Communications, Automation and Information Systems
M. Carnevale, M. Lucertini and S. Nicosia (Editors)
Elsevier Science Publishers B.V. (North-Holland)

NEURAL NETWORK MODELS AND APPLICATIONS: AN OVERVIEW

Léon PERSONNAZ and Gérard DREYFUS

Ecole Supérieure de Physique et de Chimie Industrielles
de la Ville de Paris,
Laboratoire d'Electronique
10, rue Vauquelin
75005 PARIS
FRANCE

This paper is an elementary introduction to networks of formal neurons; it endeavours to give a cursory presentation of the state of the art in basic research and applications. In a first part, we describe the usual models of formal neurons and the network architectures which are currently used: static (feedforward) nets and dynamical (feedback) nets. In the second part, we give an overview of the main potential applications of neural networks: pattern recognition (vision, speech), signal processing, automatic control. Finally, the main realizations (simulation software packages, special-purpose simulation machines, integrated circuits) are outlined.

1. INTRODUCTION

The idea of getting inspiration from the structure and operation of the nervous system in order to design machines is an old one; as soon as 1943, W. McCulloch and S. Pitts suggested a neural model which was intended to account for the ability of the brain to perform boolean functions[1]. This model is still widely used. Potential applications of these models to automatic classification and signal processing generated a high level of activity between 1960 and 1970; the realization of the limitations of the networks that were investigated at that time brought the whole field to a halt.
Completely new perspectives emerged recently, due to two significant contributions:
- the application of new concepts, stemming from the physics of disordered media, to the analysis of neural network architectures, alllowed to make quantitative predictions on the behaviour of these complex, nonlinear systems,
- the introduction of a learning algorithm which can be applied to multilayer neural networks allowed to overcome the limitations of the neural networks that were investigated in the 1960's.
These contributions are very different in nature: the first contribution is of a fundamental nature and has rich conceptual implications; the second one is purely algorithmic and may have practical applications in the future.
Researches on neural networks have been spurred by potential industrial applications, in economically important areas, such as pattern recognition, signal processing, automatic control. Such applications are usually very demanding in terms of computational speed, and are often well suited for parallel processing. At the present time, parallel computing machines range from parallel computers, made of powerful processors, to special-purpose cellular arrays (with nearest-neighbour connections for instance). Neural networks have specific features: they are made of very simple, strongly interconnected processors; their processing capabilities depend basically on the "strength" of the connections between the processors. In contradistinction to standard parallel computers, which are programmable, neural networks must be looked upon as objects whose properties and functions are determined by the connections between the neurons.
The field of neural networks is strongly interdisciplinary: it spans a spectrum including neurophysiology, neurobiology, psychology, statistical physics, applied mathematics, computer science, artificial intelligence, signal processing, automatic control, circuit design, optoelectronics, etc.

2. FROM LIVING NEURONS TO FORMAL NEURONS

The neuron is the basic building block of all nervous systems. It receives incoming electrical signals on numerous ramifications or dendrites. It transmits information through a fibre with many ramifications: the axon. Information transfer from cell to cell occurs through specific contact points called synapses. Chemical neurotransmitters are liberated in the synapse and alter the intracellular potential, thereby increasing or decreasing the probability that the neuron generate an electrical signal: if the intracellular potential is low enough, the neuron is quiescent. When the potential is large enough, the neuron becomes active. Thus, the neuron is basically a non-linear element which is continually making decisions, in response to incoming signals.
In living organisms, neurons are organized to form networks of various sizes, with rich interconnection patterns: they are basically highly parallel signal processing machines, which perform a large variety of tasks: processing of sensory inputs, motion control, memory, high-level cognitive tasks, etc. The decisions made by these networks are a result of the collective behaviour of the neurons. Some neural networks have a fixed structure resulting from evolution, whereas others are highly adaptive.
Most of the above characteristics are to be found in artificial "neural systems": relatively simple processors, high connectivity, parallel processing, cooperative behaviour, fixed or adaptive structure, wide functional variety. However, networks of formal neurons are extremely far from matching the data processing abilities of even the most rudimentary nervous system of the simplest animal.

