Artificial Neural Networks in Control and Optimization

A thesis submitted to
the University of Manchester
for the degree of
Doctor of Philosophy
in the Faculty of Technology

by

Cairo L. Nascimento Jr., B.Sc., M.Sc.

Control Systems Centre
UMIST
PO Box 88
Manchester M60 1QD

February 1994

To Sandra

for her support and personal sacrifices

Abstract

This thesis concerns the application of artificial neural networks to solve optimization and dynamical control problems.

A general framework for artificial neural networks models is introduced first. Then the main feedforward and feedback models are presented. The IAC (Interactive Activation and Competition) feedback network is analysed in detail. It is shown that the IAC network, like the Hopfield network, can be used to solve quadratic optimization problems.

A method that speeds up the training of feedforward artificial neural networks by constraining the location of the decision surfaces defined by the weights arriving at the hidden units is developed.

The problem of training artificial neural networks to be fault tolerant to loss of hidden units is mathematically analysed. It is shown that by considering the network fault tolerance the above problem is regularized, that is the number of local minima is reduced. It is also shown that in some cases there is a unique set of weights that minimizes a cost function. The BPS algorithm, a network training algorithm that switches the hidden units on and off, is developed and it is shown that its use results in fault tolerant neural networks.

A novel non-standard artificial neural network model is then proposed to solve the extremum control problem for static systems that have an asymmetric performance index. An algorithm to train such a network is developed and it is shown that the proposed network structure can also be applied to the multi-input case.

A control structure that integrates feedback control and a feedforward artificial neural network to perform nonlinear control is proposed. It is shown that such a structure performs closed-loop identification of the inverse dynamical system. The technique of adapting the gains of the feedback controller during training is then introduced. Finally it is shown that the BPS algorithm can also be used in this case to increase the fault tolerance of the neural controller in relation to loss of hidden units.

Computer simulations are used throughout to illustrate the results.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Acknowledgements

I am most grateful to my supervisor Dr. Martin Zarrop for all his patience, guidance and enthusiasm. It was an honour for me to work under his exemplary supervision.

I am also very grateful to Prof. Peter Wellstead, Prof. Neil Munro and Dr. Allan Muir for their support and encouragement. I also would like to acknowledge Dr. Daniel W. McMichael for his contributions in the early stages of this research.

I am indebted to the Aeronautics Technological Institute (Instituto Tecnológico de Aeronáutica) and to the Brazilian Research Council (CNPq - grant 200.617/88.5) for the support provided during this research.

Many thanks are due to Dr. Evandro T. de Souza, Dr. Takashi Yoneyama and Dr. J. A. M. Felippe de Souza for their long distance moral support and constant encouragement.

Finally, I wish to thank all the staff and students at the Control Systems Centre, especially Suchin Arunsawatwong, Chris Berg, Alex Bozin, Nongji Chen, Kevin Czachur, Tim Fitzpatrick, Celso Gonzalez, Peter Green, William Heath, Tariq Hussein, Duncan Hutchinson, Arne Jenssen, Tracy Jones, Alan Kjaer, Paul Lane, Panos Liatsis, Andy McCorkle, Roy Moody, Njal Pettit, Janice Prunty and Susan Simpson for all their help, moral support and friendship.

"You are never given a wish without also
being given the power to make it true.
You may have to work for it, however."
Richard Bach

Contents

List of Figures

List of Tables

Notation and Abbreviations:

Vectors and matrices are represented in bold italics, e.g. $\boldsymbol{X}$. Scalars are represented in italics, e.g. X_1. Common symbols and abbreviations are given below with the section number where they are introduced.

