韓國林學會誌 84(1) : 63-70. 1995
Jour. Korean For. Soc. 84(1) : 63-70. 1995

Performance Analysis of Neural Network on Determining The Optimal Stand Management Regimes[1]*

Joo sang Chung[2] and Joseph P. Roise[3]

임분의 적정 시업체계분석을 위한 Neural Network기법의 적용성 검토[1]*

정주상[2] • Joseph P. Roise[3]

ABSTRACT

This paper discusses applications of neural network to stand stocking control problems. The scope of this research was to develop a neural network model for finding optimal stand management regimes and examining the performance of the model for field application. Performance was analyzed in consideration of the number of training examples and structural aspects of neural network. Research on network performance was based on extensive optimization studies for pure longleaf pine(*Pinus palustris*) stands. For experimental purposes, an existing nonlinear even-aged stand optimization model with a whole-stand growth and yield simulator was used to generate data samples required for the performance analysis.

Key words : *stand stocking control, performance analysis of neural network, optimal stand management regimes*

요　약

이 논문에서는 neural network기법에 의해 소규모 임분의 시업계획을 분석하는 방법과 적용성을 평가하였다. 이를 위해서 적정한 임분시업체계를 계산하기 위한 neural network 모델을 개발하고, neural network의 구조체계와 network을 교육시키기 위해 요구되는 자료량의 측면에서 적용성을 검토하였다. 연구목적상 모델의 교육 및 비교분석에 요구되는 적정 시업체계에 대한 자료는 기존의 비선형 시업체계분석모델을 이용하였다. 이 시업체계 분석모델은 동령급 구조의 긴잎 소나무(*Pinus palustris*) 단순림의 적정시업체계를 분석하는 모델로서 전림수확생장함수에 의해 임분의 생장이 예측되는 모델이다. neural network 모델의 적용성 검토에 요구되는 분석자료들은 이 비선형 시업체계분석모델에 의해 제시된 긴잎소나무 임분의 적정 시업체계분석 결과들을 이용하였다.

[1] 접수 1994년 11월 27일 Received on November 27, 1994.
[2] 서울대학교 산림자원학과 Seoul National University, Suwon, Korea
[3] North Carolina State University, U.S.A.
* 이 논문은 1993년도 Chile의 Valdivia에서 개최된 'System Analysis and Management Decisions in Forestry' 국제심포지움에 발표된 내용의 일부임.

INTRODUCTION

During the last three decades, various mathematical programming approaches have been proposed as decision-making tools for forest stand management. These approaches overcome some of the limitations of conventional silvicultural approaches by finding optimal stand management regimes that consider both biological and financial elements.

However, an optimization model with a comprehensive stand management simulator also has shortcomings as far as use in the field is concerned. Usually forest stand management problems include a number of natural and artificial features that are determined by the management objectives and the current forest stand status. The dynamic nature of stand structure adds to complexity in determining which stand management strategy should be selected from a number of available alternatives.

As a result, optimization models were limited in their use to specific stand characteristics or stand management objectives. In addition, it requires a large amount of computation and very high computing costs for complicated problems despite current advances in mathematical programming technology. This situation results in a need to find an inexpensive and easy-to-use method which can be applicable to a broad range of forest conditions and management objectives.

Because of the complexity and nonlinear patterns in forest stand management problems, conventional inference techniques based on linear decision rules will probably not provide an efficient tool. Neural networks, a recently resurging technology, may have the potential to solve such complex problems. Neural networks, known for their simplicity and outstanding performance in linear or nonlinear pattern classification, may be preferable to conventional methods (Lippmann, 1987).

The scope of this research was limited to developing a neural network model for finding optimal stand management regimes and examining the performance of the model for field applicability in terms of computational efficiency and the generalization capability of neural network through a case study.

NEURAL NETWORK

A neural network is an information processing technique, the structure of which is based on neuron systems such as found in a brain. The attempt to simulate biological computation was pioneered by McCulloch and Pitts (1943). Recently this method has been increasingly used in many fields of applied science.

The architecture of a neural network is shown in Fig. 1. The network is made of processing elements or nodes. A set of nodes composes a layer, which is connected to other adjacent layer(s). Network is composed of 3 types of layers : the input buffer through which a set of input data is presented, the output layer through which the output of process is presented, and the in-between layer(s) referred to as hidden layer(s).

