Enhanced ART1-based Self-Organizing Supervised Learning Algorithm for Channel Optimization in Mobile Cellular Networks

Kwang-baek Kim†, Cheul-woo Ro†, Kwang-eui Lee†† and Kyung-min Kim†

† Division of Computer and Information Engineering, Silla University, Busan, Korea
†† Department of Multimedia Engineering, Dongeui University, Busan, Korea

Summary
In this paper, we propose a novel approach for evolving the architecture of a multi-layer neural network. Our method uses combined ART1 algorithm and Max-Min neural network to self-generate nodes in the hidden layer. We have applied the proposed method to the optimal channel allocation problem in mobile cellular networks. Experimental results show that the proposed method has better performance than conventional neural networks and the resulting neural network computes the optimal guard channel number g within ignorable error bound for GoS.

Key words:
ART1, neural network, channel optimization, guard channel, cellular networks.

1. Introduction

The backpropagation network (BPN), which is also sometimes referred to as a multilayer perceptron (MLP), is currently the most general-purpose and commonly used neural-network paradigm [1]. The BPN achieves its generality because of the gradient-descent technique used to train the network. Gradient descent is analogous to an error-minimization process. But, BPN has a drawback. I.e., despite its popularity as an optimization tool and for neural network training, the gradient descent has several drawbacks such as local minima usually not meeting the desired convergence criterion [2,3,4]. Grossberg and Carpenter developed the adaptive resonance theory (ART). ART was developed to solve the learning instability problem suffered by standard feed-forward network [5,6]. The weights, which have captured some knowledge in the past, continue to change as new knowledge comes in. There is therefore a danger of losing the old knowledge with time. The weights have to be flexible enough to accommodate the new knowledge but not so much so as to lose the old. This is called the stability plasticity dilemma and it has been one of the main concerns in the development of artificial neural network paradigm [7]. Max-Min neural network uses fuzzy logic to update the weights in a multi-layer perceptron rather than the delta value which uses multiplication and addition operator [8].

Optimal channel allocation [9] is one of the important problems in mobile cellular networks. In this paper, we solve the problem using the combined ART1 algorithm and Max-Min neural network.

There are two types of call in cellular networks: new call and handoff call. Handoff call occurs when a mobile station moves across a cell boundary. Optimality is achieved by minimizing the GoS (Grade of Service) in the sense of call blocking rate. GoS is defined by the following equation [10],

$$GoS = Pb + \omega Pd \tag{1}$$

where *Pb* is the probability of blocking a new call, *Pd* is the probability of blocking a handoff call (dropping) and ω is a weighting factor that decides how much emphasis is placed on handoff calls.

To reduce the dropping probability of handoff calls, a fixed number of guard channels is reserved exclusively for the handoff calls [11]. By using the guard channel policy, dropping probability can significantly be reduced. However, reserving guard channels exclusively for handoff call could result in blocking probability increase. To find a balance between these two measures, we use equation (1) to consider the composite effect of dropping and blocking probabilities. To minimize the GoS, an optimal number of guard channels should be computed.

This paper presents an enhanced self organizing supervised learning algorithm using combined self-generating model of ART1 and Max-Min neural network for enhancing recognition ratio and solving a problem of hidden layer's node numbers. Finally, we construct a neural network deciding the optimal g using the proposed architecture.

The remaining part of the paper is organized as follows. Section 2 describes the proposed architecture. Section 3 covers the definition of the optimal channel allocation problem in cellular networks. We show experimental results and Performance Analysis in Section 4, and the paper concludes in Section 5.

Manuscript received April 5, 2006.
Manuscript revised April 30 , 2006.

2. Enhanced neural networks

2.1 Enhanced ART1-based Self-Organizing Supervised Learning Architecture

BP learning method used widely in multi-layer neural networks has a possibility of local minima due to the inadequate weights and the insufficient number of hidden nodes. So, we propose an enhanced neural networks by using self-organization that self-generates hidden nodes by the compound ART1 algorithm and Max-Min neural network [8]. The proposed network is presented with a large number of patterns and each hidden layer neuron represents the cluster center. The prototype pattern for each cluster is represented by the weights from the hidden neuron to the input neuron. Vigilance criterion is used to achieve unsupervised learning which determines the actual number of clusters.

