Competitive Neural Networks for Fault Detection and Diagnosis in 3G Cellular Systems

Guilherme A. Barreto[1], João C. M. Mota, Luís G. M. Souza, Rewbenio A. Frota, Leonardo Aguayo, José S. Yamamoto[2], and Pedro E. O. Macedo

[1] Department of Teleinformatics Engineering,
Federal University of Ceará, Fortaleza-CE, Brazil
Email: guilherme@deti.ufc.br,
WWW: http://www.deti.ufc.br/~guilherme
[2] CPqD Telecom & IT Solutions, Campinas-SP, Brazil
Email: sindi@cpqd.com.br,
WWW: http://www.cpqd.com.br

Abstract. We propose a new approach to fault detection and diagnosis in third-generation (3G) cellular networks using competitive neural algorithms. For density estimation purposes, a given neural model is trained with data vectors representing normal behavior of a CDMA2000 cellular system. After training, a normality profile is built from the sample distribution of the quantization errors of the training vectors. Then, we find empirical confidence intervals for testing hypotheses of normal/abnormal functioning of the cellular network. The trained network is also used to generate inference rules that identify the causes of the faults. We compare the performance of four neural algorithms and the results suggest that the proposed approaches outperform current methods.

1 Introduction

The third generation (3G) of wireless systems promise to provide mobile users with ubiquitous access to multimedia information services, providing higher data rates by means of new radio access technologies, such as UMTS, WCDMA and CDMA2000 [1]. This multi-service aspect brings totally new requirements into network optimization process and radio resource management algorithms, which differ significantly from traditional speech-dominated second generation (2G) approach. Because of these requirements, operation and maintenance of 3G cellular networks will be challenging.

The goal of this paper is to propose straightforward methods to deal with fault detection and diagnosis (FDD) of 3G cellular systems using competitive learning algorithms [2]. We formalize our approach within the context of statistical hypothesis testing, comparing the performance of four neural algorithms (WTA, FSCL, SOM and Neural-Gas). We show through simulations that the proposed methods outperform current standard approaches for FDD tasks. We also evaluate the sensitivity of the proposed approaches to changes in the training parameters of the neural models, such as number of neurons and the number of training epochs.

The remainder of the paper is organized as follows. In Section 2, we describe the neural models and the data for training them. In Section 3, we introduce a general approach for the fault detection task and a method to generating inference rules from a trained neural model. Computer simulations for several scenarios of the cellular system are presented in Section 4. The paper is concluded in Section 5.

2 Competitive Neural Models

Competitive learning models are based on the concept of *winning neuron*, defined as the one whose weight vector is the closest to the current input vector. During the learning phase, the weights of the winning neurons are modified incrementally in order to extract *average features* from the input patterns. Using Euclidean distance, the simplest strategy to find the winning neuron, $i^*(t)$, is given by:

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \quad (1)$$

where $\mathbf{x}(t) \in \Re^n$ denotes the current input vector, $\mathbf{w}_i(t) \in \Re^n$ is the weight vector of neuron i, and t symbolizes the iterations of the algorithm. Then, the weight vector of the winning neuron is modified as follows:

$$\mathbf{w}_{i^*}(t+1) = \mathbf{w}_{i^*}(t) + \eta(t)[\mathbf{x}(t) - \mathbf{w}_{i^*}(t)] \quad (2)$$

where $0 < \eta(t) < 1$ is the learning rate, which should decay with time to guarantee convergence of the weight vectors to stable states. The competitive learning strategy in (1) and (2) are referred to as *Winner-Take-All* (WTA), since only the winning neuron has its weight vector modified per iteration of the algorithm. In addition to the plain WTA, we also simulate three simple variants of it, namely: (1) the *Frequency-Sensitive Competitive Learning* (FSCL) [3], the well-known *Self-Organizing Map* (SOM) [4], and the *Neural-Gas algorithm* (NGA) [5].

To evaluate the performance of these competitive models on FDD tasks we need to define a set of KPIs (Key Parameter Indicators), which consist of a number of variables responsible for monitoring the QoS of a cellular system. These KPIs are gathered, for example, from the cellular system's operator, drive tests, customer complaints or protocol analyzers, and put together in a pattern vector $\mathbf{x}(t)$, which summarizes the state of the system at time t:

$$\mathbf{x}(t) = [KPI_1(t) \ \ KPI_2(t) \ \ \cdots \ \ KPI_n(t)]^T \quad (3)$$

where n is the number of KPIs chosen. Among the huge amount of KPIs available for selection, we have chosen the Number of Users, the Downlink Throughput (in Kb/s), the Noise Rise (in dB), and the Other-Cells Interference (in dBm). The data to train the neural models were generated by a static simulation tool. In addition, each component x_j is normalized to get zero mean and unity variance.

