Adaptive Systems and Evolutionary Neural Networks : a Survey

M.Annunziato[1], M.Lucchetti[2], S.Pizzuti[1,3]
[1]ENEA – Energy, New technologies and Environment Agency, 'Casaccia' R.C.
Via Anguillarese 301, 00060 Rome, Italy
Phone: +39-06-30484411, Fax: +39-06-30484811
email:{mauro.annunziato, stefano.pizzuti}@casaccia.enea.it
[2]University of Rome 'La Sapienza' Department of Computer and Systems Science
Via Eudossiana 18, 00184 Rome Italy
Phone: +39-06-44585938, Fax: +39-06-44585367
email: lucchetti@dis.uniroma1.it
[3]CS – Communication Systems S.p.A.
Piazza della Repubblica 32, Milan Italy

ABSTRACT: During last decades there has been an increasing interest in artificially combining evolution and learning, in order to pursue adaptivity and to increase efficiency of control, supervision and optimisation systems. In particular the need for adaptation came out from several real-world applications in non-stationary environments ranging from non linear control tasks to manufacturing process optimisation, from time series forecasting to interactive game playing. These needs led to the birth of a new general framework for adaptive systems, namely the Evolutionary Artificial Neural Networks, where the modelling potentialities of artificial neural networks have been matched with the adaptation properties of the evolutionary algorithms. This paper briefly reviews the main results achieved and presents the state-of-the-art in this field.

KEYWORDS: evolutionary neural networks, adaptive systems, non-stationary environment, neural networks, evolutionary computation

INTRODUCTION

In several real-world dynamical systems applications, ranging from robotics to telecommunications, from process automation to biology simulations, it's impossible to formulate an *a priori* exact model of the system, taking into account all the variables influencing the evolution during time, partly because of the presence of some unobservable dynamics in the system, which result in non-stationary phenomena on the measured data, and partly because of the unavoidable and unpredictable noise that affects the system internal state and its output.
In such cases a possible approach is to extract a first rough offline model, and then to on line update it, in order to recover the gap of knowledge about unknown disturbances and to pursue an on line adaptation of the model itself. This task has been the research subject of many groups involved in the development of smart adaptive systems for real applications.
One of the most commonly ascribed approach to reach adaptive continuous updating consisted in matching the neural networks modelling capabilities with the adaptation properties of evolutionary algorithms. These investigations led to the birth of a new framework which is generally referred to as Evolutionary Artificial Neural Networks (EANNs). The model we get from a neural network undergoes the evolution superimposed by the artificial environment of the evolutionary algorithm which is related to the evolution of the system. In such a way it's possible to couple the evolution of the population and the learning process of each individual of the population, achieving better adaptation of the whole environment to a generic dynamic fitness landscape.

EVOLVING NEURAL NETWORKS: THEORETICAL ISSUES AND PREVIOUS WORK

The first attempts to conjugate evolutionary algorithms with neural networks dealt with the offline set-up of the networks (i.e. training of the connection weights [1], offline design of the neural architecture [2][3], or both [4][5]). This type of coupling is still implemented in many applications (see for example [6] [7] [8]). However the achieved model isn't able to describe the evolution of the system, its utility must be principally regarded in the higher robustness and simplicity of the topology of the resulting networks with respect to those obtained with classical training algorithms or empirically designed. For a general review of these methodologies see [9].

As we stated above, our focus is on the capability to on line follow the evolution of a partially known dynamical system via a continuous updating of the model knowledge through evolutionary optimisation [10]. This reflects upon a simultaneous evolution of the neural model. On line evolution has been introduced in neural networks at three different levels*: connection weights*, *architectures* and *learning rules* [11].

The evolution of *connection weights* extends the approach of offline training, by means of on line training of the connections in a fixed topology network, introducing a first step of adaptivity in the neural model. Several attempts in this way have already been carried out [12] [13]. Moreover it is worth to mention the recent efforts in this level to conjugate evolutionary with gradient based methods [14].

The next step in providing adaptivity to the neural model is to evolve its *architecture*. With the term architecture we intend not only the topological structure, but also the transfer functions of the neurons. The evolution of *architectures* allows neural networks to adapt their topologies to different tasks or to a non stationary changing environment without human intervention, providing an approach to automatic neural network design. Some works related to this topic are [15]. Obviously this latter class of EANNs includes the former, and several algorithms evolving both structure and weights have been proposed [16] [17].