In the proposed architecture, the connection structure between input layer and hidden layer is similar to structure of the modified ART1. The output layer of the modified ART1 is used as hidden layer in proposal structure. A node of hidden layer represents each class. The nodes in hidden layer are fully connected to nodes in input and output layers. In the case of backward propagation comparing target value with actual output value, we adopt a winner-take-all method to modify weighting factor of only the synapse that is connected to the winner class. The adaptation of weight of synapses between output layer and hidden layer is accomplished by Max-Min neural network. Fig.1 shows the proposed learning architecture.

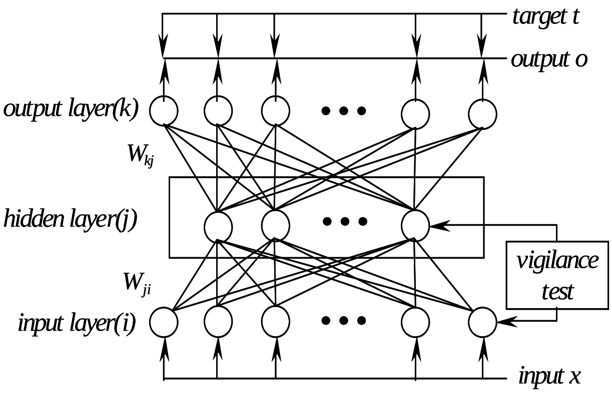

Fig. 1 Enhanced ART1-based Self-Organizing Supervised Learning Architecture

2.2 Enhanced ART1-based self-organizing supervised learning algorithm

BP learning method used widely in multi-layer neural networks has a possibility of local minima due to the inadequate weights and the insufficient number of hidden nodes. So, we propose an enhanced neural network by using self-organization that self-generates hidden nodes by the compound ART1 algorithm and Max-Min neural network. The proposed network is presented with a large number of patterns and each hidden layer neuron

The creation of the clustering layer nodes is based on the number of determining classes by input patterns. Based on ART1, we assume the number of maximum initial nodes of the hidden layer as the number of classes.

Starting with one node, we allocate related classes to the initially suggested pattern from the input layer of the node. Next input patterns choose a winner for the nodes in the present states. If all the existing nodes fail to choose a winner, they add one node and allocate it to the class for the related pattern. In this way, patterns are sequentially suggested and the nodes for the class are created dynamically. But the winner node chosen by the input pattern is not the representative class for the pattern. If the stored pattern of the winner node is similar to the input pattern, it becomes the winner. Otherwise, classification is repeated until we get a winner.

The proposed algorithm uses a winner-take-all method on conventional backpropagation learning to change weights. When we classify the connection between the input layer and the clustering layer, and the connection between the clustering layer and the target layer, the winner node chosen from the clustering layer becomes the representative class of input pattern. Therefore, we should adjust the weights connected to the winner node from the clustering layer to the input layer. To reflect target value for the input pattern to the actual output value by the representative class, we change only the connection weights related to the target layer node and its representative class.

The proposed learning algorithm as follows:

Step 1. Initialize weights, bias term and vigilance threshold.

$$b_{ji} = 1,\ w_{ji} = \frac{1}{m+1},\ w_{kj} = \textit{small} \text{ random value},\ \theta_k = \textit{small} \text{ random value} \quad (2)$$
$$0 \le i \le m-1,\ 0 \le j \le n-1,\ 0 \le k \le p-1 \qquad \textit{Set } \rho, \text{ where } 0 < \rho \le 1$$

Where b_{ji} is the value of the top-down weight from neuron i in the input layer to neuron j in the hidden layer and w_{ji} is the value of the bottom-up weight from neuron

i in the input layer to neuron j in the hidden layer. w_{kj} is the value of a weight from neuron j in the hidden layer to neuron k in the output layer. θ_k is a bias term in the output layer. ρ is the vigilance threshold, which determines how close an input has to be to correctly match a stored pattern.

Step 2. Set target value t_k and train each input data x_i .

