A critique of the connectionist hypothesis that recognition uses templates, and not rules

Kurt VanLehn[*]

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Connectionist models of cognition feature a network of nodes, whose topology is assumed to be relatively permanent. Computation (i.e., thinking) is represented by fluctuations of the activation levels of nodes and by transmission of excitation and inhibition along connections. More elaborate formulations equip nodes with small state registers instead of activations, and connections pass small messages instead of an excitatory or inhibitory quantities. The main architectural principles are (1) information transmission along connections happens in parallel, (2) there is little, if any, global control (i.e., no central processor), and most importantly, (3) a cognitive model may use as many nodes and connections as it needs, but there are severe limitations on the amount of information stored in nodes or transmitted by connections.

Historically, connectionism is analogous to the production system movement. Both schools are revisions of earlier movements. Both schools rose to recent prominence in psychology when extraordinarily good pieces of research were done within their respective paradigms. Newell and Simon's (1972) study of problem solving kicked off the production system movement. Studies by Rumelhart and his colleagues of reading, typing and speech kicked off connectionism (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982; Rumelhart & Norman, 1982; Elman & McClelland, 1983). Both production systems and connection systems have attracted the help of computer scientists interested in them for non-psychological reasons. Connection architectures like Fahlman's NETL or Hinton's Boltzmann Machine (Fahlman, Hinton & Sejnowski, 1983) are the analogs of production system languages like OPS and ACT. Unlike production systems, connection systems have attracted hardware designers who are building massively parallel computers for rapid execution of connection systems. Connection systems are as hot today, or even hotter, than production systems were a decade ago.

If the analogy between production systems and connection systems can be trusted, psychology will soon enjoy the fruit of a new formalism. It is good to have a wealth of technical notations and distinctions. Although today's cognitive scientist may not like production systems, she or he still knows what the left-hand side of a rule is, and how important conflict resolutions strategies are. Such widely-shared conceptual tools enrich and empower the field by making it easier to communicate complex ideas. Perhaps they even make it easier to generate those ideas in the first place.

[*] This work was supported by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research, under contract number N00014-82C-0067, contract authority number NR 667-477. Thanks to David Christman, Danny Bobrow and Johan de Kleer.

However, the analogy issues a warning as well as heralding a benefit. Although the research that kicked off the production system movement was outstanding psychology, some later works claimed psychological validity *solely because they used production systems to express their models.* Since one can easily express absurd cognitive models in production systems, psychologists must do much more than notate their models as a production system before they are entitled to even suggest that the model is psychologically plausible. I sincerely hope that this methodological error will not plague connectionism. A connectionist model is not a psychological plausible model just because it uses the same connection system that, say, McClelland and Rumelhart used. Even if one wired up the model with squid neurons, there would be no reason to believe it had anything to do with the mind. One can write rubbish in any representation language. It takes hard work to uncover the principles that are fundamental to a particular model, and even more hard work to show that those principles are psychologically valid. This extra work, which is over and beyond the work needed to implement the model as a connection or production system, is just exactly what yields theory (VanLehn, Brown & Greeno, 1982). Without it, one has just another program that behaves with an amusing, superficial similarity to humans. It has no more scientific merit than the "robots" hired by shopping malls.

Enough methodology! Let's move on to substantive psychological issues.

Connectionism makes an important hypothesis: For some tasks, the best models are those that achieve a rule-like behavior without rules by using a large, finite store of templates. Perhaps the most impressive demonstration of this hypothesis is Rumelhart and McClelland's interactive activation model of word recognition. It has a store of the 1179 most common four-letter English words, and it has no orthographic or phonological rules. Yet it accounts for a host of rule-like human behavior.

The experimental task goes as follows: The subject is shown a four letter string for a short time, then it is replaced by a mask (e.g., a string of "#" signs). The subject is tested on a single letter in the stimulus, using a forced choice between two letters. The subject guesses which letter occurred in that position. Three main effects are observed in such experiments. *Word advantage:* When the stimuli are English words, the subjects' guesses are correct about 17% more often than they are if the stimuli are non-words such as QXRL or ACUU. *Pseudoword advantage:* When the stimuli are pseudowords (i.e., orthographically regular, such as MAVE or SPET, but not English words), the subjects' guesses are correct about 15% more often than they are with non-words. *Wordlike consonant strings advantage:* When the stimuli are consonant strings (and hence orthographically and phonologically irregular) that are constructed by replacing a word's vowel with a consonant (e.g., SPAT becomes SPCT), then subjects' guesses are about 15% more accurate than with non-word stimuli. This third finding tends to refute any theory of word recognition based on stored orthographic or phonological rules.

