2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,
Proceedings, Companion Volume, pp. 389-398

Adaptive Multi-layer Neural Network Receiver Architectures for Pattern Classification of Respective Wavelet Images

Pythagoras Karampiperis[1], and Nikos Manouselis[2]

[1]Dynamic Systems and Simulation Laboratory
Dept. Production Engineering & Management
Technical University of Crete
pythk@dssl.tuc.gr
[2]Advanced eServices for the Knowledge Society Research Unit
Informatics and Telematics Institute (I.T.I.)
Center for Research and Technology – Hellas (CE.R.T.H.)
nikosm@iti.gr

Abstract. A difficult class of signal detection problems is detecting a non-stationary signal in a non-stationary environment with unknown statistics. One of the most interesting approaches considers the use of a neural network to compute the likelihood ratio of the received signal, by training it on different realizations of the received signal. The signal detection problem is then transformed to a pattern classification problem. It is still difficult though to determine the optimum internal structure of the neural network used, in order to achieve maximum performance of the receiver with less complexity of the network. In this paper, we demonstrate the use of self-organizing neural network. This network optimizes performance by re-configuring its internal structure regarding on whether the generalization results are satisfactory or not. The use of this network structure in the receiver architecture is also compared to a classic neural network approach of the signal detection problem.

1 Introduction

The Time Division Multiple Access (TDMA) modulation used in the Global System for Mobile communications (GSM) network requires the transmission of a training sequence consisting of 26 bits every 116 information bits, which represents wastage of about 23% throughput. Single efforts have been made to avoid use of such training sets, with the use of fixed multi-layer neural network architectures. In this paper we propose a self-organized multi-layer architecture for the design of receivers for TDMA wireless communications.

The new receiver architecture is based on the transformation of the detection problem into an adaptive pattern classification problem. This transformation enables a neural network to function as a powerful tool, which learns the underlying dynamics of a time-varying multi-path environment from data representative of that environment. This technique, originally proposed by Haykin *et al.* [1] can be altered in order

to achieve better performance than conventional Minimum Shift Keying (MSK) receivers for a Rayleigh fading multi-path channel, without the regular transmission of a training sequence.

Neural networks architectures most referenced in pattern recognition literature [4] are three: the multi-layer perceptron, the Kohonen associative memory and the Carpenter-Grossberg ART network. These networks implement algorithms of the major pattern classification paradigms: the multi-layer perceptron runs a supervised, parameter-learning algorithm the asymptotic behavior of which is that of an optimal Bayesian classifier; the Kohonen network performs vector quantization, mapping reference data onto a set of patterns representative of pattern category; the Carpenter-Grossberg network is motivated by biological relevance and brought to bear on computer-based pattern recognition, running an unsupervised algorithm that has similarities to leader clustering.

We are mainly focusing on the architecture proposed in [1] and enhancing it, by proposing an evolutionary adaptive receiver, based on a self-organizing multi-layered neural network. This architecture is different from those found in the associated literature [1], [2], [4], as it is based on a self-organized network architecture [5]. This architecture proves to provide better generalization results in pattern classification problems, compared to similar adaptive architectures.

2 Simulation Parameters

A simplified mobile communication system can be modeled with the use of a digital source, a modulator, the multi-path channel and the receiver under test. The receiver design basically involves three functional blocks: time-frequency analysis, data reduction or feature extraction and pattern recognition.

The desire in simulating the signal waveform for testing is to model the important elements of a signal, without unnecessary complications due to a particular protocol. The digital modulation system under study is a form of Frequency Shift Keying (FSK) called Minimum Shift Keying (MSK). The simulation signal parameters are representative of those in the GSM standard (Gaussian shaped pre-filtered MSK). The channel used here is in accordance with the GSM channel model. In particular, the multi-path channel is characterized by a time-varying impulse response $h(\tau,t)$ given by

$$h(\tau,t) = \sum_{1}^{\infty} \beta_i(t) e^{j\theta_i} \delta[\tau - \tau_i(t)] \ . \tag{1}$$

where $\beta_i(t)$ and $\theta_i(t)$ are time-varying amplitude and phase if the i^{th} path arriving at delay $\tau_i(t)$. Notice that $\beta_i(t)$, $\theta_i(t)$, $\tau_i(t)$ are in general random variables. However, in order to allow practical simulation, the path number is set to be finite in each of GSM channel models (rural area, hilly terrain and urban area) thereby allowing a tapped-delay line implementation. More specifically, the channel model consists of L taps (typically L=12), each of which is determined by a prescribed time delay $\tau_i(t)$, average

power P_i, and Rayleigh distributed amplitude varying according to a Doppler spectrum $S(f)$. Throughout this paper, we will use the urban area GSM model parameters.

