Critical Neural Cellular Automata

Sidney Pontes-Filho[1], Stefano Nichele[2,3] and Mikkel Lepperød[1]

[1]Department of Numerical Analysis and Scientific Computing, Simula Research Laboratory, Norway
[2]Department of Computer Science and Communication, Østfold University College, Norway
[3]Department of Computer Science, Oslo Metropolitan University, Norway
sidney@simula.no

Criticality is a behavioral state in dynamical systems that is known to present the highest computation capabilities, i.e., information transmission, storage, and processing (Langton, 1990). A system capable of this behavior may contain one or more controlling parameters that change the system's phases, such as temperature and pressure. Therefore, such parameters can be tuned to achieve the optimal behavior for computation. This behavior is observed when the system is near a phase transition. Another type of critical behavior is achieved when the same activity patterns spread in space and time across different scales, similar to a fractal in the space dimension. This is identified by a power-law distribution in the system's activity. Most of the critical systems exhibit these two types of behavior. However, a control parameter is not commonly observed and the system stays in criticality independently of the initialization or tuning, such that the critical state is an attractor to the system. A system with such characteristics displays what is called self-organized criticality (SOC) (Bak et al., 1987, 1988; Pontes-Filho et al., 2022). Natural events of this criticality were detected in some distributed dynamical systems, notably in neuronal avalanches in the cortex. Moreover, SOC is hypothesized to support intelligence in the human brain (Fontenele et al., 2019; Heiney et al., 2021). This is often referred to as the critical brain hypothesis.

Our goal is to optimize a simple and deterministic dynamical system toward criticality, such that it may be applied in a paradigm in the artificial intelligence (AI) field, known as reservoir computing (RC) (Schrauwen et al., 2007). This paradigm utilizes a dynamical system as a reservoir to perform computation; then, a linear machine learning model is trained to interpret the states of the reservoir perturbed by the input data. Our choice of optimization method is evolutionary computation, and the dynamical system to be evolved is a deterministic one-dimensional neural cellular automaton with periodic boundaries. Recently critical stochastic 1D cellular automaton (CA) was evolved by Ref. (Pontes-Filho et al., 2020). However, this CA was not effective in RC because of its stochasticity. Therefore, the present work aims to evolve a deterministic CA. A CA system consists of discrete computing units or cells regularly distributed in a grid, commonly with one or two dimensions. Those cells often have binary states that change in discrete time following a transition rule. Since the selected system is a neural cellular automaton (NCA), the transition rule is an artificial neural network (Mordvintsev et al., 2020; Nichele et al., 2017).

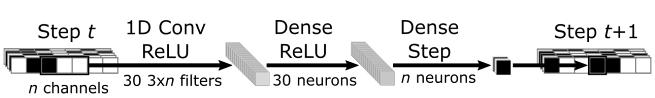


Figure 1: Architecture of the unidimensional neural cellular automata. Convolutional layer is 1D.

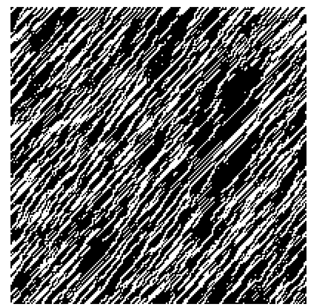
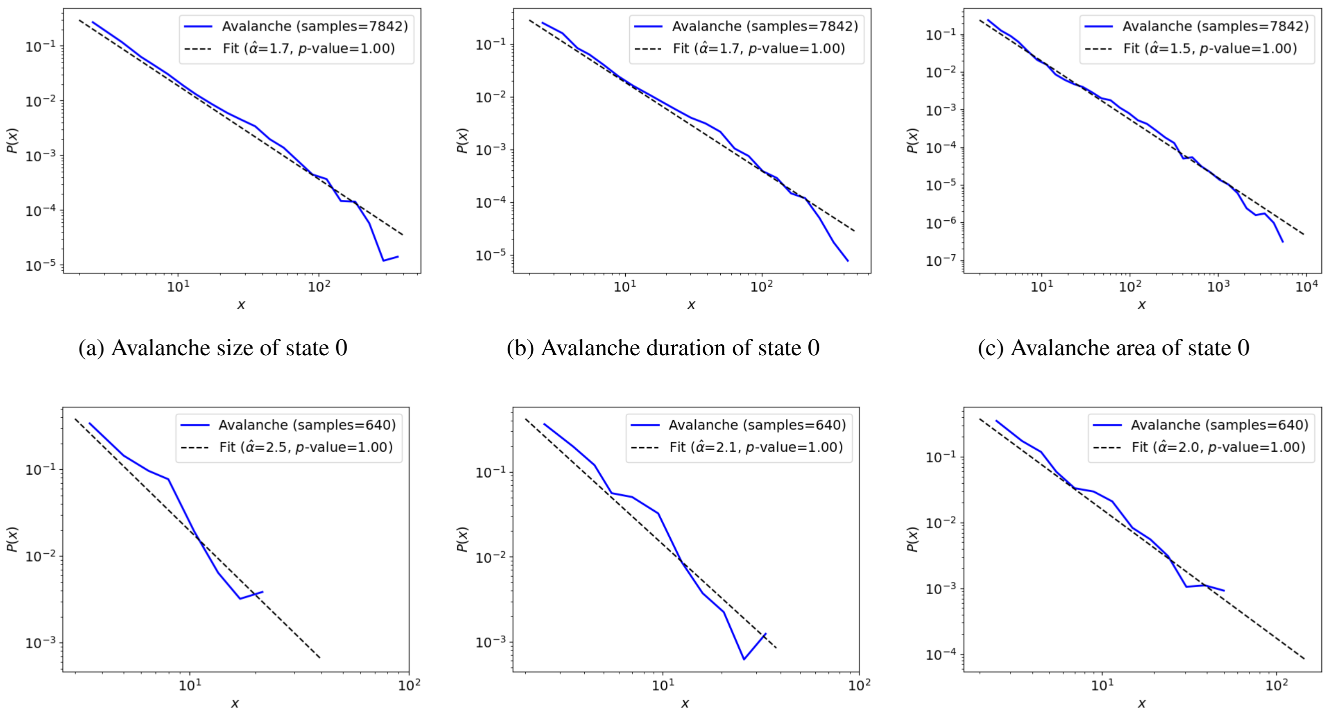


Figure 2: Sample of the selected critical NCA with 200 cells (horizontal axis) randomly initialized and ran through 200 time-steps (vertical axis from top to bottom). White cells are state 0, and black cells are state 1.

