

# Neural Networks: Implementations and Applications

**E. Vonk and L.P.J. Veelenturf**  
*University of Twente*  
and  
**L.C. Jain**  
*University of South Australia*

## KEYWORDS

Neural Network, Back Propagation, Radial Basis Function, Kohonen, ART1, Hopfield, Bidirectional Associative Memory, Character Recognition

## ABSTRACT

Artificial neural networks, also called neural networks, have been used successfully in many fields including engineering, science and business. This paper presents the implementation of several neural network simulators and their applications in character recognition and other engineering areas.

## INTRODUCTION

In the last five years the reported applications of neural networks have multiplied exponentially. These networks have proven to be successful in many applications including fault diagnosis, financial control,

forecasting, plant control, industrial process control and pattern recognition.

Despite tremendous interest in neural networks, there are still many research problems which need careful attention. For example, in the back propagation network, parameters such as learning rate and momentum term are found by trial and error.

The mathematical basis for finding the number of neurons in the hidden layer are reported in the literature [2], but there are still some problems associated with it. Often the researchers adjust the neurons in the hidden layer by trial and error. There are not enough guidelines pertaining to which architecture is suitable for a given application.

This paper presents the implementation of back propagation, radial basis function, Kohonen net, ART1, Hopfield net and BAM. A comparative study concerning a character recognition application as well as other applications are also presented.

## NEURAL NETWORK SIMULATIONS

This section describes the neural network simulators and their respective software implementations used in this study.

### The Back Propagation Network (BP)

The back propagation network consists of an input layer of neurons (not shown), an output layer of neurons and at least one hidden layer as shown in Figure 1, on next page.

---

This article is based on a short version presented in "Electronics Technology Directions to the Year 2000 Conference" in Adelaide, Australia, May 1995.

#### Authors' Current Addresses:

E. Vonk and L.P.J. Veelenturf, Control, Systems and Computer Engineering Group (BSC, Laboratory for Network Theory, Department of Electrical Engineering, University of Twente, The Netherlands; and L.C. Jain, Knowledge-Based Engineering Systems Group, School of Electronic Engineering, University of South Australia, Adelaide, Australia.

Manuscript received March 11, 1996.

0885-8985/96/ \$5.00 © 1996 IEEE

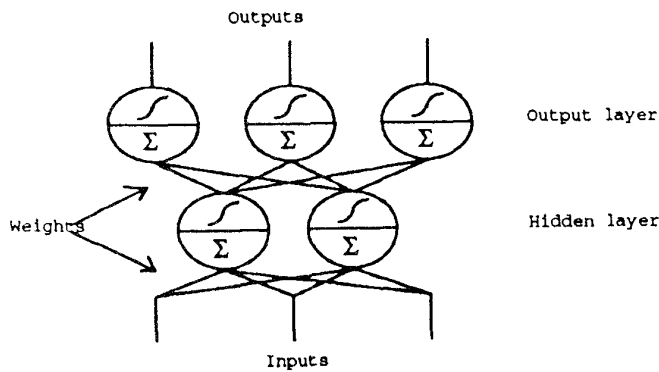


Fig. 1. A '3-2-3' Back Propagation Network

The neurons perform a weighted sum of their inputs and use this sum as the input of an activation function which is usually the sigmoid function. A supervised learning algorithm [2, 8] is used to teach the network. It consists of updating all the weights of a network until the output of the network reaches the pre-specified desired output. The factors responsible for the training and performance of the network include:

- the initial values of the weights (usually random);
- the number of training cycles;
- the number of hidden neurons;
- the training set; and
- the values of the teaching parameters (learning rate, momentum term).

#### The Radial Basis Function (RBF) Network [4, 6, 7]

The Radial Basis Function network consists of three layers. The hidden layer is used to cluster inputs of the network; neurons in this layer are, therefore, called cluster centers. Though its architecture is similar to a three layer back propagation network, its working is different. It uses a Gaussian kernel function to calculate the activations of the neurons in the first layer. The neurons in the output layer perform an ordinary lineal weighted sum of these activations. The learning in this network is done in two stages. First, the input-training patterns are clustered (unsupervised) to their nearest cluster centres by means of a clustering algorithm. The K-Means clustering algorithm is used. Second, the spread of each cluster centre is determined. This can be done in several ways; they can be made equal to the average distance between the cluster centre and the training patterns clustered with that centre [4]; or they can be set equal to the average distance between the cluster centre and its nearest cluster centres (often two) [6].

