Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Neurobiology

Recurrent neural networks as versatile tools of neuroscience research

Omri Barak

Recurrent neural networks (RNNs) are a class of computational models that are often used as a tool to explain neurobiological phenomena, considering anatomical, electrophysiological and computational constraints.

RNNs can either be designed to implement a certain dynamical principle, or they can be trained by input–output examples. Recently, there has been large progress in utilizing trained RNNs both for computational tasks, and as explanations of neural phenomena. I will review how combining trained RNNs with reverse engineering can provide an alternative framework for modeling in neuroscience, potentially serving as a powerful hypothesis generation tool.

Despite the recent progress and potential benefits, there are many fundamental gaps towards a theory of these networks. I will discuss these challenges and possible methods to attack them.

Address
Faculty of Medicine and Network Biology Research Laboratories, Technion – Israel Institute of Technology, Israel

Corresponding author: Barak, Omri (omri.barak@gmail.com)

Current Opinion in Neurobiology 2017, **46**:1–6

This review comes from a themed issue on **Computational neuroscience**

Edited by **Adrienne Fairhall** and **Christian Machens**

For a complete overview see the Issue and the Editorial

Available online 29th June 2017

http://dx.doi.org/10.1016/j.conb.2017.06.003

Introduction

The quest to understand the brain involves searching for neural correlates of behavior, and trying to explain them. By explaining, one usually means a model that links neural activity to behavior. This model can be more or less formal, depending on the scientist's preference. Here, I will focus on one class of formal models — recurrent neural networks — and discuss how they can be used as generators of complex, yet interpretable, hypotheses.

RNNs are an important model for computations that unfold over time

Recurrent neural networks (RNNs) are a class of computational models that are used both for explaining neurobiological phenomena and as a tool for solving machine learning problems [1–3]. These are networks in which a given neuron can receive input from any other neuron in the network, without a clear definition of upstream or downstream. Consequently, the activity of neurons in the network is affected not only by the current stimulus, but also by the current state of the network. This property makes such networks ideally suited for computations that unfold over time such as holding items in working memory or accumulating evidence towards a decision.

The rationale for using RNNs as models of the brain stems from both anatomy and electrophysiology. The output of almost all cortical areas has a substantial fraction targeting the area of origin [4]. Thus, the typical cortical network is recurrently connected to itself. Furthermore, an RNN is able to generate rich intrinsic activity patterns, reminiscent of ongoing activity observed in the brain [5].

Functionality, dynamics and connectivity

The functionality of an RNN is defined by its dynamics in relation to the provided inputs and required outputs. In this review I will focus on a commonly used variant known as a rate model [6,7], though other versions will be discussed below. The equations governing the evolution of the network are:

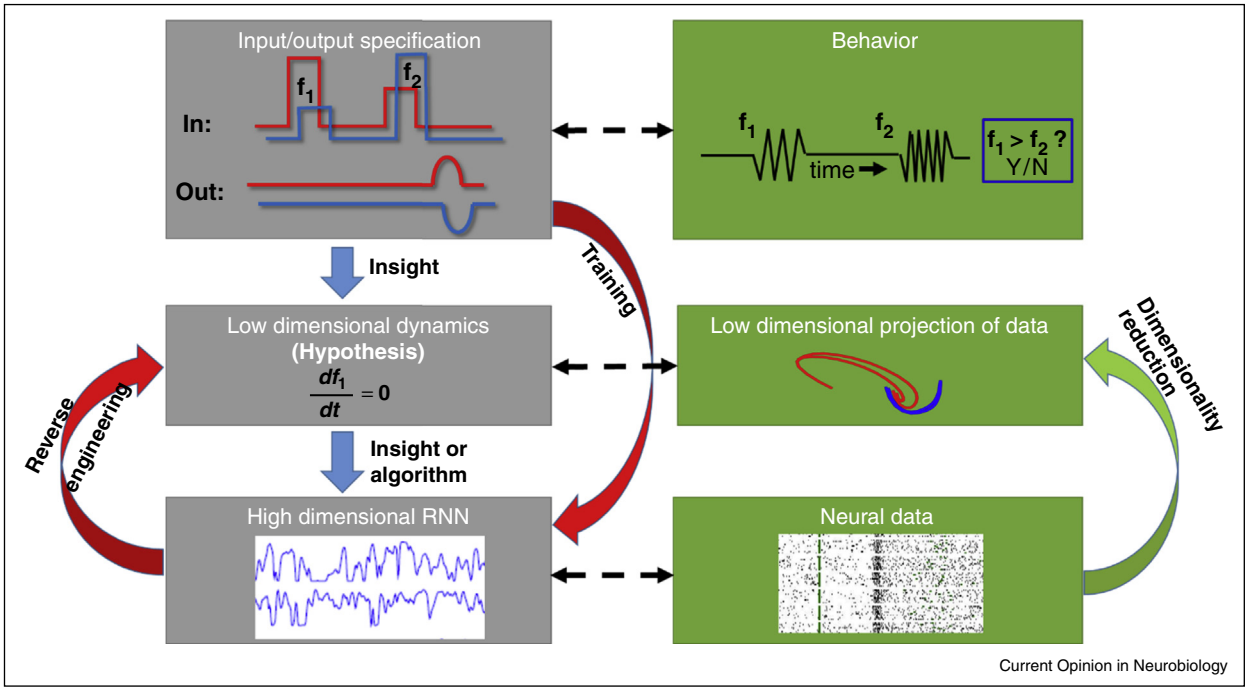
$$\frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} J_{ij}\phi(x_j) + \sum_{k=1}^{M} B_{ik}u_k(t),$$