3. MODELS OF FORMAL NEURONS

A formal neuron is a very simple automaton:
- it computes its *potential* v by performing a weighted sum of its inputs x_j; the weighting coefficients C_{ij} are termed *synaptic weights;*
- its output s (which is its state) is a non-linear function of the potential; it is obtained by applying a function f, called the transfer function of the neuron, to the potential: $s = f(v)$ (Figure 1).

The transfer function depends on the field of application and on the architecture of the network: it may be a step function (in which case the state of the neuron is binary), as illustrated on Figure 2, or a sigmoidal function (in which case the state takes on real, continuous values lying within two bounds), as illustrated on Figure 3.
Models of neurons are either static (time-independent), discrete-time (their behaviour can be described by a difference equation), or continuous-time (described by a differential equation); the state evaluation can be either deterministic or stochastic.

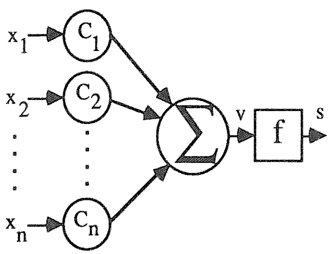

FIGURE 1
A formal neuron

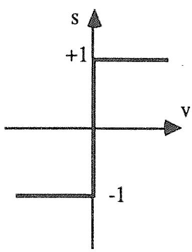

FIGURE 2
Transfer function
of a binary-state neuron

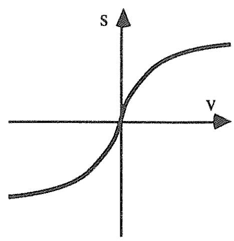

FIGURE 3
Transfer function
of a
continuous-state
neuron

Examples:

- *Deterministic, static, binary-state neuron:* the McCulloch-Pitts neuron:

$$v = \sum_j C_j x_j \ ;$$

$s = +1$ if $v>0$; $s = -1$ if $v \leq 0$.

- *Deterministic, static, continuous-state neuron:* the non-linearity is usually of the sigmoidal type (Figure 3); it is used when the learning algorithm requires the transfer function to be differentiable. Usually s lies between -1 and +1.

- *Deterministic, discrete-time (dynamical) neuron (binary or continuous-state):* the potential v is time-dependent, being a function of the past values of the input signals:

$$v(t) = \sum_j C_j \, x_j(t-\tau_j) \ , \qquad t, \tau_j = 0, 1, 2, \ldots\ldots$$

where τ_j is the delay in the transmission of signal x_j due to synapse j. The state s of the neuron is given by :

$s(t) = f[v(t)]$,

where f is the transfer function of the neuron.

4. NETWORK ARCHITECTURES.

There are basically two classes of network architectures: feedforward (static) networks and feedback (dynamical) networks.

4.1 Feedforward networks:

Feedforward networks are made of static neurons: the response of the network is obtained immediately after the application of the input signal (within propagation times). Neurons which are not output units are called "hidden" neurons. The most classical architecture of static networks is a layered structure as shown on Figure 4: each neuron of a given layer receives information from neurons of the previous layer only. This structure, however, is just a special case of a more general architecture shown on Figure 5. Such networks are used basically as classifiers, non-recursive filters and predictors.

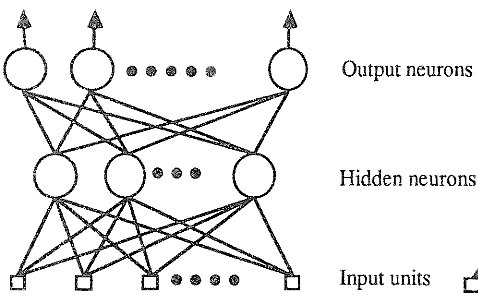

FIGURE 4
A layered feedforward network.

