Image Compression and Reconstruction Using Pai-t Sigma Neural Networks with Hybrid Evolutionary-Local Learning

Eduardo Masato Iyoda, Takushi Shibata, Hajime Nobuhara,
Kazuhiko Kawamoto, Yutaka Hatakeyama, and Kaoru Hirota

Department of Computational Intelligence and Systems Science (c/o Hirota Laboratory)
Tokyo Institute of Technology
G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan
Tel.: +81-45-924-5682/5686 Fax: +81-45-924-5676
{iyoda, takushi, nobuhara, kawa, hatake, hirota}@hrt.dis.titech.ac.jp

ABSTRACT

A high-order feedforward neural architecture, called Pi$_t$-Sigma ($\pi_t\sigma$) Neural Network with hybrid evolutionary local learning, is proposed for lossy digital image compression and reconstruction problems. The $\pi_t\sigma$ network architecture is composed of an input layer, a single hidden layer, and an output layer. The hidden layer is composed of classical additive neurons, whereas the output layer is composed of translated multiplicative neurons (π_t-neurons). In order to adjust the parameters of the $\pi_t\sigma$ network, a two-stage learning algorithm is proposed: first, a genetic algorithm (GA) is used to avoid premature convergence to poor local minima; in the second stage, a conjugate gradient method is used to fine tune the solution found by GA. Experiments using the Standard Image Database (SIDBA) and infrared satellite images show that the proposed $\pi_t\sigma$ network performs better than classical multilayer perceptron, improving the reconstruction precision (measured by the mean squared error) in about 56%, on average.

Keywords: Neural networks, image compression, multiplicative neurons, high-order neural networks, genetic algorithm.

1. INTRODUCTION

The expansion of multimedia information processing systems, associated to physical limitations on bandwidth of transmission lines and limited capacity of storage devices, resulted in an increasing interest on data compression technologies, both in industry and in academia. As an efficient image compression and reconstruction scheme, classical Multilayer Perceptron (MLP) neural networks and some of its variations have already been applied in lossy image compression problems (Namphol et al., 1996; Ma and Khorasani, 2002). Because digital images are highly nonlinear mappings, neural networks with expanded nonlinear processing abilities are needed to realize better compression using the smallest amount possible of computational resources. In this paper, therefore, a multiplicative neural network, called pi$_t$-sigma ($\pi_t\sigma$) network, is proposed for image compression and reconstruction problems. The $\pi_t\sigma$ network is a feedforward network composed of an input layer, a hidden layer of additive neurons, and an output layer composed of translated multiplicative neurons (π_t-neurons) (Iyoda et al., 2004). The $\pi_t\sigma$ network is trained using a supervised learning algorithm, composed of 2 stages: (i) a Genetic Algorithm (GA) is used to avoid local minima in the error surface; and (ii) the Scaled Conjugate Gradient (SCG) is applied to fine-tune the solution found by GA. To assess the efficacy of the proposed method, 2 image compression/reconstruction experiments (Standard Image Database (SIDBA) and Geostationary Operational Environmental Satellite (GOES)) are performed.

2. FEEDFORWARD NEURAL NETWORKS FOR IMAGE COMPRESSION

There are many possible approaches for image compression using neural networks proposed in the literature (Jiang, 1999). Among these approaches, lossy image compression using feedforward neural networks trained by supervised learning techniques have produced promising results (Namphol et al., 1996; Ma and Khorasani , 2002). These methods make use of the universal approximation capability of such neural networks to produce high quality reconstructed images, which are approximations of the original image.

The procedure adopted for image compression and reconstruction using feedforward neural networks can be described as follows: initially, a digital image of size $M\times N$ pixels is divided in $m\cdot n$ blocks and these blocks are arranged in vectors $\mathbf{x}_{11},\ldots,\mathbf{x}_{mn}$ of size $l = (M\cdot N)/(m\cdot n)$, as shown in fig. 1. These $m\cdot n$ vectors of dimension l form the training data set for the neural network. Define $g:\Re^l\rightarrow\Re^l$, such that $g(\mathbf{x}_{ij}) = \mathbf{x}_{ij}$, i=1,…,m, j=1,…,n. The objective is to design a neural network able to approximate the mapping $g(\cdot)$, i.e., a neural network whose output $\mathbf{y}_{ij}$ ($\in \Re^l$) is given by

