Evolution of Artificial Neural Networks

Stefano Nolfi Domenico Parisi
Institute of Psychology
National Research Council
15, Viale Marx, 00137 - Rome
e-mail:nolfi@ip.rm.cnr.it parisi@ip.rm.cnr.it

Introduction

Artificial neural networks are computational models of nervous systems. Natural organisms, however, do not possess only nervous systems but also genetic information stored in the nucleus of their cells (genotype). The nervous system is part of the phenotype which is derived from this genotype through a process called development. The information specified in the genotype determines aspects of the nervous system which are expressed as innate behavioral tendencies and predispositions to learn. When neural networks are viewed in the broader biological context of Artificial Life they tend to be accompanied by genotypes and to become members of evolving populations of networks in which genotypes are inherited from parents to offspring (Parisi, 1997).

Artificial neural networks can be evolved by using evolutionary algorithms (Holland, 1975; Schwefel, 1995; Koza, 1992). An initial population of different artificial genotype, each encoding the free parameters (e.g. the connection strengths and/or the architecture of the network and/or the learning rules) of a corresponding neural network, are created randomly. The population of networks is evaluated in order to determine the performance (fitness) of each individual network. The fittest networks are allowed to reproduce (sexually or a-sexually) by generating copies of their genotypes with the addition of changes introduced by some genetic operators (e.g., mutations, crossover, duplication). This process is repeated for a number of generations until a network that satisfies the performance criterion (fitness function) set by the experimenter is obtained (for a review of methodological issue see Yao, 1993).

The genotype might encode all the free parameters of the corresponding neural network or only the initial value of the parameters and/or other parameters that affects learning. In the former case of the network is entirely innate and there is no learning. In the latter networks changes both philogenetically (i.e. through out generations) and ontogenetically (i.e. during the period of time in which they are evaluated).

Evolution and development

cases, all phenotypical characteristics are coded in an uniform manner so that the description of an individual at the level of the genotype assumes the form of a string of identical elements (such as binary or floating point numbers). The transformation of the genotype into the phenotypical network is called genotype-to-phenotype mapping.

In direct encoding schemes there is a one-to-one correspondence between genes and the phenotypical characters subjected to the evolutionary process (e.g. Miller et al., 1989). Aside from being biological implausible, simple one-to-one mappings has several drawbacks. One problem, for example, is scalability. Since activation variability of the corresponding neurons was larger than a genetically-specified threshold. This simple mechanism is based on the idea that sensory information coming from the environment has a critical role in the maturation of the connectivity of the biological nervous system and, more specifically, that the maturation process is sensitive to the activity of single neurons (see Purves, 1994). Therefore the developmental process was influenced both by genetic and environmental factors (i.e. the actual sequence of sensory states experienced by the network influenced the process of neural growth).

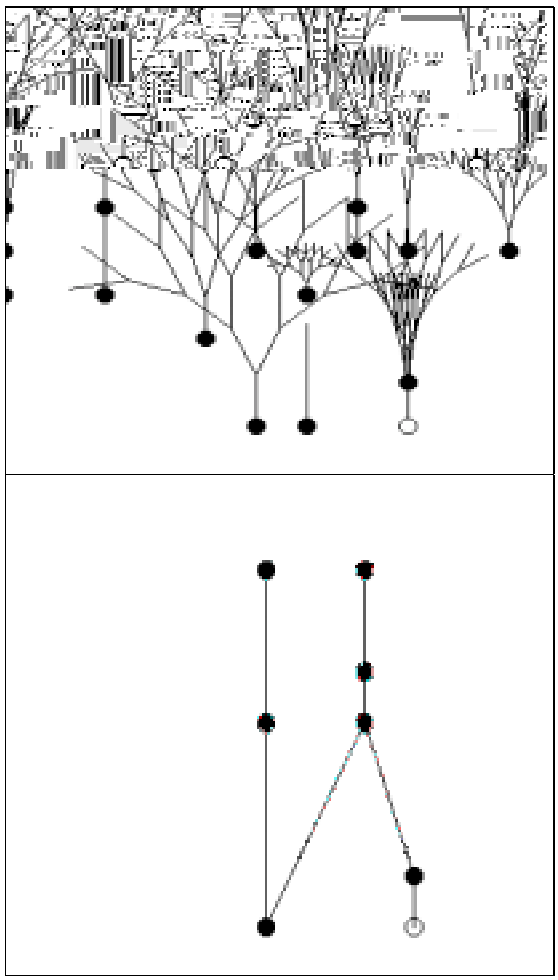

Figure 1. Development of an evolved neural network. Top: The growing and branching process of the axons. Bottom: the resulting neural network after removal of nonconnecting branches and the elimination of isolated neurons and groups of interconnected neurons.