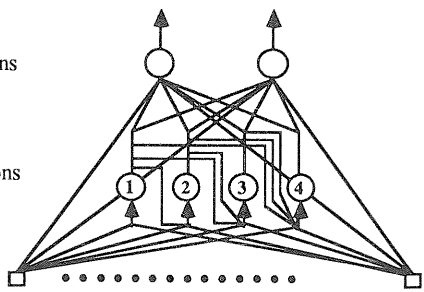

FIGURE 5
A fully connected feedforward network.

4.2 Feedback networks:

Feedback networks are made of a combination of dynamical and static neurons. In contradistinction to feedforward networks, these systems have dynamical properties which make them suitable for use in complex data processing tasks. In addition, they are more plausible than static systems from the viewpoint of biological modelling, since all nervous systems are clearly dynamical. A general dynamical network has input units, output neurons, and hidden neurons, the graph of the connections featuring cycles; static neurons may be present in the network, the only condition being that the subgraph of connections between static neurons must be acyclic. Figure 6 shows an example of such a structure.

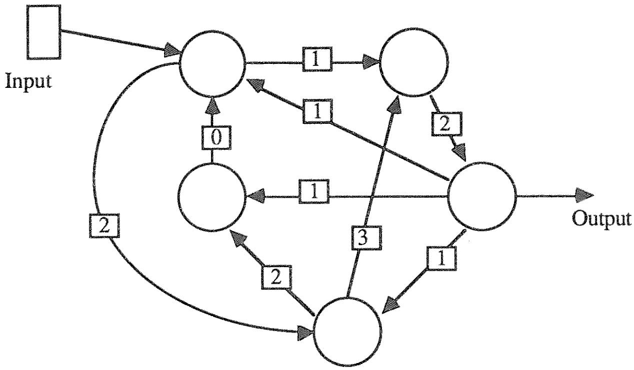

FIGURE 6
A feedback network.
Numbers indicate the values of the synaptic delays.

It can be shown that any dynamical neural network can be cast into a canonical form which consists of a static network and of unit delay loops feeding the state variables of the system back to the inputs. The static network computes the state of the system at time t as a function of the state at time t-1 and of the external inputs at time t-1. The order of the network is defined as the number of state variables. The outputs (which are assigned desired values during training) at time t are functions of the state variables at the same time t.

5. TRAINING AND USING NEURAL NETWORKS

Most neural networks currently under investigation are trainable networks: they are able to learn a task through a training procedure, whereby the synaptic weights are computed. The most frequent type of training is called "supervised learning": an "instructor" presents the network with pairs of data (the "training set") consisting of an input data and the corresponding desired response; for instance, the input data may be the picture of a handwritten numeral, and the desired output the binary code of the digit. The synaptic weights are computed so as to minimize a criterion which characterizes the quality of the response of the network to the set of data used for training; frequently, the criterion which is minimized is the squared sum of the differences between the actual and desired responses. This criterion is minimized through a gradient descent algorithm; a popular brand of gradient descent algorithm is the so-called "backpropagation" method. The speed of convergence is one of the problems encountered by that method, and many basic questions are still to be answered in order to make such a training procedure reasonably efficient.

Backpropagation can be used under various forms both for static and for dynamical networks; it was initially suggested in the framework of static networks[2], but it can be used for any dynamical network provided the latter is put in the above-mentioned canonical form.

Once learning has been performed, static networks can be used as classifiers, hetero- or auto-associative memories, predictors, and, more generally, to implement input-output mappings. The capabilities of the network are measured on a set of examples, distinct from the training set.

