Evolving neural networks to flow cytometric data interpretation

G. C. Pereira[1], A. Bonomo[2] & N. F. F. Ebecken[1]
[1]*COPPE/UFRJ - Federal University of Rio de Janeiro*
[2]*Experimental Medicine Division, Research Coordination, Cancer National Institute*

Abstract

Flow cytometry is a method largely employed in research and medical practice as a diagnostic exam on grave diseases, such as infections by HIV viruses and mainly leukemia. This paper aims to introduce a new methodological approach. Artificial neural networks based on supervised training are applied to pattern recognition of their optical blood cell population coming from patients. Nonlinear system identification via artificial neural networks consists of adjusting the network in such a way that it approximately describes the input-output mapping of the system. In its complete form, network induction involves both *parametric* and *structural* learning. The two tasks can be seen as optimization problems and, as such, they may present multiple local minima, non-differentiability and discontinuity. The idea of applying *Genetic Algorithms* to the problem of complete network acquisition comes naturally.

The preliminary results demonstrate that this approach is promising and can soon contribute to protocols related to this disease.
Keywords: flow cytometry; neural network; diagnostic.

1 Introduction

Flow cytometers are equipments which use light dispersion and quantify different cellular populations (Van Dam and Tjalma [28]). Depending of the arrangement it can obtain two or thirteen simultaneous parameters for each cell or organism. Many diseases depend on the right "classification" from flow cytometers to give an accurate diagnosis and subsequent treatment. Patients infected, for instance by HIV viruses need their CD4+ cells quantified to suitable



evolution and treatment. The determination of a DNA index (which reveals the chromosome number of a tumoral cell in relation to normal) in malign cells is important as a prognostic factor of tumoral evolution. The "imunophenotype" and classification of cellular subtypes are crucial to establish the suitable treatment (Pui [22]). However, the cytometric data analysis based on bi-dimensional visualization, limits the reach of this technique, see figure 1.

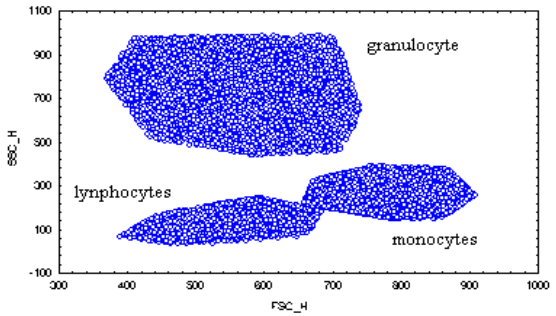


Figure 1: Three type blood cells: granulocytes, monocytes and lymphocytes.

In addition the subjectivity of the method depends on expertise and "good sense" of operator who sets, how and which cell populations are important. At this context, the application of neural networks became considerable because it is able to take the whole information in side the multi-dimensional space of data segregating it accordingly previously establish patterns.

2 Cellular kind determination

The central idea at this first approach is whether each population has self-characteristics, which differentiate themselves from other. These characteristics act on the optical properties generated by cytometers giving own "signatures" of populations. Nevertheless, depending on physiological conditions under which these cells are submitted or their different degrees of cellular development, these signatures also vary contributing to great variability of signals yielded by the equipment, as shown in table 1. Even the same phenotype variations of the same population give some oscillations on medical patterns and can become evident subpopulations. Thus the object of this work is to study the capability of artificial neural networks to "learn" these variabilities to affect the database with five optical parameters generated from three cited cell types.

Initially, was an acquired sample of periphery blood at the Cancer National Institute (INCA). The three blood cell populations were isolated. It was used a Fac Scan Becton DiKson flow cytometer equipped with a single blue laser of 488 nm. Such samples were passed through the laser beam cell by cell, and the light scatter parameters determined by their forward scatter, side scatter and three fluorescences (F1, F2, and F3), see figure 2 and retreated in list mode.

Table 1: Basic statistics of cytometer readings.

