An investigation of graceful degradation in Boolean network robots subject to online adaptation

Michele Braccini, Paolo Baldini, Andrea Roli
m.braccini@unibo.it p.baldini@unibo.it andrea.roli@unibo.it

University of Bologna

September 7, 2023

An investigation of graceful degradation in Boolean network robots subject to online adaptation

Michele Braccini, Paolo Baldini, Andrea Roli
m.braccini@unibo.it p.baldini@unibo.it andrea.roli@unibo.it

University of Bologna

September 7, 2023

Good morning, my name is Paolo Baldini, and I'm here to present the study that I conducted with Andrea Roli and Michele Braccini about the graceful degradation of performance during the online adaptation of robots controlled by Boolean networks.

Introduction

Redundancy & Degeneracy
⇓
Maintain the system alive

Introduction
Redundancy & Degeneracy
⇓
Maintain the system alive

The human body exploits a huge amount cues to function. Many are redundant while others are degenerate, meaning that the different information that they produce can be used to perform the same goal. This allows to keep the system essential variables into the living ranges also if some signals disappears. Nevertheless, some cues are much more impactful than others, and their vanishing requires an adaptation in order to use the remaining. This process often leads to an instantaneous degradation in the capabilities, that can however recover with time. Often, the recovery is partial, and the performance never returns to the origin. If the achievable performance decreases gently with the number of missing cues, we say that it is degrading gracefully.

Goal

Adaptation
⇓
Graceful degradation

Goal
Adaptation
⇓
Graceful degradation

Although desirable, a graceful degradation is often hard to achieve. Most robots are statically programmed or trained, making them sensible to lack of information or very dependent on few of them. Our goal is to assess if providing the robot with the ability to adapt induces a graceful degradation of performance. For this goal, we set up an experiment to assess how the performance are correlated to the number of faulty sensors and to the type of damage.

Experiment: setup

- Phototaxis task
- Foot-bot (24 light sensors)
- Boolean network controller
- $100 \times 100m$ arena
- 1200 epochs
- 100 replicas

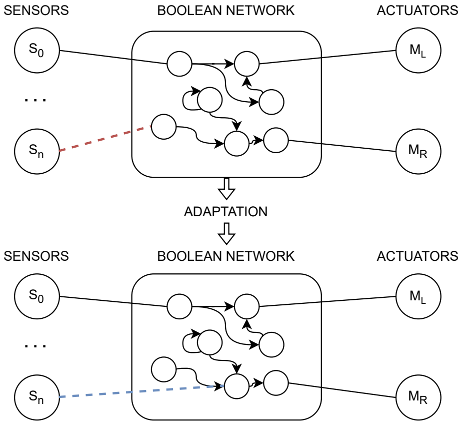
Experiment: setup

- Phototaxis task
- Foot-bot (24 light sensors)
- Boolean network controller
- $100 \times 100m$ arena
- 1200 epochs
- 100 replicas

The experiment consists in a Phototaxis task, where a robot is required to reach a light source. The model of robot used in the experiment is a "foot-bot", which is equipped with 24 light sensors. It is driven by a Boolean network controller, which takes as input the signal from the sensors and produces as output the control signal for the actuators. Since the controller works with binary inputs, the value from the sensors is binarized according to a threshold (0.2 out of 1).

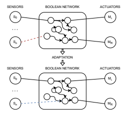
The task takes place in an arena with a 100 meters edge. The light source is placed in the center and the robot in a corner. The robot has 1 minute to get as near as possible to the light. This timespan is called an epoch. The simulation consists in 1200 epochs. In order to be statistically significant, the experiment is repeated 100 times.

Experiment: adaptation



2024-10-08

Experiment: adaptation



When a new epoch starts, the best controller discovered is adapted. The adaptation consists in the reconnection of a subset of sensors to different points of the Boolean network. The idea is that we can discover how to perturb the network such to generate a desired output. The sensors to reconnect are chosen randomly. If the new configuration performs better or equal than the best known one, this is set as the new best.

Experiment: type of damage

- No output
- Random output
- Fixed output
- Damaged area

Experiment: type of damage
- No output
- Random output
- Fixed output
- Damaged area

During the experiment we tested different kinds of sensors rupture in order to simulate different situations. The first is the "no output", where we simulate a sensor that simply detaches or stops to send signals without interfering with the controller. Then we simulated a crazy sensor providing random signals. The "fixed output" damage considers instead a sensor blocking in a specific state. Since our controller is a Boolean network, this means forcing a node value to 0 or 1. Until now, the experiments involved the rupture of random sensors; in the "damaged area" instead we consider the rupture of contiguous sensors in a randomly chosen section of the robot. This simulates externally caused damages, like passing near to a strong heat source.

