

Moody JE (1992) The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems. In: Moody JE, Hanson SJ and Lippmann RP (eds) *Advances in Neural Information Processing Systems*, vol. IV, pp. 847–854. San Mateo, CA: Morgan Kaufmann.

Murata N, Yoshizawa S and Amari S (1994) Network information criterion – determining the number of hidden units for artificial neural network models. *IEEE Transactions on Neural Networks* 5: 865–872.

Nowlan SJ and Hinton GE (1992) Simplifying neural networks by soft weight sharing. *Neural Computation* 4(4): 473–493.

Yao X (1999) Evolving artificial neural networks. *Proceedings of the IEEE* 87(9): 1423–1447.

### Further Reading

Bishop C (1995) *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Neal R (1996) *Bayesian Learning for Neural Networks*. New York: Springer-Verlag.

Read RD and Marks RJ (1999) *Neural Smoothing – Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press.

# Connectionist Implementationalism and Hybrid Systems

Intermediate article

Ron Sun, University of Missouri, Columbia, Missouri, USA

## CONTENTS

*Introduction*  
*Modeling different cognitive processes with different formalisms*  
*Integrating connectionist and symbolic architectures*  
*Tightly coupled architectures*  
*Completely integrated architectures*

*Loosely coupled architectures*  
*Localist implementations of rule-based reasoning*  
*Distributed implementations of rule-based reasoning*  
*Extraction of symbolic knowledge from connectionist models*  
*Summary*

*We may incorporate symbolic processing capabilities in connectionist models, including implementing such capabilities in conventional connectionist models and/or adding additional mechanisms to connectionist models.*

important events have brought to light ideas, issues, trends, controversies, and syntheses in this area. In this article, we will undertake a brief examination of this area, including rationales for such models and different ways of constructing them.

## INTRODUCTION

Many cognitive models have incorporated both symbolic and connectionist processing in one architecture, apparently going against the conventional wisdom of seeking uniformity and parsimony of mechanisms. It has been argued by many that hybrid connectionist–symbolic systems constitute a promising approach to developing more robust and powerful systems for modeling cognitive processes and for building practical intelligent systems. Interest in hybrid models has been slowly but steadily growing. Some important techniques have been proposed and developed. Several

## MODELING DIFFERENT COGNITIVE PROCESSES WITH DIFFERENT FORMALISMS

The basic rationale for research on hybrid systems can be succinctly summarized as ‘using the right tool for the right job’. More specifically, we observe that cognitive processes are not homogeneous: a wide variety of representations and processes seem to be employed, playing different roles and serving different purposes. Some cognitive processes and representations are best captured by symbolic models, others by connectionist

models (Dreyfus and Dreyfus, 1987; Smolensky, 1988; Sun, 1995). Therefore, in cognitive science, there is a need for 'pluralism' in modeling human cognitive processes. Such a need leads naturally to the development of hybrid systems, in order to provide the necessary computational tools and conceptual frameworks. For instance, to capture the full range of skill-learning capabilities, a cognitive architecture needs to incorporate both declarative and procedural knowledge. Such an architecture can be implemented computationally by a combination of symbolic models (which capture declarative knowledge) and connectionist models (which capture procedural knowledge). The development of intelligent systems for industrial applications can also benefit greatly from a proper combination of different techniques, because currently no one technique can do everything successfully. This is the case in many application domains.

The relative advantages of connectionist and symbolic models have been argued at length. (See, for example, Dreyfus and Dreyfus, 1987; Smolensky, 1988 and Sun, 1995 for various views.) The advantages of connectionist models include: massive parallelism; graded representation; learning capabilities; and fault tolerance. The advantages of symbolic models include: crisp representation and processing; ease of specifying detailed processing steps; and the resulting precision in processing. With these relative advantages in mind, the combination of connectionist and symbolic models is easy to justify: hybrid systems seek to take advantage of the synergy of the two types of model when they are combined or integrated.

