Constructive Neural Networks to Predict Breast Cancer Outcome by Using Gene Expression Profiles

Daniel Urda, José Luis Subirats, Leo Franco, and José Manuel Jerez

Departament of computer science,
University of Málaga. Spain
{durda,jlsubirats,lfranco,jja}@lcc.uma.es
http://www.lcc.uma.es

Abstract. Gene expression profiling strategies have attracted considerable interest from biologist due to the potential for high throughput analysis of hundreds of thousands of gene transcripts. Methods using artifical neural networks (ANNs) were developed to identify an optimal subset of predictive gene transcripts from highly dimensional microarray data. The problematic of using a stepwise forward selection ANN method is that it needs many different parameters depending on the complexity of the problem and choosing the proper neural network architecture for a given classification problem is not a trivial problem. A novel constructive neural networks algorithm (CMantec) is applied in order to predict estrogen receptor status by using data from microarrays experiments. The obtained results show that CMantec model clearly outperforms the ANN model both in process execution time as in the final prognosis accuracy. Therefore, CMantec appears as a powerful tool to identify gene signatures that predict the ER status for a given patient.

Keywords: Artificial neural networks; Constructive neural networks; Predictive modelling; Gene expression profiles; Breast cancer.

1 Introduction

Gene expression profiling strategies have been widely used in cancer studies for prediction of disease outcome [24,22] by using data from microarrays experiments. Microarrays technology is a powerful platform successfully used for the analysis of gene expression in a wide variety of experimental studies [13]. Nevertheless, this platform provides data that are in origin vulnerable to the curse of dimensionality (each sample is characterized by several thousand gene transcripts) and the curse of data set sparsity (only few examples are available for the analysis). In this sense, the estimation of prognosis models by using gene expression profiling requires the application of an additional procedure to select the most significant gene transcripts.

Methods using artifical neural networks (ANNs) were developed to identify an optimal subset of predictive gene transcripts from highly dimensional microarray data [22,14]. Recently, Lancashire et al. [7] proposed a stepwise forward

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 317–326, 2010.

selection artificial neural network approach, in which architecture size evolves sequentially by selecting and adding input neurons to the network, in order to identify an optimum gene subset based on predictive performance. This method was applied to a gene microarray dataset to identify and validate gene signatures corresponding with estrogen receptor in breast cancer.

Nevertheless, in spite of the demonstrated efficiency of ANNs in prediction and classification problems in general, the estimation of prognosis models by using stepwise forward selection procedures has, on one hand, a high computational cost, and on the other hand, a high complexity related to the design of the optimal architecture size for the neural network models. Several approaches [16,2,8,5] were proposed in this sense to solve, or alleviate, the problem of choosing the proper neural network architecture for a given problem. However, there is no general agreement on the strategy to follow in order to select an optimal neural network architecture, and the computationally inefficient "trial and error" method is still much used in applications using ANNs.

Constructive neural networks (CNN) algorithms constitute an alternative in designing artificial neural networks models [9,3,6,1,11,21,10,12,19]. This family of algorithms generate the network topology online by adding neurons during the training phase. This work presents the application of a novel constructive algorithm (CMantec) that incorporates competition and cooperation among neurons to obtain smaller in size architectures and better generalization capabilities, and analyzes its application to the problem of predicting estrogen receptor status by using data from microarrays experiments.

2 Materials and Methods

2.1 Materials

The dataset that has been used in this work was downloaded from[1] and consists of 49 samples each with 7129 corresponding variables specifying the intensity of the probe sets targeting each transcript. These data were previosly published by West et al [23]. This used microarray technology to analyse primary breast tumours in relation to estrogen receptor (ER) status. The dataset consists of 25 patients with ER+ and 24 with ER-.

2.2 Methods

For the ANN analysis, the code was written using the neural network toolbox of MATLAB and was launched on a computational cluster with 30 nodes. The CMantec algorithm was developed in Visual Studio 2008 and was launched on a personal computer. In order to compare CPU time of these algorithms, both of them were launched on the same single computer. In section 3 the reader could find time results for these algorithms.

[1] `http://data.cgt.duke.edu/west.php`

Model estimation procedure. The learning process of the stepwide method proposed by [7] and shown in Fig. 1 begins considering 7129 models of one input. On each model, the dataset is divided into three randomly subsets; 60% for training, 20% for validating and 20% for testing. CMantec does not need to validate the model so this 20% percent is added to the training subset and finally uses 80% for training and 20% for testing. Once we have the model of one input that best generalize our problem, the process now begins to consider models of two inputs where the first input is the best obtained on the last step. This process is repeated until we find a step with one more input where its generalization do not increase the generalization obtained on the previous step.

