Neural Networks Learning: Heuristic Methods and Applications

T. Carletti (Director) and D. Nicolay (PhD student)
University of Namur, Belgium

Abstract. This paper is a report of the state of progress, after one year, of our research in the framework of my PhD thesis. This research is made in collaboration with Andrea Roli from the University of Bologna. The aim of this short paper is introduce our main goal and the mathematical tools that we plan to use in order to reach it. We also explain the current trails that we follow and the ideas that we have for future research.

1 Introduction

A large number of biological networks have modular structures resulting from a Darwinian evolution on a varying environment. Our research question is to study the emergence of such structures. To get our goal we chosen the framework of artificial neural networks trained to achieve peculiar competing tasks, using genetic algorithms. Our preliminary analysis is mainly concentrated on the study of some characteristics such as redundancy\(^1\), degeneracy\(^2\) [10] and modularity\(^3\) [3]. We look for the identification of networks behaviours and patterns, in order to unravel the mechanisms responsible for the learning process.

With the aim of studying the emergence of structures following the learning phase, we investigate different aspects of our networks. We study the connections, i.e. synapses linking pieces of network responsible to solve the imposed tasks, to conclude if they are achieved independently or collectively. We also compare structures got by modular and global learning, i.e. by learning tasks separately or in parallel. In addition, the networks are analyzed to detect potential redundant connections and foresee what could happen in case of partial destruction of the networks resilience. We study these networks features both on abstract tasks and on applied ones, such as robotics.

2 Mathematical Tools

This research project is developed around two mathematical tools: artificial neural networks and genetic algorithms. As far as we know, although these tools

\(^1\) Invariable modelling of a function by the same element
\(^2\) Capacity for structurally different elements to model the same function
\(^3\) Quality measure of a sharing out in modules, each of them made of a set of nodes strongly linked together and slightly linked with nodes from other modules
Artificial neural networks have been used abundantly in the literature, they haven’t yet been used together to solve the kind of problems we are interested in.

2.1 Artificial Neural Networks

Artificial neural networks are calculation methods inspired by biological neural networks [8], [9]. They are made of inputs, outputs and hidden neurons linked together by synaptic connections. They receive information from input nodes and give answers through output nodes according to their connections and their weights. Figure 1 represents the model of neural networks that we exploit.

In this model, the topology of the network is completely unconstrained, except for the self-loops that are prohibited. The state of each neuron is activated (equal to 1) or inhibited (equal to 0). The weight of each link is 1 (the link is excitatory) or -1 (inhibitor). The updated state of each neuron is given by the perceptron rule:

\[
\text{State} = \begin{cases} 
1 & \text{if } \sum_{i=1}^{k_{in}} w_i x_i \geq \theta k_{in} \\
0 & \text{otherwise}
\end{cases}
\]

where \( k_{in} \) is the number of incoming links and \( \theta \) the threshold of the node included between -1 and 1.

Fig. 1. Illustration of our artificial neural networks model
2.2 Genetic Algorithms

Genetic algorithms (GA) are heuristic optimization methods that draw their inspiration from the biological evolution of species [7]. They are based on selection process of best individuals as well as on reproduction and mutation process.

We implemented two versions of multi-objective genetic algorithms [5]. The first one is a weighted GA, i.e. a GA in which the fitness is a combination of scores got for each task achievement. The second one is based on the pareto-optimality theory [4]. Individuals are sorted according to their rank of non-dominance and receive a fitness depending on this rank. Figure 2 represents the pareto-front got for different runs of our genetic algorithm.

![Pareto-optimality front](image)

**Fig. 2.** Pareto-optimality front obtained for different simulations

We performed a preliminary analysis to get optimal parameters for our genetic algorithms: crossover and mutation rate or number of generations. Let us note that genetic algorithms are the tools we chose to train our networks but other heuristic methods could be considered in future research.

3 Application in Robotics

Until now, we principally worked on the application of networks learning in robotics. Here is a summary of our experiments and of our ideas for the following.

3.1 Context

We work on a abstract application in robotics inspired from an article of Beaumont [1]. Indeed, virtual robots controlled by neural networks are trained to learn two different tasks in a virtual arena. This arena (see Fig. 3) is a discretized grid with a torus shape on which robots can reach any neighbouring
square at each displacement, that is the 8-squares Moore neighbourhood. The first task involves reaching a global maximum of the arena represented using a color code in the following Fig. 3 and the second one moving by avoiding zones where they lose energy, black boxes in the Fig. 3.

