Short-term Sequence Memory: Compressive effects of Recurrent Network Dynamics

Adam S. Charles (adamsc@princeton.edu)
Princeton University
Princeton, NJ, 08544, USA

Han Lun Yap (yaphanlun@gmail.com)
DSO National Laboratories
Singapore

Dong Yin dongyin@berkeley.edu
University of California, Berkeley
Berkeley, CA 94720-1776, USA

Christopher J. Rozell (crozell@gatech.edu)
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA

abstract

Neural networks have become a ubiquitous as cognitive models in neuroscience and as machine learning systems. Deep neural networks in particular are achieving near-human performance in many applications. More recently, recurrent neural networks (RNNs) are being increasingly utilized, both as standalone structures and as layers of deep networks. RNNs are especially interesting as cortical networks are recurrent, indicating that recurrent connections are important in human-level processing. Despite their growing use, theory on the computational properties of RNNs is sparse. As many applications hinge on RNNs accumulating information dynamically, the ability of RNNs to iteratively compress information into the network is particularly critical. We thus present here non-asymptotic bounds on the network's short-term memory (STM; the number of inputs that can be compressed into and recovered from a network state). Previous bounds on a random RNN's STM limit the number of recoverable inputs by the number of network nodes. We show that when the input sequences are sparse in a basis or the matrix inputs is low-rank, the number of network nodes needed to achieve an STM grows as the overall information rate. Thus RNNs can efficiently store information embedded in longer input streams, shedding light on their computational capabilities.

Keywords: Short-term memory, RNNs, sparsity, low-rank

Introduction

Artificial neural networks have long been used as simplifying models of biological neural networks with the goal of better understanding fundamental cortical processes. With advances in machine learning and neuroscience, deep and recurrent neural networks are quickly achieving human-like performance in many machine learning tasks, further encouraging the study of cortical systems via their artificial counterparts (Pitts, 1943; Hopfield, 1982; Cadieu et al., 2014; Yamins et al., 2014; Majaj, Hong, Solomon, & DiCarlo, 2015; D.L.K.Yamins & DiCarlo, 2016). Recurrent networks are and have been particularly relevant to the modeling of cortical systems (Pitts, 1943; Hopfield, 1982) as biological neural networks are not feedforward, and contain many recurrent connections. Interestingly, recurrent neural networks (RNNs) have seen a recent resurgence in cortical modeling, in part due to advances in training recurrent networks on experimental tasks (Sussillo & Abbott, 2009; DePasquale, Cueva, Rajan, Abbott, et al., 2018), and are also being increasingly applied in many machine learning applications (Jaeger, 2001; Lukoševičius, 2012; Hinaut, Petit, Pointeau, & Dominey, 2014). In both cases, these networks essentially compress temporally evolving input stimuli over time into a single network state, which can relay that information upstream for use in data processing tasks, such as classification or prediction of future events. In RNNs, temporally evolving signals $\boldsymbol{s}_t \in \mathbb{R}^L$ are iteratively input into a network evolving as

$$\boldsymbol{x}_{n+1} = f(\boldsymbol{W}\boldsymbol{x}_n + \boldsymbol{Z}\boldsymbol{s}_n + \boldsymbol{\varepsilon}_n), \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^M$ is the state of the network at time t, $\boldsymbol{Z}$ is the feed-forward input matrix, $\boldsymbol{W}$ is the matrix of recurrent connections, $\boldsymbol{\varepsilon}_t$ is potential noise in the system, and $f(\cdot) : \mathbb{R}^M \to \mathbb{R}^M$ is a potential non-linearity (Jaeger, 2001; Wilson & Cowan, 1972; Amari, 1972; Sompolinsky, Crisanti, & Sommers, 1988; Maass, Natschläger, & Markram, 2002). The dynamics of RNNs accumulate information from the inputs $\boldsymbol{s}_t$ over time into the network state $\boldsymbol{x}_t$. While in general $f(\cdot)$ is a sigmoidal or thresholding function, much of the analysis of RNNs has started from the linear RNN, where $f(\boldsymbol{x}) = \boldsymbol{x}$. Additionally, a large portion of the RNN and LSM literature has shown that random connectivity (i.e. $\boldsymbol{W}$ and $\boldsymbol{Z}$ are random) yield complex dynamics useful for accumulating information. We thus consider linear, random RNNs in this work.