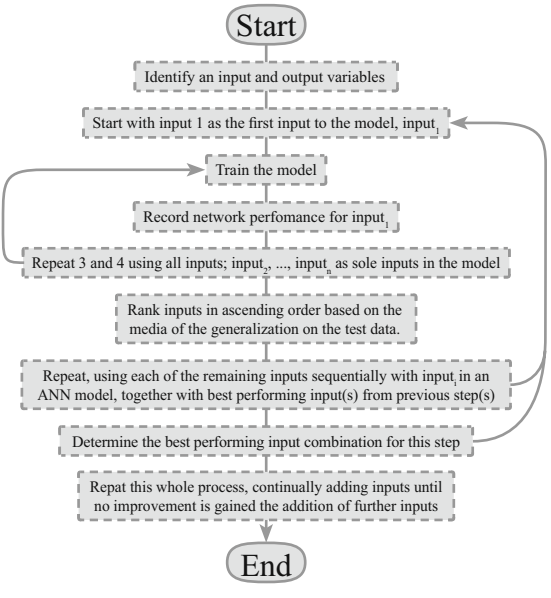

Fig. 1. Summary of the stepwise algorithm modelling process

For each model, a randomly subset of the dataset is created 50 times and for each subset the model is trained 10 times. This is 7.129 x 50 x 10 = 3.564.500 simulations per step of the model.

ANN architecture. The ANN modelling used [7] is a multilayer perceptron architecture with a sigmoidal transfer function[2] where weights are updated by a back propagation algorithm incorporating supervised learning.

Data of the dataset were scaled linearly between 0 and 1 using minimun and maximum values. The architecture shown in Fig. 2 utilised two hidden perceptrons in a single hidden layer and its initial weights were randomized between 0

[2] In the tolboox of MATLAB we used 'tansig' as the transfer function.

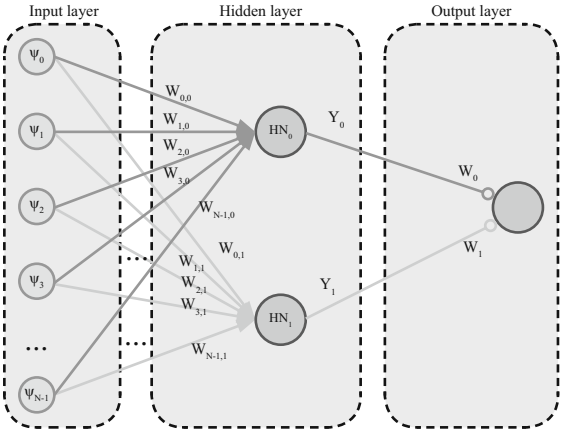

Fig. 2. Architecture of the ANN

and 1. Network training was stopped when the network error failed to improve on the test data split for 5000 epochs.

CMantec algorithm. CMantec (Competitive Majority Network Training Error Correcting) is a novel neural network constructive algorithm that combines competition between neurons using a modified perceptron learning rule to build compact architectures and cooperation between them to obtain the output of the network.

The topology of a CMantec network consists of one hidden layer that maps the information to an output neuron that uses a majority function to give us the output network (Fig. 3a). The choice of the output function as a majority gate is motivated by previous experiments in which very good computational capabilities have been observed for the majority function among the set of linearly separable functions [18]. The implementation of the majority neuron can be done establishing all the sinaptic weights to 1 and the umbral value to $M/2$ where M is the number of neurons in the hidden layer (Fig. 3b).

The results [20] show that the new algorithm generates very compact neural architectures with state-of-the-art generalization capabilities. In this paper we are applying this developed model to a real bioinformatic problem.

– Thermal Perceptron rule.
  At the single neuron level (Fig. 3c) the CMantec algorithm uses the thermal perceptron rule that ensures stability of the acquired knowledge while the architecture grows and while the neurons compete for new incoming information. Competition makes it possible that even after new units have been added to the network, existing neurons still can learn if the incoming information is similar to their stored knowledge and thus we first give some details of this algorithm. The thermal perceptron, introduced by Frean in 1992 [4] is a modification of the original perceptron learning rule [15] that

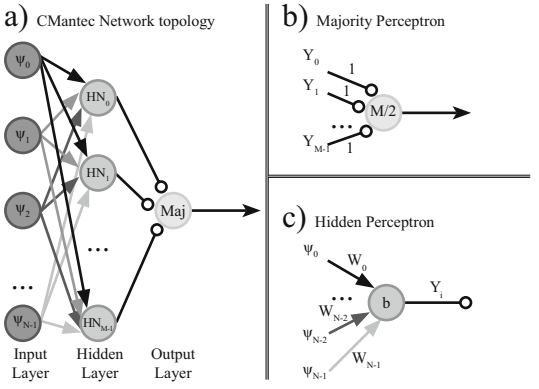

Fig. 3. Architecture of a CMantec network

aims to obtain a rule that provides a successful and stable linearly separable approximation to a non-linearly separable problem. The standard perceptron learning rule converges when the problem is linearly separable but is unstable when trying to learn non-linearly separable problems.

