Inteligencia Artificial 17(54), 30-34
doi: 10.4114/ia.v17i54.1081

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

New training approaches for classification based on evolutionary neural networks. Application to product and sigmoidal units

Antonio J. Tallón-Ballesteros

Department of Languages and Computer Systems, University of Seville, Spain

atallon@us.es

Abstract This paper sums up the main contributions of the PhD Dissertation with an homonymous name to the current article. Specifically, three contributions to train feed-forward neural network models based on evolutionary computation for a classification task are described. The new methodologies have been evaluated in three-layered neural models, including one input, one hidden and one output layer. Particularly, two kind of neurons such as product and sigmoidal units have been considered in an independent fashion for the hidden layer. Experiments have been carried out in a good number of problems, including three complex real-world problems, and the overall assessment of the new algorithms is very outstanding. Statistical tests shed light on that significant improvements were achieved. The applicability of the proposals is wide in the sense that can be extended to any kind of hidden neuron, either to other kind of problems like regression or even optimization with special emphasis in the two first approaches.

Keywords: Artificial neural networks, evolutionary algorithm, classification, product unit, sigmoidal unit, feature selection.

1 Introduction

Learning algorithms can be grouped into two categories: a) black-box methods, such as neural networks or Bayesian classifiers and b) knowledge-oriented methods, like the models created by decision trees, association or decision rules. Data Mining techniques are sensitive to the information quality on that the knowledge discovery will be carried out. The higher data quality, the higher quality decision-making models.

The goal of this paper is to get more accurate neural networks models with a greater efficiency that in previous evolutionary approaches.

The rest of this article is organized as follows: Section 2 describes the contributions; Section 3 details the final remarks; finally, Appendix A overviews the data sets used for the experimentation.

2 Contributions

The three proposed contributions are detailed in the next subsections. A deep overview of our proposals can be read in [8] along with all the results and statistical tests.

ISSN: 1988-3064(on-line)

The training of the neural network (NN) models is performed by means of a baseline evolutionary algorithm (EA) that simultaneously evolves the weights and the architecture of the NN. This EA has some common points with the approaches leaded mainly by X. Yao [13] and P. J. Angeline [1] and with a more recent work derived from that two research papers written by K. O. Stanley and R. Miikkulainen [7]. The interested reader can find the EA in Section 2.2. of [10], including the pseudo-code, their foundations and related literature. Initially, the EA was proposed [5] in the context of product-unit neural network (PUNN) models [2]. Taking as starting point the aforementioned EA, some novelty methodologies are presented for the training of neural networks with special attention to classification models containing product or sigmoidal neurons in the hidden layer. In the next subsection the EA is briefly explained due to the fact that is utilised in our three approaches. Moreover, in the subsequent subsections we will describe the hot points for each contribution.

2.1 Baseline Evolutionary Algorithm

The EA utilise two kind of operators: replication and mutation. According to the related works, crossover operator may not be convenient due to the permutation problem [4]. Two types of mutation has been used: the parametric and structural ones. The former changes the value of the model coefficients and is performed via a simulated annealing algorithm. The severity of a mutation of an individual in the population is dictated by the current temperature. The latter carries out a modification in the model structure and allows different regions in the search space to be explored while helping to maintain the diversity of the population; there are five different structural mutations, the first four ones are similar to those in the GNARL (GeNeralized Acquisition of Recurrent Links) model [1]: node addition, node deletion, connection addition, connection deletion and node fusion. All the above mutations are made sequentially in the given order, with a variable probability, in the same generation on the same network. If probability does not select a mutation, one of the mutations is chosen at random and applied to the network. The macrosteps of the EA are:

1. Random initialisation. At the beginning, the EA generates a number of individuals equal to ten times the population size with a pseudo-random number generator. Next, all individuals are assessed, sorted by decreasing fitness and a tenth part will compose the initial population.

2. Evolutionary process. It is the core of the algorithm and is divided into three tasks:

    - Replacement. The tenth best and worst portions of population individuals are swapped and the best one part goes to the next generation. The remaining individuals (nine parts out of the full population), that could be called as the intermediate population, are subjected to mutation.
    - Mutation. It is applied to the intermediate population. A ten percent of it is undergone to parametric mutation and the remainder ninety percent goes through structural mutation.
    - Stop condition. At the end of every generation the halt criterium is checked. The evolutionary process is repeated until the maximum number of generations is reached or until the best individual or the population mean fitness do not improve during twenty generations.

