Artificial neural networks and anticipatory systems

Jean-Philippe Draye

Senior Research Assistant of the Belgian National Fund for Scientific Research (F.N.R.S.)

Faculté Polytechnique de Mons
"Parallel Information Processing" Laboratory
Rue de Houdain, 9
7000 Mons (Belgium)
E-mail : jpd@pip.fpms.ac.be
Homepage : http://pip.fpms.ac.be/~jpd

Keywords : artificial neural networks, time series prediction, system identification, anticipatory systems, control.

Abstract : Neural networks are structures consisting of highly interconnected elementary computational units. The network map input patterns into output patterns. These devices are called neural not because they model the nervous systems but because they are *inspired* by them. Due to their features (adaptive behaviour, learning process, robustness, ...) , artificial neural networks have interested the anticipatory systems field of research. In particular, we will examine their interests in the field of time series prediction and system identification.

1 Introduction

Human beings are able to perform some tasks effortlessly ... nevertheless, after several decades of research, computer scientists have only made small progress toward building a device that can handled them. Among those tasks, we can cite : reading a text aloud, recognizing a face, responding to spoken commands, translating English into French, driving a car, classifying flowers, ...

However, by the fifties, shortly after the first number-calculating computers were first introduced, visionaries such as John Von Neumann and Alan Turing were speculating about the construction of machines that might exhibit *intelligent behaviour*. Turing proposed two possible lines of investigations : one line starts from the study of biological systems (the brain), searching the lowest cognitive structure and trying to mimic them. The other line starts from a study of efficient problem-solving and searching for rules by which *expert systems* approach their tasks and, finally, building machines that would apply these rules. Turing preferred this second path and many, in the field of artificial intelligence, followed his lead. On the contrary, Von Neumann was fascinated by electronic circuits whose structure was inspired by the nervous systems of animals. Unfortunately, he died before he could have followed this path. The field of *neural computation* has been built

by those who followed the Von Neumann path.

Neural networks are structures consisting of highly interconnected elementary computational units. The network map input patterns into output patterns. These devices are called neural not because they model the nervous systems but because they are *inspired* by them.

Neural networks perform a great variety of pattern mappings. A network can reconstruct a stored pattern when the input is a partial match for that pattern : the network acts as an associative memory (such as the Hopfield network). A network can retrieve a second pattern associated with a given input pattern (for example, the classical task of handwritten characters recognition). Following new developments, a neural network can even proposed a desired *temporal* signal in response to temporal input signals. It can generate a pattern representing the solution of a combinatorial problem. It may be asked to group similar patterns into clusters and provide new representative of clusters.

The best-known neural network model is, by far, the *multilayer perceptrons*. The simplest perceptrons, proposed by Frank Rosenblatt in the mid-50's, have all their units arranged in a single layer. Several authors have demonstrated that additional layers to this simple model can greatly increase its power. The interest for this model emerged in 1986 after the publication of the *backward-error-propagation algorithm* by David E. Rumelhart and his colleagues. Since them, many successful cases in which learning based on error-backpropagation has produced excellent results have been reported. The applications include recognition, process control, robot kinematics, medical diagnosis, etc.

Since then, new neural types have emerged that give impressive results in the field of auto-organizing networks (like the Kohonen network), nonlinear dynamical systems identification or control (using fully-connected recurrent neural networks) or associative memories.

We must emphasize that the power of neural networks does not arise solely from the high degree of parallelism in the computation of output patterns. The power of neural models derives from the high number of interconnections among elements and from the network's ability to alter its internal structure (or weights) via a learning algorithm.

To summarize, we can say that artificial neural network models can be superior than other methods under these conditions :

- **The data on which conclusions are to be based is fuzzy**. If the input data is human opinions, ill-defined or is subject to possible large error, the robust bevaviour of artificial neural networks is impressive.
- **The patterns important to the required decision are subtle or deeply hidden**. One of the principal advantages of an neural network is its ability to discover patterns in data which are so obscure as to be imperceptible to human researchers and standard statistical methods.
- **The data exhibits significant unpredictable nonlinearity**. Traditional time-

series prediction models are based on strictly defined models. If the data does not fit these models, results will be useless. Artificial neural networks are ideal for these cases.

