Journal of the Arkansas Academy of Science

Volume 45 Article 17

1991

ANN: A Set of Educational Neural Net Simulators

Siripong Malasri
Christian Brothers University

Stanley P. Franklin
Memphis State University

Follow this and additional works at: http://scholarworks.uark.edu/jaas

Part of the Instructional Media Design Commons

Recommended Citation

Malasri, Siripong and Franklin, Stanley P. (1991) "ANN: A Set of Educational Neural Net Simulators," *Journal of the Arkansas Academy of Science*: Vol. 45 , Article 17.
Available at: http://scholarworks.uark.edu/jaas/vol45/iss1/17

This article is available for use under the Creative Commons license: Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0). Users are able to read, download, copy, print, distribute, search, link to the full texts of these articles, or use them for any other lawful purpose, without asking prior permission from the publisher or the author.
This Article is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Journal of the Arkansas Academy of Science by an authorized editor of ScholarWorks@UARK. For more information, please contact scholar@uark.edu.

ANN: A SET OF EDUCATIONAL NEURAL NET SIMULATORS

SIRIPONG MALASRI
Christian Brothers University
Memphis, TN 38104

STANLEY P. FRANKLIN
Memphis State University
Memphis, TN 38152

ABSTRACT

ANN has been developed on MS-DOS computers primarily for educational uses. Currently, it consists of six simulation programs. ANN1 is a very simple neural net which shows how a network learns by adjusting its connection weights. ANN2 is a single processing element neural net, in which the user trains the network manually by adjusting the connection weights and the threshold value. ANN3 is a manually trained simple two layered network. It demonstrates the power of hidden neurons. ANN4 is a Bidirectional Associative Memory network. ANN5 is a Perceptron that learns from examples. ANN6 is a network based on the backpropagation of error. Graphics have been used extensively in all networks. Students can observe the way these networks learn. Hypertext is used to explain concepts, and also serves as an online user's manual.

INTRODUCTION

Artificial neural networks have become very popular in the past few years. A fascinating feature of a neural network is its ability to learn. Several network models and various training algorithms have been developed and utilized successfully in many applications (Simpson, 1990). Many schools have offered formal courses on the subject. To transfer this technology to students, hands-on simulators are needed to demonstrate concepts. At present, there are several commercial software simulators. These software packages are expensive, are too complex for class uses, and are not designed for educational purposes. There are also several public-domain simulators, which are normally specific to one particular network model and are generally not sufficiently documented.

ANN has been developed with students in mind. It runs on an MS-DOS computer with EGA or higher resolution graphics, 640-K RAM, and a color monitor. It covers six network models. Students can observe the learning process of these networks. Some networks allow students to train them manually. Graphics have been used extensively in all networks. Hypertext is used to explain concepts and to document the package. This paper describes these networks with emphasis on their educational features. For background information on artificial neural networks and hypertext, books should be consulted, such as a book on neural networks by Wasserman (1989) and a book on hypertext by Shneiderman and Kearsley (1989).

HYPERTEXT MODULE

The objectives of the hypertext module are twofold; to provide explanations of underlining concepts, and to provide an online user's manual. The first prototype (Malasri and Franklin, 1991) was developed using KnowledgePro (Knowledge Garden Inc., 1989), a software tool for expert systems and hypertext development. This tool does not provide useful navigation features, such as path history, index, searching. It is also difficult to update large amounts of textual information with KnowledgePro. Hyperties (Cognetics Corp., 1988) is a software tool designed only for hypertext applications. It guides an author through the authoring process. A programming background is not required. It has been used to develop several courseware packages at Christian Brothers University, and was used to document ANN.

When entering the hypertext module, the "introductory node is displayed. Several links are provided on this node to branch out to other key nodes, including the "network models" node. On the "network models" node, there are several links (ANN1, ANN2, ANN3, ANN4, ANN5, and ANN6). Each link leads to one network model in the ANN package. Fig. 1 shows the "ANN4" node, which provides information on the Bidirectional Associative Memory model, as well as instructions for the ANN4 simulation module. Underlined words are links (on the actual screen, links are highlighted). As shown in Fig. 1, texts are displayed on a background picture. When the "Bidirectional Associative Memory" link in Fig. 1 is selected, additional information is displayed as shown in Fig. 2. On each node, there are several default links that help the user to navigate through the hypertext module. "RETURN TO ..." takes the student back to the previous node. "EXTRA" leads to an "Index" node, where all nodes are listed in alphabetical order. Currently, there are a total of 66 nodes in the hypertext module.