The processing unit (Fig. 2) is analogous to a

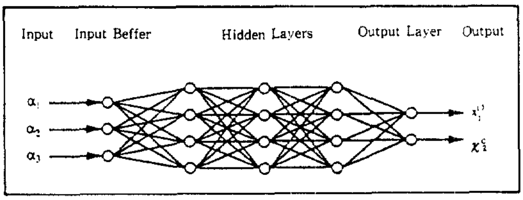

Fig. 1. A theoretical structure of the neural network used to estimate values of dependent variables, x_1^o and x_2^o, as a function of independent variables, α_1, α_2 and α_3.

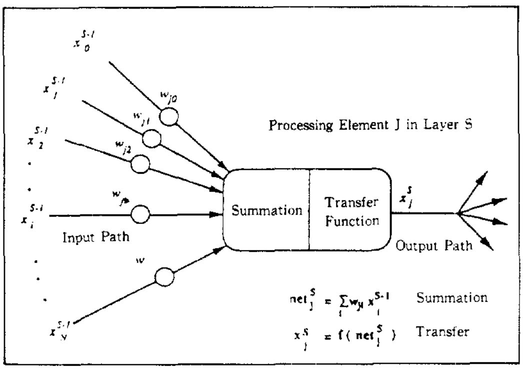

Fig. 2. A processing element j in layer S of a typical neural network showing connection weights, and summation and transfer functions.

biological neuron. It is interconnected to a set of other processing units through input and output paths. Each processing unit has functions of simple summation and activation. Input values, X_i^{s-1} (i=0, 1, ⋯, N), received through input paths from a set of other processing units are combined by summation and the combined value, net_j^s is activated for the subsequent process by a transfer function, $f(net_j^s)$. The output paths are connected to the input paths of a set of processing units in the next layer. Connections are not allowed within a layer or from a higher to a lower layer. As a value passes between the two connected processing units i and j, the input value is weighted by connection weight, w_{ji}, which is analogous to the synaptic strength of a neural connection.

MATERIALS AND METHODS

Even-aged Lognleaf Pine Stand Management Problem

The stand management problem is to estimate the optimal SEV and treatment prescriptions for both timber and pine straw production in longleaf pine stands. Even though industrial management of southern pine forests usually requires a short rotation of less than 50 years, a much longer rotation of 110 years is used in this study to accommodate red-cockaded woodpecker habitats. This long rotation for red-cockaded woodpecker habitat can be economically justified by producing longleaf pine straw as well as timber(Roise et. al., 1991). The management plan is assumed to use a three-cut shelterwood method. The control variables are thinning timing and intensity. Intensive site treatments are also included to control litter, grasses, hardwood and brown spot disease. The financial and production data used in optimization studies are shown in Table 1.

With this problem, networks are designed to infer the SEV and thinning regimes including the timing and intensity by the percent basal area as a function of site index and rotation. Thus, two input nodes (site index, rotation) and three output nodes (SEV, thin time, thin intensity) are required to construct a network.

Optimization Model

To generate data sets required for the analysis, we used a nonlinear programming stand-management optimization model developed by Roise, Chung and Lancia(1991). The optimization model solves for thinning regimes in intensively managed longleaf pine stands which produce both timber and pine straw. In the model, both timber growth and yield and pine straw production are predicted by a whole-stand simulator.

Table 1. Financial and production data used in the analysis of longleaf pine shelterwood systems.

(1) Annual interest rate 4%		
(2) Cost per unit volume		
	Sawtimber	Pulpwood
Stumpage	\$205/MBF	\$43/cord
Logging	\$0.08/ft³	\$0.03/ft³
Transport.	\$0.02/ft³	\$0.03/ft³
(3) Treatment data per unit area		
	cost	Execution times
Site preparation	\$60/acre	Year 0
Burn	\$15/acre	Year 3, 6 and after thin
Chemical treatment	\$9/acre	Every 10 years
(4) Pine straw data		
Market price	\$3.25/bale	
Baling cost	\$1.70/bale	
Weight	62 pounds/bale	
Raking efficiency	80% of needle production is recovered	
Raking frequency	Every other year starting at age 20	

Procedures of Neural Network Analysis

The procedures used here for neural network analysis of stand management consist of three parts : extensive optimization studies, neural network training and performance analysis of the trained network(Fig. 3). The extensive optimization studies provide knowledge for the network to learn. The network can absorb knowledge through repeated training procedures. Then the performance of the trained network for future use can be assessed statistically. The procedures are described as follows.