Step 3. Calculate output o_j in hidden layer.

$$o_j = \sum_{j=0}^{n-1} w_{ji} \times x_j \tag{3}$$

Step 4. Select a winner node o_j^* .

$$o_j^* = Max[o_j] \tag{4}$$

The method that selects winner node for input data is that the winner node maximize output value o_j in hidden layer.

Step 5. Compare vigilance threshold ρ between the value of input data and stored pattern in the winner node.

If $\frac{\| T \bullet X \|}{\| X \|} \geq \rho$, go to step 7. Else, go to step 6.

Step 6. Reassign zero to o_j^* in the winner node and go to step 4.

Step 7. Update the connection weights of the winner node between hidden layer and input layer.

$$\begin{aligned} t_{j^*i}(n+1) &= t_{j^*i}(n) \times x_i \\ w_{j^*i}(n+1) &= \frac{t_{j^*i}(n+1) \times x_i}{0.5 + \sum_{i=1}^{m} w_{j^*i} \times x_i} \end{aligned} \tag{5}$$

Step 8. Calculate node's NET for output layer using the winner node's output o_j^* in hidden layer and the connection weight w_{kj^*} between hidden layer and output layer. And then calculate the output o_k of output layer using max($\vee$) operator.

$$\begin{aligned} NET &= \{o_j^* \circ w_{kj^*}\} \\ o_k &= NET \vee \theta_k \end{aligned} \tag{6}$$

Where " $\circ$ " denotes max-min composition.

Step 9. Update the connection weights between output layer and hidden layer and bias term.

$$\begin{aligned} w_{kj^*}(n+1) &= w_{kj^*}(n) + \alpha \Delta w_{kj^*}(n+1) + \beta \Delta w_{kj^*}(n) \\ \theta_k(n+1) &= \theta_k(n) + \alpha \Delta \theta_k(n+1) + \beta \Delta \theta_k(n) \end{aligned} \tag{7}$$

Where α is the learning rate and β is the momentum.

$$\begin{aligned} &\Delta w_{kj^*} = \sum_{k=1}^{p} (t_k - o_k) \frac{\partial o_k}{\partial w_{kj^*}}, \Delta \theta_k = \sum_{k=1}^{p} (t_k - o_k) \frac{\partial o_k}{\partial \theta_k} \\ &\frac{\partial o_k}{\partial w_{kj^*}} = 1, \text{ where } \mathrm{o_k} = w_{kj} \ \frac{\partial o_k}{\partial w_{kj^*}} = 0, \text{ otherwise.} \\ &\frac{\partial o_k}{\partial \theta_k} = 1, \text{ where } \mathrm{o_k} = \theta_k \ \frac{\partial o_k}{\partial \theta_k} = 0, \text{ otherwise.} \end{aligned} \tag{8}$$

Step 10. For all training pattern pair, if (***TSS<Error Criteria***) then stop learning.

3. Optimal channel allocation problem

3.1 Cellular system description

We consider the performance model of a single cell in mobile cellular networks. Let λ_{vn} be the rate of the Poisson arrival stream of new calls and λ_{vh} be the rate of Poisson stream of handoff arrivals. An ongoing call (new or handoff) completes service at the rate μ_{vt} and the mobile engaged in the call departs the cell at the rate μ_{vout} . There are a limited number of channels *S*, in the channel pool. When a handoff call arrives and an idle channel is available in the channel pool, the call is accepted and a channel is assigned to it. Otherwise, the handoff call is dropped. When a new call arrives, it is accepted provided that *g+1* or more idle channels are available in the channel pool; otherwise, the new call is blocked. Here, *g* is the number of guard channels. We assume that *g* < *S* in order not to exclude new calls altogether.
Let *C(t)* denote the number of busy channels at time *t*, then $\{C(t), t \geq 0\}$ is a birth–death process as shown in Fig. 2.