- **The data is chaotic in the mathematical sense**. Chaos can be found in telephone line noise, stock market prices and other physical processes. Such behaviour is devastating to most other techniques, but neural models generally work quite well.

It is thus quite obvious that, due to their features (adaptive behaviour, learning process, robustness, ...) and due to the previous considerations, artificial neural networks have interested the anticipatory systems field of research.

In particular, artificial neural networks are very interesting to simulate externalistic anticipatory systems. In many areas of scientific research, the problem of predicting the future of dynamical systems arises. Closely related problems have also interested the scientists : process control and system identification. Unfortunately, when the observed dynamics are nonlinear with a complex dependence on time, the formulation of reliable predictions (or control or identification) becomes extremely difficult. These problems have been studied as a problem of multidimensional function approximation. This approach has produced new methodologies for the analysis of nonlinear systems or time series (including local procedures). Neural network architectures have drawn considerable attention in recent years because of their interesting learning abilities. Moreover, they are capable of dealing with the problem of structural instability. Many studies have reported exceptional results using neural networks.

2 Foundation of the artificial neural network theory

An artificial network consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections. Rulmelhart [8] enumerates the major aspects of a parallel distributed processing model as

- set of processing units ("neurons", "cells");
- a state of activation y_i for every unit, which also determines the output of the unit;
- connection between the units. Each connection is defined by a weight w_{ij} which determines the effect of the signal of unit i on unit j.
- a propagation rule (a governing equation), which determines the effective input u_i of a unit from its external inputs;
- an activation function F which determines the new level of activation based on the effective input $u_i(t)$ and the current state $y_i(t)$;

- an external input or offset θ_i for each unit;
- a method for information gathering;
- an environment within which the system must operate, providing input signals and –if necessary– error signals.

In this simple model, the simplest node sums a collection of weighted inputs and passes the result through a nonlinearity. The processing unit is therefore characterized by an internal threshold or offset θ and by the kind of nonlinearity (e.g., hard limiters, threshold logic, sigmoidal nonlinearities, ...).

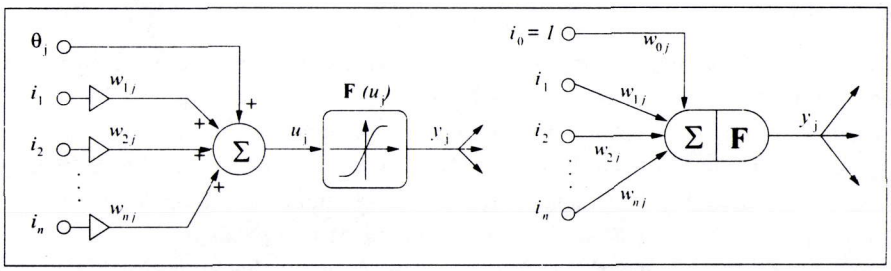

Figure 1: Simplified functional model of an artificial basic neuron cell and its symbol

Figure 1 shows a very simple processing unit using a sigmoid nonlinearity. The nonlinearity here simply seeks to classify the weighted sum of the n inputs as being greater than or less than zero and accordingly assigns the input vector $(i_1, i_2, ..., i_n)$ a continuous value between -1 or $+1$. The relative magnitudes of the weights w_{1j}, w_{2j}, ... w_{nj} decide the relative importance of the inputs i_1, i_2, ... i_n in determining the final output. The weights have the effect of determining a hyperplane in the n-dimensional hyperspace such that all points on one side of the hyperplane are distinguished, by the function output, from all those on the other side. The offset θ has the effect to translate the hyperplane.

The neuron will compute its output by two functions. It will first compute the weighted sum of its preceding inputs. This weighted sum is called the *effective input* and is given by :

$$x_i(t) \;=\; \sum_{i=1}^{n} w_{ji}\; y_i(t) \;+\; \theta_i \tag{1}$$

The current output of the neuron must be determined using the effective input and an activation function :

$$y_i(t) \;=\; F\big(x_i(t)\big) \tag{2}$$

where $F(\cdot)$ is the *activation function.*

Let us point out here that we can rewrite the governing equation (1) in order to simplify the mathematical notation. We will add a virtual input neuron (labeled 0) whose value

remains constant and equals to one. The individual offsets θ_i are replaced by adaptive weights w_{0i} associated to the virtual interconnections between the virtual neuron 0 and all the network neurons (see Figure 1).