3 Fault Detection and Diagnosis via Competitive Models

Once we choose one of the neural models presented in Section 2 and train it with state vectors $\mathbf{x}(t)$ collected during normal functioning of the cellular network (i.e. no examples of abnormal features are available for training). After the training phase is completed, we compute the quantization error associated to each state vector $\mathbf{x}(t)$, $t = 1, \ldots, N$, used during training, as follows:

$$e(t) = \|\mathbf{E}(t)\| = \|\mathbf{x}(t) - \mathbf{w}_{i^*}(t)\| \tag{4}$$

where $\mathbf{E}(t)$ denotes the quantization error vector and i^* is the winning neuron for the state vector $\mathbf{x}(t)$. In other words, the quantization error is simply the distance from the state vector $\mathbf{x}(t)$ to the weight vector $\mathbf{w}_{i^*}(t)$ of its winning neuron. We refer to the distribution of N samples of quantization errors resulting from the training vectors as the *normality profile* of the cellular system.

Using the normality profile we can then define a numerical interval representing normal behavior of the system by computing a lower and upper limits via percentiles. In this paper, we are interested in an interval within which we can find a given percentage $p = 1 - \alpha$ (e.g. $p = 0.95$) of normal values of the variable. In Statistics jargon, the probability p defines the confidence level and, hence, the *normality interval* $[e_p^-,\ e_p^+]$ is then called (empirical) confidence interval. This interval can then be used to classifying a new state vector into normal/abnormal by means of a simple hypothesis test:

$$\begin{array}{rl} \text{IF} & e^{new} \in [e_p^-,\ e_p^+] \\ \text{THEN} & \mathbf{x}^{new} \text{ is } \textbf{NORMAL} \\ \text{ELSE} & \mathbf{x}^{new} \text{ is } \textbf{ABNORMAL} \end{array} \tag{5}$$

The *null-hypothesis*, H_0, and the *alternative hypothesis*, H_1, are defined as:

- H_0: The vector $\mathbf{x}^{new}$ reflects the NORMAL activity of the cellular system.
- H_1: The vector $\mathbf{x}^{new}$ reflects the ABNORMAL activity of the cellular system.

Once a fault has been detected, it is necessary to investigate which of the attributes (KPIs) of the problematic input vector are responsible for the fault. From the weight vectors of a trained competitive neural model it is possible to extract inference rules that can determine the faulty KPIs in order to invoke the cellular network supervisor system to take any corrective action.

All the previous works generate inference rules through the analysis of the clusters formed by a subset of the NORMAL/ABNORMAL state vectors [7]. This approach is not adequate for our purposes, since the state vectors reflect only the normal functioning of the cellular network. We propose instead to evaluate the absolute values of the quantization errors of each KPI, computed for each training state vector:

$$ABS\left(\mathbf{E}(t)\right) = \begin{pmatrix} |E_1(t)| \\ |E_2(t)| \\ \vdots \\ |E_n(t)| \end{pmatrix} = \begin{pmatrix} |x_1(t) - w_{i^*1}(t)| \\ |x_2(t) - w_{i^*2}(t)| \\ \vdots \\ |x_n(t) - w_{i^*n}(t)| \end{pmatrix} \tag{6}$$

This approach is similar to that used in the fault detection task, but now we built n sample distributions using the absolute values of each component of the quantization error vector, $\mathbf{E}$. For the detection task we used only one sample distribution built from the *norm* of the quantization error vector, as described in (4). Then, for all the sample distributions, $\{\,|E_j(t)|\,\}$, $t = 1, \ldots, N$ and $j = 1, \ldots, n$, we compute the corresponding confidence intervals $[\,|E_j^-|, |E_j^+|\,]$, where $|E_j^-|$ and $|E_j^+|$ are the lower and upper bounds of the j-th interval.

Thus, whenever an incoming state vector $\mathbf{x}^{new}$ is signalized as abnormal by the fault detection stage, we take the absolute value of each component E_j^{new} of the corresponding quantization error vector and execute the following test:

$$\begin{array}{rl} \text{IF} & |E_j^{new}| \in [\,|E_j^-|, |E_j^+|\,], \\ \text{THEN} & x_j \ \text{ is normal.} \\ \text{ELSE} & x_j \ \text{ is one (possible) cause of the fault.} \end{array}$$

In words, if the quantization error computed for the KPI x_j is within the range defined by the interval $[\,|E_j^-|, |E_j^+|\,]$, then it is not responsible for the fault previously detected, otherwise it will be indicated as a possible cause of the detected fault. If none of the KPIs are found to be faulty, then a *false alarm* will be discovered and then corrected. Confidence levels of 95% and 99% are used.