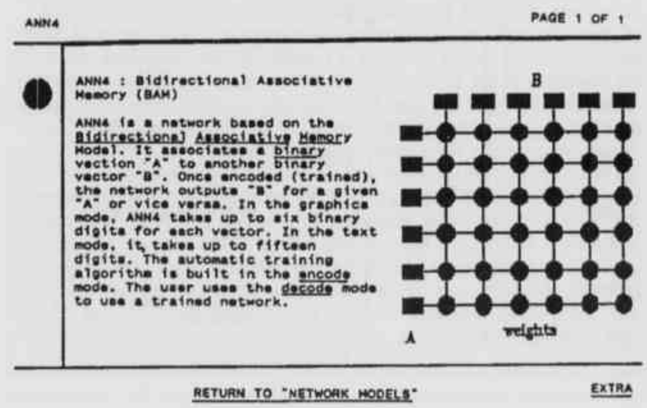

Figure 1. "ANN4" hypertext node.

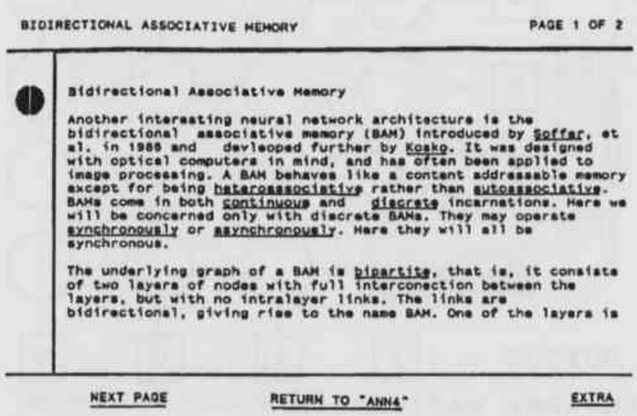

Figure 2. "Bidirectional Associative Memory" hypertext node.

NEURAL NETWORK SIMULATORS

ANN1 – Simple Neural Net

ANN1 is a simple neural network having four input cells fully connected to four output cells. The objective of this module is to introduce the concepts of weighted sum and threshold. Table 1 summarizes data for sample runs of all networks discussed in this paper. For ANN1 the sample run consists of four training pairs (Allman, 1989). Fig. 3 shows a sample screen of the input mode. The circles represent connection strengths (weights). Once the student enters a training pair consisting of an input vector and a target output vector, the network learns by adjusting its connection weights. A weight changes from zero to one when both input and output cells have values of 1. The network recalls by comparing the weighted sum with the threshold, as shown in Fig. 4. If the weighted sum if greater than or equal to this threshold, the output is set to "1"; otherwise the output is set to "0" (zero). The student can change the threshold, which is initially set to zero, until the desired output is obtained. This network may not recall all training pairs.

Table 1. Data for Sample Runs.

Network	Input	Output	Other training parameters
ANN1	1 0 1 0	1 1 0 0	
	0 1 0 1	0 0 1 1	
	1 1 0 0	1 0 0 1	
	0 0 1 1	0 1 1 0	
ANN2	1 1	1	
	1 0	1	
	0 1	1	
	0 0	0	
ANN3	1 1	0	
	1 0	1	
	0 1	1	
	0 0	0	
ANN4	1 0 1 0 1 1	1 1 0 1	
	1 1 1 0 0 1	1 0 1 1	
	1 0 0 1 1 0	0 1 1 0	
ANN5	1 1 1 1 1	1	Learning rate coef. = 1
	1 1 -1 1 -1	0	
	0 1 1 1 1	1	
	1 1 -1 -1 1	0	
	-1 1 1 -1 1	1	
ANN6	1 1	0	Learning rate coef. = 0.9
	1 0	1	Momentum coef. = 0.9
	0 1	1	Accuracy = 0.1
	0 0	0	

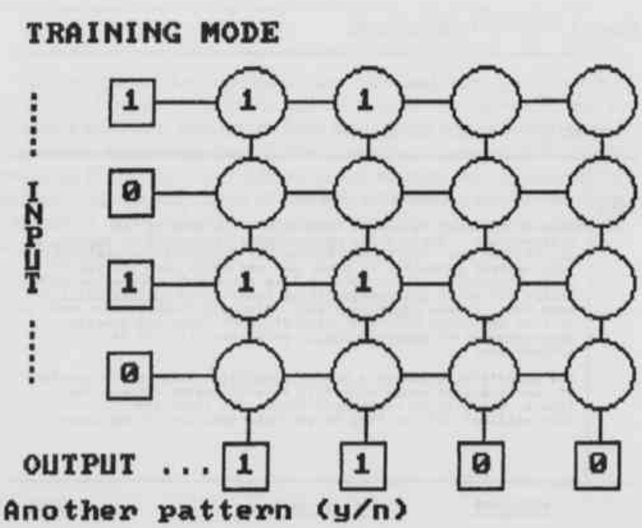

Figure 3. Training mode of ANN1.