The information available in the state of the RNN to pass onto the later processing stages depends heavily on how efficiently the RNN dynamics compresses the inputs into the state. One way to quantify this ability of RNNs is to discuss the short-term memory of the network, or the number of past inputs that can be recovered from a network at any given time (Jaeger & Haas, 2004;

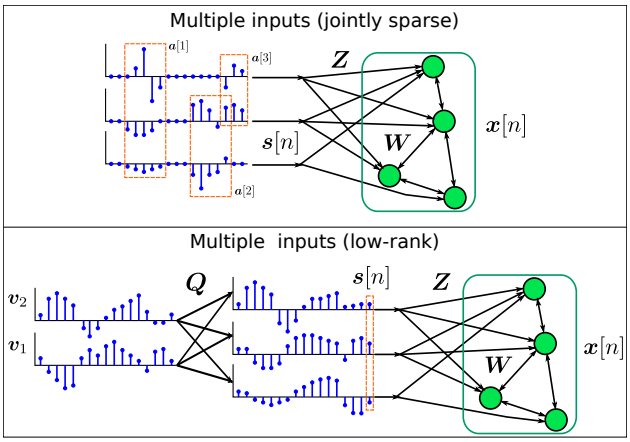

Figure 1: Network and input structures for RNNs with jointly sparse inputs (top) and low-rank inputs (bottom). In both networks the inputs $\boldsymbol{s}[n]$ are projected through the feed-forward connections $\boldsymbol{Z}$ and the temporally evolving state $\boldsymbol{x}[n]$ feeds back into itself through the recurrent connections $\boldsymbol{W}$. In the joint-sparse model a set of sparsely used features compose the input stream using the coefficients $\boldsymbol{a}$. In the low-rank case a set of prototypical signals $\boldsymbol{V}$ combine linearly through a set of coefficients $\boldsymbol{Q}$ to generate more generally correlated input streams.

Maass et al., 2002; Ganguli & Sompolinsky, 2010; Wallace, Hamid, & Latham, 2013; Verstraeten, Schrauwen, dHaene, & Stroobandt, 2007; White, Lee, & Sompolinsky, 2004; Lukoševičius & Jaeger, 2009; Buonomano & Maass, 2009; Charles, Yap, & Rozell, 2014). For example, if the state of the network at time N can recover $\boldsymbol{s}_1$ up to $\boldsymbol{s}_N$, then we can say that the number of recoverable inputs (the STM) is LN. This quantity has been extensively covered for single inputs ($L = 1$, and many canonical results have shown that in general, the number of recoverable inputs is bounded by the network size ($M \leq N$) (White et al., 2004; Wallace et al., 2013). More recent work has explored more particular cases where a potentially long input sequence is well-characterized by a small number of coefficients in a dictionary. Succinctly, this model describes $[s_1 \cdots s_N]^T = \boldsymbol{\Psi a}$, where at most K elements of the coefficients $\boldsymbol{a}$ are non-zero. In this case it was shown that the number of nodes necessary to recover N past inputs actually grows as the information rate $M > K\log(N)$, indicating that for structured inputs (e.g. audio), networks can accumulate information over much longer time-scales (Ganguli & Sompolinsky, 2010; Charles et al., 2014). In particular, our results for single inputs in (Charles et al., 2014) provide non-asymptotic bounds on recovery, with no approximations on the network dynamics, and demonstrated how such bounds can be used to infinity-length input sequences.

Results

In this work we expand on previous work bounding the theoretical capacity of RNNs to compress structured signals by proving non-asymptotic bounds on the STM of RNNs for two classes of structured multiple-input models. We show that both for sparse signals (i.e. the concatenation of all input vectors $\boldsymbol{s} = [\boldsymbol{s}_1^T, \boldsymbol{s}_2^T \cdots \boldsymbol{s}_N^T]^T = \boldsymbol{\Psi a}$ where $\boldsymbol{a}$ has at most K non-zeros) and for low-rank inputs (the matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2 \cdots \boldsymbol{s}_N] = \boldsymbol{QV}^*$ is at most rank $R < \min(L,N)$), the network size for a given STM again grows with the information rate, rather than the number of inputs. Specifically we prove two theorems that outline the necessary conditions on the structure of the input sequence needed for recovery, as well as the optimization programs that can be used to recover the signals and the bounds on the solutions to those optimization programs (Charles, Yin, & Rozell, 2017).

Our approach is based on reformulating the network dynamics such that the current network state $\boldsymbol{x}[N]$ can be expressed as a linear function applied to the past inputs, i.e. $\boldsymbol{x}[N] = \mathcal{A}(\boldsymbol{S}$, and then showing that the function $\mathcal{A}$ can be inverted under the given model for $\boldsymbol{S}$.