Consider a neuron, modelled as a threshold gate, having two response states: ON=TRUE=1 and OFF=FALSE=0, receiving input from N incoming continuous signals. The activation state (S) of the perceptron depends on the N input signals, ψ_i, and on the actual value of the N synaptic weights (w_i) and the bias (b) as follows:

$$S = \begin{cases} 1\ (ON) & \text{if } \phi \geq 0 \\ 0\ (OFF) & \text{otherwise,} \end{cases} \tag{1}$$

where ϕ is the synaptic potential of the neuron defined as:

$$\phi = \sum_{i=1}^{N} w_i\ \psi_i\ -\ b. \tag{2}$$

In the thermal perceptron rule, the modification of the synaptic weights, Δw_i, is done on-line (after the presentation of a single input pattern) according to the following equation:

$$\Delta w_i = (t - S)\ \ \psi_i\ \ T_{fac}\ , \tag{3}$$

where t is the target value of the presented input, and ψ is the value of input unit i connected to the output by weight w_i. The difference to the standard perceptron learning rule is that the thermal perceptron incorporates the factor T_{fac}. This factor, whose value is computed as shown in Eq. 4, depends

on the value of the synaptic potential and on an artificially introduced temperature (T) that is decreased as the learning process advances, in a way that resembles the well known simulated annealing procedure [17].

$$T_{fac} = \frac{T}{T_0} \exp\{-\frac{|\phi|}{T}\}. \tag{4}$$

In Eq. 4, T_0 is the initial temperature value set at the beginning of the learning process, T is the actual temperature, and ϕ is the synaptic potential defined in Eq. 2. The value of the Temperature, T, is lowered steadily as the iterations (I) proceeds from the iteration 1 to the maximum number of iterations allowed (I_{max}) according to the value of a parameter m, following the next linear equation.

$$T \ = \ T_0 - m \ I. \tag{5}$$

In the Frean original formulation [4], the evolution of T depends on the values of T_0, m and I_{max} but in order to reduce the space of parameters and as it was observed that the algorithm was not much affected, the value of m in this work was set as to reduce T to 0 at the end of the allowed number of iterations, i.e., when $I = I_{max}$. Further, we observed that it was not necessary to use different values of T_0 when the input dimension was fixed and thus all the experiments in this work has been run with a value of T_0 equals to the number of input variables, N.

– The CMantec algorithm.
The procedure for constructing an architecture for a given set of examples (inputs plus corresponding target values) of dimension N starts by putting N neurons in the input layer, with no processing capabilities except the task of introducing the incoming values to the network. The initial architecture includes a single neuron in the hidden layer and an output neuron that will compute the majority function of the activation of the neurons present in the hidden layer.

The learning process starts as usual with the presentation of randomly chosen patterns from the training data set and the single present neuron in the hidden layer tries to learn these patterns using the thermal perceptron rule in Eq. 3 described in the previous section.

The CMantec algorithm has 3 parameters to be set at the time of starting the learning procedure. The maximum number of iterations allowed for each thermal perceptron in the hidden layer per learning cycle , I_{max}. The second parameter is named the growing factor, (g_{fac}), as it determines when to stop a learning cycle and include a new neuron in the hidden layer. The third parameter is Fi_{temp} and shows how fast is going to be deleted the noisy examples from the training dataset.

Thus, patterns are learned by selecting those thermal perceptrons from the hidden layer where its output differs from the target value and its T_{fac} is larger than the set value of g_{fac}. If more than one thermal perceptron in the hidden layer satisfy these conditions, the perceptron that has the highest

T_{fac} is the selected candidate to learn the incoming pattern. A new single neuron is added when there is no thermal perceptron that comply these conditions and a new learning cycle is created. In this case I is set to 0 for every thermal perceptron in the hidden layer and noisy examples are deleted by an active learning rule.

