Letters

Analysis of the Effects of Quantization in Multilayer Neural Networks Using a Statistical Model

Yun Xie and Marwan A. Jabri

Abstract— **A statistical quantization model is used to analyze the effects of quantization when digital techniques are used to implement a real-valued feedforward multilayer neural network. In this process, we introduce a parameter that we call the effective nonlinearity coefficient, which is important in the studying the quantization effects. We develop, as functions of the quantization parameters, general statistical formulations of the performance degradation of the neural network caused by quantization. Our formulations predict (as intuitively one may think) that the network's performance degradation gets worse when the number of bits is decreased; that a change of the number of hidden units in a layer has no effect on the degradation; that for a constant effective nonlinearity coefficient and number of bits, an increase in the number of layers leads to worse performance degradation of the network; and that the number of bits in successive layers can be reduced if the neurons of the lower layer are nonlinear.**

I. Introduction

One of the first problems in the hardware implementation of real-valued artificial neural networks is determining how many bits are necessary to represent physical states, parameters, and variables in order to ensure certain learning and generalization performance.

In the analysis of the effects of quantization on learning and generalization, one of the complications is the nonlinear transfer functions commonly used to represent neurons. Assumptions on the distributions of the neuron responses are necessary to relate quantization widths in different layers.

Not much work has been done in this area, although efforts that may be of interest include those that target the determination of a relationship between the number of hidden units, the number of hidden layers, and the learning and generalization capability of a network [1].

In this paper we use statistical models to represent the quantized values of weights and neurons in a feedforward multilayer neural network, and given certain assumptions on their distributions, we investigate a relationship between bit resolution, number of hidden units and layers, and the performance degradation of the network. The paper is structured as follows. In Section II we define our statistical model and some terminology and briefly discuss the assumptions we make on the neural network. Section III develops relationships between inputs and outputs, based on our statistical models. The assumptions on the neural network are stated. We also introduce in this section the "effective nonlinearity coefficient," which plays an important role in the understanding of the effect of nonlinear units on bit resolution. In Section IV we reveal and analyze the relationship between bit resolution, network architecture, and the performance degradation of the network. Section V presents some simulation results. Finally, in Section VI some conclusions are given.

II. Statistical Model of Quantization

A. Statistical Model

The effects of quantization can be approached in many ways. For complex systems, the statistical model is a convenient method [3]. The idea is that a quantized signal can be represented by the original signal plus a quantization error (noise), $e(n)$

. We make the following assumptions:

- $e(n)$ is a stationary random process;
- $e(n)$ is independent of the signal;
- $e(n)$ is a white noise;
- Δ is the quantization width, $e(n)$ is uniformly distributed in $[-\Delta, \Delta]$; thus the mean is zero and the variance is $\Delta^2/12$, denoted by σ_Δ^2.

In the following, $E\{z\}$ represents the expectation of z and σ_z^2 denotes the variance of z.

B. Assumptions

In order to investigate the effects of quantization using the statistical model, we have to make some assumptions on the input of the neurons and the weights. In our analysis, we assume that all the inputs, the weights, and the weighted sum of the input of a neuron are distributed uniformly in certain ranges. In reality, the real distributions of these values varies with the applications at hand and from one network architecture to another. Since it is impossible to identify a universal distribution to all sort of problems and architectures, one hopes to make some assumptions and validate them through simulation on several data sets and architectures. The reason why we selected a uniform distribution in contrast to other sorts of distributions (normal for instance) is that it enables us to develop closed-form formula for the quantization effects in a network with nonlinearities. The nonlinear aspect of the network is one of its most distinguished feature. The conclusions reached in this paper on the basis of the uniform distribution have been confirmed by several simulations that we have conducted on intracardia electrogram classification. If one is to select more complicated distributions, for example normal distributions, more assumptions and approximations will have to be made to deal with the propagation of the distributions through the nonlinear network in order to obtain a clear formulation of the quantization effects.