Type 1					
	FSC-H	**SSC-H**	**FL1-H**	**FL2-H**	**FL3-H**
max.	739	987	538	479	503
min.	369	441	0	0	0
mean	558.9	724.2	211.6	215.4	222.7
s.d.	78.32	125.72	39.80	40.39	36.66
Type 2					
	FSC-H	**SSC-H**	**FL1-H**	**FL2-H**	**FL3-H**
max.	684	906	362	392	402
min.	390	33	0	0	0
mean	546.1	119.7	142.9	129.3	110.3
s.d.	54.44	37.85	52.59	52.72	54.23
Type 3					
	FSC-H	**SSC-H**	**FL1-H**	**FL2-H**	**FL3-H**
max.	908	387	405	427	591
min.	659	146	0	0	0
mean	782.8	262.3	213.9	195.9	180.6
s.d.	52.72	52.20	41.88	46.72	51.93

Each numeric value is the shining intensity received by the sensors. The resulting matrix is built of five parameters (columns) and each row related to each cell (event). We have performed 10000 readings for each kind of cell separately. Later, an additional column was settled showing the type 1 to lymphocyte, 2 to monocyte and 3 to granulocyte and their optical variabilities given under "exiting" conditions. Following, the three dataset was joint together performing a single matrix with six columns (parameters) and thirty thousands row (cases).

3 Neural network types

The behavior of neural networks, how they map input data to output data, is influenced primarily by the transfer functions of neurons, how they are interconnected and the weights of those interconnections.

Typically, an architecture or structure of a neural network is established and one of a variety of mathematical algorithms is used to determine what the weights of the interconnections should be to maximize the accuracy of the outputs. Neural networks are "trained", meaning they are previous example to establish (learn) the relationships between the input variables and the predicted variables by setting these weights. Once these relationships are established (the neural network is trained), the neural network can be presented with new input variables and it will generate predictions.

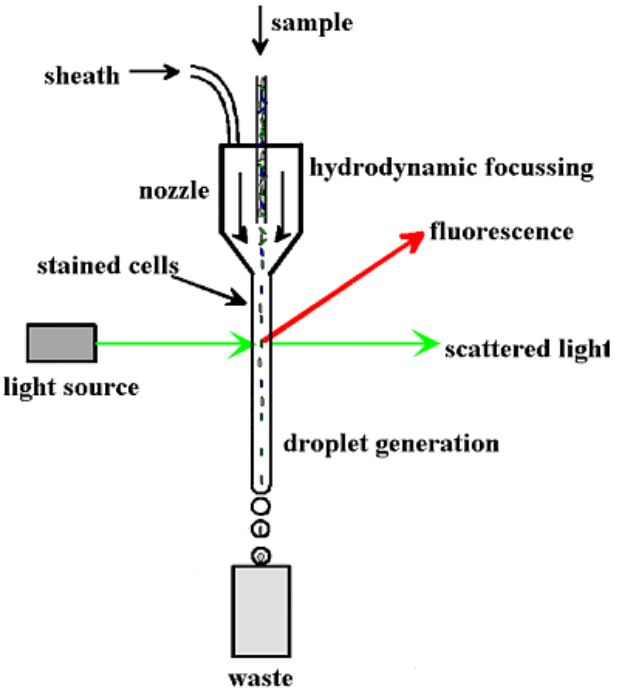


Figure 2: Flow cytometry principle.

4 MLP back propagation neural network

MLP Back Propagation neural networks are probably the most common neural architecture implemented today (Rumelhart et al. [23]). The error between the neural network outputs and desired outputs are propagated backwards through the network to aid in the determination of what the new values of weights should be. This is done for each training data record over and over until the error reaches some stable minimum.

5 Radial basis function networks

The radial Basis Functions (RBF) architecture generates a mapping between inputs and outputs using a weighted sum of basis functions (Bishop [5, 6]). The basis functions are radial Gaussians (i.e., the basis functions are invariant to rotations about their centers). Free parameters include the Gaussian center locations, the spread (standard deviation) of each Gaussian, and the weights which determine the degree to which Gaussian contributes to each output.

An RBF model can be trained by an iterative gradient descent method similar to the Back propagation algorithm used for training the MLP. However,

empirical results have shown that RBF can be trained much faster using a two-step process, without much degradation in performance. The two steps are:

1. Determine the centers and spreads of the Gaussians by minimizing the average variation of call training patterns to Gaussian centers using *K* Means clustering;
2. Adjust the weights from each Gaussian to the outputs by minimizing the mean square error. Since the weighted-sum relationship between an output variable and the Gaussian is linear, the global optimum can be computed quickly (via the use of pseudo-matrix inversion to solve a standard least-squares problem).