Optimal stand treatment regimes are determined by an optimization model. The input(initial stand conditions) and output(optimal prescriptions) of the optimization model form an appropriate training set for a network model, and a verification set to examine the generalization capability of the network model(top block in Fig. 3).

In Fig. 3, the T stand data sets are divided into two categories : the training set and the verification set. The training set is the neural network data containing *a priori* knowledge on optimal stand management prescriptions. The verification set is composed of other network data to assess the neural network performance. The two independent samples are needed to train the network and to asses the performance of the trained network. A certain number of stands, h, among T are randomly chosen and assigned to the verification set. Each input vector $X_{i(i=1,2,\cdots,T)}$ is composed of initial stand data. The optimization model produces optimal prescription vector Y_i^* when input vector X_i is presented to the model.

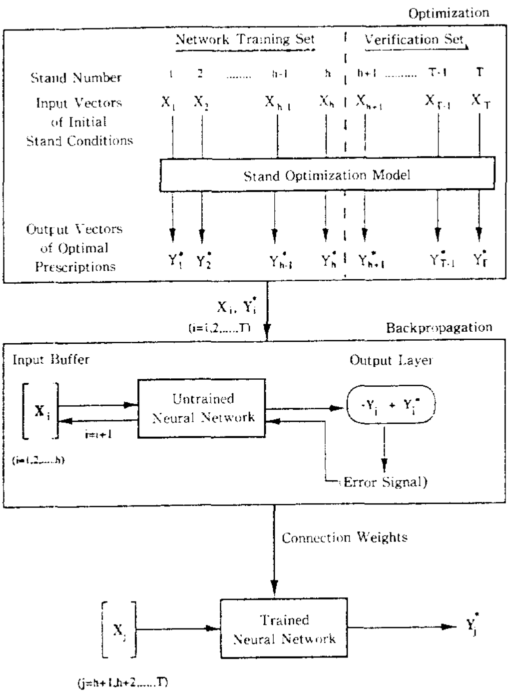

Fig. 3. Procedures to implement the neural network analysis.

During the network training session, both Xi and $Yi^*(i=1, 2, \cdots, h)$ form an input set which is sent to the network. By feeding all these training sets sequentially, the untrained network becomes knowledgeable on all patterns contained in the entire training set. This knowledge is represented by connection weights in the network. The procedures are illustrated in the middle of Fig. 3.

This learning is obtained by means of the backpropagation learning algorithm. When input vector X_i or pattern i is presented to the input buffer, this input is processed parallel in a forward direction by sequentially connected nodes in the network. Each connection between nodes contains the past experience obtained during the training process in the form of real-valued weights. The weighted inputs from all nodes in the preceding layer, via input paths, go into a simple summation function. Activated by these summed inputs, a sigmoidal function in each node transforms the inputs into the proper magnitude of output which is transferred(fired) to all nodes in the next layer. The final outputs from the output nodes form an output vectob Y_i, which is an estimate of the true optimal prescription vector Y_i^*. At each output node, the error derivative or error signal is propagated backwards to the preceding nodes and in the process used to adjust each connection weight. This error signal is used to adjust the weights involved in the node connections. The same backpropagation is repeated at each level until the input buffer is reached. When the input buffer is reached, a new training example, X_{i+1}, enters into the system and the same forward and backward procedures are repeated until all the patterns in the training set produce their output within a predefined tolerance.

Once the predefined tolerance is reached for all the training examples presented to the network, the model has a certain degree of generalization capability depending upon the complexity of actual pattern classification boundaries and the amount of network learning. The trained network can be utilized to determine inference, Y_j, on optimal stand management regimes Y_j^*, $(j=h+1, h+2, \cdots, T)$, by applying new input data X_j, $(j=h+1, h+2, \cdots, T)$, shown at the bottom of Fig. 3.

RESULTS AND DISCUSSIONS

The generalization capability of a trained network can be defined as the ability of a trained network to infer unknown optimal management regimes for a specific stand from past experience of learning during training sessions. Fig. 4 illustrate how well neural networks make inferences on optimal management regimes as a function of the number of training examples used to train the network. It contains three types of graphs for the multivariate stand stocking control problem, of which dependent variables are thinning timing(4a), percent thinning intensity for longleaf pine(4b), and the SEV(4c) when the rotation equals 110 years. In this Fig, the solid line curves represent true optimal solutions, and other scattered plots represent inferred values by networks. The trends of the true optimal stand management regimes for longleaf pine stand management to protect red-cockaded woodpecker habitats include a parabolic SEV curve(Fig. 4c) and two nonconvex thinning regime curves(Fig. 4a and 4 b).

