High Reliability Neural Networks Structure with Application to Spacecraft ASMS Tone Detection

Kourosh Rahnamai
Western New England College
Electrical Engineering Department
1215 Wilbraham Road
Spring field, MA 01119 USA
krahnama@wnec.edu

John Maleyeff
Rensselaer Polytechnic Institute
Lally School of Mangement
275 Windsor St.,
Hartford, CT 06120
Maleyeff@rh.edu

Payman Arabshahi, Tsun-Yee Yan
Jet Propoulsion Laboratory/Caltech
Digital Signal Processing Research Group
4800 Oak grove drive, MS 238-343
Pasadena, CA 91109 USA
[payman,yah]@dsp.jpl.nasa.gov

Abstract

In this study we will show that the research on N-version high-reliability software structures can be extended to neural networks architecture. In addition, we will explore the possibility of applying this structure to a spacecraft tracking problem. One such system is the Automated Spacecraft Monitoring System (ASMS), a beacon-monitoring or detection system. Four neural networks, each trained for various operating environments, are implemented in an N-version structure. The results of the networks are combined to form a composite outcome. The combined outcome is used as part of a hypothesis testing procedure to distinguish between the presence or absence of the beacon signal. The results show that any of a number of composite outcomes outperforms the use of any single neural network. Further, the simple average of network results provides the composite outcome with best performance.

1. Introduction

Computer hardware and software have become an integral part of many sophisticated and complex systems, such as systems for space exploration. This trend has been the motivation for research efforts to improve software reliability and performance by introducing redundancy in computing hardware and/or software [1-7]. Multiple-version programming has been one such approach to fault-tolerant and high-reliability software development. In this method, software faults are compensated by concurrently executing N functionally-similar or -dissimilar software programs. By devising a voting scheme, the output of the programs is checked against each other for correctness. These static or dynamic consistency checking techniques have been subject of many research papers and reports [1-2]. This research has shown that N independently developed and designed programs will improve system reliability.

In this study, we will show that N-version high-reliability software structures can be extended to neural network architecture. Furthermore, we will show the advantages of this architecture over the conventional approach. In addition, we will explore the possibility of applying this structure to a spacecraft tracking problem. Artificial neural networks (ANN's) are suitable to N-version application structures for a number of reasons:

1. Many neural network paradigms, including backpropagation, use Least Mean Square (LMS) learning laws. These methods do not guarantee convergence of the network to the global minimum in the hyperspace of the network parameters. Subsequently, networks trained under similar conditions could produce different results.

2. In most cases, multilayer networks are trained with a subset of all possible input/output data sets. A network that is properly trained can interpolate and/or extrapolate correctly on inputs outside of the trained data set. In most applications, the goal is to train the network to achieve a balance between the ability of the network to respond correctly to the trained data set (*memorization*), and the ability to produce reasonable responses to input sets that are similar but not identical to the trained data set (*generalization*). Hence, properly trained networks using different design and convergence parameters will have the capability to extrapolate beyond trained data sets. This extrapolation feature will significantly improve the performance of the network under N-version structures. This improvement is the main reason for exploring N-version structures in a neural network environment.

3. If a particular application requires the neural network to operate over a wide range of design parameters, the network's training time may be excessive or impractical. For example, the spacecraft tone detection problem considered in this paper was first attempted with single-network architecture. The network did not converge because of the wide range of signal-to-noise ratios (SNR's).

In this paper, the authors present an N-version ANN approach with temporal history statistics to detect a tone in a very noisy environment. After some background is presented, the ANN structure is described. Then, the statistical methodology used to explore the use of several composite ANN outcomes is described and implemented.

2. Application

The era of the "New Millennium" discovery mission series brings NASA's Jet Propulsion Laboratory (JPL) to a revolutionary period in spacecraft design, deployment, and tracking paradigms. NASA has been planning to use micro-technology and -instrumentation, and more frequently launch smaller spacecraft at lower cost, with narrowly-focused missions. Since budget limitations prohibit increases in mission operations staff, this vision sets forth a tremendous challenge to introduce *intelligent automation* into all aspects of mission control, telemetry equipment, and ground tracking. One such system is the Automated Spacecraft Monitoring System (ASMS), a beacon-monitoring or detection system [9]. This system will lower the cost of Deep Space Network (DSN) operation in monitoring multiple spacecraft. Currently, large 70-meter antennas are used to track spacecraft during transmission phases as well as cruise phases. The proposed system will require tracking of spacecraft using antennas as small as 5 meters during cruise phases and antennas as large as 34 meters during transmission phases of the mission. In this system, the spacecraft transmits a carrier signal modulated by a square-wave subcarrier at one of four known frequencies to indicate the health status of the spacecraft. During routine operation, the spacecraft transmits a healthy beacon signal to the ground receiver on a regular basis at a prescribed time window. If the ground receiver detects the presence of the signal, the spacecraft is declared to be healthy. If the signal is not present, either due to spacecraft anomalies or ground equipment malfunctions, the station personnel will turn on the telemetry for possible downlink of emergency data. The fact that smaller spacecraft will have much less transmission power, compounded by the use of smaller antennas (lower SNR), will require an intelligent tone detection scheme in a very noisy environment.

