Pergamon

Neural Networks, Vol 7, No. 1, pp 209–212, 1994

0893-6080/94 $6 00 + 00

BOOK REVIEW

Neural Networks: Algorithms, Applications, and Programming Techniques

By James A. Freeman and David M. Skapura, Addison–Wesley Publishing, Reading, MA, ISBN 0-201-51376-5.

Freeman and Skapura state that their interest is to borrow neuroscience concepts for use in solving problems in science and engineering. They acknowledge that artificial neural system (ANS) models may not always have biological relevance, but are generally inspired by neuroscience. References are given to other books that provide neurobiological background.

After the usual discussion of history, starting from the 1940s and going through the period of decline in the 1960s, the authors refer to the current revival of interest in ANS. This includes the growth of funding for research and development, conferences dedicated to ANS, professional societies, and programs at universities. Major journals and conferences in the field are also listed.

The authors state that the current book's material comes from short courses and seminars for computer science and engineering graduate students, and people with degrees, and that AI experience is not absolutely necessary. They state that their goal in this book is to survey basic neural network architectures that everyone should know, and to impart a solid background rather than discuss the latest research results. They envision the use of their text for self study and in the classroom. The book's level is that of advanced undergraduate or beginning graduate students in many disciplines. Mathematical prerequisites include calculus, differential equations, and advanced engineering analysis. Mathematical derivations and exercises are included in the text. The authors assume that the readers are practitioners, who will use neural nets to solve real problems. Hence, material on ANS applications in engineering is included.

The authors state that there is sufficient material for a two-semester course, and specify which chapters are prerequisites for other chapters. They also state that computer simulations (in Ada or C) are necessary for obtaining the "full benefit from the material."

Chapter 1: Introduction to ANS Technology

In Chapter 1, several questions are posed. Why can't computers think? Why can't computers comprehend shapes in images or learn from experience? The authors say that these tasks are difficult to solve on sequential computers, and that their book introduces several parallel architectures that may help solve these problems.

An example is provided to illustrate the difficulty of visual pattern recognition: spotting a Dalmatian against a cluttered background. They argue that conventional image processing would not work, even though its elements operate faster than biological neurons. Perhaps neural nets can eventually solve such problems, but the authors admit that even a neural net would have a difficult time with this example.

Next, the authors give an example of a network composed of threshold logic units, having 80 binary-valued inputs, many hidden units, and 10 output units. They use the network to recognize handwritten numerals. The authors fail to mention that conventional discriminant function methods exist for solving this example problem, methods that also utilize coefficients or weights.

After a brief discussion of the chemistry of neurons and synapses, the authors review the work of McCulloch and Pitts, and point out that propositional logic describes these nets. Hebbian learning is described and related to changes in the synapses in biological neurons. In exercise 1.3, the authors describe an autocorrelation matrix but mistakenly call it a KL matrix.

Next, the authors describe the processing element (PE) or artificial neuron. They define the net and activation functions, and show that a differential equation can be used to relate them. The concept of linear separability, as introduced by Minsky and Papert in their book, is reviewed. The scaling difficulty in neural nets is introduced.

The authors discuss methods for simulating neural nets on sequential machines. They mention the need to adapt the connectivity and number of PEs to different applications and different network types. They discuss two alternative structures for representing the network on a machine, linked lists and arrays, and conclude that arrays will lead to faster code.

At the end of the chapter, other suggested reading is listed, including the PDP series (Rumelhart, Hinton, & Williams, 1986), Lippmann (1987), Pao (1989), and many other sources.

Chapter 2: Adaline and Madaline

The authors start with a review of analog signal processing, including linear modulation techniques and frequency selective filtering. Digital filtering is motivated as a simple software simulation of analog filtering. The concepts of the discrete impulse response and discrete convolution are discussed. The authors point out that discrete convolution is similar to the inner product operation performed for net functions (Rumelhart et al., 1986) in PEs.

Next, the authors introduce the Adaline of Widrow. It is described as an adaptive linear combiner followed by a thresholding operation. The mean squared error at the output of the Adaline's linear portion is introduced as an objective function to be minimized. It is shown that there is a unique global minimum, when the autocorrelation matrix is invertible. Steepest descent is introduced as a practical means of finding the solution vector. The authors discuss how to bound the learning constant in steepest descent, including the problems that result when it is too large. However, they do not point out that there is a simple closed form solution for the learning constant, for the case where the objective function is quadratic

The authors point out that Adalines can implement transverse filters when the threshold activation is left off The application of such filters in echo cancellation is discussed. The Madaline, a network of interconnected Adalines, is introduced next. It is demonstrated that a Madaline can be configured to perform translation-invariant pattern recognition.