In Fig. 4(c), the networks provide good guesses for the optimal SEV after being trained with three of four examples. However, when the curves of the true solutions show trends with broken lines(as illustrated in Fig. 4a and 4b), the additional information gained from more than four examples results in even better inferences. Eventually, the neural networks get saturated with information, and learning more examples does not result in better inferences(Fig. 4 c). A guideline to determine the size of a training set was not found in the literature. Thus, the question of how much learning is optimal for a specific prob-

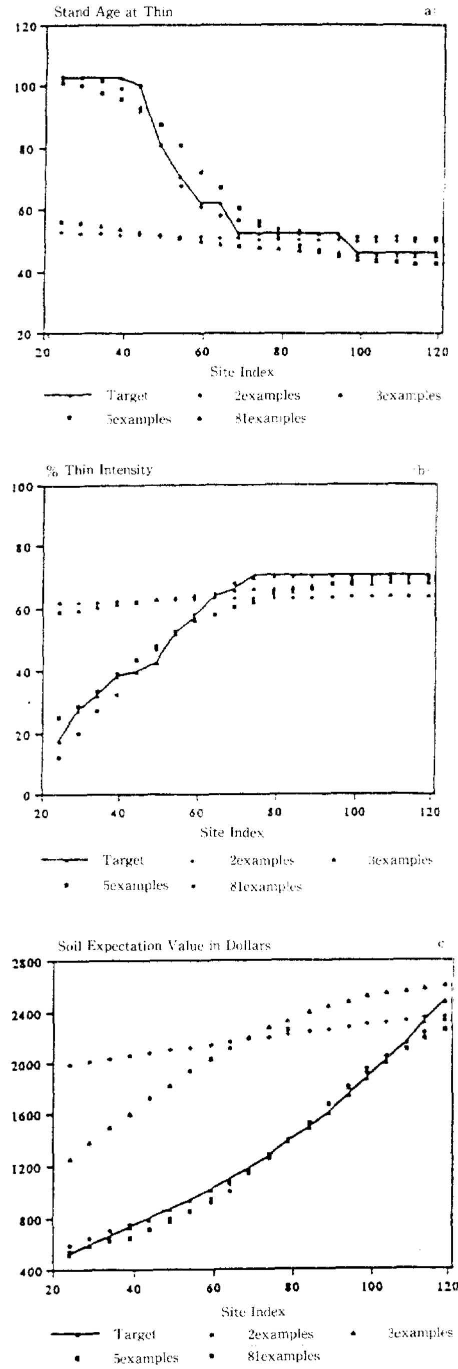

Fig. 4. Fitting neural network solutions to true optimal solutions, in which independent variables are curvilinear functions of site index(rotation=110 years). The dependent variables are (a)the stand age at thinning, (b)the percent basal area thinning intensity, and (c)the soil expectation value.

Table 2. Computation time required do train two-layer networks and quality of estimations as a function of the number of training examples. (NHN and NEX represent the number of hidden nodes per hidden layer and examples, respectively ; c.p.u. is the central processing unit time in seconds.)

NHN	2 NEX	c.p.u.	Confidence Interval Timing	 Inventory	 SEV
30	2	4	(0.40, 0.70)	(0.50, 0.86)	(0.65, 0.90)
30	3	56	(0.25, 0.55)	(0.32, 0.63)	(0.37, 0.68)
30	5	116	(0.18, 0.47)	(0.06, 0.29)	(-0.02, 0.12)
40	40	65, 954	(-0.02, 0.07)	(-0.01, 0.16)	(-0.02, 0.12)

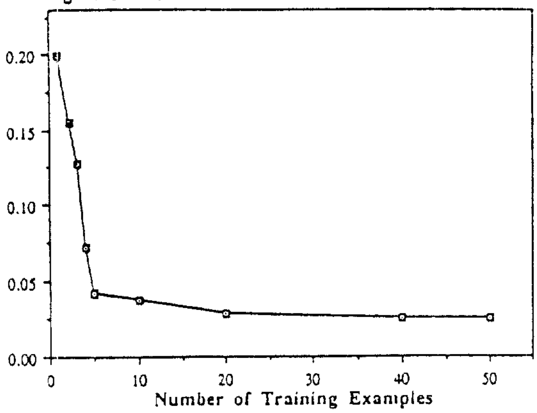

Fig. 5. The average errordistance, as a function of the number of training examples, in vector space from the true optimal point to the estimated optimal point.