Symbol		Section
a	activation value of a processing unit	2.3.2
ACE	Adaptive Critic Element	7.2.2
ADALINE	Adaptive Linear Neuron	2.4.3
AI	Artificial Intelligence	2
ANN	Artificial Neural Network	2
ASE	Associative Search Element	7.2.2
AP	Action Potential	2.1.1
BP	Back-Propagation	2.4.5
BPS	Back-Propagation with Switching	5.3
BPTT	Back-Propagation Through Time	7.2.3
c	factor of cooperation or competition	3.3.4
CNS	Central Nervous System	2.1
$\boldsymbol{D}$	network desired output vector	2.4.1
E	squared network output error	2.4.3
ext	external input	3.1
EP	Equilibrium Point	3.1
$f(\circ)$	activation function of a processing unit	3.2
FB	Feedback	2.3.4
FF	Feedforward	2.3.4
FM	Frequency Modulation	2.1.2
$g(\circ)$	output function of a processing unit	3.3.1
H	energy function to be minimized (for feedback networks)	3.1.4
IAC	Interactive Activation and Competition	3.3
J	cost function	5.3

J^*	modified cost function	5.3
L	length of the tapped delay line	7.3.2
LAM	Linear Associative Matrix	2.4.1
LMS	Least-Mean-Square	2.4.3
LTM	Long-Term Memory	2.3.2
M	delay applied to the reference signal	7.3.2
MADALINE	Multiple Adaptive Linear Neuron	2.4.3
MIMD	Multiple Instruction Multiple Data	2.3.1
MLP	Multi-Layer Perceptron	2.4.2
NC	number of possible configurations	5.1
NH	number of hidden units	6.3
net	net input value of a processing unit	2.3.2
OOP	Object-Oriented Programming	2
out	output value of a processing unit	2.3.2
PE	Processing Element	2.3.1
PDP	Parallel Distributed Processing	2
PID	Proportional-Integral-Derivative	7.3
PNS	Peripheral Nervous System	2.1
$\boldsymbol{q}_k$	joint angle vector at time instant k	7.3.1
$\boldsymbol{R}$	correlation matrix	5.5.1
RBF	Radial Basis Functions	2.5.1
ref_k	reference signal at time instant k	7.2.1
$\boldsymbol{ref}_k^*$	neural network controller input at time instant k	7.3.3
RLS	Recursive Least Square	6.1
RMS	Root-Mean-Square	5.4
$\boldsymbol{S}$	pattern to be stored in a feedback network	3.1
STM	Short-Term Memory	2.3.2
t	time	2.3.2
$\boldsymbol{T}$	target network output vector	5.3
$\boldsymbol{T}_k^F$	feedback torque at time instant k	7.3.1
$\boldsymbol{T}_k^N$	feedforward torque generated by the network at time instant k	7.3.1
TD	Temporal Differences	7.2.2
TLU	Threshold Logic Unit	2.4.1

TMR	Triple Modular Redundancy	5.1
u_k	plant input at time instant *k*	7.2.3
u_k^{FB}	output of the feedback controller at time instant *k*	7.3.2
u_k^{NN}	output of the neural network controller at time instant *k*	7.3.2
$\boldsymbol{W}$	weight matrix	2.3.2
$\boldsymbol{X}$	network input vector	2.4.1
$\boldsymbol{x}_k$	plant state vector at time instant *k*	7.2.3
x_0	optimum input value	6.1
y_k	plant output at time instant *k*	7.2.4
$\boldsymbol{Y}$	network output vector	2.4.1

Greek Letters

η	learning rate	2.4.1
α	momentum rate	2.4.6
α^*	network coefficients	7.3.3
λ^i	probability of fault for configuration *i*	5.3
ν	coefficient of asymmetry	6.3
τ	torque	7.3.6

Chapter 1 - Introduction

1.1 - Background and Motivation

This thesis concerns the application of artificial neural networks to solve optimization and dynamical control problems. Artificial neural networks are computational devices whose conception has been motivated by our current knowledge of biological nervous systems. As such, neurocomputing, that is computation using artificial neural networks, offers an alternative to the traditional computational approach based on sequential and algorithmic processing.