The main objective of this work is to automatically recognize the presence of a spacecraft beacon or the telemetry downlink carrier signal buried in the noise using the output signal of a small tracking antenna's receiver amplifier.

3. Network Structure

Figure 1 shows the general structure of the ANN and the tone detection system. The tracking-antenna amplifier output signal is fed to a non-overlapping moving window block. The output of this block, which is a sequence of noisy data, is fed to the fast Fourier transform (FFT) block for conversion to frequency domain. For a sequence of data, *x(n)*, with N points, the FFT relationship is:

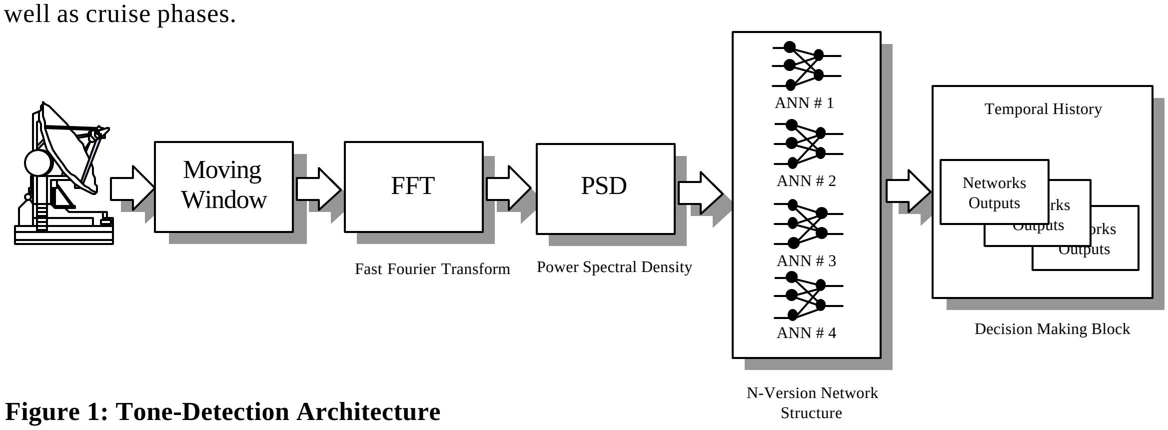

Figure 1: Tone-Detection Architecture

$$X(k) = \sum_{n=1}^{N} x(n) e^{-j2p(k-1)\left(\frac{n-1}{N}\right)} 1 \le k \le N \qquad (1)$$

Then, the power spectral density of each sequence is computed and the results are presented to the bank of neural networks.

The network block consists of 4 ANN's, each implementing a backpropagation paradigm. Networks 1 through 4 are designed to detect a carrier frequency in received signals with SNR of (greater than -3), (-3 to -9), (-9 to -12), and (-12 to -15) dB, respectively. Each network is trained using an adaptive learning rate to expedite the convergence process. Figure 2 shows a backpropagation network's general structure.

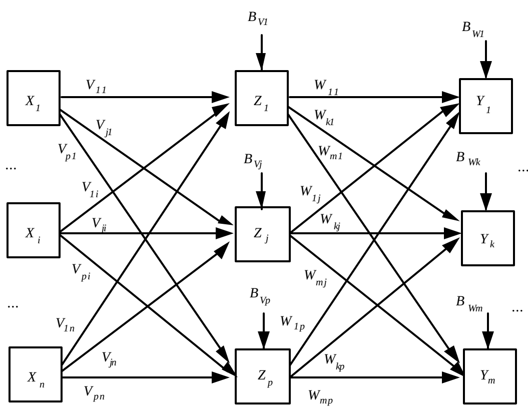

Figure 2: Structure of Each Network

with

$$\mathbf{V} = \begin{bmatrix} V_{11} & \cdots & V_{1i} & \cdots & V_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ V_{j1} & \cdots & V_{ji} & \cdots & V_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ V_{p1} & \cdots & V_{pi} & \cdots & V_{pn} \end{bmatrix}, \mathbf{W} = \begin{bmatrix} W_{11} & \cdots & W_{1j} & \cdots & W_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{k1} & \cdots & W_{kj} & \cdots & W_{kp} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{m1} & \cdots & W_{mj} & \cdots & W_{mp} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \qquad (2)$$