The authors discuss the simulations of Adalines and Madalines in the last subsections of Chapter 2. The Adaline simulation is discussed in some detail because many networks besides the Madaline resemble collections of Adalines. The data structures, output calculations, and implementations of learning are described Lastly, the authors mention the combinatorial growth of memory and math operations that results when the Madaline is simulated

Chapter 3: Back Propagation (BP)

The back-propagation network (BPN) is motivated using a character recognition example, in which network inputs are noisy binarized pixel intensities. A stated advantage of their approach is that parallel processing of pixel intensities is possible. A general discussion of training is given that includes error signals propagating backward through the network. The authors fail to mention the existence of conventional approaches for image preprocessing, feature calculation, and classification.

Generalized Delta Rule (GDR) training is described next. The authors erroneously state that each layer in the network has full connectivity only with the previous layer. This ignores the advantages of fully connected networks (in which units connect to all units in all previous layers), including fewer units and faster training. It also ignores the advantages of sparsely connected networks, which require fewer multiplications. The authors describe the concept of training data They point out that GDR is a generalization of the LMS rule and steepest descent.

The authors give formulas for the net and activation functions Only linear and sigmoidal activations are mentioned, ignoring the feasibility of using many other nonlinear functions. The authors erroneously state that the initial weights have no influence on the final solution.

The authors do a fairly good job of describing the weight update rule, error function, and gradient. They describe the full batching approach for accumulating weight changes and relate it to conventional steepest descent. However, they erroneously state that there is no advantage to it.

Update rules for hidden layer weights are given next. A summary of BPN calculations is provided for those who want to simulate the network The authors describe the training or testing error for individual patterns, but do not describe how to accumulate it over the entire training set, which is necessary to evaluate the procedure's success.

The authors then go into some practical considerations. They describe the danger that the network will forget if the training vectors are not properly ordered. Weight initialization using small random numbers is described. The authors give bounds on the learning factor that are too constrictive, but are acceptable for beginners. They compound their earlier error by stating that local minima of the error function are not a problem.

Two BPN applications are described First, a data compression network is described in which the hidden layer has fewer nodes than input or output layers. They propose this network as a method for image coding, in which 8×8 pixel blocks are compressed to 16 samples. However, the authors do not clarify whether the hidden units are linear and ignore the existence of conventional approaches to this problem that are very close to being optimal. Next, the authors describe a paint-quality inspection network that processes 30×30 pixel blocks. The authors ignore the possibility that the pixel blocks could be compressed conventionally before inputting them into the BPN.

Chapter 4: The BAM and the Hopfield Memory

The bidirectional associative memory (BAM) and Hopfield memory are described, and are motivated by examples of associative memory-like functions in humans and machines. These include the human ability to associate a face with a name and telephone number. Example devices include a memory bank that associates an address with data, and a content addressable memory that associates data with addresses of other data. The authors note that the Hopfield memory was very important for the resurgence of neural network research.

Hamming space and the Hamming distance are defined and related to Euclidean distance. Three types of linear associator are defined: heteroassociative, interpolative, autoassociative. There are four problems with the authors' presentation. First, they use the Hamming distance to measure the closeness of real vectors. Second, they should have put definitions before the beginning of subsection 4 1 2. Third, they say that no training is required for the linear associator. In fact, coefficients or weights are calculated from training vectors Fourth, the weight matrix is not defined until later on in the chapter

The BAM is discussed in an organized fashion. Its architecture is shown, along with a formula for its weight matrix, and the iterative nature of its processing. An example and three exercises are provided to clarify the BAM's operation There are some difficulties with the presentation. First, x and y have binary-valued elements, but this is mentioned in a figure rather than in the text. Second, no mathematical basis for the BAM is described The BAM energy function is presented, along with proof of a theorem concerning its minimization.

The authors derive the discrete and continuous Hopfield memories from the BAM, but do not discuss the relevance of the network's inputs. The continuous Hopfield memory is described in terms of RC circuits and the energy function. As an example, the authors discuss the application of the Hopfield memory to the Traveling Salesperson Problem (TSP). Their discussion is detailed, and is much clearer than Hopfield's own.