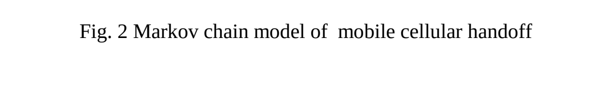

Fig. 2 Markov chain model of mobile cellular handoff

Let $\lambda = \lambda_{vn} + \lambda_{vh}$, $\mu = \mu_{vt} + \mu_{vout}$. The state-dependent arrival and departure rates in the birth–death process are given by

$$\Lambda(n) = \begin{cases} \lambda, & n = 0,1,\ldots,S-g-1 \\ \lambda_{vh}, & n = S-g,\ldots,S-1;\ g > 0 \end{cases} \quad (9)$$
$$M(n) = n\mu,\ n = 1,\ldots,S$$

Because of the structure of the Markov chain we can easily
write down the solution to the steady–state balance equations as follows. Define the steady–state probability

$$p_n = \lim_{t \to \infty} \text{Prob}\ (C(t) = n),\ n = 0,1,2\ldots,\ S \quad (10)$$

Let $\rho = \lambda / \mu$, $\rho_1 = \lambda_{vh} / (\mu_{vt} + \mu_{vout})$. Then we have an expression for p_n

$$p_n = p_0 \begin{cases} \dfrac{\rho^n}{n!}, & n \le S-g \\ \dfrac{\rho^{S-g}}{n!} \rho_1^{n-(S-g)}, & n \ge S-g \end{cases} \quad (11)$$

where

$$p_0 = \frac{1}{\sum_{n=0}^{S-g-1} \dfrac{\rho^n}{n!} + \sum_{n=S-g}^{S} \dfrac{\rho^{S-g}}{n!} \rho_1^{n-(S-g)}} \quad (12)$$

Now we can write expressions for the dropping probability for handoff calls

$$P_d(S,g) = p_S = p_0 \frac{\rho^{S-g}}{S!} \rho_1^g \quad (13)$$

Similarly, the expression for the blocking probability of new calls is

$$P_b(S,g) = \sum_{n=S-g}^{S} p_n = p_0 \sum_{n=S-g}^{S} \frac{\rho^{S-g}}{n!} \rho_1^{n-(S-g)} \quad (14)$$
$$= p_0 \rho^{S-g} \sum_{k=0}^{g} \frac{\rho_1^k}{(k+S-g)!}$$

Note that if we set $g=0$ then expression (14) reduces to the classical Erlang-B loss formula. In fact, setting $g=0$ in expression (13) also provides the Erlang-B loss formula [9].

3.2 Optimization for guard channel

We consider the problem of finding the optimal number g for guard channels such that GoS is minimized. In order to solve the optimization problem, the proposed learning algorithm is used.
With different new call arrival rates, the corresponding handoff call arrival rates vary accordingly. To capture this dynamic behavior, a fixed point iteration scheme is applied to determine the handoff arrival rates [12]. To specify and solve the Markov chain models, the tool SPNP [13] is used.

4. Experimental results

The experimental procedure includes four stages: (1) As training data, the optimal guard channel numbers under various S, ω , λ_{vn} are obtained using SPNP; (2) The proposed learning algorithm is applied to the optimal guard channel numbers; (3) To show the optimality, compare the result of the neural network to the solution of Markov chain models for unlearned combinations of S, ω, λ_{vn}; (4) And finally, the performance results are compared to the performance of conventional backpropagation algorithm.

As a first step of our procedure, we find a value of g that is minimizing the GoS according to the result shown in Fig.3

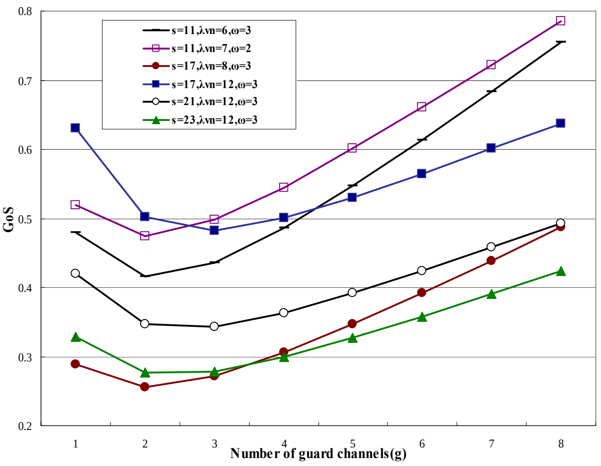

Fig. 3 GoS versus guard channel numbers

In the training experiment, 420's the optimal guard channel numbers were used. In step 3, to achieve the accuracy of our proposed algorithm, 100 random data is tested by comparing it with the Markov chain model

(SPNP) and only 4 cases have different results as shown in Table 1. The presented results from the two methods in Table 1 show negligible difference.