In our first results, we show that for the case where the input sequences are sparse, the number of nodes M required to recover N L-dimensional input vectors grows only as $M > O(K\log^{\gamma}(NL))$ - a rate that depends linearly only on the overall sparsity K and poly-logarithmically in the number of inputs NL. For the low-rank case, we find a similar bound where the number of nodes must satisfy $M > O(R(N+L)\log^{\lambda}(NL))$. This bound is linear in the total number of degrees of freedom in the signal $R(N+L)$, and is again only poly-logarithmic in the number of inputs. Interestingly, both bounds also indicate the limitations of RNNs. Specifically, in each case, a coherence parameters $\boldsymbol{\mu}$ appears, showing that random RNNs are only excellent at compressing and storing sequences that *differ* from Fourier vectors (either in the sparsity basis for the sparsity case or from the left singular vectors in the low-rank case).

Simulations

To test our theory, we simulated a number of RNNs and attempted to empirically extract the driving input stream from the network state using LASSO and nuclear norm optimization programs as in (Charles et al., 2017). In particular, we test the effect of the compression levels, defined as $\gamma = M/LN$, and the complexity of the signal, defined as $\rho = K/M$ for sparse inputs and $\rho = R(L+N-R)/M$ for low-rank inputs, on the recoverability of the inputs. Additionally, we test the effect of the input structure, specifically the level to which the input representation is coherent with a sinusoidal basis (i.e. if μ_S^2 and μ_L^2 are low or high).

Figure 2 depicts the results of 20 monte-carlo simulations for each parameter set, fixing $L = 40$ inputs and $N = 100$ time-steps while varying M and either the input sparsity K or the input rank R, as appropriate. We can see that for large parameter values, the relative mean-squared error (rMSE) is very low (essentially at the noise floor), indicating that the input can be recovered even when $M < NL$, i.e. the entirety of all the input streams have been successfully compressed into the single RNN. The simulation results also validate our theory's prediction that a high coherence with a sinusoidal basis limits the compressibility into RNNs.

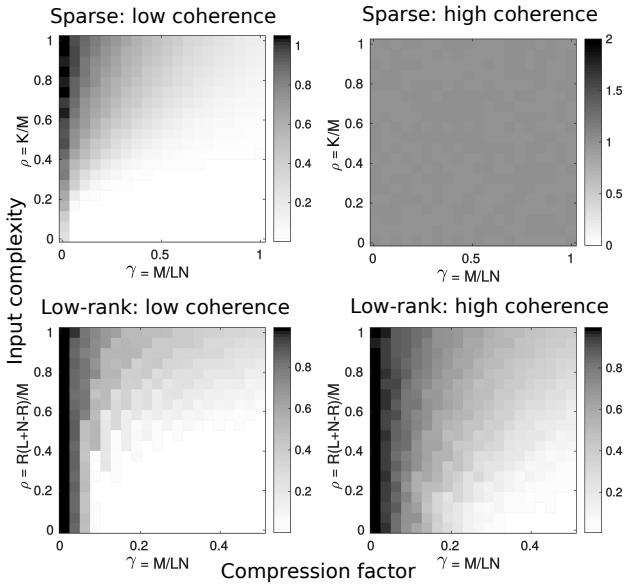

Figure 2: Simulated recovery of sparse and low-rank inputs from echo-state networks. For each plot, the intensity plots the relative mean-squared error (rMSE) over the different parameter settings. Top left: at low coherence values, the inputs are recoverable up to the noise floor for many parameter settings and $M < NL$. Top right: at high coherence the inputs are not recoverable for any $M < NL$. Bottom left: For low-rank inputs there is similarly a large range for which the theory holds. Bottom right: Similarly for low-rank inputs a high coherence value severely limits the recoverability of the RNN inputs.

Finally, our main theory focuses on finite-length input streams. For infinite length inputs, an interesting effect arises where the length of recovery is a parameter left up to the recall system. When the recall system attempts to recover too long or too short of an input stream, the estimated inputs suffer from recall or omission errors respectively (e.g. Fig. 3). In this case, our theory actually predicts an optimal recall length for a given system (Fig. 3 bottom), which can be calculated as a function of the network parameters (Charles et al., 2014).