Simulation of the BAM is discussed. Included are descriptions of its data structures, initialization, and signal processing. There are a few difficulties. The authors refer to the weight matrix as appearing on page 136, whereas it is actually described on page 134. Second, they use C for the number of columns and for a constant used for in the Hopfield network Also, there is no discussion of simulation for Hopfield networks.

Chapter 5: Simulated Annealing

The authors introduce the idea of function minimization and contrast it with the function maximization performed in Chapters 2, 3, and 4. They point out the possibility of getting stuck in local minima if only function reduction steps are performed.

Information theory is briefly reviewed, with the ensemble average energy of a system defined in terms of the Boltzmann constant, Boltzmann factor, and Boltzmann distribution. Then the authors describe the physical process of annealing a silicon boule, so that its global energy is minimized, and local energy minima are avoided.

The authors give a fairly detailed description of the Boltzmann completion network. Its system energy, architecture, recall procedure, processing cycle, annealing schedule, and learning procedure are given, and its input–output network is briefly commented on. The Cauchy machine, with its shorter annealing schedule, is briefly mentioned.

Eighteen pages of text are spent in describing the simulation of Boltzmann machines. An example application is described, but not completed, in which the machine is trained to diagnose why a car will not start. It is not clear how well the Boltzmann machine works for this example, or how well it works compared to other networks described in the book, such as the BAM and Hopfield networks. Considering the authors' admission that training is extremely slow, the amount of text spent on this network seems excessive.

Chapter 6: The Counterpropagation Network

The authors introduce the counterpropagation network (CPN) as one that can learn forward and inverse mappings. They do not specify whether they plan to use the network for classification or mapping applications.

CPN building blocks including the input layer, the competitive instar layer, and outstars are described fairly well. Differential equations are given for signals in the input units, instars, and outstars. The operation and training of the forward CPN is described. The data normalization differs from that described earlier in the chapter, but no explanation is given.

The authors point out that the competitive and outstar layers can be trained separately. However, they but do not indicate that the training is similar to K-means clustering, and do not explicitly show that linear sets of equations can be used to find the outstar weights. It is not clear whether (I_i, y_i) pairs or (x_i, y_i) pairs are used for training.

In an example application of the CPN, the network is used to estimate the angle of rotation of the principal axis of the space shuttle, from an image. There is one hidden unit for each of 12 rotation angles. The fact that the rotation angle is severely quantized is used to motivate the interpolative mode CPN. The authors admit that several conventional methods exist for solving this problem, but demonstrate that the network can work even when the object is partially occluded, which is not true of many conventional approaches. One problem is that the authors refer to this as a classification example, rather than as an estimation example.

Chapter 7: Self-Organizing Maps

The authors start with comments on the cerebral cortex's structure, centers for speech, vision, hearing, thought, and their logical ordering according to functionality. The tonotopic map of the auditory regions and somatotopic map of motor nerves in the brain are used to motivate ordered feature maps. The purpose of this section is to "investigate a method by which these maps might develop naturally." This uses competitive units that are called Kohonen units, a term not used in the previous chapter.

The authors give differential equations for the activations and learning process. They point out that neighbors of a winning unit also learn, unlike in the CPN. However, they fail to relate the self-organizing map (SOM) to conventional algorithms such as K-means clustering. The classical example from Kohonen's book is included, in which the SOM learns a rectangular grid from random vectors. The feature map classifier, which is an SOM with an extra layer, is mentioned.

As a first example application of SOMs, the authors describe the Neural Phonetic Typewriter, which translates speech phonemes into written text and has an unlimited vocabulary. In a second example, the authors present an SOM that learns a tensor matrix for a two-joint robotic arm.

Chapter 8: Adaptive Resonance Theory

The authors begin by describing the stability–plasticity dilemma. The shortcomings of the BPN are mentioned, including its tendency to forget if trained only on inputs from a new class, and its failure to recognize the novelty of an input. They use the human ability to remember things learned in the past to motivate the need for adaptive resonance theory (ART) networks.

The description of ART 1, which has binary-valued inputs, begins with a brief summary of the pattern matching cycle, 2/3 rule, and gain control ideas. Then the F_1 and F_2 layers, orienting subsystem, bottom up and top down traces are described in detail. The differential equations for the attentional subsystem are given in terms of "activities" x_k, but it is not clear how these are related to inputs. The ART 1 processing steps are given and used in an example.

