Adaptive Learning in Changing Environments

Miguel Rocha[1], Paulo Cortez[2] and José Neves[1]

[1] Dep. Informática [2] Dep. Sistemas de Informação
Universidade do Minho, PORTUGAL
mrocha@di.uminho.pt pcortez@dsi.uminho.pt jneves@di.uminho.pt

Abstract. The remarkable adaptation of living creatures to their environments comes as a result of the interaction of two orthogonal processes: *evolution* and *learning*. Within the *Machine Learning* arena, both mechanisms inspired the development of the *Evolutionary and Connectionist Computation* fields, which can be combined under two different views, the *Lamarckian* and *Baldwinian* approaches. Comparative tests reward the latter when changing environments are considered.

Keywords: Baldwin Effect, Evolutionary Programming, Hybrid Systems, Lamarckian Optimization, Multi-Layer Perceptrons.

1 Introduction

In order for living creatures to achieve a better adaptation to a natural environment, they may take advantage at two interacting and complementary processes: *evolution* and *learning*. Evolution is a slow process that takes place at the population level, determining the basic structures of an organism. The lifetime learning is responsible for the adaptation at the individual's level. In terms of a computational procedure, evolution seems suitable for global search, while learning should be used to perform local search.

When combining evolution and learning, two major frameworks can be addressed, named *Lamarckian* [2] and *Baldwinian* [1], depending on whether the acquired traits are recoded into the chromosome or not. Some work in this arena has already been put forward [5][6]. However, most of these studies consider only a subset of the possible strategies or focus primarily on the benefits of lifetime learning, being the trade-off between costs and profits rarely considered [7].

In the present work, five different learning models will be tested, being the comparisons based on computation time, so that they can be fairer. The evolution process is here materialized via *Evolutionary Programming (EP)* [3], while lifetime learning is approached by the training of a *Multi-Layer Perceptron (MLP)* using a gradient descent based procedure [4]. The proposed models will be compared in several dynamic environments; i.e., the machine learning tasks are changed at regular intervals in time.

2 Learning Models

Five different models will be defined to approach each learning task:

The *Connectionist Model (CM).* Under these circumstances, the learning is achieved by a single *MLP* with a fixed topology. The activation function was the logistic one, and the training is achieved by the *RPROP* algorithm, chosen due to its faster convergence and stability [4].

The *Population of Connectionist Models (PM).* A set of 20 *MLP's* will improve only via the learning algorithm (*RPROP*); i.e., no genetic or selection procedure is applied.

The *Darwinian Model (DM).* In this approach, the overall learning process is accomplished by *EP*, where a population of 20 real-valued chromosomes is evolving, each coding the weights of a *MLP*, similar to the ones described above. In each iteration, 50% of the individuals are kept from the previous generation, being the remaining bred through the application of a *mutation* operator. Two mutation operators were tested: replacing the gene by a random value within the range $[-1.0; 1.0]$ and applying a gaussian perturbation, which adds a value taken from a narrow Gaussian distribution with a zero mean (i.e., small perturbations will be preferred over larger ones). In both cases the mutation operator is applied to 20% of the genes in the chromosome, randomly selected.

The *Lamarckian Model (LM).* The *LM* combines both lifetime learning and evolutionary approaches. Similarly to the *DM*, the *EP* is still the engine of the process, but in this case each individual is allowed to learn during its lifetime, in this case by running the *RPROP* algorithm for 50 epochs in each iteration of the *EP* process. Then, the improved weights are encoded back into the chromosome (Figure 1).

The *Baldwinian Model (BM).* The *BM* approach is close to the *LM*, except that the lifetime learning is only used to improve the fitness of the individuals, and the new weights are not encoded back into the genome (Figure 1). This means that, in the process of reproduction, the offspring does not inherit the acquired genetic information from their ancestors.

For all models, the initial weights are randomly assigned within the range $[-1.0; 1.0]$. The training accuracy of each *MLP* is measured in terms of the *Root Mean Squared Error (RMSE)*, according to the expression:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{p} \sum_{j=1}^{m} (T_{i,j} - F_{i,j})^2}{pm}}$$

where p denotes the number of the training patterns, m the number of the *MLP* outputs, $T_{i,j}$ the target and $F_{i,j}$ the actual value, both for the output j and the i input pattern. This metric is used as the fitness value for each individual

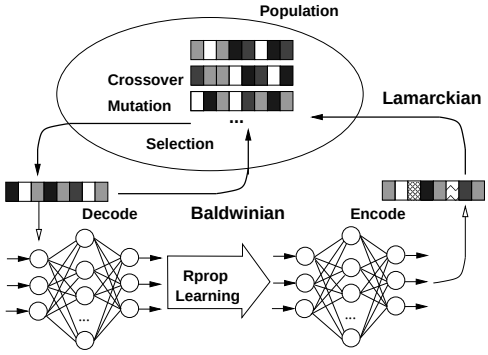

Figure 1: An illustration of the Baldwinian and Lamarckian strategies of inheritance

in the evolutionary approaches (*DM*, *LM* and *BM*). For the population based models (*PM*, *DM*, *LM* and *BM*), the overall accuracy is given by the *RMSE* of the best individual.

