Recurrent neural networks learn robust representations by dynamically balancing compression and expansion

Anonymous Authors

Abstract

Recordings of neural circuits in the brain reveal extraordinary dynamical richness and high variability. At the same time, dimensionality reduction techniques generally uncover low-dimensional structures underlying these dynamics. What determines the dimensionality of activity in neural circuits? What is the functional role of dimensionality in behavior and task learning? In this work we address these questions using recurrent neural network (RNN) models . We find that, depending on the dynamics of the initial network, RNNs learn to increase and reduce dimensionality in a way that matches task demands. These findings shed light on fundamental dynamical mechanisms by which neural networks solve tasks with robust representations that generalize to new cases.

1 Introduction

Dynamics shape computation in brain circuits. Due to the limitations in our ability to record every neuron in a circuit, it can be difficult to characterize these dynamics through direct observation alone. Bridging between machine learning and neuroscience, artificial recurrent neural networks (RNNs) are powerful tools for investigating dynamical representations in controlled settings, and enable tests of theoretical hypotheses that can be leveraged to formulate experimental predictions (reviewed in [1]). Thinking of artificial networks as dynamical brain circuits is likewise a useful way of understanding their power and flexibility. Since RNNs give rise to well-defined dynamical systems, the *neural representation* of the recurrent units is governed by the system's dynamical response to inputs. In this work we task a network with classifying inputs into one of two classes (binary classification). We treat each input as an impulse delivered at an initial time t_0, and allow the RNN a delay period to process this input before querying the network to output the class label (Fig. 1).

To reveal the essential dynamical elements at play in high-dimensional systems such as RNNs, dimensionality reduction is routinely employed [3]. These approaches reveal a surprising fact: rather than scaling with the number of neurons in the circuit, dynamics are often effectively constrained to regions whose dimensionality seems to be intimately linked to the complexity of the function, or behavior, that the neural circuit fulfills or produces [8] . This link between task and representation dimension is especially intriguing in light of fundamental ideas in learning theory. On one hand, high-dimensional representations can subserve complex and general computations that nonlinearly combine many features of inputs [4, 9]. On the other, low-dimensional representations that preserve only essential features needed for specific tasks can allow learning based on fewer parameters and examples, and hence with better generalization [4, 11].

Here we ask how an RNN balances reducing and increasing dimensionality of input data, and link this behavior to network dynamics. We find that the answer can depend on initialization; in particular, networks that are initially more *chaotic* have a tendency to expand the dimensionality of low-dimensional inputs. Frequently encountered in network models of brain function, dynamical chaos (whereby tiny changes in internal states are amplified by unstable, but deterministic, dynamics)

Preprint. Under review.

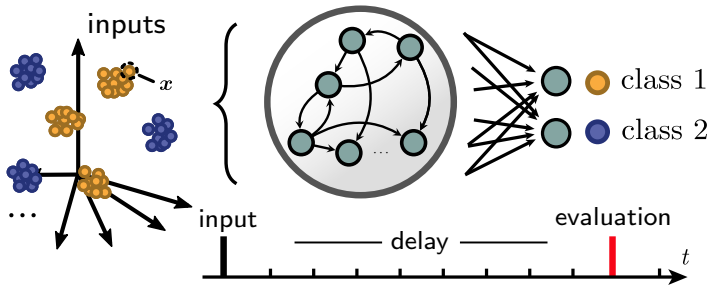

Figure 1: Task and model schematic.

provides a parsimonious explanation for both repeatable structure as well as internally generated variability seen in highly recurrent brain networks such as cortical circuits [10, 5]. While chaos-driven dimensionality expansion with fixed recurrent weights has previously been explored [6], the attributes of this phenomenon as recurrent weights evolve through training are less understood.

2 Results

We consider a standard RNN architecture with hidden unit dynamics $\boldsymbol{h}_t$ and outputs $\boldsymbol{o}_t$ given by

$$\boldsymbol{h}_t = \tanh(\boldsymbol{R}\boldsymbol{h}_{t-1} + \boldsymbol{x}_t + \boldsymbol{b}_1)$$
$$\boldsymbol{o}_t = \boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{b}_2$$

where $\tanh$ is applied element-wise. Here $\boldsymbol{R}$ and $\boldsymbol{W}$ are the recurrent and output weights, respectively, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are biases, and $\boldsymbol{x}_t$ is the input. Simulations are run with $n = 200$ hidden units. The weights $\boldsymbol{R}$ are initialized as a random perturbation of the identity to ensure smooth trajectories: $\boldsymbol{R}^0 = (1-\varepsilon)\boldsymbol{I} + \varepsilon\boldsymbol{J}$ where $\varepsilon = .01$ and $\boldsymbol{I}$ is the identity matrix. The matrix $\boldsymbol{J}$ has normally distributed entries that scale in magnitude with a coupling strength parameter g: $J_{ij} \sim \mathcal{N}(0, g^2/n)$. The coupling strength g governs the degree to which the network is chaotic, with higher values leading to more chaotic dynamics. The outputs are initialized at random: $W^0_{ij} \sim \mathcal{N}(0, .3^2/n)$.

