# Using a Combination of Artificial Neural Networks for the Diagnosis of Multiple Sclerosis

**Angel GUTIÉRREZ**
**Department of Computer Science, Montclair State University**
**Montclair, NJ 07043, U.S.A.**

**and**

**Carlos FERNÁNDEZ GARCÍA**
**Departamento de Matemática Aplicada, Universidad de Oviedo**
**33007 Oviedo, Spain**

## ABSTRACT

Very often the number of data available in the average clinical study of a disease is small. This is one of the main obstacles in the application of neural networks to the classification of biological signals used for diagnosing diseases. A rule of thumb states that the number of parameters (weights) that can be used for training a neural network should be around 15% of the available data, to avoid overlearning. This condition puts a limit on the dimension of the input space

In this paper we work with the Radial Basis Function and Functional Link artificial neural networks. To have enough data to train both neural networks, we increment the number of training elements, using randomly expanded training sets. This way the number of original signals does not constraint the dimension of the input sets.

Once the radial basis function has been trained, we train four functional link neural networks using samples of positives, false positives, negatives and false negatives results of the previous one. We then test the Radial Basis Function neural network by itself, and the chain of networks. A comparison with results obtained using other methods is presented.

**Keywords:** Neural Networks, Radial Basis Functions, Functional Link Architecture, Signal Processing, Wavelets, Health Sciences, Multiple Sclerosis.

## 1. INTRODUCTION

Doctors utilized Brain Stem Auditory Evoked Potentials (BSAEP) to diagnose patients with multiple sclerosis. MS can reveal, among other symptoms, a decrease of the wave V amplitude, an increase on absolute latencies and interpeak intervals latencies I-III, I-V, III-V. But the border between pathological and normal values sometimes is not well defined [1].

.

Figures 1 and 2 show the BSAEP of one of the healthy people, that it is called healthy # 1, and one of the sick people, called sick # 12. As can be seen, the discrimination between one case and the other is not very simple. Therefore when doctors diagnose this disease they often find difficult to state the rules they use to reach their conclusions and their percentage of success in the diagnosis is around 80%, with recognition for healthy people in the order of 95.7%, and recognition for sick people around 73.9%.

This diagnosis involves the estimation of the effects of the disease on the form of the waveform components. These components, which are localized in time and frequency, are given a physiological interpretation

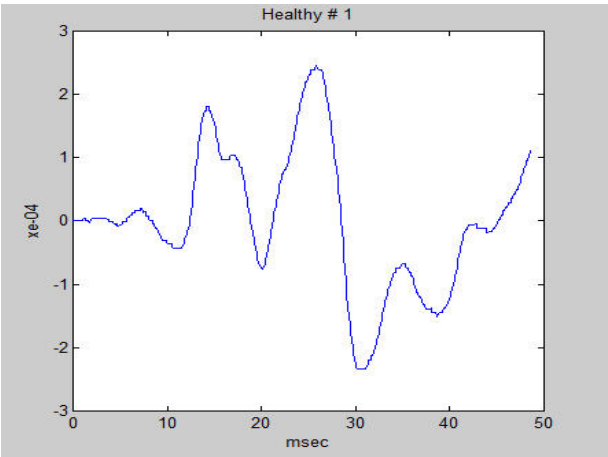


**Figure 1: Healthy # 1**

A Fourier expansion of the signals would allow us to classify the potentials according to their frequency, but would lose the phase information. Therefore we have used the wavelet transform that can be easily implemented and it is time localized as well as frequency localized.

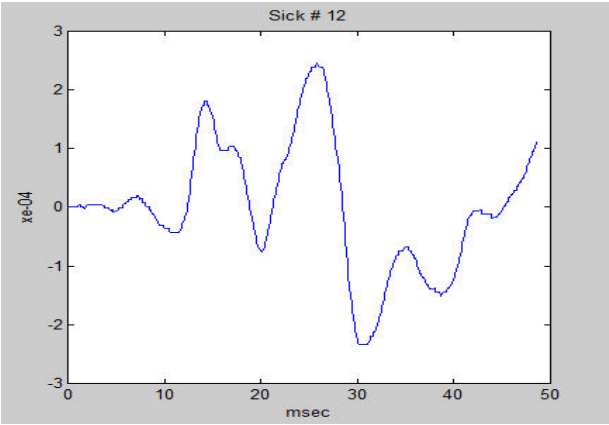


**Figure 2: Sick # 12**

Note that even cases corresponding to sick people may look completely different (See Figure2 and Figure 3).