The channel output is corrupted by additive noise that is assumed to be Gaussian, with zero-mean and variance σ^2. The received signal is then led into the receiver whose function is to detect the transmitted signal a_k which is multi-path (frequency selective) faded and noise corrupted.

Since the GSM channel model tap delays are multiples of 0.1 μs, a sample rate of 10MHz (100 ns sampling period) was used in the simulation. Taking 36 samples per bit yielded a bit period of 3.6 μs, thus a bit rate of about 278 KHz, representative of the GSM bit rate. The GSM system uses a training sequence to characterize the channel impulse response for a Viterbi receiver that considers a group of 5 consecutive bits at a time. Similarly, the receiver described in this paper operated on a sliding window block of 5 consecutive bits.

3 Signal Transformation

A noisy received signal can be represented in such a way that the signal components belonging to different classes (e.g. symbol 1 or symbol 0) have as more distinct representations as possible. Transforming the one-dimensional received signal into a two-dimensional image with time and frequency as coordinates is the idea used in this simulation. Since we are considering an FSK signal, the information is conveyed by a change in instantaneous frequency with time, so it is natural to examine time-frequency analysis methods for this transformation. From the two methods that were studied, that is Wigner-Ville transformation and wavelet analysis, the latter was chosen since it produced better receiver performance.

The wavelet used in the receiver is the Morlet wavelet, whose computation was carried out using the fast Fourier transform (FFT) algorithm. The squared amplitude of the Morlet wavelet, known as a scalogram, was used as the overall output of the time-frequency analyzer.

Since the wavelet image is highly redundant and considerably large, it is necessary to compress the image so that the design of the pattern classifier can be eased. However, we must ensure that the significant features contained in the image are extracted. Although principal components analysis (PCA) is widely used as a tool for feature extraction, it is proved that it cannot be satisfactorily used for the task at hand, since it is not particularly sensitive in changes in the instantaneous frequency, which is a major characteristic of the GMSK (Gauusian MSK) signal. Instead, a similar method is used, referred to as the energy profile. This method computes the energy values for a set of frequencies within the duration of one bit. Specifically, 5 scalogram values corresponding to scale bin 3 to 7, for each time index n, were used in computing the energy profile. Each bit's duration is divided into 4 segments, with each segment being associated with 9 samples. Then, with 5 bits, 4 time segments per bit and 5 frequency bins per bit, we have a total of 5x4x5=100 energy values, with each one being the result of adding 9 pertinent scalogram values. The motivation behind the use of multiple scalograms in computing the energy profile is to exploit the contextual information contained in a corresponding number of adjacent data bits.

4 The Network Receiver Architecture

The purpose of pattern classification is to recognize binary symbols 1 ad 0 by classifying the patterns in the respective wavelet images. In previous approaches [1], neural networks were used, consisting of multi-layer perceptrons trained with the back-propagation algorithm. Most of these approaches used static combinations of multi-layer perceptrons. In order to improve the generalization capability of the pattern classifier, we propose a self-organizing, multi-layer neural network, capable of adapting to the nature of the problem.

Reformulation of the signal detection problem as an adaptive pattern classification problem provides improved detection of a non-stationary target signal embedded in a non-stationary background. Pattern classification deals with assigning an unknown input pattern using supervised learning to one of several pre-specified classes, based on one or more properties that characterize the given class, as they were defined in the previous paragraph.

4.1 Network structure

The neural network used is a growing multi-layered perceptron, which begins from a basic structure of one node and one hidden layer and is then self-altering until it reaches the optimum structure for the given problem. The self-organization process consists of two phases: a growing one and a shrinking one (Figure 1). In the growing phase of self-optimization, two basic principles must always be valid:

- Every hidden layer has the same number of hidden nodes as the rest of the hidden layers.
- There are two ways of growing: horizontal (by incrementing the number of hidden layers) or vertical (by incrementing the number of hidden nodes).

The growth rule of the network is the optimization of the generalization error. At every step of the algorithm of growth, the following potential steps must be examined:

1. Calculate generalization error, using the current structure.
2. Calculate generalization error, after horizontal growth.
3. Calculate generalization error, after vertical growth.

Then the following conditions are examined:

- If horizontal growth is proven better than the vertical growth and the current structure, grow horizontally and return to the beginning.
- If vertical growth is proven better than the horizontal growth and the current structure, grow vertically and return to the beginning.
- If the current structure is proven better than the vertical growth and the horizontal growth, optimization stops.

