Te Photonic Synapses for Physical Reservoir Computing



Prof. Hongseok Oh

Department of Physics, Department of Intelligent Semiconductors, and Integrative Institute of Basic Sciences (IIBS), Soongsil University, South Korea

Reference: Hyerin Jo[#], Jiseong Jang[#], Hyeon Jung Park[#], Huigu Lee, Sung Jin An, Jin Pyo Hong^{*}, Mun Seok Jeong^{*}, and Hongseok Oh^{*}, "Physical reservoir computing using tellurium-based gate-tunable artificial photonic synapses", *ACS Nano*, **18**, 44 (2024): 30761–73.

What is Reservoir Computing



Nonlinear dynamical systems



Natural systems

Engineering systems



Social systems



Biological systems



Recurrent Neural Network (RNN)



- For static input
- Perceptrons are used once for each input

Recurrent Neural Network



- For dynamic input
- Perceptrons are used repeatedly during the given input
- Output of nodes at 'n-1'th series becomes input of 'n'th series

Disadvantages of RNN

Source: https://iq.opengenus.org/disadvantages-of-rnn/

- Vanishing or exploding gradient problem
- Slow and complex training procedure



Reservoir computing

Ref: Lukoševičius, Mantas, and Herbert Jaeger. Computer Science Review 3, no. 3, 127–49.



- Reservoir: a set of *randomly created, fixed* recurrent neural network
 - It is passively excited by the input signal and maintains nonlinear transformation of the input history in its state.
- The desired output signal is generated as a linear combination of the readout nodes of the reservoir driven by the input signal
 - The linear combination is obtained by linear regression, using a teacher signal as a target

UAT for dynamical nonlinear systems



UAT for dynamical nonlinear systems



Reservoir computing for cost saving

Code by ChatGPT

Recurrent Neural Network

Reservoir Computing





of connections to train: 2651

=== RNN Computational Costs === Training time: 3.83 seconds Inference time: 0.2095 seconds Memory used: 63.27 MB



of Input size: 1 (20 time steps)# of nodes in reservoir: 100# of readout nodes: 100

of connections to train: 100

=== RC Computational Costs === Training time: 0.02 seconds Inference time: 0.0015 seconds Memory used: 5.57 MB

From Reservoir Computing To Physical Reservoir Computing



'Physical' reservoir computing

A reservoir can be a physical system...

Reservoir Computing (RC)

Physical Reservoir Computing (PRC)



Trained connections

As long as the physical system has the following features:

- Have a short-term internal memory fading memory
- Nonlinear transformation of the input data
- Given the same input, same output should be guaranteed

Can water compute?



- Same drops: same ripples propagate (Same input same output)
- Multiple ripples can interfere (Nonlinear transform)
- Ripples will fade away (Fading memory)

Reservoir computing from a water bucket

Source: TU Berlin (Dr. Manish Yadav et al.)



Physical Intelligence: AI in a Bucket of Water

Supervisor: Dr. Manish Yadav; Participants: Filip Gerhard Landin, Yueyin Luo, Luis Emilio Zaldivar Hanke, Samuel Gorzalnik, Mahdi Ghorbaniaghooyeh,Tatiana Vasilyeva

Technical University Berlin

Berlin University Alliance



Introduction

The presented work demonstrates the potent al of reservoir computing as a resource-ef cient alternative to traditional deep learning models. By leveraging a complex dynamical system as the computational core, it highlights the capability of reservoir computing to process and analyze data in innovative ways.

Reservoir Computer

Reservoir computing employs a fixed, randomly interconnected network—the reservoir—to transform input signals into a higher-dimensional space. This transformation facilitates the application of straight orward linear methods. Renowned for its proficiency with time-series data and dynamic systems, reservoir computing demands minimal training. The diagram below illustrates this computed to nal framework.



