State space properties of Boolean networks trained for sequence tasks

Andrea Roli[1], Matteo Amaducci[1], Lorenzo Garattoni[2], Carlo Pinciroli[2], and Mauro Birattari[2⋆]

[1] Dipartimento di Informatica: Scienza e Ingegneria - DISI
Alma Mater Studiorum Università di Bologna
Cesena (Italy)
[2] IRIDIA-CoDE
Université libre de Bruxelles
Brussels (Belgium)

Abstract. In a recent work, it has been shown that Boolean networks (BN), a well-known genetic regulatory network model, can be utilised to control robots. In this work, we use a genetic algorithm to train robots controlled by a BN so as to accomplish a sequence learning task. We analyse the robots' dynamics by studying the corresponding BNs' phase space. Our results show that a phase space structure emerges enabling the robot to have memory of the past and to exploit this piece of information to choose the next action to perform. This finding is in accordance with previous results on minimally cognitive behaviours and shows that the phase space of Boolean networks can be shaped by the learning process in such a way that the robot can accomplish non-trivial tasks requiring the use of memory.

1 Introduction

Dynamical systems provide metaphors and tools which can be effectively used to analyse artificial agents, such as robots [3, 15]. The dynamical systems metaphor has also been advocated as a powerful source of design principles for robotics [8]. The core idea supporting this viewpoint is that information processing can be seen as the evolution in time of a dynamical system [12]. In this paper, we show that a dynamical systems perspective makes it possible to analyse the behaviour of a robot controlled by Boolean networks and explain it in terms of trajectories in the Boolean network's state space.

Boolean networks (BNs) have been introduced by Kauffman [4] as a gene regulatory network (GRN) model. BNs have been proven to reproduce very important phenomena in genetics and they have also received considerable attention in the research communities on complex systems [1, 4]. A BN is a discrete-state and discrete-time dynamical system whose structure is defined by a directed graph of N nodes, each associated to a Boolean variable x_i, $i = 1, \ldots, N$, and a Boolean function $f_i(x_{i_1}, \ldots, x_{i_{K_i}})$, where K_i is the number of inputs of node i. The arguments of the Boolean function f_i are the values of the nodes whose outgoing arcs are connected to node i. The state of the system at time t, $t \in \mathbb{N}$, is defined by the array of the N Boolean variable values at time t: $s(t) \equiv (x_1(t), \ldots, x_N(t))$. The most studied BN models are characterised by having a *synchronous* dynamics—i.e., nodes update their states at the same instant—and *deterministic* functions. However, many variants exist, including asynchronous and probabilistic update rules [13]. BN models' dynamics can be studied by means of usual dynamical systems methods [2, 12], hence the usage of concepts such as state (or phase) space, trajectories, attractors and basins of attraction. BNs can exhibit complex dynamics and some special ensembles have been deeply investigated, such as that of Random BNs [4, 11].

In a recent work, it has been shown that BNs can be utilised to control robots [10]. A BN is coupled with a robot by defining a set of input nodes, whose values are imposed by the robot's sensor readings, and a set of output nodes, which are used to maneuver the robot's actuators. The BN is trained by means of a learning algorithm that manipulates the Boolean functions. The

⋆ Mauro Birattari acknowledges support from the Belgian F.R.S.-FNRS.

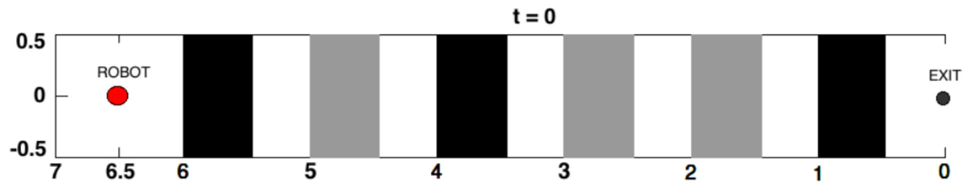

Fig. 1. An example of the BN-robot's working environment. The order of black and grey stripes is randomly chosen at each trial.

algorithm employs as learning feedback a measure of the performance of the BN-controlled robot (in the following, BN-robot) on the task to perform.