III. Effects of Quantization

A. First Hidden Layer

In the following, x_k^0 is the input signal from input node k, w_{ik}^0 is the weight connecting node k in the input layer and node i in the first hidden layer, and y_i^0 is the input of node i while x_i^1 is the output:

$$y_i^0 = \sum_{k=0}^{K_1-1} w_{ik}^0 x_k^0 \qquad (1)$$

Manuscript received May 9, 1991. This work was supported in part by the Australian Research Council and the Australian Telecommunication and Electronic Research Board. The first author acknowledges the support of the National Education Committee of the People's Republic of China.

Y. Xie was with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, People's Republic of China. He is now with the Department of Electrical Engineering, University of Sydney, N.S.W. 2006, Australia.

M. A. Jabri is with the Department of Electrical Engineering, University of Sydney, N.S.W. 2006, Australia.

IEEE Log Number 9104050.

1045-9227/92$03.00 © 1992 IEEE

$$x_i^1 = f(y_i^0). \tag{2}$$

The quantity $f(\cdot)$ is the nonlinear transfer function of a node and the bias is treated as an input.

Assuming that x_k^0 and w_{ik}^0 are quantized by N bits (one bit for sign), that Δ_0 is the quantization width, and that the quantized value falls in $\left[-\Delta_0(2^{N-1}-1), \Delta_0(2^{N-1}-1)\right]$, which can be approximated to $\left[-\Delta_0 2^{N-1}, \Delta_0 2^{N-1}\right]$ when N is large enough, then

$$y_i^0 = \sum_{k=0}^{K_1-1} \left(w_{ik}^0 + \Delta w_{ik}^0\right)\left(x_k^0 + \Delta x_k^0\right) \simeq \sum_{k=0}^{K_1-1} w_{ik}^0 x_k^0 + \Delta y_i^0 \tag{3}$$

$$\Delta y_i^0 \overset{\text{def}}{=} \sum_{k=0}^{K_1-1} w_{ik}^0 \Delta x_k^0 + \sum_{k=0}^{K_1-1} x_k^0 \Delta w_{ik}^0 \tag{4}$$

where the second-order items are omitted. The quantities Δx_k^0 and Δw_{ik}^0 are quantization noises and are independent of each other, w_{ik}^0 and x_k^0. Assuming x_k^0 and w_{ik}^0 are uniformly distributed in $\left[-\Delta_0 2^{N-1}, \Delta_0 2^{N-1}\right]$ and independent of each other, we have

$$E\{\Delta y_i^0\} = 0 \tag{5}$$

$$E\left\{\left(x_k^0\right)^2\right\} = E\left\{\left(w_{ik}^0\right)^2\right\} = \int_{-\Delta_0 2^{N-1}}^{\Delta_0 2^{N-1}} \frac{x^2}{\Delta_0 2^N}\, dx = \frac{\Delta_0^2 2^{2N}}{12} \tag{6}$$

$$\sigma_{\Delta y_i^0}^2 = \left(\sum_{k=0}^{K_1-1} E\left\{\left(x_k^0\right)^2\right\} + \sum_{k=0}^{K_1-1} E\left\{\left(w_{ik}^0\right)^2\right\}\right)\sigma_{\Delta_0}^2 = \zeta_0 K_1 \Delta_0^4 2^{2N} \tag{7}$$

$$\zeta_0 = 1/72. \tag{8}$$

In order to relate quantization widths of different layers, we have to make some approximations on the distribution of y_i^0:

$$\sigma_{y_i^0}^2 = \sum_{k=0}^{K_1-1} E\left\{\left(x_k^0\right)^2\left(w_{ik}^0\right)^2\right\} = \left(K_1\Delta_0^4 2^{4N}\right)/144 \tag{9}$$

where noise is omitted.

