F}\right]\left[\boldsymbol{s}_1^T, \boldsymbol{s}_2^T, \cdots, \boldsymbol{s}_L^T\right]^T + \boldsymbol{\epsilon}, \\ &= \boldsymbol{U}\boldsymbol{F}\sum_{l=1}^{L} \boldsymbol{s}_l + \boldsymbol{\epsilon}, \end{aligned} \quad (9)$$

which clearly indicates that only the sum of the input streams can ever be recovered and the different inputs cannot be distinguished from one another. Instead, we utilize random feed-forward vectors to allow the network to disambiguate different input streams with high probability, choosing $\boldsymbol{Z}$ to be a set of random Gaussian vectors with i.i.d. zero-mean, variance $1/M$ entries. In this way, each input stream projects differently onto the evolving network state.

IV. RIP for multiple inputs

Using the approach from the prior section, we can show a result similar to Theorem 4.1 in [8]. The main differences involve necessary modifications to the signal model and the resulting coherence term to accommodate the new signal input structure. For a single input we could describe the input model as $\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{a}$, i.e. $\boldsymbol{s}$ is sparse in $\boldsymbol{\Psi}$. We can similarly describe $\widetilde{\boldsymbol{s}} = \boldsymbol{\Psi}\widetilde{\boldsymbol{a}}$, i.e. the composite of all input signals is sparse in a basis $\boldsymbol{\Psi} \in \mathbb{R}^{NL\times NL}$. This means that each signal stream can be written as $\boldsymbol{s}_l = \sum_{k=1}^{L} \boldsymbol{\Psi}^{l,k}\boldsymbol{a}_k$ where $\boldsymbol{\Psi}^{l,k}$ is the $\{l,k\}^{th}$ $N\times N$ block of $\widetilde{\boldsymbol{\Psi}}$. This signal model is very rich in that a given coefficient can influence multiple channels, and the network memory can accommodate the interdependencies. With this model, we find it necessary to generalize the coherence parameter used in [8] to

$$\mu(\boldsymbol{\Psi}) = \max_{l,k=1,\ldots,L} \max_{n=0,\ldots,N-1} \sup_{t\in[0,2\pi]} \frac{\left| \sum_{m=0}^{N-1} \boldsymbol{\Psi}_{m,n}^{l,k} e^{-jtm} \right|}{\|\boldsymbol{\Psi}_n^{l,k}\|_2}. \quad (10)$$

[1]More generally, $\boldsymbol{x}[0]$ can be non-zero. As long as it is a known quantity, it can be subtracted from both sides and the remainder of the analysis holds with the left hand side being $\boldsymbol{x}[N] - \boldsymbol{x}[0]$.

In the single input case, the coherence parameter focused on the deviation of the sparsity basis from the Fourier basis. In this multiple input case, each $N \times N$ block must have low coherence with the Fourier basis. This restriction is reasonable, since if a single sub-block of $\mathbf{\Psi}$ was coherent with the Fourier basis, then at least one input stream would be sparse in a Fourier-like basis and hence would be unrecoverable. Since we are seeking uniform recovery, this situation is not acceptable. We note that for the case of L = 1, the generalized definition of coherence reduces to the definition for single inputs.

Theorem IV.1. *Suppose $NL \geq M$, $N \geq K$ and $N \geq O(1)$. Let $\boldsymbol{U}$ be any unitary matrix of eigenvectors (containing complex conjugate pairs) and the entries of $\boldsymbol{Z}$ be i.i.d. zero-mean Gaussian random variables with variance $\frac{1}{M}$. For M an even integer, denote the eigenvalues of $\boldsymbol{W}$ by $\{e^{jw_m}\}_{m=1}^{M}$. Let the first $M/2$ eigenvalues ($\{e^{jw_m}\}_{m=1}^{M/2}$) be chosen uniformly at random on the complex unit circle (i.e., we chose $\{w_m\}_{m=1}^{M/2}$ uniformly at random from $[0, 2\pi)$) and the other $M/2$ eigenvalues as the complex conjugates of these values. Then, for a given RIP conditioning δ and failure probability $(NL)^{-\log^4 NL} \leq \eta \leq \frac{1}{e}$, if*

$$M \geq C\frac{K}{\delta^2}\mu^2\left(\mathbf{\Psi}\right)\log^5\left(NL\right)\log(\eta^{-1}), \tag{11}$$

where the coherence $\mu\left(\mathbf{\Psi}\right)$ is defined as in Equation (10), $\boldsymbol{A}$ satisfies RIP-(K, δ) with probability exceeding $1 - \eta$ for a universal constant C.

The proof of Theorem IV.1 follows very closely to the proof of Theorem 4.1 in [8]. The main difference is that the correlations between the Gaussian feed-forward vectors need to be taken into account. Since these correlations end up only affecting an expectation for a supremum of a sum of random variables, the number of input streams only shows up with the input history length N in the poly-logarithmic factor. This means that many streams can be stored with only a moderate increase in the number of nodes. Additionally, the sparsity number K in Theorem IV.1 represents the total joint sparsity of all input streams. This implies that if there is significant structure between inputs, the total sparsity may be very small compared to the product L N. Finally, we note that when L = 1, Theorem IV.1 reduces to Theorem 4.1 in [8], indicating that the single stream result can be considered a special case of this more general theorem.