In this work, we use a genetic algorithm to train a BN-robot so as to accomplish a task concerning *sequence learning* and we analyse their dynamics by studying the characteristics of the corresponding BNs' state space. Our results show that a state space structure emerges enabling the robot to have memory of the past and to exploit this piece of information to choose the next action to perform. In the following of this brief contribution, we outline the task to accomplish and we illustrate the main results achieved. For completeness, we include a description of materials and methods.

The task

Sequence learning is one of the most prominent activities in humans, animals, as well as artificial agents and systems [14]. Sequence tasks involve the use of some kind of *memory* which enables the agent to choose the next action depending on the past. The main kinds of sequence tasks are: sequence prediction, generation, recognition and sequential decision making. Sequence learning is clearly a difficult task, due to the fact that forms of memory structures are needed. Several techniques exist to tackle the problem, including recurrent neural networks, hidden Markov model, dynamic programming, reinforcement learning and evolutionary computation techniques, such as the ones used in this work.

In our experiment, the BN-robot must learn to recognise a sequence of colours, by performing certain actions. The environment in which the BN-robot operates is a straight corridor. Along the corridor, the ground is painted in three different colours: white (W) represents the background, while black (B) and grey (G) denote the symbols of a sequence to be recognised. See Figure 1 for an example of the environment. The BN-robot, placed at the beginning of the corridor, moves along it, turning its LEDs on when it encounters a black or grey stripe in the right sequence and keeping the LEDs off when the colour is not in the right order or it is the background colour. In our case, the sequence to be recognised is a cyclic repetition of black and grey. For example:

```
Colours along the corridor: W    B    W    G    W    G    W    B    W    B
BN-robot's LEDs status:    OFF  ON   OFF  ON   OFF  OFF  OFF  ON   OFF  OFF
```

This task is dynamic, in that the robot needs to decide whether to switch on or off the LED, on the basis of information concerning the past. To carry out this task, the robot needs to exploit some sort of memory.

Results

A successful BN-robot is one which correctly switches its LEDs on and off according to the desired sequence, when encountering different colours on the ground. Since this task requires some kind of memory structure to be constructed, we analysed the state space traversed by the BN controlling a robot with the aim to understand its operation and dynamics. A similar approach has also been