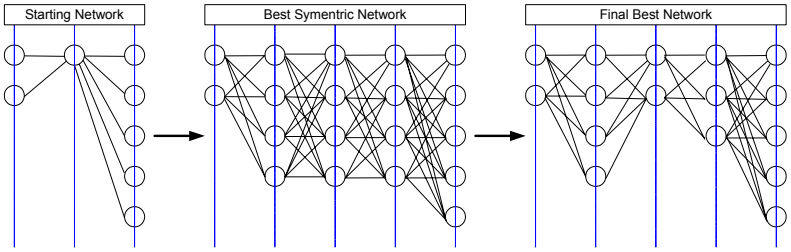

Fig. 1. A 2x4 network is first growing and then pruning in order to end up to the best possible architecture.

This is the growing algorithm that starts from the simple one node, one hidden layer perceptron and ends in a MLP network that gives optimum generalization. After the network structure is chosen, then pruning techniques are used in order to deduct certain nodes from this structure, with minimization of the generalization error. In our application we have used Optimal Brain Damage (OBD) as a pruning technique, but it is obvious that the growing phase of the algorithm does not depend on the chosen pruning algorithm.

4.2 Training algorithm

The network is consisted of neurons, which have an activation function of the form $\Phi_{(U)} = a\tan(bU)$, and locating the values of the elements of the network requires employing the back-propagation algorithm.

The feed-forward error back-propagation (BP) learning algorithm is the most famous procedure for training artificial neural networks (ANNs). BP is based on searching an error surface (error as a function of ANN weights) using gradient descent for point(s) with minimum error. Each iteration in BP constitutes two sweeps: forward activation to produce a solution, and a backward propagation of the computed error to modify the weights.

There has been much research on improving BP's performance. The Extended Delta-Bar-Delta (EDBD) variation of BP attempts to escape local minima by automatically adjusting step sizes and momentum rates with the use of momentum constants. To reduce the possibility of trapping into a local minimum even more, we use an extension of EDBD, which assumes that every node has a different activation function and every synaptic weight has its own learning rate. So we consider the following quantities as free parameters in each neuron:

- w: weight of every synaptic connection,
- a, b: activation function parameters,
- r_{wi}: learning rate of w_i,
- r_a: learning rate of a,

- r_b: learning rate of b.

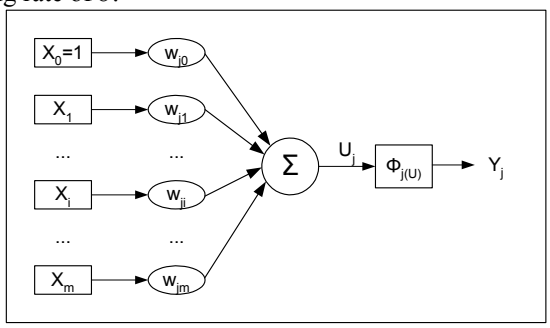

Fig. 2. Model for Neuron j

In order to avoid trapping into a local minimum, we adopted a momentum constant equal with (1-r_x) for every x free parameter of the network. The corrections of free parameters for epoch n become so:

$$\Delta w_{ji_{(n)}} = r_{w_{ji}} * \delta_j * X_i + (1 - r_{w_{ji}}) * \Delta w_{ji_{(n-1)}} \quad . \tag{2}$$

$$\Delta a_{j_{(n)}} = r_{a_j} * \delta_j * \frac{Y_j}{a_{j_{(n)}} * \Phi'_{j_{(U_j)}}} + (1 - r_{a_j}) * \Delta a_{j_{(n-1)}} \quad . \tag{3}$$

$$\Delta b_{j_{(n)}} = r_{b_j} * \delta_j * \frac{U_j}{b_{j_{(n)}}} + (1 - r_{b_j}) * \Delta b_{j_{(n-1)}} \quad . \tag{4}$$

where $r_{w_{ji}}$ is the learning rate parameter of $w_{ji}\ and\ \left(0 < r_{w_{ji}} < 1\right) \forall weight(j,i)$, r_{a_j} is the learning rate parameter of $a_j\ and\ \left(0 < r_{a_j} < 1\right) \forall node\ j$ and r_{b_j} is the learning rate parameter of $b_j\ and\ \left(0 < r_{b_j} < 1\right) \forall node\ j$.

Backpropagation is applied on the training set, with cost function the average square error, which must be minimized. As a training stopping criterion we use the generalization error. The average squared error (for minimization) can be calculated from:

$$E_{av} = \frac{1}{2TM} * \sum_{n=1}^{T} \left[\sum_{j \in C} \left[\sum_{k=1}^{M} \left(D_{j_{(n)}} - Y_{j_{(n)}} \right)^2 \right] \right]. \tag{5}$$

where:

- T is the total number of examples in the training set
- C is the set of all the neurons in the output layer

- M is the number of outputs.

