

Connectionist Models: Not Just a Notational Variant, Not a Panacea

David L. Waltz

Thinking Machines Corporation

and

Brandeis University

Abstract

Connectionist models inherently include features and exhibit behaviors which are difficult to achieve with traditional logic-based models. Among the more important of such characteristics are 1) the ability to compute nearest match rather than requiring unification or exact match; 2) learning; 3) fault tolerance through the integration of overlapping modules, each of which may be incomplete or fallible, and 4) the possibility of scaling up such systems by many orders of magnitude, to operate more rapidly or to handle much larger problems, or both. However, it is unlikely that connectionist models will be able to learn all of language from experience, because it is unlikely that a full cognitive system could be built via learning from an initially random network; any successful large-scale connectionist learning system will have to be to some degree “genetically” prewired.

1 Prologue

My current research centers on *memory-based reasoning*, a connectionism-informed descendant of associative memory ideas. Memory-based reasoning holds considerable promise, both for cognitive modeling and for applications. In this model, rote memories of episodes play the central role, and schemas are viewed as epiphenomenal. This model is described in considerable detail in [35] and will not be explained here; however, as I have prepared this paper, it has served as the background against which I have critically examined both connectionist and more traditional AI paradigms.

2 Connectionist and Heuristic Search Models

For most of its history, the heuristic search, logic, and “physical symbol system” [19] paradigms have dominated AI. AI was conceived at about the same time that protocol analysis was in vogue in psychology [16]; such protocols could be implemented on the then-new von Neumann machines fairly well. Protocol analysis suggested that people operate by trial and error, using word-like objects as primitive units. AI has stuck almost exclusively with heuristic search and symbol systems, using them in a wide variety of natural language processing models and programming languages, ranging from ATN’s, most other natural language parsing systems, and planning based models (e.g. for pragmatics) to Prolog and Planner [9].

Meanwhile, it seems highly implausible that anything resembling heuristic search is used much below the level of consciousness; certainly no one would believe that a neuron executes heuristic search. The small amount of evidence marshalled to support the hypothesis of subconscious search [15] could be explained in many other ways. Such models as Marcus’ deterministic parser [29] have attempted to move away from heuristic search, yet were cast largely in heuristic search terms ¹.

¹One problem that Marcus’ parser was attempting to solve was the mismatch between psychological data and heuristic search models; garden path sentences were an exception, where backtracking seems an appropriate model. Even there, it seems that to understand garden path sentences, people generally back up and *completely* reprocess sentences, using a “trace” stored in a sort of audio buffer [26].

Connectionist systems have stirred a great deal of excitement for a number of reasons: 1) They're novel. Connectionism seems to be a good candidate for a major new paradigm in a field where there have only been a handful of paradigms (heuristic search; constraint propagation; blackboard systems; marker passing). 2) They have cognitive science potential. While connectionist neural nets are not necessarily analogous to neurons, they do seem brain-like and capable of modeling a substantial range of cognitive phenomena. 3) Connectionist systems exhibit non-trivial learning; they are able to self-organize, given only examples as inputs. 4) Connectionist systems can be made fault-tolerant and error-correcting, degrading gracefully for cases not encountered previously [37]. 5) Appropriate and scalable connectionist hardware is rapidly becoming available. This is important, both for actually testing models, and because the kinds of brain and cognitive models that we build are very heavily dependent on available and imaginable hardware [23] [1]; and 6) Connectionist *architectures* also scale well, in that modules can be interconnected rather easily. This is because messages passed between modules are activation levels, not symbolic messages.

Nonetheless, there are considerable difficulties still ahead for connectionist models. It is probably premature to generalize based on our experience with them to date. So far all systems built have either learned relatively small numbers of items, or they have been toy systems, hand built for some particular task. The kinds of learning shown to date are hardly general. It seems very unlikely to me that it will be possible for a single, large, randomly wired module to learn *everything*. If we want to build a system out of many modules, we must devise an architecture for the system with input and output specifications for modules and/or a plan for interconnecting the internal nodes of different modules. Finally, connectionist models cannot yet argue that they offer a superset of traditional AI operations: certain operations such as variable binding cannot yet be performed efficiently in connectionist networks.