2.1 Importance of the activation function

In neuron-like units, the conceptual models exhibit a linear term (a weighted sum of the inputs) followed by a nonlinear function. This *activation function* $F(\cdot)$ determines the new level of activation based on the effective input $u_i(t)$ and the current state $y_i(t)$. The essential characteristics of these activation functions are the fact that they introduce some nonlinearity in the conceptual model of the neuron-like unit.

Without this nonlinearity, the neural models would reduce to pure linear systems. The linear systems theory is a powerful branch of systems theory and a number of applications do indeed exploit the methods of linear algebra and linear systems. However, with fixed input, a linear device has only a single equilibrium point state whereas a nonlinear system may, depending on its structure, exhibit multiple equilibrium states, limit cycles or even chaotic behaviour. These particular behaviours are interesting : indeed, artificial neural networks are useful alternatives when traditional techniques (and among them linear models) fail to success. It is thus these behaviours (multiple equilibrium states, limit cycles or even chaotic behaviour) whose are exploited by many neural networks applications.

Most common activation functions are the sigmoid (S-shaped) function. Such a sigmoid function can mathematically be described (approximated) for example as :

$$y_j = \tanh(\gamma u_j) = \frac{1 - \exp^{-2\gamma u_j}}{1 + \exp^{-2\gamma u_j}} \tag{3}$$

for a symmetrical (bipolar) representation or :

$$y_j = \frac{1}{1 + \exp^{-2\gamma u_j}} \tag{4}$$

for a non-symmetric unipolar representation.

Another important aspect of the activation function is the fact that this one should ideally be continuous. We have thus to avoid using the simple step function (the Heaviside function). There are two main reasons for this :

- the first one is the noise resistance : a step function can amplify noise which a sigmoid function may smooth out. Nevertheless, this may be at the price of postponing a binary decision until after further statistical analysis has been made.
- the second one is related to the training methods which exploits methods of the differential calculus to adjust the free parameters of the model in order to solve the task to handle. However, most widely used training algorithms (such as the well-known backpropagation algorithm) make use of the fact that that the activation functions are continuous and thus differentiable.

2.2 Neural networks architectures

Obviously, a neural network is a network of neurons. This high-level definition applies to both biological neural networks and artificial neural networks (ANNs). This section is mainly concerned with the various ways in which neurons can be interconnected to form the networks or network topologies used in ANNs, even though some underlying principles are also applicable to their biological counterparts. The term *neural network* is therefore used to stand for *artificial neural network* in the remainder of this work, unless explicitly stated otherwise. The nodes of an artificial neural network must be called artificial neuron or neuron-like unit but given the length of this term and the need to frequently use it, it is not surprising that its abbreviated form, *neuron*, is often used as a substitute instead.

The *architecture* or *topology* of a neural network is related to the way in which the neurons are organized to form the network. The network architecture is thus the relationship of the neurons by means of their connections. The network topology is characterized by the number of neurons that provide the information (also called the *input neurons*), the number of neurons that produce the results of the network computation (*the output neurons*), the number of neurons whose inputs and outputs remain within the network (these neurons that have no interaction with the "outside world" are called *hidden neurons*) and the way these neurons are connected to each other (the number of connection weights). The topology of a neural network thus consists of its framework of neurons, together with its interconnection structure or *connectivity*.

We can divide neural frameworks into two categories :

- feedforward networks where the information flows from the input neurons to the output neurons without any feedback connections. These neurons of these models are usually organized in layers.
- recurrent or feedbacks networks are characterized by cycles (associated to adaptive connection weights) in its graph. We can find in these networks feedforward, feedback or self-connections.