$$\mathbf{y}_{ij} \approx g(\mathbf{x}_{ij}) = \mathbf{x}_{ij}\,. \tag{1}$$

Fig. 2 presents an overview of this learning procedure, where an MLP with a single hidden layer with K hidden neurons is used to approximate the mapping $g(\cdot)$. Note that, to reconstruct the original image, it is necessary to have the outputs of the hidden neurons for each training pattern and the weights between hidden neurons and output layer. In fig. 2, there are $K\cdot l$ weights between hidden and output layers, and $K\cdot m\cdot n$ numerical values representing the hidden neurons outputs for each training pattern. Therefore, the compression rate ρ is defined by

$$\rho = \frac{K(m\cdot n+1)}{M\cdot N}\,. \tag{2}$$

If ρ < 1, then a successful compression has been achieved. In the following, image compression methods employing the procedure presented above are briefly described.

Namphol et al. (1996) propose a hierarchical neural network architecture for image compression, composed of 3 hidden layers, denoted combiner, compressor, and decombine. An image is divided in a number of sub-scenes and each of the sub-scenes is processed by a group of neurons in the combiner layer. The decombine layer is also divided in groups of neurons, each of them responsible for reconstructing a sub-scene from the signals generated by the compressor layer.

Ma and Khorasani (2002) apply single hidden layer neural networks designed by a constructive learning algorithm to image compression. The learning algorithm is based on the Cascade Correlation algorithm (Fahlman and Lebiere, 1991). Several experiments are performed to compare the constructive approach and fixed structure networks. Comparisons with baseline JPEG are also provided.

Although these 2 methods may perform better than existing image compression techniques, they employ neural networks composed of classical additive neurons only. It has been shown that neural architectures employing multiplicative neurons may outperform classical neural network architectures (Ghosh and Shin, 1992; Leerink et al., 1995; Schmitt, 2002), both in terms of approximation accuracy and computational cost. This happens because multiplicative neurons can extract high-order information from learning data more efficiently than additive neurons.

3. TRANSLATED MULTIPLICATIVE NEURON (π_t-NEURON)

Artificial neuron models are usually composed of 2 blocks: an aggregation operator followed by an activation function (Duch and Jankowski, 1999). The aggregation operator combines the neuron's inputs to produce a signal called level of internal activation v ($\in \Re$). The output s ($\in \Re$) of the neuron is then given by $s = f(v)$, where f:$\Re \rightarrow \Re$ is the model's activation function.

Neuron models can be classified according to the type of aggregation operator and activation function employed (Duch and Jankowski, 1999). When v is produced by an additive weighted composition of neuron's inputs, then the model is called additive neuron (or σ-neuron, for short), defined by

$$\begin{aligned} v &= \sum_{i=1}^{m} w_i p_i, \\ s &= f(v), \end{aligned} \tag{3}$$

where w_i ($\in \Re$), i=1,…,m, are the weights (adjustable parameters) of the neuron and p_i ($\in \Re$), i=1,…,m are the neuron's inputs. The model in (3) is the traditional McCulloch-Pitts neuron model.

The multiplicative neuron (π-neuron) (Zhang, 2000) is defined by

$$\begin{aligned} v &= \prod_{i=1}^{m} w_i p_i, \\ y &= f(v), \end{aligned} \tag{4}$$

where w_i ($\in \Re$), i=1,…,m, are the weights of the model.

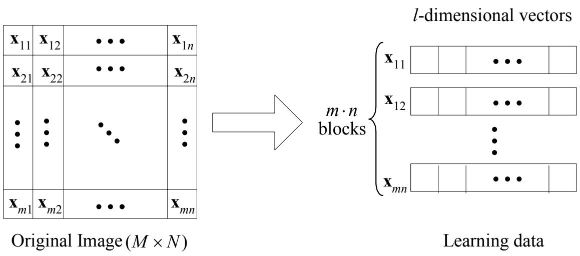

Fig. 1. Construction of neural network training data set. Here, $l = (M{\cdot}N)/(m{\cdot}n)$.