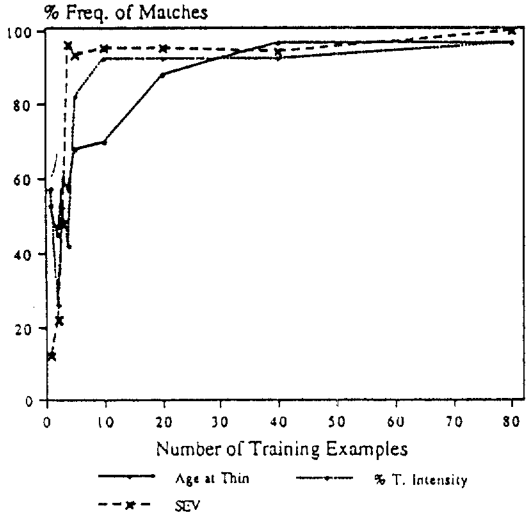

Fig. 6. The percent successful matches as a function of the number of training examples.

lem is a fruitful area for future research.

For the multivariate problem, the average Euclidean distance from an estimated point, which represents a vector of inferred solutions, to a true optimal point is a measure of the error in vector space. Fig. 5 illustrates the average error distance as a function of the number of training examples. In the Fig., the average vector distance was calculated using error values scaled to a range of (0, 1) to unify the different measurement-units of the variables. Each variable was linearly scaled using the upper and lower boundary values in the list of raw data. As shown in Fig. 5, the generalization capability of a trained network is sensitive to the number of training examples for the longleaf pine stand problem. After training with twenty examples for the problem, the networks do not provide significantly better inferences, which indicates saturation of the network information storage capacity.

Fig. 6 illustrates the percent frequency of close matches between inferences and true solutions. It is a direct measure of the networks' capability to infer unknown optimal stand management regimes. The output of a network is considered a close match if the value of each dependent variable is within a margin of 10% of the correct values. Notice the rapid learning rate with a small number of training examples for all variables.

The convergence time required to train each network is another measure of neural network performance. We measured the computing time by running a commercial neural network software using 12 Mhz micro-computers. The convergence time is not a consistent measure of the computational efficiency because it varies according to the training samples chosen. Nonetheless, the convergence time shown in CPU column of Table 2 indicates a very sensitive

Table 3. The c.p.u. time bequired to train single-hidden-layer networks with a different number of hidden nodes when NEX=10.

NHN	c.p.u.	Confidence interval		
		Timing	Intensity	SEV
2	378	(0.16, 0.44)	(0.02, 0.23)	(0.01, 0.19)
4	216	(0.16, 0.44)	(0.02, 0.23)	(0.01, 0.19)
10	674	(0.16, 0.44)	(0.06, 0.29)	(-0.01, 0.16)
30	1,352	(0.16, 0.44)	(-0.01, 0.16)	(-0.02, 0.07)

Table 4. Computation time required to train networks as a function of the number of hidden layers and the number of training examples.

NEX	NHL	NHN	TNC	c.p.u	Time Ratio
3	1	50	250	77	-
3	2	14	266	150	0.51
5	1	50	250	87	-
5	2	14	266	217	0.40
10	1	50	250	1,972	-
10	2	14	266	2,155	0.92
20	1	50	250	6,322	-
20	2	14	266	3,742	1.69

response to the change in the number of training examples. The Table shows sharp curvilinear increases in the computing time as a function of the number of training examples.

In addition, the last three columns in the Table show the tendency for confidence intervals to narrow as the number of training examples increases. Each confidence interval represents an estimate with 95% confidence of a population mean for p1-p2, where p1 and p2 are approximated by point estimates for the percent success, respectively. The above discussions were based on performance of a fixed structure two -layer neural network.

What would be the effect on overall neural network performance if network structure changed? To answer this question, we changed the number of hidden layers and the number of hidden nodes separately, varying the network structures. To measure the effect of changing the number of hidden nodes, the c.p.u. time required to train a series of single -layer neural networks with a fixed number of training examples(NEX=10) and the quality of solutions for verification data sets were observed.

