18th International Workshop on Database and Expert Systems Applications

Dermatology Disease Classification via Novel Evolutionary Artificial Neural Network

Antonia Azzini and Stefania Marrara
University of Milan
Information Technology Department
via Bramante, 65 – 26013 Crema, Italy

Abstract

Neuro-genetic systems are biologically inspired computational models that use evolutionary algorithms (EAs) in conjunction with neural networks (NNs) to solve problems. They are especially useful in classification problems in which classifier systems are not able to provide easy answers. In this paper a novel neuro-genetic approach is used in order to predict a known classification problem, related to dermatology diseases.

1. Introduction

Evolutionary algorithms represent a more integrated and rational way of designing Artificial Neural Networks (ANNs) as classifier systems, and are especially useful for complex optimization problems where the number of parameters is large and the analytical solutions are difficult to obtain, without requiring expert knowledge of the problem.

In this work a multi-classifier system has been presented, based on the evolution of artificial neural networks, that are useful in those classification problems in which classifier systems are not able to provide easy answers. The neuro-genetic algorithm used in this setting has been previously presented in [2, 1]. The novelty in this paper is that we carry out classification in presence of white noise in the dataset and implement a novel fitness function in the evolutionary process. This improves the general approach since it reduces the algorithm parameters and it is able to classify data affected by noise without reducing the algorithm performances and obtaining better results then other approaches used in literature to face the same biological problem [5].

The aim of the dermatology classification problem considered in this work is to determine the type of eryhemato-squamous diseases when 34 clinical and histopatological features are defined as inputs.

2. Dermatology Classification Problem

The differential diagnosis of the dermatology problem related to the well-known erythemato-squamous diseases is a difficult problem in medicine, since all these diseases share the clinical features of erythema and scaling, with very little differences. The diseases regard psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. Usually a biopsy is necessary for the diagnosis but unfortunately these diseases share many histopathological features as well. Another difficulty for the differential diagnosis is that a disease may show the features of another disease at the beginning stage and may have the characteristic features at the following stages.

Several works have been carried out in the literature [3, 5, 6] in order to define classifier systems able to solve this problem. In this work we present a novel neuro-genetic approach, that uses evolutionary algorithms to optimize a particular kind of neural networks, that would work as multi-classifier systems in this dermatological application.

2.1 Neuro-Genetic Algorithm

The primary aim of the neuro-genetic approach considered in this work concerns the optimization of neural network design, basing on the simultaneous evolution of network weights and topology.

The neuro-genetic approach considered in this application has been presented in the literature and it has been validated on different benchmarks and real-world problems [2, 1]. Our approach takes advantage of the Backpropagation (BP) as local search optimization algorithm. The idea is to exploit the ability of the EA to find a solution close enough to the global optimum, together with the ability of the BP algorithm to finely tune a solution and reach the nearest local minimum. In the overall evolutionary process BP is used to decode a *genotype* into a *phenotype* NN. Accordingly, it is the genotype which undergoes the genetic

1529-4188/07 $25.00 © 2007 IEEE
DOI 10.1109/DEXA.2007.71

operators and which reproduces itself, whereas the phenotype is used *only* for calculating the genotype's fitness.

The overall evolutionary process can be described by the following pseudo-code:

1. Initialize the population by generating new random individuals.
2. Create for each genotype the corresponding MLP, and calculate its fitness value.
3. Save the best individual as the best-so-far individual.
4. While not termination condition do
   (a) Apply the genetic operators to each network.
   (b) Decode each new genotype into the corresponding network.
   (c) Compute the fitness value for each network.
   (d) Save statistics.

Then, in the evolutionary process, the genetic core implemented is applied to each neural network by the following pseudo-code:

1. Select from the population (of size n) $\lfloor n/2 \rfloor$ individuals by truncation and create a new population of size n with copies of the selected individuals.
2. For all individuals in the population:
   (a) Perform crossover with probability p_{cross}.
   (b) Mutate the topology and the weights of the offspring.
   (c) Train the resulting network.
   (d) Calculate the training fitness f and the test fitness $\hat{f}$.
   (e) Save the individual with lowest $\hat{f}$ as the best-so-far individual if the $\hat{f}$ of the previously saved best-so-far individual is higher (worse).
3. Save statistics.

