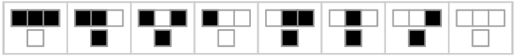
Learning Graph Cellular Automata

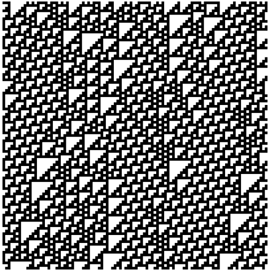
Daniele Grattarola, Lorenzo Livi, Cesare Alippi

Neural Information Processing Systems 2021

Cellular automata

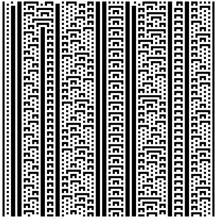


(a) Transition rule

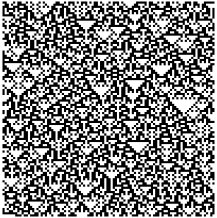


(b) Evolution of CA

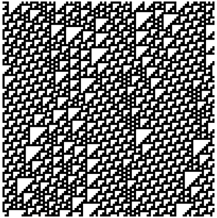
Cellular automata



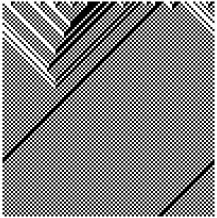
(a) Rule 73



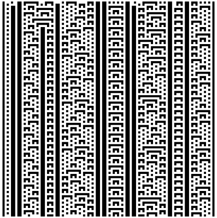
(b) Rule 90



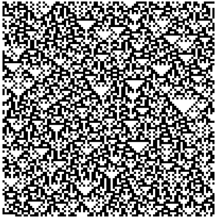
(c) Rule 110



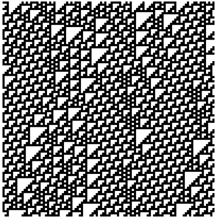
(d) Rule 184



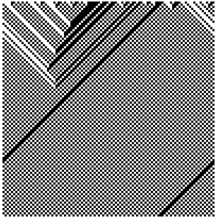
(a) Rule 73



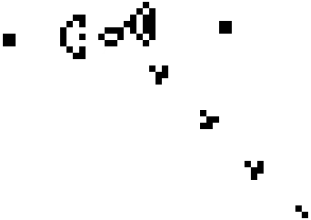
(b) Rule 90



(c) Rule 110



(d) Rule 184

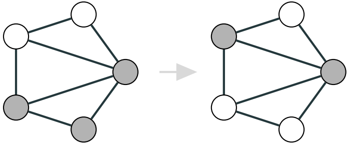


(e) Conway's Game of life

Graph cellular automata

A Graph CA is the generalisation of typical CA:

- Cells arranged in a graph;
- State space is any vector space;
- Transition rule τ as a function of neighbours $\mathcal{N}(i)$;

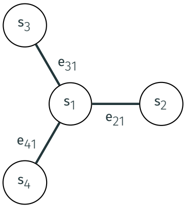


Graph cellular automata

The transition rule has the form:

$$\tau(\mathsf{s}_i) : \{\mathsf{s}_i\} \cup \{\mathsf{s}_j, \mathsf{e}_{ji} \mid j \in \mathcal{N}(i)\} \mapsto \mathsf{s}_i',$$

where e_{ji} encodes type, distance, direction, or unique ID of neighbour.



Learning GCA

Problem: how to design a useful rule?

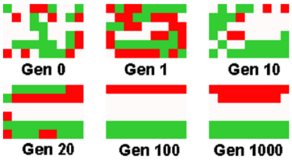
Learning GCA

- **Wulff and Hertz [1]**: learning 1D and 2D CA using "$\sum$-$\prod$ networks with weight sharing";

[1] N Wulff *et al.*, "Learning cellular automaton dynamics with neural networks," 1992.

