A Model of Memristive Nanowire Neuron for Recurrent Neural Networks

Veronica Pistolesi[1], Andrea Ceni[1], Gianluca Milano[2], Carlo Ricciardi[3], and Claudio Gallicchio[1] *

1- Department of Computer Science, University of Pisa, Italy
2- Istituto nazionale di ricerca metrologica (INRiM), Italy
3- Department of Applied Science and Technology, Politecnico di Torino, Italy

Abstract. We propose a novel neural processing unit for artificial neural networks, inspired by the memristive properties of nanowires. Our analysis, framed within the Reservoir Computing paradigm, demonstrates the stability, short-term memory, and fading memory capabilities of the unit. Further experiments on assemblies of nanowire-inspired neurons show promising results in time-series classification tasks. Our introduced approach bridges analog neuromorphic hardware and AI applications, enabling efficient time series processing.

1 Introduction

Memristive nanowire-based neurons offer a novel and promising approach to neuromorphic computing. Leveraging the unique properties of memristors-electronic components whose resistance changes based on the history of applied voltage-these neurons emulate the dynamics of biological systems, particularly synaptic plasticity [1]. Memristors' inherent ability to "remember" past electrical states makes them ideal candidates for simulating temporally dynamic processes such as short-term plasticity, a cornerstone of adaptive neural computation. In neuromorphic systems, memristors are gaining traction as efficient analog components for replicating the complex dynamics of biological memory [2]. However, to integrate the realism of their continuous-time physics into digital Artificial Neural Networks (ANNs), discrete-time modeling is essential. Such adaptation bridges the gap between analog hardware and digital frameworks, enabling memristor-inspired neurons to be deployed within existing ANN architectures. This paper introduces a novel discrete-time model for a neural processing unit based on the physical principles of memristive nanowires. The model approximates the conductance dynamics of memristors and casts them as artificial neuron dynamics, making it suitable for digital systems. By adopting a Reservoir Computing (RC) framework [3], we analyze the stability, memory, and non-linear response of the proposed model, disentangling its intrinsic architectural properties from specific learning algorithms. Our analysis, further supported by preliminary experiments on classification tasks, paves the way for its application in time series processing, where short-term memory and adaptive nonlinear dynamics are critical.

*This work has been supported by NEURONE, a project funded by the European Union - Next Generation EU, M4C1 CUP I53D23003600006, under program PRIN 2022 (prj code 20229JRTZA), and by and EU-EIC EMERGE (Grant No. 101070918).

2 From Physics to a Single Neuron Memristive Nanowire Model

In [1], an ODE modeling the dynamics of a nanowire-based memristor is proposed. Such a model gives us the opportunity to draw a parallel between the physical properties of the memristor and neural mechanisms. Conductance in the memristor can represent the activation of a neuron (or hidden state), while the applied voltage acts as the external input to the neuron, as shown in Table 1.

Physics	**Informatics**
Continuous-time	Discrete-time
Conductance	Hidden state
Voltage	Input

Table 1: Short concept table linking physical and computational perspectives.

The conductance changes based on a potentiation-depression rate balance equation, which governs how the neuron's state adapts in response to voltage inputs over time. The corresponding ODE for the nanowire network system is:

$$\frac{dh}{dt} = k_p \cdot (1 - h) - k_d \cdot h, \tag{1}$$

where $k_p = \kappa_{p_0} \cdot e^{\eta_p V}$ and $k_d = \kappa_{d_0} \cdot e^{-\eta_d V}$. Here, the variable h models the normalized conductance of the nanowire memristors, and V is the applied input voltage. The constants $\kappa_{p0}, \kappa_{d0}, \eta_p, \eta_d$ represent physical quantities defining the nanowire material response to electrical stimulation. Physical constraints require $\kappa_{p0}, \kappa_{d0}, \eta_p, \eta_d > 0$, and $0 \leq h \leq 1$. eq. (1) captures the ability of a biological neuron to increase or decrease its conductance, mimicking how biological synapses strengthen or weaken temporarily based on recent activity.

