Parallel Machine Learning Prediction of Network Dynamics

Joy Hamlin (SUNY Stony Brook), Nolan J. Coble (SUNY Brockport),
Keshav Srinivasan (UMD), Michelle Girvan (UMD), Ed Ott (UMD), Tom Antonsen (UMD)

Summary

- Machine learning prediction of a network of dynamic systems is a task with many potential applications
- Using a single machine learning device becomes increasingly difficult as the system size increases
- We devised and tested a prediction scheme that uses many machine learning devices acting in parallel
- This prediction method scales naturally to arbitrarily large systems
- Proof of principle tests were done on the Kuramoto oscillator model

Background: Reservoir Computing Prediction

- Machine learning technique that uses a neural network known as a "reservoir" to predict the evolution of a dynamical system
  - Neural network: sparsely connected nodes that evolve over time, similar to neural activity
- Reservoir has no knowledge of system dynamics; it adapts to known training data for the system
- Method:
  1) Listening: input system states **u**(t) and obtain output state estimates $\tilde{\mathbf{u}}(t+\Delta t)$
  2) Training: adjust output layer such that $\tilde{\mathbf{u}}(t+\Delta t)$ closely approximates **u**(t+Δt)
  3) Prediction: use $\tilde{\mathbf{u}}(t+\Delta t)$ from output as the next input to reservoir
- Reservoir must be large enough to predict a system, but for large interconnected systems this size requirement can be computationally infeasible

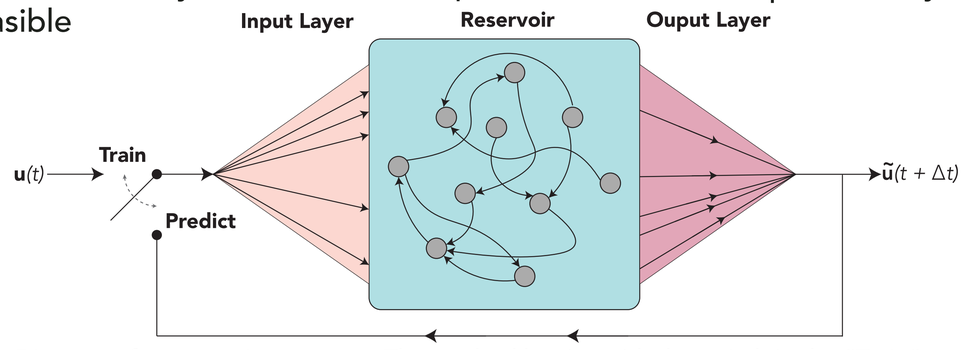

Figure 1. Diagram of a reservoir computer. The method begins with the switch in the "Train" position. An input layer couples the system state to the reservoir nodes, and an output layer retrieves an evolved state from the reservoir. Once the reservoir has processed the training data, the output layer is adjusted so that the output states closely approximate the true states for the system. The switch is then moved to the "Predict" position. Now, the most recent reservoir state is fed back into the reservoir, and a new output state is obtained. This process continues, allowing the prediction to evolve without any data of the true system state.

Method: Parallel Prediction

- Assign a small reservoir to each node, i, of a network-coupled system
- Train each reservoir with known data from its assigned node and that node's neighbors
- Predict the assigned node to obtain output vector $\tilde{\mathbf{u}}_i(t+\Delta t)$ and send this output to each neighboring reservoir to use for the next prediction step
- A similar parallel prediction scheme has been shown to work for spatiotemporally dynamic systems [J.Pathak, et al., Phys. Rev. Lett. (2018)]