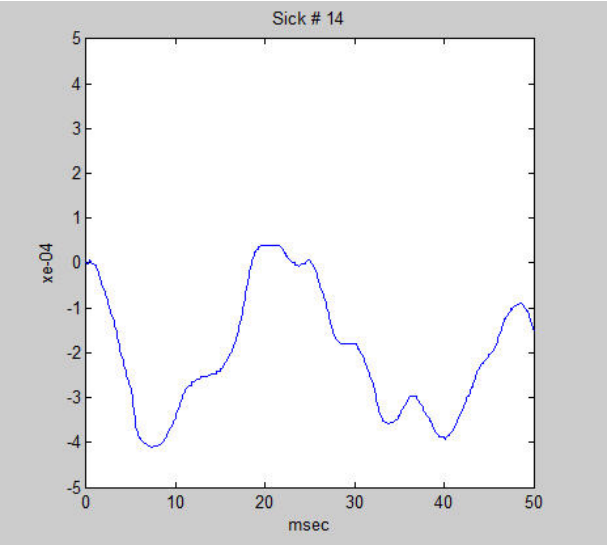


**Figure 3: Sick # 14**

Although the biological signals studied in this paper are BSAEPs, the techniques that we applied to them could be easily applied to study any time series related to the evolution of biological parameters. For instance, they could easily be applied to VEP, ECG's , EEG or EMG 's potentials, [2], [3], [4], [5].

The main obstacle in the application of this approach is the limited number of available signals. We have a set of 193 BSAEP signals, obtained from the Hospital Ramon y Cajal, Madrid (Spain), where 70 are normal signals, i.e., corresponding to healthy people, and 123 belong to patients diagnosed with multiple sclerosis. Small samples impose a limit on the number of parameters that can be learned by neural networks.

The signals are therefore first preprocessed and then compressed. The preprocessing begins by using the same time interval for all signals. Then the signals are digitized and incremented to 256 points using cubic splines, and finally we normalize them [6]. For the compression we use wavelet transforms that, as we mentioned, allow us to capture the decrease of the wave V amplitude, and an increase of interpeak intervals latencies. This reduces the loss of significant information contained in the signals.

Once they have been compressed, the authors, in collaboration with other colleagues, [7], selected a small number of the most significative features, according to different statistical criteria, like Kolmogorov-Smirnov, largest coefficients, Shannon's Entropy. These selected coefficients were then used as inputs. It is clear that the classification. would be better the more features we could feed into the network.

In this paper we first increment the number of training elements, using randomly expanded training sets [8] and we use them to train radial functions networks, following the ideas on [9], [10]. Clustering algorithms were used previously to find centers and radii for the radial basis functions [11]. The availability to generate an arbitrary number of samples removes not only the need to find centers and radii, but also the constraint that the number of original signals places on the dimension of the input set of the network. For each neuron we can determine the coordinates of the center (the same number as the inputs), the radius and the output weight. Thus, an n input network, with m radial functions, would require the fitting of $m*(n + 2) + 1$ parameters. So we still, from the hundreds of wavelet coefficients, must select only a handful of them and they must be the coefficients that contain the most significant features [12]. We use these networks with different kinds of wavelets and different selection criteria.

Once the radial basis function has been trained, we test them and record our results. This will allow us to determine the strength of this approach [13]. Since we have at our disposal as many training and testing sets as we need, we can generate enough samples of positives, false positives, negatives and false negatives results to train the other kind of neural network, the functional link neural network. The purpose of this approach is to complement the learning of the radial basis functions network with this chain of networks, where the knowledge acquired by the first one is passed to the second one, to improve the training. We then test the original data using first the radial basis function neural network by itself, and the chain of networks.