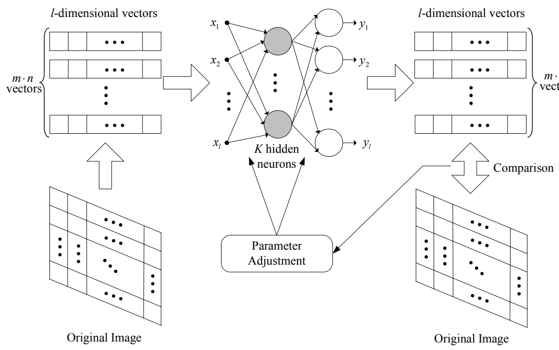

Fig. 2. Construction of neural network training data set.

Note that the model in (4) has 2 properties that may limit its applicability in complex problems (Iyoda et al., 2004): (i) it has an excessive number of parameters and (ii) decision surfaces generated by this model are always centered in the origin of the neuron's input space $\Re^m$.

To expand the capabilities of the π-neuron model, the translated multiplicative neuron (π_t-neuron) has been proposed (Iyoda et al., 2004). The π_t-neuron model is defined by

$$\begin{aligned} v &= b\prod_{i=1}^{m} (p_i - t_i), \\ s &= f(v), \end{aligned} \tag{5}$$

where the parameters t_i ($\in\Re$), i=1,…,m, represent the coordinates of the center of the decision surface generated by the model, and b ($\in\Re$) is a scaling factor. The π_t-neuron model has 2 advantages, when compared to the traditional multiplicative neuron: (i) it has a meaningful set of adjustable parameters and (ii) the decision surfaces generated by this model can be placed in any point of its input space.

The π_t-neuron model has been tested in some supervised learning problems, including nonlinear regression (Iyoda et al., 2004) and pattern classification (Iyoda et al., 2003a). In most of these problems, neural networks employing π_t-neurons could achieve better performance than classical neural network models. It has also been shown that a single π_t-neuron can solve the N-bit parity problem (Iyoda et al., 2003b). These results confirm that the π_t-neuron has expanded information processing capabilities when compared to the classical additive neuron model. These capabilities seem to be more evident when the data to be approximated has a high degree of nonlinearity.

Because typical digital images are mappings containing high degree of nonlinearity, a neural network should have improved nonlinear processing capabilities to realize the mapping in (1) using the smallest amount possible of computational resources (i.e., number of hidden neurons and connection weights). Since networks using π_t-neurons have expanded nonlinear processing abilities, a neural network architecture containing π_t-neurons as processing elements is proposed for digital image compression and reconstruction problems. The proposed network is described in detail in section 4.

4. THE PI$_t$-SIGMA ($\pi_t\sigma$) NEURAL NETWORK

The pi$_t$-sigma ($\pi_t\sigma$) neural network used for image compression and reconstruction is depicted in fig. 3. The

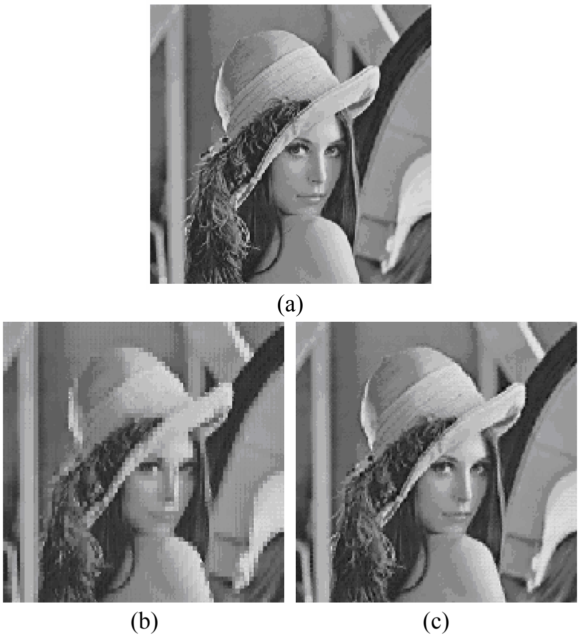

Fig. 4. Comparison of proposed learning algorithm. (a) Original image. (b) Compressed and reconstructed with SCG only. (c) Compressed and reconstructed with GA+SCG.

