FORUM ON CONNECTIONISM

Questions about Connectionist Models of Natural Language

Mark Liberman
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

MODERATOR STATEMENT

My role as interlocutor for this ACL Forum on Connectionism is to promote discussion by asking questions and making provocative comments. I will begin by asking some questions that I will attempt to answer myself, in order to define some terms. I will then pose some questions for the panel and the audience to discuss, if they are interested, and I will make a few critical comments on the abstracts submitted by Waltz and Sejnowski, intended to provoke responses from them.

I. What is a "connectionist" model?

The basic metaphor involves a finite set of nodes interconnected by a finite set of directed arcs. Each node transmits on its output arcs some function of what it receives on its input arcs; these transfer functions are usually described parametrically, for instance in terms of a linear combination of the inputs composed with some nonlinear threshold-like function; the transfer function may involve a random variable.

A subset of the nodes (or arcs) are designated as inputs and/or outputs, whose values are supplied or used by the "environment."

"Time" is generally quantized and treated in an idealized way, as if all connections involved a transmission delay exactly equal to the time quantum; this is presumably done for convenience and tractability, since neural systems are not like this. The nodes' transfer function may contain some sort of memory, e.g. an "activation level." The state of the network at time step t determines its state at time step t+1 (at least probabilistically, if random variables are involved); the network calculates its response to a change in its input by executing a sequence of time-steps sufficient to permit information to propagate through the required number of nodes, and to permit the system to attain (at least approximately) a fixed point, that maps back into itself or into a state sufficiently close.

Thus the system as a whole is usually defined so that it will settle into a static configuration for a static input pattern; (models whose dynamics exhibit limit cycles or chaotic sequences are easy to devise, but I am not aware that they have been used).

Connectionist models (at least those with static fixed points) define a relation on their set of input/output node values. Without further constraints on the number of hidden nodes, the nodes' transfer function, etc., the defined relation can obviously be anything at all.

In fact, the circuits of a conventional digital computer can obviously be described in terms that make them "connectionist" in the very general sense given above. The most interesting connectionist models, such as the so-called "neural nets" of Hopfield and Tank, or the "Boltzmann machine," are defined in much more specific ways.

II. How can we categorize and compare the many different types of such models that have been proposed?

The situation is reminiscent of automata theory, where the basic metaphor of finite control, read/write head(s), input and output tape(s) has many different variations. The general theory of connectionist machines seems to be at a relatively early stage, however. Some particular classes of machines have been investigated in detail, but at the level of generality that seems appropriate for this panel, a general mathematical characterization does not exist.

Some crude distinctions seem worth making:

Some models "learn" while others have to be programmed in every detail. This is a gradient distinction, however, since the "learning" models require an appropriate network architecture combined with an appropriate description and presentation of the training material.

Some models represent category-like information diffusely, through ensembles of cooperating nodes and arcs, while others follow the principle of "one concept, one node."

III. Why are (some) connectionist models interesting?

The term "interesting" is obviously a subjective one. The list that follows expresses my own point of view.

1. Connectionist models are vaguely reminiscent of neurological systems. The analogy is extremely loose, at best; neuronal circuits are themselves apparently quite diverse, but they all share properties that are quite different from the connectionist models that are generally discussed. Still, it may be that there are some deep connections in terms of abstract information-processing methods.
2. Connectionist information processing is generally parallel and cooperative, with all calculations completed in a small number of time steps. For certain kinds of algorithms, network size scales gracefully with problem size, with at worst small time penalties.
3. In some cases, learning algorithms exist: training of the network over appropriate input/output patterns causes the network to remember the patterns and/or to "summarize" them according to statistical measures that depend on the network structure and the training method. The trained network "generalizes" to new cases; it generalizes appropriately if the new cases fit the design implicit in the network structure, the training method, and the training data. The same mechanisms also give the system some capacity to complete or correct patterns that are incomplete or partly errorful.

4. Some models (especially those that learn and that represent patterns diffusely) blur distinctions among rule, memory, analogy. There need be no formal or qualitative distinction between a generalization and an exception, or between an exception and a subregularity, or between a literal memory and the output of a calculation. For some cognitive systems (including a number relevant to natural language) this permits us to trade the possibly harmful consequences of giving up on finding deeper generalizations for the immense relief of not looking for perfectly regular rules that aren't there.
5. Some aspects of human psychology can be nicely modeled in connectionist terms -- e.g., semantic priming, the role of spaced practice, frequency and recency effects, non-localized memory, restoration effects, etc.
6. Since connectionist-like networks can be used to build arbitrary filters and other signal-processing systems, it is possible in principle to build connectionist systems that treat signals and symbols in an integrated way. This is a tricky point -- an ordinary general-purpose computer reduces a digital filter and a theorem-prover to calculations in same underlying instruction set, so the putative integration must be at a higher level of the model.

IV. What do connectionist models have to tell us about the structure of infinite sets of strings?

So far, well-defined connectionist models all deal with relations over a finite set of elements; at least, no one seems to have shown how to apply such models systematically to the infinite sets of arbitrarily-long symbol-sequences that form the subject matter of classical automata theory.

Connectionist models can deal with sequences of symbols in at least two ways: the first is to connect the symbol sequence to an ordered set of nodes, and the second is to have the network change state in an appropriate way as successive symbols are presented.