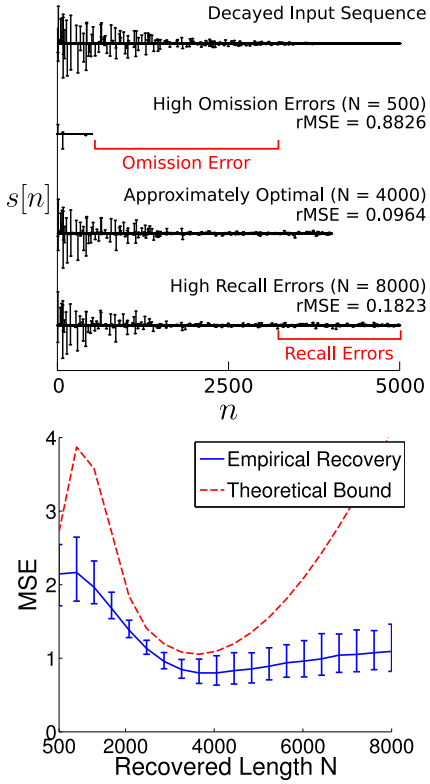

Figure 3: Predicted optimal recall length. When perturbed by an infinite-length input, the network effectively stores a decaying version of the input sequence (top). The system then has an option of how many inputs to attempt to recall (or utilize in computation). Recovering too few inputs or too many inputs results in sub-optimal recovery, due to recall (trying to recover inputs that have long since faded) or omission (ignoring inputs still promonantly effecting the network state) errors. Our theory predicts that there exists an optimal memory length between these two regions that allows for optimal recall of the input stream (bottom).

Conclusions

We demonstrated here two theorems which bound the STM of linear RNNs when the inputs are structured. Our theory predicts both that there can be very high streaming compression of such inputs into randomly connected networks as well as the limitation that the inputs must be incoherent in their representation from a set of sinusoids. When the inputs are too oscillatory, the bounds we prove revert back to the general case $M > N$.

By demonstrating STM bounds, recovery algorithms and accuracy guarantees, the culmination of these bounds (both for the single input networks and multi-

ple input networks), provide some of the most precise STM characterizations of RNNs. While these bounds only address the linear networks, we believe that the intuition gained from proving these bounds begin to provide a method for discussing the computational capabilities of neural networks. More specifically, despite our bounds being derived for a more abstract model, the predictions in terms of the number of items retrievable from the network state and of the existence of an optimal recall time might generalize beyond our simplifying assumptions. We thus believe that further investigation is warranted — both theoretically and empirically — to assess the applicability of these predictions in cortical systems.

References

Amari, S.-I. (1972). Characteristics of random nets of analog neuron-like elements. *IEEE Transactions on Systems, Man and Cybernetics*(5), 643–657.

Buonomano, D. V., & Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, *10*(2), 113–125.

Cadieu, C., Hong, H., Yamins, D., Pinto, N., Ardila, D., Solomon, E., ... DiCarlo, J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS Computational Biology*, *10*(12), e1003963.

Charles, A. S., Yap, H. L., & Rozell, C. J. (2014). Short-term memory capacity in networks via the restricted isometry property. *Neural computation*, *26*(6), 1198–1235.

Charles, A. S., Yin, D., & Rozell, C. J. (2017). Distributed sequence memory of multidimensional inputs in recurrent networks. *Journal of Machine Learning Research*, *18*(7), 1–37.

DePasquale, B., Cueva, C. J., Rajan, K., Abbott, L., et al. (2018). full-force: A target-based method for training recurrent networks. *PloS one*, *13*(2), e0191527.

D.L.K.Yamins, & DiCarlo, J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, *19*(3), 356–365.

Ganguli, S., & Sompolinsky, H. (2010). Short-term memory in neuronal networks through dynamical compressed sensing. *Conference on Neural Information Processing Systems*.

Hinaut, X., Petit, M., Pointeau, G., & Dominey, P. (2014). Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Frontiers in neurorobotics*, *8*.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554–2558.

Jaeger, H. (2001). Short term memory in echo state networks. *GMD Report 152 German National Research Center for Information Technology*.

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science*, *304*(5667), 78-80.

Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade* (pp. 659–686).

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*(3), 127–149.

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

Majaj, N., Hong, H., Solomon, E., & DiCarlo, J. (2015). Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *J. Neuroscience*, *35*(39), 13402-13418.

Pitts, W. (1943). The linear theory of neuron networks: The dynamic problem. *The bulletin of mathematical biophysics*, *5*(1), 23–31.

Sompolinsky, H., Crisanti, A., & Sommers, H. (1988). Chaos in random neural networks. *Physical Review Letters*, *61*(3), 259.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*(4), 544–557.

Verstraeten, D., Schrauwen, B., dHaene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, *20*(3), 391–403.

Wallace, E., Hamid, R. M., & Latham, P. E. (2013). Randomly connected networks have short temporal memory. *Neural Computation*, *25*, 1408–1439.

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

Wilson, H., & Cowan, J. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal*, *12*(1), 1.

Yamins, D., Hong, H., Cadieu, C., Solomon, E., Seibert, D., & DiCarlo, J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Science*.