2.1 Best Match vs. Exact Match

It is not possible to specify completely the conditions for any sort of decision—including decisions on natural language understanding and parsing—in a manageable set of logical rules and heuristics. By inserting a sentence in an appropriate context, even extremely rare or unusual structures and interpretations can be made to seem the most natural.

Rule systems can be constructed to handle such cases, but at the expense of requiring arbitrarily large numbers of rules with arbitrarily long sets of conditions. Connectionist models inherently integrate *all* available evidence, most pieces of which will be irrelevant or only weakly relevant for most decisions. Moreover, one does not have to find logically necessary and sufficient conditions; connections between actions and the facts of the world can be represented as statistical correlations. In Feldman's terms [32], connectionist reasoning is *evidential* rather than *logical*.

Reasoning that is *apparently* logical can arise from connectionist models in at least two ways. 1) A programmer can encode individual alternatives for lexical selection, phrase structure, etc. as nodes which compete with or support each other; the processing of a sentence then involves clamping the values of some input word nodes, and allowing the whole network to settle. For "regular" inputs, strong pathways, which "collaborate" in reinforcing each other, can give the appearance of rule-like behavior. Given similar inputs, one can expect similar outputs. Most natural language connectionist work has been rule-like in this sense [37] [3] [33] [31]. 2) The appearance of rule-based behavior can also result from learned connectionist networks or associative memory models. If a system can find the activation pattern or memory which is closest to a given current event or situation, it can exhibit highly regular behavior. Such systems degrade gracefully. Unlike connectionist models, associative memory models can also tell when a new event does not correspond well to any previous event, they can "know that they don't know" [35]. (See also Grossberg [7].)

In contrast, systems based on logic, unification and exact matching are inevitably brittle (i.e. situations even slightly outside the realm of those encoded in the rules fail completely, and the system exhibits discontinuous behavior). We see no way to repair this property of such systems. ²

²See also [21] and [22].

2.2 Match with Psychological Results

Psychological research on categorization [34][25][13][2] has shown that category formation cannot be explained in a classical logical model. That is, the conditions of category membership are not merely logical conditions (result of expressions with connectives ‘and’ ‘or’ and ‘not’). Rather, categories are organized around “focus concepts” or prototypes, and exhibit graceful degradation for examples that differ from the category focus along any of a number of possible dimensions [13]. Connectionist systems seem well-suited for modeling such category structure (though such modeling has not been explored very extensively [11]).

2.3 Massive Parallelism

Restricted natural language is not natural language. One cannot make progress in natural language understanding unless one can run large problems and see the results of experiments in finite time. Small scale experiments (involving on the order of hundreds of nodes or less) are inadequate to really explore the issues in computational linguistics. One needs a model with a realistically large vocabulary and range of possible word senses and interpretations, in order to convincingly argue that the model is appropriate and adequate.

Fortunately, dramatic strides are being made in computer architecture at just the time that connectionist theoretical models are being explored. These fields are not unrelated. Connectionist models [24][4] served as initial inspiration to designers of new generation hardware (e.g. [10]), though many parallel architectural ideas were already being explored in the pursuit of greater speed. This followed the realization that we were approaching asymptotes for speeds possible with serial uniprocessors. I believe that developing appropriate hardware will prove to be the easiest part of building full-scale natural language systems.

2.4 Integration of Modules

Connectionist models allow for much easier integration of modules than is possible with symbolic/heuristic search-based systems. Generally, symbolic systems require either a very simple architecture (e.g. the traditional phonetic—syntactic—semantic—pragmatic bottom-up model of classical linguistics) or a sophisticated communications facility (for example, a blackboard [20]) in order to build a system composed of many modules. In the blackboard model, each module must in general have a *generator* for complex messages as well as an *interpreter* for such messages.

In contrast, connectionist models allow an integration of modules by links that can go directly to the nodes (concepts or microfeatures) that co-vary with the activation patterns of other modules, and messages themselves can be extremely simple (e.g. numerical activation levels, or markers [8]). In some cases, link weights can be generated based on an analysis of the statistical correlations between various concepts or structures; in other cases weights can be generated by learning schemes [27]. Nonetheless, still there is a potentially large set of cases where weights will have to be generated by hand, or by yet-to-be-discovered learning methods. Clearly, every concept cannot be connected to every other directly. (This would require n^2 connections for n concepts, where n is at least 10^6 .) Some solutions have been suggested (e.g. the microfeature ideas in [37]) but none seems easy to program.