Learning GCA

- **Wulff and Hertz [1]**: learning 1D and 2D CA using "$\sum$-$\prod$ networks with weight sharing";
- **Elmenreich and Fehérvári [2], Nichele, Ose, Risi, *et al.* [3]**: neuroevolution to learn rules with target behaviour;

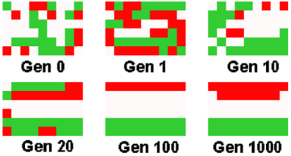


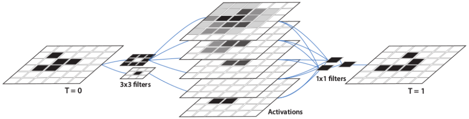
[2] W. Elmenreich *et al.*, "Evolving self-organizing cellular automata based on neural network genotypes," 2011.

[3] S. Nichele *et al.*, "CA-NEAT: evolved compositional pattern producing networks for cellular automata morphogenesis and replication," 2017.

Learning GCA

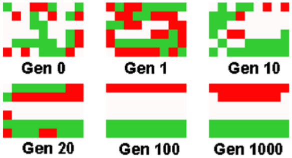
- **Wulff and Hertz [1]**: learning 1D and 2D CA using "$\sum$-$\prod$ networks with weight sharing";
- **Elmenreich and Fehérvári [2], Nichele, Ose, Risi, *et al.* [3]**: neuroevolution to learn rules with target behaviour;
- **Gilpin [4]**: universal CNN architecture for M-state 2D CA;

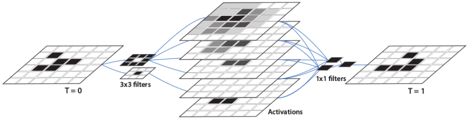




[4] W. Gilpin, "Cellular automata as convolutional neural networks," 2019.

- **Wulff and Hertz [1]**: learning 1D and 2D CA using “∑-∏ networks with weight sharing”;
- **Elmenreich and Fehérvári [2], Nichele, Ose, Risi, *et al.* [3]**: neuroevolution to learn rules with target behaviour;
- **Gilpin [4]**: universal CNN architecture for M-state 2D CA;
- **Mordvintsev, Randazzo, Niklasson, *et al.* [5]**: learning to *grow* a given configuration (inspired by flatworms).



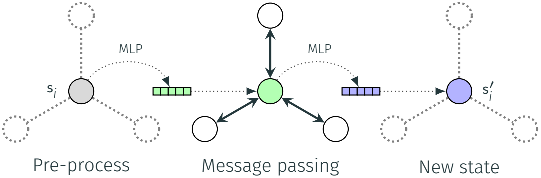


[5] A. Mordvintsev *et al.*, “Growing neural cellular automata,” 2020.

Learning GCA

GCA transition rules are message-passing functions:

$$\mathsf{s}_i' = \gamma\left(\mathsf{s}_i, \sum_{j \in \mathcal{N}(i)} \phi\left(\mathsf{s}_i, \mathsf{s}_j, \mathsf{e}_{ji}\right)\right).$$

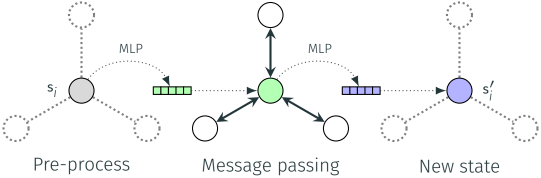


Graph Neural Cellular Automata: GCA with GNN transition rule.

Learning GCA

Extend the results of Gilpin [4] to implement any M-state GCA:

- MLP for one-hot encoding states;
- Message-passing for pattern matching;

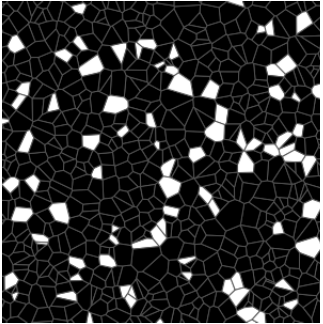


[4] W. Gilpin, “Cellular automata as convolutional neural networks,” 2019.

GNCA on Voronoi tessellation

Binary GCA on Voronoi tessellation (equiv. Delaunay triangulation).