6 Network architecture optimization

Once it was possible to have data from isolated populations, the problem now falls over what is the most fitted neural network architecture to solve this data classification problem. At the supervised training we have tested two systems: Multilayer Perceptron Neural Network (MLP) and Radial Basis Function Neural Network (RBF). In both cases, the so-called back-propagation algorithm was used (Holland [15] and Opitz [20]). The optimization of number of computational layers, number of neurons in each layer, connectivity degree between neurons and what would be the best set of input variables, a genetic algorithm, a heuristic search technique, was left to evolve until the maximum of two hidden layers to avoid high computational costs. All computational experiments usually employed 50% of data for training, 25% for testing and 25% for validation. The composition of this data profile was randomly chosen. In another step, other database having a mix of signatures of the three populations but at this time without the indicating column was submitted to the best models.

The training of a RBF network can be seen as an optimization problem. If the centers and widths of the hidden RBFs are not fixed, it turns into a difficult nonlinear task. In the case where the number of units in the hidden layer is to be optimized the problem becomes still harder, with a non-differentiable cost function. In such cases, using a genetic algorithm (Holland [15] and Goldberg [14]) as an optimization tool is a natural choice. There are several possibilities of using a GA to configure a RBF network. A straightforward approach is to fix the topology and use the GA as an optimization method to compute all free parameters. It has been used by Sheta for time series forecasting.

In Mak [18], the number of hidden units was fixed, and the GA optimized only the location of the centers. The widths of the functions and the outputs weights were computed by the K-nearest neighbor heuristic and SDV (singular value decomposition), respectively. Whitehead and Choate [29] also fixed the number of RBFs, and evolved their locations and widths, but on a completely different and interesting fashion: instead of encoding an entire model in each individual, the whole set of chromosomes cooperate to constitute a RBF network, each one being a basis function. Whitehead [30] presents an alternative approach to evaluate each individual, based on SVD.



However, using a GA to configure only the parameters of the RBF network does not seem to be the best choice, specially considering there are specific (and faster) algorithms to do this task. So, a better alternative would be to adopt the GA to also determine the network topology. The literature offers a large number of works with these proposals. In this way it seems to support this hypothesis. Whitehead and Choate [31], Billings [4], Carse [7], Maillard [12], Liu [13], Zhao [24], de Lacerda [11], and Barreto et al. [1] are some good examples. In these approaches, the GA is responsible for setting the number of hidden units, as well as the location of their centers and their widths, while the weights in the output layer are computed by an algebraic or gradient-descent method. Naturally, there are many differences between them. In Whitehead and Choate [31], for example, an indirect representation is used; the locations of centers are governed by space-filling curves, whose parameters evolve genetically. Another example is the work of Maillard and Gueriot [12] in which the basis function is not restricted to Gaussians and is also subjected to evolution. Based in statements given by Zhao [24], the work of Barreto et al. [1] employs a minimalist initialization procedure within the GA, which makes the evolution process somewhat similar to connectionist constructive methods.

Another idea is hybridize the configuration process, using the GA as a support tool for another algorithm. In Burdsall [2], the GA was used to optimize the number and initial cluster centers of the K-means algorithm; the RBF network training then proceeded as in Moody (1989).

Chen et al. [25,26] have present a two-level hybrid learning method for RBF networks which employs both GAs and the OLS algorithm. A regularized orthogonal least squares algorithm (ROLS) was employed to construct the RBF networks at the inner level, while the two main parameters of this algorithm, the global RBFs width and the regularization parameter were optimized by a GA process at the outer level.

7 Results

Many computational experiments were done and the six best models (three MLP and three RBF) separated and its results compared following table 2.

In general, all models shows good robustness and discriminative capabilities between the three classes.

It was allowed the model evolution until a maximum of twenty computational units in hidden layer. It is easy to verify that MLPs had good correlation between the true class and those indicated by the models (0,95%). The model MLP-3 shows the worst performance just it have to use 18 hidden neurons to get almost the same result of MLP-1 which have only one hidden neuron. It is worth to see, both models were nor able to optimize their respective architecture using all five input variables. However, MLP-2 with only three inputs and one hidden unit reach better performance with RMSE 0,175380 and AME of 0,246011 contrasting this class of models. Actually, MLP-2 was able to classify all 30000 events saving four unknown cases.



Table 2: Model obtained results.

