1 Why neural computing? a personal view

I. Aleksander

Department of Electrical Engineering, Imperial College of Science and Technology, London, UK

Abstract

The sudden growth of interest in neural computing is a remarkable phenomenon that will be seen by future historians of computer science as marking the 1980s in much the same way as research into artificial intelligence (AI) has been the trademark of the 1970s. There is one major difference, however: in contrast with AI, which was largely an outlet for a minority of computer scientists, neural computing unites a very broad community: physicists, statisticians, parallel processing experts, optical technologists, neurophysiologists and experimental biologists. The focus of this new paradigm is rather simple. It rests on the recognition by this diverse community that the brain 'computes' in a very different way from the conventional computer.

This is quite contrary to the focus of the AI paradigm, which is based on the premise that an understanding of what the brain does represents a true understanding only if it can be explicitly expressed as a set of rules that, in turn, can be run on a computer which subsequently performs artificially intelligent tasks. Those who contribute to neural computing believe that the brain, given sensors and a body, builds up its own hidden rules through what is usually called 'experience'. When a person activates his muscles in complex sequences driven by signals from his eyes, from sensory receptors in his muscles and even from his ears when performing an every-day act such as getting on a bus, or when he notices a 'polite chill' in a colleague's voice, these are examples of large numbers of implicit rules at work in a simultaneous and coordinated fashion in the brain. In neural computing it is believed that the cellular structures within which such rules can grow and be executed are the focus of important study as opposed to the AI concern of trying to extract the rules in order to run them on a computer.

Neural computing is thus concerned with a class of machines that compute by absorbing experience, and in that sense is a class which includes the brain, but may include other forms with similar properties. It is for reasons of extending interest to these other forms that we have

chosen to use the word *architectures* in the title of this book. Its authors are not latter-day Frankensteins in the business of *making brains.* They are, however, united in trying to *understand* computing structures that are brain-like in the sense that they acquire knowledge through experience rather than preprogramming. So, *Neural Computing Architectures* is not about the details of mimicking the neurons of the brain and their interconnections, but more about the nature of the broad class of machines which behave in brain-like ways and, through this, adding to both our armoury of knowledge in computing and to our ability to apply such knowledge through the design of novel machinery.

Perhaps, from all this it may be possible to draw out a definition of neural computing:

> *Neural computing is the study of cellular networks that have a natural propensity for storing experiential knowledge. Such systems bear a resemblance to the brain in the sense that knowledge is acquired through training rather than programming and is retained due to changes in node functions. The knowledge takes the form of stable states or cycles of states in the operation of the net. A central property of such nets is to recall these states or cycles in response to the presentation of cues.*

1. Origins

There is undoubtedly a certain degree of hype associated with this field. Phrases such as 'the dawn of a new era' are used by conference organizers, and the press talks of 'new computers that are built like the brain and really think for themselves'. But there is nothing new about neural computing: it is as fundamental as the more conventional or 'algorithmic' mode. Norbert Wiener in his 1947 book *Cybernetics* writes:

> *Mr. Pitts had the good fortune to come under Dr. McCulloch's influence* (*in 1943*) *and the two began working quite early on problems concerning the union of nerve fibres by synapses into systems with given overall properties... They added elements suggested by the ideas of Turing in 1936: the consideration of nets containing cycles.*[1]

So some of the discussions that fill the pages of this book and echo in the auditoria of conferences were begun more than ten years before the invention of the computer that we know and love. The McCulloch and Pitts model of the neuron is still the basis for most neural node models, and Turing's concern about nets and cycles is the very stuff of neural computing. Indeed, the 1960s were most productive in this area. The work of Rosenblatt of Cornell University on 'perceptrons' is well known, as is the destruction of its credibility in 1969 by Minsky and Papert of MIT which led to a halt to such work in the USA.[2] More detailed reference to these events may be found in other parts of this book. It is as a reaction to this mistaken criticism that the current revival started in 1983 with analyses that are summarized in Part IV of this book.

