Information processing using time-delay autonomous Boolean networks

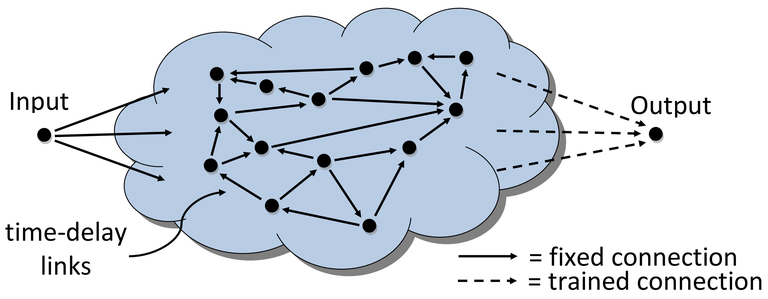
Nicholas D. Haynes,[1] Stefan Apostel,[1,2] Otti D'Huys,[1] and Daniel J. Gauthier[1,3]

[1]Department of Physics, Duke University, Durham, NC, USA

[2]Institut für Theoretische Physik, Technische Universität Berlin, Germany, EU

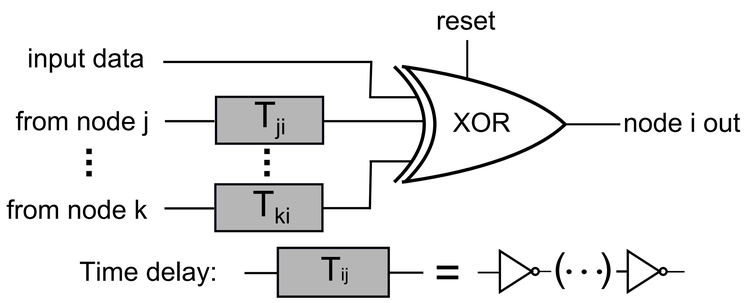
[3]Department of Physics, The Ohio State University, Columbus OH, USA

Reservoir Computing



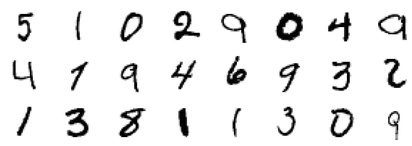
- Reservoir computing is a method for **efficiently training recurrent networks of nonlinear elements** to perform a task [1, 2]
- The "reservoir" is a network of nodes with **arbitrary (but fixed) connections**
- Input and reservoir connections are kept fixed, **output connections are determined by linear regression** with target outputs

Experimental system



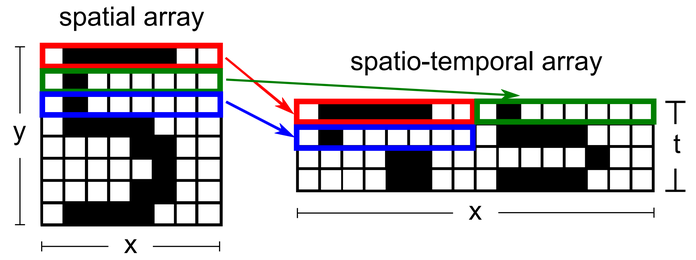
- Field-programmable gate arrays (FPGAs) contain $10^4 - 10^6$ digital logic elements that can be **wired arbitrarily** using a hardware description language + compiler
- An autonomous Boolean network of N nodes with time-delay links is constructed using an Altera Cyclone IV FGPA
- Reservoir nodes constructed from **unclocked (autonomous) exclusive-OR (XOR) gates**
- Time delays are added along reservoir links with pairs of inverter gates
- M output lines are connected to random nodes ($1 \leq M \leq N$)

The MNIST dataset

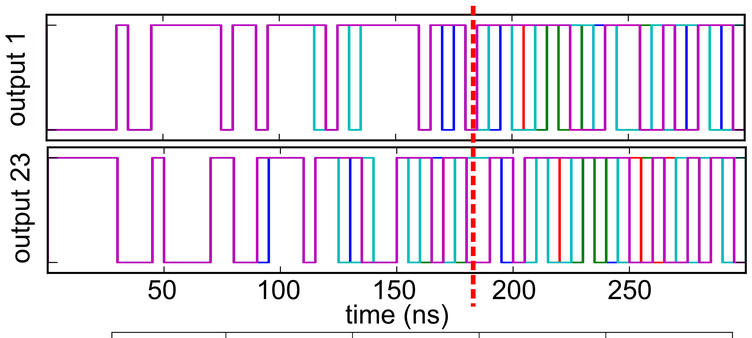


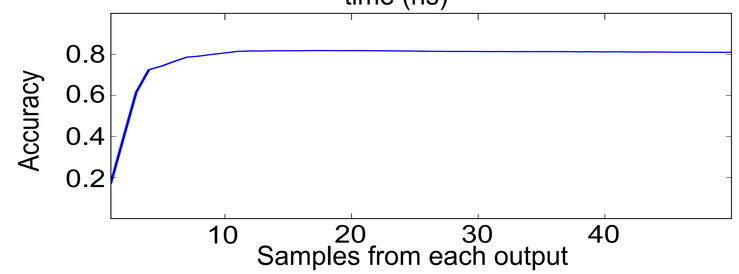
- Dataset of 70,000 handwritten digits (0-9) and labels
- Each image is 28 x 28 grayscale (0-255) pixels
- Standard benchmark problem commonly used to test new machine learning algorithms

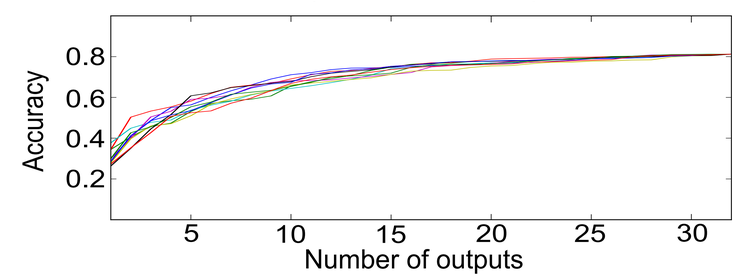
Results



- Reservoir is 112-node Erdös–Rényi network with an average connectivity of 3 and random (uniform) delays between 30 and 45 ns
- Pixels are Booleanized and reshaped from a 28 x 28 spatial array to a 112 x 7 spatio-temporal array; each of 112 reservoir nodes receives a unique 7-pixel input time series
- Once excited, **reservoir evolves chaotically** with a characteristic time scale of ~ 0.5 ns and a consistency window of ~ 200 ns
- 32 random reservoir nodes are chosen as output nodes, and time series samples are collected once every 5 ns from each output
- Classifiers for each digit are **trained using ridge regression** [3], and accuracy scores are calculated using **5-fold cross-validation**
- Performance is not affected by samples collected after $\sim$ 2 x mean delay time
- Additional outputs increase performance with **diminishing returns**







Future Work

- Examine effects of network parameters such as connectivity, delay, and Boolean sensitivity on MNIST performance
- Implement classification with digital logic for high-speed classification
- Explore the use of input masks for compressing input data

References

[1] Maass, W., Natschlaeger, T., and Markram, H. *Neural Comput.* **14**(11), 2531–2560 (2002).

[2] Lukoševičius, M. and Jaeger, H. *Comp. Sci. Rev.* **3**, 127–149 (2009).

[3] Horel, A. E. and Kennard, R. W. *Technometrics* **12** (1970).

The authors gratefully acknowledge the financial support of U.S. Army Research Office Grant #W911NF-12-1-0099 and National Science Foundation Grant #DGE-1068871.