We define

$$\max\left|y_i^0\right| = \sqrt{3\sigma_{y_i^0}^2} = \eta_0\sqrt{K_1}\,\Delta_0^2 2^{2N} \tag{10}$$

$$\eta_0 = 1/\left(4\sqrt{3}\right). \tag{11}$$

The distribution of y_i^0 is approximated by a uniform distribution in $\left[-\max\left|y_i^0\right|, \max\left|y_i^0\right|\right]$. When K_1 is very large, a normal distribution is a better approximation, but this is not true when K_1 is not large enough and makes subsequent analysis very difficult and necessitates further approximations in order to obtain a clear formulation of the quantization effects.

The uniform distribution we use gives the same variance (power) of $\sigma_{y_i^0}^2$; it is the second-order approximation.

For simplicity, we use the nonlinear transfer function as

$$y = f(x) = \begin{cases} \Delta_1 2^{N-1}, & x > \Delta_1 2^{N-1} \\ x, & \text{between} \\ -\Delta_1 2^{N-1}, & x < -\Delta_1 2^{N-1}. \end{cases}$$

Here Δ_1 and N are the quantization width and the number of bits respectively in the quantization of the outputs of the first hidden layer and the weights between the first and the second hidden layers.

We define the *effective nonlinearity coefficient* of the nodes in the first hidden layer as follows:

$$E_1 \overset{\text{def}}{=} \frac{\max\left|y_i^0\right|}{\Delta_1 2^{N-1}} = 2\eta_0\sqrt{K_1}\, 2^N \frac{\Delta_0^2}{\Delta_1} \tag{12}$$

$$\Delta_1 = \frac{2\eta_0\sqrt{K_1}\,\Delta_0^2 2^N}{E_1} \tag{13}$$

$$\sigma_{\Delta_1}^2 = \frac{\eta_0^2}{3\zeta_0 E_1^2}\,\sigma_{\Delta_{y_i^0}}^2 \tag{14}$$

Since neural networks are generally nonlinear, E_1 is greater than unity.

Now we can derive the distribution of x_i^1. The probability density of x_i^1 in $\left(-\Delta_1 2^{N-1}, \Delta_1 2^{N-1}\right)$ is

$$p_{x_i^1} = \frac{1}{\Delta_1 2^N E_1} \tag{15}$$

and

$$P\left(x_i^1 \in \left(-\Delta_1 2^{N-1}, \Delta_1 2^{N-1}\right)\right) = \frac{1}{E_1} \tag{16}$$

$$P\left(x_i^1 = \Delta_1 2^{N-1}\right) = P\left(x_i^1 = -\Delta_1 2^{N_1}\right) = \frac{E_1 - 1}{2E_1}. \tag{17}$$

The above two equations offer a method to measure E_1 in a neural network. We can also consider it as another definition form of E_1.

B. Second Hidden Layer

The quantity x_l^1 is the input signal from the first hidden layer; w_{il}^1 is the weight connecting node l in the first hidden layer and node i in the second hidden layer; and y_i^1 is the input of node i in the second hidden layer and x_i^2 is its output:

$$y_i^1 = \sum_{l=0}^{K_2-1} w_{il}^1 x_l^1 \tag{18}$$

$$x_i^2 = f(y_i^1). \tag{19}$$

Just as in the first hidden layer, because to quantization,

$$y_i^1 = \sum_{l=0}^{K_2-1} \left(w_{il}^1 + \Delta w_{il}^1\right)\left(x_l^1 + \Delta y_l^1\right) \simeq \sum_{l=0}^{K_2-1} w_{il}^1 x_l^1 + \Delta y_i^1 \tag{20}$$

$$\Delta y_i^1 \overset{\text{def}}{=} \sum_{l=0}^{K_2-1} w_{il}^1 \Delta x_l^1 + \sum_{l=0}^{K_2-1} x_l^1 \Delta w_{il}^1 \tag{21}$$

where Δw_{il}^1 is the quantization noise with zero mean and variance of $\sigma_{\Delta_1}^2 = \Delta_1^2/12$. Because of nonlinearity of $f(\cdot)$,

$$\Delta x_l^1 = \begin{cases} \text{QN} + \Delta y_l^0, & x_l^1 \in \left(-\Delta_1 2^{N-1}, \Delta_1 2^{N-1}\right) \\ 0, & x_l^1 = -\Delta_1 2^{N-1}, \text{ or } \Delta_1 2^{N-1} \end{cases}$$

where QN is the quantization noise. This means E_1^{-1} percent of x_l^1 has $\Delta x_l^1 \neq 0$.