4 Computer Simulations

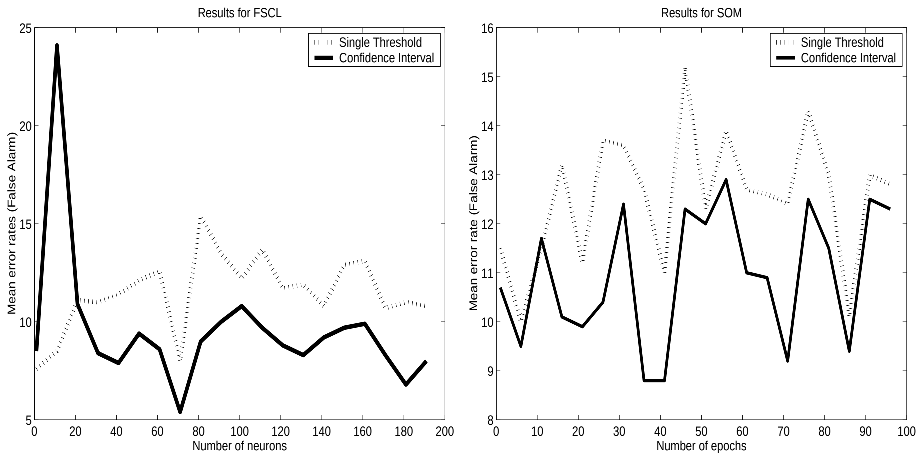
The 3G cellular environment used for system simulations is macrocellular, with two rings of interfering cells around the central one, resulting in a total of 19 cells. Other configurations are possible, with 1, 7 or 37 cells. All base stations use omnidirectional antennas at 30 meters above ground level, and the RF propagation model is the classic Okumura-Hata for 900MHz carrier frequency. Quality parameters, such as E_b/N_t^{Target} and maximum Noise Rise level are set to 5dB and 6dB, respectively. The number of initial mobile users is 60, which can be removed from the system by a power control algorithm. For each Monte Carlo simulation (drop) of the celular environment, a set of KPIs is stored and used for ANN training/testing procedures.

The first set of simulations evaluates the performance of the neural models, by quantifying the occurrence of false alarms after training them. The chosen network scenario corresponds to 100 mobile stations initially trying to connect to 7 base stations. No shadow fading is considered, and only voice services are allowed. Each data set corresponding to a specific network scenario is formed by 500 state vectors (collected from 500 drops of the static simulation tool), from which 400 vectors are selected randomly for training and the remaining 100 vectors are used for testing the neural models.

The results (in percentage) are organized in Table 1, where we show the intervals found for two confidence levels (95% and 99%). For comparison purposes, we show the results obtained for the single threshold approach. The error rates were averaged for 100 independent training runs. For all neural models,

Table 1. False alarm (FA) rates and confidence intervals for the various neural models.

Model	Proposed Approach		Approach by [6]	
	CI, FA (95%)	CI, FA (99%)	CI, FA (95%)	CI, FA (99%)
WTA	[0.366, 1.534], 12.43	[0.074, 1.836], 5.41	[0.000, 0.465], 17.91	[0.000, 1.018], 7.13
FSCL	[0.214, 1.923], 10.20	[0.136, 4.584], 1.80	[0.000, 1.126], 12.20	[0.000, 0.385], 3.00
NGA	[0.277, 1.944], 9.50	[0.1651, 4.218], 2.10	[0.000, 1.329], 10.10	[0.000, 0.941], 2.30
SOM	[0.361, 1.815], 8.75	[0.187, 2.710], 1.43	[0.000, 1.122], 13.28	[0.000, 1.191], 2.71



(a) (b)

Fig. 1. Error rates of false alarms versus (a) the number of neurons (FSCL) and (b) the number of training epochs (SOM).

the number of neurons and the number of training epochs were set to 20 and 50, respectively. As expected, the NGA/SOM models performed much better than the WTA/FSCL models.

The second set of simulations evaluates the sensitivity of the neural models to changes in their training parameters. The goal is to understand how the number of neurons and the number of training epochs affect the occurrence of false alarms after training the neural models. The results are shown in Figure 1. For each case, we compare the interval-based approach proposed in this paper with the single threshold presented in [6]. The chosen network scenario corresponds to 120 mobile stations initially trying to connect to 7 base stations, for which fast and shadow fading are considered this time. Voice and data services are allowed. For the sake of simplicity, results are shown for one neural model only, since similar patterns are observed for the others.

Table 2. Results (in percentage) for the joint fault detection and diagnosis tasks. FA=false alarm, AA=Absence of alarm.