The fact that we can generate as many training sets as needed is very important for the generation of new training elements for the functional link neural networks, since we need to generate enough elements to obtain four kinds of different sets to train the four functional link neural networks.

In the neural network architectures section of this paper we describe the type of radial basis neural network that was used, with its number of input and hidden nodes, and how the network is trained using 37 wavelet bases offered in MATLAB: all biorthogonal bases (bior11- bior68), all Coiflets bases (coif1-coif5), the first 10 Daubechies bases (db1-db10) and the 7 first Symlets bases (sym2- sym8). In the same section we describe the functional link neural network architecture that was used in the four different

trainings. One for each of the four kinds of input elements, sick people that were recognized as such by the previous network, sick people that were diagnosed as healthy, and the other two opposite cases. Some of the results obtained using the different trainings are exposed in the following section, and in the conclusion we comment on the results and suggest some ideas for future work.

## 2. NEURAL NETWORK ARCHITECTURES

The radial basis function network architecture used for this work can be seen in Fig. 1. There are *n* input nodes in the fanout layer, *m* nodes and a bias in the hidden layer, and one output node.

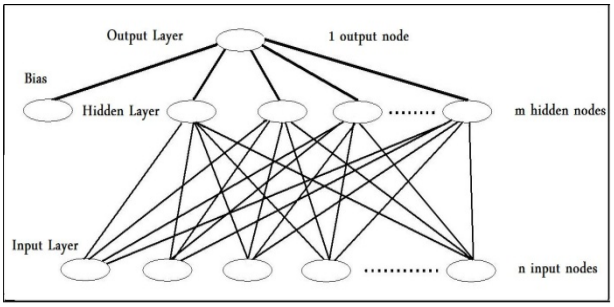


**Figure 4: Radial basis function neural network**

The value of *n* used was 10, and so was the number of hidden nodes. This will imply a set of 121 free parameters that needs to be fitted. This number of free parameters is consistent with the requirement that the number should be, at most175, and at best around 112. This will avoid the overlearning of the training set and allow the neural network to be able to generalize. This condition was imposed by the fact that the number of randomly generated input training vectors was 750. Of course if we increment this last number, the values of *m* and/or *n* could also be incremented correspondingly.

The functional link neural network architecture appears in Fig. 2, where only a few powers of the input data and a few connections among them have been shown.

In this case we use 10 nodes as the ones that will generate the polynomials, the same number that was used in the input layer of the radial basis function network. Instead of a bias, we use the output of the previous network as one extra input, but it will not be considered for the formation of higher order products between the original input nodes. In fact, for the rest of the input nodes, we compute the square of each value, the cubic value, and the forth power value, and we use them, together with the original values, as part of the new input set.

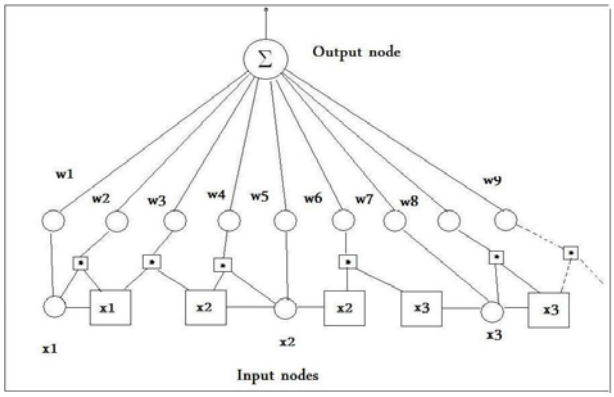


**Figure 5: Functional Link Neural Network**

Then we multiply the first value times each of the next 4 values. The second values times the next 3 values, until we compute the fourth value times the fifth value. All these values will also be part of the input nodes. The next set of input values will be generated taking the square of the first 5 values, and multiplying them for the other values in the set of the first 5 elements. This will end the generation of the input set, consisting of 71 values. Again, this number is consistent with the number of 500 input vectors that we used to train the functional link neural networks.