Where **X** is the input vector; X_i is the i^{th} unit input; **t** is the output target vector; α is the learning rate; B_{Vj} is the J^{th} hidden unit bias; Z_j is the j^{th} hidden unit output; B_{Wk} is the k^{th} unit bias; and Y_k representing the k^{th} unit output. Then with *f(.)* designating the activation function of each neuron, the network information propagation equations are:

$$z_{in_j} = B_{Vj} + \sum_{i=1}^{n} V_{ji} X_i = \mathbf{B}_V + \mathbf{V} * \mathbf{X} \qquad (3)$$

$$Z_j = f(z_{in_j}) \qquad (4)$$

$$y_{in_k} = B_{Wk} + \sum_{j=1}^{p} W_{kj} Z_j = \mathbf{B}_W + \mathbf{W} * \mathbf{Z} \qquad (5)$$

$$Y_k = f(y_{in_k}) \qquad (6)$$

Let $p(\mathbf{X})$, $p(\mathbf{X},t_j)$, and $p(t_j|\mathbf{X})$ signify the unconditional density function of the networks' input data, the joint probability density function of the training set $(\mathbf{X}, t_i)$, and the conditional probability density function of the training set respectively. Using these definitions the networks' error as shown in [8] can be written as:

$$E = \sum_{j=1}^{m} \int \left[Y_j(\mathbf{X}, \mathbf{W}, \mathbf{V}) - t_j \right]^2 p(\mathbf{X}, t_j) d\mathbf{X} dt_j \qquad (7)$$

using

$$\begin{aligned} & p(\mathbf{X}, t_j) = p(\mathbf{X}) * p(t_j \mid \mathbf{X}) \\ & E = \sum_{j=1}^{m} \int \left[Y_j(\mathbf{X}, \mathbf{W}, \mathbf{V}) - \langle t_j \mid \mathbf{X} \rangle \right]^2 p(\mathbf{X}) d\mathbf{X} + \\ & \sum_{j=1}^{m} \int \left[\langle t_j^2 \mid \mathbf{X} \rangle - \langle t_j \mid \mathbf{X} \rangle^2 \right] p(\mathbf{X}) d\mathbf{X} \end{aligned} \qquad (8)$$

The first term of Equation 8 is dependent on network weights, and for minimum value of the network error, *E*, this term should vanish. Also note that the residual of this equation is the variance of the training data set. This important result is used to properly set the termination parameters during training phases of the networks.

For training purposes, the data stream is partitioned into non-overlapping windows, and the power spectral density of each time series is calculated before presenting them to the neural networks. A supervised backpropagation paradigm based on gradient descent is used to train each network with a single output assuming two integer target values: 10 indicating the presence of tone and -10 indicating only noise. During simulation and operational phase of the network the input data is prepared in a similar fashion to the training phases. Output values of the neural networks are collected over a fixed-length time interval and a statistical method is devised for final decision making.

4. Statistical Methodology

A hypothesis testing routine is employed to detect the presence of a signal in a noisy environment. The procedure incorporates the following hypotheses:

H_0: No signal exists
H_1: A signal exists

In the system described, the method would statistically evaluate the one or more ANN scores, by calculating a *test statistic* that would be compared to a critical value. If the test statistic exceeds the *critical value*, the decision would be to Reject H_0 in favor of H_1 and conclude that a signal exists. The most effective test statistic and critical value would be chosen such that both the probability of a Type I error (rejecting H_0 when no signal exists) and the probability of a Type II error (failing to reject H_0 when a signal exists) are minimized. For this system, the most effective test statistic and associated critical values are determined empirically, based on MATLAB simulations of various noise environments. The following steps were followed:

1. Characterization of the statistical behavior of scores resulting from each ANN
2. Determination of the degree of cross-correlation among the ANN scores
3. Generation of a variety of potential composite test statistics
4. Characterization of the statistical behavior of each composite test statistic and determination of corresponding critical values
5. Analysis of the operating characteristic (OC) of each test statistic

4.1 Analysis of Network Scores

It is assumed that a data stream consisting of pure noise will follow a Gaussian distribution with mean zero and variance σ^2. Since the statistical behavior of the ANN scores may depend on the variance of the noise, simulations were written to generate network scores over a range of noise levels. The objective of the statistical analysis was to determine, for each ANN, whether or not a relationship existed between the noise level and the scores, and if so, to determine the nature of the relationship.