The last step is represented by the capability of the network to evolve its *learning rules*. To do this we have to encode in each chromosome a dynamic behaviour. Yao in [11] formalizes the concept of learning rule giving it a mathematical structure depending on a large number of parameters and evolution has to set those parameters. Research into the evolution of learning rules is still in its early stages, but adapting a learning rule through evolution is expected to greatly and efficiently enhance the neural network's adaptivity in a dynamic environment (see for example [18] and [19]).

CURRENT IMPLEMENTATIONS AND APPLICATIONS

The theoretical issues described above found a lot of diverse implementations and several applications have been developed. Just to have an overview we introduce the most meaningful contributions.

The first attempt to evolve the weights of a fully-connected recurrent network can be tracked down in [20]. In this work Wieland describes a GA based adaptive approach to a typical control problem, the double pole balancing (also known as double inverted pendulum). The fitness function of each generated set-up of the network was the length of time that the pole remained balanced. Although the topology of the evolving network is fixed and is empirically established offline, the weights of the network are evolved as a consequence of the dynamics of the controlled system.

The double pole balancing task was referred to as a benchmark in following papers and inspired the work of Saranavan and Fogel [12]. They tackled the problem by introducing an evolutionary programming scheme in a population of neural networks. The output of the cart system was used as input for each network, all with the same fixed topology, which is able to evolve its weights and biases. The selected network's output response is in turn used as a force applied to the cart system. This is the first closed continuous loop between a system and an evolving environment of neural networks.

We stressed above that in real applications it's important to online update the model. An interesting contribution is offered by [21] where it's analysed an online evolution of a population of networks in a real-time interactive environment. The problem is that it's required a strict adaptivity to the changing environmental conditions imposed by the opponent interactive player. The approach presented is then tested on a gaming application, but its inspiring philosophy can be applied in many fields such as robot control in unstructured environments or traffic management.

One more distinction we have to point out is the codification of each individual. In the previous paragraph we distinguished among three possible codifications of the genome of each individual (connection weights, architecture, learning rules). All of these require that each individual refers to a complete neural network. In this sense we can speak of an evolving population of neural networks. Another approach is that of [22] where each individual refers to a single neuron, and the neural networks formed are the results of the evolution of a population of these individuals via co-evolution and speciation. The competition between individuals is based on how well, on average, the network which they participate in perform. This approach is called symbiotic evolution and the developed base methodology (called SANE, Symbiotic Adaptive Neuro-Evolution) has been applied to various problems with very promising results ([23] [24] [25]).

Another important contribution to the development of evolving neural network came from [26]. In particular Stanley et al. propose a method of evolving topologies of networks with unbounded complexity (NEAT). The evolution of topologies allows the networks to support plastic synapses by designating which connection should be active and in which way. Moreover it has been noted that NEAT strengthens the analogy with natural evolution by simultaneously optimising and complexifying solutions. Experimentation of this methodology mainly concerned the already mentioned double-pole balancing control problem.

It is worth to mention the framework proposed by [27] concerning evolving fuzzy neural networks (EfuNN). In this work EfuNNs evolve their structure and parameter values through incremental, online learning and adaptation to the changing environment is performed by evolutionary methods. This approach has been tested on time series prediction. Similar works are [28] and [29].

Recent real world applications range from image processing, to vehicle control, from evolutionary robotics to telecommunications.

In image processing [14] focus on the on-line training of neural networks in slowly varying environments with a Hybrid Evolutionary Algorithm (HEA) applied to texture classification and tumour detection problems. [30] [31] [32] focus on the detection and extraction of objects in images under variable perceptual conditions.

In vehicle control [33] applies evolutionary algorithms to on-line adapt control modules represented by neural networks in order to achieve fuel injection control.

In evolutionary robotics [34] [35] [36] provide examples of robotics applications in varying environments using different evolutionary and dynamical neural networks.

In telecommunications the call admission control, regarded as non stationary time series forecasting [37] has been faced. Another promising application in that field has been the dynamic bandwidth allocation problem [38].

CONCLUSIONS

In this paper we briefly reviewed the theoretical and methodological approaches to develop evolutionary neural network. We focussed our attention on those works where a continuous on line adaptation of the model to a non stationary environment is required. It's commonly thought that this is one of the most troubling features in real-world applications and, even if the first theoretical approaches to the problem in terms of adaptive artificial systems can be fixed in the first 90's, real implementations started later and at present they are not consolidated yet. Recent applications show good adaptivity to unpredictable variations in time of the dynamics of the system and the results obtained can be considered both a promise and a challenge for future research in smart adaptive systems.