where x_i is the input to neuron i, J is the connectivity matrix, ϕ is a nonlinear activation function, and the external inputs u are fed through weights B. The current state of the network is defined by the N-dimensional vector x, and its evolution in time defines trajectories in that space. Readouts of the form $z = \sum_{i=1}^{N} w_i\phi(x_i)$ are often defined to represent the output of the network. The dynamics of the network, and hence its functionality, are determined by the connectivity matrices J, B. But how does one set the connectivity to match a desired function?

Designing RNNs often involves implementing low-D intuition

One approach that yielded many important results is to design the connectivity based on intuition (Figure 1, blue arrows). As an example, consider the delayed discrimination task studied by the Romo lab — a monkey receives two vibrotactile stimuli separated by a few seconds and has to report whether the frequency of the first stimulus

Figure 1



Conceptual framework. In order to link behavior and neural data (green), a model (gray) is needed. The common path (blue arrows) includes a formalization of the behavior, followed by an insightful hypothesis about the low dimensional dynamics that can solve the task. These dynamics are then implemented as the connectivity of a high dimensional recurrent neural network. Training RNNs offers an alternative pathway (red arrows), whereby a low dimensional component could implicitly arise without hypothesizing it a priori. Completing the pathway via reverse engineering can provide a powerful hypothesis generation scheme. Studying the regularities in implicitly formed low-D dynamics can lead to new forms of dimensionality reduction, and more principled methods for comparing models with neural activity. Details (clockwise from top right): time course of a typical trial; trajectories of neural dynamics projected to the first two principal components, with colors denoting trial outcome; raster plot of several trials; activity of two units in a trained rate network; equation describing a line attractor; input and output pairs denoting the two trial categories.

was larger than that of the second one (Figure 1, top right) [8]. As a first modeling step, we idealize the task by describing the essence of the behavior as an input-output transformation. The input is composed of two pulses whose amplitude corresponds to the vibration frequency in the experiment, and the output transiently increases only if the first input amplitude is larger than the second one, and decreases otherwise [9]. Now comes the insightful stage — one can realize that a useful component for solving this task is a representation of the first frequency that does not vary in time (observing the neural activity can also provide insights [10]). Such a representation would allow comparison of the two stimuli. In the language of dynamical systems, the network should represent an abstract variable f_1, that obeys the following low dimensional dynamics during the delay: $df_1/dt = 0$. These dynamics can be translated to the connectivity of an RNN using insight [11–13] or algorithmic methods [14–16]. Every value of the variable f_1 corresponds to a certain point x in the N-dimensional phase space. The collection of such points forms a line attractor — a one-dimensional manifold of fixed points — that was built into the network dynamics as part of the design process. Different tasks give rise to different dynamical objects such as saddle points mediating decisions [17], line attractors implementing accumulation of evidence [18,19], point attractors representing associative memories [1] and many more [20–22].

In all of these cases, the underlying mechanism is hypothesized using a low dimensional dynamical system, and then implemented in a high dimensional RNN. The implementation serves as a consistency check for the hypothesis [23], because it makes all assumptions explicit. Furthermore, the activity of model neurons can be compared to those observed experimentally, and possibly provide predictions.