Assuming the distribution of w_{il}^1 is uniform in $\left[-\Delta_1 2^{N-1}, \Delta_1 2^{N-1}\right]$,

$$\sigma_{\Delta y_i^1}^2 = \frac{K_2}{E_1} E\left\{\left(w_{il}^1\right)^2\right\}\sigma_{\Delta x_l^1}^2 + K_2 E\left\{\left(x_l^1\right)^2\right\}\sigma_{\Delta_1}^2 = \zeta_1 K_2 \Delta_1^4 2^{2N} \tag{22}$$

$$\zeta_1 \overset{\text{def}}{=} \frac{1}{48}\left(\frac{\zeta_0 E_1}{\eta_0^2} - \frac{1}{3E_1} + 1\right) \tag{23}$$

$$\sigma_{y_i^1}^2 = \sum_{l=0}^{K_2-1} E\left\{\left(w_{il}^1\right)^2\right\}E\left\{\left(x_l^1\right)^2\right\} = \frac{K_2\Delta_1^4 2^{4N}}{48} - \frac{K_2\Delta_1^4 2^{4N}}{72E_1}. \tag{24}$$

Just as in the first hidden layer,

$$\max\left|y_i^1\right| \overset{\text{def}}{=} \sqrt{3\sigma^2_{y_i^1}} = \eta_1\sqrt{K_2}\,\Delta_1^2 2^{2N} \tag{25}$$

$$\eta_1 \overset{\text{def}}{=} \sqrt{\frac{1}{16} - \frac{1}{24E_1}}\,. \tag{26}$$

The effective nonlinear coefficient of the second hidden layer is defined as

$$E_2 \overset{\text{def}}{=} \frac{\max\left|y_i^1\right|}{\Delta_2 2^{N-1}} = \frac{2\eta_1\sqrt{K_2}\,\Delta_1^2 2^N}{\Delta_2} \tag{27}$$

where Δ_2 and N are the quantization width and the number of bits in the quantization of the outputs of the second hidden layer and the weights between the second and the third hidden layers. From the above equation,

$$\Delta_2 = \frac{2\eta_1\sqrt{K_2}\,\Delta_1^2 2^N}{E_2} \tag{28}$$

$$\sigma^2_{\Delta_2} = \frac{\eta_1^2}{3\zeta_1 E_2^2}\,\sigma^2_{\Delta y_i^1}. \tag{29}$$

C. Generalization to n Layers

Following assumptions and approximations similar to those made above, we can generalize the above analysis to higher hidden layers.

For the nth hidden layer, y_i^{n-1} is the input of node i, K_n is the number of nodes, E_n is the effective nonlinearity coefficient, and Δ_{n-1} and N are the quantization width and the number of bits in the quantization of the outputs of the $(n-1)$th hidden layer and the weights between the $(n-1)$th and the nth hidden layers. Thus,

$$E_n = \frac{2\eta_{n-1}\sqrt{K_n}\,\Delta_{n-1}^2 2^N}{\Delta_n} \tag{30}$$

$$\eta_n = \sqrt{1/16 - 1/(24E_n)} \tag{31}$$

$$\zeta_n = \frac{1}{48}\left(\frac{\zeta_{n-1}E_n}{\eta_{n-1}^2} - \frac{1}{3E_n} + 1\right) \tag{32}$$