V. Simulations

Since the effects of the sparsity and input length have been demonstrated in [8], we focus here on showing the effects of adding input streams to the network. In Figure 2, we show the results of recovering L length $N = 256$ input sequences with a total sparsity of $K = 30$ from the resulting node values of a random orthogonal network of size M. The input sequences were constructed from K random Gaussian coefficients in a Daubechies-1 basis. We vary the number of input sequences L, and for each input sequence, we look for the smallest size M for which Equation 5 fails to recover all the LN inputs. We see that as L increases, the number of nodes needed for recovery increase in logarithmically, as demonstrated by the similarity to the displayed best logarithmic fit.

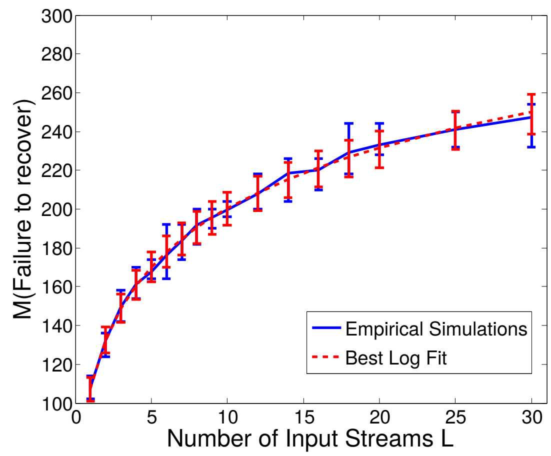

Fig. 2. Driving a network with more input sequences has a logarithmic effect on the number of nodes needed to effectively store the inputs driving the system. Empirically, as we increase the number of input streams, the number of nodes needed to recover the signal increases in a logarithmic manner (shown in solid blue). Shown here are the mean M_{failure} over 10 trials, as well as error bars showing the maximum and minimum M_{failure}. The best fit logarithmic function to this curve (and the maximum and minimum values) has an exponent of 1.1 (1.08, and 1.077 for the maximum and minimum respectively).

VI. Conclusions

In this work we have expanded the STM results for randomly connected networks to networks with multiple input streams. We considered here the case when the input streams had joint sparsity structure, (i.e. the composite vector of all input streams can be sparsely represented in a basis) and used the properties of random orthogonal networks with Gaussian feed-forward vectors to show that the network preserves distances between different joint-sparse input streams. Our result effectively generalizes earlier work in [8] which only considers a single input stream. While in this work we focused on joint-sparse statistics, this is only a single type of low dimensional structure that can be considered for multiple streams. Another prominent low-dimensional structure treats the matrix created by stacking the different inputs as a low-rank matrix. Essentially, this model treats each input as a linear sum of a smaller number ($R < L$) of base input streams. Future work aims to further expand STM recovery guarantees to cover these types of input statistics by drawing on recent advances in low-rank signal recovery [16].

Acknowledgement

This work was partially supported by NSF grants CCF-0830456 and CCF-1409422.

REFERENCES

[1] H. Jaeger and H. Haas, "Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[2] O. L. White, D. D. Lee, and H. Sompolinsky, "Short-term memory in orthogonal neural networks," *Physical Review Lett.*, vol. 92, no. 14, p. 148102, 2004.

[3] S. Ganguli, D. Huh, and H. Sompolinsky, "Memory traces in dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 105, no. 48, p. 18970, 2008.

[4] M. Hermans and B. Schrauwen, "Memory in linear recurrent neural networks in continuous time," *Neural Networks*, vol. 23, no. 3, pp. 341–355, 2010.

[5] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *IEEE Proceedings - Special Issue on Applications of Compressive Sensing & Sparse Representation*, Oct 2008.

[6] E. J. Candes, J. Romberg, and T. T, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[7] S. Ganguli and H. Sompolinsky, "Short-term memory in neuronal networks through dynamical compressed sensing," *Conference on Neural Information Processing Systems*, 2010.

[8] A. S. Charles, H. Yap, and C. J. Rozell, "Short term memory capacity in networks via the restricted isometry property," *Neural Computation*, vol. 26, no. 6.

[9] L. Büsing, B. Schrauwen, and R. Legenstein, "Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons," *Neural computation*, vol. 22, no. 5, pp. 1272–1311, 2010.

[10] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[11] J. Mayor and W. Gerstner, "Signal buffering in random networks of spiking neurons: Microscopic versus macroscopic phenomena," *Physical Review E*, vol. 72, no. 5, p. 051906, 2005.

[12] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural circuit models," *Neural Networks*, vol. 20, no. 3, pp. 323–334, 2007.

[13] E. Wallace, R. M. Hamid, and P. E. Latham, "Randomly connected networks have short temporal memory," *Neural Computation*, vol. 25, pp. 1408–1439, 2013.

[14] H. Jaeger, "Short term memory in echo state networks," *GMD Report 152 German National Research Center for Information Technology*, 2001.

[15] T. Strauss, W. Wustlich, and R. Labahn, "Design strategies for weight matrices of echo state networks," *Neural Computation*, vol. 24, no. 12, pp. 3246–3276, 2012.

[16] A. Ahmed and J. Romberg, "Compressive multiplexing of correlated signals," *arXiv preprint arXiv:1308.5146*, 2013.