2.1 The single neuron memristive nanowire model

We cast the physical nanowire memristive model into a single neural processing unit discretizing eq. (1) via the forward Euler numerical scheme, as follows:

$$h(t+1) = h(t) + \Delta_t\big(k_p(x(t+1)) - [k_p(x(t+1)) + k_d(x(t+1))]h(t)\big), \tag{2}$$

where Δ_t is a *step size* hyper-parameter for the numerical integration. Remarkably, a unique feature of this neural unit is its non-linearity. In fact, $k_p(x(t))$ and $k_d(x(t))$ provide a distinctive non-linear encoding of the time series $x(t)$, represented by a sum of the following two exponential functions:

$$k_p(x(t)) = \kappa_{p_0} \cdot e^{\eta_p x(t)}, \qquad k_d(x(t)) = \kappa_{d_0} \cdot e^{-\eta_d x(t)}. \tag{3}$$

This creates an asymmetric and strongly non-linear response that differs from conventional artificial neurons that use tanh or ReLU activations. The resulting neuron exhibits strongly nonlinear temporal dynamics, making it particularly suited for applications requiring short-term adaptive responses and quick-switching temporal behaviors.

3 Stability analysis

We aim to exploit the dynamical system eq. (2) as the recurrent unit of a recurrent neural network. Thus, we need to ensure eq. (2) converges to a stable state when driven with generic inputs, a stability requirement known as *echo state property* for RC networks [4]. An asymptotically stable input-driven solution ensures that the recurrent dynamics have reliable behavior, preventing unbounded responses and enabling consistent processing of sequential data. In the following theorem, we provide conditions under which eq. (2) admits an asymptotically stable input-driven solution.

Theorem 3.1. *If* $|1-\Delta_t(k_p(x(t+1))+k_d(x(t+1)))|<1$*, for all* $t>0$*, then it exists a unique input-driven solution that attracts all trajectories. Moreover, if the input* $x(t)$ *varies sufficiently slowly, then such a unique input-driven solution closely follows the moving fixed point* $h^*(t)=\frac{k_p(x(t+1))}{k_p(x(t+1))+k_d(x(t+1))}$.

Proof. For the sake of brevity, in this proof, we denote $k_p(x(t+1))$ and $k_d(x(t+1))$ simply as $k_p(t)$ and $k_d(t)$. Let be given two initial conditions $h(0),\hat{h}(0)$, from which follow two $x(t)$-driven trajectories $h(t),\hat{h}(t)$. Then, by a few algebraic manipulations and the triangle inequality, we get:

$$
\begin{aligned}
&|h(t+1)-\hat{h}(t+1)|= \\
&=\Big|h(t)+\Delta_t\big(k_p(t)-[k_p(t)+k_d(t)]h(t)\big)-\hat{h}(t)-\Delta_t\big(k_p(t)-[k_p(t)+k_d(t)]\hat{h}(t)\big)\Big| \\
&=|(h(t)-\hat{h}(t))\big(1-\Delta_t[k_p(t)+k_d(t)]\big)|\leq|h(t)-\hat{h}(t)|\,\big|1-\Delta_t[k_p(t)+k_d(t)]\big|.
\end{aligned}
$$

By recursion we have that $|h(t+1)-\hat{h}(t+1)|\leq|h(0)-\hat{h}(0)|\,\big|1-\Delta_t[k_p(t)+k_d(t)]\big|^{t+1}$. By hypothesis $\big|1-\Delta_t[k_p(t)+k_d(t)]\big|<1$, therefore $\lim_{t\to\infty}|h(t+1)-\hat{h}(t+1)|=0$. Necessarily, the system asymptotically converges towards a unique input-driven solution. Now, assume a constant input $x(t)\equiv x^*$, which is an infinitely slowly varying input signal. We have that a fixed point h^* must satisfy the equation $h^*=h^*+\Delta_t\big(k_p(x^*)-(k_p(x^*)+k_d(x^*))h^*\big)$, thus it must have the following form $h^*=\frac{k_p(x^*)}{k_p(x^*)+k_d(x^*)}$. If the input $x(t)$ varies slowly and the quantity $|1-\Delta_t[k_p(t)+k_d(t)]|$ is small, then the system quickly approaches the slowly varying fixed point $h^*(t)=\frac{k_p(t)}{k_p(t)+k_d(t)}$. □

The quantity $\rho(t)=|1-\Delta_t[k_p(x(t+1))+k_d(x(t+1))]|$ regulates the rate of convergence towards the unique attracting solution. The closer to zero this quantity, the faster the forgetting of the initial condition, i.e., the stronger the fading memory. For this reason we call $\rho=\sup_{t>0}\rho(t)$ the *(fading) memory factor*. Differently from the continuous time model eq. (1), in the discrete time version of eq. (2) there is the possibility of unstable single neuron dynamics, e.g. a too large Δ_t might cause wildly oscillating dynamics. For this reason, we introduce another hyper-parameter, ω, that we call *input scaling*, whose role is to tune the amplitude of the input, basically replacing $x(t)$ with $\omega x(t)$.