In the first mode, can we do anything that adds to our understanding of the algorithms involved? For instance, it seems straightforward to implement a parallel version of standard context-free parsing algorithms, by laying out a 2D matrix of cells (corresponding to the set of substrings) for each of the nonterminal symbols, imposing connectivity along the rows and up the columns for calculating immediate domination relations, and so on. Can such an architecture be persuaded to learn a grammar from examples? It is limited to sentences of fixed maximum length -- is this enough to make learning possible? Under what circumstances can the resulting "trained" network be extended to longer inputs without retraining? Are there more interesting spatial-layout parsing models?

Many connectionist models are "finite impulse response" machines; that is, the consequences of an input pattern "die out" after the pattern is removed, and the network's propensity to respond to further patterns is left unchanged. If this characteristic is removed, and the network is made to calculate by changing state in response to a sequence of inputs, we can of course imitate classical automata in a connectionist framework. For instance, a push down store can be built out of connectionist piece parts. Can a connectionist approach to processing of sequentially presented information do something more interesting than this? For instance, can the potentially very complex dynamics of of such networks be exploited in a useful way?

V. Comments on Sejnowski

In evaluating Sejnowski's very interesting demonstration of letter-to-sound learning, it is worth keeping a few facts in mind.

First, the success percentages reported are by letter, not by word (according to a personal communication from Sejnowski). Since the average word length was presumably about 7.4 (the average length of the 20000 commonest words in the Brown corpus), the success rate by word of the generalization from the 1000-word set to the 20000-word set must have been approximately .8^7.4, or about 19%. With the "additional training" (presumably training on the same set it was then tested on), the figure of 92% translates to .92^7.4, or about 54% correct by word.

Second, the training did not just present words and their pronunciations, but rather presented words and pronunciations with the correspondences between letters and phonemes indicated in advance. Thus the network does not have to parse and/or interrelate the two symbol sequences, but only keep track of the conditional probability of various possible translations of a given letter, given the surrounding letter sequences. My guess is that a probabilistic n-gram-based transducer, trained in exactly the same way (except that it would only need to see each example once), would outperform Sejnowski's network. Thus the interesting thing about Sejnowski's work is not, I think, the level of performance (which is not competitive with conventional approaches) but some perhaps lifelike aspects of its mode of learning, types of mistakes, etc.

The best conventional letter-to-sound systems rely on a large morph lexicon (Hunnicutt's "DECOMP" from MITalk) or systematic back-formation and other analogical processes operating on a large lexicon of full words (Coker's "nounce" in the current Bell Labs text-to-speech system). Coker's system gives 100% coverage of the dictionary, in principle; more interestingly, it gives better than 99% (by word) coverage of random text, despite the fact that only about 80% of the words are direct hits. In other words, it is quite successful at guessing the pronunciation of words that it doesn't "know" by analogy to those that it does. To take an especially trivial, but very useful, example, it is quite good at decomposing unknown compound words into pairs of known words, with possible regular prefixes and suffixes.

Thus I have a question for Sejnowski: what would be involved in training a connectionist network to perform at the level of Coker's system? This is a case that should be well adapted to the connectionist approach -- after all, we are dealing with a relation over a finite set, training material is easily available, and Coker's success proves that the method of generalizing by analogy to a large knowledge base works well. Given this situation, is the poor performance of Sejnowski's network due only to its small size? Or was it set up in a way that prevents it from learning some relevant morphographemic generalizations?

VI. Comments on Waltz

Waltz is very enthusiastic about the connectionist future. I agree that the possibilities are exciting. However, I think that it is important not to depreciate the future by overselling the present.

In particular, Waltz's statement that Sejnowski's NETtalk "learned the pronunciation rules of English from examples" is a bit of a stretcher -- I would prefer something like "summarized lists of contextual letter-to-phoneme correspondences, and generalized from them to pronounce about 20% of new words correctly, with many of its mistakes being psychologically plausible ones."

Waltz comments that connectionist models "promise to make the integration of syntactic, semantic, pragmatic and memory models simpler and more transparent." The four-way categorization of syntax, semantics, pragmatics, and memory strikes me as an odd way of dividing the world up; but I agree with what I take to be Waltz's main point. A little later he observes that "connectionist learning models... have demonstrated surprising power in learning concepts from example.." I'm not sure how surprising the accomplishments to date have been, but I agree that the possibilities are very exciting. What are the prospects for putting the "integrated processing" opportunities together with the "learning" opportunities?

If we restrict our attention to text input rather than speech input, then the most interesting issues in natural language processing, in my opinion, have to do with systems that could infer at least the lexical aspects of linguistic form and meaning from examples, not just for a toy example or two, but in a way that would converge on a plausible result for a major fraction of a language. Here, few of the basic questions seem to have answers. In fact, from what I have seen of the literature in this area, many of the questions remain unposed.

Here are a few of the questions that come to mind in relation to such a project. What would such a system have to learn? What kind of inputs would it need to learn it, given what sort of initial expectations, represented how? How much can be learned without knowledge of non-linguistic aspects of meaning? How much of such knowledge can be learned from essentially linguistic experience? Are current connectionist learning algorithms adequate in principle? How big would the network have to be? Is a non-toy version of such a system computationally tractable today, assuming it would work in principle? If only toy versions are tractable, can anything be proved about how the system would scale?