Inputs are drawn from Gaussian distributed clusters lying within a d-dimensional random subspace of the n-dimensional neural activity space. The means of these clusters are distributed uniformly at random within a bounded region of this subspace. Each cluster is assigned one of two class labels, as illustrated in Fig. 1. While this schematic shows six clusters for clarity, in our simulations we use 60 clusters with half being assigned label 1 and half label 2. Recurrent and output weights, as well as biases, are adjusted via stochastic gradient descent with momentum (RMSProp) to minimize a cross-entropy loss function, and classification performance is evaluated on held-out inputs.

RNNs learn to compress high-dimensional inputs

We start by considering the classification of high-dimensional inputs $d = n$ that allow a linear class separation boundary (Fig. 2a). We compare a network initialized near the edge of chaos (EOC) to one that is initially strongly chaotic (SC). Both networks achieve perfect testing accuracy after training (not shown). Fig. 2d shows the top two principal components (PCs) of the network responses to 600 input samples at snapshots in time. The EOC network forms two stable fixed point attractors, one for each class. The SC network separates the classes into two chaotic attractors that begin to mix back together after the evaluation time $t_{\text{eval}} = 10$. This compression phenomenon is partly captured by the *effective dimensionality* (ED) of the representation [7]. The equation for the ED of a set of points is $\text{ED} = 1/\left(\sum_i \tilde{\lambda}_i^2\right)$ where $\tilde{\lambda}_i = \lambda_i / \sum_j \lambda_j$ are the normalized eigenvalues of the covariance matrix of the points. ED can be roughly thought of as the number of PCs needed to capture most of the variance of the points. In Fig. 2b the ED of the representation at times t is plotted over t. The EDs of the trained networks are approximately equal to that of the input at time $t = 0$, since the initial states only differ from the inputs by one application of the nonlinearity. The dimensionality drops with increasing t and is highly compressed at the evaluation time $t_{\text{eval}} = 10$. This drop results both from increasing distances between different classes as well as decreasing distances within classes. Similar phenomena have recently been reported in deep neural networks [2].

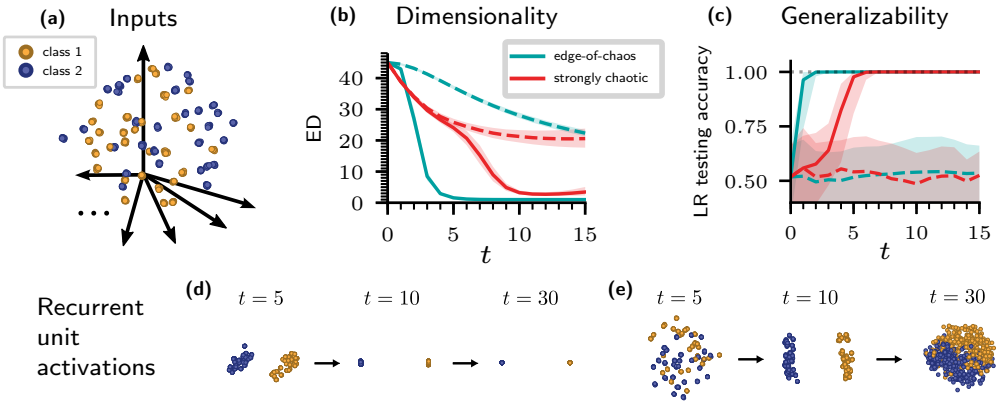

Figure 2: Comparison of edge-of-chaos (blue-green) and strongly chaotic (red) networks classifying high-dimensional inputs. Input color (yellow-orange or purple) denotes true class label. Shaded regions indicate 75% probability mass of gamma distribution fits over 30 network and input realizations, with lines indicating medians. (a) Visual schematic of inputs. (b) Effective dimensionality (ED) of the network representation through time t. Dashed and solid lines denote before and after training the network, respectively. (c) Mean testing accuracy of an LR classifier on network response to held-out input clusters. Dashed lines and legend as in (b). (d) Activations of recurrent units responding to inputs, plotted as "snapshots" in time in PC space, for the edge-of-chaos network. (e) Same as (d) but for the strongly chaotic network.