Psychologists have proposed many cognitive dichotomies on the basis of experimental evidence, such as: implicit versus explicit learning; implicit versus explicit memory; automatic versus controlled processing; incidental versus intentional learning. Above all, there is the well-known dichotomy between procedural and declarative knowledge. The evidence for these dichotomies lies in experimental data that elucidate various dissociations and differences in performance under different conditions. Although there is no consensus regarding the details of the dichotomies, there is a consensus on the qualitative difference between two types of cognition. Moreover, most researchers believe in the necessity of incorporating both sides of the dichotomies, because each side serves a unique function and is thus indispensable. Some cognitive architectures have been structured around some of these dichotomies.

Smolensky (1988) proposed a more abstract distinction of conceptual versus subconceptual

processing; and he related the distinction to that between connectionist and symbolic models. Conceptual processing involves knowledge that possesses the following three characteristics: public access; reliability; and formality. This is what symbolic models capture. There are other kinds of cognitive capacities, such as skill and intuition, that are not expressible in linguistic forms and do not share the above characteristics. It has proved futile to try to model such capacities with symbolic models. These capacities should belong to a different level of cognition: the subconceptual level. The subconceptual level is better modeled by connectionist subsymbolic systems, which can overcome some of the problems faced by symbolic systems modeling subconceptual processing. Therefore, the combination of the two types of models can capture a significantly wider range of cognitive capacities. These ideas provide the justification for building complex hybrid cognitive architectures. For detailed accounts of a variety of examples of the synergistic combination of connectionist and symbolic processes, see Dreyfus and Dreyfus, 1987; Sun, 1995; Waltz and Feldman, 1986; and Wermter and Sun, 2000.

## INTEGRATING CONNECTIONIST AND SYMBOLIC ARCHITECTURES

Hybrid models are likely to involve a variety of types of processes and representations, in both learning and performance. Therefore, they will involve multiple heterogeneous mechanisms interacting in complex ways. We need to consider how to structure these different components; in other words, we need to consider architectures. Questions concerning hybrid architectures include:

- Should hybrid architectures be modular or monolithic?
- For modular architectures, should we use different representations in different modules, or the same representations throughout?
- How do we decide whether the representation of a particular part of an architecture should be symbolic, localist, or distributed?
- What are the appropriate representational techniques for bridging the heterogeneity likely in hybrid systems?
- How are representations learned in hybrid systems?
- How do we structure different parts to achieve appropriate results?

Although many interesting models have been proposed, including some that correspond to the cognitive dichotomies outlined above, our understanding of hybrid architectures is still limited. We need to look at the proposed models and analyze

their strengths and weaknesses, to provide a basis for a synthesis of the existing divergent approaches and to provide insight for further advances. Below we will provide a broad categorization of the existing architectures.

Architectures of hybrid models can be divided into 'single-module' and 'multi-module' architectures. Single-module systems can be further divided according to their representation types: symbolic (as in symbolic models); localist (i.e. using one distinct node for representing each concept – see, for example, Lange and Dyer, 1989; Sun, 1992 and Shastri and Ajjanagadde, 1993); and distributed (i.e. using a set of overlapping nodes for representing each concept – see, for example, Pollack, 1990 and Touretzky and Hinton, 1988). Usually, it is easier to incorporate prior knowledge into localist models, since their structures can be made to directly correspond to that of symbolic knowledge. On the other hand, connectionist learning usually leads to distributed representation (as in the case of back-propagation learning). Distributed representation has some useful properties.

Multi-module systems can be divided into 'homogeneous' and 'heterogeneous' systems. Homogeneous systems are similar to the single-module systems discussed above, except that they can contain several replicated copies of the same structure, each of which can be used for processing the same set of inputs, to provide redundancy for various reasons; alternatively, each module (of the same structure) can be specialized for processing inputs of a particular type (of content).

For heterogeneous multi-module systems, several distinctions can be made. First, a distinction can be made in terms of the representations of the constituent modules. There can be different combinations of types of constituent modules: for example, a system may be a combination of localist and distributed modules (as in CONSYDERR, described in (Sun, 1995), or it may be a combination of symbolic and connectionist modules, either localist or distributed (as in CLARION, described in (Sun and Peterson, 1998)).