2.5 Fault Tolerance

Since a large number of nodes (or modules) have a bearing on a single connectionist decision (e.g. lexical selection or prepositional phrase attachment) then not all of them need to be active in order to make a correct decision; some variation of values can be tolerated. In a modular connectionist model, input information can be fed to syntactic, semantic, and pragmatic modules directly. Thus, an unparsed string of terms can suggest a particular topic area to a pragmatic context module, even without any syntactic processing; such topical context can in turn be used to influence lexical selection. At the same time, the range of possible syntactic structures allows certain lexical assignments and precludes others; semantic information such as case role restrictions likewise can have a bearing on lexical selection (see [37] for further discussion).

2.6 Learning

Learning is one of the most exciting aspects of connectionist models for both the AI and psychology communities. For example, the back propagation error learning [26] and Boltzmann machine [30] methods have proved quite effective for teaching input/output patterns. However, such learning is not a panacea. Some researchers believe that one can start with a very large randomly interconnected and weighted network, and potentially generate a fully intelligent system, simply by presenting it with enough raw sensory inputs and corresponding desired outputs. I doubt it: the learning space corresponding to raw sensory inputs (e.g. visual and audio) is astronomically large, and learning to perceive via feedback (“punishment/reward”?) seems both cognitively and technically unrealistic.

3 Key Problems for Connectionist Language Models

3.1 Learning from “Experience”

As suggested above, learning is both a key achievement of connectionism, and a key open issue for a full cognitive system.

The difficulty for cognitive learning theories of any sort is the observation that perception has to be prior to language. In turn, perception itself seems to require *a priori*, innate organization. Just how large must an innate component be? I believe it will have to account at least for such phenomena as figure/ground organization of scenes, the ability to appropriately segment events, (both to separate them from the experiences that precede and follow them and also to articulate their internal structure); the notion of causality; and general structuring principles for creating memory instances. This suggests to me that a large portion of a learning system must be wired initially, probably into fairly large internally regular modules, which are subject only to rudimentary learning via parameter adjustment. This conclusion follows from the observation that if brains could completely self-organize, this method, being simpler than present reality, would have been discovered first by evolution. My guess is that such total self-organization would require far too long, since it requires exploring vast space of weight assignments. Even given extensive *a priori* structure, humans require some twenty years to mature. I think that we cannot avoid programming cognitive architecture.

3.2 Variable Binding

Some operations that programmers have traditionally taken for granted have proven difficult to map onto connectionist networks. One such key operation is variable binding. Assume that we have devised a good schema representation or learning system, and stored a number of schemas: what happens when a new natural language input triggers a schema and we would like to store this instance in long-term memory? It seems that we need to create an instance of the schema with the particular agents, objects, patients, so on, bound to case roles. It is not obvious how this ought to be done in a connectionist model. Some experiments in designing general connectionist schemes for variable binding have been performed [36], but these methods seem very awkward and expensive in terms of the numbers of nodes and links required to store even a single relation.

Another possibility is to make a copy of the entire schema structure for each new instance, but this seems to lack neurophysiological plausibility. A more appealing direction is suggested both by Minsky [18][17] and Feldman and Shastri [5] [32]: a very large number of nodes are randomly connected to each other such that nodes that have never been used before form a kind of pool of potential binding units for novel combinations of schemas and role fillers. When a new instance is encountered, all the participants which are active can be bound together using one or more of these previously unutilized binding nodes, and those nodes can then be removed from the “free binders pool”.

There are important open questions in any case: for example, are different modules responsible for sentence processing, perceptual processing, short-term memory and long-term memory [6]? If so, how are these interconnected and “controlled”? If not, how can we account for these different processing modes?