Results

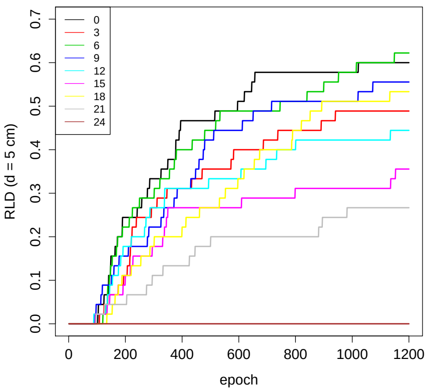
- Run Length Distribution
- Average distance along epochs

Results
- Run Length Distribution
- Average distance along epochs

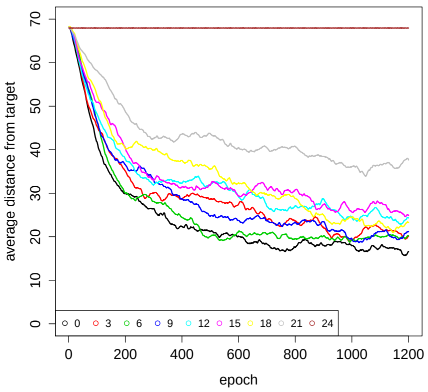
We analyzed the results according to two different metrics. The Run Length Distribution (RLD) displays the number of robots that obtained a performance higher than a given threshold in any moment before a given epoch. The average distance from the target at each epoch shows how near the robots are to the target. For each type of damage, this is averaged over all the 100 robots.

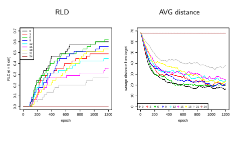
Results: no output

RLD



AVG distance

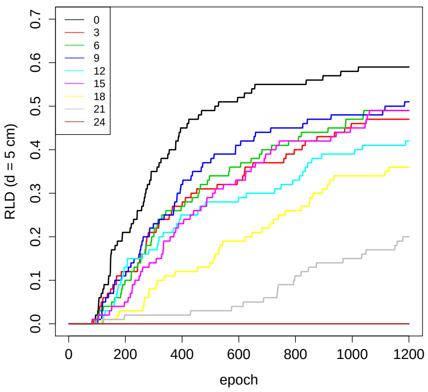




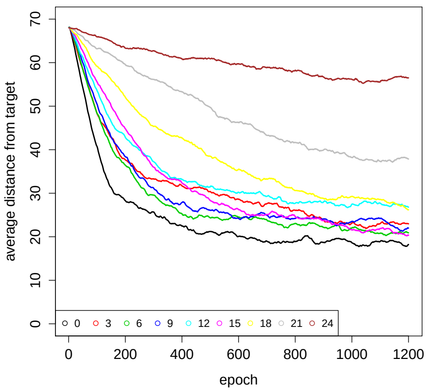
The first results to analyze are the ones of the "random input". Those clearly show that the amount of damage directly impacts the performance. Indeed, we can see that increasing the amount of damage reduces the performance. However, we can also see that the performance degrades somehow gracefully, without showing any abrupt variation. Indeed, those only happen when the number of fault sensors is more than 80%, i.e., the case were we have 21 and 24 faulty sensors.

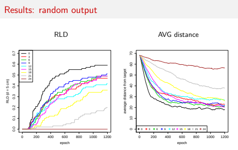
Results: random output

RLD



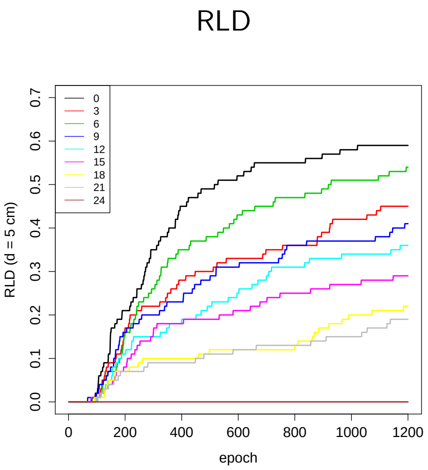
AVG distance

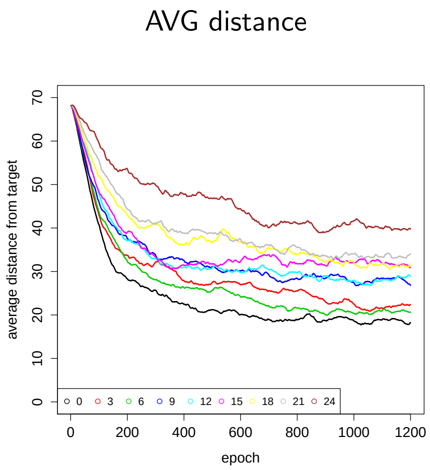




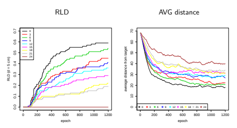
For the “random output” instead we notice a slight difference. Although similar, these results show that a random signal reduces the amount of robots succeeding in crossing the RLD threshold. Conversely, we also see that this allows also the 24 faulted sensors robot to move. This suggests that adapting to a crazy sensor is particularly difficult, but on the other hand allows to maintain a basic behavior.