It should be noted that the values of the input vectors were obtained sorted by the power of discrimination of features, with the higher one placed in the first position. Therefore we try to maximize the benefit of the functional link neural network using the five elements with more discrimination power, when generating mixed higher order products.

For the generation of all the extra new input training vectors the ideas found in [8] were followed. For each of the two clusters corresponding to sick and healthy people, an estimation of the values for the elements in the probability density function, $f_{kME}(z)$, also denoted as $N_k(\mathbf{U}, R)$, k =1,2 Eq.(1), that maximized the differential entropy for that cluster, were computed.

$$\mathbf{N_k}(\mathbf{U}, \mathbf{R}) = \frac{1}{(\sqrt{2\pi})^{n+1}\,|R_k|^{\frac{1}{2}}}\, e^{-\frac{1}{2}(z-U_k)^T R_k^{-1}(z-U_k)} \qquad (1)$$

Here z denotes an input-output data vector, $U_k$ is the mean vector of the cluster k, $R_k$ is the covariance matrix of the same cluster, $|R_k|$ is its determinant, and T denotes the operation that performs the vector transpose operation. We denote the estimation of the mean vector as $\hat{U}_k$, and of the covariance matrix as $\hat{R}_k$, where here a diagonal load was added to insure its invertibility.
With this information, data were drawn for each cluster using the formula given in Eq. (2)

$$Z^i = \hat{U}_k + \hat{L}_k s^i \qquad (2)$$

where $s^i$ is an independently identically distributed (i.i.d.) vector sequence drawn from N(0,1), and $\hat{L}_k$ is the

Cholesky lower triangular matrix from the decomposition of $\hat{R}_k$.

Using this technique we generated the 750 input vectors for the training of the radial basis function neural network, and thousands of input vectors to obtain the sets to train the four different functional link neural networks.

The first network was trained using the 37 different wavelet bases. For the input space we selected the coefficients of the wavelet transforms that discriminated more according to the Kolmogorov-Smirnov test.

The input-output space of our data requires that all the values of every coefficient, on our sample, are normalized, with mean zero, and standard deviation of one. This avoids the problem that the output values, being far greater than any of *n* inputs in the case of sick people, could dominate the making of the partitions and in doing so, defeat the purpose of the algorithm. The mean value for each coefficient and the corresponding standard deviation is kept, to be utilized for the normalization of any future input vector that needs to be tested.

Each training process consisted of 3,000 random presentations, beginning with different random values. The random number generator utilized was the one included in MATLAB.

The learning rates η(k) for the centers, the radii and the weights were given by the linear function

$$\eta(k) = \eta_0 + (\eta_1 - \eta_0) * \frac{k}{NPR} \qquad (3)$$

where *k* is the iteration step, NPR is the number of presentations, $\eta_0$ is the initial learning rate, set at 0.001, and $\eta_1$ is the final rate, set at 0.08. These values for the initial and final learning rate for both the hidden and input layers were known to be acceptable.

Once the radial basis function network was trained, we used the thousands of generated new input vectors, and test them using the trained network. This way four sets of training set for the functional link network were created. The first set contained the all the input vectors corresponding to sick people whose diagnosis by the radial basis function network was correct. The second set contained those whose result was incorrect. Similarly the third and four set contained the correct and incorrect results of the healthy people.

We trained the four functional link neural networks with these new sets. Here we also use the learning rate given by (3), but the initial and final learning rates were ten times smaller. When the functional link neural networks were trained, we took our original set of data and tested using the radial basis function neural network. We recorded the results, and according to them, we test again the same original set using the functional link neural networks. If the result of the previous neural network suggested that the input vector corresponded to a sick people, the input vector would be tested using the functional link neural network that gave correct results for the sick people and the functional link neural network that gave incorrect results for the healthy people. A similar approach was used when the radial basis function neural network associated an input vector with a healthy person.

This way we could determine the cases of the positive-positive, false-positive, false-negative and negative-negative cases on the testing of the radial basis function neural network. This in fact improves the accuracy of our results.