Neural Model	$p = 95\%$			$p = 99$ %		
	FA	**AA**	**PERF**	**FA**	**AA**	**PERF**
WTA	5.54	0.00	91.50	2.20	0.04	95.80
FSCL	4.30	0.00	92.83	1.10	0.00	98.17
NGA	5.54	0.00	91.50	0.65	0.00	99.00
SOM	4.67	0.00	92.80	0.98	0.00	98.50

For a given value of a parameter (e.g. the number of neurons), the neural model is trained 100 times with different initial weights. For each training run, state vectors are selected randomly for the training and testing data sets. Also, the ordering of presentation of the state vectors for each training epoch is changed randomly. Then, the final value of the false alarm error rate is averaged for 100 testing runs. These independent training and testing runs are necessary to avoid biased estimates of the error rate.

In Figure 1a, the number of neurons is varied from 1 to 200, and each training run lasts 50 epochs. In Figure 1b, the number of epochs is varied from 1 to 100, while the number of neurons is fixed at 30. As a general conclusion, we can infer that in average the proposed approach produces better results than the single threshold method.

The last set of simulations evaluate the proposed methods for generating inference rules from competitive ANNs. Table 2 depicts the obtained results, averaged over 100 Monte Carlo simulations. ERROR I refers to the false-alarm rate, while ERROR II refers to the absence-of-alarm rate. The indicator $PERF$ (%) denotes the mean accuracy of the FDD system and it is computed as $PERF = 100 \cdot (1 - ERRORS/S)$, where S is the total number of state vectors used for testing. For each simulation, there were 8 state vectors corresponding to ABNORMAL conditions of the cellular network, and 52 state vectors reflecting NORMAL conditions. Thus, we have $S = 60$. One can infer that the maximum possible value of $ERRORS$ is S, reached only in the case of a very unreliable FDD system.

Two faulty state vectors per KPI were simulated by adding or subtracting random values obtained from Gaussian distributions with standard deviations greater than 1. The underlying idea of this procedure is to generate random values outside the range of normality of each KPI, and then, to test the sensitivity of the FDD system. It is worth emphasizing that all neural models performed very well, irrespective to their performances in the fault detection task. The only remaining error is the false alarm, which is the less crucial in a cellular network. Even this type of error has presented a very low rate of occurrence. All the ABNORMAL vectors have been found and his causes correctly assigned, i.e., all the faulty KPIs inserted in each ABNORMAL state vector have been detected.

5 Conclusion

In this paper we proposed general methods for fault detection and diagnosis in 3G cellular networks using competitive neural models. Unlike the available qualitative methods [8–10], the approach we took is focused on *quantitative* (numerical) results, more adequate for online performance analysis, being based on a statistically-oriented and widely accepted method of computing confidence intervals. Our methods outperformed current available single-threshold methods for FDD tasks.

Acknowledgements

The authors thank CPqD Telecom&IT Solutions/Instituto Atlântico for the financial support. The first author also thanks CNPq (DCR: 305275/2002-0).

References

1. Prasad, R., Mohr, W., Konäuser, W.: Third Generation Mobile Communication Systems - Universal Personal Communications. Artech House Publishers (2000)
2. Principe, J.C., Euliano, N.R., Lefebvre, W.C.: Neural and Adaptive Systems: Fundamentals through Simulations. John Wiley & Sons (2000)
3. Ahalt, S., Krishnamurthy, A., Cheen, P., Melton, D.: Competitive learning algorthms for vector quantization. Neural Networks **3** (1990) 277–290
4. Kohonen, T.: The self-organizing map. Proceedings of the IEEE **78** (1990) 1464–1480
5. Martinetz, T.M., Schulten, K.J.: A 'neural-gas' network learns topologies. In Kohonen, T., Makisara, K., Simula, O., Kangas, J., eds.: Artificial Neural Networks. North-Holland, Amsterdam (1991) 397–402
6. Laiho, J., Kylväjä, M., Höglund, A.: Utilisation of advanced analysis methods in UMTS networks. In: Proceedings of the IEEE Vehicular Technology Conference (VTS/spring), Birmingham, Alabama (2002) 726–730
7. Hammer, B., Rechtien, A., Strickert, M., Villmann, T.: Rule extraction from self-organizing networks. Lecture Notes in Computer Science **2415** (2002) 877–882
8. Binzer, T., Landstorfer, F.M.: Radio network planning with neural networks. In: Proceedings of the IEEE Vehicular Technology Conference (VTS/fall), Boston, MA (2000) 811–817
9. Raivio, K., Simula, O., Laiho, J.: Neural analysis of mobile radio access network. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), San Jose, California (2001) 457–464
10. Raivio, K., Simula, O., Laiho, J., Lehtimäki, P.: Analysis of mobile radio access network using the Self-Organizing Map. In: Proceedings of the IPIP/IEEE International Symposium on Integrated Network Management, Colorado Springs, Colorado (2003) 439–451