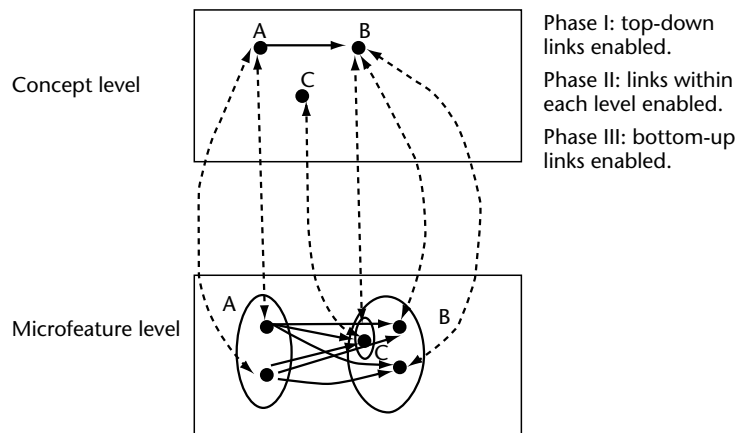
Second, a distinction can be made in terms of the coupling of modules: a set of modules may be loosely or tightly coupled. In loosely coupled architectures modules communicate with each other, primarily through message passing, shared memory locations, or shared files. This allows for some loose forms of cooperation among modules. One form of cooperation is in terms of the type of processing: while one or more modules take care of preprocessing (e.g. transforming input data) or postprocessing (e.g. rectifying output data),

another module focuses on the main part of the task. Preprocessing and postprocessing are commonly done using a neural network, while the main task is accomplished by symbolic methods. Another form of cooperation is through a master-slave relationship: while one module maintains control of the task at hand, it can command other modules to handle some specific aspects of the task. For example, a symbolic expert system, as part of a rule, may invoke a neural network to make a specific classification decision. Yet another form of cooperation is an equal partnership of multiple modules. In this form, the modules (the equal partners) may represent complementary processes; functionally equivalent but structurally and representationally different processes; or differentially specialized and heterogeneously represented 'experts'.

In tightly coupled architectures on the other hand, the constituent modules interact through multiple channels (for example, various possible function calls); or they may even have node-to-node connections between modules (as in CONSYDERR (Sun, 1995) and ACT-R (Anderson and Lebiere, 1988)). As in the case of loosely coupled systems, there are several possible forms of cooperation among modules.

## TIGHTLY COUPLED ARCHITECTURES

Let us examine briefly a tightly coupled, heterogeneous, multi-module architecture: CONSYDERR (Sun, 1995). It consists of a concept level and a microfeature level. The representation is localist at the concept level, with one node for each concept, and distributed at the microfeature level, with an (overlapping) set of nodes for representing each concept. Rules are implemented, at the concept level, using links between nodes representing conditions and nodes representing conclusions, and weighted sums are used for evaluating evidence. Rules are diffusely duplicated at the microfeature level in a way consistent with the meanings of the rules. Rules implemented at the concept level capture explicit and conceptual knowledge that is available to a cognitive agent, and diffused representations of rules at the microfeature level capture (to some extent) associative and embodied knowledge. Figure 1 shows a sketch of the model. There are two-way (gated) connections between corresponding representations at the two different levels; that is, each concept is connected to all the related microfeature nodes, and vice versa. The operation of the model is divided into three phases: the top-down phase, the settling phase, and the bottom-up



**Figure 1.** The CONSYDERR architecture.

phase. In the top-down phase, microfeatures corresponding to activated concepts are themselves activated, enabling similarity-based reasoning at the microfeature level. In the settling phase, rule-based reasoning takes place at each level separately. Finally, in the bottom-up phase, the results of rule-based and similarity-based reasoning at the two levels are combined.

Because of the interaction between the two levels, the architecture is successful in producing, in a massively parallel manner, a number of important patterns of common-sense human reasoning: for example, evidential rule application, similarity matching, mixed rule application and similarity matching, and both top-down and bottom-up inheritance (Sun, 1995).

## COMPLETELY INTEGRATED ARCHITECTURES

An even tighter coupling between symbolic and connectionist processes exists in ACT-R (Anderson and Lebiere, 1998). ACT-R consists of a number of symbolic components, including declarative memory (a set of structured chunks), procedural memory (a set of production rules), and goal stacks. Retrieval in declarative memory is controlled by activations of chunks, which spread in a connectionist fashion and are affected by the past history of activations, similarity-based generalization, and stochasticity. Learning of associations among chunks and selection of procedural knowledge also happen in a connectionist fashion. Thus, the learning and the use of symbolic knowledge are partially controlled by connectionist processes. Through this tight integration of the two types of process, ACT-R has been successful in modeling

human learning in areas such as arithmetic, analogy, scientific discovery, and human-computer interaction.