2.3 Neural network learning algorithms

The essential difference between neural network models and classical algorithms is the fact that they do not learn by being programmed; they learn by being trained. The *learning phase* of a neural network consists in a fine tuning of its internal parameters (classical its interconnection weights) in response to external stimuli. This parameters adjustment is coherent with biological neural networks where both memory and the formation of thoughts involve *neuronal synaptic changes*; and we have seen that the artificial neural networks models the synaptic strength using scalar weights. When the learning process is successful, all the network weights values stabilized.

The learning phase can be *supervised* or *unsupervised* :

- when the process is *supervised*, the external stimuli (which form the network input data) and the corresponding desired target data are provided. The learning process adjusts in some fashion the internal parameters so that for a given input, the network will produce the desired associated target. The supervised learning algorithm will compare the network output for a given input with the corresponding target data and compute the error difference between both data. This error difference provides the external feedback (determined using the target data) that is necessary to adjust the network variables.

- in *unsupervised learning*, the network adjusts itself its parameters by using the *inputs only*. No target data are available and it is impossible to determine errors upon which to base external feedback for learning. The *unsupervised* learning method groups similar sets of inputs into clusters predicated upon a predetermined set of criteria relating the components of the data. The *unsupervised* learning algorithm and their related networks are out of the scope of this work but readers can find more information in the works of Teuvo Kohonen, or ....

The *training phase* of a neural network refers to the presentation of the inputs (and possibly the targets if the learning process is supervised) to the network. The goal of the training and hence learning is to enable the network to exhibit good *generalization* capabilities. *Generalization* is the ability of the network to produce reasonable outputs associated with new inputs (that were present in the training sets of input data).

3 Neural networks and anticipatory systems

In the context of externalistic anticipation, i.e. time series prediction, a lot of scientific research have already proved the great interest of using artificial neural networks. Another field of numerous applications is system identification and process control.

Moreover, when using properly these applications, we can even get features that bring new insights in the field of *real* artificial intelligence : anticipation. For example, neural models were successfully used in preventing babies' central apneas (which is an important cause of death). In this particular application, the system *anticipates* dramatic accidents by a thorough analysis of current and past data.

We consider in the following the particular cases of time series prediction and system identification.

3.1 Time series prediction

3.1.1 Classical prediction methods

Classical prediction methods are usually linear; indeed, these ones have two particularly desirable features : they can be understood in great detail and they are straightforward to implement. The obvious penalty for this convenience is the fact that they may be entirely inappropriate for even moderately complicated systems.

The most known models are usually based on linear combination of previous observed data, on previous prediction errors or on both types of data. These models are respectively called *autoregressive process* (AR), *moving average process* (MA) or *mixed autoregressive-moving average process*. They have been introduced by Box and Jenkins in 1976 [2].

The ARMA class of stochastic models generalizes and includes the AR and MA models. It depends upon two values the AR order p and the MA order$q$, the process can be of p or q degree. Thus

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \cdots + \beta_p x_{t-p} + \epsilon_t - \psi_1 \epsilon_{t-1} - \psi_2 \epsilon_{t-2} - \cdots - \psi_q \epsilon_{t-q} \quad (5)$$

where $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$ are the observed time series, ϵ_t, ϵ_{t-1}, ..., ϵ_{t-q} are zero-mean forecasting errors. The model is denoted ARMA(p,q) [2].

Unfortunately, the disadvantages of the Box-Jenkins method are multiple : identifying the order of the model to be used can be difficult, especially for multivariate time series; the model identification and validation process of the Box-Jenkins process need expert knowledge; it is not easy to understand the statistical mechanism used in the model identification procedure and the method is primarily applicable for short term forecasting, so it is inaccurate for long term predictions (see [1]).

New models are thus necessary : neural networks can help in many circumstances [6, 9, 10].

3.1.2 Neural prediction methods

In the Sante Fe Time Series Prediction Competition (see [11]), the majority of contributions and also the best predictions for each proposed set of data used connectionist methods.