This method allows the evolutionary process to select neural network topologies that are suited to the task chosen. Moreover, the developmental process, by being sensitive to the environmental conditions, might display a form of plasticity. Indeed, as shown by the authors, if some aspects of the task are allowed to vary during the evolutionary process, evolved genotypes display an ability to develop into different final phenotypical structures that are adapted to the current conditions.

Cellular Encodings

In natural organisms the development of the nervous system begins with a folding in of the ectodermic tissue which forms the neural crest. This structure gives origin to the mature nervous system through three phases: the genesis and proliferation of different classes of neurons by cellular duplication and differentiation, the migration of the neurons toward their final destination, and the growth of neurites (axons, dendrites). The growing process described in the previous section therefore characterizes very roughly only the last of these three phases. A number of attempts have been made to include other aspects of this process in artificial evolutionary experiments.

Cangelosi et al. (1994), for example, extended the model described in the previous section by adding a cell division and migration stage to the already existing stage of axonal growth. The genotype, in this case, is a collection of rules governing the process of cell division (a single cell is replaced by two "daughter" cells) and migration (the new cells can move in the 2D space). The genotype-to-phenotype process therefore starts with a single cell which, by undergoing a number of duplication and migration processes, produces a collection of neurons arranged in a 2D space. These neurons grow their axons and establish connection until a neural controller is formed (for a related approaches see Dellaert and Beer, 1994).

Gruau (1994) proposed a genetic encoding scheme for neural networks based on a cellular duplication and differentiation process. The genotype-to-phenotype mapping starts with a single cell that undergoes a number of duplication and transformation processes ending up in a complete neural network. In this scheme the genotype is a collection of rules governing the process of cell divisions (a single cell is replaced by two "daughter" cells) and transformations (new connections can be added and the strengths of the connections departing from a cell can be modified). In this model, therefore, connection links are established during the cellular duplication process.

The instructions contained in the genotype are represented as a binary-tree structure as in genetic programming (Koza, 1992). During the genotype-to-phenotype mapping process, the genotype tree is scanned starting from the top node of the tree and then following each ramification. The top node represents the initial cell that, by undergoing a set of duplication processes, produces the final neural network. Each node of the genotype tree encodes the operations that should be applied to the corresponding cell and the two sub-trees of a node specify the operations that should be applied to the two daughter cells. The neural network is progressively built by following the tree and applying the corresponding duplication instructions. Terminal nodes of the tree (i.e. nodes that do not have sub-trees) represents terminal cells that will not undergo further duplications. Gruau also considered the case of genotypes formed by many trees where the terminal nodes of a tree may point to other trees. This mechanism allows the genotype-to-phenotype process to produce repeated phenotypical structures (e.g. repeated neural sub-networks) by re-using the same genetic informations. Trees that are pointed to more than once, in fact, will be executed more times. This encoding method has two advantages: (a) compact genotypes can produce complex phenotypical networks, and (b) evolution may exploit phenotypes where repeated sub-structures are encoded in a single part of the genotype. Since the identification of sub-structures that are read more than once is an emergent result of the evolutionary process, Gruau defined this method Automatic Definition of Neural Subnetworks (ADNS) (Gruau, 1994).

Discussion

Artificial evolution can be seen as a learning algorithm for training artificial neural networks. From this point of view, one distinctive feature is the limited amount of feedback required. Supervised learning algorithms require immediate and detailed desired answers as a feedback. Reinforcement learning algorithms require less - only a judgement of right or wrong which should not be necessarily immediate. Viewed as a learning algorithm, artificial evolution requires still less - only an overall evaluation of the performance of the network over the entire evaluation period. A second distinctive feature is that any parameter of the neural network (e.g. the connection strengths, the network topology, the learning rules, the transfer function of the neurons) can be subjected to the evolutionary process.

Although systematic comparison between artificial evolution and other algorithms are not been done yet, it is reasonable to claim that artificial evolution tend to produce better results when detailed feedback is not available. This is the case, for example, of a neural networks that should control mobile robots (Nolfi and Floreano, 2000). In this case in fact, although the experimenter can provide a general evaluation of how much the behavior of a robot approximates the desired behavior, he or she cannot usually indicate what the robot should do each time step to produce such a desired behavior. Moreover artificial evolution might result more effective in those cases in which certain features of the network (such as the network topology or the transfer functions) that cannot be properly set by hand are crucial. Artificial evolution, in fact, provide a way to co-adapt different type of parameters.

The analogy with natural evolution however can also be considered more strictly. In this case the evolutionary process is not seen as an abstract training algorithm but as a process that mimics some of the key aspects of the evolutionary process in nature. From this point of view neural networks tend to be viewed as a part of a population of artificial organisms that adapt autonomously by interacting with the external environment.