The second stage of learning is the supervised learning of the weights of the output layer. This is usually done by the least mean square algorithm. The

initial value of the output weights are usually set to small random values.

The performance of the Radial Basis Function network depends on the way the inputs are clustered. As with the back propagation network, the number of neurons in the first layer greatly influences the performance of the network. The training time of this network is typically orders of magnitude smaller than that for the back propagation network as training is split up in two parts both of which can be done quickly. In our program, the only teaching parameter is the learning rate. The calculation of training and test errors are identical to the back propagation program.

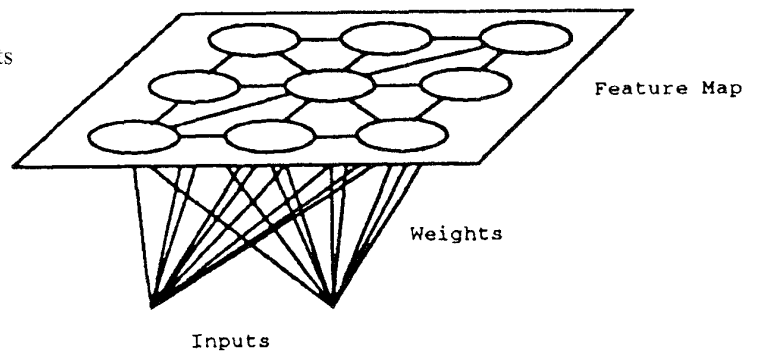


Fig. 2. A '2-3-3' 2-Dimensional Kohonen Network

#### The Kohonen Self-Organising Feature Map [5]

The Kohonen self-organising feature map has two layers of neurons. The signals from input neurons (also called inputs) are fed to every single neuron. While learning (unsupervised), the network makes a two-dimensional representation of the input space. The multi-dimensional input space is transferred to a two-dimensional grid using this network. Figure 2 shows a Kohonen network that has two inputs and a grid (or feature map) consisting of 3 by 3 = 9 output neurons.

Learning in the Kohonen network is performed by selecting the winning neuron for a certain training input. This neuron has the weights that most closely resemble the values of the inputs. The weights of this neuron, as well as of its neighbours, are updated to resemble like the input vector. The neighbours are determined by a neighbourhood function. The radius of this neighbourhood function determines how many neurons around the winning neuron are included in the neighbourhood. There are several shapes possible for the neighbourhood function. A square shape, a bubble (circle) shape or a hexagonal shape are the ones usually used.

In order to train the network properly both the learning rate and the radius of the neighbourhood function should increase in time. The initial weights of the network should be different from each other and are often set to random values in the range of the possible input values.

The learning process consists of two phases. In the first phase, coarse learning, the learning rate is quite large

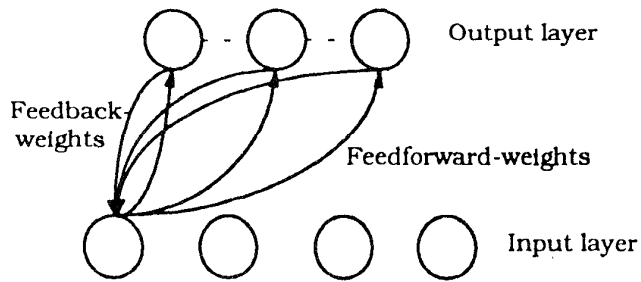


Fig. 3. A '4-3' ART1 Network

and the neighbourhood radius decreases from encompassing a large part of the grid to encompassing only the winning neuron itself (radius = 0) or the winning neuron and its closest neighbours (radius = 1). In the second phase the learning rate slowly decreases to zero (this is usually the stopping criterion) while the neighbourhood radius remains 0 or 1.

The learning rate as well as the neighbourhood radius can decrease by a number of functions; e.g., linearly, exponentially. According to [5], this is not very critical. The total number of training steps (= the number of times a single input training pattern is presented to the network) is typically 500 times the number of network units or less, with the first phase of learning lasting only about 1000 cycles.