$\pi_t\sigma$ is a feedforward neural network, composed of an input layer with l nodes, a hidden layer of K σ-neurons, and an output layer of l π_t-neurons. The network's output y_p ($\in\Re$), p=1,…,l, is given by

$$y_p = b_p \prod_{i=1}^{K} \left(f\left(\sum_{j=1}^{l} (w_{ij}x_j - w_{0j}) \right) - t_{pi} \right), \tag{6}$$

where w_{ij} ($\in\Re$), i=1,…,K, j=1,…,l, is the weight connecting the input x_j to the hidden neuron i and w_{0j} ($\in\Re$) is the weight connecting the bias term x_0=+1 to the i-th hidden neuron. The parameters b_p ($\in\Re$) and t_{pi} ($\in\Re$), p=1,…,l, i=1,…,K, are the parameters of the π_t-neurons in the output layer of the network.

The σ-neurons in the hidden layer work as a feature extractor, thus reducing the dimensionality of the input space. Because of the multiplicative characteristic of the π_t-neuron model, the output layer of the p*t*s network is able to (indirectly) detect higher-order correlations in the learning data. In other words, it has expanded information processing capabilities in comparison with the classical MLP model. Therefore, the $\pi_t\sigma$ network has the potential to better describe the true input-output mapping using a smaller number of hidden neurons.

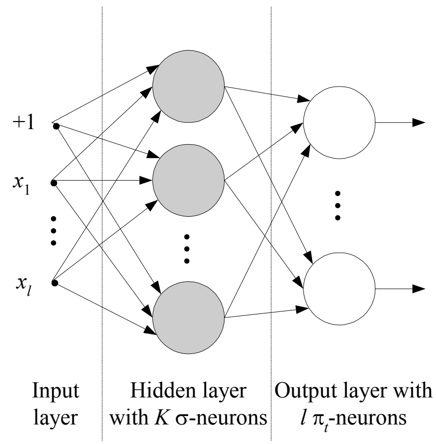

Fig. 3. The Pi$_t$-Sigma ($\pi_t\sigma$) neural network architecture.

The $\pi_t\sigma$ network can be considered as an extension of the pi-sigma neural architecture (Ghosh and Shin, 1992), which employs the π-neuron model defined in (4) in its output layer. Since the original pi-sigma neural architecture has the universal approximation property, it is clear that the $\pi_t\sigma$ network is also a universal approximator.

4.1. LEARNING ALGORITHM OF $\pi_t\sigma$ NEURAL NETWORK

Although there are numerous heuristics and strategies to initialize classical MLP networks (Haykin, 1999, ch. 4), strategies to initialize the weights of networks containing of π_t-neurons have not been established yet. During the initial experiments stage, the Scaled Conjugate Gradient (SCG) algorithm (Møller, 1993) was used to train $\pi_t\sigma$ networks, but convergence to poor local optima was observed very often.

To avoid the premature convergence problem, a two-stage learning procedure is adopted. First, a Genetic Algorithm (GA) is used to find a suboptimal solution. After the termination of the genetic search, the SCG is used to fine-tune the solution found by GA.

The standard GA (Michalewicz, 1996) is employed and its main characteristics are as follows.

- Codification: a chromosome is a floating-point vector **p** of dimension (l+1)K+(K+1)l, where each element of the vector represents an adjustable parameter (w_{ij}, b_p, and t_{pi}) of the network described in (6) and depicted in fig. 3.
- Fitness function: each chromosome is evaluated by

  $$fit(\mathbf{p}) = \frac{1}{MSE(\mathbf{p})},$$

  where $MSE(\mathbf{p})$ is the mean squared error of the network codified by **p**, defined as

  $$MSE(\mathbf{p}) = \frac{1}{2m\cdot n} \sum_{i=1}^{m\cdot n} \sum_{j=1}^{l} \left(y_j^{(i)} - x_j^{(i)} \right)^2, \tag{7}$$

  where $y_j^{(i)}$ is the j-th network output for i-th training pattern, and $x_j^{(i)}$ is its corresponding target value.
- Selection operator: chromosomes are selected for the next generation using the roulette wheel operator, which assigns selection probabilities proportional to the fitness of the individual.
- Crossover operator: uniform crossover, where the elements of 2 parent chromosomes are exchanged with a certain probability.
- Mutation operator: gaussian mutation, where an element p_i of a chromosome selected for mutation is modified according to

  $$p_i = p_i + N(0,1),$$

  where $N(0,1)$ is a gaussian random variable with 0 mean and standard deviation 1.