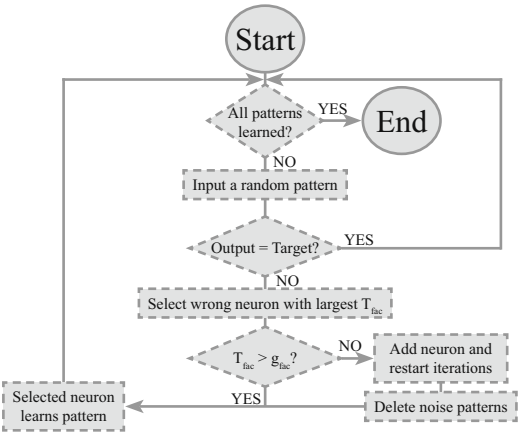

Fig. 4. Flow diagram of the CMantec algorithm

In Fig. 4 a flow diagram of the algorithm is shown. It includes a filtering stage, showed at the lower right part of the figure ("Delete noisy patterns"). This stage filters those patterns that not satisfy a filtering condition. This condition is established as:

$$nlp < mean(nap) + \phi * std(nap)$$

where nlp is the number of times that a certain pattern must be learned and nap is a vector of nlp values corresponding to the resting patterns in

Table 1. ANN and CMantec identified genes

ANN model						CMantec model					
Num. Genes	Probe set ID	Mean (%)	TPR* (%)	TNR* (%)	CPU time (h)	Probe set ID	Mean (%)	TPR* (%)	TNR* (%)	CPU time (h)	Neurons H. Layer
1	X17059_s_at	81,96	76,51	90,05	178	X76180_at	85,72	88,69	83,16	21	3,46 ± 0,4
2	HG273-HT273_at	84,3	82,66	88,65	243	HG4749-HT5197_at	94,04	95,39	92,25	25	1,86 ± 0,35
3	U05340_at	89,26	87,11	92,27	274	M31520_rna1_at	97,44	98,76	96,14	12	1 ± 0
4	J03778_s_at	92	90,72	93,85	273	U20325_at	98,32	97.74	98,65	1	1 ± 0
5	X16396_at	93,4	92,22	94,8	263						
6	X90857_at	96,3	94,73	97,99	298						

* TPR (**T**rue **P**ositive **R**ate) - TNR(**T**rue **N**egative **R**ate)

the dataset. The parameter ϕ is initially set to 2 and every time a neuron is added it is reduced on a $Fi_{temp}\%$.

In all the simulations that has been launched for this model, the parameters of the algorithm were initialize to: $I_{max} = 10000$, $g_{fac} = 0.2$ and $Fi_{temp} = 10$.

3 Results and Conclusion

The final results on both ANN and CMantec algorithm are shown in table 1, in which two important results could be highlighted. On one hand, execution time costs using CMantec algorithm are considerably smaller than using ANN algorithm. The column labelled as CPU time indicates an orientative execution time to estimate any model with n genes (first column in table 1). An approximation to the ratio between the execution times for both models is computed by averaging over values in columns CPU times, obtaining 254.83 and 14.75 hours for ANN and CMantec models respectively. This means that CMantec algorithm needs aproximately less than 17 times the execution time of the ANN model.

On the other hand, CMantec model needs less genes than ANN model in order to get a better generalization over the test data subset. CMantec requires only 4 genes while ANN model needs 2 more genes in order to get their respective maximum predictive performance. Moreover, CMantec outperforms the ANN model in prognosis accuracy (98.32% against 96.3%). Also, sensitivity (TPR) and especificity (TNR) were computed, and the results showed the improvement in both pronostic indexes by the CMantec model.

An analysis of variance (ANOVA) test was then applied to compare the mean values for prognosis accuracy between CMANTEC and ANN algorithms. A p-value$<$0.001 indicated that differences in prognosis accuracies were found statistically significant.

As conclusion, this work presents a novel contructive neural network (CMantec) that has been applied to identify the bests gene transcript signatures predictive for estrogen receptor. The CMantec model clearly outperforms the ANN model in process execution time and final prognosis accuracy. Thus, CMantec appears as a powerful tool to identify gene signatures that predict the ER status for a given patient, also reducing the number of gene transcripts selected as inputs to the system. We will continue studying on this research line testing and validating our model against other datasets and other types of cancer.

Acknowledgments

The authors acknowledge support from CICYT (Spain) through grant TIN2008-04985 (including FEDER funds) and from Junta de Andalucía through grant P08-TIC-04026. Leonardo Franco acknowledges support from the Spanish Ministry of Science and Innovation (MICIIN) through a Ramón y Cajal fellowship.