#### The ART1 Clustering Algorithm [8]

The ART1 network consists of two layers; a layer to which the inputs are applied and a top layer of neurons (outputs). The two layers are fully interconnected with feedforward (from input-layer to top-layer) weights (the "W-weights") as well as feedback weights (the "V-weights").

The "W-weights," or clustering weights are used to select the winning output neuron (the cluster) and serve as one long term memory of the network. The winning neuron is the neuron with the maximum response (= weighted sum) for a certain input pattern. The "V" or "vigilance weights" are used for the vigilance test and serve as the short term memory of the network. When a training input is similar enough to the last pattern clustered with the winning neuron, it will be clustered with this neuron. Otherwise, the neuron with the second largest response is tried. The required similarity is determined by the vigilance threshold. If the input does not cluster with any of the existing clusters (output neurons), then a new cluster centre is made. The vigilance threshold (0 to 1) determines how similar two patterns should be before they are clustered together. The larger the threshold, the less patterns are clustered together and, therefore, the more clusters will be made. When an input pattern is clustered with an existing or a new cluster centre, its weights are updated. The "V" weights are set equal to this input vector and the "W" weights are set to the normalized input vector. While learning this way, the network works as a "follow the leader" system.

Testing or recall of ART1 is done by feeding a pattern into the input layer and selecting the neuron with maximum response (the winning neuron) using the feedforward weights. The output of the network (i.e., the winning neuron) is the one whose weights are most similar (the smallest Hamming distance) to the input (test) pattern.

Figure 3 shows an ART1 network with 4 inputs and 3 output neurons (clusters); a '4-3' network. Only the weights for one input neuron are shown in the figure.

The ART1 network can only be used for binary input values. An enhanced version of the algorithm exists, ART2, that can handle continuous valued inputs.

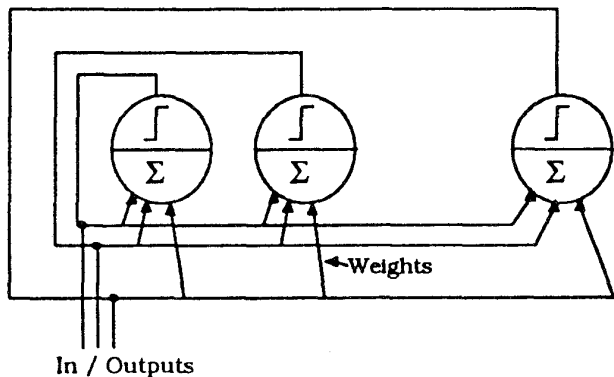


Fig. 4. A 3-Neuron Hopfield Network

#### The Hopfield Network [3, 4, 8]

The Hopfield network is a fully connected single layer network. After an input pattern is presented to the network, it will converge by means of a state update rule until it resides in a stable pattern. The state of a neuron is simply its activation (-1 or +1) and the starting state of the network is just the input. The network can only handle bipolar inputs or binary inputs, with a slightly altered update rule. This is the test or recall procedure. The learning or storing is done simply by setting the weights in the network according to  $w_{ij} = s_i \cdot s_j$ , where  $w_{ij}$  is the weight between neurons  $i$  and  $j$ , and  $s_i, s_j$  are the inputs of neurons  $i$  and  $j$ . Figure 4 shows a 3-neuron Hopfield network.

Usually the weights from a neuron back to itself (i.e., all the weights  $w_{ij}$ ) are set to zero. When an input is presented to the network, the output of the network will be the stable (trained) pattern that has the minimum Hamming distance from the input pattern. Because of the limited storage capacity, the Hopfield network is not very popular in pattern recognition applications.

In [1], a neural network system is used that consists of a single Hopfield network for every training pattern.

There is virtually no such thing as training time for the Hopfield network. The weights are calculated after the training patterns are presented and are set to their appropriate values. There is no updating of weights. The

recall phase does take time because the network has to undergo several convergence steps before the network has found its stable state.