2.2 Initialization

Our evolutionary approach considers Multi Layer Perceptron, (MLPs), for individual encoding. They are a kind of feedforward NNs with a layer of input neurons, a layer of one or more output neurons and zero or more 'hidden' (i.e., internal) layers of neurons in between; neurons in a layer can take inputs from the previous layer only.

When a new population is generated, the networks corresponding to the genotype will be initialized with different hidden layer sizes, using two exponential distributions to define the number of hidden layers and the corresponding number of neurons for each layer. This process is carried out by the following pseudo-code:

$NumLayers = \|-h * ln(rand(1))\|$
if $NumLayers = 0$
  $NumLayers = 1$
$k = ln(Ninput/Noutput)/NumLayers$
for $i = 1$ **to** $NumLayers$ **do**
  $MeanNeurons = Ninput\, e^{(-ki)} - 1$
  $Phenotype(i) = \|-MeanNeurons * ln(rand(1))\| + 1$

where $NumLayers$ corresponds to the number of hidden layers, and h is the mean for the exponential distribution used for the hidden layer definition. k is the parameter used to define $MeanNeurons$, corresponding to the mean value of neurons set in each layer i, and which are defined into the corresponding array $Phenotype(i)$. Finally, $Ninput$ and $Noutput$ refer, respectively, to the number of input and output neurons. They are two of the algorithm parameters, set at first time during the initialization and maintained constant for all individuals during the entire simulation.

Then we use a normal distribution to determine the weights and bias values for each individual at the initialization step. Variance matrices are also defined for all weights and bias matrices, that will be applied in conjunction with evolutionary strategies in order to perturb network weights and bias. Variance matrices are initialized with matrices of all ones.

Unlike other approaches [7], in the neuro-genetic algorithm the maximum size and the number of the hidden layers are not determined in advance, nor bounded, even though the evolution may penalize large networks.

Table 1 lists all the parameters of the algorithm, and specifies the default values that they assume in this work at the initialization step.

Table 1. Parameters of the Algorithm.

Symbol	Meaning	Default Value
n	Population size	60
seed	Previously saved population	none
bp	Backpropagation selection	1
p^{+}_{layer}	Probability to insert a hidden layer	*
p_{layer}	Probability to delete a hidden layer	*
p^{+}_{neuron}	Probability to insert a neuron in a hidden layer	*
p_{cross}	Probability to crossover	0
r	Parameter for use in weight mutation for neuron elimination	1.5
h	Mean for the exponential distribution	2

*) Simulations with several settings.

Note that the use of the bp parameter set to 1 allows to employ *indirect encoding* of networks, where the phenotype is obtained by the training of an initial (embryonic) network using BP.

Each individual is then encoded in a structure in which basic information are maintained as illustrated in Table 2. During the entire evolution, the dimension, on average, of each individual is about 20 Kbytes.

The values of all these parameters are affected by the genetic operators during evolution, in order to perform incre-

Table 2. Individual Representation.

Element	Description
l	Length of the topology string, corresponding to the number of layers.
topology	String of integer values that represent the number of neurons in each layer.
$\mathbf{W}^{(0)}$	Weights matrix of the input layer neurons of the network.
$\mathbf{Var}^{(0)}$	Variance matrix of the input layer neurons of the network.
$\mathbf{W}^{(i)}$	Weights matrix for the ith layer, $i = 1, \ldots, l$.
$\mathbf{Var}^{(i)}$	Variance matrix for the ith layer, $i = 1, \ldots, l$.
b_{ij}	Bias of the jth neuron in the ith layer.
$Var(b_{ij})$	Variance of the bias of the jth neuron in the ith layer.

mental (adding hidden neurons or hidden layers) and decremental (pruning hidden neurons or hidden layers) learning.

2.3 Fitness Function

Although it is customary in EAs to assume that better individuals have higher fitness, in this case we adopt the convention that a lower fitness means a better NN. This maps directly to the objective function of our problem, which is a fitness minimization problem.

