Of course, these interesting speculations about the functions of extended, nonlocal reactivations are just that: speculations. After many years, it has still not been demonstrated that the sleep reactivation is involved in memory consolidation; it has not been shown that reverse replay is involved in the credit assignment problem; it has not been shown that the extended sweeps at choice points are involved in VTE behavior. It is possible that these results are all epiphenomena of hippocampal circuitry and plasticity mechanisms, which may be necessary for proper functioning during active locomotion but which have no well-defined functions during sleep, restfulness, or pauses. Proving that these phenomena serve specific functional roles in cognition will be a difficult task that requires clever behavioral tasks and neuronal activity markers that may be beyond current technological limitations. For example,

it would be interesting to see if one could train the rat to produce specific nonlocal representations during pauses and SW-Rs, which would suggest that these events are under the control of the rat. Further, the increasing development of sophisticated imaging tools and molecular-genetic techniques may someday allow the manipulation of the specific neural circuits that generate these events. Future research may ultimately expand on these results to decipher the neural mechanisms underlying internally generated, conscious thought that is removed from external sensory stimulation. Imagine the possibilities!

REFERENCES

Buzsaki, G. (1986). Brain Res. *398*, 242–252.

Davidson, T.J., Kloosterman, F., and Wilson, M.A. (2009). Neuron *63*, this issue, 497–507.

Diba, K., and Buzsaki, G. (2007). Nat. Neurosci. *10*, 1241–1242.

Foster, D.J., and Wilson, M.A. (2006). Nature *440*, 680–683.

Hassabis, D., Kumaran, D., Vann, S.D., and Maguire, E.A. (2007). Proc. Natl. Acad. Sci. USA *104*, 1726–1731.

Jackson, J.C., Johnson, A., and Redish, A.D. (2006). J. Neurosci. *26*, 12415–12426.

Johnson, A., and Redish, A.D. (2007). J. Neurosci. *27*, 12176–12189.

Lee, A.K., and Wilson, M.A. (2002). Neuron *36*, 1183–1194.

Louie, K., and Wilson, M.A. (2001). Neuron *29*, 145–156.

O'Neill, J., Senior, T., and Csicsvari, J. (2006). Neuron *49*, 143–155.

Skaggs, W.E., and McNaughton, B.L. (1996). Science *271*, 1870–1873.

Wilson, M.A., and McNaughton, B.L. (1994). Science *265*, 676–679.

Harnessing Chaos in Recurrent Neural Networks

Dean V. Buonomano[1,2,3,*]
[1]Department of Neurobiology
[2]Department of Psychology
[3]Brain Research Institute
University of California, Los Angeles, Los Angeles, CA 90095, USA
*Correspondence: dbuono@ucla.edu

In this issue of Neuron, *Sussillo and Abbott describe a new learning rule that helps harness the computational power of recurrent neural networks.*

One of the ironies of the human condition is that while many of our cognitive abilities derive from the complex dynamics of recurrent networks of neurons, we are quite inept at understanding neural dynamics in these same networks. The challenge of understanding the behavior of complex systems is, of course, not unique to neuroscience. Scientists have long struggled with how macroscopic properties emerge from interactions between a large number of simple elements, such as how complex social structures emerge from the interaction of individual members of insect colonies, or

how biomolecules give rise to life itself. However, the problem seems particularly perverse in neuroscience, since no one would describe a neuron as a simple element, and the synaptic strengths—which ultimately determine the nature of the interactions—continuously vary as a result of short- and long-term forms of plasticity.

Over the past 3 decades, neuroscientists have sought to understand the rules governing synaptic weights and have examined how computations can be achieved by appropriately adjusting these weights. These studies have resulted

in experimentally derived learning rules, such as spike-timing-dependent plasticity (Dan and Poo, 2004), and biologically implausible but computationally powerful algorithms, such as backpropagation (Rumelhart et al., 1986). The computational potential of these rules has focused primarily on feedforward networks, largely because they are not burdened with the complexities that can result in chaotic dynamics or epileptic-like "runaway excitation."

Despite some progress with artificial (Hopfield, 1982) and experimentally derived (Izhikevich, 2006; Legenstein

et al., 2008) learning rules, figuring out how to reliably adjust synaptic weights in recurrently connected networks has proven elusive. The good news is that such networks are able to perform useful computations even with weights set at random—as long as they are not too weak or strong. Specifically, when all units in a randomly connected recurrent network are in turn connected to an output ("readout") neuron, it is possible to adjust the weights from the neurons onto the output neuron (W_{Out}) so that it will generate selective responses to complex stimuli, such as spoken words. This setup—referred to as a state-dependent network or a liquid-state machine (Buono-mano and Maass, 2009)—relies on regimes in which recurrent connectivity is strong enough to influence the behavior of the network, but not so strong that network behavior is governed by internal dynamics or is capable of generating self-perpetuating patterns of activity. Because of this, state-dependent networks are better suited to discriminate input patterns than generate complex output patterns.