Probably the main feature that characterizes the artificial neural networks approach is the simultaneous use of a large number of relatively simple processors, instead of using few very powerful central processors, as is nowadays the standard in most man-made computers. This is also the computational architecture selected by natural evolution for the central nervous systems of the most developed animals, where the basic computational unit is the neuron.

The use of a large number of simple processors makes it possible to perform parallel computation and to have a very short response time for tasks that involve real-time simultaneous processing of several signals. Furthermore it is also possible to have a decentralized architecture, which is much more fault tolerant to loss of individual processors than centralized architectures.

Another important feature of artificial neural networks is that, although each processor is very simple in terms of computational power and memory, they are adaptable nonlinear devices. Consequently, artificial neural networks can be used to approximate nonlinear models, an essential property for solving many real-world problems. The adaptable parameters of artificial neural network models are the connections that link the processors. This is similar to "learning" in biological neural networks that is supposed to be the result of changes in the strength of the connections between neurons.

Research in artificial neural network models began at the same time as the first digital computers where being developed in the 1940's. In 1943 McCulloch and Pitts [McPi43] proposed modelling the biological neuron as a simple threshold device, i.e. it could be in only two states, on or off. In 1958 Rosenblatt proposed the Perceptron [Ros58] as a model for visual pattern recognition. In 1960 Widrow and Hoff [WiHo60] proposed the ADALINE (adaptive linear neuron). However, when in the late 1960's Minsky and Papert [MiPa69] pointed out some limitations of the artificial neural network models available at the time (e.g. the lack of a reliable algorithm to train multi-layer networks), interest in neurocomputing was greatly reduced and efforts were shifted to the area of Artificial Intelligence and expert systems.

Interest in neurocomputing only reappeared in the mid 80's mainly as a result of a combination of the following factors: a) the popularization of Hopfield's work to solve optimization problems using feedback networks [Hop85]; b) the rediscovery of the Back-Propagation algorithm, used to train multi-layer feedforward networks, by Rumelhart, Hinton and Williams [RHM86], c) the realization of the limitations of Artificial Intelligence and expert systems approaches, and d) the availability of powerful and cheap digital computers that could be used to simulate, test and refine artificial neural network models.

Nowadays artificial neural network models are the subject of study in many areas as diverse as medicine, engineering and economics, to tackle problems that cannot be easily solved by other more established approaches.

In this work our motivation is to develop techniques that exploit the properties of nonlinear modelling, adaptability and tolerance to internal damage exhibited by artificial neural network models in order to solve the problems of: a) extremum control of static systems; and b) adaptive control of nonlinear dynamical systems under feedback.

In the extremum control problem the aim is to estimate on-line, i.e. as new data are made available, the input that maximizes the system output, considering that the input-output relationship is unknown. Only noisy measurements of the output and the respective input values used to produce them are available. Previous work [WeZa91] has dealt with the case when the system is assumed to be governed by a quadratic function with unknown parameters. In this work we develop a neural solution for the case when the input-output relationship is non-quadratic but with a unique maximum over the interval of interest.

Control theory in the areas of analysis and design of time-invariant dynamical linear systems is well developed due to the intense research effort expended since the early 60’s. However, several dynamical systems of interest may contain severe nonlinearities and therefore a linear model will not be entirely suitable. Unfortunately, the nonlinear control field is much less advanced than its linear counterpart and few general approaches exist. In this work we modify and analyze the neural control architecture known as feedback-error-learning, which operates under feedback control by using an artificial neural network as a feedforward controller.

1.2 - Structure of the Thesis and Contributions

In chapter 2 the basic concepts relating to artificial neural network models are presented. The chapter begins with a simplified section about the human nervous system and the human brain. A general framework for artificial neural network models is later introduced and the most important feedforward artificial neural network models, i.e. network models that perform static mappings, are then presented. Finally, some limitations of the artificial neural network approach are discussed.