Table 1: Comparison of GoS

S	λ_{vn}	ω	*SPNP g (GoS)*	*Proposed Algorithm g (GoS)*	*difference*
11	**7**	**3**	2 (0.4996)	3 (0.5046)	0.0050
13	**8**	**3**	2 (0.4560)	3 (0.4578)	0.0018
9	**8**	**2**	1 (0.1640)	2 (0.1668)	0.0028
23	**12**	**3**	2 (0.2769)	3 (0.2784)	0.0015

As shown in Table 1, all 420's the optimal guard channel numbers were successfully trained from the proposed algorithm and 96's the optimal guard channel numbers were recognized successfully out of a total of 100's test data. Therefore, the accuracy of recognition rate is 96% in this study.
Table 2 represents comparison of epoch's number and TSS through proposed method and conventional backpropagation algorithm.

Table 2: Learning results of each learning method

	Epoch number	*TSS*
Conventional Backpropagation	1305	0.089961
The Proposed Algorithm	701	0.077842

The initial connection weights used for training in each algorithm were set to values between –1 to 1. And the learning rate and the momentum were set to 0.5 and 0.5 for the two recognition algorithms, respectively. For the proposed algoritm, the vigilance parameter used for the creation and update of clusters was empirically set via the priority test. Based on simulation results the optimum value for the vigilance parameters for traning data was set as 0.8. Table 1's results represent result of training with error limit 0.09. In backpropagation algorithm, when we set 10~30 nodes of hidden layer, we obtained the fact that the case of 18 nodes has good performance (fast training time, high convergence). Therefore, Table 1 represents a result of training (case of 18 hidden nodes). In the proposed method, because we applied ART1 as structure of between input and hidden layer, it produced 20 hidden nodes after training.

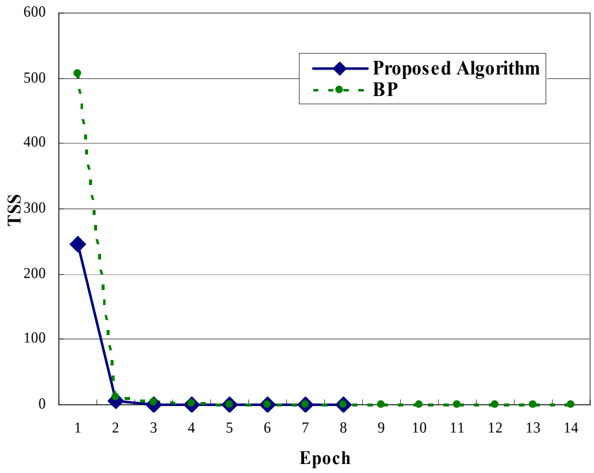

Fig. 4 Variance of TSS according to learning methods

Fig. 4 shows the curve that is the sum of the square of errors in the backpropagation and the proposed method. As shown in Fig. 4, the proposed method wins the conventional methods in terms of the speed of the initial convergence and training time. Through experimental results, we know that the proposed method spend less training time than conventional training method, and have a good convergence ability. This is based on the fact that winner-take-all method is adopted to the connection weight adaptation, so that a stored pattern for some pattern gets updated. Moreover, the proposed method reduced the possibility of local minima due to the inadequate weights and the insufficient number of hidden nodes.

5. Conclusions

This paper proposes an enhanced supervised learning algorithm by using self-organization that self-generates hidden nodes by the compound Max-Min neural network and modified ART1. From the input layer to hidden layer, a modified ART1 was used to produce nodes. And winner-take-all method was adopted to the connection weight adaptation, so that a stored pattern for some pattern gets updated.
Using the proposed architecture, we construct the neural network algorithm for optimal channel allocation problem in mobile cellular networks.
The Experimental result shows that the proposed method did not sensitively responded about moment, had good convergence ability, and had less training time than conventional backpropagation algorithm. We must enhance hidden node number's increase according to vigilance variable's changes.