3 Learning Tasks

Two artificial classification tasks will be endorsed in this study:

3.1 The *N Bit Classification (NBC)*

This task is inspired by the *N Bit Parity (NBP)* problem, which is a famous benchmark, defined by 2^N patterns of N inputs (N was set to 6 in the experiments) and one output, whose value is set to 1, if the total number of input bits set to 1 is odd, and 0 otherwise [4].

Based on this benchmark, two changing environments were defined:

NBC_1 - at startup, outputs are randomly set; then, at each t_1 seconds, one output is randomly selected to change (from 0 to 1 or the reverse); and

NBC_2 - the *NBP* task, considering that all the desired outputs will change periodically each t_2 seconds like before.

3.2 The *Three Color Cube (TCC)*

This is a simple task that consists in learning how to paint a large 3D cube, made up by a 3x3 grid of blocks (27 smaller cubes) (Figure 2) [2]. Each input stands for the blocks' coordinates on the X, Y and Z axis, while each output represents a different color (100 for black, 010 for gray and 001 for white). In the static environment, the corners are black, the cubes in the center are white, being the others gray (Figure 2).

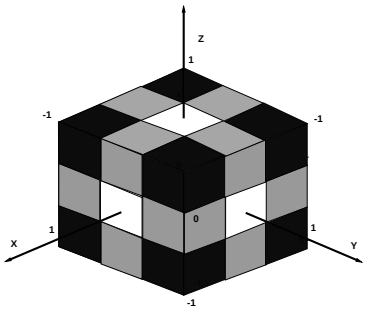

Figure 2: The *Three Color Cube* in its static variant

Again, two different approaches were followed when building the dynamic environments. In the former case (TCC_1), each cube is initially assigned a random color (black, gray or white), and then, at each t_1 seconds, the color of one randomly chosen cube is changed. In the latter one (TCC_2), the static discrimination rules prevail, although all the cubes are periodically repainted (each t_2 seconds), following a predefined order (black follows gray, gray follows white and white follows black).

4 Experimental Results

All experiments reported in this work were conducted using programming environments developed in *Java*, under the *Linux* operating system. The results obtained were compared in terms of two orthogonal parameters, the overall learning's *accuracy* (*RMSE*), and the process' *efficiency*, measured by the time elapsed (in seconds). For all models, at each time/generation slot, one considers the average of the results obtained in 20 independent runs.

In terms of the experiments' setup, the *MLP* topologies were fixed to $6-6-1$ and $3-8-3$ for the *NBC* and *TCC* tasks (the form $n_i-n_h-n_o$ denotes a network with n_i input, n_h hidden and n_o output nodes). Based on previous experiments, the gaussian perturbation was applied on the *DM* approach while the random mutation was used on the *LM* and *BM* strategies. For the "soft" changing environments, defined by NBC_1 and TCC_1 (since only one pattern is changed periodically), the dynamic changes were set to $t_1 = 1s$, while $t_2 = 20s$ was used for the "hard" changing tasks (NBC_2 and TCC_2). Finally, the termination criteria was set by a time limit T, here set to $T = 1000s$.

When considering the soft changing experiments (Figure 3), one can observe the inability of the *CM* to cope with changes in the environment, presenting the worst performance, since it is incapable of escaping the initial learning bias. Both *PM* and *DM* show a stable pattern, but seem unable to improve. When combining evolution and learning (*LM* and *BM* approaches), better results are achieved, with the *BM* behaving has the best alternative for both tasks. Since

models overlap, and to simplify the visual analysis, only the best strategies (*BM* and *LM*) were plotted for the hard changing tasks. Again, and despite the more radical changes, the *BM* still manages to get a steady performance over time, being the best solution.

5 Conclusions and Future Work

The results obtained do support the idea that the combination of evolution and learning makes itself a very interesting approach, when facing changing environments, that constitute most of the real-world applications. Referring to the *LM* vs *BM* debate, this work supports the belief that the latter reveals greater robustness facing changes. One explanation may be that the *BM* is evolving not a solution to a given task, but instead structures with a good learning capability.

In the future it is intended to enlarge the experiments by tackling real world domains, namely *intensive care units*, *time-series forecasting* or *bioengineering*.

References

[1] J.M. Baldwin. A New Factor in Evolution. *American Naturalist*, (30):441–451, 1896.

[2] P. Cortez, M. Rocha, and J. Neves. A Lamarckian Approach for Neural Network Training. *Neural Processing Letters*, 15(2):105–116, 2002.

[3] L. J. Fogel. *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*. John Wiley, New York, 1999.