Next, the authors describe the differences between ART 1 and one of Carpenter and Grossberg's versions of ART 2. They show that the architecture is basically the same for both, although the F_1 layer of ART 1 splits into several sublayers in ART 2. They list the ART 2 processing steps and follow the steps in an example.

Chapter 9: Spatiotemporal Pattern Classification

This chapter deals with spatio-temporal networks (STNs). The STNs are networks that can recognize spatio-temporal patterns (STP), which are time-correlated sequences of patterns. This kind of recognition is important in speech, radar, and sonar echo recognition. A speech recognition application is used as an example throughout the chapter. The preprocessing step for the chapter's networks is a spectrum analyzer, whose output is a time-varying vector.

The authors introduce Grossberg's formal avalanche, which has a simple architecture consisting of multiple outstars. They use this network, for which they give differential equations, as a basis for this chapter's other networks.

The operation of a formal avalanche for single word recognition is described The notation seems confused. Both $P_1(t)$ and $Q_1(t)$ represent spectrum analyzer output. Perhaps one STP represents training data and the other testing data. The authors mention the idea of using averaged spectra from several speakers for tuning or training the network. They discuss the sequential nature of recognition and show that recognition fails if a word is input backwards.

The Sequential Competitive Avalanche Field (SCAF) is motivated by the need to reduce the number of redundant units in the formal avalanche A two-node SCAF example is given. They show how temporal order is introduced through the differential Hebbian learning law. The authors go on to discuss time-dilation effects and STN simulation.

Chapter 10: The Neocognitron

The authors describe how the Neocognitron was specifically designed for the recognition of handwritten characters, and is based upon the architecture of the visual cortex, as discovered by Hubel and Wiesel. They describe the hierarchical structure of the network, along with its connectivity. They explain how each level has two layers, one of simple or S-cells followed by a layer of complex or C-cells. They discuss the different tasks assigned to S-cells and C-cells. The authors give equations for S-cell processing in terms of net and activation functions, and show that the net has excitatory and inhibitory inputs. They point out how weight sharing in S layers simplifies training.

The C-cell net and activation functions are given. The authors discuss the use of lateral inhibition and feedback to handle ambiguous overlapped characters.

Summary (by the reviewer)

Freeman and Skapura have done a good job compiling detailed descriptions of the most commonly used ANS paradigms. They give pseudocode at the end of almost every chapter, as a guide to programming that chapter's paradigm. Processing summaries are used to describe the processing and learning algorithms in many chapters. Exercises are found throughout many of the chapters, along with programming exercises at the end.

The theory behind each paradigm is discussed in more detail than in other neural network books. Relevant differential equations for processing and training are given for many of the paradigms. The relationships between the various paradigms are made clear The authors' description of ART 1 and ART 2 is much more complete than those in other texts.

The authors have successfully accomplished the goals stated in the preface to provide a solid background in ANS technology to the reader, and to provide a book suitable for use in self-study and in the classroom. However, the book does have several shortcomings. First, it treats ANS in isolation, and ignores the existence of conventional algorithms that can sometimes perform as well as or better than the paradigms discussed. This is clear in the example applications, such as character recognition, speech recognition, rotation angle estimation, and image coding. Second, the authors fail to adequately motivate their use of differential equations to describe network operation and learning. The differential equations are not used in the simulations, and it is not clear whether their inclusion would speed up network operations. Perhaps the equations would be useful in analog VLSI implementations of the networks. Third, the approximation capabilities of the BPN, including its ability to learn Bayes discriminants, should have been included. In spite of these shortcomings, the book will be quite valuable to anyone who wants to gain an understanding of neural networks I highly recommend it

M T Manry
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX 76019

REFERENCES

Lippmann, R P (1987) An introduction to computing with neural nets *IEEE ASSP Magazine*, **April,** 4–22

Pao, Y-H (1989) *Adaptive pattern recognition and neural networks* Reading, MA· Addison-Wesley

Rumelhart, D E, Hinton, G E, & Williams, R J (1986) Learning internal representations by error propagation. In D. E Rumelhart & J L McClelland (Eds.), *Parallel distributed processing (Vol 1)* Cambridge, MA: MIT Press