The SCG algorithm is chosen because it does not require any line search procedure and does not have any critical user defined parameter. Details about SCG can be found in Møller (1993).

To confirm the efficacy of the proposed learning scheme, a simple compression/reconstruction experiment is

Table 1. Mean squared error for SIDBA reconstructed images.

Image	*Number of hidden neuron* 4		5		6		7	
	MLP	$\pi_t\sigma$	MLP	$\pi_t\sigma$	MLP	$\pi_t\sigma$	MLP	$\pi_t\sigma$
Airplane	0.0478	0.0185	0.0285	0.0141	0.0461	0.0111	0.0353	0.0095
Barbara	0.0741	0.0222	0.0337	0.0156	0.0308	0.0122	0.0335	0.0082
Boat	0.0189	0.0089	0.0175	0.0054	0.0170	0.0045	0.0220	0.0043
Bridge	0.0830	0.0412	0.0606	0.0322	0.0494	0.0271	0.0370	0.0243
Building	0.0335	0.0135	0.0427	0.0103	0.0255	0.0066	0.0245	0.0065
Cameraman	0.0397	0.0232	0.0336	0.0188	0.0401	0.0147	0.0418	0.0145
Girl	0.0214	0.0072	0.0164	0.0064	0.0164	0.0049	0.0107	0.0043
Lax	0.0732	0.0434	0.0494	0.0333	0.0543	0.0285	0.0641	0.0243
Lenna	0.0250	0.0111	0.0201	0.0091	0.0285	0.0081	0.0170	0.0063
Lighthouse	0.0688	0.0322	0.0457	0.0294	0.0326	0.0204	0.0474	0.0165
Text	0.0302	0.0294	0.0305	0.0204	0.0274	0.0178	0.0299	0.0151
Woman	0.0268	0.0116	0.0390	0.0085	0.0208	0.0068	0.0237	0.0061

performed using the benchmark image lenna. In this experiment, the performance of a $\pi_t\sigma$ neural network with 5 hidden neurons trained using SCG only is compared to a $\pi_t\sigma$ network trained by the proposed GA+SCG learning algorithm. The parameters of the GA are as follows:

- Population size: 150;
- Maximum number of generations: 10000;
- Crossover probability: 0.8;
- Mutation probability: 0.0001.

For the SCG, the maximum number of epochs is set to 10000, whereas the stop criterion is defined as norm of error gradient smaller than 0.001.

The images compressed and reconstructed by the compared learning algorithms are shown in figs. 4(b) and 4(c). The MSE of the image obtained using SCG only is 0.027, whereas the MSE of the image obtained by GA+SCG is 0.007. Therefore, the efficacy of the proposed GA+SCG learning algorithm is confirmed.

(a) (b)

Fig. 5. Woman image compressed and reconstructed by: (a) MLP; (b) $\pi_t\sigma$ network. Both networks have 5 hidden neurons.

5. COMPRESSION AND RECONSTRUCTION OF THE STANDARD IMAGE DATABASE AND INFRARED SATELLITE IMAGES

To confirm the validity of the proposed approach, 2 experiments are performed. In the first experiment, images from the Standard Image Database (SIDBA) are employed to evaluate the approximation capability of the proposed $\pi_t\sigma$ neural network. In the second experiment, the generalization capability of $\pi_t\sigma$ neural network is investigated using infrared images taken by a Geostationary Synchronous Satellite. In both experiments, the performance of $\pi_t\sigma$ network is compared to that of classical multilayer perceptrons.

5.1. COMPRESSION AND RECONSTRUCTION OF THE STANDARD IMAGE DATABASE

The Standard Image Database (SIDBA) is composed of 12 grayscale images, some of them widely used to test the performance of image processing algorithms. All the images are of size 256×256, i.e., N=M=256. To construct the training data set for the neural networks, all the images are divided in blocks of size 4×4, i.e., n=m=4. Therefore, the dimension of the input space is l=16 and there are 16556 training patterns. The images of the SIDBA can be downloaded from http://www.sp.ee.musashi-tech.ac.jp/app.html.