References

Cangelosi, A., S. Nolfi, and D. Parisi, 1994. Cell division and migration in a 'genotype' for neural networks, Network- Computation in Neural Systems, 5:497-515.

Dellaert F. and R.D. Beer (1994). Toward an evolvable model of development for autonomous agent synthesis, in Proceedings of the Forth Conference on Artificial Life, (R. Brooks, and P. Maes, eds.), Cambridge, MA: MIT Press.

Gould, S.J., 1977. Ontogeny and Phylogeny. Cambridge, MA: Harward University Press.

Gruau F., 1994. Automatic definition of modular neural networks, Adaptive Behavior, 3:151-183.

Holland, J.J., 1975, Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press.

Husbands, P., I. Harvey, D. Clnfl08 (y)1. Miller, 1994. The use of genetic algorithms for the development of sensorimotor control systems, in From Perception to Action, (P. Gaussier08 (y)J-D. Nicoud, eds.), Los Alamitos CA: IEEE Press.

Kitano, H., 1990. Designing neural networks using genetic algorithms with graph generation system, Complex Systems, 4:461-476.

Koza, J.R., 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA: MIT Press.

Miller, G.F., P.M. Todd08 (y)S.U. Hedge (1989). Designing neural networks using genetic algorithms, in Proceedings Third International Conference on Genetic Algorithms, (L. Nadel, (y)D. Stein, eds.), San Mateo, CA: Morgan Kaufmann, pp. 379-384.

been provided by Hinton and Nowlan (1987). The authors considered a simple case in which (a) the genotype of the evolving individuals consists of 20 genes that encode the architecture of the corresponding neural networks, and (b) only a single architecture (i.e. only a single combination of gene values) confers added reproductive fitness. Individuals have a genotype with 20 genes that can assume two alternative values (0 or 1). The only combination of genes that provide a fitness value above 0 consists of all ones. In this extreme case, the probability of finding the good combination of genes would be very small given that the fitness surface looks like a flat area with a spike in correspondence of the good combination. Indeed, on such a surface, artificial evolution does not perform better than random search. Finding the right combination is like looking for a needle in a haystack. The fitness surface is a metaphor often used to visualize the search space on an evolutionary algorithm. Any point on the search space corresponds to one of the possible combinations of genetic traits and the height of each point on the fitness surface corresponds to the fitness of the individual with the corresponding genetic traits.

The addition of learning simplify significantly the evolutionary search. One simple way to introduce learning is to assume that, in learning individual, genes can have three alternative values [0, 1, and ?] where question marks indicate modifiable genes whose value is randomly selected within [0, 1] each time step of the individuals' lifetime. By comparing learning and non-learning individuals one can see that performance increases throughout generations much faster in the former than in the latter. The addition of learning, in fact, produces an enlargement and a smoothing of the fitness surface area around the good combination that, in this case, can be discovered much more easily by the genetic learning and evolution display also other forms of interactions that are mutually beneficial.

providing a mean to master changes that occur too fast to be tracked by the evolutionary process. However, as we will see in this section, the combination of learning and evolution deeply alter both processes so that, in individuals that evolve and learn, adaptive characteristics emerge as the result of the interaction between evolutionary and lifetime adaptation and cannot be traced back to only one of the two processes.

Nolfi and Parisi (1997), evolved neural controllers for a small mobile robot that was asked to explore an arena of 60 x 20 cm surrounded by walls. The robot was provided with 8 infrared sensors that could detect walls up to a distance of about 4 cm and two motors that controlled the two corresponding wheels. The colors of the walls switched from black to white and viceversa each generation. Given that the activity of the infrared sensors is highly affected by the color of the reflecting surface (white reflect much more that black walls), to maximize their exploration behavior, evolved robots should modify their behavior on the fly. In the environment with dark walls, in fact, robots should move very carefully when sensors are activated given that walls are detected only when they are very close. In the environment with white walls, on the contrary, robots should begin to avoid walls only when the sensors are strongly activated in order to explore also the area close to the walls.

Individuals learn during lifetime by means of a self-generated teaching signals. The genotype of the evolving individuals encoded the connection strengths of two neural modules: a teaching module that each time step receives the state of the sensors as input and produce a teaching signal as output and an action module that receives the state of the sensors as input and produce motor actions as output. The self-generated teaching signal is used to modify the connection strengths of the action module (for a similar architecture see also Ackley and Littman, 1991). This implies that not only the initial behavior produced by the evolving individuals but also what individuals learn is the result of the evolutionary process and is not determined by the experimenter.