Two “versions” of the Hopfield network have been implemented. One program is just the ordinary Hopfield network. The other consists of a system of multiple Hopfield networks. After the network is trained, it generates as many Separate Hopfield networks as there are training patterns and stores one training pattern in every one of them. In the recall phase, the test input is presented to each of the networks. The output (after convergence) of each network is compared with the input pattern and the final output is that output pattern which most closely (smallest Hamming distance) resembles the input pattern.

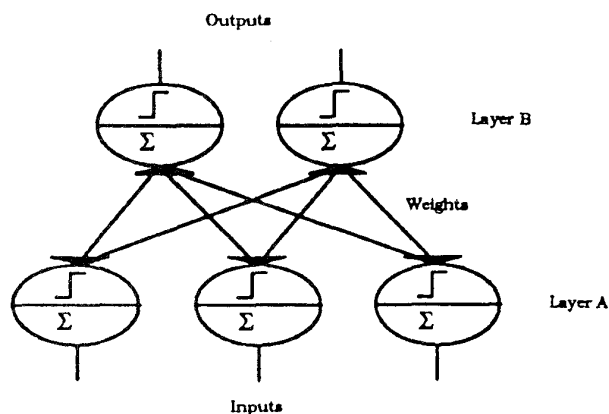


Fig. 5. A 3\*2-Neuron Bidirectional Associative Memory

#### The Bidirectional Associative Memory (BAM) Network

The Bidirectional Associative Memory is quite similar to the Hopfield network. This type of network has two layers of neurons, which are fully interconnected. In the recall phase, an input (test) pattern is presented to one of the two layers. After the recall (test) phase, the network has come to a stable state so that each of the two layers has a stable pattern as output. The updating of the states of the neurons (-1/+1) is done for the two layers alternatively, which means that there is a bidirectional data flow while the network is converging.

The store (learning) procedure is similar to the Hopfield network except that the BAM network needs inputs to both layers. Figure 5 shows a BAM network which has a layer of three neurons (layer A) and a layer of two neurons (layer B).

For the BAM network, just like the Hopfield network, the storage capacity is probably its main limitation. From [8], it follows that an optimistic value of the maximum storage capacity is:  $\min(m,n)$ , where  $m,n$  are the number of neurons in each of the layers. Thus, at most, the smallest of  $m$  and  $n$  patterns can be stored in the network. A more conservative value is  $[1]:\sqrt{\min(m,n)}$ .

## CASE STUDIES

### Character Recognition

A simple comparative study was performed on the neural network simulators described above. The various simulation programs were compared on properties such as speed and generalization capability (i.e., the performance on distorted input patterns). The training set consisted of ten digits, '0' to '9' in a  $5 \times 7$  grid as shown below.

```

00100 01110 11111 00010 11111 00110 11111 01110 01110 01110
01100 10001 00010 00110 10000 01000 00001 10001 10001 10001
00100 00001 00100 01010 11110 10000 00010 10001 10001 10001
00100 00010 00010 10010 00001 11110 00100 01110 01111 10001
00100 00100 00001 11111 00001 10001 00100 10001 00001 10001
00100 01000 10001 00010 10001 10001 00100 10001 00010 10001
01110 11111 01110 00010 01110 01110 00100 01110 01100 01110

```

The network architectures which were determined after various trial runs are: a  $35 \times 10 \times 10$  BP network, a  $35 \times 10 \times 10$  RBF network, a 2-dimensional  $6 \times 6$  Kohonen network and a multiple Hopfield network consisting of 10 separate networks. The single Hopfield network and a  $35 \times 10$  BAM network were not able to learn all the training patterns. Due to the limited storage capacity of these networks only five patterns could be stored.

pixels distorted:	BP	RBF	Kohonen	ART1& Hopfield
0	0.006	0.00013	-	-
1	0.031	0.180	-	-
2	0	0	0	0
3	0.023	0.396	-	-
4	0	0	0	0
5	0.103	0.742	-	-
	0	1	0	0
	0.334	1.08	-	-
	1	1	2	0
	1.04	1.60	-	-
	1	2	4	1

Fig. 6. Cumulative Errors (top left), and Number of Incorrectly Classified Patterns (bottom right) on Training and Test Sets

Figure 6 shows the cumulative training and test errors as well as the number of incorrectly classified facts for all the test sets. Since only one test set was made for each level of distortion, conclusions from this should be drawn carefully. For the Kohonen, ART1 and Hopfield networks, no cumulative errors exist.