## LOOSELY COUPLED ARCHITECTURES

Loosely coupled multi-module architectures, unlike the tightly coupled models discussed above, involve only loose and occasional interaction among components. For example, CLARION (Sun and Peterson, 1998), a model for capturing human skill learning, consists of two levels: a symbolic rule level and a connectionist network level. The two levels work rather independently, but their outcomes are combined in decision-making. The network level consists of back-propagation networks, which work through spreading activation and learn by reinforcement. The rule level works according to symbolic rules, which are learned by extracting information from the network level. Through the loose, outcome-based interaction of the two types of processes, the system is able to model a variety of types of human skill learning.

## LOCALIST IMPLEMENTATIONS OF RULE-BASED REASONING

Among single-module or homogeneous multi-module models, localist implementations of symbolic processes, especially rule-based reasoning, stand out as an interesting compromise between connectionist networks and purely symbolic models. The representational techniques described below are shared by a number of localist models of rule-based reasoning (see, e.g. Lange and Dyer, 1989; Sun, 1992 and Shastri and Ajanagadde, 1993).

The simplest way of mapping the structure of a rule set into that of a connectionist network is by associating each concept in the rule set with an individual node in the network, and implementing a rule by connecting each node representing a concept in the condition of the rule to each node representing a concept in the conclusion of the rule. The weights and activation functions can be set to carry out binary logic or fuzzy evidential reasoning.

To express relations, especially relations between large numbers of variables, we need to introduce variables into rules in connectionist implementations. We can represent each variable in a rule as a separate node. We assign values to these variable nodes dynamically and pass values from one variable node to another, based on links that represent variable binding constraints. Such values can be simple numerical signs (Lange and Dyer, 1989; Sun, 1992) or activation phases (Shastri and Ajjanagadde, 1993).

For example, in first-order predicate logic, each argument of a predicate is allocated a node as its representation; a value is assigned to represent an object (i.e., a constant in first-order logic) and thus is a sign of the object. This sign can be propagated from one node to other nodes, when the object that the sign represents is being bound to other variables from an application of a rule.

For each predicate in the rule set, an assembly of nodes is constructed. The assembly contains  $k + 1$  nodes if the corresponding predicate contains  $k$  arguments. We link up assemblies in accordance with rules. With this network, we can perform forward-chaining inference. We first activate the assemblies that represent known facts; then activations from these assemblies will propagate to other assemblies to which they are connected. This propagation can continue to further assemblies. For backward chaining, we first try to match the hypothesis with conclusions of existing rules; if a match is found, then we use the conditions of the matching rule as our new hypotheses to be proved: if these new hypotheses can be proved, the original hypothesis is also proved. To implement backward chaining in assemblies, we need, in addition to a predicate node, another node for indicating whether the predicate node is being considered as a hypothesis. Backward flow of activation through hypothesis nodes leads to backward-chaining inference.

Why should we use connectionist models (especially localist ones) for symbolic processing, instead of symbolic models? There are two reasons in particular why researchers explore such models. First, connectionist models are believed to be a more apt

framework for capturing many (or even all) cognitive processes (Waltz and Feldman, 1986). The inherent processing characteristics of connectionist models often make them more suitable for cognitive modelling. Second, learning may be more easily incorporated into connectionist models than symbolic models: using, for example, gradient descent and its various approximations, expectation maximization, or the Baun–Welch algorithm. This is especially true of distributed models, but is also true of localist ones to some extent.