Evolved robots displayed an ability to discriminate the two types of environments and to modify their behavior accordingly thus maximizing their exploration capability. The analysis of the obtained results revealed that this ability resulted from a complex interaction between the evolutionary and learning process. For example, evolved individuals displayed an inherited ability to behave so to enhance the perceived differences between the two environments. This Thies ee52.5()-0(e)eah1

Other experiments conducted by co-evolving two competing populations of predator and prey robots (Nolfi and Floreano, 1998) emphasized how lifetime learning might allow evolving individuals to achieve generality, i.e. the ability to produce effective behavior in a variety of different circumstances. Predators consisted of small mobile robots provided with infrared sensors and a linear camera with a view angle of 36° with which they could detect the prey. Prey consisted of mobile robots of the same size provided only with infrared sensors but that had a maximum available speed set to twice that of the predators. Predators were selected for their ability to catch prey while prey were selected for their ability to escape predators.

What is interesting about this experimental situation is that, given that both populations changes throughout generations, predators and prey are facing ever-changing and potentially progressively more complex challenges. Interestingly the authors observed that in this situation, evolution alone displayed severe limitations and progressively more effective solutions could be developed only by allowing evolving individuals to adapt on the fly through a form of lifetime learning. Indeed, any possible fixed strategy was able to master different type of competitors and therefore only by combining learning and evolution the authors were able to synthesize individuals able to deal with competitors adopting qualitatively different strategies. Indeed, by evolving learning individuals, the authors observed the emergence of predators able to detect the current strategy adopted by the prey and to modify their behavior accordingly.

Other advantages

Floreano and Urzelai (in press) conducted a set of experiments in which the genotype of the evolving individuals encoded the learning properties of the neurons of the corresponding neural network. These properties included one of four possible hebbian learning rules, the learning rate, and the sign of all the incoming synapses of the corresponding neuron. When the genotype is decoded into a neural controller, the connection strengths are set to small random values. As reported by the authors, after some generations, the genetically specified configuration of learning rules tend to produce changes in the synaptic strengths that allow individuals to acquire the required competencies through lifetime learning. By comparing the results obtained with this method with a control experiment in which the strength of the synapses were directly encoded into the genotype, the authors observed that evolved controllers able to adapt during lifetime can solve certain tasks faster and better than standard non-adaptive controllers. Moreover they demonstrated that their method scales up well to large neural architectures.

The authors applied this method to evolve neural controllers for a mobile robots. Interestingly, the analysis of the synaptic activity of the evolved process (e.g., neural networks trained with supervised methods) learning is usually accomplished by ignoring the characters of the individual prior to learning (which are typically generated at random), in evolving plastic individuals learning exploits such starting conditions. Moreover, when the learning process itself (i.e. what it is learn during lifetime) is subjected to evolution and not determined in advance, learning does not necessarily tend to incorporate the right solution to the problem but rather it tends to pull the learning individual in a direction that, given the initial state of the individual, maximizes the chances of acquiring adaptive characters.

References

Ackley, D.H., and M.L. Littman, 1991. Interaction between learning and evolution, in Proceedings of the Second Conference on Artificial Life, (C.G. Langton et. al eds.), Reading, MA: Addison-Wesley, pp. 487-509.

Belew, R.K., J. McInerney J., and N.N. Schraudolph N.N., 1992. Evolving networks: using the genetic algorithm with connectionistic learning, in Proceedings of the Second Conference on Artificial Life, (C.G. Langton et. al eds.), Reading, MA: Addison-Wesley.

Di Paolo, E.A., 2000. Homeostatic adaptation to inversion in the visual field and other sensorimotor disruptions, in From Animals to Animats 6. Proceedings of the VI International Conference on Simulation of Adaptive Behavior, (J-A Meyer, A. Berthoz, D. Floreano, H.L. Roitblat, and S.W. Wilson eds.), Cambridge, MA: MIT Press, pp. 440-449.

Floreano, D., and J. Urzelai, in press. Evolving robots to face unpredictable change, Robotics and Autonomous Systems.

Nolfi, S., 1999. How learning and evolution interact: The case of a learning task which differs from the evolutionary task, Adaptive Behavior, 2:231-236.

*Nolfi, S., and D. Floreano, 1999. Learning and evolution, Autonomous Robots, 1: 89-113.

Nolfi S., and D. Floreano, 1998. Co-evolving predator and prey robots: Do 'arm races' arise in artificial evolution?, Artificial Life, 4:311-335.

Nolfi, S., and D. Parisi, 1997. Learning to adapt to changing environments in evolving neural networks, Adaptive Behavior, 1:75-98.

Nolfi, S., J.L. Elman, and D. Parisi, 1994. Learning and evolution in neural networks, Adaptive Behavior, 1:5-28.

Todd, P.M., and G.F. Miller, 1991. Exploring adaptive agency II: simulating the evolution of associative learning, in From Animals to Animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior, (J. A. Meyer and S.W. Wilson eds.), Cambridge, MA: MIT Press.