Time increment: $\mathbf{r}_{t+1} = (1 - \alpha)\mathbf{r}_t + \alpha \cdot \sigma(\mathbf{W}_{res} \cdot \mathbf{r}_t + \mathbf{W}_{in} \cdot \mathbf{x}(t))$

Training: w_i : mink $W_{out}\mathbf{r}(t) - \mathbf{y}(t)k$

- Predict on: $\mathbf{y}(t) = \mathbf{W}_{out}\mathbf{r}(t)$
- $\mathbf{W}_{res} \in \mathbb{R}^{d \times d}, \mathbf{r} \in \mathbb{R}^{d \times 1}, \mathbf{x}(t) \in \mathbb{R}^{m \times 1}, \mathbf{W}_{in} \in \mathbb{R}^{d \times m}, \mathbf{W}_{out} \in \mathbb{R}^{n \times d}$
- Explainable model
- Reservoir: Fixed (randomly generated) graph
- . Ef clent training: Only the readout weights W at need to be trained

Autonomous Reservoir Comput ng



- Autoregressive model: Predict ons based on learnt weights and previous states (self-feeding loop)
- Temporal Informat on Storage: Reservoir can retain informat on from past inputs, enabling the network to ut lize memory capabilit es for

t me-dependent predict ons or dynamic temporal pat ern recognit on

Conference XX

The figure below illustrates our experimental setup for problem-solving using a physical reservoir computer, represented here by a container of water. Input signals are converted into mechanical perturbations, which generates awares on the water's surface. These waves interact in complex pat erns, act ng as the computat onal medium. An image of the wave pat erns is then analyzed to extract the system's response to the inputs.

Physical Reservoir - Setup



Solving XOR with a Physical Reservoir

- Classical non-linear binary classif cat on problem
- . Can be solved linearly in a higher dimensional space
- Solut on: Map inputs into higher dimensional feature space by using the inherent non-linearity of the waters dynamics



Input:(1,1), class: 1 Input:(0,0), class: 1 Input:(1,0) class: 0 Input:(0,1) class: 0

Procedure:

- Encode XOR Input: Signals trigger the motors → generat on of waves in the physical reservoir
- Recording Image Data: Take a picture of the water surface shortly af er motor act vat on
- Image Preprocessing: Cropping the edges → edge detect on with sobel filter (taking magnitude pixelwise) → non-max-suppression → resizing
- 4. Model: CNN with linear act vat on funct ons (linear model) → predict ons



Solving XOR with a Physical Reservoir - Results We used a model comprising one convolut onal layer with 32 filters (3× 3), a Max-Pooling-Layer (2× 2), a Fully Connected Neural Network with 64 neurons and one ouput neuron. In the hidden layers we only used linear act vide on funct ons.



Model	BCE-Loss	Accuracy
Linear Model	0.5491	82.57%
Non-Linear Model (ReLU)	0.3007	90.04%

Music Generation with a Reservoir Network

We used the "Children's Song Dataset" for training the model and introduced following simplifying constraints:

- Only seven dist nct musical notes (C-scale)
- . In case of chords only the lowest note is used
- No rest and equal durat ons

The reservoir consists of a Erdös-Renyi graph $W_{BB} \in \{0,1\}^{n\times N}$ of N nodes and connect on probability δ . An input vector $x \in C^n$ of length n will first be transformed into a matrix $W_{n} \in \{0,1\}^{N\times n}$, where the 1-th column of W_{n} is a one-hot encoding of the 1-th note in the sequence x. Then, the state of the reservoir at time-step (t+1), R_{n+1} is computed. For predict ons a Feed-Forward Neural Network was applied. The model learns a distribut on of the next note to play given R_{n+1} i.e. on the given musical sequence. To generate music we feed back the predict ons learn vely to compute the next states.

Music Generation with a Physical Reservoir

- Time series need to be converted into signals for motor act vat on
- Memory-capacity of the fading waves allows for predict ons of future states based on previous inputs
- · Pretrained linear model predicts based on reservoir inputs
- Predict ons need to be converted to signals → motor act vations → new reservoir state

manishyadav@tu-berlin.de

Towards practical applications



- Electronic devices are generally preferred for their versatile applicability
- Making an electronic reservoir requires complicated circuits and connections
- Simple and programmable method to utilize electronic devices as physical reservoir

From Delay-coupled RC to Dynamic devices RC

Ref: X. Liang, J. Tang, Y. Zhong, B. Gao, H. Qian, H. Wu, Physical reservoir computing with emerging electronics. Nature Electronics **7**, 193–206 (2024).