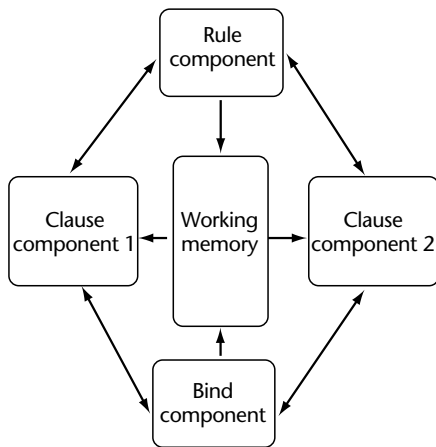
## DISTRIBUTED IMPLEMENTATIONS OF RULE-BASED REASONING

A stronger notion of integration emphasizes developing symbolic processing capabilities in truly connectionist models, rather than juxtaposing symbolic codes with neural networks, or adopting a compromise as in localist implementations. This approach is more parsimonious explanatorily and thus potentially a more interesting form of cognitive modelling if it can be properly developed. Hence there is considerable interest in symbolic processing capabilities of distributed (or ‘true’) connectionist models.

An early example is Touretzky and Hinton’s (1988) DCPS, which implements a production system in connectionist models. There is a working memory, which stores initially known facts and derived facts; there are two clause components, each of which is used to match one of the two conditions of a rule (each rule is restricted to have two conditions); there is a rule component, which is used to execute the action of a matching rule in the working memory; and a bind component is used to enforce constraints that may exist in a rule regarding variables. Each component is a connectionist network. See Figure 2.

The working memory consists of a large number of nodes, each of which has a randomly assigned ‘receptive field’. A *triple* (a fact) is stored in the working memory by activating all the nodes that include the triple in their receptive fields. Many such triples can be stored in the working memory. The two clause components are used to ‘pull out’ two triples that can match two conditions of a rule. That is, they are used to match triples (in the working memory) with rules (in the rule component). Each node in working memory is connected to a corresponding node in each clause component. A clause component is a kind of ‘winner takes all’ network.

The rule component is made up of mutually inhibiting clusters. It is also a kind of ‘winner



**Figure 2.** The overall structure of a connectionist production system.

takes all' network. Each rule is represented in the rule component by a cluster of identical nodes. The connections from the rule component to the clause components are used to help to pull out the triples that match a rule. In turn, these pulled-out triples also help a particular rule to win in the rule component. After successfully matching a rule with two triples in working memory, actions of the rule are carried out by the gated connections from rule nodes (in the rule component) to nodes in working memory. If the action of the rule includes adding a triple, then the gated connections will excite those nodes in the working memory that represent the triple; if the action includes deleting a triple, then the gated connections will inhibit those nodes in the working memory that represent the triple to be deleted. Overall, it is a complex system designed specifically to implement a limited production system.

## EXTRACTION OF SYMBOLIC KNOWLEDGE FROM CONNECTIONIST MODELS

Many hybrid models involve extracting symbolic knowledge, especially rules, from trained connectionist networks. For example, some researchers proposed a search-based algorithm to extract conjunctive rules from networks trained with back-propagation (see Fu, 1989 and Wermter and Sun, 2000). To find rules, the algorithm first searches for all the combinations of positive conditions that can lead to a conclusion; then, with a given combination of such positive conditions, the algorithm searches for negative conditions that should be added to guarantee the conclusion. In the case of

three-layered networks, the algorithm can extract two separate sets of rules, one for each layer, and then integrate them by substitution. Other researchers (e.g. Towell and Shavlik, 1993) tried rules of an alternative form, the 'N of M' form: 'If N of the M conditions  $a_1, a_2, \dots, a_M$  are true, then the conclusion  $b$  is true.' (It is believed that some rules can be better expressed in such a form, which more closely resembles the weighted-sum computation in connectionist networks, in order to avoid the combinatorial explosion and to discern structures.) A four-step procedure is used to extract such rules, by first grouping similarly weighted links and eliminating insignificant groups, and then forming rules with the remaining groups.

These early rule extraction algorithms are meant to be applied at the end of the training of a network. Once extracted, the rules are fixed; there is no modification 'on the fly', unless the rules are completely extracted again after further training of the network. In some more recent systems, rules can be extracted and modified dynamically. Connectionist learning and rule learning can work together, simultaneously. Thus the synergy of the two processes may be utilized to improve learning (Sun and Peterson, 1998). Dynamic modification is also suitable for dealing with changing environments, allowing the addition and removal of rules at any time.