References

[1] James and Freeman, Neural Networks: Algorithm, Application and Programming Techniques, Addison-Wesley, 1991.
[2] R. Hecht-Nielsen, "Theory of Backpropagation Neural Networks," Proceedings of IJCNN, Vol.1, pp.593-605, 1989.
[3] Y. Hirose, K. Yamashita, S. Hijihya, "Backpropagation Algorithm Which Varies the Number of Hidden Units," Neural Networks, Vol.4, pp.61-66, 1991.
[4] K. B. Kim, M. H. Kang and E. Y. Cha, "Fuzzy Competitive Backpropagation using Nervous System," Proceedings of WCSS, pp.188-193, 1997.
[5] S. N. Kavuri, V. Ventatasubramanian, "Solving the Hidden Node Problem in Neural Networks with Ellipsoidal Units and Related Issues," Proceedings of IJCNN, Vol.1, pp.775-780, 1992.
[6] M. Georipoulos, G. L. Heileman and J. Huang, Properties of Learning Related to Pattern Diversity in ART1," Neural Networks, Vol.4, pp.751-757, 1991.
[7] K. B. Kim, S. W. Jang, and C. K. Kim, "Recognition of car license plate by using dynamical thresholding method and enhanced neural networks," LNCS 2756, pp.309-319, 2003.
[8] T. Saito and M. Mukaidono, "A Learning algorithm for Max-Min Network and its Application to Solve Relation Equations," Proceedings of IFSA, pp.184-187, 1991.
[9] Kishor Trivedi, "Loss Formulas and Their Application to Optimization for Cellular Networks", IEEE Trans. Vehicular Technology, Vol.50,p664-673,2001.
[10] A Guide to DECT features that influence the traffic capacity and the maintenance of high radio link transmission quality, including the results of simulations. http://www.etsi.org, July 1991.
[11] Kishor S. Trivedi, Xiaomin Ma and S. Dharmaraja, "Performability modeling of wireless communication systems.," Internationnal Journal of Communication Systems, pp.561-577, 2003.
[12] V.Mainka and K.S.Trivedi, "Sufficient conditions for existence of a fixed point stochastic reward net-based iterative models",IEEE Trans. Software Engineering, Vol.22,pp.640-653, 1996.
[13] G.Ciardo, K.S.Trivedi, "SPNP Usrs Manual, Version 6.0", Technical report, Duke Univ., 1999.

Kwang-Baek Kim received the M.S. and the Ph.D. degrees from Busan National University, Busan, Korea, in 1993 and 1999, respectively. At present, he is an Associate Professor at the Department of Computer Engineering, Silla University. He is currently an Associate Editor for Journal of Korea Multimedia Society. Also, He is currently a director for Korean Intelligent Information Systems Society. His research interests include Fuzzy Neural Networks and Application, Biomedical Image Analysis, Image Processing, Biomedical system.

CheulWoo Ro received his B.S., M.S. and the Ph.D. degrees from Sogang, Dongguk, Sogang University, Seoul, Korea in 1980, 1982, and 1995, respectively. From 1982 to 1991, he joined ETRI (Electronics and Telecommunications Research Institute) as a senior research member. Since 1991, He has been a professor of Silla University. His research interests include performance analysis of wireless networks, Petri Nets modeling, and embedded network system.

KwangEui Lee received his B.S., M.S. and the Ph.D. degrees from Sogang University, Seoul, Korea in 1990, 1992, and 1997, respectively. From 1997 to 2001, he joined ETRI as a senior research member. Since 2001, He has been an assistant professor of Dongeui University. His research interests include computation theory, artificial life, context awareness and their applications

KyungMin Kim received her B.S. and M.S. degrees from University of Silla, Busan, Korea in 1993 and 2000, respectively.
She is currently a Ph.D. candidate in the Department of Computer Engineering University of Silla. Her research interests include performance analysis of wireless networks, ad-hoc networks and embedded programming.