The average network score across data stream standard deviation (σ) values is shown in Figure 3, while the standard deviation of network scores across data stream σ values is shown in Figure 4. It is clear that not only do the average and standard deviation of network scores vary over the four networks, but, for each network, the average and standard deviation of network score varies across the range of data stream variation.

The distribution of network scores was analyzed and changes in distribution over the range of data stream σ values were evaluated. All data streams were in statistical control over time, which was verified by

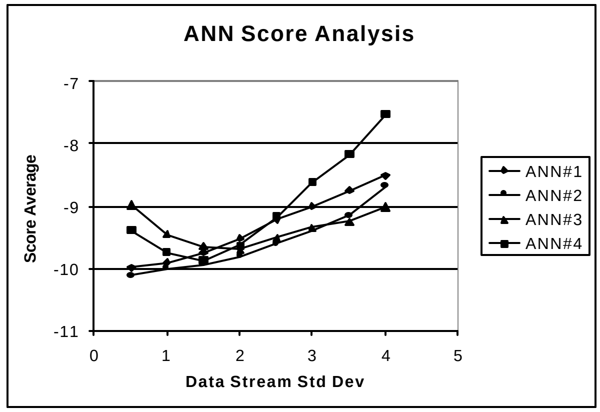

Figure 3: Average of ANN Scores

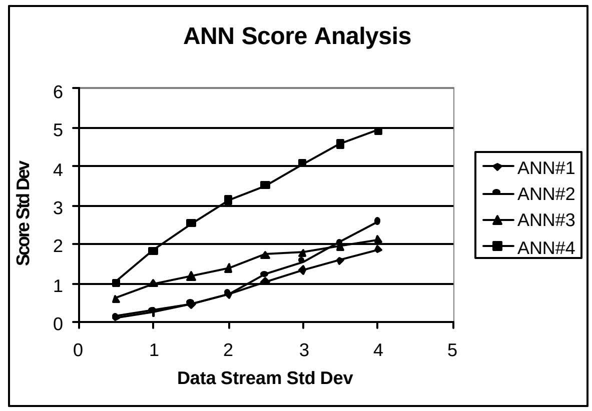

Figure 4: Standard Deviation of ANN Scores

analysis of the autocorrelation of network scores for consecutive data streams using time series analysis procedures. ANN score histograms followed patterns as illustrated in Figures 5 and 6. The examples shown are typical of the all distributions analyzed in terms of their positive skew.

4.2 Cross-Correlation of ANN Scores

The cross-correlation of ANN scores among the networks was evaluated to determine if network scores for similar data streams provide mutually independent outcomes. It was evident from scatter plots that network scores are correlated. Table 1 provides a sample of Pearson product moment correlation coefficients for each pair of networks scores. Correlation coefficients for scores from other data sets yielded similar results. As a result of this analysis, it cannot be assumed that the network scores provide independent scores when evaluating a data set.

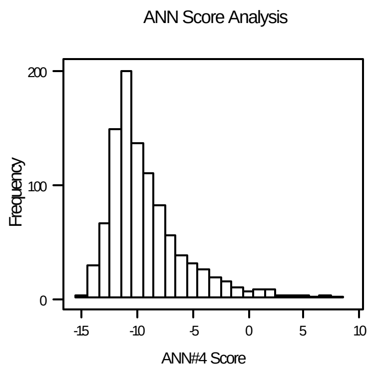

Figure 5: Sample Histogram of ANN Scores (Data Stream Std Dev=2.5)

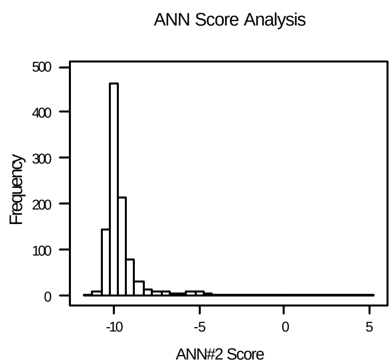

Figure 6: Sample Histogram of ANN Scores (Data Stream Std Dev=2.5)

	ANN#		
ANN#	1	2	3
2	0.637		
3	0.554	0.581	
4	0.476	0.446	0.530

Table 1: Sample Correlation Coefficients (Data Stream Std Dev=2.5)

4.3 Determination of Composite Test Statistics

As a result of the analysis above, it can be concluded that any hypothesis testing procedure must use the standard deviation of the input data to determine the expected distribution of scores for each network. The plots in Figures 3 and 4 indicate the need for second-order regression equations. The equations were developed and used to determine expected values and variances of ANN scores given the input data set standard deviation.