A novel aspect of this approach is that, despite other previous approaches [2, 1], the fitness of an individual is not given by the overall cost and the mean square error (mse) of each neural network, but based on the confusion matrix, in order to decrease the algorithm parameters and consequently reduce the fitness function complexity.

Indeed, in this approach the fitness function is defined as:

$$f = Noutput - Trace \tag{1}$$

where $Noutput$ has been already set to the number of possible classes of membership, $Trace$ corresponds to the sum of the diagonal elements of the normalized confusion matrix. Generally, a confusion matrix is defined as a visualization tool typically used in supervised learning, and contains information about actual and predicted classifications done by a classification system. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The performance of such systems is commonly evaluated using the data in the matrix, and the ideal results are obtained when all elements of the confusion matrix belong to the diagonal of that matrix. This matrix represents the conditional probability between the predicted outputs obtained from the neuro-genetic approach, and the desired outputs, defined as target values in the database. In this work we consider, for the $Trace$ computation, the normalized confusion matrix, since the normalization process ables us to solve many of the problems related to the classification of unbalanced datasets, as in this application.

One of the main advantages of using this fitness is related to the remarkable reduction of algorithm parameters, that were necessary, during the evolutionary process, for network cost and accuracy definition. Furthermore, this kind of fitness also penalizes large networks in a satisfactory way, avoiding to check the mse for each neural network.

In this approach the $Trace$ value is calculated for each individual during the overall evolutionary process and shows the goodness of such neural network as a multi-classifier system. The best individual will have the lowest fitness value (since we have defined our as a fitness minimization problem), and the corresponding normalized confusion matrix would be near as soon as possible to an Identity matrix.

In the neuro-genetic approach two fitness values are actually calculated for each individual: the fitness f, used by the selection operator, and a test fitness $\hat{f}$.

Then, following the commonly accepted practice of machine learning, the problem data are partitioned into three sets: training set, used to train the network, test set, used to decide when to stop the training and avoid overfitting, and finally, a validation set, to test the generalization capabilities of a network.

Now, $\hat{f}$ is calculated according to Equation 1 by using the $Trace$ value over the test set. When BP is used, i.e., if $bp = 1$, $f = \hat{f}$; otherwise ($bp = 0$), f is calculated according to Equation 1 by using the $Trace$ over the training and test sets together.

2.4 Genetic Operators

The evolutionary process is based on the genetic operators of selection and mutation. Here they are briefly presented, while a detailed description of such operators is reported in the literature [2, 1].

- *Selection* The selection method implemented in this work follows the breeder genetic algorithm [1]. The selection strategy used by the algorithm is truncation, in which, starting from a population of n individuals, the worst $\lfloor n/2 \rfloor$ (with respect to f) are eliminated. The remaining individuals are then duplicated in order to replace those eliminated, and finally, the population is randomly permuted.

- *Mutation* The two mutation operators implemented in the genetic core refer to *Weight Mutation*, that is applied before the BP learning rule and perturbs the weights and biases of the network by using variance matrices and evolutionary strategies applied to the number of synapses of the entire network. During the evolutionary process, after this perturbation has been used, neurons whose contribution to the network outputs is negligible are eliminated from the corresponding structure. *Topology Mutation* considers three mutation operators that affect the network architecture

(i.e., the number of neurons in each layer and the number of hidden layers), with three different independent probabilities, respectively, p^{+}_{layer}, p^{-}_{layer}, and p^{+}_{neuron}, that refer to the insertion and the elimination of one hidden layer, and the insertion of one neuron. The case of neuron elimination is considered, as indicated in Weight Mutation, like a contribution-dependent probability.

3 An Application to erythemato-squamous disease

As previously reported in Sect. 2, the differential diagnosis of erythemato-squamous diseases is a difficult problem in dermatology, since such diseases all share the clinical features of erythema and scale with very few differences. Furthermore a disease may show the histopathological features of another disease at the beginning stage and may have the characteristic features at the following stages.