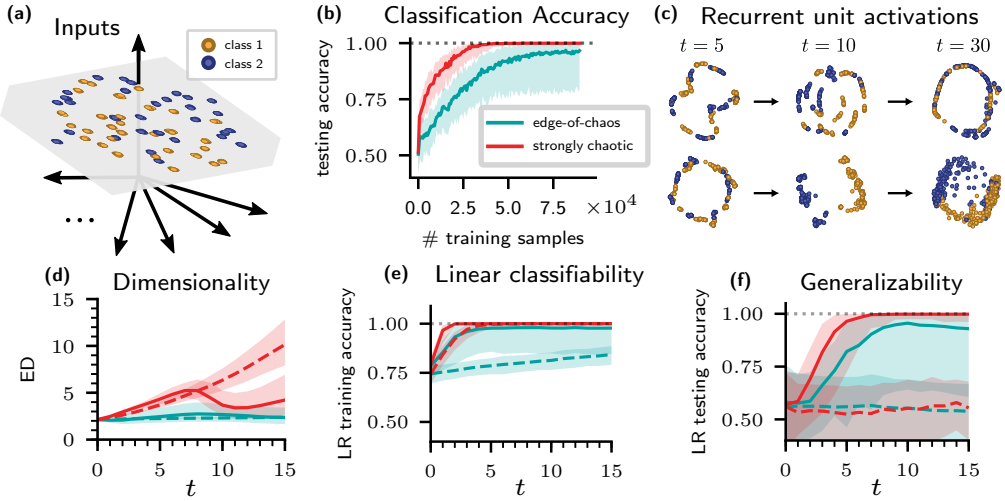

Figure 3: Comparison of edge-of-chaos and strongly chaotic networks classifying low-dimensional inputs. (a) Inputs lie on a two-dimensional plane. (b) Testing accuracy. (c) Top: as in Fig. 2d. Bottom: as in Fig. 2e (d) As in Fig. 2b (e) Mean training accuracy of a logistic regression (LR) linear classifier trained on the recurrent unit activations at each timepoint t. (f) As in Fig. 2c

We next study the coding properties of these representations. To capture how close the representations are to being linearly separable, we train a logistic regression (LR) linear classifier on the representations at each timepoint. In this case, the LR classifier achieves perfect training accuracy on the representations of the SC and EOC networks for all time up to at least $t = 15$ (not shown), both before and after the networks are trained. This confirms that linear separability is preserved by the network dynamics. The interesting properties of the compressed representation can be seen in Fig. 2c, which measures generalization by first training an LR classifier on the network response to inputs drawn from a fixed 80% of the input clusters, and then measuring the accuracy of this classifier on the network response to samples drawn from the remaining 20% of the clusters. The

dashed lines indicate that the representations of the untrained networks do not generalize well to held-out clusters, even while they allow for linear separation boundaries. In contrast, after training, the network representations become increasingly generalizable through time t, eventually allowing for perfect classification accuracy on held-out clusters.

Strongly chaotic RNNs learn to expand and then compress low-dimensional inputs

We next consider inputs embedded in a two-dimensional space, $d = 2$ (Fig. 3a). In this case, boundaries must be highly curved and nonlinear to separate the classes. Fig. 3b illustrates a surprising fact: the SC network learns to classify near-perfectly, while the EOC network is not as successful. PC plots (Fig. 3c) indicate that the SC network is better at separating the classes. This seems to result from the increasing dimensionality through time t exhibited by the SC network in Fig. 3d, much more so than for the EOC network. After training, the SC network first increases dimensionality up to time $t = 7$, and then decreases up to time $t = 11$. This allows it to achieve linear separability of the two classes (Fig. 3e) as well as good generalization (Fig. 3f).

3 Conclusion

We find that in tasks where inputs are linearly separable by class label, RNNs generically reduce the dimension of their inputs over the delay period, in the process forming a representation that lends itself to good generalization. Next, we find that in harder tasks where inputs are low-dimensional and class separation boundaries are highly nonlinear, only networks with sufficiently chaotic initializations are successful. We explain this by showing that chaos-driven dimensionality expansion results in representations with linear separation boundaries. Taken together, we find evidence that RNNs learn representations that have the minimal dimensionality needed to support relatively simple class separation boundaries, provided that the initialization is sufficiently chaotic. These findings invite further exploration of learning strategies through the lens of modulating dimensionality and suggest functional roles for variability found in brain circuits.

References

[1] O. Barak. Recurrent neural networks as versatile tools of neuroscience research. *Current Opinion in Neurobiology*, 46, 2017.

[2] U. Cohen, S. Chung, D. D. Lee, and H. Sompolinsky. Separability and Geometry of Object Manifolds in Deep Neural Networks. *bioRxiv*, page 644658, 2019.

[3] J. P. Cunningham and B. M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, 2014.

[4] S. Fusi, E. K. Miller, and M. Rigotti. Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74, 2016.

[5] G. Lajoie, K. Lin, and E. Shea-Brown. Chaos and reliability in balanced spiking networks with temporal drive. *Phys Rev E*, 87(5):2432–2437, 2013.

[6] R. Legenstein and W. Maass. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3), 2007.

[7] K. Rajan, L. F. Abbott, and H. Sompolinsky. Stimulus-dependent suppression of chaos in recurrent neural networks. *Phys Rev E*, 82(1):1–5, 2010.

[8] M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013.

[9] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. Cambridge, MA, 2002.

[10] C. van Vreeswijk and H. Sompolinsky. Chaotic balanced state in a model of cortical circuits. *Neural computation*, 10(6):1321–71, 1998.

[11] V. N. Vapnik. *Statistical Learning Theory*. New York, 1 edition, 1998.