Two neural architectures are compared:

(i) Multilayer Perceptron (MLP) with sigmoidal activation function in the hidden neurons and linear activation function in the output neuron. The MLP is trained using the SCG algorithm. The maximum number of epochs is set to 10000 and the stop criterion is norm of error gradient smaller than 0.001.

(ii) Pi$_t$-sigma ($\pi_t\sigma$) neural network, with logistic activation function in the hidden neurons and linear activation function in the output neurons. The parameters for the GA are as follows:

- Population size: 150;
- Maximum number of generations: 10000;
- Crossover probability: 0.8;
- Mutation probability: 0.0001.

  For the SCG, the maximum number of epochs is 10000, whereas the stop criterion is norm of error gradient smaller than 0.001.

For each neural architecture, the number of hidden neurons is varied between 4 and 7.

Table 1 shows the mean squared error of the images compressed and reconstructed by MLP and $\pi_t\sigma$ networks. For all the cases the proposed $\pi_t\sigma$ network obtained better performance than traditional MLP. Fig. 5 shows an example of images compressed and reconstructed by the two neural networks considered. From these figures, it is possible to notice the higher quality of the images reconstructed using the proposed method.

From the results it is confirmed that, using the same number of hidden neurons, the proposed $\pi_t\sigma$ network can achieve better performance than the classical MLP.

5.2. COMPRESSION AND RECONSTRUCTION OF INFRARED SATELLITE IMAGES

To evaluate the generalization capability of the proposed $\pi_t\sigma$ neural network, a set of infrared images taken by a Geostationary Operational Environmental Satellite (GOES) orbiting Japan is used. The images generated by this satellite are grayscale, of size 800×800, i.e., M=N=800. Six images taken in January, 2000 are chosen for this experiment; 1 image is used for to train the neural networks considered and the other 5 images are used to test their generalization ability.

To construct the training data set, the training image is divided in blocks of size 8×8, i.e., m=n=8. Thus, the training data has dimension l=64 and there are 10000 training instances.

Two neural architectures are compared:

(i) Multilayer Perceptron (MLP) with 8 hidden neurons using logistic activation function and output neurons using linear activation function. The MLP is trained using the SCG algorithm. The maximum number of epochs is set to 10000 and the stop criterion is norm of error gradient smaller than 0.001.

(ii) Pi$_t$-sigma ($\pi_t\sigma$) neural network, with logistic activation function in the hidden neurons and linear activation function in the output neurons. The parameters for the GA are as follows:

- Population size: 150;
- Maximum number of generations: 10000;
- Crossover probability: 0.4;
- Mutation probability: 0.0001.

For the SCG step, the maximum number of epochs is 10000, whereas the stop criterion is norm of error gradient smaller than 0.001.

Table 2 compares the results obtained by MLP and $\pi_t\sigma$ neural networks. The performance measured used is the mean squared error, defined in (7). In this experiment, $\pi_t\sigma$ neural network can again achieve better performance than classical MLP, in all the images considered. The results of this experiments show that, besides having better approximation properties, $\pi_t\sigma$ network has also better generalization capability than the traditional MLP.

Table 2. Mean squared error of GOES reconstructed images. Both networks have 8 hidden neurons. The image 00012709.h is the training image.

Image	MLP	$\pi_t\sigma$ network
00012709.h	0.16976	0.07611
00010309.h	0.18273	0.08387
00010609.h	0.18585	0.08502
00010709.h	0.18570	0.08415
00010809.h	0.18656	0.08261
00011009.h	0.18705	0.08399

6. CONCLUSIONS

A neural architecture called Pi$_t$-Sigma ($\pi_t\sigma$) neural network is proposed for digital image compression and reconstruction. The $\pi_t\sigma$ neural network is composed of an input layer, a hidden layer of additive neurons, and an output layer of translated multiplicative neurons (π_t-neurons) (Iyoda et al., 2004). The multiplicative characteristic of π_t-neuron model enables the proposed $\pi_t\sigma$ network to extract (indirectly) high-order information from the training image data.

The learning algorithm of $\pi_t\sigma$ network is composed of 2 stages. First, a floating-point Genetic Algorithm (GA) is used to avoid local minima in the network's error surface. After the evolutionary process, the Scaled Conjugate Gradient (SCG) (Møller, 1993) is used to fine-tune the solution found by GA. Experiment results show that the combined GA+SCG learning algorithm produces reconstructed images with Mean Squared Error (MSE) about 20% lower than that produced using SCG only.

