SYMBOLS VERSUS NEURONS?

JOACHIM STENDER
and
TOM ADDIS
EDITORS

IOS
Amsterdam, Washington, Tokyo

FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS

Symbols versus Neurons?

Edited by

Joachim Stender
Brainware, Berlin

and

Tom Addis
University of Reading, UK

IOS Press

1990

Amsterdam • Washington, DC • Tokyo

ISBN 90 5199 039 1

Publisher:

IOS
Van Diemenstraat 94
1013 CN Amsterdam
Netherlands

Distributor in the U.S.A. and Canada:

IOS Press, Inc.
P.O. Box 10558
Burke, VA 22009-0558
U.S.A.

Distributor in Japan:

IOS Japan Dept.
Highway Development Co. Ltd.
1st Golden Bldg, 8-2-9 Ginza
104 Tokyo - Chuoku
Japan

PRINTED IN THE NETHERLANDS

SYMBOLS VERSUS NEURONS?

FRONTIERS IN ARTIFICIAL INTELLIGENCE
AND APPLICATIONS

Previously published in this series:

B. Wielinga et al. (Eds.), Current Trends in Knowledge Acquisition
H. Kangassalo et al. (Eds.), Information Modelling and Knowledge Bases
G. Schuyten, M. Valcke (Eds.), Teaching and Learning in Logo-based Environments
K. Voss et al. (Eds.), Computer Analysis of Images and Patterns
H. Jaakkola, S. Linnainmaa (Eds.), Scandinavian Conference on Artificial Intelligence — 89
T. Danielsen (Ed.), Scandinavian Conference on Artificial Intelligence (SCAI '88)
D. Bierman et al. (Eds.), Artificial Intelligence and Education
C. Queinnec, J. Chailloux (Eds.), LISP Evolution and Standardization

ISSN: 0922-6389

PREFACE

In recent years, transputer-based parallel computers have gained in significance as a platform for the development of AI applications and tools. Subsymbolic or neoconnectionist approaches are occupying a growing position at the side of classical symbolic approaches. This book highlights this development with a comparison between symbolic and connectionist approaches and their implementation. In posing the question "Symbols versus Neurons?", a forum is provided for directly tackling the central conflict in the AI debate today.

This volume contains the papers of the 2nd International Conference of the OUG Artificial Intelligence SIG, held at IEE London on 1st October 1990. The conference was organised by the Institution of Electrical Engineers (Professional Group Committee C4 - Artificial Intelligence) in collaboration with the Occam User Group (Artificial Intelligence Special Interest Group). The Occam User Group is an informal organisation concerned with all aspects of the programming and application of transputer-based architectures.

The Occam User Group is pleased to acknowledge the support and sponsorship for this conference provided by Inmos. On a personal note, we should like to express our thanks to all people at Brainware who supported us in compiling this volume, especially Ms. Eva Hillebrand, Ms. Catriona Kennedy, Ms. Sylvia Beamish and Mr. Ilian Chorbadjiev, as well as Pam Addis.

Joachim Stender
Brainware GmbH
August 1990

Tom Addis
University of Reading

Contents

Symbols versus Neurons?

Some Remarks on the Central Debate in Intelligent Systems

Tom Addis[1], Catriona Kennedy[2], Joachim Stender[2]
[1]University of Reading, Department of Computer Science, P.O. Box 220, Whiteknights, Reading RG6 2AX, United Kingdom, and [2]Brainware GmbH, Gustav-Meyer-Allee 25, D-1000 Berlin 65, West Germany

Abstract
The title "symbols vs neurons" in the field of information technology refers to the distinction between Turing Machine symbol processing and the "evolutionary" approach based mainly on Neural Network architectures. However, this title refers to a much deeper and long standing division between these two approaches than mere technique. The division separates different ways of thinking about problem solving, understanding and design.

1. The "symbolic" approach in Intelligent Systems

In discussing the issue of "Symbols vs. Neurons?" in intelligent systems, it is first necessary to characterise more precisely what is meant by "symbol processing" and to distinguish it from the more "evolutionary" approach based mainly on Neural Network architectures. "Symbol processing" focusses on *representation*, in particular *knowledge representation*, as the central problem in constructing intelligent systems. The main hurdle is to translate a programmer or domain expert's mental concepts into formal symbols. Once this formal encoding exists then any computer can process it and therefore behave in a similar way to the expert or problem solver. This formal symbol processing is regarded as the "intelligence" of the system.