To account for these three findings, the interactive activation model stores all common four-letter English words. This is the key feature. One can get adequate empirical accuracy, I contend, without a connection system as long as there is a word store and it is used in certain ways. That is, the credit for explaining the main effects belongs to the hypothesis that people recognize words with templates instead of rules. The success of the explanation does not depend on the representation language, which is good. To demonstrate this point, a simplified version of the McClelland/Rumelhart model is presented. Let the function Friends(S,L,P,I) return the set of all words in the store that share I letters with the stimulus S and have letter L at position P. Of course, $1 \leq I \leq 4$ and $1 \leq P \leq 4$. If the stimulus were the pseudoword "MAVE", then

Friends("MAVE","H",1,3) = {"HAVE"}
Friends("MAVE","M",1,4) = {}
Friends("MAVE","A",2,3) = {"HAVE","SAVE","MALE",...}

Given this function to access the word store, the percentage of correct guesses is predicted using the following formuli:

$$\text{Activation}(S,L,P) = \sum_{i} a_i |\text{Friends}(S,L,P,i)|$$

$$\%\text{Correct}(S,P) = \frac{\text{Activation}(S,P,R)}{\text{Activation}(S,P,R) + \text{Activation}(S,P,W)}$$

where R is the right letter choice, W is the wrong one, the a_i are task parameters, and $|X|$ is the cardinality of set X.

Let's see how this model behaves with each of the stimuli kinds. If the stimulus S is a non-word, then both R and W will have few 2-, 3- or 4-letter friends (i.e., $\text{Friends}(S,R,P,I) \approx \text{Friends}(S,W,P,I) \approx \{\}$ for all $I \neq 1$, for all P). They will both have many 1-letter friends. So $\text{Activation}(S,P,R) \approx \text{Activation}(S,P,W)$, and %Correct is roughly 50%. If S is a word, then R will have exactly one 4-letter friend (i.e., S) and W won't have any. Because W is chosen by the experimenter so that it forms a word when substituted into S, W has exactly one 3-letter friend. On the other hand, R usually has many 3-letter friends. Similarly, R will generally have more 2-letter friends than W. Since $|\text{Friends}(S,R,P,I)| > |\text{Friends}(S,W,P,I)|$ for all $I>1$, $\text{Activation}(S,R,P) > \text{Activation}(S,W,P)$, and hence %Correct is greater than 50%. The case for pseudowords and wordlike consonant strings is just like the case for words, except that R will have no 4-letter friends. Hence, the %Correct will be a tad lower than the %Correct for words, but it is still greater than the 50% correct of nonwords. These predictions are qualitatively similar to the main findings. To get quantitative accuracy would require fitting the a_i parameters. Parameter a_4 controls the relative advantage of words over pseudowords and wordlike consonants. Parameters a_3 and a_2 control the advantage of pseudowords and wordlike consonants over nonwords. Interestingly, if subjects are not instructed to expect pseudoword stimuli, then the pseudoword advantage disappears. Under these conditions, $a_3 = a_2 = 0$.

The above model is a simplification of the one actually used by Rumelhart and McClelland. To compete with theirs, it would need an input/output model wrapped around it in order to account for phenomena involving the duration and image quality of the stimuli, the kind of masking, serial position effects, and so on. The interactive activation model can explain some of these effects, although some extra, non-connectionist mechanisms must added (e.g., a clock and a gated response buffer) in order to do so. Extensibility of a model is important, but it is not as important as accounting for the main effects. I take it that Rumelhart and McClelland have convincingly demonstrated that the main findings are best explained by the hypothesis of word storage rather orthographic or phonological rule storage. Moreover, it doesn't matter whether one expresses the hypothesis with connections, as in the interactive activation model, or with a simple additive model, as above.

To return to the production/connection analogy, this "templates, not rules" hypothesis is analogous to Newell's problem space hypothesis (Newell, 1980). Neither of the two hypotheses mention the representation language (which is good), both are quite general (which is excellent), and both can be tested in specific cases (which is best of all). As it turns out, both are controversial, and that's good too. Contention over hypotheses like these will advance cognitive science, but fights about connections versus rules are mere religious squabbles.