Dynamical neural networks can be used for performing a variety of tasks:
- Associative memories can be designed with dynamical neural networks by capitalizing on the fact that feedback networks are non-linear dynamical systems which exhibit *attractors*. In that context, learning consists in computing the synaptic coefficients in order to impose the stored information as the attractors of the system. The most popular attractor neural network is the *Hopfield model*, which is a single layer network with complete connectivity [3], made of binary dynamical neurons with unit delay for every synapse. These networks have been studied in depth with the tools of statistical mechanics [4-6], and many variants of these networks have been proposed [7,8]. One of the salient features of this class of networks is their *robustness* with respect to inaccuracies of the synaptic weights. Associative memories can also be designed with feedback networks featuring hidden units [9-11].
- Adaptive signal processing tasks and adaptive non-linear automatic control tasks can be performed by dynamical networks; in that case, learning consists in computing the synaptic weights in order to assign a given dynamical behaviour to the system in response to time-varying inputs [12]. Continually adaptive systems can be designed.

6. APPLICATIONS

The above general considerations will be illustrated by some typical applications which are currently being developed.

6.1 Automatic recognition of handwritten digits:

A recent sudy by AT&T Bell Labs uses several neural networks in order to perform the various tasks necessary for the automatic recognition of handwritten digits (postal codes).
Ad hoc (non trainable) networks, implemented in hybrid analog-digital VLSI technology, perform the skeletonizing of the numerals and the extraction of topological features; after these preprocessing tasks, data is transmitted to an adaptive network (still simulated), used as a classifier [13] (Figure 7). The input data consists of binary, normalized 16 x 16 pixels pictures; the preprocessing phase yields a vector featuring 180 components (topological features). Various types of classifiers, both neural and non-neural, have been used with a 1200 digits database, and their performances have been compared systematically [14]. With this database, neural networks exhibit performances which are close to those of standard classifiers, or slightly better.
The same group has used a multilayer neural network starting directly from the pixel representation of the digits.The network has an error rate of 1% with a rejection rate of 12% on a data base of several thousands of handwritten digits. The system uses a hardware accelerator based on a signal processing chip in order to implement the normalization task and the recognition task; this hardware allows a recognition rate of ten handwritten digits per second.

6.2 Speech recognition

During the past two years, the use of neural networks for speech recognition has attracted much attention [15]. Several investigations have been carried out by research groups having a previous experience in the field of speech recognition, with specical emphasis on preprocessing and on the use of Hidden Markov Chain models. Various kinds of tasks have been performed, with several types of neural networks.

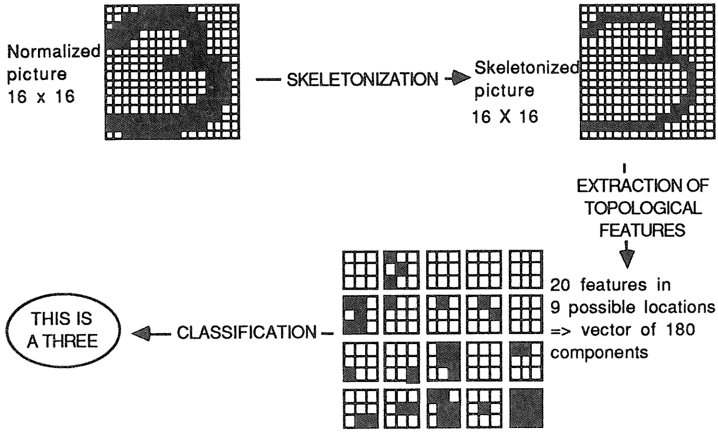

FIGURE 7
A recognition system for handwritten numerals.