This paradigm is generally "user" centred. It is most important that the user's problem is accurately represented and that the outputs of the system should seem intelligent to a user or problem solver. The internal structure of the system does not matter a great deal. The pattern follows that of the traditional Von Neumann computer architecture: INPUT -> PROCESS -> OUTPUT. The characteristics of such a system are the same as those for a formal system. i.e. the "symbols" are syntactically correct strings to be processed. The processing involves transformation and matching of these strings to produce other strings. An external interpretation is necessary for these strings to have any sense.

The advantage lies in the *generality* of formal systems. AI languages such as LISP can have enormous representational power. One criticism, however, is that it is only the user's own intelligence that comes into effect. There is no model of "meaning" or "symbol" within the system itself and it could be regarded only as a very crude pattern processor whose actions are analogous to shuffling around papers in a filing cabinet.

Probably one of the most serious limitations of such formal string processing systems is that all the responses must normally be explicitly programmed beforehand. This problem of explicit programming is also a cause of the "software crisis" in complex real-time systems. It is simply not possible to program for every sequence of events that can actually happen in a noisy environment.

If an unusual combination of events occurs then the system can "crash" or respond senselessly. This problem of brittleness also applies to present day expert systems. Such predefined formal systems are often called "closed" systems because there is no capacity to react to the *environment* where novel situations can occur and bring about orderly (but perhaps unpredictable) changes in the system's reponses.

2. The "evolutionary" approach

In contrast to the symbolic paradigm, the "evolutionary" approach emphasises *learning* and *adaptation* as the key problems in intelligent behaviour. Of particular importance are *emergent properties* and *self-organisation*. The applications at present are mainly in the areas of pattern recognition and adaptive control systems. The "intelligence" of this type of system is most usually its ability to distinguish between one general class of patterns and another general class without being directly programmed. For example, a hand-written letter such as "T" forms a class of patterns which is different from "H" say. The ability to adapt to and respond sensibly to new or unexpected occurrences is also an important feature of such systems.

The evolutionary approach is more "system" centred than the symbolic approach. The internal structure of the system itself is critical. For example, in a neural-network based system, the changes in the connections between the units indicate the measure of learning that has taken place. The influence of the environment is emphasised much more here [the environment is part of the system]. As a result many such systems (especially those that change their own structure in a non-deterministic fashion) may be called "open" systems.

In such systems, however, there is no symbolic communication with the user. In other words, they are said not to do "symbolic processing". The responses of the different units must be carefully analysed. As a result, present day neural networks do not make use of the representational power of AI formalisms.

3. Historical Background

In order to fully appreciate the development of the two separate philosophies of "symbols" and "neurons", it is necessary to consider the roots and history of the whole field. In 1943 a group of scientists, engineers and mathematicians associated with both Arturo Rosenblueth and Norbert Wiener (who had previously met at a series of informal discussions at the Harvard Medical School) became aware of the essential unity of the set of problems addressed by communication, control and statistical mechanics. This new field was called Cybernetics and was intended to be the study of self-organising systems; systems that may have many different material forms. These forms could be artificial (e.g. machines) or natural (e.g. living tissue) and they could be single entities (e.g. an animal) or groups of entities (e.g. societies).

The main theoretical framework was information theory (proposed by Claude Shannon) supported by analogue and discrete mathematics. The subject was strongly influenced by Pavlovian psychology and the physiological structures found in animal nervous systems. Models of natural systems were constructed from observations of the behaviour of the components and the manner in which these components were connected with each other and with the environment. It was believed that nature was parsimonious and as such intelligence would emerge from the complex interactions of essentially simple mechanisms. As a result much work was done during the period 1940 to 1960 on the construction of neural nets and the theoretical framework to support them by such people as Warren McCulloch, Walter Pitts, Marvin Minsky and Frank Rosenblatt.

It was in 1956, at a conference held at Dartmouth College (USA), stimulated mainly by John McCarthy but also supported by Marvin Minsky, Nathaniel Rochester and Claude Shannon that a new discipline called Artificial Intelligence was established. Artificial Intelligence was eventually defined to be the symbolic modelling of intelligent behaviour; it was considered to be primarily an engineering activity rather than a science. However, the definition specifically excluded an important element of Cybernetics; an element that allowed for adaptive systems which included the environment as a non-symbolic component.

The main theoretical framework for Artificial Intelligence emerged from the work of Alan Newell and Herbert Simon; it was constructed from utility theory in conjunction with the search of problem space. The subject was strongly influenced by formal logic as well as Skinnerian and Cognitive Psychology. It was believed that symbol manipulation was sufficient to model human reasoning and that human intelligence would emerge through the extrapolation of techniques derived from computation.

