Proceedings of the
International Conference on Complex Systems
Nashua, NH, 21–26 Sept. 1997.

Y. Bar-Yam (ed.), New England Complex Systems Institute (1997)
Also on-line in the *InterJournal*.

An integrated theory of nervous system functioning embracing nativism and constructivism

Brian D. Josephson
University of Cambridge, Department of Physics, Cambridge CB3 0HE, UK

Neural constructivists have proclaimed, on the basis of arguments which it is suggested in the following are unconvincing, a view of the nervous system which has little place for innate knowledge relating to specific domains. In this paper an alternative position is developed, which by instantiating innate knowledge in a flexible manner provides a more credible alternative. It invokes dynamical systems whose behaviour initially accords with specific algorithms but is modifiable during development, in combination with an object-oriented programming architecture that provides a natural means of specifying systems that have special knowledge of how to approach specific classes of situation.

1 Introduction

The algorithmic approach to describing the mind that characterised the discipline of artificial intelligence has proved inadequate in contexts such as the recognition of handwriting. In recent years an alternative, called neural constructivism, has received considerable attention [1, 2]. It is based on neural network models that are treated not merely as models of skill acquisition but also as models of the developmental process overall.

A number of constructivists (whom I shall refer to as evangelical constructivists) have proclaimed in their writings the falsity of nativism, which is the hypothesis that there exist innate neural systems dedicated to specific domains of activity such as language. They hypothesise instead that flexible learning networks, not oriented towards any specific domain, can achieve all that can be achieved by domain-specific networks.

Close examination of the arguments given against nativism indicates that they are based on a restricted view of the possible forms that innateness might take whilst, on the other hand, the arguments in favour of flexible systems rather than domain-specific ones are based on optimistic extrapolations of what has been achieved by the existing constructivist simulations. The comprehensive case that has been built up by workers such as Pinker [3] for there being a language instinct is dismissed on the basis of a small number of possible flaws in particular arguments which, in the author's opinion, affects very little the cogency of the overall case made by these workers. Crucially, the critics fail to address the central question: if, as the arguments of workers such as Pinker suggest, specific devices can considerably facilitate the acquisition of language, why should nature employ general-purpose networks alone?

It must nevertheless be acknowledged that the arguments in support of domain-specific mechanisms for language have been spelt out within the paradigm of algorithms as opposed to that of neural networks, and thus cannot give the full picture, which should be subject neither to the familiar limitations of the algorithmic point of view nor to the limitations of the evangelical constructivist position which denies the presence in any form in the initial system of algorithmic structures related to specific domains. The question poses itself how we may get beyond both categories of limitation, thereby combining the specificity and precision of an algorithm with the flexibility of constructivist neural networks. Models with this feature can be expected to have a validity and potentiality considerably in excess of both the standard algorithmic and the evangelical constructivist points of view. This paper proposes initial steps in such a direction.

2 Fundamentals of an integrated theory

2.1 Algorithmic and constructivist approaches compared

We begin by discussing in somewhat more detail the two basic approaches referred to above, the one based on algorithms and the other based on neural networks. The basic idea behind neural constructivism is that of taking as primary the level of the neuron, and looking for network architectures and weight-changing rules, often associated with an optimisation process, that can give a system a powerful ability to acquire a range of skills. The rules that a trained network instantiates, unlike the kinds of rules normally associated with the term algorithm, are submerged in the details of the network, and to the extent that rules that emerge as a result of training can be codified, they reflect primarily the nature and structure of the environment.

The alternative, consisting of a system that works purely on the basis of algorithms, is very often conceptualised and/or implemented in the context of digital computation, in which case the algorithm concerned is instantiated by a piece of code written in an appropriate language. This kind of model implies a discreteness and a corresponding lack of flexibility that contrasts significantly with neural network models.

2.2 A third option

The problems of discreteness and lack of flexibility can be circumvented by changing the basic concept from algorithms as such to dynamical systems whose behaviour simulates or equivalently is approximated by an algorithm as conventionally understood. The possibility of such systems undergoing transformations over time introduces a kind of flexibility not present in the traditional algorithmic models such as those of artificial intelligence. As with traditional neural network models, such transformations could come about in response to assessments of the observed performance. The mechanics of change might involve either changing weights in the style of traditional neural network models or, more creatively, linking in an appropriate new process as a response to a problem or a challenge, in the manner of traditional artificial intelligence models.