But in Europe, neural net researches were not as prone to the winds of change that blew from the direction of MIT as their colleagues in the USA. Eduardo Caianiello in Italy and Teuvo Kohonen in Finland continued to develop an understanding of neural computers to great depth and elegance. I am pleased to have been able to include their contributions in this book. I, too, largely due to a fascination with how well and fast the brain performs tasks of pattern recognition using components much slower than those found in computers, continued designing machines based on the neuron models that I first defined in 1965. These are characterized by the fact that they are easily implemented in electronics and can be understood using formal logic. Part II of this book is concerned with this approach, which has led to the commercialization of practical systems and which points to new high-performance systems for the future.

While this approach is not new, there is no doubt that the work of the 'Parallel Distributed Processing' (PDP) group in the USA has been fundamental in nailing down both the language and the targets of the current paradigm, and it is for this reason that Part IV of this book is an extended review of the pair of books generated by this group under the PDP banner (Rumelhart & McClelland, 1986).[3] But what do the rapidly expanding band of workers in neural computing hope to achieve?

2. Four promises

There appear to be four major reasons for developing neural computing methods, the first of which is a rebuttal of the Minsky and Papert criticism. Although this is not the place to debate the technical issues, it is helpful to note that the criticism was founded on a demonstration that there are simple pattern recognition tasks that neural nets appeared not to be able to accomplish. It is now clear that this conclusion was mistaken because it was founded only on a restricted class of neural system. In fact, the first promise of neural computing is that it is *computationally complete*. This means that, given an appropriate neural structure, and appropriate training, there are no computational tasks that are not available to neural nets. This does not mean that a neural net is as efficient at performing certain tasks as a conventional computer. For example, in order to perform multiplications, the net may have to learn multiplication tables in the way that a human being does, and it can be easily outperformed by a fast arithmetic unit in a conventional computer. But there are tasks for which the neural net not only outperforms the conventional computer but is the *only* way of performing the task.

This leads to the second promise: *functional use of experiential knowledge*. It is here that the neural net can perform functions beyond the capability of rule-based, conventional systems. Typical are the Achilles' heels of artificial intelligence: speech, language and scene understanding. The problem with conventional approaches to these tasks is either that rules are difficult to find, or the number of such rules explodes alarmingly even for simple problems. Imagine having to distinguish between the faces of two people. What information should be extracted? What should be measured in this information? How can we be sure that what we measure will distinguish between the faces? Although a considerable amount of study may provide the answers to some of these questions and, when compiled into a program, may actually differentiate between the faces in question, there is no guarantee that the same measures can be applied to another pair of faces. In contrast, 20 seconds of exposure to a neurally based system such as the WISARD (Aleksander *et al.*, 1984) will allow the net to select among a vast number of rules (node functions) in a very short time in order to provide the best discriminators between the images in question.

The third promise is *performance*: rapid solutions to problems which in conventional computers would take a long time. For example it has been possible to solve the 'travelling salesman problem'* in many fewer steps than by conventional (exhaustive) algorithm.

But there is a snag to the exploitation of this performance: neural systems have actually to be built or run on general purpose *parallel* machines. It is worth pointing out that machines such as the connection machine (Hillis, 1986) are not neural systems. They are general purpose parallel systems that require programs as much as any conventional machine. But the program could be the structure of a neural net, that is, an emulation which, due to the parallelism of the host machine, exploits the speed with which the neural system is capable of solving some problems. Indeed, several 'neural computers' that are appearing on the market are emulations of this type. A useful function that they perform is to provide a tutorial vehicle that gives their users experience in the way such systems work. The first serious neural computer capable of solving real-life problems in real time is still to be built. Although this book

*The travelling salesman problem concerns the finding of the shortest route between geographically scattered points. This is traditionally difficult for conventional machines because it relies on the testing of an astronomically large number of paths. The neural computer, by performing local operations in parallel and then allowing these to interact, finds solutions very rapidly.

contains no specific designs from which such a system might emerge, it does contain information that may be important for anyone wishing to embark on the design of such a machine. There are many opportunities open for the design of the neural node (eg by optical means, conventional memory chips or special very large scale integrated systems (VLIS)).