## 3. EMPIRICAL RESULTS

The empirical results had a wide range of discrepancy. We selected original results on the range of success rate in the neighborhood of 75%, and then we applied the second set of functional link neural networks.

In the case of db4 wavelet, this second set improved the accuracy rate about 5%, with all the improvement, except one, being for the sick people. On the other extreme, for the bior13, the improvement rate was 12%, but in this case, we had a higher number of healthy people whose diagnosis was improved. In fact this number was more than double the corresponding number for the sick people. Looking into the original results, there is an explanation for these differences. Since the db4 wavelet was very accurate diagnosing the healthy people, most of the improvement occurred in the sick people. On the hand, the bior13 wavelet was more accurate for the sick people, and the improvement went to the healthy people.

| Wavelet | RBFANN | | | CHAINED | | |
|---|---|---|---|---|---|---|
| | Sick | Healthy | Total | Sick | Healthy | Total |
| db4 | 74.0 | 77.1 | 75.1 | 79.7 | 78.6 | 79.3 |
| bior1.3 | 83.7 | 61.4 | 75.6 | 88.6 | 85.7 | 87.5 |

Table 1: Percent of success rate for the different cases.

A view of these results can be found in Table 1. The first row indicates whose results are we considering: Using the single radial basis function neural network (RBFANN), or the chain of neural networks. Each other row corresponds to the success rates for a particular wavelet basis whose name appears in the first column. The columns reflect the general success rate for that wavelet basis and for the sick and healthy people. ,

There are other samples that were computed, but the results were similar to those shown, so we have omitted them.

## 4. CONCLUSIONS

Radial basis function networks had been used to diagnose Multiple Sclerosis. They provide an automatic, fast and reliable way to discriminate the signals from sick and healthy people, provided that we use a high number of hidden nodes. But we can improve their accuracy using a chained set of neural networks. Since the selection of the wavelet coefficients was done in order of their power of discrimination, it is more beneficial, for the same number of free parameters, to use more hidden nodes than to use more inputs beyond a certain number. Although we are making the selection from the several hundreds of wavelet

coefficients, an input dimension of ten is appropriate, and selecting a larger number does not enhance the learning of the networks.

One of the problems that we encounter with both networks is that the learning was bad when a local minimum for the error function was reached. Although we thought that we could avoid the clustering of the data to find centers and radii for the Gaussian functions, the need to avoid local minima suggests otherwise. In the future we should use a clustering algorithm to find the starting centers and radii of these functions. Besides we will have the advantage that in this case, before we start the clustering algorithm, we will know the number of clusters that we want, the number of hidden nodes in the network. This will avoid the problem of not knowing the number of clusters that should be created, a common problem in many of the clustering algorithms.

For the functional link neural network we could use another learning algorithm, to avoid the local minima.

In most of the previous approaches, the sick people had a higher success rate in their discrimination. But as we can see from the table, the success rate was higher for sick people in one case, and lower in the other.

For future research, we will try to use the random number generator described in [14]. Different discriminating criteria, like the largest coefficient in absolute value, Shannon entropy, or the Student t-test could be used. We could also consider a margin based feature selection criterion and apply it to measure the quality of sets of extracted features [13].

To work with other type of data, samples could be taken from the original data instead of using the wavelet transform coefficients, to generate the sets of extra new values. Another possibility is to select the even or odd values in the set of original data when they are expanded using cubic splines. This will generate twice as many numbers of starting data for the randomly generated expanded training set. Of course we could use a combination of all these approaches to compare the results with those obtained in this paper.

All these ideas for future research follow the line of using the expansion of training sets. We can apply different architectures and training algorithm to the functional link neural network. Instead of using regular polynomials involving all the data, we could use Chebyshev polynomials. Similarly instead of simple LMS learning, we could use swarm optimization to approach the global minimum [15] or prepare the functional link using the ideas of genetic algorithms [16]. Considering the approach to clustering we could obtain results using the ideas found in [17] and [18] to proceed as in [11] and [12], and compare all the results based in performance and cost.