We thus have these three main ways of implementing a rule:

(i) as a piece of code running on a digital computer. This is the most rigid way and corresponds to traditional artificial intelligence;

(ii) as something that emerges through problem-solving activity in an environment, owing its structure almost entirely to the environment and the nature of the solutions to problems. This is the method favoured by the evangelical constructivists;

(iii) as a consequence of the initial dynamics of a modifiable structure. This provides a flexible mechanism for generating adaptive behaviour that can be modified by problem-solving activity, and combines the merits of (i) and (ii). The existence of universal behaviour patterns, apparently not learnt, in the developing child, which gradually transform into activity well-adapted to the solution of problems, provides a strong argument in favour of (iii) being the actuality.

2.3 Dynamical systems perspective

The essential requirement for a neural network to implement an algorithm is for there to be a limited number of degrees of freedom, corresponding to a specific state space that is not too complex. The parameters of the system (which govern the dynamical laws governing the dynamics of this state space) have then to be adjusted so that the resulting behaviour accords with the algorithm concerned.

We are in reality concerned not with just a single algorithm but with a large collection of algorithms. Correspondingly, our concern is with a collection of interacting dynamical systems. These systems are autonomous to the extent that they can be considered in isolation for the purposes of analysis, but are not necessarily physically separate from each other.

2.4 The object-oriented programming architecture and other concepts from computer science

The idea of practically relevant computations being performed by a collection of subsystems sending messages to each other is found both in Minsky's *Society of Mind* model [4] and in the powerful methodology of *object oriented programming* [5]. The *class construct* of the latter seems particularly relevant in regard to the way it treats the creation of new objects or subsystems. Objects belong to classes that are associated with specific types of situation, and when a new instance of that class is encountered a new object of that class is constructed, according to rules associated with the given class. This is, if we like, a form of innateness in the world of computer programs, this parallel being enhanced by the fact that there normally resides on the computer system concerned a large collection of classes with their defining algorithms and protocols.

What would correspond to this in the nervous system would be a modification of the architecture described by Quartz and Sejnowski [2], incorporating a collection of modules that can play the role of objects. These objects may not be initially domain specific, but mechanisms may exist which will permit domain-specific systems to take them over at critical times and use them for memory purposes related to the needs of the domain. For example, when one was listening to language, processes specific to language would take over a number of such systems and modify them so as to represent the information contained in the instances of language involved, for example by creating structures homologous to the deep structures of language. Such processes would work more effectively if the information that they were dealing with had been organised in a way that reflected the specific kinds of linguistic entities. Neural counterparts of the class construct of object-oriented programming can be expected to facilitate such activity.

The above account, with its reference to "objects belonging to classes that are associated with specific types of situation", may appear to imply that our model contains *only* systems primed with expectations specific to specific domains. This is not the case, because of the possibility of there being an *uncommitted class*. Objects in this class are constructed whenever a new situation of interest arises that has no previous parallels.

In the above it has been postulated that the architecture of the nervous system parallels in particular respects the design of conventional computer programs. This would be a very logical state of affairs, since modern computer programs and the nervous system are both systems designed to perform tasks successfully in very complex situations. It might be expected that, subject to the constraints imposed by hardware, both kinds of system would have evolved similar solutions to problems. Creating structures reflecting very general regularities in nature, as done with the classes of object-oriented programming, is one example of a kind of evolution in computer science that was dictated by the important requirement of minimising the unnecessary duplication of computational structures (which is assisted by means of object-oriented programming's *inheritance mechanisms*, something equally relevant to nervous system design). Two other examples from computer science that may similarly be of value are *assignment operations* and *threads*.

In the computing context, assignment of a value to a variable is a routine component of a computation. Its instantiation in the form of a pointer from a register representing the variable to a register containing a value is one that might be readily implementable in the nervous system.

A thread [5] is a computational mechanism representing a process that has continuity over time but has dormant phases, and protocols for starting up again. A neural equivalent would be useful in a number of ways, for example to implement a process which is interrupted until a problem connected with it is resolved, or in the context of delayed reinforcement, where the success or otherwise of a process may not be known until some time after the process is complete, and a thread may be programmed to wake up at the appropriate time. Again, in the context of working memory, a thread could hold the current state of a search process and start up the search process again if the current trial proved to be unsuccessful.

It is hypothesised, then, that inventions from computer science may be of considerable relevance in modelling and understanding nervous system design and function.