RNNs can be trained without intuition

There is an alternative method to achieve functional RNNs, that is commonly used in the machine learning community — training (Figure 1, red arrows). The computational power of RNNs stems from the fact that the activity of a neuron is affected not only by the current

input to the network but also by the network's state, containing traces of past inputs. This blessing comes with a price — it is difficult to train RNNs [24]. Recently, however, there has been significant progress in training methods [25–31], allowing the use of RNNs for a variety of tasks. Consider the vibrotactile discrimination task mentioned above: the connectivity can be trained by presenting the network with many example trials and modifying connectivity according to some learning rule. The result will be a high dimensional neural network that solves the task, and the activity of the model neurons can be compared to those measured experimentally [18,32••,33•,34]. In some cases, the correspondence to real neurons is higher in the trained case compared to designed networks [9].

Unlike the designed networks, the resulting trained network is somewhat of a black box. We do not know the underlying mechanism that allows the network to solve the task. By skipping the 'intelligent hypothesis' stage, it seems that we replaced one ill-understood complex system (the brain) with yet another complex system that we do not understand (RNN). This replacement, however, is vastly more accessible for research [35], as will be discussed below.

Reverse engineering can uncover implicit low-D structures

Despite the fact that no explicit low-D structure was built into trained networks, such a structure might have formed implicitly during training. This possibility provides hope for reverse engineering trained networks. A recent method approaches this problem from the framework of dynamical systems — trying to find fixed points of the dynamics [36]. The rationale is that line attractors, saddle points and other slow areas of phase space can be the main defining points of the computation performed by the network. This method has uncovered such elements in several examples, leading to better insights on the computations performed by the analyzed networks [18,30,32••,34,37•,38]. Another approach is to assume a parametric form of the underlying (latent) low dimensional dynamics, and then try to infer both the dynamics, and their projection onto the high dimensional phase space [39–43].

The coming years will probably provide us with more tools for this challenge. Dynamical objects such as limit cycles, heteroclinic orbits [44], and strange attractors can be sought after, expanding the range of analyzed functionalities. Networks can be trained while constrained to partially match recorded data [33•]. Reverse engineering during the training phase can provide links to learning and offer new insights on training algorithms. The last point is especially relevant due to the recent advances in using reinforcement learning [31,45–48], which has a potential for a stronger link to biological learning.

Hypothesis generation

If reverse engineering the learned artificial network is successful, we have in a sense discovered a hypothesis, rather than actively thought of its possibility [3]. This was the case for context dependent accumulation of evidence, where a pair of line attractors with a particular relation between their left-eigenvectors and right-eigenvectors was discovered [18]. This mechanism was not hypothesized a priori, although its discovery was still not fully an automation of the scientific process [49] because defining what level of reverse engineering is truly interpretable is a subjective assessment. For the vibrotactile discrimination example, the mechanism uncovered was both expected and surprising. On the one hand, requiring variable delay times led to the formation of a line attractor, as expected for such a task. On the other hand, the decision itself was found to be mediated by a saddle point on some networks, and by a gradual deformation of the output pulse on other networks (Figure 2).

A need for theory

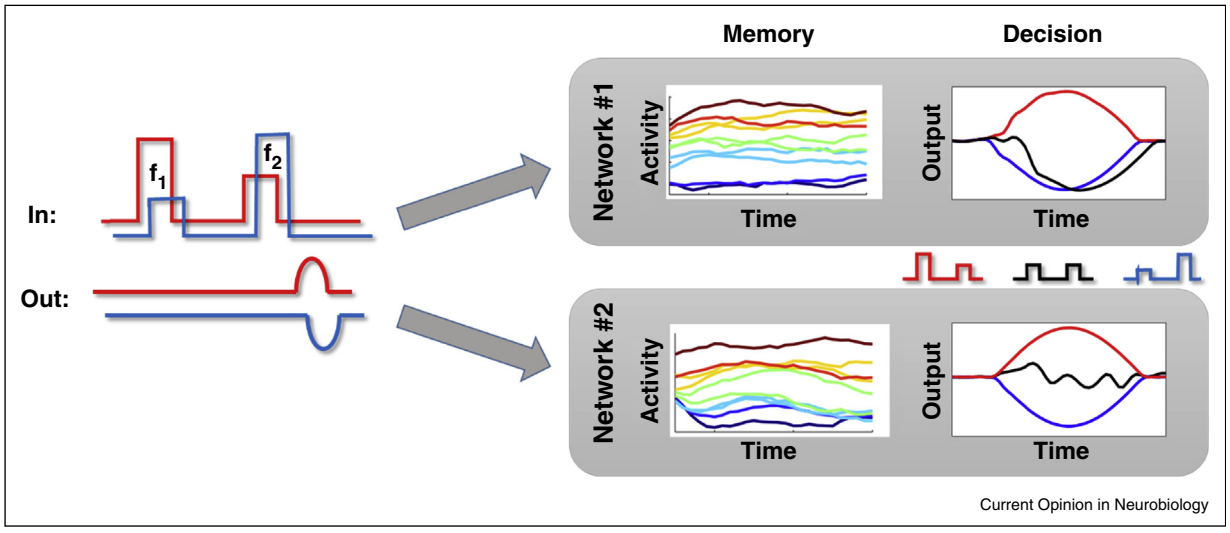
The recent results and insights gained from training RNNs for various tasks are promising. And yet, there is a large gap between our understanding of such networks compared to what we know of designed [1,20,50] or random networks [5,51–53]. What are the limits of this approach? Which tasks can be trained, and where will the networks fail? How invariant are the solutions found by training algorithms? How do the solutions depend on the learning algorithm, or on the details of the single neuron model used?