In this issue, Sussillo and Abbott (2009) extend a related framework and demonstrate that their new approach has an impressive ability to learn complex output patterns and input-output transformations. Imagine a network of 500 units, each connected to 50 others, and that we set the weights according to a Gaussian distribution with a mean of 0 scaled by an internal gain g (we will pretend that the "neurons" take on analog values between −1 and 1 and that synapses can be positive or negative). It is easy to see that, in response to a brief input, if all the synaptic strengths are very weak (g close to 0), activity quickly fades away. On the other hand, if the synapses are strong (g is large), with the same input the network might generate complex ongoing patterns of activity. Indeed, different g values can produce a range of dynamic regimes; generally values below 1 do not generate
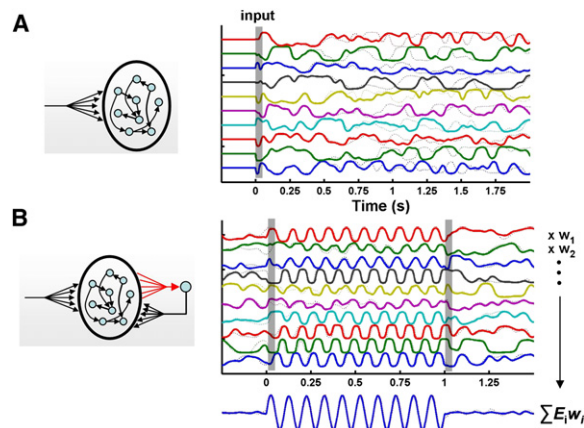

Figure 1. Dynamics of Randomly Connected Recurrent Neural Networks
(A) Complex self-maintaining activity patterns are observed in response to a brief stimulus (gray) in a recurrent network (ellipse on left, with blue circles representing neurons, and arrows, synapses) in which the weights are randomly assigned strong values (g = 1.5). Each line represents the activity of a single unit of a large recurrent network. The dashed lines represent the same simulation in which the activity of a single unit was altered at t = 20 ms. The divergence indicates a high sensitivity to noise, suggestive of chaotic behavior.
(B) FORCE learning rule applied to a network with g = 1.5 and trained to generate a 10 Hz sinusoid at the onset of a brief input (gray; there was also an offset signal at t = 1 s). Dashed lines represent the same simulation when the activity of a single units was altered at t = −750 ms. This network includes an external feedback unit that receives inputs (red) from the recurrent network. Only the W_{Out} (red) were modified during training.

self-sustaining activity, and above 1, they generate increasingly complex and chaotic activity (Figure 1A) (Sompolinsky et al., 1988).

Let's assume that we want our network of 500 neurons (E units) to do something useful: generate a 10 Hz sinusoid output. First we will add a single output unit receiving input from all 500 E units. We want to achieve a behavior in which the output is steady, but will generate a continuous sinusoidal pattern in response to a brief input stimulus. How do we set the weights between the neurons in the recurrent network (W_{EE}) and those onto the output unit (W_{Out})? Clearly if the W_{EE} weights are too weak, activity will die out, making it impossible to generate a periodic output. One way to solve this problem is to add an input to the network that just happens to generate precisely our desired target: a sine function. Now each E unit will receive a sinusoidal input as well as input from other units in the network. If the input weights are stronger than the recurrent weights, most of the E units will oscillate with the same period

as the input. Given this scenario it is easy to find a set of W_{Out} weights that produces a sinusoidal output in the readout unit. So now we have our desired sinusoidal output; of course, we are cheating because we are feeding in the precise input that we wanted as an output! But we can solve this by simply using the output as the input—in other words, using feedback. This approach, first described in the context of artificial neural networks by Jaeger and Haas (2004), has been termed an echo-state network. During training the feedback is clamped to the desired output and only the weights of the W_{Out} synapses are adjusted so that the output matches the desired target—afterwards, the feedback can be unclamped and run autonomously.

The challenge in changing weights in a network with feedback is instability: changing the weights onto the readout → alters feedback from the readout → which modifies the dynamics of the recurrent network → requiring further changes in W_{Out}. In the echo-state network this problem was avoided by clamping the output to the desired target during training. Here, the authors describe a novel, more powerful technique to train these feedback recurrent networks (inset of Figure 1B)—even when internal weights are strong enough to generate complex spontaneous activity. In this method, referred to as FORCE (first-order reduced and controlled error) learning, the feedback loop is always operational, but the error (the difference between target and output) is kept very small by rapid changes in W_{Out}. As shown in Figure 1B, FORCE generates a sinusoid output from a network that exhibits complex patterns of spontaneous activity in the absence of feedback (g = 1.5). But the FORCE learning rule goes far beyond generating arbitrary waveforms. Indeed, as shown in the paper, multiple input and output units can be incorporated into the circuit to generate networks that can operate in different modes, governed by different

input "switches." In one compelling example the authors simulate human locomotion using a set of 95 different outputs (each representing a joint). Depending on the input pattern, the network was capable of generating either walking or running behavior. These results are fundamental in that the same recurrent network can operate in different "modes."