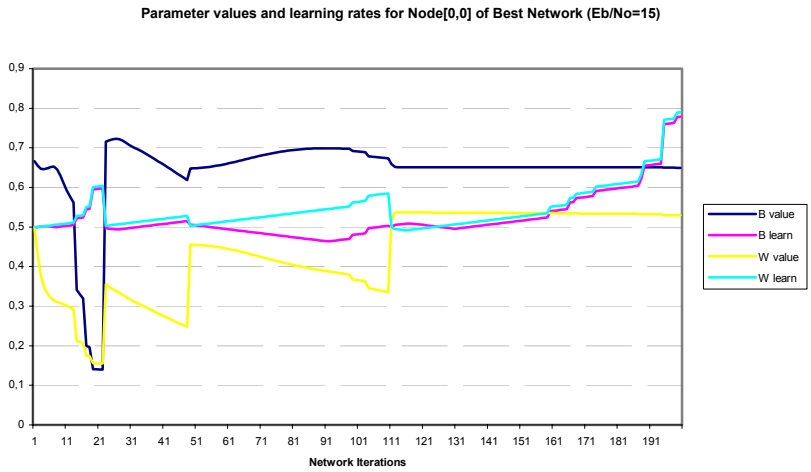

Fig. 3. Parameters and learning rates for the proposed structure, Node [0,0].

Every adjustable network parameter (free) of the cost function has its own learning rate parameter, given by:

$$R_{(n)} = -x_{(n)} \frac{\Delta x_{(n)}}{r_{x_{(n)}}}. \tag{6}$$

Rule 1. $(R_{(n+1)} > 0)\, and\, (R_{(n)} > 0)$ then increase r_x ($r_x = r_x + 0.001$)

Rule 2. $(R_{(n+1)} < 0)\, and\, (R_{(n)} < 0)$ then decrease r_x ($r_x = r_x - 0.001$)

As an initial value for every r_x we define 0.5. This is a value that can be changed as epochs pass, for each adjustable parameter. In Figure 3, some network parameters of a specific node and their corresponding learning rates are presented.

4.3 Initial data processing

Input data should be initially processed, in three stages (Figure 4):

1. Mean removal: mean values for each input node is removed, in order to centralize the original data values.
2. Decorrelation: the training set input data should be uncorrelated, so we use Principal Components Analysis at this stage.
3. Scaling: normalization is carried out in order to make covariances of the decorrelated input variables approximately equal.

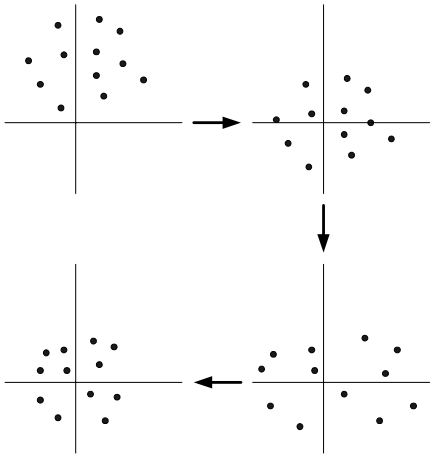

Fig. 4. Input data at the initial conditions and the three following stages of Mean Removal, Decorrelation and Scaling.

Splitting the training data set into estimation and validation subsets is originally such that 70% corresponds to training data and 30% to validation data. Since the growth algorithm is executed, an optimal network structure is chosen, where each hidden layer contains the same number of nodes with all the others. It is possible so to calculate the number W of the free parameters and re-define the validation subset according to the following formula:

$$V = [1 - \frac{\sqrt{2W-1}-1}{2(W-1)}] \cdot T_f \,. \tag{7}$$

where V is the validation set and T_f the full set of the training data. It is only after this split that we apply the pruning techniques, in order to increase generalization in this new validation set.

4.4 Initialization

The synaptic weights w_{ji} for neuron j are drawn from a uniformly distributed set of numbers with mean: $\mu_w = 0$ and variance: $\sigma_w = \frac{1}{\sqrt{m}}$, for all (j,i) pairs, where m is the number of synaptic connections of neuron j. Other initial values are learning rate: $r_x = 0.5$, for every adjustable network parameter x, parameter a: $a_j = 1.7159$ and parameter b: $b_j = \frac{2}{3}$, for every node j.