## SUMMARY

Overall, we can discern two approaches toward incorporating symbolic processing capabilities in connectionist models: combining symbolic and connectionist models; and using connectionist models for symbolic processing. In the first approach, the representation and learning techniques from both symbolic processing and neural network models are used to tackle complex problems, including modeling cognition, which involves modeling a variety of cognitive capacities. The second approach is based on the belief that one can perform complex symbolic processing using neural networks alone, with, for example, tensor products, RAAM, or holographic models (see Wermter and Sun, 2000). We may call the first approach 'hybrid connectionism' and the second 'connectionist implementationalism'.

Despite the differences between them, both approaches strive to develop architectures that bring together symbolic and connectionist processes, to achieve a synthesis and synergy of the two paradigms. Many researchers in this area share the belief that connectionist and symbolic methods

can be usefully combined and integrated, and that such integration may lead to significant advances in our understanding of cognition.

## References

- Anderson J and Lebiere C (1998) *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.
- Dreyfus H and Dreyfus S (1987) *Mind Over Machine*. New York, NY: The Free Press.
- Fu L (1989) Integration of neural heuristics into knowledge-based inferences. *Connection Science* 1(3): 240–325.
- Lange T and Dyer M (1989) High-level inferencing in a connectionist network. *Connection Science* 1: 181–217.
- Pollack J (1990) Recursive distributed representation. *Artificial Intelligence* 46(1,2): 77–106.
- Shastri L and Ajjanagadde V (1993) From simple associations to systematic reasoning: a connectionist representation of rules, variables and dynamic bindings. *Behavioral and Brain Sciences* 16(3): 417–494.
- Smolensky P (1988) On the proper treatment of connectionism. *Behavioral and Brain Sciences* 11(1): 1–74.
- Sun R (1992) On variable binding in connectionist networks. *Connection Science* 4(2): 93–124.
- Sun R (1995) Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence* 75(2): 241–295.
- Sun R and Peterson T (1998) Autonomous learning of sequential tasks: experiments and analyses. *IEEE Transactions on Neural Networks* 9(6): 1217–1234.
- Touretzky D and Hinton G (1988) A distributed connectionist production system. *Cognitive Science* 12: 423–466.
- Towell G and Shavlik J (1993) Extracting rules from knowledge-based neural networks. *Machine Learning* 13(1): 71–101.
- Waltz D and Feldman J (eds) (1986) *Connectionist Models and Their Implications*. Norwood, NJ: Ablex.
- Wermter S and Sun R (eds) (2000) *Hybrid Neural Systems*. Heidelberg: Springer.

## Further Reading

- Barnden JA and Pollack JB (eds) (1991) *Advances in Connectionist and Neural Computation Theory*. Hillsdale, NJ: Erlbaum.
- Giles L and Gori M (1998) *Adaptive Processing of Sequences and Data Structures*. New York, NY: Springer.
- Medsker L (1994) *Hybrid Neural Networks and Expert Systems*. Boston, MA: Kluwer.
- Sun R (1994) *Integrating Rules and Connectionism for Robust Commonsense Reasoning*. New York, NY: Wiley.
- Sun R and Alexandre F (eds) (1997) *Connectionist Symbolic Integration*. Hillsdale, NJ: Erlbaum.
- Sun R and Bookman L (eds) (1994) *Architectures Incorporating Neural and Symbolic Processes*. Boston, MA: Kluwer.
- Wermter S, Riloff E and Scheler E (eds) (1996) *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*. Berlin: Springer.

# Consciousness

Introductory article

Adam Zeman, University of Edinburgh, Edinburgh, UK

## CONTENTS

Introduction  
 What do we mean by 'conscious', 'aware', and 'self-conscious'?  
 The science of consciousness

Theories of consciousness  
 The philosophy of consciousness  
 Conclusion

*Consciousness refers both to wakefulness and to the contents of our experience. The subjective aspect of consciousness poses a philosophical problem for objective science.*

## INTRODUCTION

Since the early 1980s there has been a major effort to make better sense of consciousness. The

current fascination with the subject flows from several sources: work by neuroscientists is steadily revealing details of the brain processes which make consciousness possible; psychologists have underlined the existence of a wide range of unconscious brain processes which can be contrasted informatively to conscious ones; computer scientists and engineers are designing sophisticated brain-like systems which can rival human intellectual