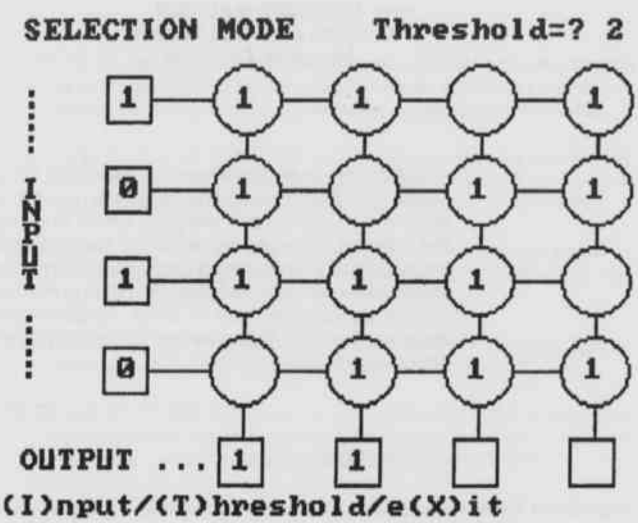

Figure 4. Selection (recall) model of ANN1.

ANN2 – Simple Perceptron

ANN2 produces a single processing element neural network, which allows two binary digits of input and one binary digit of output. The objective of this module is similar to that of ANN1; to demonstrate the concepts of connection weights, weighted sum, and threshold. The weights are, however, not limited to "0" and "1" as in ANN1. A set of training pairs is shown in Table 1. After entering a training pair, the user manually adjusts the connection weights and threshold until the desired output is obtained, as shown in Fig. 5. To be solved by ANN2, the problem must be linear separable. The classic "Exclusive OR" problem cannot be solved with this model.

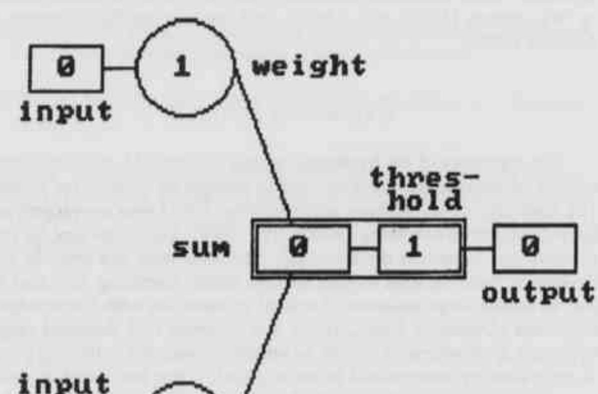

Figure 5. A sample screen of ANN2.

ANN 3 – Simple Multiple Layered Network

ANN3 creates a network with one hidden layer. It has two input neurons (cells), two hidden neurons, and one output neuron. The objective is to demonstrate the power of hidden cells. This network is capable of solving the "Exclusive OR" or "XOR" problem, as shown in Table 1. The user can adjust connection weights, as well as the threshold values until all input vectors produce desired output. Fig. 6 shows a sample screen of this network.

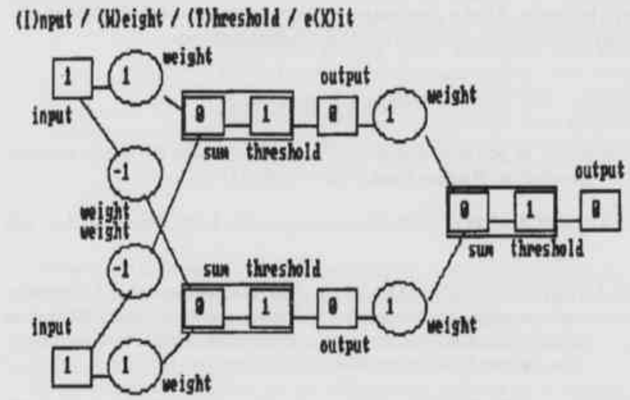

Figure 6. A sample screen of ANN3.