4 Experiments

We study the properties of the single nanowire neuron by analyzing the influence of some specific parameters on its activation and memory factor. We also test the capabilities of the single unit by evaluating an ensemble of uncoupled neurons in classification tasks. The single neuron memristive nanowire model has a total of six hyper-parameters, $\Delta_t, \eta_p, \eta_d, \kappa_{p_0}, \kappa_{d_0}, \omega$. We pursue an experimental analysis focusing on the two hyper-parameters that stem from our informatics-based modelization, namely Δ_t and ω, and keep the physics-based hyper-parameters to the values in Table 2, derived in a neuromorphic experimental setting [5].

Hyper-parameter	κ_{p_0}	κ_{d_0}	η_p	η_d
Value	0.0001	0.5	10	1

Table 2: Physics-based hyper-parameters used in experiments.

Response to input pulses. We analyze the dynamics of a single neuron model under varying conditions, focusing on the evolution of the activation state $h(t)$ and its response to changes in the discretization step Δ_t (Fig.1 left) and input scaling ω (Fig.1 right). In the left plot, we apply an input pulse from time step 30 to 40. Observe that smaller Δ_t gives slower convergence to the moving fixed point, where the larger transient is the effect of a greater memory factor. Note also that different values of Δ_t give different responses in the activation amplitude. This is because the system is slower in the convergence towards the new fixed point induced by the forcing of the input. A similar behavior can be seen in the right plot, where larger values of ω result in wider amplitude responses and slower convergence rates to the fixed point. As expected, the prolonged tails in the response curves are in line with the theoretical analysis in the previous section, proving that the two hyper-parameters Δ_t and ω play a crucial role in the dynamical properties of the model.

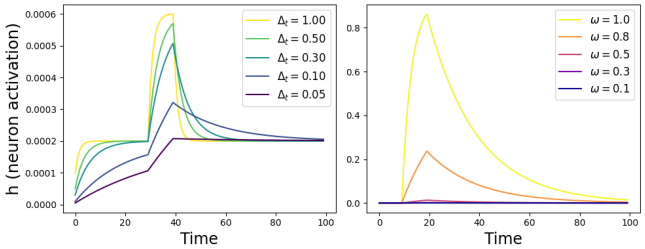

Fig. 1: Neuron activation h over time for different discretization step sizes Δ_t (left, with $\omega = 0.1$ between timesteps 30 and 40, $\omega = 0$ otherwise) and input pulse amplitudes ω (right, with $\Delta_t = 0.1$).

Memory Factor. The memory factor ρ is a function of all hyper-parameters $(\Delta_t, \eta_p, \eta_d, \kappa_{p_0}, \kappa_{d_0}, \omega)$. Fig. 2 illustrates how ρ varies with different discretization steps Δ_t and input scaling values ω. The results show that for small Δ_t (approximately $\Delta_t < 0.1$), the ω becomes less influential on the variation of ρ. In contrast, for relatively large values of Δ_t, a specific range of ω is necessary to

maintain $\rho \approx 1$, ensuring longer short-term memory and better performance in temporal tasks requiring greater memory retention.

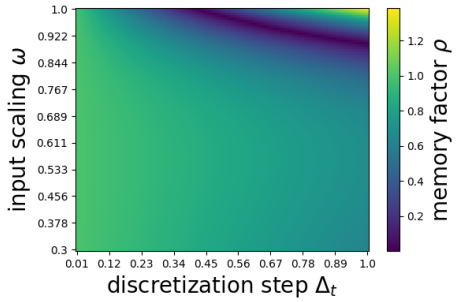

Fig. 2: Memory factor ρ as a function of the discretization step Δ_t and input scaling ω. The heatmap illustrates how ρ varies with different combinations of time step size and input scaling, with ρ values represented by the color intensity.