It follows that the ART1 and the multiple Hopfield network give the least misclassifications on the test sets. These two networks also showed the best performance in speed; learning took a few seconds as compared to 4 minutes for the RBF network, 10 minutes for the

BP network and about 20 minutes for the Kohonen network.

In all, the ART1 showed the best performance on this simple test, as it is less computationally expensive than the multiple Hopfield network.

#### **Further Applications**

The following is a brief summary of some of the projects undertaken on the application of neural networks in the Department of Electrical Engineering of the University of Twente. Other projects include handwritten digit recognition, automatic recognition of registration plate numbers of vehicles, identification of the dynamic pattern of signatures, speech recognition, machine condition monitoring, signal source separation, electric power load forecasting, automatic detection of spike-wave-complexes in EEG-signals, detection of cancer tumours in X-ray photographs, automatic control of a metal sheet bending machine and process parameter identification in plastic production.

#### *Adaptive Recruitment Learning Rule for Hyphenation*

This project involves the use of a neural network in developing a hyphenation system. During the last decade, several software programs were developed using conventional techniques to break words with a hyphen. These programs are, in general, complicated because they are mostly based on grammatical rules and exceptions of these rules. In our experimental study, we have used 70,000 words of a dictionary of hyphenated words of a Dutch dialect: "Fries." These words were broken up into word segments with a length of 7 characters and the network scanned whether to hyphenate after the third character of a word segment or not. A three layer binary perceptron was used and it was trained using the adaptive recruitment learning rule. This is a supervised training rule that adds new hidden neurons to the network when a training pattern is misclassified. This way, a 100% correct performance is achieved. The language alphabet was coded into 26 8-bit vectors. The network's input consists of the word segments coded as binary vectors of length  $7 \times 8 = 35$ , which means an input layer consisting of 35 neurons is used. A single output neuron is used to determine whether the word segment should be hyphenated or not. After learning the 250,000 word segments, a binary neural network of approximately 20,000 hidden layer neurons was obtained.

#### *A Back Propagation Network in Machine Condition Monitoring*

It is useful to monitor the condition of a machine to avoid major damage. It is necessary to know the condition of the machine continuously to predict the time at which a machine needs maintenance. It is achieved by observing the vibration characteristics of the machine. A set of consecutive samples of the vibration signal is taken and with the Fast Fourier Transform, the frequency spectrum

of the signal at the particular time interval is determined. This frequency spectrum is then represented by a spectral vector which is used as the input to a three layer back propagation network. These spectral vectors are divided into classes corresponding to different categories of machine condition. We have used the vibration signals of four ball-bearings where one of the bearings was slightly damaged. The spectral vectors consisted of 64 components each and a  $64 \times 4 \times 1$  back propagation network was used, where the value of the output neuron is used to distinguish between a damaged and an undamaged bearing. Correct classification was observed even on data that was not a part of the training set.

#### *The Kohonen Network for Picture Color Quantization*

Picture color quantization is needed when, for example, a picture in True Color Format with 16 million RGB (Red, Green and Blue) colors has to be displayed on a EPIC monitor with only 256 colors. Two things have to be done: the number of colors has to be reduced and we have to map every color of the original color onto one of the 256 colors. The selected set of colors, the color palette, can be chosen such that the different colors are evenly distributed over the total range of RGB colors. However, by selecting the colors according to the probability of colors in the original picture, a better picture quality can be achieved. This can be accomplished by a vector quantization of the colors in the original picture to the new color palette. We have chosen a Kohonen network for this task. 10,000 randomly chosen vectors from a picture were used to train the network. The network used a 3-dimensional network with size 256 in every dimension. After learning the weights of the network represent the color palette for the specific picture. This was done for four different pictures and the quality of the final pictures was very good. This result was compared to the classical vector quantization method of Heckbert. It was found that the quantization noise was less when the Kohonen network was used.

#### **CONCLUSIONS**

This paper has described the implementation of neural network simulators in brief. It is obvious from the applications that neural networks are suitable for a wide range of applications in business, science and engineering.