Chapter 3 is concerned with feedback artificial neural network models. Because of the presence of feedback connections, feedback networks are complex nonlinear dynamical systems. The two main application areas for feedback networks are as associative memories and to solve quadratic optimization problems. Two models, the Hopfield and IAC (Interactive Activation and Competition) neural networks are presented and analyzed. The main contribution of this chapter is the mathematical proof that the IAC network can also be used to solve quadratic optimization problems, much like the Hopfield network. As far as we are aware, this is the first time that it is shown that another feedback neural network model can be used to solve quadratic optimization problems.

One of the limitations of current feedforward artificial neural network models, as it is pointed out in chapter 2, is that a large number of iterations is needed if the current training algorithms are used. The contribution in chapter 4 is the presentation of a novel method that speeds up learning by constraining the location of the decision surfaces defined by the values of the weights arriving at the hidden units. The same method can be used to provide a better initialization procedure for the network weights. The performance of the method is evaluated through computer simulations.

Chapter 5 is concerned with fault tolerant artificial neural networks, i.e. networks that are tolerant to loss of weights and hidden units. The problem of training an artificial neural network can be seen as an optimization problem. Therefore it is not surprising that current algorithms, such as Back-Propagation, will not necessarily result in fault tolerant solutions since they do not explicitly search for a fault tolerant solution. In this chapter we propose the BPS algorithm (involving Back-Propagation with Switching). The proposed algorithm switches during training between the different fault configurations, i.e. all possible fault configurations are trained and forced to share the same set of network weights. The conventional Back-Propagation algorithm can be considered a special case of the proposed algorithm where the set of possible configurations contain only the no-fault configuration. The benefits of the proposed algorithm are illustrated using a bit-mapped image recognition problem.

The main contribution of chapter 5 is the mathematical analysis that shows that, by considering network fault tolerance, the problem of training the network is regularized, that is the number of local minima is reduced. We show that in some cases when the weights of a layer are fixed there is only one set of weights that minimizes the cost function.

Chapter 6 deals with the extremum control problem. First the extremum control problem is introduced and the limitations of the quadratic model approach are presented. The contribution of chapter 6 is the development of a novel non-standard artificial neural network model. Since the network model is flexible enough to accommodate non-quadratic functions, the optimum input for static systems with an asymmetric performance index can be estimated with a small error, even if the system is excited by a dither with a large amplitude. The standard Back-Propagation algorithm, with the necessary modifications for the specific network model developed in this chapter, is used to adapt the network parameters. We mathematically prove that the proposed network model can also be used in the multi-input case (theorem 6.1). Two simulation examples are presented, one for the single input case and the other for a two input case.

In chapter 7 we address the use of artificial neural networks for control of dynamical systems. First we review the main approaches proposed in the literature to integrate feedforward artificial neural networks into the general control structure. The concept of feedback-error-learning, proposed by Kawato [Kaw90], is then introduced. The first contribution in this chapter is the development of a modified feedback-error-learning control structure which aims to perform closed-loop identification of the inverse dynamical system. Such a modified control structure is then mathematically analyzed and we show that, at least for the case of a single input single output linear dynamical system, when certain requirements are satisfied, there exists an artificial neural network (a linear filter in this case) that is a close approximation of the inverse dynamical model of the system under control. A computer simulation for the linear case is used to illustrate the use of the proposed neural control structure.

The second contribution in chapter 7 is the introduction of the technique of variable (or adaptive) feedback to be used in the proposed neural control structure. Simulations of the control of a two-joint robot, a two-input two-output nonlinear control problem, are presented and we show that use of the variable feedback technique improves the generalization of the neural network controller in relation to trajectories not used during training.

The third contribution in chapter 7 is the application of the BPS algorithm, presented in chapter 5, to improve the fault tolerance of the neural controller in relation to faults in the neural network. The control of an inverted pendulum is used as the simulation example in this case.

Finally the last chapter presents general conclusions and suggestions for further work.