$$\sigma^2_{y_i^{n-1}} = K_n\Delta_{n-1}^4 2^{2N}\eta_{n-1}^2/3 \tag{33}$$

$$\sigma^2_{\Delta y_i^{n-1}} = \zeta_{n-1}K_n\Delta_{n-1}^4 2^{2N}. \tag{34}$$

IV. Results Analysis

A. Signal to Noise Ratio

We can define the signal to noise ratios in the second and the third hidden layers. In the second hidden layer, the signal to noise ratio, R_2^p, is

$$R_2^p \overset{\text{def}}{=} \frac{\sigma^2_{y_i^1}}{\sigma^2_{\Delta y_i^1}} = \frac{\eta_1^2 2^{2N}}{3\zeta_1}\,. \tag{35}$$

Similarly, in the third hidden layer,

$$R_3^p = \frac{\eta_2^2 2^{2N}}{3\zeta_2}\,. \tag{36}$$

We define

$$\max\left|\Delta y_i^1\right| = \sqrt{3\sigma^2_{\Delta y_i^1}}\,. \tag{37}$$

First we consider the case where the second layer is the output layer. We assume Δy_i^1 has a uniform distribution in $[-\max\left|\Delta y_i^1\right|, \max\left|\Delta y_i^1\right|]$; the output nodes of the network are assumed to have the function

$$y = f(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0. \end{cases}$$

Now we can calculate the probability, P_2^f, that an output node gives a result different from the output when no quantization is involved:

$$\begin{aligned} P_2^f &= 2\int_0^{\max\left|\Delta y_i^1\right|} \frac{1}{2\max\left|y_i^1\right|}\int_{-\max\left|\Delta y_i^1\right|}^{-x} \frac{1}{2\max\left|\Delta y_i^1\right|}\,dy\,dx \\ &= \frac{\max\left|\Delta y_i^0\right|}{4\max\left|y_i^1\right|} = \frac{1}{4\sqrt{R_2^p}} \end{aligned} \tag{38}$$

If the third layer is the output layer, the results are similar:

$$P_3^f = \frac{1}{4\sqrt{R_3^p}}\,. \tag{39}$$

It is clear that the signal to noise ratio is directly associated with the performance degradation of the network.

B. The Effects of Number of Bits

From (35) and (36) it is clear that increasing the number of bits results in an improvement of signal to noise ratios and that the ratio is independent of the number of nodes in each layer. This means that if we change the number of nodes in a layer the signal to quantization noise ratio in the output of the layer remains the same, although the network's performance may or may not degrade.

Then, is it necessary for all the layers to have the same number of bits? Let us consider the second layer. Suppose we use M bits to represent the output of the first hidden layer. Then (12) will become

$$E_1 = \frac{\max\left|y_i^0\right|}{\Delta_1 2^{M-1}} = \frac{2\eta_0\sqrt{K_1}\,\Delta_0^2 2^{2N-M}}{\Delta_1} \tag{40}$$

$$\Delta_1 = \frac{2\eta_0\sqrt{K_1}\,\Delta_0^2 2^{2N-M}}{E_1}\,. \tag{41}$$

From (7) and (41) we have

$$\sigma^2_{\Delta_1} = \frac{\Delta_1^2}{12} = \frac{\eta_0^2 2^{2N-2M}}{3\zeta_0 E_1^2}\,\sigma^2_{\Delta y_i^0}. \tag{42}$$

$\sigma^2_{\Delta y_i^0}$ being the variance of the input noise to the second hidden layer and $\sigma^2_{\Delta_1}$ the variance of the quantization noise produced by the second hidden layer.