3.3 Timing and Judging When Sentence Processing is Complete

Connectionist systems for language processing have assumed that sentences will be preceded and followed by quiescent periods. The resulting pattern of activations on nodes in the system can then be read whenever appropriate, and the time sequence of node activations interpreted as desired (Pollack and I are guilty of this sloppiness). There is a real difficulty in knowing how and when one should interpret the internal operation of a system. Should we wait until activation levels on nodes have settled, i.e. changed less than a certain amount on each cycle? Should we wait for activity to either be completely on or completely off in various nodes? Should we wait a fixed amount of time and then evaluate the network activation pattern? If so, how do we set the clock rate of the relaxation network relative to the rate at which input words arrive? What should be done to the activation pattern of a set of nodes after a sentence has been "understood"? Should the levels be zeroed out? Should they remain active? Under what circumstances and by what methods should items be transferred to (or transformed into) long-term memory? Are the nodes used in understanding the same ones responsible for long-term memory storage or is there some sort of copying or transfer mechanism?

All these questions need crisper answers and principles. It does seem clear that processing must be approximately complete soon after the completion of a sentence so that processing of the next sentence can start, since sentences or clauses can occur with very little separation. This suggests that expectations play an important role in sentence processing and further that really important material ought to appear or be expected well before the end of a sentence if the processing of the next sentence is not to be interfered with.

3.4 Debugging and Understanding Systems

In general, it is difficult to tell exactly what systems with distributed knowledge representations know or don't know. Such systems cannot explain what they know, nor can a person look at their structures and tell whether they are in fact complete and robust or not, except in very simple cases [12]. The only way to test such systems is by giving them examples and judging on the basis of their performance whether they are suitable or not. This problem is a quite serious one for systems that are designed to be fault tolerant. A fault tolerant system, for instance, might usually work quite well, even though one module is seriously defective; however in marginal cases, a counterproductive module could cause performance to be much worse than it ought to be. The problems of debugging a system in which some modules may compensate for and cover up the errors of others seem quite intractable.

3.5 Generating Applications

Natural language processing work has suffered and still suffers from a shortage of good ideas for applications. We don't know quite what we'd do with such systems even if we could successfully build them. In part this is because the actions that a computer can easily carry out are radically different from those that a person can do. In part the difficulty is that typing is a slow and error prone input method; if speech were available, natural language processing might rapidly increase in importance. On the other hand, bulk processing of text databases [28] seems a promising applications area.

It may be impossible to use human-like learning methods for connectionist systems (or for any computer-based language processing system). It may also be undesirable. Unlike people, computers are capable of remembering literally the contents of large text files and complete dictionaries while at the same time they lack perceptual and reasoning facilities. The combination suggests that infant-like learning may not be appropriate for computer-based language systems, even if a brain-like machine can be built.

References

- [1] Backus, J., "Can programming be liberated from the von Neumann style? A functional style and its algebra of programs," (1977 ACM Turing Award Lecture.) *Communications of the ACM* **21** (8), 613-641, August 1978.