The results are shown as a function of the number of hidden nodes(NHN) in Table 3. In the Table, the c.p.u. time is very sensitive to changes in the number of hidden nodes. The results also indicate there is an optimal network size which minimizes the needed training time for a given neural network problem. However, the optimal number of hidden nodes seems to depend upon the number of dependent and independent variables of the neural network problem and the complexity of patterns presented to the network. There was little evidence, in the the literature searched for this study, that the optimal network structure for a certain neural network formulation can be determined without a series of experiments like the ones shown above.

The results shown in Table 3 also indicate that the quality of neural network solutions is rarely affected by the changes in network structure. Instead, from the results shown in Table 2 and Table 3, it can be concluded that the accuracy of neural network solutions is definitely affected by the number of training examples rather than network structures.

To investigate the effect of changing the number of hidden layers on computing time, two networks, single hidden-layer and two-hidden-layer, were built(Table 4). The number of hidden nodes for each hidden layer was determined for each competing pair of networks to have approximately the same number of node connections(TNC). Then, both types of networks were trained using the same training examples. For comparison purposes, the c.p.u. time needed to train each network is displayed in Table 4 as a function of the number of training examples.

To compare the needed training time between the different number of hidden layers in the above Table, the difference is expressed as the ratio of time, which is a ratio of the time required to train a single-hidden-layer network over the time needed to train a two-hidden-layer network. The ratio of time changes as a function of the number of training examples. As the number of training examples increases, the values of the ratio of time increases. In other words, a network with two hidden layers performs more efficiently than one with a single hidden layer when the number of training examples is arbitrarily large. When th number of training examples is small, the results indicates that a single hidden layer is more efficient.

CONCLUSION

Neural networks were built to determine optimal thinning timing and intensity, and the SEV as a function of site index. Then, neural network performance was measured in terms of the generalization capability as a function of the number of training examples and training time by varying network structures.

For the multivariate problem of longleaf pine shelterwood management the neural networks were efficient in recognizing nonlinear trends of optimal solutions with a small number of training examples.

The efficiency of neural network learning was measured by the c.p.u. time needed for convergence considering network structures and the number of training examples. As the number of examples increases, the convergence time increases exponentially. When the number of training examples is small, single-hidden-layer networks converge faster than two-hidden-layer networks. However, the reverse is true when an arbitrarily large number of training examples is used.

We suggest that further analysis of neural network applications to obtain optimal stand management prescriptions be undertaken. In this research, an intensively-managed longleaf pine stand problem was chosen for experimental purposes. This conclusion can not be necessarily extended to more complicated stand management problems, i.e. mixed hardwood stand problems of which solution patterns are generally less discernible because of ecological complexity of stand dynamics. Thus, a broader range of case studies should be undertaken for future applications of the neural network in forestry. The neural network approach is still in its developing stage and seems to have potential in reducing computational burden for finding optimal stand management regimes.

LITERATURE CITED

1. Furnival, G.M. 1957. Yield prediction of even-aged longleaf pine in the Gulf and Atlantic Coastal Plains. Ph. D. Dissertation, Duke University.
2. Gresham, C.A. 1982. Litterfall patterns in mature loblolly and longleaf pine stands in Coastal South Carolina. Forest Sci. 28(2) : 223-231.
3. Guiver, J.P. and C.C. Klimasauskas. 1988. NeuralWorks : Networks II. NeuralWorks user's guide. (ed.) Klimasaukas, C.C., NeuralWare, Inc., Pittsburgh. p.437-465. 4. Josin, G. 1988. Integrating neural network with robots. AI Expert 8 : 50-58.
5. Lippmann, R.P. 1987. An introduction to computing with neural nets. IEEE ASSP Magazine(April) : 4-22. Neural Systems Incorporated. 1990.
6. Roise, J.P., J. Chung and R. Lancia. 1991. Red-cockaded woodpecker habitat management and longleaf pine straw production : an economic analysis. Southern J. of Applied Forestry 15(2) : 88-92.
7. Roise, J.P., J. Chung, R. Lancia and M. Lennartz. 1990. Red-cockaded woodpecker habitat and timber management : production possibilities. Southern J. of Applied Forestry 14(1) : 6-11.
8. Rumelhart, D.E., G.E. Hinton and R.J. Williams. 1986. Learning internal representations by error propagation. In Rumelhart, D.E., J.L. McClelland(editor), Parallel Distributed Processing : Explorations in the Microstructure of Cognition, Chapter 8. MIT Press, Cambridge, England.