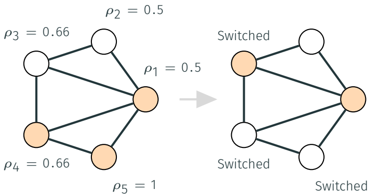
Simplest extension to GCA.



GNCA on Voronoi tessellation

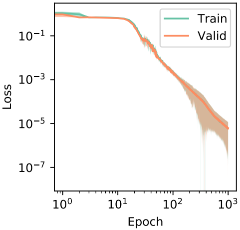
Outer-totalistic rule only depends on the density ρ_i of alive neighbours:

$$\tau(\mathsf{s}_i) = \begin{cases} \mathsf{s}_i, & \text{if } \rho_i \leq \kappa \\ 1 - \mathsf{s}_i, & \text{if } \rho_i > \kappa. \end{cases}$$

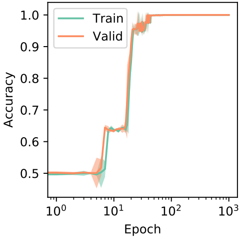


(a) Example transition

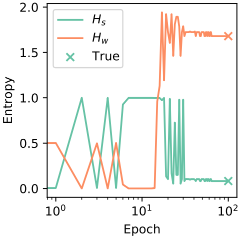
GNCA on Voronoi tessellation



(a) Loss.



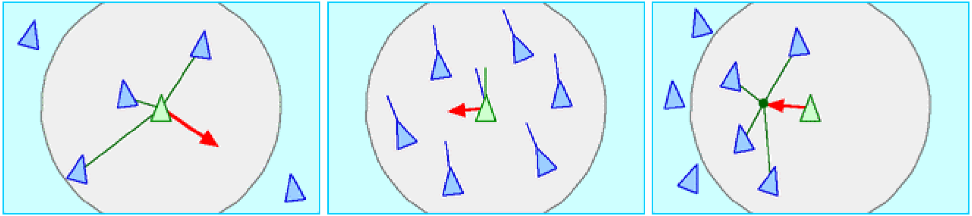
(b) Accuracy.



(c) Entropy.

GNCA for agent-based modelling

Continuous-state GCA with dynamic graph based on the Boids algorithm [6]:

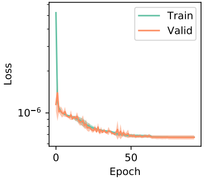


(a) Separation (b) Alignment (c) Cohesion

[6] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," 1987.

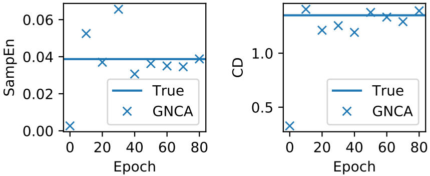
GNCA for agent-based modelling

Loss goes to 10^{-6} almost immediately, but approximation is not good:



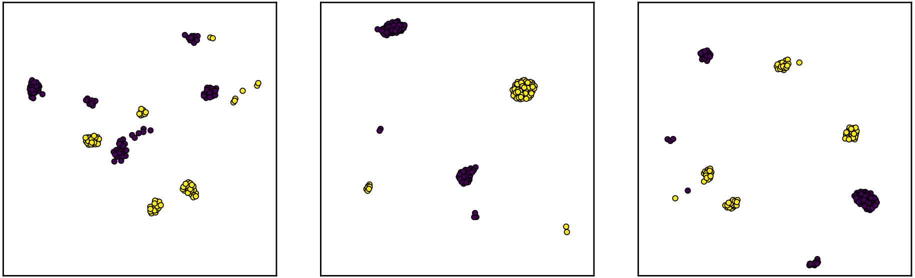
(a) Loss.

Use sample entropy and correlation dimension to evaluate how good the learned rule is.



(a) SampEn and CD during training.