ANN4 – Bidirectional Associative Memory

ANN4 is based on the Bidirectional Associate Memory model. There are two modes: graphic mode and text mode. In the graphic mode, the user observes the changes of weights during a training (encoding) session through the network configuration shown in Fig. 7. Due to the screen display constraint, only six binary digits are allowed for input and output. In the text mode, the number increases to fifteen binary digits for input and output since only matrices of numbers are displayed. The learning process is taken from a textbook (Soucek and Soucek, 1988).

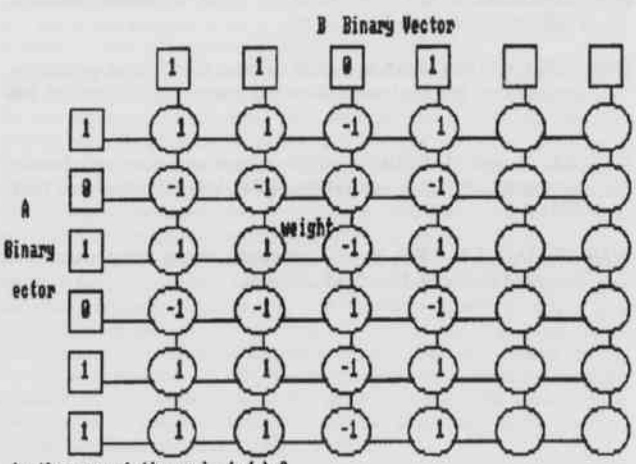

Figure 7. Encoding mode of ANN4.

Fig. 8 shows a recall process, which the vector B can be recalled from the given vector A or vice versa. The user can let the flow go back and forth in both directions until the system converges. Fortunately, this model always rapidly converges, and will not oscillate (Kosko, 1987).

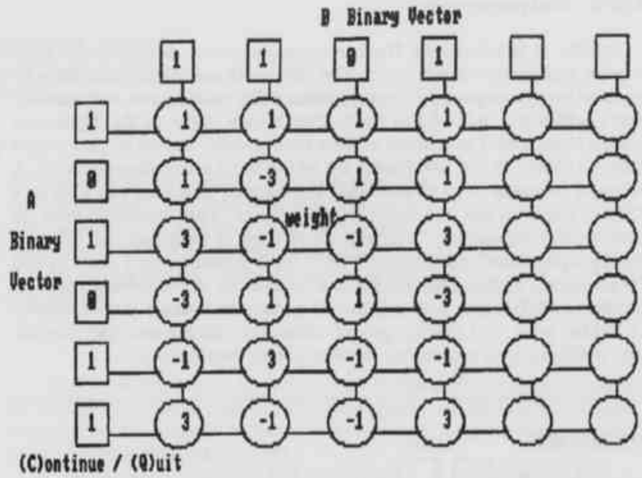

Figure 8. Decoding mode of ANN4.

ANN5 – Threshold Logic Unit

ANN5 provides a single processing element network, similar to ANN2. It, however, allows up to seven binary digits for input as shown in Fig. 9. An automatic learning process is built-in, so that there is no need for a trail-error process as in ANN2. The objectives are to introduce the concept of Hebbian learning rule, delta rule, and learning rate coefficient. "Slow" mode allows the user to observe the weight changes for each cycle. "Fast" mode displays the weights after each cycle continuously. "Bias", an additional input that always has the value of 1, is added to the processing element to speed up the training time. The training process implemented in ANN5 follows an algorithm appearing in a textbook (Soucek and Soucek, 1988). In this process, the weights are adjusted in proportion to the input with the user's specified training rate coefficient. When the training starts, all weights are set to zero. Weights are then adjusted in each cycle until desired output is obtained for all pairs. The recall process uses a similar graphic screen, as previously shown in Fig. 9, with the "desired output" box omitted.

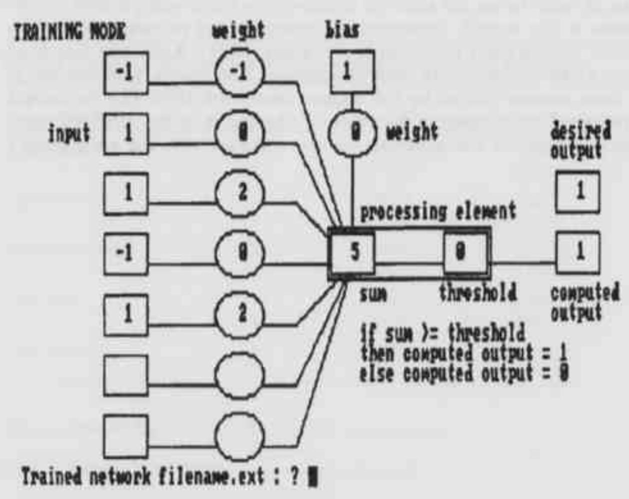

Figure 9. ANN5 network configuration.