Results: fixed output





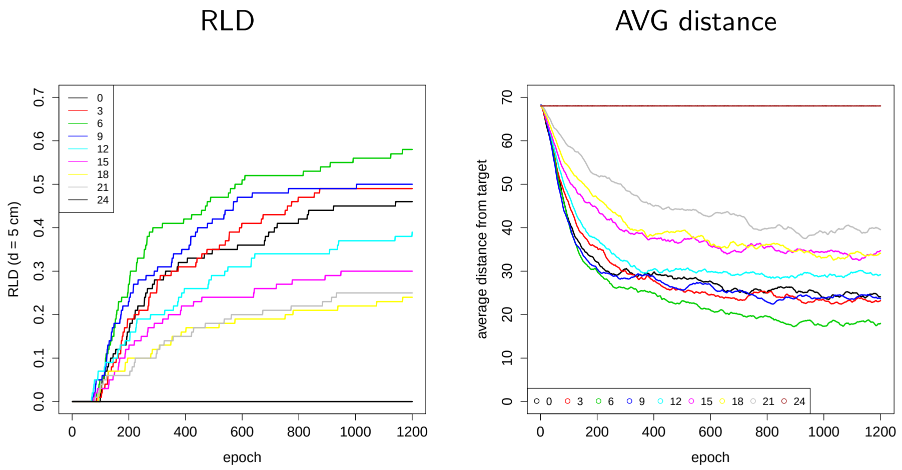
2024-10-08

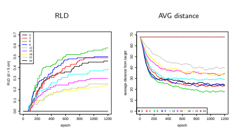


The "fixed output" results are even more interesting. Here, the performance decreases in a much more noticeable way. However, also the 24 faulted sensors robot succeed in going nearer to the light.

Our hypothesis is that the adaptation is not able to segregate those sensors to avoid their erroneous influence, generally leading to higher decreases in performance with the increase of faulted sensors. On the other side, it seems to be able to exploit them to adapt the control to a specific instance of the task. This means that the adaptation succeeds in understanding which endogenous signal can be used to correctly drive the robot. Therefore, we can say that this is not a behavioral adaptation, but an adaptation to a specific instance of the task.

Results: damaged area





Finally, the “damaged area” shows that the fault of a subsection of the robot sensors allows an increase in performance. This was honestly unexpected, and we hypothesis that it may be related to an easier adaptation. The idea is that a lower amount of sensors may be reconnected more easily in an effective way. Moreover, this may be related to the importance of the sensors in achieving the task. Indeed, not all the sensors have the same impact. We can consider the robot as divided in 4 main sections: the front one (i.e., the most important), the side ones (i.e., less important), and the rear one (i.e., almost negligible). Since the damaged area affects contiguous sensors, we can expect that this kind of damage will often leave at least one of the most important sections unaffected. This would therefore simplify the adaptation to the few remaining sensors, possibly allowing to overrun the not-faulted robot.

Conclusions

Considerations:

- Adaptation provides graceful degradation;
- Not all the sensors are needed, and less can be better.

Work in progress:

- Recover time after fault;
- Comparison with an offline trained robot.

Conclusions

Considerations:

- Adaptation provides graceful degradation;
- Not all the sensors are needed, and less can be better.

Work in progress:

- Recover time after fault;
- Comparison with an offline trained robot.

From these results we can draw some considerations. First, an online adaptation seems to allow graceful degradation. Indeed, although a static controller can still provide good performance after a fault, its is very unlikely that they will be the best obtainable. An online adaptation allows instead to optimize also suboptimal situations where only few weak sensors remains available. Second, not all the sensors are really needed for achieving a task. Instead, reducing the number of sensors may even increase the performance. This is visible especially in the damaged area task, where a smaller amount of sensors allowed to obtain better performance.

Finally, we are currently working on few other tasks. We aim to verify the time needed to the system to recover from a fault. This is important in order to guarantee a fast recovery from damage, such that the robot does not incur in worse breakdowns due to dangerous behaviors. Additionally, we want to assess how much the adaptation improves the performance compared to a static controller. To do so, we want to simulate the same type of rupture in an offline trained robot. We expect the performance of the latter to decrease abruptly, and obviously without the possibility to improve.