The tasks that have been attacked exhibit various degrees of complexity: recognition of vowels, recognition of a few phonemes with their context (single and multiple speakers), recognition of ten isolated digits; to the best of our knowledge, the most complex task was the recognition of a few sentences. The networks used for these tasks have varying complexity, from single-layer feedforward networks to multi-layer feedback networks. Some authors suggest combinations of neural networks with standard techniques.
One of the most interesting study was aimed at the recognition of the consonants "B, D, G" of a single speaker, using a feedforward network with two layers of hidden units [16]. The input data is a spectrogram with 16 frequency values sampled every 10 ms with a total duration of 150 ms. The phonemes used for training (BA, DA ,GA, BO, DO, GO, etc.) are isolated from a spoken text. The neurons of the first layer are split into 8 independent groups. Ad hoc constraints imposed on the architecture and on the synaptic weights allow the system to extract 8 masks which are useful for classification. The second hidden layer is similarly organized. The results obtained with 1200 examples (600 for training and 600 for testing the network), for a single speaker, lead to an error rate of 1.5%. The authors compared their results to those obtained with a system using hidden Markov chains, which has an error rate of 6%.

6.3 Signal processing, automatic control.

Signal processing and automatic control have not been as extensively investigated as pattern recognition for instance. This is somewhat paradoxical since the early work by Widrow [17] dealt with filtering, noise cancelling, etc. However, the units that were used in these studies were linear. The main interest of the current preliminary investigations lie in the use of non-linear neurons.
Speech encoding is one of the signal processing problems that have been investigated recently. The problem is the following: how to code the speech signal, which is intrinsically redundant, on a number of bits which should be as small as possible, in order to make the best use of the limited channel capacity of the transmission lines, without audible distortion of the received signal? Multilayer feedback networks have been investigated for this purpose. Although preliminary results are encouraging, much is still to be done in order to make these systems operational and economical.

Source separation is another difficult problem which has been attacked with simple, ad hoc, analog networks [18]. Again the results are satisfactory, but efforts must be done in order to clarify the operation of such systems and measure the performances in a rigorous way.
Another notable exploratory work has been carried out in the field of signal prediction [19]. The authors used neural networks to predict the solution of non-linear differential equations having a chaotic behaviour. The use of multilayer networks yielded results of the same quality as those obtained by traditional methods.
The possible use of neural networks in robotics has also been investigated. In most cases, the investigations were aimed at computing the trajectory of a robot arm [20] and the determination of the inverse kinematics [21]. The adaptive and non-linear character of the neurons should make them suitable for adaptive automatic control of non-linear processes.

7. IMPLEMENTATIONS

Naturally enough, the first investigations of neural networks were performed by simulations on standard computers. Packaged simulators are available commercially. However, real-life applications, which often involve hundreds of neurons, require a large computing power; moreover, training algorithms tend to be exceedingly slow. Therefore, several attempts at building special-purpose machines are under way in order to speed up the neural network simulations. Various kinds of machines have been designed:

- special purpose machines built around commercially available arithmetic chips; all the so-called "neurocomputers" are just personal computers with a number-crunching add-on card;
- special purpose parallel machines with a suitable architecture for neural network simulations [22] (machines built around Transputers for instance);
- massively parallel microprogrammed machines[23];
- hybrid analog-digital machines, possibly using optoelectronic components[24].

Beyond these special-purpose machines, which are just sophisticated simulators, a large research activity is devoted to the design and manufacturing of silicon chips actually implementing "artificial neurons". Numerous circuits have been designed or built [25]. Among those, a particularly interesting class of artificial neural networks is emerging: networks which are intended to mimick the operation of the retina or of the cochlea [26]. These circuits seem particularly well adapted as preprocessing chips for fully "neural" systems.

8. CONCLUSION

Beyond the current overexcitement about this field, neural networks are being studied very seriously in a number of places, both in Universities and in industry. The field has conceptual foundations which are much sounder than they were twenty years ago, when the first wave of interest for neural networks came and died. A few solid results about the limitations of the capabilities of neural networks have been derived, and many more studies should be performed along those lines. These new developments are likely to make real-life applications possible in the next years; many efforts are aimed at comparing seriously the performances of neural networks to those of classical methods; many prototypes are operating satisfactorily in laboratories. There is a long way, however, from prototypes to actual, significant, economically valid, real-life applications. Indeed, there are no serious commercial applications of neural networks at present. Huge research and development efforts have to be performed before neural networks become a real technology.