BIBLIOGRAPHY

[1] D. Montana and L. Davis, *Training feedforward neural networks using genetic algorithms*, in Proc. of *Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA, USA, pp. 762-767, 1989.

[2] Y. Liu, X. Yao, *Evolutionary Design of Artificial Neural Networks with different nodes*, in Proc. of *IEEE International Conference on Evolutionary Computation*, ICEC 96, Nagoya, Japan, pp. 670-675, 1996.

[3] Y. Liu, X. Yao, *Evolving neural networks for Hang Seng stock index forecast*, in Proc. of the *2001 Congress on Evolutionary Computation*, CEC 01, Seoul, Korea, vol. 1, pp.256-260, 2001.

[4] P. Kocalka, V. Vojtek, , *Problem Solving Based on Evolutionary Neural Network Algorithms*, Proc. of *23rd Int. Conf. On Information Technology Interfaces*, ITI 2001, Pula, Croatia , pp.145-150, 2002.

[5] D. Wicker, M. M. Rizki, L. A. Tamburino, *E-Net: Evolutionary neural network synthesis*, Neurocomputing, vol. 42, pp. 171-196, 2002.

[6] V. P. Plagianakos, M. N. Vrahatis, *Training neural networks with threshold activation function and constrained integer weights*, in Proc. of the *IEEE Int. Joint Conference on Neural Networks*, IJCNN 2000, Como, Italy, vol. 5, pp. 161-165, 2000.

[7] J. Arifovic and R. Gençay, *Using Genetic Algorithms to select architecture of a Feedforward Artificial Neural Network*, Physica A, vol. 289, pp. 574-594, 2001.

[8] E.M. Iyoda, F.J. Von Zuben, *Evolutionary Hybrid Composition of Activation Functions in Feedforward Neural Network*, in Proc. of *International Joint Conference on Neural Networks*, IJCNN 99, Washington DC, USA, vol. 6, pp. 4048-4053, 1999.

[9] K. Balakrishnan and V. Honovar, *Evolutionary Design of Neural Architectures - A Preliminary Taxonomy and Guide to Literature*, Technical Report CS-TR 95-01, Iowa State University, Ames, IA, 1995.

[10]J. Branke, *Evolutionary Optimisation in Dynamic Environments*, Kluver Academic Publishers, 2001.

[11]X. Yao, *Evolving Artificial Neural Networks*, in *Proceedings of the IEEE*, 87(9): 1423-1447, 1999.

[12]N. Saravanan and D. B. Fogel, *Evolving neural control systems*, IEEE Expert, vol. 10, n. 3, pp. 23-27, 1995.

[13]G. W. Greenwood, *Training partially recurrent neural networks using evolutionary strategies*, IEEE Transactions on Speech and Audio Processing, vol. 5, n. 2, pp. 192-194, 1997.

[14]G. D. Magoulas, V. P. Plagianakos, M. N. Vrahatis, *Hybrid Methods Using Evolutionary Algorithms for On-Line Training*, in *Proc. of the IEEE International Joint Conference on Neural Networks*, IJCNN 2001, Washington DC, USA, vol.3, 2218-2223, 2001.

[15]X. Yao and Y. Liu, *A new evolutionary system for evolving artificial neural networks*, IEEE Trans. on Neural Networks, vol. 8, n. 3, pp. 694-713, 1997.

[16]V. Maniezzo, *Genetic evolution of the topology and weight distribution of neural networks*, IEEE Trans. on Neural Networks, vol. 5, no. 1, pp. 39-53, 1994.

[17]J. C. F. Pujol, R. Poli, *Evolving the topology and the weights of neural networks using a dual representation*, Applied Intelligence, vol. 8, pp. 73-84, 1998.

[18]S. Nolfi and D. Parisi, *Neural Networks in an Artificial Life perspective*, in Proc. of the *7th International Conference on Artificial Neural Networks*, ICANN97, Berlin: Springer Verlag, pp. 733-738, 1997.

[19]B. Carse and J. Oreland, *Evolution and Learning in Neural Networks: Dynamic Correlation, Relearning and Thresholding*, Adaptive Behaviour, vol. 8, n. 3/4, pp. 297-312, 1999.

[20]A. P. Wieland, *Evolving neural network controllers for unstable systems* in Proc. of *1991 IEEE International Joint Conference on Neural Networks*, IJCNN 91, Seattle, WA, USA, vol. 2, pp. 667-673, 1991.