If we want $\sigma^2_{\Delta_1} = \sigma^2_{\Delta y_i^0}$, which means that the power of newly introduced noise in the output of the first hidden layer is the same as the power of the original noise in the input (this requirement is commonly used in signal processing systems), then we get

$$M = N - \frac{1}{2} - \log_2 E_1. \tag{43}$$

If N bits are used for the quantization of the output of the $(n-1)$th hidden layer and the weights connecting this layer to the following and if the number of bits used for the quantization of the output of the nth hidden layer and the noise passed to the output of the $(n-1)$th hidden layer from lower layers is ignored, a similar equation can be developed:

$$M = N - \log_2 E_n - \log_2\left(\frac{3E_{n-1} - 1}{3E_{n-1} - 2}\right). \tag{44}$$

Let $n = 1$ and $E_0 = 1$; then (44) becomes (43). Since $E_n > 1$, $N < M$. The greater E_n, the smaller M is compared with N. The

TABLE I
EFFECTS OF THE NUMBER OF LAYERS ON SIGNAL TO NOISE RATIO

	2 layer	3 layer
$E_1 = E_2$	$R_2^p/2^{2N}$	$R_3^p/2^{2N}$
2	0.30	0.22
4	0.23	0.13
6	0.18	0.07

TABLE II
TRAINING RESULTS OF CS ON ICEG DATA WITH ONE HIDDEN LAYER

Number of Bits	Average Performance on Training Sets (%)	Standard Deviation on Training Sets (%)	Average Performance on Testing Sets (%)	Standard Deviation on Testing Sets (%)
10	98.7	0.712	90.0	1.09
9	98.4	0.771	89.6	0.978
8	98.5	0.740	89.2	1.46
7	97.5	2.24	87.7	5.29
6	95.7	2.70	86.4	5.85
5	89.2	8.80	81.2	8.63

TABLE III
TRAINING RESULTS ON ICEG DATA WITH 9 BIT RESOLUTION FOR INPUT AND ONE HIDDEN LAYER ARCHITECTURE

Number of Bits for Upper Layers	Average Performance on Training Sets (%)	Standard Deviation on Training Sets (%)	Average Performance on Testing Sets (%)	Standard Deviation on Testing Sets (%)
9	98.4	0.771	89.6	0.978
8	98.2	0.950	89.7	0.908
7	97.1	0.646	89.6	1.03
6	95.3	1.30	87.4	1.66

reason for this is that as E_n increases, less unprocessed information is passed to the upper layer, so fewer bits are needed to represent the information. So from layer to layer the necessary number of bits decreases.

C. The Effects of the Number of Layers

This is not so explicit. Combining the definitions of ζ and η, the signal to noise ratios becomes

$$R_2^p = \frac{(3E_1 - 2)2^{2N}}{2E_1^2 + 3E_1 - 1}$$

$$R_3^p = \frac{(9E_1E_2 - 6E_1 - 6E_2 + 4)2^{2N}}{2E_1^2E_2^2 + 3E_1E_2^2 - E_2^2 + 9E_1E_2 - 6E_2 - 3E_1 + 2}.$$

Table I shows that as the number of layers and the nonlinearity of the nodes increase, which indicates that the network needs to extract the small changes in the input signal, the signal to noise ratio decreases with a fixed number of bits. In this case more bits are needed to keep the signal to noise ratio constant.

D. The Nonlinearity Coefficient

E_i is the only parameter which makes our neural network different from linear ones. It is a measure of a neural network's nonlinearity with respect to its inputs. According to the above analysis the performance of a network is closely associated with this parameter.

Actually we can interpret E_i as a measure of the permeability of a node. It controls the amount of information that is allowed to "pass through" and the amount that is compressed (abstracted) into one bit by the node after it produces its input weighted sum. Therefore the effective nonlinearity coefficient controls the nonlinear amount of information flowing in the network from its input to its output.

V. SIMULATIONS

We present in this section the results of two simulation experiments that we have conducted to demonstrate the effects of quantization on learning and to verify the conclusions reached by our paper in terms of the relationship between architecture, quantization levels, and training and testing performance.