ANN6 – Backpropagation

ANN6 is based on the Backpropagation model, which is the most popular architecture in use today. The objectives are to introduce the concepts of backpropagation of error, hidden cells, training rate and momentum coefficients. ANN6 has up to four binary digits of the input, one hidden layer with a maximum of four hidden cells, and up to four output cells, as shown in Fig. 10. Biases are added to all processing elements. A sigmoid function is used to obtain the output. The user can choose a "Slow" mode to see one cycle at a time or a "Fast" mode to have all cycles run continuously. The algorithm used is based on the "Vanilla Backpropagation" (Simpson, 1990) with a momentum term added (Wasserman, 1989). This model can be used to solve several practical problems. Difficulties with this model include choosing a proper number of hidden cells and finding proper values for training rate and momentum coefficients. It sometimes requires a long training time.

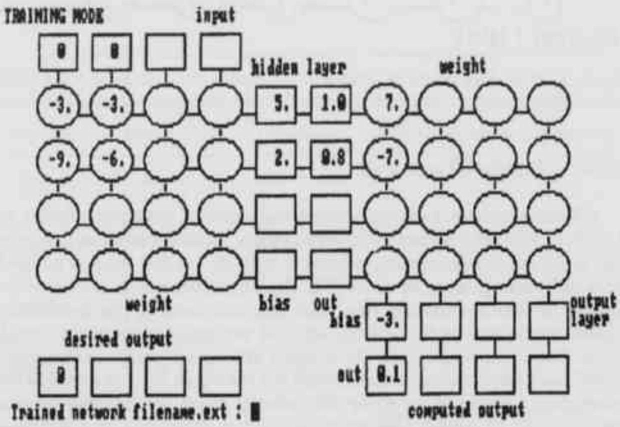

Figure 10. ANN6 network configuration.

CONCLUSION

This paper briefly describes each network model in the ANN package. In order to use the software effectively, a set of well-planned assignments is also equally important. At present, some of these assignments are in printed form (Franklin and Malasri, 1991). ANN was first used with 19 faculty members from 18 academic institutions, who enrolled in a short course offered by the authors under the 1990-91 Chautauqua Program. One question in the course evaluation is on the ANN software. Each participant was asked to rate the software with the scale from 1 (poor) to 5 (excellent). The average rating was 4.43. We are very encouraged with this result. Several other network models are being developed and will be added to the package. The next version will be tested with a graduate class on "Knowledge Engineering" under the Engineering Management Program at Christian Brothers University, which covers expert systems. hypertext, and artificial neural networks (Malasri *et al.*, 1992). At that time, more extensive evaluation on the hypertext modules will be made. After a few years, we hope to cover most of popular network models with extensive hypertext documentation.

LITERATURE CITED

ALLMAN, W.F. 1989. Apprentices of wonder, inside the neural network revolution. Bantam Books, New York, 211 pp.

COGNETICS CORP. 1988. Hyperties version 2.35j. Princeton Jct, 156 pp.

FRANKLIN, S.P. and S. MALASRI. 1991. Neural networks: An exciting new paradigm for computing and for cognitive science. Workshop notes. Chautauqua program. Christian Brothers University Chautauqua field center, Memphis, 117 pp.

KNOWLEDGE GARDEN INC. 1989. KnowledgePro version 1.43. Nassau, 360 pp.

KOSKO, B. 1987. Constructing an associative memory. BYTE. Hightstown, 12(9):137-144.

MALASRI, S., R.C. BLACKMAN, and K.W.TOUCHSTONE. 1992. Teaching Emerging Technologies: A CBU Experience. Proceedings of the 3rd International Conference on Management of Technology, Miami (in press).

MALASRI, S. and S.P. FRANKLIN. 1991. Educational artificial neural networks, an integration of hypertext and simulation software. Proceedings of the annual meeting (New Orleans). American Society for Engineering Education, 2:1009-1013.

SHNEIDERMAN, B. and G. KEARSLEY. 1989. Hypertext hands-on!. Addison-Wesley, Reading, 165 pp.

SIMPSON, P.K. 1990. Artificial neural systems, foundations, paradigms, applications, and implementations. Pergamon press, Elmsford, 209 pp.

SOUCEK, B. and M. SOUCEK. 1988. Neural and massively parallel computers, the sixth generation. Wiley-Interscience, New York, 460 pp.

WASSERMAN, P.D. 1989. Neural Computing, theory and practice. Van Nostrand Reinhold, New York, 230 pp.