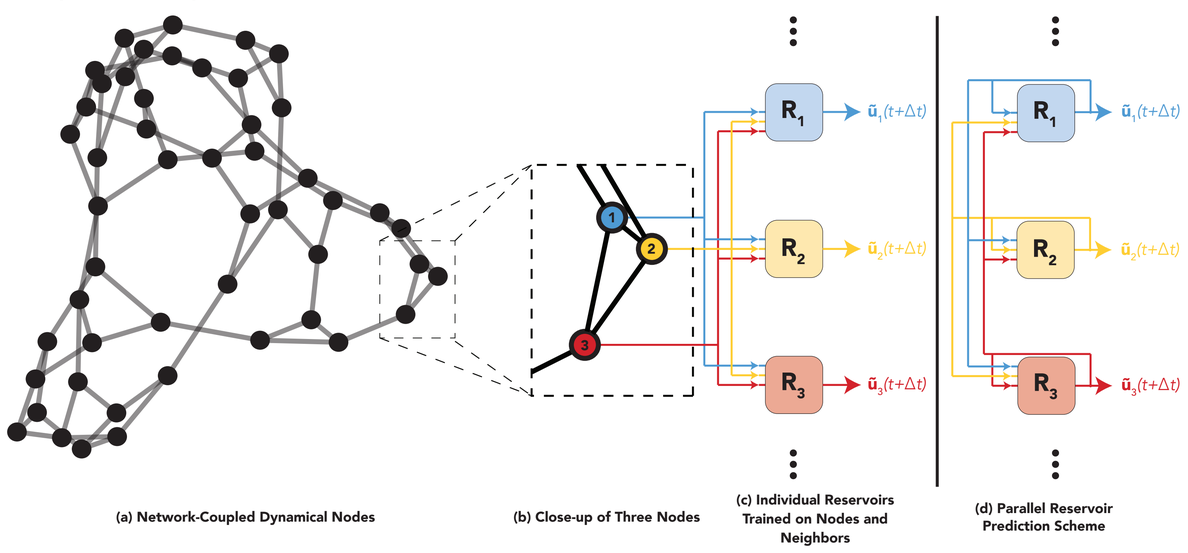

Figure 2. (a) Example of a network with dynamical nodes. (b) Portion of the network. (c) Training Phase: a single, small reservoir is assigned to each node of the system. Each reservoir receives an input state $\mathbf{u}_i(t)$ from its assigned node and the nodes that neighbor it. Output states $\tilde{\mathbf{u}}_i(t+\Delta t)$ are obtained from each reservoir, and each output layer is individually adjusted so that every $\tilde{\mathbf{u}}_i(t+\Delta t)$ is approximately $\mathbf{u}_i(t+\Delta t)$. (d) Prediction phase: the output states $\tilde{\mathbf{u}}_i(t+\Delta t)$ are fed back into the assigned and neighboring reservoirs. The predictions then evolve without any knowledge of the true dynamical state of each node. In panels (c) and (d) only the connections between R_1, R_2, and R_3 are shown. Other inputs are omitted for simplicity.

Test System: Kuramoto Model

- System of N oscillators described by their phase angles, θ_i

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{D}\sum_{j=1}^{N} A_{ij}\, sin(\theta_j - \theta_i)$$

- ω_i: natural frequencies chosen from a uniform distribution from -π/2 to π/2
- K: strength of coupling
- D: constant number of neighbors per oscillator
- A: connectivy matrix with frequency assortativity (oscillators with similar ω_i are preferentially connected)
- Our test system uses N = 50, natural scales because of our parallel structure
- For the reservoir, vectors $\mathbf{u}_i(t)$ are described by $[\sin(\theta_i),\cos(\theta_i)]$

Results

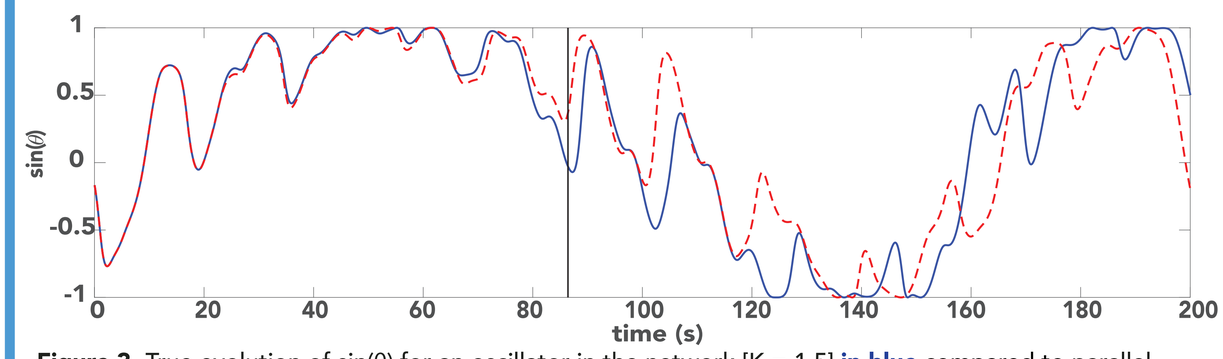

Figure 3. True evolution of sin(θ) for an oscillator in the network [K = 1.5] **in blue** compared to parallel prediction result for the same oscillator **in red**. The vertical black line marks the valid time of the prediction when the error reaches the threshold set (0.4). Even after the valid time, the prediction evolves similarly to the true state.