While the operational principles described are the same as in the echo-state network, FORCE learning seems significantly more powerful in finding optimal sets of W_{Out} weights and in tapping into the computational power of networks exhibiting chaotic behavior. Indeed, up to a point, the presence of chaotic behavior (higher g values) improved the ability of the network to learn to generate a complex periodic output. It is suggested that this result supports the notion that networks operating on "the edge of chaos" are optimally suited for some computations. Networks with higher g values are likely more effective because they generate a richer set of activity patterns and a longer lasting internal "memory."

Perhaps the most exciting contribution is the demonstration that the learning rule can be applied to the synapses within the recurrent network. One previous report (Maass et al., 2007) has taken a related approach, but Sussillo and Abbott show that *all* synapses can be modified using FORCE and that the learned component reflects the desired target pattern. In these simulations all units were updated according to the same rule, but interestingly, the same global error signal, based on the output unit, was used for all cells. The applicability of FORCE to all synapses in a recurrent network is biologically important since it is well established that recurrent synapses between pyramidal neurons are plastic. Intuitively, it might seem that plasticity within the recurrent network should offer computational advantages; surprisingly, performance did not appear to be significantly better than when plasticity was limited to the W_{Out} synapses. It is possible that future studies will reveal situations in which this is not the case;

however, it is also possible that since the feedback from the readout neuron is a particular case of a recurrent network (imagine the readout unit being dragged into the recurrent network), these architectures are related.

FORCE provides a powerful algorithm for training recurrent networks, but it is far from clear whether it should be thought of as a biologically plausible model. First, all simulations used a firing rate model as opposed to spiking neurons. Maass et al. (2007), however, have implemented related feedback principles using spiking neurons, thus FORCE may generalize to more realistic neural simulations. Second, the authors focus primarily on a weight update algorithm termed recursive least-squares (Haykin, 2002). This rule relies on a running estimate of the correlation matrix of the inputs onto a postsynaptic neuron (and requires that each synapse "know" the input to other synapses), and while powerful, it is not biologically plausible. However, the authors also show that FORCE works with the more biologically plausible, yet less powerful, delta rule (coupled with a variable learning rate). Third, to maintain the small error in the feedback loop, the change in the synaptic weights has to be in real time and very quick. Given the speed requirements and complexity of the rule, it seems likely that future research aimed at adapting FORCE to the biology may prove more fruitful than searching for biological mechanisms that fit FORCE.

As the authors acknowledge, FORCE might best be viewed as a valuable technique for building and, perhaps more importantly, understanding recurrent networks. For example, in Figure 1B it is instructive to ask where the sinusoid is stored. Clearly the modified W_{Out} weights are critical, but so are the random weights within the recurrent network (shuffling W_{EE} produces an entirely different behavior). It is rather illuminating to realize that having the complete connectome of the recurrent network (in the absence of the nature of the input and feedback signals) would not provide any insight into the fact that it was involved in generating a sinusoid or in explaining

where the sine wave was "stored." One potential lesson is that perhaps much of the synaptic plasticity occurring within recurrent cortical networks is not aimed at learning specific stimuli, but at ensuring that networks are in an optimal regime. Homeostatic forms of plasticity would appear likely candidates; however, experimentally derived forms of homeostatic plasticity, such as synaptic scaling (Turrigiano, 2007), are unstable in recurrent networks (Buonomano, 2005; Houweling et al., 2005). Yet, it is possible that homeostatic plasticity in combination with correlation-based forms of plasticity may play an important role in tuning cortical networks "off-line" to ensure that they operate in a receptive regime—one equivalent to the optimal g values reported in Sussillo and Abbott (2009).

REFERENCES

Buonomano, D.V. (2005). J. Neurophysiol. *94*, 2275–2283.

Buonomano, D.V., and Maass, W. (2009). Nat. Rev. Neurosci. *10*, 113–125.

Dan, Y., and Poo, M.M. (2004). Neuron *44*, 23–30.

Haykin, S. (2002). Adaptive Filter Theory (Upper Saddle River: Prentice Hall).

Hopfield, J.J. (1982). Proc. Natl. Acad. Sci. USA *79*, 2554–2558.

Houweling, A.R., Bazhenov, M., Timofeev, I., Steriade, M., and Sejnowski, T.J. (2005). Cereb. Cortex *15*, 834–845.

Izhikevich, E.M. (2006). Neural Comput. *18*, 245–282.

Jaeger, H., and Haas, H. (2004). Science *304*, 78–80.

Legenstein, R.A., Pecevski, D., and Maass, W. (2008). PLoS Comput. Biol. *4*, e1000180.

Maass, W., Joshi, P., and Sontag, E.D. (2007). PLoS Comput. Biol. *3*, e165.

Rumelhart, D.E., McClelland, J.L., and University of California San Diego, PDP Research Group. (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Cambridge, MA: MIT Press).

Sompolinsky, H., Crisanti, A., and Sommers, H.J. (1988). Phys. Rev. Lett. *61*, 259–262.

Sussillo, D., and Abbott, L.F. (2009). Neuron *63*, this issue, 544–557.

Turrigiano, G. (2007). Curr. Opin. Neurobiol. *17*, 318–324.