Time-series classification with neuronal assemblies. We construct an assembly of uncoupled nanowire neurons and evaluate its generalization capabilities in time-series classification tasks. These experiments compare the performance of nanowire neuron assemblies with conventional uncoupled leaky integrator neurons with $tanh(\cdot)$ activation function [6], using an RC setup [7]. In this framework, training is restricted to the output layer, implemented as a ridge classifier. For assemblies of uncoupled neurons, we use input scaling parameters to introduce heterogeneity in the neurons' responses. Specifically, the effective input scaling for each neuron is defined as $\omega = \omega_0 \pm \Delta\omega$, where ω_0 is the center of the interval and $\Delta\omega$ introduces variability across the neurons. For the *tanh* case, we set $\omega_0 = 0$. This approach ensures diverse dynamics within the assembly, potentially enhancing the overall classification performance. Input preprocessing and scaling were guided by insights from heatmap analysis (Fig. 2), and an additional check on $\rho < 1$ was enforced to ensure system stability at a preprocessing level. The hyper-parameters explored during model selection were determined by grid search and are summarized in Table 3, for reservoir layer sizes of 100, 500, and 1000.

Neurons	**Hyper-parameters**	**Values**
Nanowire	`input scaling` ω_0	{0.1, 0.5, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.5, 2}
	`input scaling variability` $\Delta\omega$	{0.001, 0.01, 0.1, 0.5, 1, 1.5, 2}
	`discretization time` Δ_t	{0.01, 0.1, 0.15, 0.2, 0.25, 0.5, 1}
Tanh	`leaking-rate` α	{0.1, 0.3, 0.5}
	`input scaling variability` $\Delta\omega$	{0.1, 0.5, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.5, 2}
	`bias scaling` β	{0.001, 0.1, 1}
	`spectral radius` ρ	{0.8, 0.9, 0.95, 0.99}

Table 3: Values of the hyper-parameters explored during model selection for the different types of uncoplued neurons used in the experiments. For *tanh*-based reservoirs, we explored conventional Echo State Network hyper-parameters (see [7] for details).

Fig. 3 shows the accuracy of uncoupled nanowire neurons (blue) and un-

coupled *tanh* neurons (orange) on the ECG5000 and SyntheticControl datasets[1] across different reservoir layer sizes. Notably, nanowire neurons demonstrate higher or comparable accuracy for both tasks, suggesting their enhanced expressivity, as well as their good potential for these kinds of application.

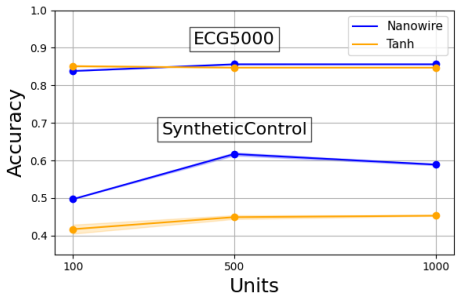

Fig. 3: Accuracy comparison between different numbers of uncoupled nanowire neurons (blue) and leaky integrator neurons based on *tanh* activation (orange) on the ECG5000 and SyntheticControl classification tasks. For each reservoir layer size, namely of 100, 500, and 1000, the model selection has been performed by grid search as detailed in Table 3.

5 Conclusions

We have introduced a novel discrete-time model for dynamically driven neural processing units, inspired by the physical principles of memristive nanowire hardware. Through an analysis framed within the Reservoir Computing paradigm, we demonstrated the model's stability, fading memory, and adaptive response capabilities. Preliminary experiments on time-series classification tasks validated the potential of the introduced nanowire-inspired neurons, achieving competitive performance when compared to Echo State Network-like systems using traditional tanh activation function.

Future research will focus on coupling strategies for the proposed neuron model in densely connected networks, aiming to unlock its full potential for broader applications in both reservoir-based and fully trainable systems.

References

[1] E. Miranda, G. Milano, and C. Ricciardi. Modeling of short-term synaptic plasticity effects in zno nanowire-based memristors using a potentiation-depression rate balance equation. *IEEE Transactions on Nanotechnology*, 19:609–612, 2020.

[2] V. K. Sangwan and M. C Hersam. Neuromorphic nanoelectronic materials. *Nature nanotechnology*, 15(7):517–528, 2020.

[3] K. Nakajima and I. Fischer. *Reservoir computing*. Springer, 2021.

[4] A. Ceni, P. Ashwin, L. Livi, and C. Postlethwaite. The echo index and multistability in input-driven recurrent neural networks. *Physica D: Nonlinear Phenomena*, 412:132609, 2020.

[5] G. Milano, E. Miranda, and C. Ricciardi. Connectome of memristive nanowire networks through graph theory. *Neural Networks*, 150:137–148, 2022.

[6] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3):335–352, 2007.

[7] M. Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer science review*, 3(3):127–149, 2009.

[1] From the UCR archive at `http://timeseriesclassification.com/`