For this reason, we implement the neuro-genetic approach in order to carry out, without any human expert intervention, this diseases classification problem. The evolutionary neural networks considered in this approach, the MLPs, will be defined, as previously reported in Sect. 2.2, with 34 neurons in the input layer and 6 neurons in the output layer. During the evolutionary process, the output, i.e. the class of membership, obtained from each neural network will correspond to the highest activation value of the 6 output neurons of the network.

3.1 Dataset

All data included in the training, validation and test sets are acquired from the UCI Machine Learning Repository [4]. It is a repository of databases and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. In this repository, different and several benchmark problems are collected and all the corresponding data can be fully downloaded. As previously described, the dataset has been divided into the three datasets corresponding to training, test and validation set, with respectively, 200, 100 and 66 cases. For each instance the first 34 attributes are considered as input to the neural networks, containing 12 clinical features and 22 histopathological features, while the following 6 values refer to the target output. Samples with target outputs psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris are given the binary target values of (1,0,0,0,0,0), (0,1,0,0,0,0), (0,0,1,0,0,0), (0,0,0,1,0,0), (0,0,0,0,1,0), and (0,0,0,0,0,1), respectively.

Furthermore, in order to study the capabilities of our approach in presence of white noise in the dataset, some missing data are also added into the three datasets, by randomly setting some input feature values to -1.

3.2 Experimental Results

In this application some experiments have been carried out in order to find the evolutionary MLP that works as good classifier. All the parameters of the algorithm are set to the default values shown in Table 1, and several runs of this approach have been carried out in order to find out optimal settings of the genetic parameters p^{+}_{layer}, p^{-}_{layer}, and p^{+}_{neuron}. For each run of the evolutionary algorithm, up to 200000 network evaluations (i.e., simulations of the network on the whole training set) have been allowed, including those performed by the backpropagation algorithm, with an average computational time of about 15 minutes. The results obtained are presented in Table 3: here are reported data about the average and the standard deviation of the test fitness values about the best solutions found for each parameter settings over 10 runs.

Table 3. Dermatology Disease Classification Experimental Results.

Setting	Parameter Setting			BP=1	
	p^{+}_{layer}	p^{-}_{layer}	p^{+}_{neuron}	avg	stdev
1	0.05	0.05	0.05	0.4155	0.0627
2	0.05	0.05	0.1	0.3642	0.0642
3	0.05	0.05	0.2	0.3824	0.0982
4	0.05	0.1	0.05	0.3922	0.0554
5	0.05	0.1	0.1	0.4332	0.0911
6	0.05	0.1	0.2	0.4266	0.0722
7	0.05	0.2	0.05	0.4004	0.1163
8	0.05	0.2	0.1	0.4456	0.0716
9	0.05	0.2	0.2	0.3822	0.0814
10	0.1	0.05	0.05	0.3864	0.1208
11	0.1	0.05	0.1	0.4134	0.0955
12	0.1	0.05	0.2	0.4075	0.2813
13	0.1	0.1	0.05	0.4108	0.0650
14	0.1	0.1	0.1	0.4355	0.0770
15	0.1	0.1	0.2	0.3994	0.0632
16	0.1	0.2	0.05	0.4108	0.0650
17	0.1	0.2	0.1	0.4355	0.0771
18	0.1	0.2	0.2	0.4135	0.0453
19	0.2	0.05	0.05	0.40337	0.0832
20	0.2	0.05	0.1	0.3775	0.0981
21	0.2	0.05	0.2	0.3810	0.0968
22	0.2	0.1	0.05	0.4097	0.0762
23	0.2	0.1	0.1	0.3797	0.0638
24	0.2	0.1	0.2	0.3293	0.0806
25	0.2	0.2	0.05	0.3871	0.0560
26	0.2	0.2	0.1	0.3631	0.0481
27	0.2	0.2	0.2	0.3726	0.0656

The best solutions, on average, have been found with $p^{+}_{\text{layer}} = 0.2$, $p^{-}_{\text{layer}} = 0.1$, and $p^{+}_{\text{neuron}} = 0.2$. The best model is a multi-layer perceptron with a phenotype of type [5,6], which obtained a $Trace$ value equal to 0.1736 for the test set and to 0.19643 for the validation set. The performance and the accuracy obtained with the best solution found are defined by the statistical sensitivity parameter cal-

culated on the validation set, explained in Table 4, that correspond to the percentage of the number of true positive decisions over the number of actually positive cases. A graphical representation of the best classification results is also shown in Figure 1. This figure outlines how most of the results are concentrated on the diagonal of the normalized confusion matrix and therefore our algorithm performs particularly well.