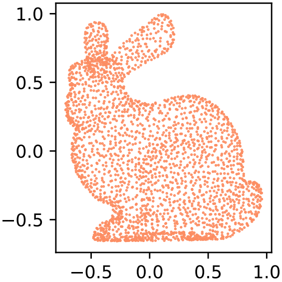
GNCA for agent-based modelling



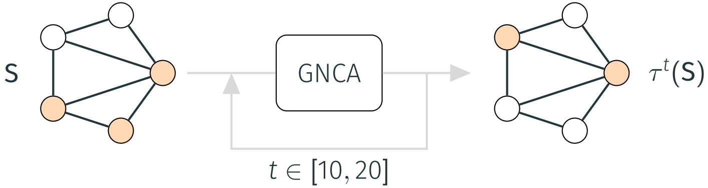
(a) Examples of flocks from the true system and the GNCA.

GNCA that converge to a fixed target

Goal: design a rule with a desired behaviour (converging to target).

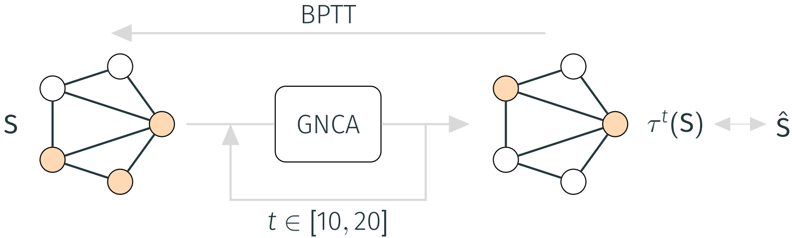


GNCA that converge to a fixed target

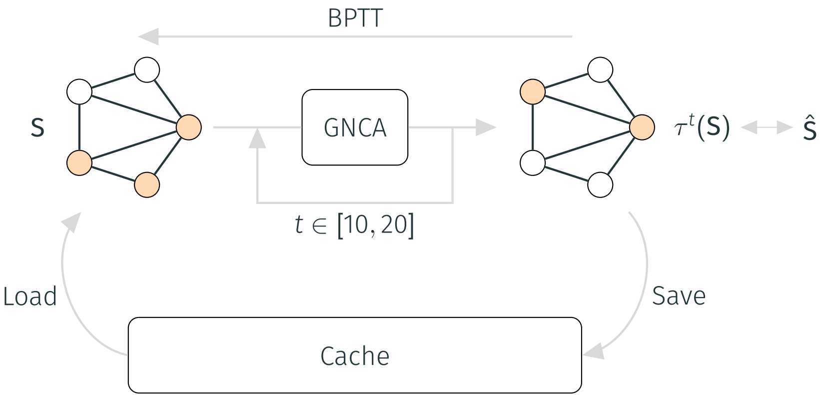


Apply rule for t steps starting from S.

GNCA that converge to a fixed target



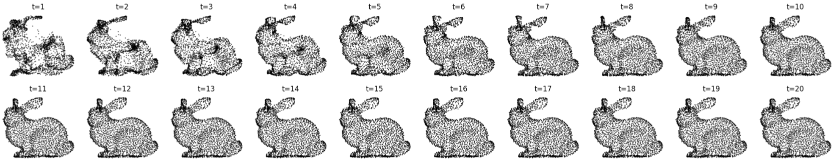
Compute loss w.r.t. target state $\hat{\mathsf{S}}$.

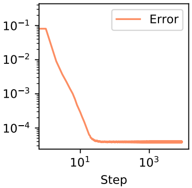


Use a cache to ensure stable attractor and adequate state space exploration [5].

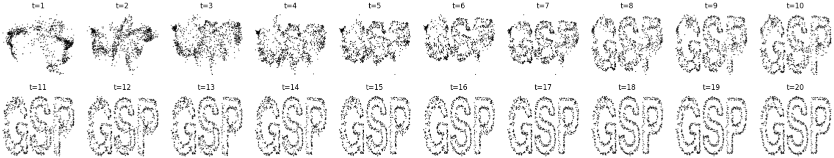
[5] A. Mordvintsev *et al.*, "Growing neural cellular automata," 2020.