REFERENCES

[1] McCulloch, W. S., and Pitts, W. H., Bull. Math. Biophys. 5 (1943) 115.

[2] Rumelhart, D. E., Hinton, G. E., and Williams R. J., in Parallel Distributed Processing (M.I.T. Press, 1986), vol. 1.
[3] Hopfield, J. J., Proc. Natl. Acad. Sci. USA 79 (1982) 2554.
[4] Amit, D., Gutfreund, H., and Sompolinsky, H., Phys. Rev. A 32 (1985) 1007.
[5] Gardner, E., Europhys. Lett. 4 (1987) 481.
[6] Peretto, P., Biol. Cybern. 50 (1984) 51.
[7] Personnaz, L., Guyon, I., and Dreyfus, G., Phys. Rev. A 34 (1986) 4217.
[8] Krauth, W. and Mézard, M., J. Phys. A 20 (1987) L745.
[9] Almeida, L. B., 1987, Proc. of the First IEEE International conference on Neural Networks, (San Diego, 1987).
[10] Pineda, F., Phys. Rev. Lett. 59 (1987) 2229.
[11] Rohwer, R. and Renals, S., 1989, in Personnaz, L., Dreyfus, G., eds., Neural Networks from Models to Applications, (I.D.S.E.T., Paris, 1989), p. 207.
[12] Robinson, A. J. and Fallside, F., in Anderson, D.Z. , ed., Neural Information Processing Systems (American Institute of Physics, 1988).
[13] Graf, H. P., and Hubbard, W., in Personnaz, L., Dreyfus, G., eds., Neural Networks from Models to Applications, (I.D.S.E.T., Paris, 1989), p. 725.Farhat N.H., Psaltis D., Prata A. and Paek E.G., 1985, Appl. Optics 24, p. 1469.
[14] Guyon, I., Poujaud, I., Personnaz, L., Dreyfus, G., Denker, J. and Le Cun, Y., in Proceedings of the IEEE International Joint Conference on Neural Networks (Washington, 1989).
[15] Lippmann, R. P., Neural Computation 1 (1989) 1.
[16] Waibel, A., Neural Computation 1 (1989) 39.
[17] Widrow, B. and Stearns, S.D., Adaptive Signal Processing (Prentice-Hall, 1985).
[18] Hérault, J., Jutten ,C. and Guérin, A., in Personnaz, L., Dreyfus, G., eds., Neural Networks from Models to Applications, (I.D.S.E.T., Paris, 1989), p. 493.
[19] Lapedes, A. and Farber, R., in Anderson D.Z. , ed., Neural Information Processing Systems (American Institute of Physics, 1988).
[20] Ritter, H., Martinetz, T., and Schulten, K., 1989, in Personnaz, L., Dreyfus, G., eds., Neural Networks from Models to Applications, (I.D.S.E.T., Paris, 1989), p. 579.
[21] Kawato, M., Uno, Y., Isobe, M. and Suzuki, R., IEEE Control Systems Magazine, April 1988, p. 8.
[22] Petrowski, A., Personnaz, L., Dreyfus, G. and Giraud, C., in André, F., Verjus, J.P., eds, Hypercube and Distributed Computers (North Holland, 1989)
[23] Guérin, A., Jutten, C. and Hérault, J., 1989, in Personnaz L., and Dreyfus G., eds., Neural Networks from Models to Applications, (I.D.S.E.T., Paris, 1989), p. 756.
[24] Farhat, N., IEEE Circuits and Devices Magazine 5 (1989) 32.
[25] Personnaz, L., Johannet, A. and Dreyfus, G., 1989, in Pfeifer, R., Schreter, Z., Fogelman-Soulie, F., Steels, L., eds, Connectionism in Perspective (Springer, 1989).
[26] Mead C.A., Mahowald, M.A, Neural Networks 1 (1988) 91.