- Delay-coupled RC requires signal masking and delayed feedback in a node, which is usually a circuit module
 - State richness is achieved by time multiplexing
- Dynamic devices RC utilizes nonlinearity and fading memory characteristics of a single device and utilize multiple devices in parallel
 - State richness is achieved by *device-to-device variation*

Case study: Physical Reservoir Computing Using Tellurium Photonic Synapses

Reference: Hyerin Jo[#], Jiseong Jang[#], Hyeon Jung Park[#], Huigu Lee, Sung Jin An, Jin Pyo Hong^{*}, Mun Seok Jeong^{*}, and Hongseok Oh^{*}, "Physical reservoir computing using tellurium-based gate-tunable artificial photonic synapses", *ACS Nano*, **18**, 44 (2024): 30761–73.

Te thin film photonic synapse



Collaboration with:

Prof. Munseok Jeong (Hanyang Univ.)

Prof. Jin-Pyo Hong (Hanyang Univ.) Co-first authors: Ms. Hyerin Jo Mr. Jiseong Jang Dr. Hyeon Jung Park

C. Zhao, C. Tan, D.-H. Lien, X. Song, M. Amani, M. Hettick, H. Y. Y. Nyein, Z. Yuan, L. Li, M. C. Scott, A. Javey, Evaporated tellurium thin films for p-type field-effect transistors and circuits. *Nat. Nanotechnol.* **15**, 53–58 (2020).

- Te thin film is now receiving a lot of interest as a candidate for high performance p-type materials
- Potential for synaptic CMOS system
- Multi-colored light applications (Small bandgap)

Te thin film photonic synapse



• Te thin film device with back-gate structure

Te TFT as p-type TFT



Small MNIST classification task



- No binarization: Used value as a LED intensity
- Final conductance values were used for learning
- # of connections for training: 64 (No RC) -> 16 (RC)
- In principle parallel computation is possible

System Implementation



 Repeat over entire "Small digit" dataset (1697 images from Scikit Learn)

System Implementation



Results



- Sample #: 1796, 70% for training, 30% for validation

Temperature dependent PPC



- Temperature dependent PPC with decreasing time constant at elaborated temperature
- The time constant can be tuned by the gate bias
- The energy related to trapping of photo-generated carriers are expected to be 166 – 419 eV depending on the gate bias

Temperature dependent PPC



- Negative bias attracts holes to the front channel. They recombine with electrons in the trap sites.
- Positive bias pushes holes to the back channel, while the electrons are attracted to the front channel, leading to the prolonged recombination.