A study of the performance of several learning algorithms (backpropagation, weight perturbation, modified weight perturbation, combined weight perturbation and random search) and their behavior in response to quantization levels can be found in [4]. Because of space limitations, only an extract of the study is reported.

The test problem is the classification of intracardia electrogram (ICEG) signals. Each pattern contains 21 inputs, the training set consists of 278 patterns, and there are 2463 patterns in the testing set. All patterns need to be placed into one of three classes. The neural network architecture consists of a three-layer network with 21 inputs, 10 hidden units, and 3 outputs. Results shown here are for the combined search algorithm, which incorporates modified weight perturbation [2] and random search. Table II shows the training performance on ten trials for different quantization precision. As analysis has predicted, the performance degradation is roughly an exponential function of the number of quantization bits. The number of bits in the following tables exclude sign bits.

One of the important conclusions reached by our analysis is that the number of quantization bits can be reduced from the lower layers to the higher layers without performance degradation.

First, it is found that the effective nonlinearity coefficients of the nodes in the hidden layer vary from node to node. With 9-bit resolution, the average value is 1.724, and according to (43), the number of bits in the second layer can be 1.29 less than that of the first layer.

Now, we use 9 bits for the quantization of the first layer, the inputs and the weights connecting them to the hidden units. The remaining values in the higher layers are quantized by the same number of bits, which varies from 9 to 6; the results are given in Table III.

Comparing Tables II and III, we can see that the number of bits used for the upper layers can be 2 less than that of in the first layer. This indicates that the information in the output of a hidden unit is compressed in comparison with the information contained in the input of the node.

VI. CONCLUSIONS

In this paper, we have developed statistically based relationships between 1) the quantization levels of weights and neuron input/output states; 2) network architecture (number of hidden units and layers); and 3) the performance of a feedforward multilayer network. To develop these relationships, we have made the assumption that the distribution of the response of hidden units is uniform in order to relate the distributions between successive layers. Following this assumption, we have developed the general formulation of the probability (P^f) that a network output node gives a result different from the output when no quantization is involved. Our formulation predicts (as intuitively one may think) that 1) P^f increases when the number of bits is decreased; 2) a change in the number of hidden units in a layer has no effect on P^f; 3) for a constant "effective nonlinearity coefficient" and number of bits, an increase in the number of layers leads to an increase in P^f; and 4) the number of bits in successive layers can be reduced if the neurons have an effective nonlinearity coefficient that is greater than 1 (nonlinear neurons).

References

[1] A. Krogh, J. Hertz, and R. G. Palmer, *Introduction to the Theory of Neural Computation.* Reading, MA: Addison Wesley, 1990.

[2] M. A. Jabri and B. Flower, "Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," in *Neural Computation,* MIT Press, to be published; also available as SEDAL Tech Report.

[3] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975, ch. 9.

[4] Y. Xie, "Study on the training of multi-layer neural networks with limited precision," SEDAL Tech. Rep. 1991-8-3, Department of Electrical Engineering, University of Sydney, Aug. 1991.

A Neural Detector for Seismic Reflectivity Sequences

Li-Xin Wang

Abstract— **A commonly used routine in seismic signal processing is deconvolution [1]–[3], which comprises two operations: reflectivity detection and magnitude estimation. Existing statistical detectors [2]–[5] are computationally expensive. In this letter, a Hopfield neural network is constructed to perform the reflectivity detection operation. The basic idea is to represent the reflectivity detection problem by an equivalent optimization problem and then construct a Hopfield neural network to solve this optimization problem. The neural detector is applied to a synthetic seismic trace and 30 real seismic traces. The processing results show that the accuracy of the neural detector is about the same as that of the existing detectors [2]–[5], but the speed of the neural detector is much faster.**