The central limit theorem can be used to characterize the distribution of each network score under the noise assumption (H_0 true). Assuming that 4 data streams are evaluated before a decision is made, the distribution of average scores from neural network model i can be expected to follow a near-normal distribution. That is, for network i, the following statistic approximates a standard normal random variable:

$$Z_i = \frac{\overline{X}_i - m_i}{s_i / \sqrt{4}} \qquad (9)$$

This statistic can be used to determine if the assumption of pure noise should be rejected. The mean of the standardized Z-scores for each ANN as well as other test statistics (the median, and the four order statistics) were considered below.

4.4 Statistical Analysis of Composite Test Statistics

Each composite statistic's average and standard deviation were used to explore its effectiveness. Limits for rejecting H_0 are determined to ensure that the probability of falsely claiming a signal exists (a type I error) is less than 1%.

4.5 Testing OC of Composite Test Statistics

The procedure was tested using the seven potential test statistics to determine the one that performed best. The results are shown in Figure 7. Figure 7 shows the probability of accepting H_0 for a range of data stream variation given that a tone exists. This display shows the Type II error probability, which should be as low as possible for a procedure to be powerful. In Figure 7, Z_j is the score for the properly trained ANN, Zbar and median are the average and median Z-score, and Z(1)-Z(4) are the four order statistics.

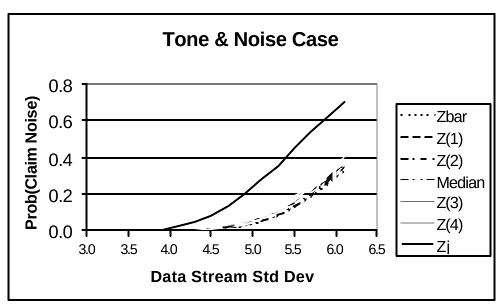

Figure 7: Ability to Detect Tone

It is clear that any of the composite statistics perform better than the use of a single neural network score. As shown in Figure 7, the use of the average score outperforms the other composite test statistics. While the improvement over the other test statistics is relatively small, this improvement has been shown to be statistically significant.

References

[1] Caglayan, A.K., and G.L. Zacharias, "Study of Fault Tolerant Software Technology for Dynamic Systems," Charles River Analytics Inc., NASA Contractor Report 172618, Cambridge, Massachusetts, USA, September 1985.

[2] Lorczak, P.R., and A.K. Caglayan, "A Large Scale Second Generation Experiment in Multiversion Software: Testing for Version Reliability," Charles River Analytics Inc., Report R8806, Cambridge, Massachusetts, USA, July 1988.

[3] Rahnamai, K.R., Caglayan, A.K., and D. Moerder, "Detection and Identification of Control Impairment in a Self-Repairing Flight Control System," *Proceedings of the National Aerospace and Electronics Conference*, Dayton, OH, May 1986.

[4] Rahnamai, K.R., Caglayan, A.K., Moerder, D., and N. Halyo "A Hierarchical Reconfiguration Strategy for Aircraft Subjected to Actuator Failure/Surface Damage," Report (AFWAL-TR- 87-3024), Flight Dynamics Laboratory, Wright-Patterson Air Force Base, March 1987.

[5] Rahnamai, K.R., Caglayan, A.K., and C.L. Stubbs, "Enforced Software Diversity For A Tracking Application," Final Report No. R8908 for NASA Contract No. NAS1-17705, August 1989.

[6] Rahnamai, K.R., "Situation Awareness Metric For Cockpit Configuration Evaluation" (with G.L. Zacharias), Report No. R8910 for Dr. Michael E. Venturino, AAMRL/HEA, Wright-Patterson Air Force Base, December 1989.

[7] Rahnamai, K.R., Huang, C., Allen, S.M., and A.K. Caglayan "Control Reconfigurable Combat Aircraft", Report (AFWAL-TR-88-3118), Flight Dynamics Laboratory, Wright-Patterson Air Force Base, June 1990.

[8] Bishop, C.M., "Novelty Detection and Neural Network Validation," IEE 94

[9] Lanyi, G., and R. Kahn, "Tome Detection Via Incoherent Averaging of Fourier Transforms to Support the Automated Spacecraft-Monitoring Concept," TDA Progress Report 42-129, Jet Propulsion Laboratory, Pasadena, California, May 15, 1997.