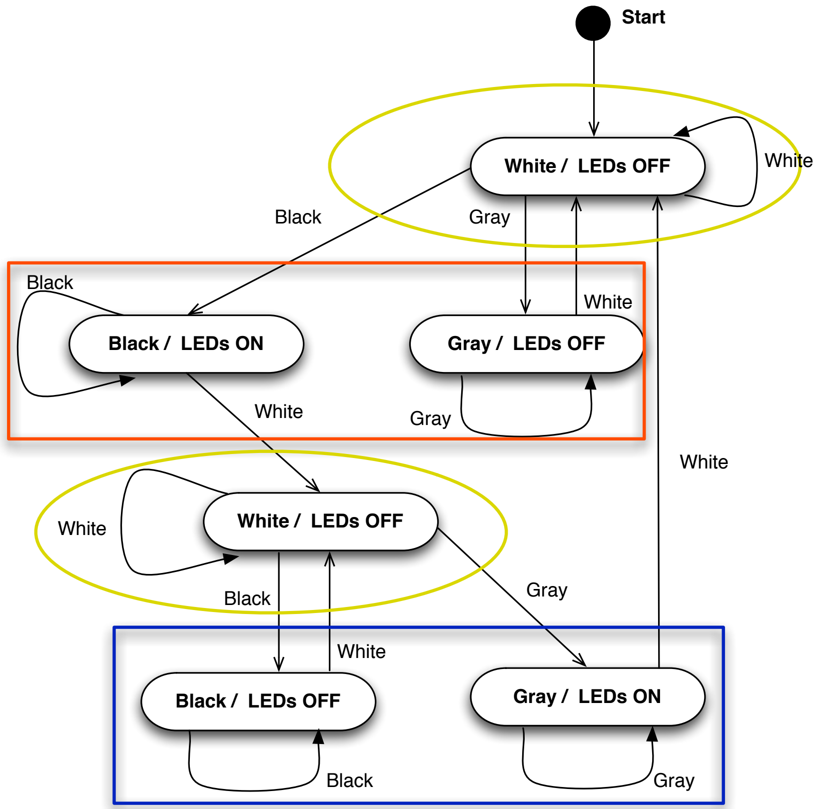

Fig. 2. Finite state automaton representation of the trajectory graph over the state space for a typical BN. A state in the automaton represents a cluster of BN's states in which the BN remains until a specific input is received.

used in previous works in evolutionary robotics [6, 15]. To analyse a BN, we extracted a sample of 1000 trajectories in the state space by simulating the robot in corridors with colours in random order. We gathered such trajectories and generated a graph of the observed state transitions. The first relevant observation we derived from this analysis is that the size of the state space traversed by the BN is a very tiny fraction of the whole potential state space, which is of size 2^{30}. Indeed, the number of states in the collected trajectories is about 200, on average; hence, the learning process shapes the BN in such a way that its dynamics is confined in a limited portion of the state space. A further notable property of the BN dynamics of the robot is that memory is implicitly represented by connecting different areas of the state space, each devoted to a specific set of actions. A compact view of the state space can be provided in the form of a finite state automaton (FSA), in which states represent clusters of connected states in the BN phase space. Indeed, the BN phase space can be clustered in sets of states which encode the memory of the previous colour encountered by the robot. In Figure 2, we report the FSA of a typical successful BN-robot. At the beginning of the trial, the BN is in a state space area in which the values of the output nodes are such that the BN-robot goes straight and keeps its LEDs off until a coloured stripe is found. Then, depending on the detected colour, the BN goes into either of two regions, which we will denote as the upper (red) and lower (blue) rectangles. As we can see, the mechanisms in the two clusters is dual: in the upper one, the robot switches its LEDs on when it encounters a black stripe either if it is the first non-background colour it detects or if a grey stripe has been previously found; conversely, the second state space cluster is devoted to recognising grey stripes, after a black one has been encountered.

In summary, we can assert that the information concerning the last seen colour is implicitly stored in the state space area in which the BN operates. This finding is in accordance with previous results on minimally cognitive behaviours [15] and shows that the state space of BNs can be shaped by the learning process in such a way that the BN-robot can accomplish non-trivial tasks requiring the use of memory.

Materials and methods

In this experiment, we control an *e-puck* robot [5] by means of a BN. The robots are simulated with the the ARGoS simulator [9]. The values of a set of network nodes (BN input nodes) are imposed by the robot's sensor readings, and the values of another set of nodes (BN output nodes) are observed and used to encode the signals for maneuvering the robot's actuators. The BN controlling the robot has 30 nodes in total and the function of each node depends on the value of 3 other nodes, chosen at random. Four nodes are used as inputs and encode the proximity sensors (North, South, East, West)—the node is set to 1 if an obstacle is near the robot—and two encode the ground colours (00↔black, 01↔grey, 11↔white). Two nodes are used to control the robot wheels, which can be individually either set to a constant non-zero speed or stopped.