These, and many other open questions call for the development of new theories for trained RNNs. There are several directions for such theories. Numerically, it is possible to train many networks on the same task, while systematically varying a given aspect (e.g. network architecture) [54]. A recent study advanced this approach by training a network on a task that had several different phases [32••]. The authors showed that forcing an explicit representation of the task phase, as opposed to having such a representation implicitly form through training, improved performance and provided a better match to electrophysiology data from monkeys performing this task. Another approach is to use simplified tasks that can yield to more rigorous analysis, and provide building blocks for more complex settings. Because fixed points have been found to be part of many trained networks, analyzing networks trained to possess several predefined fixed points is a natural first step in this direction. A recent combination of mean field and systems theory tools provided analytical results on the dynamics of such networks [55••].

The correct level of realism in models

The models I discussed so far were of rate networks. In principle, training can also work for spiking networks, and

Figure 2



Invariance and variability in network solutions. Two networks were trained on the same parametric working memory task. The memory component was always captured by a line attractor, demonstrated here by constant activity for different f_1 values (denoted by different colors). The implementation of the decision process varied between networks. The right panels show the output of the networks in response to $f_1 > f_2$ (red), $f_1 > f_2$ (blue), and $f_1 \approx f_2$ (black). Network #1 formed a saddle point, as revealed by the slowing down of the dynamics for a borderline stimulus. Network #2 formed a smooth transition between the two outputs.

it seems that one should strive for this more biologically plausible version. The results of a recent method for training such networks [56,57••], however, contain a lesson about the quest for realistic models. In this work, a spiking network was constructed to be functionally equivalent to a trained rate network. The correspondence, however, was not at the single neuron level — every rate neuron corresponded to the entire spiking population. If one were to consider the actual brain as yet another equivalent network performing the same task, then there is no justification to compare individual model neurons to their experimental counterparts. Comparison should be done instead at the population level, guided by those elements that are invariant to the particularities of the network implementation. This viewpoint also implies that progress in understanding trained RNNs could lead to new insights on dimensionality reduction of recorded neural activity [58].

On the other hand, it is possible that a certain biological element such as modularity, spiking or short-term plasticity, qualitatively changes the set of solutions available to the network [22]. In that sense, trained RNNs can provide a new tool to assess the functional significance of anatomical or physiological findings.

Conclusions

Recurrent neural networks are a valuable tool for neuroscience research. Training these networks has opened an alternative framework to model neural systems that holds much promise. Besides providing a richer set of candidate networks for specific tasks, this framework can pave the way to new categories of scientific questions. What is the correct level of realism when modeling the brain? What is the space of network solutions for a given task? How do biological constraints modify this space? Can we understand the process of learning from reward as a gradual formation and deformation of dynamical objects? In order to reap these larger benefits, a strong theoretical foundation is needed, and will hopefully be developed in the coming years.

Trained and designed networks can be thought of as two ends of a spectrum. In biological networks, training occurs on a background of existing anatomical and physiological constraints. These constraints can be considered as designed elements within the trained network. From a statistical viewpoint, prior knowledge may be the formal way of incorporating such constraints into training.