[21]A. Agogino, K. Stanley and R. Miikkulainen, *Real-Time Interactive Neuro-Evolution*, Neural Processing Letters, vol. 11, pp. 29-37, 2000.

[22]D. E. Moriarty and R. Miikkulainen *Efficient Reinforcement Learning through Symbiotic Evolution*, Machine Learning, vol. 22, pp. 11-32, 1996.

[23]F. J. Gomez, D. Bourger and R. Miikkulainen, *A Neuroevolution Method for Dynamic Resource Allocation on a Chip Multiprocessor*, in Proc. of the *International Joint Conference on Neural Networks*, IJCNN 2001, Washington DC, USA, pp. 2355-2360, 2001.

[24]A. E. Conradie, R. Miikkulainen, and C. Aldrich, *Intelligent Process Control Utilizing Symbiotic Memetic Neuro-Evolution*, in Proc. of *2002 Congress on Evolutionary Computation*, CEC 02, Honolulu, HI, USA, vol. 1, pp. 623-628, 2002.

[25]A. Lubberts and R. Miikkulainen, *Co-Evolving a Go-Playing Neural Network*, in *Genetic and Evolutionary Computation Conference Workshop*, GECCO 2001, San Francisco, CA, USA, pp. 14-19, 2001.

[26]K. O. Stanley and R. Miikkulainen, *Evolving Neural Networks through Augmenting Topologies*, Technical Report AI01-290, Department of Computer Sciences, University of Texas at Austin, USA, 2001.

[27]N. Kasabov, *Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning*, IEEE Transactions on Systems, Man and Cybernetics, Part B, vol.31, 6, 2001.

[28]A. Topchy et al., *Adaptive training of radial basis function networks based on co-operative evolution and evolutionary programming*, in Progress in Connectionist-Based Information Systems, N.Kasabov et al. Eds, New-York, Springer-Verlag, 253-258, 1998.

[29]M. Watts and N. Kasabov, *Genetic algorithms for the design of fuzzy neural networks*, Proc. ICONIP, Kitakyushu, Japan, IOS Press, 793-796, 1998.

[30]S. W. Balk and P. Pachowicz, *Adaptive object recognition based on the radial basis function paradigm*, in Proc. of the *IEEE International Joint Conference on Neural Networks*, IJCNN 1999, Washington DC, USA, vol. 6, pp. 600-605, 1999.

[31]P.Pachowicz and S.W. Balk, *Adaptive RBF classifier for object recognition in images sequences*, Proc. of the *IEEE International Joint Conference on Neural Networks*, IJCNN 2000, Como, Italy, vol. 5, pp. 3138-3142, 2000.

[32]A.D. Doulamis, N.D. Doulamis, S. D. Kollias, *On-line retrainable neural networks : improving the performance of neural networks in image analysis problems*, IEEE Transactions on Neural Networks , vol. 11, pp. 137-155, 2000.

[33]I. Kamihira, M. Yamaguchi, H. Kita, *On-line Adaptation of Vehicles by means of an Evolutionary Control System*, in Proc. of *IEEE International Conference on Systems Man and Cybernetics*, IEEE SMC 99, Tokyo, Japan, vol. 5, pp. 553-558, 1999.

[34]E. Tuci, I. Harvey, M. Quinn, *Evolving integrated controllers for autonomous learning robots using dynamic neural networks*, in Proc. of *Simulation of Adaptive Behaviour*, SAB02, Edinburgh, UK, 2002.

[35]P. K. Kim, P. Vadakkepat, T. Lee, Peng, X., *Evolution of Control Systems for Mobile Robots*, in Proc. of the *2002 Congress on Evolutionary Computation*, CEC 02, Honolulu, HI, USA, vol. 1, pp. 617-622, 2002.

[36]S. Nolfi, D. Floreano, *Learning and Evolution*, Autonomous Robots, vol. 7, n. 1, pp. 89-113, 1999.

[37]Y. Huang, W. Yan, Z. Song, *Call Admission Control in ATM Networks based on Evolutionary Neural Networks*, in Proc. of the *IEEE National Aerospace and Electronics Conference*, NAECON 2000, Dayton, OH, USA, pp. 327-331, 2000.

[38]R. Bolla, F. Davoli, P. Maryni, T. Parisini, *An adaptive neural network admission controller for dynamic bandwidth allocation*, IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 28, n. 4, pp. 592-601, 1998.