-NN type	Layers	Correlation	RMSE	AME
MLP-1	05:01:01	0.955364	0.301196	0.246011
MLP-2	03:01:01	0.955147	0.244372	0.175380
MLP-3	05:18:01	0.953830	0.369072	0.314557
RBF-1	02:12:01	0.985542	0.138341	0.097539
RBF-2	02:04:01	0.993029	0.151542	0.131009
RBF-3	02:05:01	0.985542	0.138341	0.097536

When analyzing Radial Basis Function Neural Networks an interesting case could be noted about their performance. It is clear this type of model have better performance (correlation of 0,99%, 0,98%) while MLP (0,95%). Really, RBF-1 shows twenty hidden neurons, correlation of 0,98%, RMSE 0,138341and AME of 0,097539, but RBF-3 smaller architecture (2:5:1), the same numerical correlation and RMSE yet minor AME (0,097536), possibly due to its topological optimization. This fact gives an evidence of several neural network can reach similar results in spite of their architectures, the difference in this case stays on the higher or lesser computational effort.

Whatever the same table, any one can see the model RBF-2 which gets the best correlation index amongst whole models reaching the brand of 0,99%. Its RMSE is 0,151542 and AME 0,131009 displaying an excellent performance even so numerically "inferior" to those RBF-1 and RBF-3. This architecture was able to place all studied events at their concerning classes and present yet the best approximation.

8 Discussion

As much the systems MLP as RBF are able to classify flow cytometry patterns with plenty accuracy. The problem denote very simple despite a large optical variability of each cell population be linearly separated and this database can be segmented though networks only with the forward and side scatter parameters. The promising potential application of neural network in order to detection of Minimal Residual Disease (MRD) in leukemias by the persistence of a small number of sick cells after the first treatment cycle have primordial prognosis importance and will determine the evolution and subsequent treatment. Detection of MRD through molecular biology technique, PCR in real time, is unequivocal (Panzer-Grümayer et al. [21]), but its cost is extremely high, forbidden nowadays. The custom option is flow cytometry, which use phenotype characteristic of leukemia cells (Counstan-Smith et al. [8]; Campana, 2000; Counstan-Smith et al. [10]). With the utilization of fluorescent markers of many wavelengths, besides physical cell characteristics given by light dispersion parameters will be possible the identification of specific molecules of sick cells

and characterize their phenotype. However, leukemic cells come from normal cells of osseous medulla and in most cases have very similar phenotypes. In such case, using the conventional cytometric analysis, from bidimensional visualization an overlap of two populations exists, causing false MRD. At this context, the multivariate analysis of neural network capabilities can be a good aid.

References

[1] Barreto, H. Barbosa, N. Ebecken, Growing compact RBF networks using a genetic algorithm, in: Proc. of the VII Brazilian Symposium on Neural Networks, Recife, PE, pp. 61–66, 2002.

[2] Burdsall, C. Giraud-Carrier, GA-RBF: A self-optimising RBF network, in: Proc. of the Third International Conference on Artificial Neural Networks and Genetic Algorithms, Springer-Verlag, pp. 348–351, 1997.

[3] URL citeseer.nj.nec.com/71534.html.

[4] Billings, S.A., G.L. Zheng, Radial basis function network configuration using genetic algorithms, Neural Networks 8 (6) (1995) 877-890.

[5] Bishop, "Improving the Generalization Properties of Radial Basis Function Neural Network", Neural Computation, 3(4): 579-588, 1991.

[6] M. Bishop, "Neural Networks for Pattern Recognition", Clarendon Press, Oxford, 1995.

[7] Carse, B. A.G. Pipe, T.C. Fogarty, T. Hill, Evolving radial basis function neural networks using a genetic algorithm, in Proc. Of the IEEE Int. Conf. On Evolutionary Computation, IEEE, pp. 300-305, volume 1, 1995.

[8] Coustan-Smith E, Behm, F.G., Sanchez, J., Boyett, J.M., Raimondi, S.C., Rubnitz, J.E., Riviera, G.K., Sandlund, J.T., Pui, C.H. "Campana, D. Imunological Detection of Minimal Residual Disease in Children with Acute Lymphoblastic Leukemia", Lancet: 351(9102): 550-4, 1998.