in Table 4, we can see how our approach generally performs better than the approaches reported in recent literature [5, 6]. Even if this neuro-genetic approach considers the same dataset used in these other approaches, no a direct comparison is showed in this section, since white noises and different distributions for train, test and validation sets are used in this approach.

However, in order to provide a more satisfactory idea about results obtained in this evolutionary approach, we also report the sensitivity values obtained from the experiments presented in [5]. In particular, in that work, the authors showed that the sensitivity values for the six erythemato-squamous diseases were equal respectively to 96.4% for psoriasis, 96.7% for seboreic dermatitis, 97.1% for lichen planus, 91.7% for pityriasis rosea, 95.8% for chronic dermatitis, and 90.0% for pityriasis rubra pilaris disease.

Table 4. Sensitivity of the best solution found with data affected by white noise.

Setting	Sensitivity (%)
Psoriasis	100
Seboreic dermatitis	100
Lichen planus	100
Pityriasis rosea	92.8
Chronic dermatitis	87.5
Pityriasis rubra pilaris	100

4. Conclusions

The work described in this paper demonstrates an approach to the joint optimization of neural networks weights and topology which takes advantage of both evolutionary algorithms and the backpropagation algorithm. Its effectiveness has been validated by applying this approach to the well-known erythemato-squamous diseases classification problem. The experiments show very satisfactory results obtained from the best MLP neural network identified by the evolutionary algorithm, together with a comparison with other approaches previously discussed in the literature.

5 Acknowledgments

This work was partly funded by the Italian Ministry of Research under FIRB contract n. RBNE05FKZ2_004, TEKNE. The authors also would like to thank Prof. Andrea G.B. Tettamanzi and Prof. Ernesto Damiani for their precious suggestions.

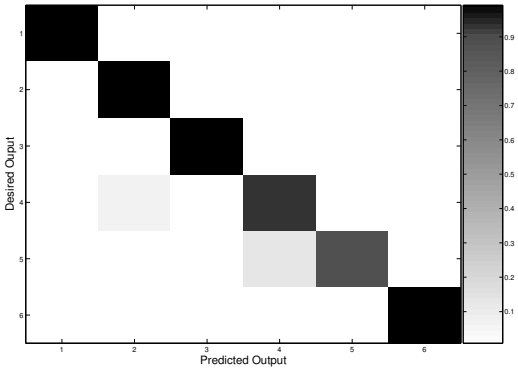

Figure 1. Best Classification Results on Validation Set.

References

[1] A. Azzini and A. Tettamanzi. A neural evolutionary approach to financial modeling. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'06*, volume 2, pages 1605–1612. Morgan Kaufmann, San Francisco, CA, 2006.

[2] A. Azzini and A. Tettamanzi. A neural evolutionary classification method for brain-wave analysis. In *Proceedings of the European Workshop on Evolutionary Computation in Image Analysis and Signal Processing, EVOIASP 2006*, volume 3907, pages 500–504. Springer, Berlin, 2006.

[3] H. A. Guvenir, G. Demiroz and N. Ilter. Learning differential diagnosis of eryhemato-squamous diseases using voting feature intervals. *Aritificial Intelligence in Medicine*, 13:147–165, 1998.

[4] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[5] E. Ubeyli and I. Guler. Automatic detection of erythemato-squamous diseases using adaptive neuro-fuzzy inference systems. *Computers in Biology and Medicine*, 35:421–433, 2005.

[6] L. Nanni. An ensemble of classifiers for the diagnosis of erythemato-squamous diseases. *Neurocomputing, Letters*, 842–845, 2006.

[7] X. Yao and Y.Liu. Towards designing artificial neural networks by evolution. *Applied Mathematics and Computation*, 91(1):83–90, 1998.