In 1963 Edward Feigenbaum and Julian Feldman published a collection of papers under the title *Computers and Thought* (published by McGraw-Hill). This book contained most of the major techniques of the time. They were techniques that effectively circumscribed the subject by restricting the notion of Artificial Intelligence to just those techniques; techniques such as heuristic search, production rules, weighted decision making, semantic structures, primitive induction and parsing.

After 1968, when Minsky and Papert published their famous critique of the perceptron, this argument for pure symbolic processing gained further strength. It was supported by the theoretical proof that a Universal Turing Machine (UTM) could simulate any computable process. A symbol processing system (which is a variant of a UTM) could therefore also be made arbitralily powerful. Neural networks could not possibly do anything *more* than this (since they were just parallel versions of the same thing) and indeed it appeared that the primitive neural networks so far invented could do much less. So why use neural networks?

By the end of the 1970s, a considerable amount of work had been done in this field of "symbolic" Artificial Intelligence (or AI) and the critical issue had become knowledge representation. This led to the discipline of "expert systems" and "knowledge engineering" as we know them today. In contrast, very little work had been done on neural networks during this time.

4. Philosophical and Conceptual Issues

What is of great significance about the two fields is the different underlying perception which each field had of human nature. Cybernetics saw people as components in a much larger system (i.e. the world and society). Intelligence was not just found in the complexity of the central nervous system; it was the result of the central nervous system having an extension into the world through an active body. "Symbolic" Artificial Intelligence, on the other hand, saw intelligence as being solely the result of the central nervous system. This underlying difference in philosophy reflected the attitudes of the society in which each subject flourished and as a result Artificial Intelligence became dominant in the USA and Cybernetics became dominant in the USSR.

Hubert Dreyfus published a strong argument against machine reasoning of all kinds and in particular against symbolic processing (*What Computers Can't Do*, 1972, published Harper & Row). His attack consisted of two main points. Firstly, formal systems were, in a real sense, detached from the world. This detachment had the serious consequence that human interpretation was always necessary. Secondly, machines of any kind were materially different in that they were at best models but usually simulcra of the world. He was not suggesting by this argument that there was any supernatural component but only that the perception of an object in the world was always in part less than the object itself, since such a perception can only be an abstraction. Further, any machine that models an abstraction will always embellish such an abstraction with its own properties.

Ten years later, Dreyfus extended his book with an extensive introduction reviewing the apparent lack of success of Artificial Intelligence. Supporters of Artificial Intelligence responded to his criticism by simply pointing out that compared with other disciplines Artificial Intelligence was still in its infancy and that all human understanding of the world must be derived from theories that model the world. An Artificial Intelligence program was in some sense a complex theory that was best understood by tracing its behaviour by computer.

However, the argument still holds that these theories (and hence symbolic representations based on them) are "hand-crafted" by a particular intelligent system (namely humans) who have adapted to a specific environment (the ecosystem) with a specific goal (to survive). Surely these theories and concepts must be restricted by our very specific evolution and experience (needless to say culture and society also play a large part here). The set of all problems solvable by pre-programmed systems could in practice be much smaller than that for a UTM simply because the correct programming (hand-crafted) can never exist for some types of problems.

This is really where the argument for "evolutionary" systems based on Cybernetics becomes very strong. In a self-organising neural network, for example, it may be possible that totally new things can arise, which a human programmer or expert would never have thought of. The potential for this type of system is particularly high in environments where humans have not adapted (e.g. masses of data being received by a space telescope). A non neural net example of this "innovation" is in the use of simulated evolution. In some simple cases this has yielded engineering designs which have surprised the experts and has turned out to be more "suitable" (or "fit") for a specific environment than one based on a theory[1,2].

5. The PDP-"Resurrection"

At the beginning of the 80's work in the general area of parallel distributed processing (PDP) was established by such people as David Rumelhart, James McClelland and Geoffrey Hinton (*Parallel Distributed Processing,* 1986, published by MIT press). This was, in effect Neo-Cybernetics since this work centred around ideas drawn from information theory (e.g. the

Boltzmann Machine) and involved the essential notion that intelligence would be the emergent property of the connectivity between simple elements.