The BNs controlling the robots are designed by means of a genetic algorithm (GA),[3] according to the evolutionary robotics approach [7]. The genetic algorithm adopts a proportional selection and applies mutation and crossover operators. Mutation is implemented by randomly choosing a node and an entry in its Boolean function truth table and flipping it.[4] The crossover operator is a single point crossover, operating on the binary string given by the concatenation of the the nodes' Boolean functions. The training process changes the Boolean functions, while the BN topology is kept constant (it is generated according to the random BN model, as described by Kauffman [4]). The fitness function is computed as the average of the performance across 10 trials, in which the sequence of colours is randomly generated. The performance of the BN-robot in a trial is computed as the distance it can walk along the corridor by correctly switching its LEDs and avoiding the walls. The GA is run with the following parameter setting: population size equal to 20, elitism set to 2, $p_{\mathrm{mut}} = 0.02$, $p_{\mathrm{cx}} = 0.1$, the number of generations is set to 5000. The GA is run 60 times, starting from randomly generated BNs. The successful runs, i.e., those returning BN-robots correctly performing the sequence task, were 10 out of 60.

Conclusion

In this brief contribution, we have outlined the results of the analysis of the behaviour of BN-robots trained to accomplish a sequential task. The behaviour of a BN-robot is studied by means of the phase space analysis of the corresponding BN. The results show that the training process shapes the phase space so as to restrict the BN dynamics to few, relatively small, areas. Phase space areas play the role of memory, as they implicitly store the information concerning the past which is relevant for the BN-robot to choose the next action. In addition, we observe that the analysis of the BN phase space can be simplified by clustering the set of states and studying the corresponding finite state automata. This method may also be subject to formal verification, making it possible to validate the robot's behaviour.

References

1. Aldana, M., Coppersmith, S., Kadanoff, L.: Boolean dynamics with random couplings. In: Kaplan, E., Marsden, J., Sreenivasan, K. (eds.) Perspectives and Problems in Nonlinear Science. A celebratory volume in honor of Lawrence Sirovich. Springer Applied Mathematical Sciences Series, Springer, Berlin, Germany (2003)

---

[3] Other search techniques have also been used and we obtained the same qualitative results.
[4] Details can be found in [10].

2. Bar–Yam, Y.: Dynamics of Complex Systems. Studies in nonlinearity, Addison–Wesley, Reading, MA (1997)
3. Beer, R.: A dynamical systems perspective on agent-environment interaction. Artificial Intelligence 72, 173–215 (1995)
4. Kauffman, S.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, UK (1993)
5. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: Gonçalves, P., Torres, P., Alves, C. (eds.) Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions. vol. 1, pp. 59–65 (2009)
6. Nolfi, S.: Categories formation in self-organizing embodied agents. In: Cohen, H., Lefebvre, C. (eds.) Handbook of Categorization in Cognitive Science, pp. 869–889. Elsevier (2005)
7. Nolfi, S., Floreano, D.: Evolutionary robotics. The MIT Press, Cambridge, MA (2000)
8. Pfeifer, R., Bongard, J.: How the Body Shapes the Way We Think: A New View of Intelligence. MIT Press, Cambridge, MA (2006)
9. Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Stirling, T., Gutiérrez, A., Gambardella, L., Dorigo, M.: ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), pp. 5027–5034. IEEE Computer Society Press, Los Alamitos, CA (September 2011)
10. Roli, A., Manfroni, M., Pinciroli, C., Birattari, M.: On the design of Boolean network robots. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcázar, A., Merelo, J., Neri, F., Preuss, M., Richter, H., Togelius, J., Yannakakis, G. (eds.) Applications of Evolutionary Computation, Lecture Notes in Computer Science, vol. 6624, pp. 43–52. Springer, Heidelberg, Germany (2011)
11. Serra, R., Villani, M., Graudenzi, A., Kauffman, S.: Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. Journal of Theoretical Biology 246, 449–460 (2007)
12. Serra, R., Zanarini, G.: Complex Systems and Cognitive Processes. Springer, Berlin, Germany (1990)
13. Shmulevich, I., Dougherty, E.: Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks. SIAM, Philadelphia, PA (2009)
14. Sun, R., Giles, C.: Sequence learning: From recognition and prediction to sequential decision making. IEEE Intelligent Systems 16(4), 67–70 (2001)
15. Yamauchi, B., Beer, R.: Sequential behavior and learning in evolved dynamical neural networks. Adaptive Behavior 2(3), 219–246 (1994)