[9] Coustan-Smith E., Sanchez, J., Hancock, M.L., Boyett, J.M., Behm, F.G., Ribeiro, R.C., Pui, C.H., Campana, D. 1998. "Clinical Importance of Minimal Residual Disease in Childhood Acute Lymphoblastic Leukemia". Blood; 96(8): 2691-2696.

[10] Coustan-Smith E., Sanchez, J., Hancock, M.L., Boyett, J.M., Behm, F.G., Ribeiro, R.C., Pui, C.H., "Campana, D. Clinical Importance of Minimal Residual Disease in Childhood Acute Lymphoblastic Leukemia". Blood; 96(8): 2691-2695, 2000.

[11] G. M. de Lacerda, A. C. P. L. F. de Carvalho, T. B. Ludermir, Evolutionary optimization of RBF networks, in: VIth Brazilian Symposium on Neural Networks, Sociedade Brasileira de Computac¸˜ao (SBC), IEEE, pp. 219–224, volume I, 2000.

[12] P. Maillard, D. Gueriot, RBF neural network, basis functions and genetic algorithm, in: Int. Conf. on Neural Networks, IEEE, pp. 2187–2192, volume 4, 1997.

[13] P. Liu, V. Kadirkamanathan, Multiobjective criteria for neural network structure selection and identification of nonlinear systems using genetic



algorithms, in: IEEE Proc. Part P: Control Theory and Applications, 146(5), 1999, pp. 373–382.
[14] Goldberg, D. Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Readings, MA, 1989.
[15] Holland, J.H., 1975, "Adaptation in natural and Artificial Systems, Ann Arbor, The University of Michigan Press.
[16] K. O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies.
[17] URL citeseer.nj.nec.com/stanley01evolving.html.
[18] Mak, M.W., K.W. Cho, Genetic evolution of radial basis function centers for pattern classification, in: Proceedings of The 1998 IEEE International Joint Conference on Neural Network, pp. 669-673, volume 1, 1998.
[19] Moody, J., C. Darken, Fast learning in networks of locally tuned processing units, Neural Computation 1 (2) 281-294, 1998.
[20] Optiz D. Shavilik, J, "Connectionist Theory Refinement: Genetically Searching the Space of Network Topologies, Journal of Artificial Intelligence Research, 6, pp. 177-209, 1997.
[21] Panzer-Grümayer E.R., Schneider, M., Panzer, S., Fasching, K., Gadner, H., "Rapid Molecular Response During Early Induction Chemotherapy Predicts A Good Outcome In Childhood Acute Lymphoblastic Leukemia". Blood; 95(3): 790-794, 2000.
[22] Pui, C.H. 1997. "Acute Lymphoblastic Leukemia". Pediatric Clin North Am.; 44(4): 831-846.
[23] Rumelhart, D.E., Hilton, G.E. and Willians, R.J., 1986, "Learning Internal Representation by Error Propagation", Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations (edited by D.E. Rumelhant and J.L. McClelant). MIT Press, Cambridge, MA.
[24] Q. F. Zhao, O. Hammami, K. Kuroda, K. Saito, Cooperative co-evolutionary algorithm how to evaluate a module, in: IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks, pp. 150–157, 2000.
[25] S. Chen, Y. Wun, K. Alkadhimi, A two-layer learning method for radial basis function networks using combined genetic and regularised OLS algorithms, in: Proceedings of Genetic Algorithms in Engineering Systems: Innovations and Applications, Vol. 414, IEEE, pp. 245–249, 1995.
[26] S. Chen, Y. Wu, B. L. Luk, Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks, IEEE-NN 10 (5) 1239, 1999.
[27] Sheta, A.F., K.D. Jong, Time-series forecasting using GA-tuned radial basis functions, Information Sciences Special issue.
[28] Van Dam P.A., Tjalma W.A.A., "Clinical Applications of Flow Cytrometry". In Recent Advances in Histopathology, 18, Lowe D6, Underwoord JCE (eds), 1999.



[29] Whitehead, B.A., T.D. Choat, Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction, IEEE Trans. On Neural Networks 7(4) 869-880, 1995.
[30] Whitehead, B.A., Genetic evolution of radial basis function coverage using orthogonal niches, IEEE Trans. On Neural Networks 7 (6) 1525-1528, 1996.
[31] Whitehead, B.A., T.D. Choat, Evolving space-filling curves to distribute radial basis functions over an imput space, IEEE Trans. On Neural Networks 5 (1) 15-23, 1993.