In the interest of controversy, I'll try to indicate some problems that the templates-not-rules hypothesis might have in accounting for other kinds of recognition than word recognition. I undertake this with some reluctance. I prefer to evaluate specific hypotheses against specific empirical facts. However, I can find no problems with the way that the templates-not-rules hypothesis accounts for the word recognition phenomena, so I have no alternative but to attack it with general observations.

First, a few quick shots. Any straightforward realization of the templates-not-rules hypothesis means that the recognizer outputs templates from its finite store. A word recognizer outputs words. A word sense disambiguator outputs word senses. But there are plenty of cognitive domains where there isn't a finite set of classes to recognize. Take natural language understanding. If there were a finite number of things that an utterance could mean, then they could be the templates in the store, and the interactive activation model might be a perfectly adequate explanation of natural language understanding. But clearly there are not a finite number of meanings. It's likely that meanings, and maybe even utterances as well, are not countable. Where does one stop and another begin? This suggests that the templates-not-rules hypothesis has significant trouble with domains that don't admit of finite classifications.

Here's another quick shot. A big advantage of template systems is that new knowledge is easy to acquire. To learn a new word, one just adds it to the word store. That's a bit too simple, of course, because it is very rare for a recognition stimulus to *exactly* match any of the training instances. There is a certain amount of abstraction, differentiation and noise removal that has to go on in getting from a training example to a recognition stimulus. A rule-based system does most of this work during learning. A system that stores training instances does most of the work during recognition. Which do people do? A general observation about human behavior is that learning something is much harder than recognizing it. It might take several seconds of rehearsal to add a new word to one's vocabulary, but once that word is learned, recognition takes mere milliseconds. This observation tends to refute the templates-not-rules hypothesis.

I'll finish with a longer criticism of the hypothesis, which may ultimately be more interesting to computer scientists than to psychologists. When Elman and McClelland (1983) applied the interactive activation model to speech perception, they had a little problem, which turns out to be symptomatic of a nearly fatal flaw in the connectionist approach. They assumed that the template store held words, represented as sequences of phonemes. They assumed that the input (stimulus) was a mixture of phonemes, where the strength of each phoneme varied with each tick of the clock. (Actually, they used phonological features as input, but that's irrelevant to the current account.) The model had two difficulties. (1) It depended too much on finding a clear occurrence of the initial phoneme of a word in the input stream. (2) The model had difficulty recognizing words spoken more slowly or more rapidly than usual. Both difficulties can be traced to the same underlying problem: the word's phonemes must be brought into registration with the phonemes in the input sequence. For the word recognition task described above, registration is not a problem. There are always exactly four stimulus letters, which can be matched in only one way against the stored four-letter words. If instead the stimulus were, say, a 14 letter string with a four letter word buried somewhere inside it, then there would be 10 possible ways to match the input with a stored word. This would be a 10-way ambiguous registration problem. The speech recognition task has this registration problem, and more. It has a second source of ambiguity because the phonemes are not guaranteed to be certain cannonical lengths, nor must they be adjacent. If a word is spoken slowly, noises may intervene. These are not problems for just the Elman/McClelland model. They are inherent in the speech recognition task. All speech recognizers must deal with them.

In fact, all recognizers of any kind must deal with the registration problem. In vision, for instance, it is not enough to store an image of an object. The system must be able to recognize the object under translation, rotation, scaling and possibly other kinds of transformation. Perhaps the clearest cases of the registration problem occur in non-metric tasks, such as story recognition. Suppose a stimulus story has N actors, and an old, stored story has M actors. There are N^M possible ways to map stimulus actors to the stored actors. The number of possible registrations decreases if one adds the constraint that two stimulus actors can't fill the same role in the stored story. The number of registrations decreases even more in tasks with more constrained topologies, such as reading, speech or vision. As illustrated a moment ago, there are only $N - M$ registrations to check in order to find an M-long word in an N-long stimulus string.

Not only is the registration problem universal, its intransigent. The typical recognition procedure for a serial machine has two nested loops:

```
(For each template in the store do
   (For each legal registration of the parts of the template
       with the parts of the stimulus do
       (... some matching of the template to the stimulus
           under the mapping of the registration ...)))
```

Connection machines basically eliminate the outer loop by doing all instantiations of its body in parallel. That is, each template is associated with a distinct group of nodes. The stimulus is broadcast all at once to each group, which then trys to recognize its template in the stimulus. Each node group has the connectionist equivalent of the inner loop, which does registration. If the registration problem is trivial, then the node group can be small. For instance, each group consists of a single node in the Rumelhart/McClelland model because there is only one way to register a four-letter stimulus against a four-letter template. On the other hand, if the registration problem is complex, then node groups are large. For instance, the general case of registering an N-part stimulus to an M-part template could use a tree of nodes that is M levels deep with a uniform branching factor of N.* There are $O(N^M)$ nodes per node group. Replacing the inner loop by a parallel scheme merely replaces time complexity by space complexity. The registration problem is inherently complex as well as universal.