The sketch of the EA is completed with the description of the topology and the error function. We keep the original architecture published in [5], that is a feed-forward three-layer $k : m : j$ topology, with k nodes in the input layer, m ones and a bias one in the hidden layer and j nodes in the output layer. The transfer function of each node in the hidden and output layers is the identity function. We have considered a standard soft-max activation function, associated with each output node of the R network model, given by:

$$R_j(\mathbf{x}) = \frac{exp\ f_j(\mathbf{x})}{\sum_{j=1}^{J} exp\ f_j(\mathbf{x})} \quad j = 1, ..., J \tag{1}$$

where J is the number of classes in the problem, $f_j(\mathbf{x})$ is the output of node j for pattern $\mathbf{x}$ and $R_j(\mathbf{x})$ is the probability that this pattern belongs to class j. Given a training set $D = (x_i, yi)\ i = 1, ..., N$, a

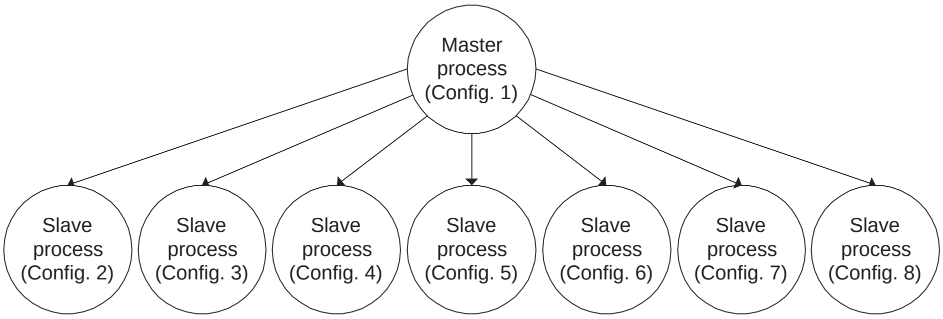
Master process (Config. 1)
Slave process (Config. 2)
Slave process (Config. 3)
Slave process (Config. 4)
Slave process (Config. 5)
Slave process (Config. 6)
Slave process (Config. 7)
Slave process (Config. 8)

Figure 1: EDD model

function of cross-entropy error is used to evaluate a network R with the samples of a problem, which is reflected in the following expression:

$$l(R) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J}(y_i^j ln(R_j(\mathbf{x}_i))) \tag{2}$$

and substituting R_j defined in (1),

$$l(R) = -\frac{1}{N}\sum_{i=1}^{N}\left(-\sum_{j=1}^{J} y_i^j f_j(\mathbf{x_i}) + ln(\sum_{j=1}^{J} exp\ f_j(\mathbf{x}_i))\right) \tag{3}$$

where y_i^j is the target value for class j with pattern $\mathbf{x_i}$ ($y_i^j = 1$ if $\mathbf{x_i} \in$ class j and $y_i^j = 0$ otherwise), $f_j(\mathbf{x_i})$ is the output value of the neural network for the output neuron j with pattern $\mathbf{x_i}$. Observe that soft-max transformation produces probabilities that sum to one and therefore the outputs can be interpreted as the conditional probability of class membership. On the other hand, the probability for one of the classes does not need to be estimated because of the normalization condition. Usually, one activation function is set to zero; then $f_J(\mathbf{x}_i) = 0$ and we reduce the number of parameters to estimate. Thus, the number of nodes in the output layer is equal to the number of classes minus one in the problem.

2.2 Experimental Design Distribution (EDD)

In this first proposal [9], some parameters regarding either the topology of the PUNN model or the EA are distributed throughout all the processing elements of the computation system. More concretely, these parameters are the maximum number of neurons in the hidden layer (neu), the maximum number of generations (gen) and the α_2 (associated with the parametrical mutation) parameter value. Recently, this contribution has been extended to consider sigmoidal neurons in the hidden layer and the resultant model has been named Experimental Design Distribution with Sigmoidal units (EDDSig) [11].

This approach was implemented following a master-slave model where the master process fixs a base configuration that is distributed to the slaves which update the received configuration changing a parameter value. Once all the modifications have been made each node, including the master and the slave processes, will execute the new configuration. Figure 1 depicts the structure of the EDD model.

This model was tested in thirteen problems (eleven from the University of California at Irvine and two real-world complex problems) and the results ([9] and [11]) show that the product units are slightly more accurate than the sigmoidal units.

2.3 Two-Stage Evolutionary Algorithm (TSEA)

The second contribution diversifies the neural network architecture and is composed of two stages. During the first stage, two populations with different properties are created and evolved for a small number of generations. Next, the half best of each population are merged in a new population. In the second stage, the new population is evolved for a full evolutionary cycle. More details about this approach are provided in [10]. Also, the algorithm is available upon request. The natural extension to sigmoidal units has been called Two-Stage Evolutionary Algorithm for neural networks with Sigmoidal units (TSEASig) [11].

This proposal changes the initialisation step of the baseline EA considering two independent populations each one with a configuration with a different number of hidden neurons that are evolved during a tenth part of a complete evolution. After that, the two best halves are mixed and are undergone to a typical evolutionary process.

The results ([10] and [11]) shed light on that TSEA (or TSEASig) is significantly better than EDD (or EDDSig). The experimentation was carried out in fourteen binary and multi-class data sets with a number of patterns between 63 and 5000 and the number of inputs in the range [3, 83].

2.4 Two-Stage Evolutionary Algorithm with Feature Selection (TSEAFS)

This third proposal [12] involves a data preprocessing –applying feature selection methods implemented as filters– on the data set in order to improve the efficacy and the simplicity of the obtained models. Some feature selectors are independently applied to the training set of the problem at hand getting a list of attributes, that will be used to obtain the reduced training and test sets for the learning of the PUNN models.