2.5 Algorithms, mappings and order

[section added following the talks and discussions at the ICCS conference]
It is useful to think of an algorithm as implementing a function, not only in the usual informal sense (a specialised means of achieving an end) but also in the sense of a *mapping* from an initial state to a goal state. Furthermore, the functionality in a biological context generally takes the form of a restriction of possibilites or local reduction of entropy (which does not contradict the second law of thermodynamics since there is a corresponding source of negentropy elsewhere). Thus the nervous system structure is a special kind of structure having this capacity to reduce entropy locally. The idea can be taken one step further: *classes* and their corresponding constructors are systems specialised to the *generation* of locally entropy-reducing structures.

2.6 Relevance of hyperstructure theory

In §2.2 there was reference to 'transformations coming about in response to assessments of the observed performance'. Baas [6], and private communication, has suggested that processes of this kind may lead to the generation of new systems through what he calls *observational emergence*, essentially the emergence of systems associated with a process or structure of behaviour in some sense selected or induced by a particular observation process, acting together with the constraints imposed by the environment. This construction process may be very specific to the details of the domains concerned and build up in a cumulative, multi-level manner.

2.7 Illustrative analogy

A very basic account of the kinds of processes that might be involved is contained in the following simple analogy, which makes no attempt to fit accurately the demands of actual situations. In it, note carefully that the engineer, manual and instructions are simply devices to aid the imagination and do not in general correspond to actual component systems. The system as a whole behaves *as if there were a person following instructions*, in the same way as a thermostatic system behaves as if there were a person carrying out instructions to turn on the heating if the room got too cold, but in neither case is there a person, or explicit instructions that a person might follow (cf. [7]).

In the analogy, an engineer is in charge of a complicated electronic system which he is constantly upgrading, following the rules given in a manual. The system is composed of a large number of modules which come successively into use over time. Many of the modules (in accord with the experimental evidence relating to the cerebral cortex) are initially general purpose, but as they become functional the engineer marks them with tags where appropriate to indicate their function. Lights on the modules light up to indicate which ones are currently in use or have recently been in use.

A buzzer indicates the occurrence of a new event, but generally an existing system is able to handle such an event and the buzzer is then switched off. If this is not the case, a hitherto unused module responds to the buzzer and puts on a special light to indicate its availability. The engineer then connects it with other illuminated modules in accord with the rules in the manual. When this has been done the new module can start to play its functional role. It remains illuminated in a special manner for some time afterwards to indicate its status as a newly operational module deserving special attention, and the behaviour of the system at the times when it is performing its function is evaluated and appropriate adjustments are then made by the engineer. This activity is supported by systems which can remember the situation in which the new system was functioning, which allows for the corresponding activity to be repeated as often as is necessary in order that the processes occurring during learning can take place over a shorter period of time. How the above proposals are likely to work in the case of language is discussed in §3.

2.8 Evolutionary aspect

The existence of systems that work harmoniously together to yield a high level of proficiency in a given domain can be understood on evolutionary grounds. Mutations give rise to simple capabilities in a new domain which offer some increase in fitness. The way is then opened for a succession of further mutations which extend the existing domain or improve functioning in that domain. At every step the new features have to be coordinated with the existing ones (cf.

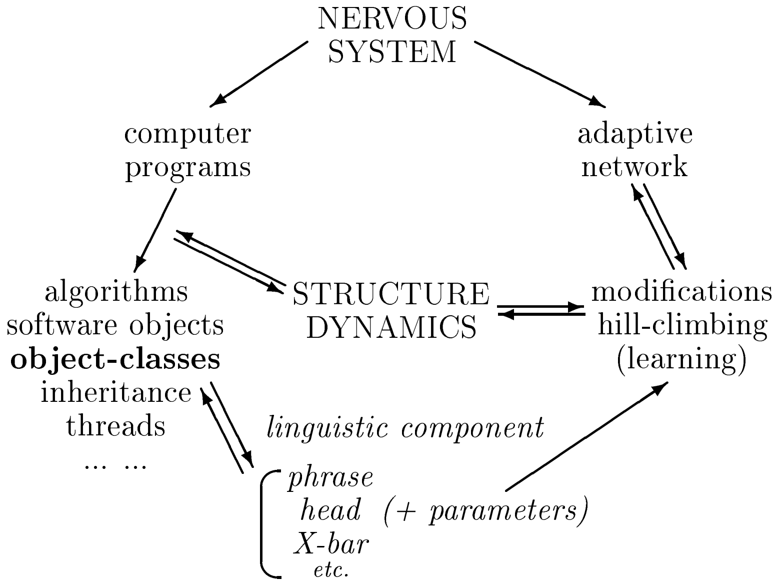

Figure 1: Diagram showing relationships between the various concepts discussed in this paper (see text for detailed explanations). Upper case is used for physical entities, and italics for constructs relating to language.

the robot models of Brooks [7]), so at every step one has a properly coordinated system.