#### **ACKNOWLEDGMENTS**

E. Vonk implemented the networks during his ten weeks work experience training in the Knowledge-Based Engineering Systems Group, University of South Australia, Adelaide, in 1993. Thanks are due to the Department of Electrical Engineering, University of Twente, for permitting us to undertake this research work.

## REFERENCES

- [1] L. Chen, J. Fan and Y. Chen, 1991, A Modified High Speed Hopfield Neural Network and Its Character Recognition System, *Proceedings ISIC-91*, pp. 135-140.
- [2] G. Mirchandani, 1989, On Hidden Nodes for Neural Nets, *IEEE Transactions on Circuits & Systems*, 36, 5, pp. 661-666.
- [3] J. Hopfield, 1988, Artificial Neural Networks, *IEEE Circuits and Devices Magazine*, pp. 3-10.
- [4] D. Hush and B.G. Horne, 1993, Progress in Supervised Neural Networks, What's New Since Lipmann, *IEEE Signal Processing Magazine*, pp. 8-39.
- [5] T. Kohonen, 1990, The Self-Organizing Map, *Proceedings of the IEEE*, 78, 9, pp. 1464-1480.
- [6] J. Moody and C.J. Darken, 1989, Fast Learning in Networks of Locally-Tuned Processing Units, *Neural Computation*, 1, pp. 281-294.
- [7] M.T. Musavi, et al., 1992, On the Training of Radial Base Function Classifiers, *Neural Networks*, 5, pp. 595-603.
- [8] J.M. Zurada, 1992, (*Introduction to*) *Artificial Neural Systems*, West Publishing Company, St. Paul.



# Meeting Summary of Spring AESS Board of Governors Meeting

Held at the National Radar Conference,  
Ann Arbor, Michigan ♦ May 15, 1996

President Charles Gager called the meeting to order and welcomed the 15 BoG members, Past President, Editors, and guests present.

The minutes of the last BoG meeting (January 25, 1996) were approved as written.

President Gager reviewed his report to the Board. The Board discussed and agreed to the following preliminary meeting schedule for 1997:

- Winter:** January 16, Battery Conference,  
Long Beach, California;
- Spring:** May, NAECON, Dayton, Ohio;
- Fall:** September, Electronics Museum,  
Baltimore, Maryland

Past President Ed Reedy reported on the recent TAB meeting. The previously passed motion allowing Societies to charge fees to Life Members was withdrawn and tabled at the TAB meeting. A subcommittee, including Myron Kayton, has been formed to develop a position on Life Member dues and assessments. Funds were approved for use by the Chapter Coordinator to travel to the Sections Congress (\$1,000.00).

The President reminded all Officers that they should write their reports focusing on the items assigned to their position in the Five Year Operating Plan.

Myron Greenbaum provided the Executive Vice President's report. Discussion followed on possible publishing opportunities

with the IEEE Press. David Shu of Litton was identified as a potential author of a book that AESS could sponsor. Walter Fried is responsible for assembling the team of reviewers for the Hughes Aircraft book, *Introduction to Airborne Radar*. Bob Hill suggested that the Radar Panel Chairman be a part of this effort.

During this discussion on publishing, it was suggested that the AESS Technical Panels be contacted every year about possible books for AESS sponsorship. The Executive Vice President will be responsible for making the contacts.

Dan Snyder, Treasurer, distributed copies of his report. He reviewed the 1995 budget actuals, the 1996 budget forecast, and the 1997 budget inputs. The key element to keeping the AESS budget positive in 1996 is the collection of voluntary page charges.

The Treasurer reviewed the conference income chart he prepared. There are several conferences that owe money to AESS. The Treasurer and Vice President-Conferences will be contacting these conferences. The NTC has sent in a check for \$3,000.00. A check for \$25,000.00 from the 94 PLANS Conference has also been received.

First budget inputs for the 1997 budget are due to IEEE in May. Dan reviewed the proposed budget actions in his report. The proposal would charge non-IEEE members \$110.00 for *Systems Magazine*; publish four *Transactions* with a total of 1,216 pages; and charge non-IEEE members \$250.00 for the *Transactions*. The Board also discussed establishing prices for

*Continued on page 26*