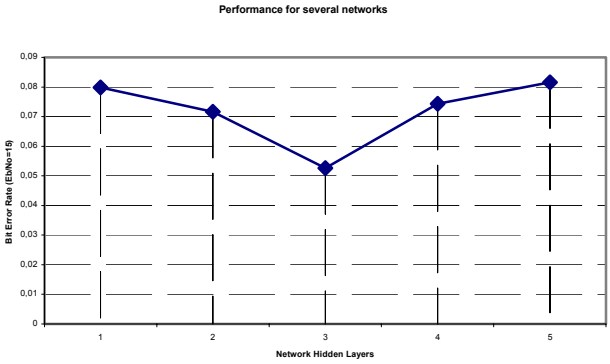

Fig. 5. The bit-error rate for several network structures with different number of hidden layers.

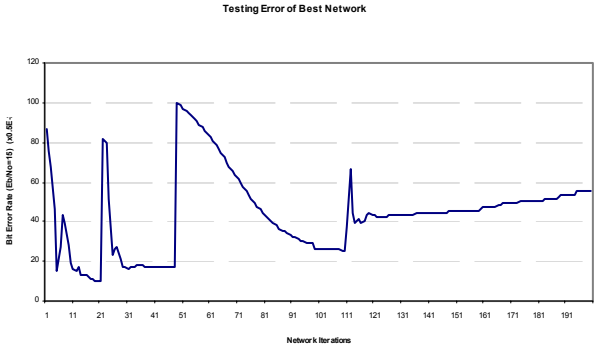

Fig. 6. Transforming the initial network to the best network found throughout several iterations.

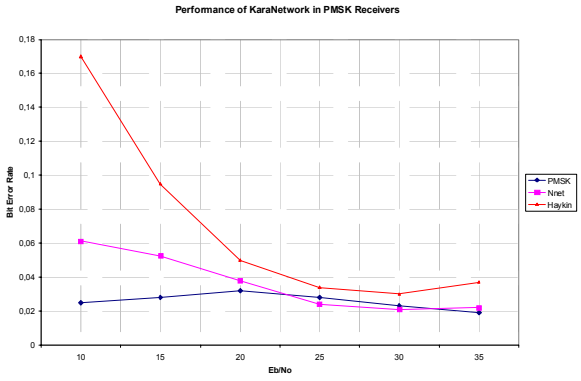

Fig. 7. Proposed Network (Nnet), Haykin *et al.* (Haykin) and PMSK receiver performance.

5 Results

We evaluated the architecture based on the proposed structure for the receiver, compared to the original neural network receiver structure tested by Haykin *et al*. in [1]. In Figure 5, the performance for several different cases of hidden-layers is shown, measured in bit-error rate. Figure 6 depicts the bit-error rate for the best network, according to the number of iterations of the training process of the specific network. In Figure 7 the bit-error rate for different values of *Eb/No* is presented, as far the Haykin *et al.* network and the proposed structure are concerned, for the classic PMSK receiver. It is clear that the use of the proposed network architecture enhances the receiver structure originally proposed by Haykin *et al.* [1] by substantially improving its performance. Moreover, it is clear in Figure 7 that when the value of *Eb/No* exceeds the value 25, performance can be compared even to that of the conventional PMSK receiver.

6 Conclusions

The transformation of a signal detection problem to a pattern classification problem is a technique found often in the literature, also providing very good results. In this paper we studied the neural network based receiver structure proposed by Haykin *et al.* and then substituted the classic MLP architecture with a specially designed adaptive architecture [5]. Results have proven that this self-organizing architecture greatly improves performance in such cases of pattern classification problems, and especially in the case of classification of respective wavelet images. Future research concerns dealing other pattern classification problems with the proposed architecture and extending it to other fields of practice that neural networks are used, as time-series prediction.

References

1. S. Haykin, J. Nie, B. Currie, “Neural Network-based Receiver for Wireless Communications”, Electronic Letters, Vol. 35, Issue 3, February 1999, pp. 203-205.
2. S. Haykin, D.J. Thomson, “Signal Detection in a Nonstationary Environment Reformulated as an Adaptive Pattern Classification Problem”, Proceedings of the IEEE, Vol. 86, Issue 11, November 1998, pp. 2325-2344.
3. D.T. Pham, S. Sagiroglu, “Training multilayered perceptrons for pattern recognition: a comparative study of four training algorithms”, International Journal of Machine Tools & Manufacture 41, 2001.
4. A. Mitiche, M. Lebidoff, “Pattern classification by a condensed neural network”, Neural Networks, Vol. 14, 2001, pp. 575-580.
5. P. Karampiperis, N. Manouselis, T. Trafalis, “Architecture selection for neural networks”, to appear in Proc. of IEEE World Congress on Computational Intelligence, Hawaii, May 2002.