3 The case of language

How the above proposals may work in the case of language is indicated by an adaptation of a proposal of Elman et al. [1]:

> "If children develop a robust drive to solve [the problem of mapping non-linear thoughts on to a highly constrained linear channel], and are born with processing tools to solve it, then the rest may simply follow because it is the natural solution to that particular mapping process."

If the possibility of innate systems dedicated to language is not ruled out in accord with the dictates of the evangelical constructivists, then we can change the above account to allow for a collection of *specific drives relevant to aspects of language acquisition*, and for *specific tools that take into account universals of linguistic structure*, and the corresponding classifications.

The existence of drives equates to specific mechanisms being liable to be activated. When this happens the outcome is monitored and weight changes made that amount to learning. When one kind of skill has been developed, another level of process can spring into action.

Some proposals regarding how this might be implemented in practice for language have been made by Josephson and Blair [8], whose proposals for the acquisition of linguistic skills are consistent with the specific facts about language described for example by Pinker, and with the general picture of systems dedicated to handling specific *classes of situation* and *entities* developed here.

4 Diagrammatic representation of relationships discussed

Fig. 1 shows digrammatically the various relationships discussed in the paper. First of all, the nervous system has been considered in its two complementary aspects, as an adaptive network and as a system functionally equivalent in some respects to a computer program. The concept of a computer program can be opened out into the various aspects listed (algorithms, software objects, etc.). These aspects are related to the structure and dynamics of the nervous system (for example, the software objects are dependent upon the structure and its interconnectivities, and the functioning of an algorithm to the dynamics of the structure). On the other hand, the adaptive network aspect of the nervous system reveals itself as modifications of the structure which may be related to hill-climbing.

One section of the diagram is concerned specifically with language. It lists a number of constructs of language, such as the phrase, which may have specific correlates on the computing side, such as specific object-classes. According to this schema, the linguistic capacities of the nervous system arise in a process that involves two steps: implicit in descriptions of language such as those of universal grammar there are certain *classes* with corresponding algorithms, and these computational constructs are then implemented in the neural hardware. The parameters of universal grammar may enter via a different mechanism, for example as the outcome of a particular tendency of a learning mechanism's activities to continue in a way corresponding to the initial learning.

5 Summary

It has been shown that a dynamical systems approach can integrate the algorithmic and neural network approaches to development, permitting the respective advantages of both schemes to be both retained and integrated. This appears to correspond more closely to the reality than do the anti-nativist ideas popular among neural constructivists. Methodologies deriving from computer science that are likely to have correlates in the design of the nervous system, such as object-oriented programming with its way of handling specific categories of situation and threads, were discussed. It is to be hoped that the generic proposals of the present paper can be confirmed in the future by more detailed specification and model building.

6 Acknowledgements

I am grateful to Profs. Nils A. Baas and Michael Conrad for illuminating discussions on the concept of hyperstructure and questions of the flexibility of algorithms respectively.

References

[1] ELMAN, Jeffrey L., Elizabeth A. BATES, Mark H. JOHNSON, Annette KARMILOFF-SMITH, Domenico PARESI, and Kim PLUNKETT, *Rethinking innateness: A connectionist perspective on development*, MIT (1996).

[2] QUARTZ, Steven R. and Terrence J. SEJNOWSKI, "The neural basis of cognitive development: A constructivist manifesto", *Behavioral and Brain Sciences*, to be published.

[3] PINKER, Steven, *The language instinct*, Penguin (1994).

[4] MINSKY, Marvin, *The society of mind*, Heinemann (1987).

[5] CAMPIONE, Mary and Kathy WALRATH, *The Java tutorial: object-oriented programming for the Internet*, Addison-Wesley (1996), or on the World Wide Web at http://java.sun.com/docs/books/tutorial/index.html

[6] BAAS, Nils A., "Emergence, hierarchies and hyperstructures", *Artificial Life III, SFI Studies in the Sciences of Complexity, vol. XVII*, (Christopher G. LANGTON ed.), Addison-Wesley, (1994), 515–537.

[7] BROOKS, Rodney A., "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation* **2**(1) (1986), 14–23.

[8] JOSEPHSON, B.D. and D.G. BLAIR, *A holistic approach to language* (1982), World Wide Web, at URL http://www.tcm.phy.cam.ac.uk/~bdj10/language/lang1.html