A different theoretical framework is being developed for PDPs to that of symbolic processing. A different framework is needed since the "mechanism" itself cannot be taken as a model whose behaviour can be understood. The need for an abstraction of the mechanisms that would provide a unified view had already been encountered in the 50's and 60's by those mathematicians, physiologists, psychologists and engineers who created models of animal behavioural systems. Sadly, the neo-cyberneticists make little reference to this earlier work; work such as that done by Grey Walter who first introduced the Machina Speculatrix, The Conditioned Reflex Analogue and the electric model of the nerve (1953, *The Living Brain*) or the work of Ross Ashby who created the Homeostat and an elegant mathematical theory of multi-connected units (e.g. neurons or people) (1952, *Design for a Brain*) or the later work of Igor Aleksander who built Wisard, a mechanism that uses a variant of the neural net for industrial pattern recognition and some others. They do, however, acknowledge Marvin Minsky's and Seymour Papert's excellent analysis and criticism of the Perceptron and many others' related works (e.g. D.O. Hebb, 1949, *Organisation of Behaviour*).

6. Bridging the Gap

We now enter a decade in which it is desired that a new set of directions will arise from addressing the problem of bridging the long standing division between those who manipulate symbols and those who connect neurons. To a very small extent this is already happening. In the early 1980s there was a resurgence of interest in the branch of AI known as machine learning. Some models of machine learning which involve no "neurons" (and probably belong to the "symbolic" tradition of Artificial Intelligence) can behave like "open" or "evolutionary" systems in the sense that they change their structure and hence their behaviour in response to an unpredictable environment. In contrast, some simple rule induction programs could be said to process data that can only appear in certain forms and convert it into a decision tree. The decision tree might then be externally interpreted as a set of rules.

Another area of blurred distinction is in massively parallel processing. It is possible to program each processor sequentially (in the classical Von Neumann way) so that it behaves like a neuron. Then when many of these processors interact together (perhaps by sending interprocess signals or messages), the whole system could in theory act like a neural network. It could in fact demonstrate emergent properties and "evolution" even though each processor was individually programmed in the "symbolic" way. Transputer - based architectures are at present widely used for simulating massively parallel processing systems because those are not widely available, even for research purposes. Because of this, transputer-based systems are playing an important role in this area; even more, as also representatives of the symbolic paradigm are using more and more transputer-based systems for solving complex problems, for example, by implementing parallel inference engines on transputers. It must be said, however, that most uses of the transputer are purely for speed and reliability reasons. A homogenous architecture has not yet been developed which combines the "representational" advantages of AI with the "adaptive" advantages of neural networks.

It must also be said that both approaches create mechanisms that remain passive to the world. The next step must be a move towards reactive machines; machines which are driven by an underlying process and will draw the "meaning" of their internal structure directly from the environment. It may be possible to combine both approaches so that the development of such a system becomes possible.

This collection of papers shows that the search for the right way to understand intelligent systems and their relationships with the world is still continuing. Each paper describes either a "symbolic" or a "subsymbolic" approach to solving a particular problem. An overview of the "connectionist" papers is first given.

In "Connectionism - a link between Psychology and Neurosciences?", Zoltan Schreter tries to connect three traditionally different disciplines - *cognitive psychology, behavioural sciences* and *brain sciences* - using neural networks. A model is described which aims to account for psychological/behavioural data on the one hand and neurosciences data on the other. This is a model of attentional learning with context-dependent filtering of irrelevant features.

In "Simulating Neural Networks in Distributed Environments", Kimmo Kaski and Jukka Vanhala look at neural networks from the viewpoint of *computer science and parallel processing.* Two neural network architectures - Hopfield nets and Kanerva's Sparse Distributed Memory - are modelled using a network of transputers.

In "Real-time Reinforcement Learning Control of Dynamic Systems applied to an Inverted Pendulum", van Luenen represents the discipline of *control engineering*. A neural network version of the Adaptive Heuristic Critic (AHC) learning algorithm is presented and analysed. Experiments are done where the network learns to control an inverted pendulum with no specific a priori programming.

In "Neural Nets - Applications in Medicine" , von Goldammer and Paul describe the use of Kohonen nets to solve complex problems in medical diagnosis. These are implemented on a transputer network.

In "A Systolic Algorithm for Back Propagation: Mapping onto a transputer Network" Pau Bofill and José del Millán present a supervised, iterative, gradient-descent connectionist learning rule where a single systolic ring carries out sequentially the three main steps of the learning rule. It is implemented on a linear ring of transputer processors.

The following papers take the "symbolic" approach.

In "Knowledge and the Structure of Machines", Addis and Nowell propose a paradigm that may lead to a new perspective which will unify the different components of knowledge systems. This paper represents the relatively new discipline of *knowledge-based systems*, which is a branch of AI.

In "Relational and Differential Logic for Knowledge Processing", Stoytchev and Antonov represent mainly the viewpoint of *mathematical logic*. Relational Logic (RL) and Differential Logic (DL) are introduced. They look at intelligent systems using the paradigm of a *state space* and *transitions* between those states which begin from a START state and end with a GOAL state. This is very much a traditional AI viewpoint in that the problem solving process is represented symbolically.