My presentation of this framework assumed the existence of a low dimensional dynamical system in all networks. But should one always exist? One reason to expect such low dimensionality stems from the reduced nature of most experimental preparations [35]. Modeling such tasks is likely to produce a low dimensional solution. When thinking of natural stimuli and behavior, it is not clear whether this will continue to be the case [59]. Even in this scenario, however, the sensory stimuli are generated by underlying regularities of objects in the outside world, and the networks that adapt to them may represent such regularities as low dimensional dynamical objects [60,61].

It remains to be seen how well the current methods will scale to such cases, and whether new approaches will be required for this challenge.

This review focused on recurrent neural networks, but the overall framework of using machine learning techniques as a hypothesis generation engine generalizes beyond these models. Trained feed forward models are being probed for similarity with the visual cortex [62], and attempts are made to advance a theory in that domain as well [63,64]. As machine learning provides us with ever increasing levels of performance [48,65–67], accompanied by a parallel rise in opaqueness, cross field fertilization holds great promise.

Conflict of interest statement

Nothing declared.

Acknowledgements

I thank Larry Abbott, Naama Brenner and Ron Meir for helpful comments. This work was supported by the Israel Science Foundation (grant No. 346/16) and by European Research Council FP7 Career Integration Grant 2013-618543.

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

1. Hopfield JJ: **Neural networks and physical systems with emergent collective computational abilities**. *Proc Natl Acad Sci U S A* 1982, **79**:2554.

2. Graves A, Mohamed A, Hinton G: **Speech recognition with deep recurrent neural networks**. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); IEEE: 2013:6645-6649*.

3. Sussillo D: **Neural circuits as computational dynamical systems**. *Curr Opin Neurobiol* 2014, **25**:156-163.

4. Douglas RJ, Martin KA: **A functional microcircuit for cat visual cortex**. *J Physiol* 1991, **440**:735-769.

5. van Vreeswijk C, Sompolinsky H: **Chaos in neuronal networks with balanced excitatory and inhibitory activity**. *Science* 1996, **274**:1724-1726.

6. Wilson HR, Cowan JD: **Excitatory and inhibitory interactions in localized populations of model neurons**. *Biophys J* 1972, **12**:1-24.

7. Dayan P, Abbott LF: *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press; 2005.

8. Romo R, Brody CD, Hernández A, Lemus L: **Neuronal correlates of parametric working memory in the prefrontal cortex**. *Nature* 1999, **399**:470-473.

9. Barak O, Sussillo D, Romo R, Tsodyks M, Abbott LF: **From fixed points to chaos: three models of delayed discrimination**. *Prog Neurobiol* 2013, **103**:214-222.

10. Machens CK, Romo R, Brody CD: **Functional, but not anatomical, separation of "What" and "When" in prefrontal cortex**. *J Neurosci* 2010, **30**:350-360.

11. Machens CK, Romo R, Brody CD: **Flexible Control of Mutual Inhibition: A Neural Model of Two-Interval Discrimination**. *Am Assoc Adv Science* 2005.

12. Miller P, Brody CD, Romo R, Wang XJ: **A recurrent network model of somatosensory parametric working memory in the prefrontal cortex**. *Cereb Cortex* 2003, **13**:1208-1218.

13. Miller P, Wang XJ: **Inhibitory control by an integral feedback signal in prefrontal cortex: a model of discrimination between sequential stimuli**. *Proc Natl Acad Sci U S A* 2006, **103**:201-206.

14. Boerlin M, Machens CK, Deneve S: *Balanced Spiking Networks can Implement Dynamical Systems with Predictive Coding*. 2012.

15. Singh R, Eliasmith C: **Higher-dimensional neurons explain the tuning and dynamics of working memory cells**. *J Neurosci* 2006, **26**:3667-3678.

16. Thalmeier D, Uhlmann M, Kappen HJ, Memmesheimer R-M: **Learning universal computations with spikes**. *PLOS Comput Biol* 2016, **12**:e1004895.

17. Wang XJ: **Decision making in recurrent neuronal circuits**. *Neuron* 2008, **60**:215.

18. Mante V, Sussillo D, Shenoy KV, Newsome WT: **Context-dependent computation by recurrent dynamics in prefrontal cortex**. *Nature* 2013, **503**:78-84.

19. Seung HS: **How the brain keeps the eyes still**. *Proc Natl Acad Sci U S A* 1996, **93**:13339-13344.

20. Ben-Yishai R, Bar-Or RL, Sompolinsky H: **Theory of orientation tuning in visual cortex**. *Proc Natl Acad Sci U S A* 1995, **92**:3844-3848.

