Reservoir Computing: Harnessing a Universal Dynamical System

By Daniel J. Gauthier

T

here is great current interest in develop-
ing artificial intelligence algorithms for processing massive data sets, often for classification tasks such as recog-
nizing a face in a photograph. But when we think of learning to perform a deterministic dynamical system? Relevant applications include forecasting the weather, control-
ing complex dynamical systems, and fing-
gering radio-frequency transmitters to secure the internet of things.

Training a "universal" dynamical sys-
tem to predict the dynamics of a desired system is one approach to this problem that is well-suited for a reservoir computer (RC): a recurrent artificial neural network for processing time-dependent informa-
tion (see Figure 1). It can operate in many modes, including prediction mode, the task described above. While researchers have studied RCs for well over 20 years [1] and applied them successfully to a variety of tasks [2], there are still many open questions about how the dynamical systems community may find interesting and be able to address. An RC is distinguished from traditional feed-forward neural networks by the following qualities:

• The network nodes each have distinct dynamical properties
• Time delays of signals may occur along the network links
• The network’s hidden part has recur-
rence connections
• The inputs and internal weights are fixed and chosen randomly
• Only the output weights are adjusted during training.

The last point drastically speeds up the training process.

Mathematically, an RC is described by the set of autonomous, time-delay differential equations given by

\[
\frac{dx}{dt} = -\gamma_i x + f_i \sum W_{ii} u_i(t) + \sum W_{ij} x_j(t - \tau_{ij}) + b_i,
\]

where \(J \in \mathbb{R}^N \) is the input layer, \(W_{ii} \) is a reservoir node index, and \(K \) is the output layer with \(y \). Here, \(\gamma_i \) are decay constants, \(W_{ii} \) are fixed input (internal) weights, \(\tau_{ij} \) time delay, \(b_i \) biases, and \(\mathbf{w}_i \) are the output weights whose values are optimized for a particular task. The non-
linear function \(f \) is typically sigmoidal, which we can take to the limit of an on-off thresholding (Boolean) function, as is done in traditional Hopfield net-
works. The reservoir maps the input signals to the output signals as is done in traditional Hopfield nets. A single RC can learn a large number of tasks, while simultaneously allowing the network to different inputs and increases noise tolerance. We can also find a solution to (2) using gradient descent meth-
ods, which are helpful when the matrix dimensions are large, and leverage tool-
kits from the deep learning community that take advantage of graphical process-
ing units. Use of recursive least-squares is another approach.

An RC can work very well in the pre-
diction task. For example, it is possible to learn the parameter of a dynamical sys-
tem when the reservoir dynam-
ics is projected to a lower-dimensional phase space before training [3]. We can also learn the attractor with standard training approaches and accurately predict Lyapunov exponents from the time series produced by the RC, even for spatial-temporal dynamics [7]. Furthermore, we can utilize the predicted time series as an observer in a control system [4] or for data assimilation of large spatiotemporal systems without use of an underlying model [6]. These results suggest that an RC is a powerful tool for characterizing complex dynamical systems.

While these conclusions are compel-
ling, designing an RC for a particular task is largely a trial-and-error undertaking, and authors tend to present results that work without dwelling on those that fail. The following is an open question: how can we optimize the parameters in (1) and (2) to obtain the most accurate prediction in either the prediction or classification tasks, while simultaneously allowing the RC to function well on data that is simi-
lar to the training data set? Early studies focused on the so-called echo state prop-
erty of the network—where the output should eventually forget the input—and the consistency property, where outputs from identical trials should be similar over some period. These conditions were initially assumed to be guaranteed when the spectral radius of \(\mathbf{w}_i \) is less than one (for the case when \(\gamma_i \) = 0). However, this scenario ignores the input dynamics and is not always a simple and straightforward answer. Based on the known chaotic behavior of the reser-
voir map, the spectral radius does not guarantee the stability of the network. Recent work is beginning to address this shortcoming for the case of a single input channel, demonstrating that there must be a single output solution given the input [5].

While a base of past research exists, many questions that demand quantita-
tive answers remain open. For example, how large must \(\gamma \) be to achieve a desired error rate? How should we adjust \(\gamma \), rela-
tive to the timescale of the dynamical sys-
tem? Why do sparsely-connected reservoirs often perform best?

At the 2017 SIAM Conference on Applications of Dynamical Systems, held in Snowbird, Utah, last May, Edward Ott and I organized a minisymposium on RCs to discuss these and other problems. Ott showed that RCs can learn the “climate” of a dynamical system and accurately forecast spatiotemporal chaos in a scalable manner. Roger Brockett indicated that dense network connections might give rise to partial or full synchronization of the reservoir nodes, thus diminishing the diversity of waveforms that an RC can learn. Brian Hunt suggested that an RC must synchronize to the input data in a generalized sense when used for the prediction task. Finally, I discussed a hardware-based RC capable of predicting at a rate exceeding tens of MHz.

In summary, RCs can serve as a universal dynamical system capable of learning the dynamics of other systems. This may prove advantageous when obtaining data for the learned dynamical system is expensive or difficult, for example. While the field is progressing rapidly, there are still substan-
tial openings for others to join the effort.

Acknowledgments: I gratefully acknowledge discussions of this work with Roger Brockett, Daniel Canaday, Ingo Fischer, Michelle Girvan, Aaron Griffith, Alexander Hartemink, Nicholas Haynes, Brian Hunt, Zhixun Lu, Edward Ott, and Jaideep Pathak, and the financial support of the U.S. Army Research Office Grant No. W911NF12-1-0099.

Figure 1. Illustration of the reservoir computer architecture. Figure credit: Daniel J. Gauthier.

ACKNOWLEDGMENTS

The 2018 SIAM Annual Meeting is co-
located with the 2018 SIAM Conference on Mathematical Aspects of Materials Science. My invited talk, which is part of both meetings, will develop the afore-
tioned topics, focusing on recent examples in which the identification of energy scaling laws has produced some interesting surprises.

References


tems. Chaos, 27, 041102.


ing to replicate chaotic attractors and cal-

Daniel J. Gauthier is a professor of phys-
ics at Ohio State University and is inter-
ested in the broad range of topics in dynami-
cal systems, quantum optics, and quantum information science.

Wrinkles and Folds

Continued from page 9

bound demonstrate the adequacy of the model when they are close to agreement, and the underlying arguments help explain why certain configurations are preferred.

Wrinkling is observed in a diverse array of different situations. When a sheet is under tension, any wrinkles must be par-
allel to the tensile direction, and under-
standing the length scale of wrinkling is the main goal. The situation is more com-
plex when wrinkling serves to avoid biaxial compression. In such situations, even the orientation of the wrinkling is far from clear.