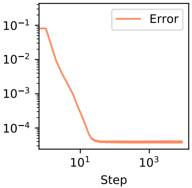
GNCA that converge to a fixed target





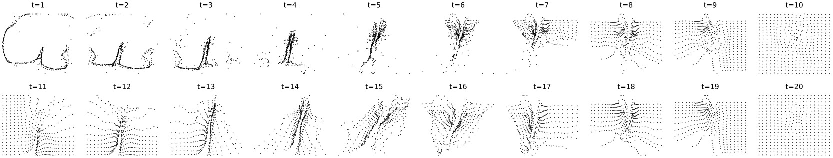
(a) Logo, GNCA trained with $t = 20$, example of convergence.

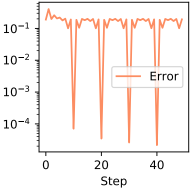




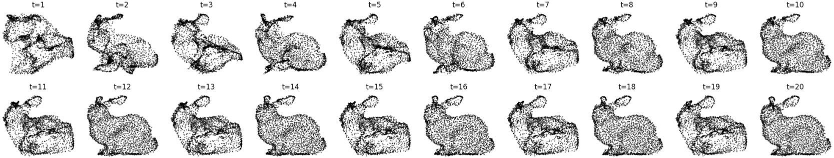
(a) Logo, GNCA trained with $t = 20$, example of convergence.

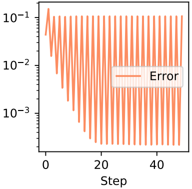
GNCA that converge to a fixed target





(a) Grid, GNCA trained with $t = 10$, example of periodic behaviour.





(a) Bunny, GNCA trained with $t = 20$, example of periodic behaviour.

Future research

- Predict global properties (e.g. graph classification);
  - Something similar was done by Gori, Monfardini, and Scarselli [7];

[7] M. Gori *et al.*, “A new model for learning in graph domains,” 2005.

Future research

- Predict global properties (e.g. graph classification);
    - Something similar was done by Gori, Monfardini, and Scarselli [7];
- Decentralised control (e.g. IoT, autonomous vehicles);

[7] M. Gori *et al.*, "A new model for learning in graph domains," 2005.

Future research

- Predict global properties (e.g. graph classification);
  - Something similar was done by Gori, Monfardini, and Scarselli [7];
- Decentralised control (e.g. IoT, autonomous vehicles);
- Epidemiological networks;

[7] M. Gori *et al.*, “A new model for learning in graph domains,” 2005.

Future research

- Predict global properties (e.g. graph classification);
    - Something similar was done by Gori, Monfardini, and Scarselli [7];
- Decentralised control (e.g. IoT, autonomous vehicles);
- Epidemiological networks;
- Modelling biological systems (e.g. neurons).

[7] M. Gori *et al.*, “A new model for learning in graph domains,” 2005.

Conclusion

Learning Graph Cellular Automata

Daniele Grattarola, Lorenzo Livi, Cesare Alippi

github.com/danielegrattarola/GNCA

Get in touch: daniele.grattarola@usi.ch

Summary:

- Learn GCA rules with GNNs;
- Universal architecture for M-state GCA;
- Enable design of GCA from high-level specification;
- Solve tasks through emergent computation.

References i

[1] N Wulff and J. A. Hertz, "Learning cellular automaton dynamics with neural networks," *Advances in Neural Information Processing Systems*, vol. 5, pp. 631–638, 1992.

[2] W. Elmenreich and I. Fehérvári, "Evolving self-organizing cellular automata based on neural network genotypes," in *International Workshop on Self-Organizing Systems*, Springer, 2011, pp. 16–25.

[3] S. Nichele, M. B. Ose, S. Risi, and G. Tufte, "Ca-neat: Evolved compositional pattern producing networks for cellular automata morphogenesis and replication," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 687–700, 2017.

[4] W. Gilpin, "Cellular automata as convolutional neural networks," *Physical Review E*, vol. 100, no. 3, p. 032 402, 2019.

[5] A. Mordvintsev, E. Randazzo, E. Niklasson, and M. Levin, "Growing neural cellular automata," *Distill*, vol. 5, no. 2, e23, 2020.

[6] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[7] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, IEEE, vol. 2, 2005, pp. 729–734.