21. Burak Y, Fiete IR: **Accurate path integration in continuous attractor network models of grid cells**. *PLoS Comput Biol* 2009, **5**:e1000291.

22. Laje R, Buonomano DV: **Robust timing and motor patterns by taming chaos in recurrent neural networks**. *Nat Neurosci* 2013, **16**:925-933.

23. Abbott LF: **Theoretical neuroscience rising**. *Neuron* 2008, **60**:489-495.

24. Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J: **Gradient flow in recurrent nets: the difficulty of learning long-term dependencies**. *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press; 2001.

25. Bengio Y, Boulanger-Lewandowski N, Pascanu R: **Advances in optimizing recurrent networks**. *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013:8624-8628.

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

27. Maass W, Natschläger T, Markram H: **Real-time computing without stable states: a new framework for neural computation based on perturbations**. *Neural Comput* 2002, **14**:2531-2560.

28. Martens J, Sutskever I: **Learning recurrent neural networks with Hessian-free optimization**. *Proc. 28th Int. Conf. on Machine Learning*. 2011.

29. Sussillo D, Abbott LF: **Generating coherent patterns of activity from chaotic neural networks**. *Neuron* 2009, **63**:544-557.

30. Song HF, Yang GR, Wang X-J: **Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework**. *PLoS Comput Biol* 2016, **12**:e1004792.

31. Hoerzer GM, Legenstein R, Maass W: **Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning**. *Cereb Cortex* 2012.

32. •• Enel P, Procyk E, Quilodran R, Dominey PF: **Reservoir computing properties of neural dynamics in prefrontal cortex**. *PLOS Comput Biol* 2016, **12**:e1004967.
The authors use trained RNNs to compare model and experimental data in a multi-phase task. The internal representation of task phase is either explicitly or implicitly represented by the network, and consequences for stability and agreement with data are discussed.

33. • Rajan K, Harvey CD, Tank DW: **Recurrent network models of sequence generation and memory**. *Neuron* 2016, **90**:128-142.
This paper provides an example of constraining network models with experimental data while training an RNN. The connectivity of the resulting network is analyzed to gain mechanistic insights.

34. Sussillo D, Churchland MM, Kaufman MT, Shenoy KV: **A neural network that finds a naturalistic solution for the production of muscle activity**. *Nat Neurosci* 2015, **18**:1025-1033.

35. Gao P, Ganguli S: **On simplicity and complexity in the brave new world of large-scale neuroscience**. *Curr Opin Neurobiol* 2015, **32**:148-155.

36. Sussillo D, Barak O: **Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks**. *Neural Comput* 2013, **25**:626-649.

37. • Carnevale F, de Lafuente V, Romo R, Barak O, Parga N: **Dynamic control of response criterion in premotor cortex during perceptual detection under temporal uncertainty**. *Neuron* 2015, **86**:1067-1077.
An example of training followed by reverse engineering and comparison to experimental data. The authors showed that the proximity to a separatrix can serve as a representation of prior knowledge in the temporal domain.

38. Zhao Y, Park IM: **Interpretable nonlinear dynamic modeling of neural trajectories**. *Advances in Neural Information Processing Systems*. 2016:3333-3341.

39. Sussillo D, Jozefowicz R, Abbott LF, Pandarinath C: **LFADS-latent factor analysis via dynamical systems**. *ArXiv Prepr* 2016. ArXiv160806315.

40. Pfau D, Pnevmatikakis EA, Paninski L: **Robust learning of low-dimensional dynamics from large neural ensembles**. In *Advances in Neural Information Processing Systems 26.* Edited by Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ. Curran Associates, Inc.; 2013:2391-2399.

41. Lakshmanan KC, Sadtler PT, Tyler-Kabara EC, Batista AP, Yu BM: **Extracting low-dimensional latent structure from time series in the presence of delays**. *Neural Comput* 2015.

42. Bongard J, Lipson H: **Automated reverse engineering of nonlinear dynamical systems**. *Proc Natl Acad Sci U S A* 2007, **104**:9943-9948.

43. Talmon R, Coifman RR: **Intrinsic modeling of stochastic dynamical systems using empirical geometry**. *Appl Comput Harmon Anal* 2015, **39**:138-160.