[4] M. Riedmiller. Supervised Learning in Multilayer Perceptrons - from Backpropagation to Adaptive Learning Techniques. *Computer Standards and Interfaces*, 16, 1994.

[5] T. Sasaki and M. Tokoro. Adaptation toward Changing Environments: Why Darwinian in Nature? In *Proceedings of the Fourth European Conference on Artificial Life*, 1997.

[6] T.Iba and Y.Takefuji. Adaptation of neural agents in dynamic environments. In L. C. Jain and L. K. Jain, editors, *2nd International Conference on Knowledge-Based Intelligent Electronic Systems*. IEEE Press, 1998.

[7] P. Turney. Myths and Legends of the Baldwin Effect. 13th International Conference on Machine Learning (ICML96), Workshop on Evolutionary Computation and Machine Learning, Bari, Italy, July, 135-142, 1996.

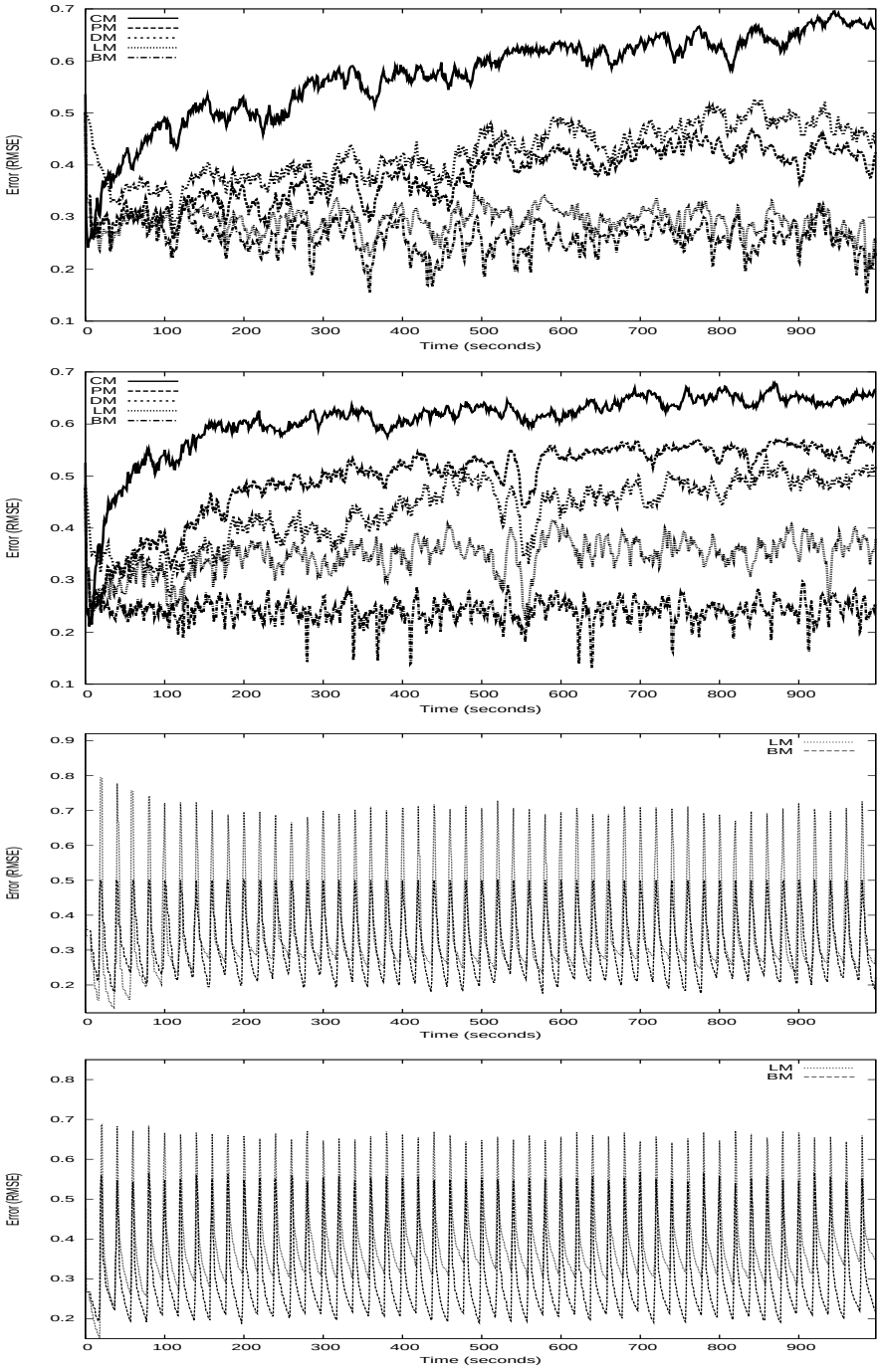

Figure 3: Results for the experiments (NBC_1, TCC_1, NBC_2 and TCC_2)