Predicting Network Dynamics with a Parallel Machine Learning Approach

Nolan J. Coble (SUNY Brockport), Joy Hamlin (SUNY Stony Brook)

Keshav Srinivasan, Michelle Girvan, Ed Ott, and Tom Antonsen
Network Problem

- Predicting large dynamical networks (social networks, biological networks) is important to research

- Approach: Use methods of reservoir computing (a type of machine learning) to predict network dynamics

- Extension for large systems: Use multiple reservoirs in parallel to predict the dynamics
Reservoir Computing

- Machine learning technique for predicting time series
- Uses an artificial neural network capable of complex dynamics, known as a "reservoir"
- Reservoir: sparse, randomly connected nodes which contain information that evolves over time
Reservoir Training

Adjust weights in output layer so that $\tilde{u}(t + \Delta t)$ closely approximates $u(t+\Delta t)$
Steps for prediction phase:

1. Use the last training state as input to the reservoir.
2. At each time step, use the previous output as the new input.
Parallel Training Scheme for the Network Problem:

1. Assign a small reservoir to each node
2. Train each reservoir with data from its assigned node and that node’s neighbors
3. Adjust the output weights so the reservoir output matches the assigned node’s data
Parallel Training Scheme for the Network Problem:

1. Assign a small reservoir to each node
2. Train each reservoir with data from its assigned node and that node's neighbors
3. Adjust the output weights so the reservoir output matches the assigned node's data
Parallel Training
Scheme for the Network Problem:

1. Assign a small reservoir to each node
2. Train each reservoir with data from its assigned node and that node’s neighbors
3. Adjust the output weights so the reservoir output matches the assigned node’s data
Parallel Training Scheme for the Network Problem:

1. Assign a small reservoir to each node
2. Train each reservoir with data from its assigned node and that node’s neighbors
3. Adjust the output weights so the reservoir output matches the assigned node’s data
Parallel Training Scheme for the Network Problem:

1. Assign a small reservoir to each node
2. Train each reservoir with data from its assigned node and that node’s neighbors
3. Adjust the output weights so the reservoir output matches the assigned node’s data
Parallel Training Scheme for the Network Problem:

1. Assign a small reservoir to each node
2. Train each reservoir with data from its assigned node and that node’s neighbors
3. Adjust the output weights so the reservoir output matches the assigned node’s data
Parallel Prediction Scheme

1. Use the last training state of each reservoir as input to the assigned and neighboring reservoirs.
2. For each time step, use the reservoir outputs as the next input.
Test System: Kuramoto Model

- System of N oscillators described by their phase angles, $\theta_i$

\[
\frac{d\theta_i}{dt} = \omega_i + \kappa \sum_{j=1}^{N} A_{ij} \sin(\theta_j - \theta_i)
\]

- $\omega_i$: natural frequency
- $\kappa$: strength of coupling
- $A$: connectivity matrix with frequency assortativity
Results: Individual Node Performance

Blue: True State  Red: Prediction

Oscillator #10  \(\omega: -0.18129\)

Oscillator #20  \(\omega: 0.64201\)
Blue: True State
Red: Prediction

Results: System Performance
Future Work

- Test parallel scheme on other network models (e.g. FitzHugh-Nagumo model for neuron activity)
- Reduce the number of reservoirs by having each predict multiple nodes (beneficial for very large networks)
- Extend the parallel scheme for situations where the network connections are unknown
Thank You!

- Advisors: Michelle Girvan, Ed Ott, and Tom Antonsen
- Keshav Srinivasan
- Daniel Serrano, Taylor Prendergast
- TREND Cohort