On this model, the idea is to act on the raw training data to select relevant features. The list of characteristics that is collected after the feature selection step is projected onto the test set. The reduced sets are taken as input to the TSEA approach. In relation with the methods for data preparation, we utilised feature subsets selectors which main property is that the output result is a feature set instead of an ordered list of attributes.

TSEAFS was applied to nineteen complex problems, including one real-world data set related with liver transplantation in Spain, with a number of inputs between 7 and 114. According to the results, excluding those of the real-world problem, [12], it could be asserted that all the considered filters reached significantly better results than TSEA. Deepening a bit more, we could conclude that the confidence level is greater for the methods based on a correlation measure [3], followed by a filter based on a consitency metric (CNS, [6]). The best feature selector for the liver transplantation problem was CNS showing a better performance that some classifiers of the state-of-the-art such as the powerful Multilayer Perceptron (MLP).

3 Final remarks

The experimentation was conducted using a great deal of classification data sets (refer to Appendix A). The achieved conclusions are as follows. First, EDD is more accurate than EDDSig, however the latter is faster than the former. Second, TSEA has a significant better efficacy and is about a fourty percent faster than EDD. Third, TSEASig has significant greater accuracy than EDDSig. Fourth, TSEAFS obtains simpler models with a dimensionality reduction greater than a fifty percent compared with TSEA.

Acknowledgements

This work has been partially subsidized by TIN2007-68084-C02-02, TIN2008-06681-C06-03 and TIN2011-28956-C02-02 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds, P08-TIC-3745 and P11-TIC-7528 projects of the ”Junta de Andalucía” (Spain).

References

[1] P. J. Angeline, G. M. Saunders, and J. B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65, 1994. doi: 10.1109/72.265960.

[2] R. Durbin and D. Rumelhart. Products units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1(1):133–142, 1989. doi: 10.1162/neco.1989.1.1.133.

[3] M.A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the seventeenth International Conference on Machine Learning(ICML 2000)*, pages 359–366, San Francisco, CA, 2000. Morgan Kaufmann.

[4] P. J. B. Hancock. Genetic algorithms and permutation problems: A comparison of recombination operators for neural net structure specification. In *Combinations of Genetic Algorithms and Neural Networks, 1992., COGANN-92. International Workshop on*, pages 108–122. IEEE, 1992.

[5] C. Hervás, F. J. Martínez, and P. A. Gutiérrez. Classification by means of evolutionary product-unit neural networks. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1525–1532. IEEE, 2006.

[6] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *Proceedings of the thirteenth International Conference on Machine Learning(ICML 1996)*, pages 319–327, Italy, 1996. Morgan Kaufmann.

[7] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002. doi: 10.1162/106365602320169811.

[8] A.J Tallón-Ballesteros. *Nuevos modelos de Redes Neuronales Evolutivas para Clasificación. Aplicación a Unidades Producto y Unidades Sigmoide.* PhD thesis, University of Seville (Spain), 2013. Available at: http://fondosdigitales.us.es/tesis/tesis/1990/nuevos-modelos-de-redes-neuronales-evolutivas-para-clasificacion-aplicacion-unidades-producto-y-unidades-sigmoide/.

[9] A.J. Tallón-Ballesteros, P.A. Gutiérrez-Peña, and C. Hervás-Martínez. Distribution of the search of evolutionary product unit neural networks for classification. *arXiv preprint arXiv:1205.3336*, 2012.

[10] A.J. Tallón-Ballesteros and C. Hervás-Martínez. A two-stage algorithm in evolutionary product unit neural networks for classification. *Expert Systems with Applications*, 38(1):743–754, 2011. doi: 10.1016/j.eswa.2010.07.028.

[11] A.J. Tallón-Ballesteros, C. Hervás-Martínez, and P.A. Gutiérrez. An extended approach of a two-stage evolutionary algorithm in artificial neural networks for multiclassification tasks. In *Innovations in Intelligent Machines-3*, pages 139–153. Springer, 2013. doi: 10.1007/978-3-642-32177-1_9.

[12] A.J. Tallón-Ballesteros, C. Hervás-Martínez, J.C. Riquelme, and R. Ruiz. Feature selection to enhance a two-stage evolutionary algorithm in product unit neural networks for complex classification problems. *Neurocomputing*, 114:107–117, 2013. doi: 10.1016/j.neucom.2012.08.041.

[13] X. Yao and Y. Liu. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 8(3):694–713, 1997. doi: 10.1109/72.572107.

Appendix

A Data sets

Appendicitis, Australian credit approval, Balance, Breast (Cancer, Tissue and Wisconsin), Cardiotocography, Heart (Statlog and disease Cleveland), Hepatitis, Horse colic, Thyroid disease (Hypothyroid and Newthyroid), Ionosphere, Labor Relations, Led24, Liver disorders, Lymphography, Parkinsons, Pima Indians diabetes, Steel Plates Faults, Molecular Biology, SPECTF, Vowel, Waveform, Wine Quality, Yeast, BTX, Listeria monocytogenes and Liver-transplantation problems were used in the experiments.