Nevertheless, one can wonder what differs a neural network from a polynomial regression for example. Given that feedforward networks with hidden units (see Section 2.2) implement a nonlinear regression of the output onto the inputs, what features do they have that might give them an advantage over more traditional methods (i.e. a polynomial regression). For the polynomial regression, the components of the input vector (x_1, x_2, ..., x_d) (where d is the dimension of the input vector) can be combined in pairs (x_1x_2, x_1x_3, ...), in triple ($x_1x_2x_3$, $x_1x_2x_4$, ...), etc, as well as combinations of higher powers. This vast number of possible terms can approximate any desired output surface. The difference between the neural regression and the polynomial one is the kinds of constraints

they impose. For the polynomial case, the number of possible terms grows rapidly with the input dimension, making it sometimes impossible to use even all of the second-order terms. Thus, the necessary selection of which terms to include implies a decision to permit only specific pairwise or perhaps three-way interactions between components of the input vector. A feedforward network, rather than limiting the order of the interactions, limit only the total *number* of interactions and learns to select an appropriate combination of inputs [11].

Recently, recurrent neural networks have even exhibited more powerful results (see [4, 5]).

3.2 Identification of systems

System identification is a fundamental problem in systems theory; it is connected to the problem of system characterization. The problem of characterization is concerned with the mathematical representation of a system as an operator $\mathcal{P}$ which maps input signals into output signals. The problem of identification is to approximate $\mathcal{P}$ in some sense using an identification model. By identifying a particular system, obviously, we will be able to *anticipate* its behaviour [7].

Specifically, the application of neural networks to the task of identification of systems is supported by their learning capacity, their ability to internally represent related states, and their good behaviour to approximate continuous functions. Particularly, in the last years, several works concerned with approximation of continuous functions defined on a compact set (in a space of finite dimensions) have been presented in the neural community.

Furthermore, in practice, there are situations where we need to compute functionals such as the output of a dynamical system (at any particular time, it can be viewed as a functional). At this level, Chen demonstrated that continuous functionals can be approximated by neural networks and showed their application for dynamical systems modeling [3].

4 Conclusion

Artificial neural networks are often considered as mysterious or *black box* systems without being more interesting than other existing techniques. We have seen in this paper that they can bring new insights in the field of anticipatory systems, especially for time series prediction, system identification or process control.

An important challenge would be to extent the theory of neural networks in order that these neural models become *autonomous* : that would be the real artificial intelligence.

Acknowledgments

This work is supported by the Belgian National Fund for Scientific Research and by the EC through the ESPRIT program NeuroCOLT (Working Group Nr 8556).

References

[1] A. Andersen, S. Makridakis, and R. Carbone. *The forecasting accuracy of major time series methods.* John Wiley and Sons, 1994.

[2] G.E.P. Box and G.M. Jenkins. *Time series analysis : forecasting and control.* Holden-Day Inc., San Francisco, 1976.

[3] T. Chen and H. Chen. Approximation of continuous functionals by neural networks with application to dynamical systems. *IEEE Transactions on Neural Networks*, 4(6):910–918, 1993.

[4] J.P. Draye, D. Pavisic, G. Cheron, and G. Libert. Adaptive time constants improve the prediction capability of recurrent neural networks. *Neural Processing Letters*, 2(3):12–16, 1995.

[5] J.P. Draye, D. Pavisic, G. Cheron, and G. Libert. Dynamic recurrent neural networks : a dynamical analysis. *IEEE Transactions on Systems, Man, and Cybernetics– Part B: Cybernetics*, 26(5):692–706, 1996.

[6] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks. Technical Report LA-UR87-2662, Los Alamos National Laboratory, 1987.

[7] K.S. Narendra. Identification and control. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 477–481. MIT Press, 1995.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing : Explorations of the Microstructure of Cognition*, volume I. Cambridge MA, Bradford Books, 1986.

[9] R. Sharda and R. Patil. Neural networks as forecasting experts. *IEEE Transactions on Neural Networks*, 1990.

[10] Z. Tang. Feedforward neural networks as models for time series forecasting. Technical Report TR91-008, University of Florida, 1991.

[11] A.S. Weigend and N.A. Gershenfeld. *Time series prediction : Forecasting the Future and Understanding the Past.* Addison-Wesley Publishing Company, Reading (MA, USA), 1994.