In "Artificial Intelligence for Genomic Interpretation", Sallantin and Pingand present an Artificial Intelligence environment for the description of certain types of biological knowledge, in particular protein folding. Techniques used include Conceptual Graphs and Iterative Learning.

The following papers can help to bridge the conceptual gap between symbolic and evolutionary approaches.

The paper "Symbolic Constraint-based Reasoning in PANDORA" by Reem Bahgat is a deliberate attempt to bridge this gap. A non-deterministic parallel programming language (PANDORA) is introduced. The language is abstract and logic-based so it is ideal for knowledge representation. It is also designed to be used with parallel processing architectures which can act in a similar way to neural networks.

In "Inductive Protein Structure Analysis using Transputers", Steffen Schulze-Kremer uses representational/symbolic methods in conjunction with machine learning and parallel processing. A special symbolic language (Protein Representation Language or PRL) is introduced to represent knowledge on protein structure. A technique of machine learning known as Conceptual Clustering (introduced by Michalski) is also applied. It could be argued that this has similarities with evolutionary systems in that the structure of the environment is divided into classes (or "clusters") by the system.

In "Using the Genetic Algorithm to adapt Intelligent Systems", Terence C. Fogarty describes two applications of this algorithm which is based on natural evolution. The first application is to enable a symbolic rule-based system to adapt to its environment (i.e. develop "better" rules). The second application is in the "growth"of neural networks to solve a particular problem. (One could compare this to the biological evolution of the visual system).

In "Subsymbolic Inductive Learning Framework for Large-Scale Data Processing", Stender and Chorbadjiev are introducing a theoretical framework for describing machine learning techniques to analyse large-scale data sets. This also attempts to partially bridge the gap by combining subsymbolic mechanisms for knowledge representation with classical traditions in machine learning.

It is clear that this field of interest cuts across many disciplines. It is hoped that this collection will help to illuminate the contrasts between the different strands of thinking.

References:

[1] E. Gottzein, K. Janschek, W. Oesterlin and J. Colrat. *Design of the Hermes Orbital Flight Controller.* In: T. Nishimura (Ed.), Automatic Control in Aerospace. Selected papers from the IFAC Symposium, Tsukuba, Japan, 17-21 July 1989.

[2] Ralph Neumann. *Solving Complex Configuration Problems with Genetic Algorithms.* In: Pragmatica, IPSIG Spring 1990.

Symbols versus Neurons?
J. Stender, T. Addis, Eds.
IOS Press, 1990

Knowledge and The Structure of Machines

T. R. Addis and M. C. C. Nowell

Knowledge Systems Group
Department of Computer Science
University of Reading
Whiteknights
Reading
Berkshire RG6 2AX

Abstract. A paradigm is proposed based upon a taxonomy of knowledge; a taxonomy that has been strongly influenced by the need to represent knowledge for machine processing. The importance of such a paradigm is to show an equivalence of activity in all spheres of system design from knowledge systems to machine architectures and thus open up the possibility of cross fertilization of techniques and a redistribution of tasks across fields. Both these possibilities will improve the design of knowledge systems within the framework of a rapidly evolving technology. We first describe what is meant by a deterministic mechanism and then explain deduction in these terms. The new paradigm is then used to illustrate the coherence of CARDS. CARDS is an example of a family of front end processors based upon the transputer. Three transputers are formed into a pipeline that supports a content addressable relational mass storage system (1.6 Gbytes). This relational system is used, in turn, to support a Functional Database (FDB) language called FAITH. We show that a Functional Database Language has formal coherence with the representation of knowledge and that there is a natural link from functions (tuple at a time processing) to relations (set at a time processing) via the ZF function.

1. Introduction

The awkwardness of many large knowledge-based systems is, in part, due to the incoherence of the different 'technologies' that have come together to manipulate knowledge. Experts in hardware design do not necessarily have the breadth of understanding of data structures to create the 'best' kind of machines to support large databases and experts in databases are not completely aware of all the issues involved in manipulating and representing knowledge. Both these classes of experts may be forced to modify greatly their approach to their designs in the light of a complex application.

Part of the difficulty lies in the widely and often unrelated theoretical paradigms that each 'expert' uses in order to accomplish a design. Even within a single sphere of activity, such as the representation of knowledge, there can be many different and apparently irreconcilable points of view. For example, in knowledge engineering there are those who work in 'rules' or 'frames' or 'semantic nets' or 'PROLOG' and in systems analysis (i.e.