44. Rabinovich MI, Huerta R, Varona P, Afraimovich VS: **Transient cognitive dynamics, metastability, and decision making**. *PLOS Comput Biol* 2008, **4**:e1000072.

45. Song HF, Yang GR, Wang X-J: **Reward-based training of recurrent neural networks for cognitive and value-based tasks**. *eLife* 2017, **6**:e21492.

46. Miconi T: **Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks**. *eLife* 2017, **6**:e20899.

47. Matsuki T, Shibata K: **Reward-based learning of a memory-required task based on the internal dynamics of a chaotic neural network**. *Neural Information Processing*. Cham: Springer; 2016, 376-383.

48. Levine S, Finn C, Darrell T, Abbeel P: *End-to-End Training of Deep Visuomotor Policies*. 2015:. ArXiv150400702 Cs.

49. King RD, Whelan KE, Jones FM, Reiser PGK, Bryant CH, Muggleton SH, Kell DB, Oliver SG: **Functional genomic hypothesis generation and experimentation by a robot scientist**. *Nature* 2004, **427**:247-252.

50. Gardner E: **The space of interactions in neural network models**. *J Phys Math Gen* 1988, **21**:257.

51. Mastrogiuseppe F, Ostojic S: **Intrinsically-generated fluctuating activity in excitatory-inhibitory networks**. *PLoS Comput Biol* 2017, **13**:e1005498.

52. Aljadeff J, Stern M, Sharpee T: **Transition to chaos in random networks with cell-type-specific connectivity**. *Phys Rev Lett* 2015, **114**:088101.

53. Kadmon J, Sompolinsky H: **Transition to chaos in random neuronal networks**. *Phys Rev X* 2015, **5**:041030.

54. Collins J, Sohl-Dickstein J, Sussillo D: *Capacity and Trainability in Recurrent Neural Networks*. 2016:. ArXiv Prepr. ArXiv161109913.

55. •• Rivkind A, Barak O: **Local dynamics in trained recurrent neural networks**. *Phys. Rev. Lett* 2017, **118**:258101.
The first study to use mean field and systems theory tools to gain analytical results on trained RNNs. The results enable prediction of stability, resonance, and offer an explanation for output robustness in the presence of internal variability.

56. Abbott LF, DePasquale B, Memmesheimer R-M: **Building functional networks of spiking model neurons**. *Nat Neurosci* 2016, **19**:350-355.

57. •• DePasquale B, Churchland MM, Abbott LF: *Using firing-rate dynamics to train recurrent networks of spiking model neurons*. 2016:. ArXiv Prepr. ArXiv160107620.
The authors harness an equivalence between population level dynamics in firing rate and spiking networks to train spiking RNNs to perform tasks. The fact that each rate neuron is represented by the entire spiking population raises questions on the level of realism models should strive to.

58. Williamson RC, Cowley BR, Litwin-Kumar A, Doiron B, Kohn A, Smith MA, Yu BM: **Scaling properties of dimensionality reduction for neural populations and network models**. *PLOS Comput Biol* 2016, **12**:e1005141.

59. Fusi S, Miller EK, Rigotti M: **Why neurons mix: high dimensionality for higher cognition**. *Curr Opin Neurobiol* 2016, **37**:66-74.

60. Chandler DM, Field DJ: **Estimates of the information content and dimensionality of natural scenes from proximity distributions**. *JOSA A* 2007, **24**:922-941.

61. Lungarella M, Pegors T, Bulwinkle D, Sporns O: **Methods for quantifying the informational structure of sensory and motor data**. *Neuroinformatics* 2005, **3**:243-262.

62. Yamins DLK, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ: **Performance-optimized hierarchical models predict neural responses in higher visual cortex**. *Proc Natl Acad Sci U S A* 2014, **111**:8619-8624.

63. Saxe AM, McClelland JL, Ganguli S: *Exact solutions to the nonlinear dynamics of learning in deep linear neural networks*. 2013:. ArXiv Prepr. ArXiv13126120.

64. Soudry D, Carmon Y: *No bad local minima: Data independent training error guarantees for multilayer neural networks*. 2016:. ArXiv Prepr. ArXiv160508361.

65. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M *et al.*: **Mastering the game of go with deep neural networks and tree search**. *Nature* 2016, **529**:484-489.

66. He K, Zhang X, Ren S, Sun J: *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015:1026-1034.

67. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G *et al.*: **Human-level control through deep reinforcement learning**. *Nature* 2015, **518**:529-533.