To evaluate the performance of $\pi_t\sigma$ network in image compression and reconstruction problems, 2 experiments are conducted. In the first experiment, images from the Standard Image DataBase (SIDBA) are used to evaluate the approximation capability of $\pi_t\sigma$ network. The images compressed and reconstructed using the $\pi_t\sigma$ network have MSE about 57% lower than those obtained using classical Multilayer Perceptrons (MLP).

The generalization capability of $\pi_t\sigma$ network is evaluated using a set of infrared images obtained by a Geostationary Operational Environmental Satellite (GOES). The set is composed of 6 images, where 1 is used for training and the remaining 5 are used for testing. The test images compressed and reconstructed by $\pi_t\sigma$ network have an MSE 55% smaller than those obtained by MLP.

The results confirm that the proposed $\pi_t\sigma$ network has better nonlinear approximation and generalization capabilities than the classical MLP architecture. They also confirm the suitability of the proposed network to digital image compression and reconstruction problems.

The proposed method requires long training times, due to the burden imposed by the GA and the big size of the training data. Since the proposed approach operates offline, this does not limit the applicability of the method. Furthermore, with the rapid advance of hardware computational power, it is certain that the proposed method will be running much faster in the near future. Another way to shorten the training time is to develop smart heuristics for initializing a $\pi_t\sigma$ network, thus eliminating the need for GA. This is certain a topic for future research.

Another future research direction is to investigate the performance of $\pi_t\sigma$ network will in compression and reconstruction of color images. Furthermore, the applicability of $\pi_t\sigma$ network in video compression problems will also be considered.

REFERENCES

Duch, W. and Jankowski, N. (1999). Survey of Neural Transfer Functions. Neural Computing Surveys, 2, 163-212.

Fahlman, S. E. and Lebiere, C. (1991). The Cascade-Correlation Learning Architecture. Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Ghosh, J. and Shin, Y. (1992). Efficient Higher-Order Neural Networks for Classification and Function Approximation. International Journal of Neural Systems, 3, 323-350.

Haykin, S. (1999). Neural Networks - A Comprehensive Foundation. 2nd ed. Prentice Hall.

Iyoda, E. M., Nobuhara, H., Kawamoto, K., Yoshida, S., and Hirota, K. (2003a). Sigma-pi$_t$ Cascaded Hybrid Neural Network and its Application to the Spirals and Sonar Pattern Classification Problems. In Proceedings of the 4th International Symposium on Advanced Intelligent Systems, Jeju Island, Korea, pp. 158-161.

Iyoda, E. M., Nobuhara, H., and Hirota, K. (2003b). Solving the *N*-bit Parity Problem Using a Single Translated Multiplicative Neuron. Neural Processing Letters, 18, 213-218.

Iyoda, E. M., Nobuhara, H., and Hirota, K. (2004). Translated Multiplicative Neuron: An Extended Multiplicative Neuron that can Translate Decision Surfaces. Journal of Advanced Computational Intelligence and Intelligent Informatics (to appear).

Jiang, J. (1999). Image Compression with Neural Networks - a Survey. Signal Processing: Image Communication, 14, 737-760.

Leerink, L. R., Lee Giles, C., Horne. B. G., and Jabri, M. A. (1995). Product unit learning. Technical Report CS-TR-3503, University of Maryland, College Park, MD.

Ma, L. and Khorasani, K. (2002). Application of Adaptive Constructive Neural Network to Image Compression. IEEE Transactions on Neural Networks, 13, 1112-1126.

Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs, 3rd ed. Springer.

Møller, M. F. (1993). Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks, 6, 525-533.

Namphol, A., Chin, S. H., and Arozullah, M. (1996). Image Compression With a Hierarchical Neural Network. IEEE Transactions on Aerospace Electronic Systems, 32, 326-337.

Schmitt, M. (2002). On the Complexity of Computing and Learning with Multiplicative Neurons. Neural Computation, 14, 241-301.

Zhang, B. -T. (2002). A Bayesian Evolutionary Approach to the Design and Learning of Heterogeneous Neural Trees. Integrated Computer-Aided Engineering, 9, 73-86.