In conclusion we can say that our findings are a good sign that chaining artificial neural networks with radial basis functions and functional link architectures could be used to help doctors when they are diagnosing cases of multiple sclerosis.

## 5. REFERENCES


[1] A. Blinowska, J. Verroust and D. Malapert, Bayesian statistics as applied to multiple sclerosis diagnosis by evoked potentials, *Electromyogr. Clin. Neurophysiol.*, Madrid, 32(1-2), 1992, 17-25.

[2] J.A. Sigüenza, S. González, J.R. Dorronsoro and Vicente López. ``Automatic Classification of Visual Evoked Potentials by Feedforward Neural Nerworks", Artificial Neural Networks (*Proc. of the International Conference on Artificial Neural Networks*). (North-Holand Elsevier T. Kohonen et al Edi.) (1991) 1117 - 1120.

[3] J. Raz and B. Turetzky, "Wavelet Models of Event-Related Potentials" pp. 571-590, in **Wavelets in Medicine and Biology**, A. Aldroubi and M.Unser eds. CRC Press 1996.

[4] M. Akay, **Detection and Estimation Methods for Biomedical Signals,** New York, Academic Press Inc., 1996.

[5] A. Subasi , M Yilmaz and H. R. Ozcalik, "Classification of EMG signals using wavelet neural network*", Journal of Neuroscience Methods* 156 (2006) 360–367.

[6] Charles K. Chui, **Wavelets: A Mathematical Tool for Signal Analysis**, (Philadelphia, SIAM 1997).

[7] C. Fernández-García, A. Gutiérrez and A. Somolinos, Diagnosis of multiple sclerosis using radial basis functions, *Proceedings of the IASTED98, International Conference on Modeling and Simulation,* Pittsburgh, PA, 1998, 3-6.

[8] G.N. Karystinos, and D. A Pados, On Overfitting, Generalization, and Randomly Expanded Training Sets, *IEEE Trans. Neural Networks*, *11*(5), 2000, 1050-1057.

[9] A. Gutiérrez, "Processing Brain Stem Auditory Evoked Potential for Improving Diagnosis of Multiple Sclerosis", *Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition (AIPR -09)*, Orlando, Florida, July 13-16, 2009, 193-197.

[10] A. Gutiérrez, "Diagnosis of Multiple Sclerosis Using Brain Stem Auditory Evoked Potentials", *Proceedings of the 13th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, Florida, July 10-13, Vol. III, , 2009, 45-50.

[11] A. Gutiérrez and A. Somolinos, Preprocessing of Brain Stem Auditory Evoked Potentials for Diagnosing Multiple Sclerosis, *Proceedings of the IASTED International Conference on Advances in Computer Science and Technology*, Puerto Vallarta, Mexico, 2006, 196-201.

[12] A. Gutiérrez and A. Somolinos, Extracting Features for Brain Stem Auditory Potential Signals, *Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, FL, 2004, 140-144.

[13] R. Gilad-Bachrach, A. Navot , and N. Tishby, Margin Based Feature Selection- Theory and Algorithms, *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Alberta, Canada, 2004, 43-50.

[14] K. Marse and S. D. Roberts, "Implementing a Portable FORTAN Uniform (0, 1) Generator", **Appl. Statist., 34**, 1983, pp. 135-139.

[15] S. Dehuri, and Sung'Bae Cho, "A comprehensive survey on functional link neural networks and an

adaptive PSO-BP learning for CFLNN", *Neural Comput. & Applic* (2010), 19:187-205.

[16] A. Sierra, J. A. Macias and F. Corbacho, "Evolution of Functional Link Neural Networks", *IEEE Trans. on Evolutionary Computation*, vol. 5, # 1, 2001, 54-65.

[17] B. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points", **SCIENCE, 315**, 2007, pp. 972-976

[18] V. Elser, I. Rankenburg, and P. Thibault, "Searching with iterated maps". **Proceedings of the National Academy of Sciences USA**. 2007, **104**, pp. 18-42