The fourth and final promise of neural computing is the provision of an *insight into the computational characteristics of the brain.* This is very much the stated aim of the authors of the PDP books, but is not emphasized strongly in the structure of *Neural Computing Architectures.* In fact, it is becoming apparent that the nature of the research that one does in neural computing will differ according to whether one is concerned with (1) the understanding of principles and the design of machines on one hand, or with (2) brain modelling on the other.

In (1) above, general structures are investigated, while in (2) certain structure characteristics may be ruled out of court should they not conform to what is known of the brain, even if such structures may be computationally highly competent. But this book does not completely ignore an interest in brain modelling: in Part I, some concern is shown for the mechanisms of language understanding in humans which may contribute both to the creation of novel machinery as well as providing a deeper analysis of what may be happening in the brain when it is 'understanding' language.

3. About this book

As already stated, the aims of this book are to provide a guide to architectural issues that arise in neural computing. We see these as being complementary to those of the PDP volumes which are subtitled 'Explorations in the microstructure of cognition'. Clearly, a better understanding of the subject matter constituting cognition in humans is of common concern, but in this book it is the influence of physical structure which unites the authors. The book also aims to present a view (by no means exhaustive) of concerns in Europe, while the PDP books represent the work of researchers in the USA.

In Part I, 'The Classical Perspective', some of the longest-standing European contributors make their statements alongside some related work from more recent arrivals. This perspective is not only fundamental in its own right, but also adds substantially to understanding in a way that is complementary to similar work in the USA.

Part II, 'The Logical Perspective', represents lesser known work in which the neuron is modelled as a logic truth function which can be implemented in a direct way as a silicon read-only-memory. However, it is not only the implementability that is significant in this perspective, but also the fact that neural computing is seen from the mathematical perspective of logic, automata theory and probability theory. This too is seen as being complementary to the favoured approaches in the PDP style which use statistical mechanics as the analytic substrate, the two approaches together forming a more solid pillar for the understanding of neural computing than either of the methodologies on their own.

Part III, 'Analysis and Implementation', presents new material both in the form of analytical tools and models, as well as a suggestion for implementation in optical form.

Part IV, 'The PDP Perspective', is a single extended chapter that summarizes the considerable wealth of theory, application and speculation represented by researchers in the USA. This is not intended to be a replacement for reading the PDP volumes: it is included for completeness so that the authors of *Neural Computing Architectures* can make reference to this work without leaving the reader in the dark, until such time as he can get around to reading the PDP work.

4. The future?

The considerable hype surrounding much current work on neural computing is by no means constructive, but it is at least self-defeating. Many laboratories new to neural computing are discovering that it is not fruitful to cobble together *any* simulation of a neural net, and then hope that it will compute the first thought-of task. This quickly diverts the thrust towards the need to understand what can and cannot be expected of a particular net, and the way the parameters of a net are optimized. The aim of the authors of this book is to contribute to such understanding, which is the best way of fighting the exaggeration.

So what is the ultimate neural computing architecture of the future likely to be? This is an area on which the authors may differ, mainly due to their dedication to the understanding of specific approaches. But one thing does seem to be evident. Neural computing of the future is not likely to be a *replacement* of conventional computing and AI programs, but, rather, is likely to form a complementary technology. It would border on the silly to create with difficulty neural computations that can be performed with ease through conventional methods. The key issue, however, is that the two methods must be able to exist under the same roof (or metal box). So the ultimate challenge for experts in computer architecture is to exploit the two technologies within the box, while presenting a single, flexible interface to the user.

References

1. Wiener, N. *Cybernetics* (Cambridge Mass: MIT Press, 1947).
2. Minsky, M. & Papert, S. *Perceptrons: An Introduction to Computational Geometry* (Boston: MIT Press, 1969).
3. Rumelhart, D. E. & McClelland, J. L. (eds.) *Parallel Distributed Processing*, Vols. 1 and 2 (Cambridge Mass: MIT Press, 1986).