 Reservoir computing can be used to predict the solution of nonlinear equation

SynapSSU v0.11 Connection Control Box Data location and filename Time Display Measurement SMU1 (Drain) USB0::0x05E6::0x2450::04570654::INSTR Check Location: D:/Data Current date & time Time elansed for the current measurment BHN Select folder KEITHLEY INSTRUMENTS, MODEL 2400,04570654, 1, 7, 12b Filename: noname 2024/04/23 20:22:15 00:45:34.36 ABOBT SMII2 (Gate) USB0::0x05E6::0x2450::04584339::INSTR Check Parameter setup KEITHLEY INSTRUMENTS, MODEL 2400, 04584339, 1, 7, 12b Drai Equation Input sequence (random numbers) SMU3 (Light) USB0::0x05E6::0x2450::04603670::INSTR Check <Measurement settings> Drain (Memristor/Memtransistor) 율 KEITHLEY INSTRUMENTS MODEL 2400 04603670 1 7 12h Auto Zero 0 0: OFF/1: ON Gate (Synaptic transistor) Grapt WWWWW LED (Photonic synapse) Voltage source range 20 V 120 140 160 180 200 240 260 300 20 220 280 Refresh 40 Drain Bias -10 V Current sensing range 1E-6 A System log-0 V Gate Bias Trigger delay 0.0 s Graph1 IS DADIUS/UTHENU CHADU-UX02ED-0X2490-04970094-0-03VATD-72 10 V Pulse Voltage <'USBInstrument'('USB0::0x05E6::0x2450::04584339::0::INSTR')> Source delay 0.0 9 'USBInstrument'('USB0::0x05E6::0x2450::04603670::0::INSTR')>] Unit Pulse Width 200 ms 20 NPLC 1 number [2024-04-23 16:40:03] Measurement start! [2024-04-23 17:00:58] Measurement done! Unit Delay Width 200 ms [2024-04-23 18:01:06] Run Measurement: NARMA-2 Class Current compliance 10E-3 A Encoded voltage input signal (To LED) [2024-04-23 18:01:06] SMU_list: Response Encoding manager <'USBInstrument'('USB0::0x05E6::0x2450::04570654::0::INSTR')> Gate bias setup <'USBInstrument'('USB0::0x05E6::0x2450::04584339::0::INSTR')>, <'USBInstrument'('USB0::0x05E6::0x2450::04603670::0::INSTR')>1 <Measurement settings> Equation Length 350 Device response [2024-04-23 18:01:06] Measurement start! 0 0: OFE/1: ON Auto Zero Load equation [2024-04-23 19:32:51] Measurement done 20 4N 60 80 100 120 140 [2024-04-23 19:34:44] Run Measurement: NARMA-2 Class Voltage source range 20 V Calculate nulse sen Λ [2024-04-23 19:34:44] SMIL list: [<'USBInstrument'('USB0::0x05E6::0x2450::04570654::0::INSTR')>; Current sensing range 1E-6 A ('USBInstrument'('USB0::0x05E6::0x2450::04584339::0::INSTR')> Encoding Modulation -20 -2e-08 Trigger delay 0.0 s <'USBInstrument'('USB0::0x05E6::0x2450::04603670::0::INSTR')>1 [2024-04-23 19:34:44] Measurement start! Source delay 0,0 s [2024-04-23 19:36:00] Measurement done! -40 -4e-08 Vgs (start) 20 V [2024-04-23 19:36:07] Run Measurement: NARMA-2 Class Gate NPLC 1 number [2024-04-23 19:36:08] SMU_list: -20 V Vgs stop <'USBInstrument'('USB0::0x05E6::0x2450::04570654::0::INSTR')> -60 -6e-08 Current compliance 100E-3 A ('USBInstrument'('USB0::0x05E6::0x2450::04584339::0::INSTB'))> Vgs step -2,5 V <'USBInstrument'('USB0::0x05E6::0x2450::04603670::0::INSTR')>] LED bias setup -80 -8e-08 [2024-04-23 19:36:08] Measurement start! Reset parameters [2024-04-23 19:36:19] Measurement done! <Measurement settings> (V10-01) [2024-04-23 19:36:19] Run Measurement: NARMA-2 Class Drain pulse -100 0 0: OFF/1: ON -1e-07 Auto Zero [2024-04-23 19:36:20] SMU_list: Gate pulse <'USBInstrument'('USB0::0x05E6::0x2450::04570654::0::INSTR')> 20 V Voltage source range 'USBInstrument'('USB0::0x05E6::0x2450::04584339::0::INSTR')> 3 -120 -1,2e-07 LED pulse <'USBInstrument'('USB0::0x05E6::0x2450::04603670::0::INSTR')>1 Current sensing range 100E-3 A [2024-04-23 19:36:20] Measurement start! Drain Bias n v [2024-04-23 20:21:54] Measurement done! 140 -1.4e-07 0,0 s 3 Trigger delay 0 V Gate Bias Source delay 0,0 s 0 V -160 -1.6e-07 Pulse Voltage Gate NPLC 1 number Unit Pulse Width 10000 ms -180 -1.8e-07 Current compliance 100E-3 A Unit Delay Width 500 ms Recorded -200 -2e-07 current -220 -2,2e-07 SW Information -240 -2.4e-07 SynapSSU (v0,1 January 2024) 20 4٢ 80 100 120 140 Created by Prof. Honoseok Ob Time (s) Department of Physics, Soongsil University (SSU), South Korea Email: hoh331@gmail.com





• Accurate prediction of the solution is possible from this PRC approach

Conclusion



- Reservoir computing is a new energy-saving framework for learning nonlinear dynamical systems
- Physical reservoir computing replaces the software reservoir with a physical one to achieve maximized cost saving
- Dynamic devices can be used as a physical reservoir a new way to implement physical reservoir computing with existing electronics

Acknowledgement



Group members Professor:

Prof. Hongseok Oh Prof. Jaegoo Lee (Adjunct Professor w/ Samsung Electronics)

Graduate students:

Hyerin Jo, Hyunsoo Kim, Wonsuk Oh, Kyusun Han

Collaborators

Prof. Mun Seok Jeong (Hanyang Univ.) Prof. Jin Pyo Hong (Hanyang Univ.)