The robots that we consider are made of 17 sensors and 4 motors. The 9 first sensors help robots to check the height of the square on which it stays and the squares around them. The 8 others detect if there are “loss of energy” zones around them. The 4 motors control the movements of robots. Two of them are dedicated to the west-east movements and the two others to the north-south movements. Hence, the neural networks that control the robots are made of 43 nodes: 17 inputs, 4 outputs (corresponding to the number of sensors and motors) and 22 hidden neurons (a number necessary to let sufficient possibilities of connections and to compare global and modular learning). They are trained for the two tasks with genetic algorithms that we implemented.

In order to analyze characteristics of our final networks, we ran different simulations with the weighted genetic algorithm. First, we trained robots to manage only one task. Secondly, robots are trained to achieve both tasks together. Finally, we took the neural networks got after the training on one task and we added a training on both tasks. We hope to find differences between robots that learn both tasks together and robots that learn one task before the other. We also ran different simulations with the genetic algorithm based on pareto optimality. In this case, both tasks are trained together and we got a set of solutions instead of one unique solution (see Fig. 2). We took advantage of this preliminary analysis phase to compare our two genetic algorithms. For the same number of generations, we got better fitness for weighted GA than for the other one. Our conjecture is that this is due to the way of defining the fitness according to the rank of non-dominance and it could be interesting to perform more experiments to check this hypothesis.
3.2 Useless Links

Although neural networks so far obtained lead to the expected robots behaviours, they exhibit many “useless” links. So, we decided to consider different methods to decrease the number of links.

Firstly we tried to decrease the number of links by adding a third objective, namely a penalty on the number of links, in our genetic algorithms. But, concerning the weighted GA, the choice of the combination of weights to introduce is not trivial and the reduction obtained is very low and slow. On the other hand, for the GA based on pareto optimality, adding a third objective prevent best networks from reaching equivalent tasks performance to the one that they have in the case of two objectives optimization.

Another idea was to replicate the best networks got with simulations performed without any links constraints by removing “useless” links as long as possible. To remove these links, we begin by choosing one link randomly. Then we remove it and we check if the fitness has changed. If it’s the case than the link is reinserted in the network, otherwise it is definitely removed and we go on by choosing another link. We stop this part of the algorithm when we get a sequence of 50 randomly chosen links that can not be removed. Next, we check every remaining link to see which ones can be removed without changing the fitness and we choose one of them that we removed. We iterate this algorithm until it’s no more possible to keep the same fitness by removing any links. We saw that this method had one big limit. It’s a random method and the final matrix depends on the order in which we removed the links. Moreover we observed the following fact: removing a “useless” link will sometimes reduce the fitness. Indeed, for each node, the sum of stimuli are divided by the number of incoming links and compared with a threshold to see if the neuron is activated or not. By removing links, we change the divisor and we influence the comparison.

In order to overcome this limit, we thought to modify the thresholds each time that we remove a link. Some preliminary experiments showed that this change leads to networks with a lower number of connections. We could also consider to introduce this approach in the genetic algorithms themselves. Another way that we could explore would be to develop a kind of simulated annealing for removing links with a certain probability. This method could bring to less random networks.

3.3 Modularity

We realize that two kinds of modularities can be defined. The first one is a topological modularity [6] that can be studied with some community detection methods such as the Louvain method [2] or by using measures of centrality or similarity. In order to analyze the second one, namely the functional modularity, that is to group together neurons that have similar dynamic behaviours, we
follow two trails. We plan to use some recent information-like indicators [11]. We also decide to check the node relevance by randomly removing hidden nodes. Results are not yet available at the moment of writing this report.

4 Conclusion

This paper is the state of progress, after one year, of our research in the framework of a PhD thesis. We present the goal of our research, namely the emergence of modular structures in artificial neural networks trained to achieve peculiar tasks by using genetic algorithms. We introduce the mathematical tools that are exploited and an application in the field of robotics. The preliminary analysis are detailed and current research is broached. We finish by a presentation of work that we plan to study the modularity in our networks.

Acknowledgements

We would like to thank Andrea Roli for his help and his involvement in this project. This research used computational resources of the “Plateforme Technologique de Calcul Intensif (PTCI)” located at the University of Namur, Belgium, which is supported by the F.R.S.-FNRS.

References