I. Introduction

The most commonly used method in oil and natural gas exploration is to ignite an explosive on the surface of the earth and then measure the reflected signals from inside the earth, as shown in Fig. 1. The earth can be approximately modeled by a *reflectivity sequence* $u(k)$. which characterizes the positions and the nature of the reflection layers inside the earth. The information provided by the reflectivity sequence can help geophysicists to determine whether there is oil or natural gas. The explosive is modeled by a *source wavelet* $w(k)$. The measurement $z(k)$ can be represented as a convolution of the reflectivity sequence $u(k)$ and the source wavelet $w(k)$, plus a white noise $n(k)$ [1]–[3]. That is, the geophysical experiment of Fig. 1 can be mathematically modeled as

$$z(k) = \sum_{i=1}^{N} w(k-i)u(i) + n(k), \tag{1}$$

where $k = 1, 2, \cdots, N$, N is the length of the measurements $z(k)$, $n(k)$ is assumed to be zero-mean and white, and $w(j) = 0$ for $j < 0$. The seismic deconvolution problem is, assuming that the source wavelet $w(k)$ is given, estimate the reflectivity sequence $u(k)$ ($k = 1, 2, \cdots, N$) based on the measurements $[z(1), z(2), \cdots, z(N)]$.

The reflectivity sequence $u(k)$ is sparse [2] and is often modeled as a Bernoulli–Gaussian sequence [2], [3]; i.e.,

$$u(k) = q(k)r(k), \tag{2}$$

where $q(k)$ is a Bernoulli sequence with $\Pr[q(k) = 1] = \lambda$ and $\Pr[q(k) = 0] = 1 - \lambda$, and $r(k)$ is a zero-mean white Gaussian sequence. The $q(k)$ indicates whether there is a reflection ($q(k) = 1$) or not ($q(k) = 0$); the $r(k)$ represents the magnitude of the reflection (if there is any). Based on the Bernoulli-Gaussian model of $u(k)$, the seismic deconvolution problem has two parts: one is to estimate $q(k)$; the other is to estimate $r(k)$. The first part is called detetction (because $q(k)$ is a 0–1 sequence); the second part is called magnitude estimation. The magnitude estimation can be performed by using the minimum-variance deconvolution (MVD) algorithm [2]. The detection is much more difficult than the magnitude estimation [2]–[5]. Some statistical detectors were developed in [2]–[5], but they are computationally expensive and have become the bottleneck of the whole seismic deconvolution procedure.

In this letter, we develop a new detector using the Hopfield neural network [6]. Since the Hopfield neural network is suitable for VLSI implementation, our new detector can be realized in hardware and therefore has the potential to greatly speed up the seismic deconvolution procedure. This has practical importance because deconvolution is a routine operation in seismic signal processing, which is one of the most time consuming computational tasks in the world.

II. The Neural Detector

The Hopfield neural network has been used to solve a wide variety of optimization problems (e.g., [7] and [8]). The basic idea is to relate the cost function of an optimization problem to the energy function of the Hopfield neural network, so that when the network reaches its stable state, for which the energy function is locally minimized, the output of the network gives the solution to the optimization problem. We use this basic idea to develop our neural detector.

Suppose that the source wavelet $w(k)$ is given and the magnitude sequence $r(k)$ is estimated using the MVD algorithm [2]; then the detection problem can be reformulated as minimizing the cost function

$$E = \frac{1}{2}\sum_{k=1}^{N}[z(k) - \sum_{i=1}^{N} w(k-i)q(i)r(i)]^2. \tag{3}$$

The energy function of the Hopfield neural network is [6]

$$E_0 = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T(i,j)q(i)q(j) - \sum_{i=1}^{N} I(i)q(i), \tag{4}$$

with $T(i,i) \equiv 0$ for $i = 1, 2, \cdots, N$. Equation (3) can be rewritten as

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}[\sum_{k=1}^{N} w(k-i)w(k-j)r(i)r(j)]q(i)q(j)$$

Manuscript received July 17, 1991.

The author is with the Signal and Image Processing Institute, Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089–0781.

IEEE Log Number 9104051.

1045–9227/92$03.00 © 1992 IEEE