- [2] Berlin, B., and Kay, P. *Basic color terms: Their universality and evolution*. Berkeley and Los Angeles: University of California Press, 1969.
- [3] Cottrell, G. W., & Small, S.L. "A connectionist scheme for modelling word sense disambiguation." *Cognition and Brain Theory* **6**, 89-120, 1983.
- [4] Fahlman, S.E., *NETL: A System For Representing and Using Real-World Knowledge*, Cambridge, MA: MIT Press, 1979.
- [5] Feldman, J.A., Ballard, D.H., "Connectionist Models and Their Properties," *Cognitive Science* **6** (3), 205-254, 1982.
- [6] Fodor, J. *The Modularity of Mind*, Cambridge, Massachusetts: MIT Press, 1982.
- [7] Grossberg, S. "Competitive Learning: From Interactive Activation to Adaptive Resonance," to appear in *Cognitive Science* **11** (1).
- [8] Hendler, J. "Integrating Marker-passing and Problem Solving", doctoral dissertation, Brown University, 1986.
- [9] Hewitt, C., "Description and theoretical analysis of PLANNER", doctoral dissertation, Department of Mathematics, MIT, 1972.
- [10] Hillis, D., *The Connection Machine*, Cambridge, MA: MIT Press, 1985.
- [11] Hinton, G., and Anderson, J. (eds.), *Parallel Models of Associative Memory*, Hillsdale: Lawrence Erlbaum Associates, 1981.
- [12] Hinton, G.E., McClelland, J.L., and Rumelhart, D.E. "Distributed Representations," *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
- [13] Lakoff, G. *Women, Fire and Dangerous Things*. Chicago: University of Chicago Press, to appear 1979.
- [14] Marcus, M.P. *A theory of syntactic recognition for natural language*. Cambridge, MA: MIT Press, Cambridge, 1980.
- [15] Marslen-Wilson, W., & Tyler, L.K. "The temporal structure of spoken language understanding". *Cognition* **8**, 1-72, 1980.
- [16] Miller, G.A., Galanter, and Pribram, K., *Plans and the Structure of Behavior*, 1954.
- [17] Minsky, M.L., *The Society of Mind*, Simon & Schuster, 1986 (to appear).
- [18] Minsky, M.L. "K-lines: A theory of memory," *Cognitive Science* **4**, 117-133, 1980.
- [19] Newell, A. "Physical Symbol Systems," *Cognitive Science* **4** (4), 135-183, 1980.
- [20] Nii, H.P. "Blackboard Systems Part Two: Blackboard Application Systems," *AI Magazine* **7** (3), 82-106, 1986.
- [21] Nilsson, N.J. "Artificial Intelligence Prepares for 2001," *AI Magazine* **4** (4), 7-14, 1983.
- [22] Pentland, A.P. & Fischler, M.A. "A More Rational View of Logic or, Up Against the Wall, Logic Imperialists!" *AI Magazine* **4** (4), 15-18, 1983.
- [23] Pylyshyn, Z.W. "Computation and cognition: Issues in the foundations of cognitive science." *The Behavioral and Brain Sciences* **3**, 111-169, 1980.
- [24] Quillian, M.R. "Semantic memory". In M. Minsky (Ed.), *Semantic information processing*. Cambridge, MA: MIT Press, 1968.
- [25] Rosch, E., & Mervis, C. "Family resemblances: Studies in the internal structure of categories". *Cognitive Psychology* **7**, 573-605, 1975.

- [26] Rumelhart, D.E., Hinton, G.E., & Williams, R.J. "Learning internal representations by error propagation," In D.E. Rumelhart, McClelland, J.L. & the PDP research group (eds), *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
- [27] Rumelhart, D.E. and McClelland, J.L. and the PDP Research Group (eds.) *Parallel Distributed Processing: Explorations in the microstructure of cognition*, Volumes 1 & 2. Cambridge, MA: MIT Press, 1986.
- [28] Sabot, G. "Bulk Processing of Text on a Massively Parallel Computer," Tech. Rpt. No. 86-2, Thinking Machines Corporation, 1986.
- [29] Seidenberg, M.S., Tanenhaus, M.K., & Leiman, J.M. "The time course of lexical ambiguity resolution in context," Center for the Study of Reading, Tech. Rpt. No. 164., University of Illinois, Urbana, March, 1980.
- [30] Sejnowski, T.J., & Rosenberg, C.R., "NETtalk: A Parallel Network that Learns to Read Aloud." Tech. Rpt. JHU/EECS-86-01, The Johns Hopkins University, Electrical Engineering and Computer Science, 1986.
- [31] Selman, B. and Hirst, G. "A Rule-Based Connectionist Parsing System". *Proc. of the Conf. of the Cognitive Science Society*, Irvine, CA, 212-221, August 1985.
- [32] Shastri, L. and Feldman J.A. "Evidential Reasoning in Semantic Networks: A Formal Theory". *Proc. IJCAI*, Los Angeles, 465-474, August 1985.
- [33] Small, S. "Word expert parsing: A theory of distributed word-based natural language understanding". Department of Computer Science Tech. Rpt. No. 954. University of Maryland, 1980.
- [34] Smith, E.E., and Medin, D. *Categories and Concepts*. Cambridge, MA: Harvard University Press, 1981.
- [35] Stanfill, C., and Waltz, D.L., "Toward Memory-Based Reasoning," to appear in *Communications of the ACM*, December 1986.
- [36] Touretzky, D.S. "Symbols Among Neurons: Details of a Connectionist Inference Architecture," *Proc. IJCAI*, Los Angeles, 238-243, August 1985.
- [37] Waltz, D.L. and Pollack, J.B. "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation," *Cognitive Science* 9(1), 51-74, 1985.