A common approach to coping with the registration problem is to reduce M, the number of parts in the template to be matched. Instead of M parts, a template has J subtemplates as its parts where J<<M, and each subtemplate has its J parts, which in turn may have parts, and so on. An old, flat template becomes a part-whole hierarchy. As it stands, this doesn't reduce the combinatorics of registration. Consider a flat template with 6 parts. When it matches an input with N objects, their are N^6 registrations to check. Suppose the new template has two parts, A and B, each of which is a three-part subtemplate. There are N^3 possible matches for A, and N^3 matches for B. If there are no constraints on A and B that are independent of the main template, then the main template has to check all N^3 bindings for A against all N^3 bindings for B. Since $N^3N^3 = N^6$, the registration problem in no easier. Simply dividing flat templates into subtemplates doesn't reduce the combinatorics at all. Combinatorics only begin to decrease when constraints can be added at the subtemplate level. Sharing subtemplates among templates also helps.

* The details: Each node has N descendents. The root node passes a message to its first daughter that pairs the first template part to the first of the N stimulus parts. The second daughter gets a message pairing the first template part to the second stimulus part, and so on for all N daughters. So the first level (= root node) takes care of pairing the first template part to each of the N stimulus parts. The second level (= the daughters of the root node) pairs off the second template part in similar fashion. In order to bind all M template parts, the tree has M levels. So it has $(N^{M+1} - 1)/(N - 1)$ nodes.

Here is a template-subtemplate hierarchy that shares subtemplates and has lots of constraints among subtemplates. I think you will find the notation familiar.

```
S      →  NP  VP
NP     →  Determiner  NBAR
NBAR   →  Adjective*  Noun
VP     →  Auxiliary  VBAR
VBAR   →  Verb  (NP)  (NP)  PP*
PP     →  Preposition  NP
```

A templates-not-rules recognizer that "optimizes" its performance by adopting a part-whole hierarchy with constraints is no longer a templates-not-rules system. It is a rule-based system.

The conclusion is that templates-not-rules systems are infeasible for any recognition problems that require non-trivial, non-metric registration. Neither serial computers nor connection machines can run them fast enough. On the other hand, rule-based systems are feasible.

A psychologist can draw one of two conclusions from this. Either, (1) people are subject to the same "laws of information processing" as machines, therefore they must use rule-based recognizers, and therefore the templates-not-rules hypothesis is generally false, or (2) people have templates-not-rules recognizers, but they run them (so to speak) on some as yet undiscovered information processing architecture that somehow solves registration problems very quickly.

References

Elman, J.L. & McClelland, J.L. Speech perception as a cognitive process: The interactive activation model. To appear in N.Lass (Ed.) *Speech and Language, Vol. 10.* New York: Academic Press. Currently available as ICS Report No 8302, University of California, San Diego, 1983.

Fahlman, S.E., Hinton, G.E. & Sejnowski, T.J. Massively parallel architectures for AI: NETL, Thistle, and Boltzmann Machines. In *Proceedings of 1984 AAAI Conference*, Los Altos, CA: Kaufman, 1983.

McClelland, J.L. & Rumelhart, D.E. An interactive activation model of context effects in letter perception: Part1. An account of basic findings. *Psychological Review*, 1981, *88*, 375–407.

Newell, A. Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.) *Attention and Performance VIII*, New York: Erlbaum, 1980.

Newell A. & Simon H.A., *Human Problem Solving*, Englewood Cliffs, New Jersey: Prentice Hall, 1972.

Rumelhart, D.E. & McClelland, J.L. An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review, 89*, 60–94.

Rumelhart, D.E. & Norman, D.A. Simulating a skilled typist; A study of skilled cognitive-motor performance. *Cognitive Science*, 1982, *6*, 1–36.

VanLehn, K., Brown, J.S. & Greeno, J.G. Competitive argumentation in computational theories of cognition. In W. Kinsch, J. Miller & P. Polson (Eds.) *Methods and Tactics in Cognitive Science.* New York: Erlbaum, forthcoming. Currently available as Xerox Parc report CIS-14, Palo Alto, CA, 1982.