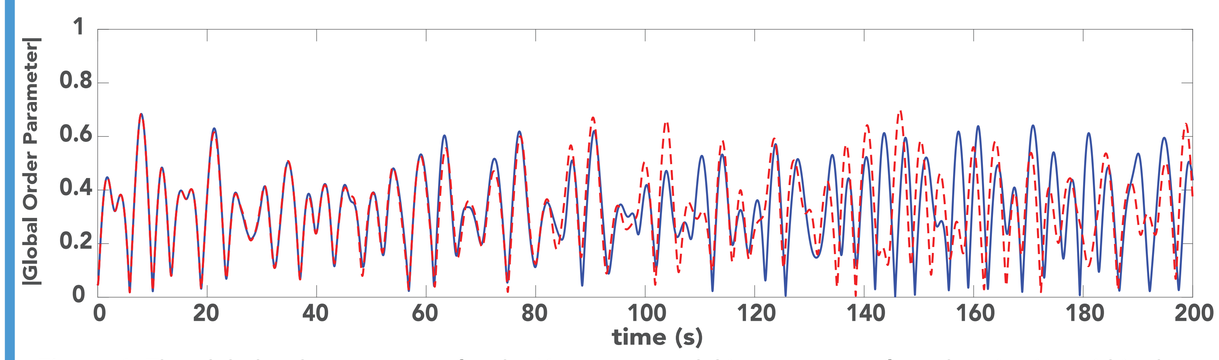

Figure 4. The global order parameter for the Kuramoto model is a measure of synchronization within the system, and it is calculated from the angles of each oscillator at a given time. A magnitude of 1 describes a completely synchonized system whereas a magnitude of 0 describes a desynchronized state. The figure above shows the true magnitude of the global order parameter over **in blue** compared to the results from the parallel prediction **in red**.

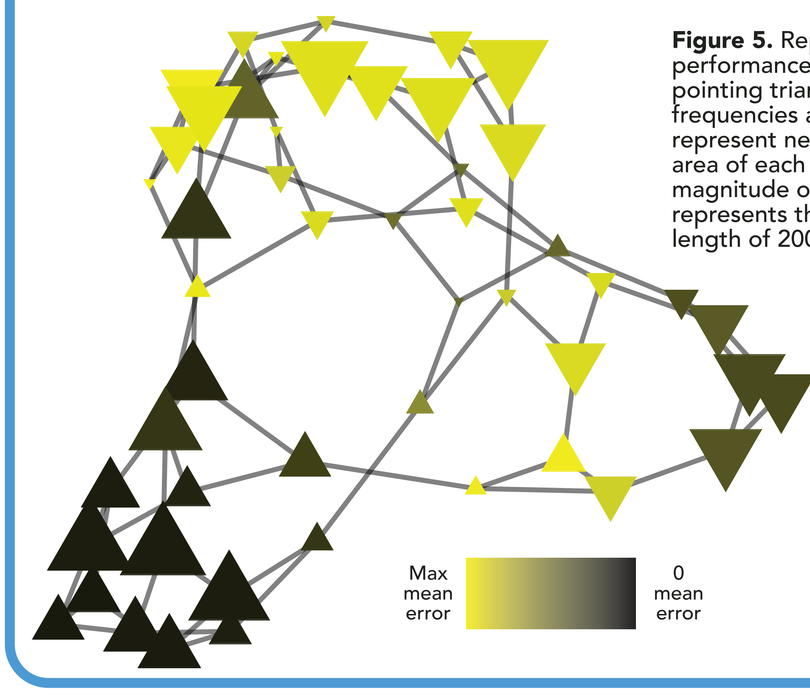

Figure 5. Representation of parallel prediction performance for our 50 node system. Upward pointing triangles represent positive natural frequencies and downward pointing triangles represent negative natural frequencies. The area of each triangle is proportional to the magnitude of the natural frequency. Color represents the mean error across the prediction length of 200 seconds.

Conclusion

- For networks of interest in applications the number of nodes can be very large, e.g., ~10^6 for some social networks.
- Prediction of global and individual nodal state dynamics is often of great use in such situations.
- We have presented a machine learning scheme which, through parallelization, makes it feasible to accomplish such tasks.

TREND 2019
Training and Research Experiences in Nonlinear Dynamics

REU program sponsored by the
National Science Foundation
Award Number: PHY1756179