References

1. Andree, H.M.A., Barkema, G.T., Lourens, W., Taal, Vermeulen, J.C.: A comparison study of binary feedforward neural networks and digital circuits. Neural Networks 6, 785–790 (1993)
2. Baum, E.B., Haussler, D.: What size net gives valid generalization? Neural Computation 1, 151–160 (1989)
3. Frean, M.: The upstart algorithm: A method for constructing and training feedforward neural networks. Neural Computation 2, 198–209 (1990)
4. Frean, M.: Thermal perceptron learning rule. Neural Computation 4, 946–957 (1992)
5. Gómez, I., Franco, L., Jerez, J.M.: Neural Network Architecture Selection: Can function complexity help? Neural Processing Letters (in press, 2009) doi:10.1007 s11063-009-9108-2
6. Keibek, S.A.J., Barkema, G.T., Andree, H.M.A., Savenlie, M.H.F., Taal, A.: A fast partitioning algorithm and a comparison of binary feedforward neural networks. Europhys. Lett. 18, 555–559 (1992)
7. Lancashire, L.J., Rees, R.C., Ball, G.R.: Identification of gene transcript signatures predictive for estrogen receptor and lymph node status using a stepwise forward selection artifical neural network modelling approach. Artificial Intelligence in Medicine 43, 99–111 (2008)
8. Lawrence, S., Giles, C.L., Tsoi, A.: What Size Neural Network Gives Optimal Gener- alization? Convergence Properties of Backpropagation. Technical Report UMIACS-TR-96-22 and CS-TR-3617, University of Maryland (1996)
9. Mezard, M., Nadal, J.P.: Learning in feedforward layered networks: The tiling algorithm. J. Physics A 22, 2191–2204 (1989)
10. Nicoletti, M.C., Bertini, J.R.: An empirical evaluation of constructive neural network algorithms in classification tasks. International Journal of Innovative Computing and Applications 1, 2–13 (2007)
11. Parekh, R., Yang, J., Honavar, V.: Constructive Neural-Network Learning Algorithms for Pattern Classification. IEEE Transactions on Neural Networks 11, 436–451 (2000)
12. García-Pedrajas, N., Ortiz-Boyer, D.: A cooperative constructive method for neural networks for pattern recognition. Pattern Recognition 40, 80–98 (2007)
13. Pellagatti, A., Vetrie, D., Langford, C.F., Gama, S., Eagleton, H., Wainscoat, J.S., Boultwood, J.: Gene Expression Profiling in Polycythemia Vera Using cDNA Microarray Technology. Cancer Res. 63, 3940–3944 (2003)
14. Linder, R., Richards, T., Wagner, M.: Microarray data classified by artificial neural networks. Methods Mol. Biol. 382, 345–372 (2007)
15. Rosenhlatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65, 386–408 (1959)
16. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by backpropagating errors. In: Rumelhart, D., Mc-Clelland, J. (eds.) Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
17. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
18. Subirats, J.L., Franco, L., Gòmez, I., Jerez, J.M.: Computational capabilities of feedforward neural networks: the role of the output function. In: Proceedings of the XII CAEPIA'07, vol. II, pp. 231–238 (2008) ISBN: 978-84-611-8848-2

19. Subirats, J.L., Jerez, J.M., Franco, L.: A New Decomposition Algorithm for Threshold Synthesis and Generalization of Boolean Functions. IEEE Transactions on Circuits and Systems I 55, 3188–3196 (2008)
20. Subirats, J.L., Franco, L., Molina, I., Jerez, J.M.: Competition and Stable Learning for Growing Compact Neural Architectures with Good Generalization Abilities: The C-Mantec Algorithm (2009) (sent for publication)
21. Utgoff, P.E., Stracuzzi, D.J.: Many-Layered Learning. Neural Computation 14, 2497–2539 (2002)
22. Wei Jun, S., Greer Braden, T., Frank, W., Steinberg Seth, M., Chang-Gue, S., et al.: Prediction of Clinical Outcome Using Gene Expression Profiling and Artificial Neural Networks for Patients with Neuroblastom. Cancer Res. 64, 6883–6891 (2004)
23. West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., et al.: Predicting the clinical status of human breast cancer by using genes expression profiles. Proc. Natl. Acad. Sci. U.S.A. 98, 11462–11467 (2001)
24. Xu, Y., Selaru, F.M., Yin, J., Zou, T.T., Shustova, V., Mori, Y., Sato, F., et al.: Prediction of Clinical Outcome Using Gene Expression Profiling and Artificial Neural Networks for Patients with Neuroblastom. Cancer Res. 62, 3493–3497 (2002)