The proposed 1D NCA contains 1,000 binary cells that have four extra binary states serving as communication channels. Its architecture consists of one 1D convolution layer with 30 kernels of size 3 (neighborhood) and rectified linear unit (ReLU) as activation function, a dense layer with 30 neurons with ReLU, ending with a dense layer with 5 binary neurons (step activation function) to define the center cell state and its four extra states. The architecture uses $n = 5$ and is illustrated in Fig. 1. The weights and biases of this network were evolved with covariance matrix adaptation evolution strategy (CMA-ES) (Hansen and Ostermeier, 1996). The fitness function used to guide the evolution of



(a) Avalanche size of state 0

(b) Avalanche duration of state 0

(c) Avalanche area of state 0

(d) Avalanche size of state 1

(e) Avalanche duration of state 1

(f) Avalanche area of state 1

Figure 3: Example of the avalanche distributions of the selected NCA. The estimated power-law slope $\hat{\alpha}$ and goodness-of-fit p-value are included in the legends.

the NCA is from Refs. (Pontes-Filho et al., 2020, 2022). It takes the distributions of the six avalanche types, which are size, duration, and area of avalanches for states 0 and 1 (see Ref. (Pontes-Filho et al., 2022) for the avalanche definitions) and calculates the similarities with their power-law distributions estimated with a linear fitting method called least squares regression applied in the 10 leftmost points of the distribution. The similarity measurements are the coefficient of determination of complete linear fitting and the Kolmogorov-Smirnov statistic. Additional fitness scores are the percentage of non-zero bins in the distribution and the percentage of unique states during the simulation. Moreover, small adjustments in the fitness function were made when getting the maximum or minimum feature values of the six avalanche distributions. Now the maximum or minimum values are selected from the average of the three avalanche measurements for the two cell states. For calculating the fitness score, a simulation randomly initialized and executed for 1,000 time-steps is performed.

The evolution is executed for 200 generations with 96 individuals. Several evolutionary runs were performed and one high-performing evolved NCA was selected as a result to be shown in this work, which is depicted in Fig. 2. The selection criterion for the optimizations is the goodness-of-fit p-value based on the Kolmogorov-Smirnov statistic of the six avalanches (Clauset et al., 2009). The avalanche distributions of the selected NCA are presented in Fig. 3. All the distributions have p-value equal to 1 which confirms they are all power-law distributions and the NCA is in the criticality regime.

In conclusion, this method is the first one (to the best of our knowledge) that presents an evolutionary algorithm guiding an NCA toward criticality. While our method was successful, it required several evolutionary runs to achieve a satisfactory level for our criteria of goodness-of-fit. Our plan for ongoing work is to apply the evolved critical NCA in reservoir computing benchmarks. The reason for that is that AI systems are rather narrow and rigid, and an alternative to improve adaptation is through distributed dynamical systems optimized to proper computation regimes, such as the one in this work and applied in an RC framework.

References

Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the 1/f noise. *Physical review letters*, 59(4):381.

Bak, P., Tang, C., and Wiesenfeld, K. (1988). Self-organized criticality. *Physical review A*, 38(1):364.

Clauset, A., Shalizi, C. R., and Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

Fontenele, A. J., de Vasconcelos, N. A., Feliciano, T., Aguiar, L. A., Soares-Cunha, C., Coimbra, B., Dalla Porta, L., Ribeiro, S., Rodrigues, A. J., Sousa, N., et al. (2019). Criticality between cortical states. *Physical review letters*, 122(20):208101.

Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, pages 312–317. IEEE.

Heiney, K., Huse Ramstad, O., Fiskum, V., Christiansen, N., Sandvig, A., Nichele, S., and Sandvig, I. (2021). Criticality, connectivity, and neural disorder: a multifaceted approach to neural computation. *Frontiers in computational neuroscience*, 15:611183.

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: nonlinear phenomena*, 42(1-3):12–37.

Mordvintsev, A., Randazzo, E., Niklasson, E., and Levin, M. (2020). Growing neural cellular automata. *Distill*, 5(2):e23.

Nichele, S., Ose, M. B., Risi, S., and Tufte, G. (2017). Ca-neat: evolved compositional pattern producing networks for cellular automata morphogenesis and replication. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):687–700.

Pontes-Filho, S., Lind, P., Yazidi, A., Zhang, J., Hammer, H., Mello, G. B., Sandvig, I., Tufte, G., and Nichele, S. (2020). A neuro-inspired general framework for the evolution of stochastic dynamical systems: Cellular automata, random boolean networks and echo state networks towards criticality. *Cognitive Neurodynamics*, 14(5):657–674.

Pontes-Filho, S., Lind, P. G., and Nichele, S. (2022). Assessing the robustness of critical behavior in stochastic cellular automata. *Physica D: Nonlinear Phenomena*, 441:133507.

Schrauwen, B., Verstraeten, D., and Van Campenhout, J. (2007). An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007*, pages 471–482.