Dynamic Construction of Mental Models in Connectionist Networks

Venkat Ajjanagadde*
Wilhelm-Schickard-Institut
Universität Tübingen
Sand 13, W-7400 Tübingen, Germany
email: nnsaj01@mailserv.zdv.uni-tuebingen.de

Abstract

The task of "model construction", which is the one of constructing a detailed representation of a situation based on some clue, forms an important component of a number of cognitive activities. This paper addresses the problem of dynamic model construction from a connectionist perspective. It discusses how to represent models as patterns of activity within a connectionist network, and how dynamic generation of such patterns can be efficiently achieved.

1 Introduction

The subject matter of this paper is what we refer to as the task of "model construction", which forms an important component of a number of cognitive activities. Informally, the task of "model construction" is the one of *constructing a detailed representation of a situation based on some clue*. Consider the problem of language understanding for example. Suppose we are told that "John drove to the supermarket". "Understanding" this sentence involves infering many more things other than what is explicitly stated in the sentence. For example, we would have inferred that "John *went* to the supermarket" using our knowledge that "driving to a place" implies "going to that place". We would have also done "plan recognition" (Charniak & McDermott, 1985), i.e., we would have inferred the most likely reason behind John's supermarket visit (such as "to shop there", "to work there" etc.). We would also have inferred the sequence of actions taking place such as "John reached the supermarket", "He parked his car in the supermarket parking lot", "He got out of the car and grabbed a shopping cart" and so on... In this example, the sentence explicitly provided only a *clue* about the situation; all that it explicitly stated was just *drive-to(john, supermarket1)*. Based on this clue, we inferred many more facts about the situation thereby constructing a detailed representation, or a model, of the situation corresponding to John's supermarket visit.

In the case of other perceptual tasks, the problem is similar. In vision for example, the 2D image on the retina provides the clue about the situation in the world; based on that clue, the problem is that of constructing the representation of object configuration in the 3D world that would give rise to that image.

In addition to perception, "model construction" plays a role in other cognitive tasks as well. For example, Mannes and Kintsch (1991) argue that a number of mundane planning problems are problems of "understanding". But, then, "understanding" in turn is a perceptual task and involves model construction.

It is very natural to expect that other intelligent activities such as solving problems/puzzles, playing games etc. make extensive use of the apparatus that already exist to accomplish perception. Hence, it is not at all surprising that "model construction", which is an important component of perception, has been found to play a major role in tasks such as syllogistic reasoning (Johnson-Laird, 1983) as well.

In this paper, we deal with this all-pervading task of model construction from a connectionist perspective. Specifically, we examine how to represent models in a connectionist network and how efficient, dynamic construction of such models can be achieved. Our treatment is at a general abstract level wherein the details of the individual cognitive activities are suppressed. In their details, there are a number of differences in the model construction process as it takes place in different cognitive activities; thus, for example, low level image processing differs in a number of ways from speech processing. But, when we disregard the details and examine the problem in a rather abstract fashion, there appears a great deal of similarity in the model construction process as it takes place in different cognitive activities such as vision, language processing, problem solving etc. It is

*This work was supported by DFG grant Schr 275/7-1.

at such an abstract level that we treat the problem of model construction in the rest of this paper.

2 Models and their Dynamic Construction

In the previous section, we informally described the task of model construction as the one of constructing a model of a situation based on some clue. In order to proceed, we need to formalize the notions of "model" and "clue".

Our attempt at formalizing the notion of "model" is inspired by the Tarskian semantics of predicate calculus. The idea is to describe the "world" in terms of a set of objects and relations between those objects. Following that scheme, we define a model to be an *explicit* representation of which relations hold between which objects in the "world"[1].

With that definition of "model", the task of model construction can be stated as follows: We are given some of the relation instances that hold in a situation (These constitute the "clue"); the task is to infer all the relation instances that hold in that situation (All those relation instances together constitute the "model").

Thus the task of model construction involves reasoning. What is the nature of that reasoning activity? The answer is that it is an integrated combination of a variety of reasoning that have traditionally been dealt with rather separately within AI. Upon hearing the sentence "John drove to the supermarket", a model constructed in our mind might consist of facts such as "John went to the supermarket", "John used a car", "he would be shopping", etc. Among these, infering that *goto(john,supermarket1)* is an instance of *deductive reasoning* since *drive-to(x,y)* necessarily means that *goto(x,y)* for any x and y. Infering that "John used a car" is an instance of *default reasoning* since with a few exceptions it is usually the case that when one says "x drove to y", the vehicle driven happens to be a car. Infering that the purpose behind John's driving to the supermarket must be one of shopping, is an instance of *abductive reasoning* (Charniak & McDermott, 1985). It involves examining the different possible purposes behind one's driving to a place (such as "to work there", "to shop there", "to meet someone there" etc.) and picking the most likely purpose in the given context. Model construction involves performing all these different kinds of inferences. In order to arrive at a system that dynamically constructs models, it *may not be necessary* that we make these distinctions between different kinds of reasoning. It may be possible to arrive at such a system directly (via learning or via designing) without ever thinking about the differences between the various inferences that constitute the overall model construction process. But, it so happens that the system being presented here was not arrived at directly. It began with a deductive, rule-based, backward reasoning system (Ajjanagadde & Shastri, 1989); later on, a deductive forward reasoning system was developed (Ajjanagadde & Shastri, 1991) and some enhancements in reasoning power were achieved (Shastri & Ajjanagadde, 1990). Then, a system which combines forward and backward chaining was developed. The work was extended to deal with evidential rules and facts, negation, and abductive reasoning in (Ajjanagadde, 1991). Since it is difficult to provide all the relevant details of our system for dynamic model construction here, we will take another approach to describing it. It so happens that our system for dynamic model construction has some important resemblances to Rumelhart et al's (1986) connectionist model of schemata. Since a typical reader can be assumed to be familiar with the work reported in (Rumelhart et al., 1986), it appears that a rough outline of our system can be provided by relating it to the work of Rumelhart et al. We will point out some important similarities and differences between Rumelhart et al.'s schema model and our system. It is hoped that this comparative discussion will provide the reader with a rough understanding of the ideas underlying our system. Details about our system can be found in the publications referred to.

[1]To contrast between explicit and implicit representations, let us consider a simple example. Suppose that the objects in the domain are a and b. Now, consider the statement $\forall x \forall y P(x,y)$. This representation *implicitly* represents the relations between the objects in the domain. An equivalent *explicit* representation would consist of the following statements: $P(a,a)$, $P(a,b)$, $P(b,a)$, and $P(b,b)$.

3 Schema Model of Rumelhart et al.

In order to illustrate how schemata can be realized in connectionist networks, Rumelhart et al take as an example the problem of representing knowledge about various kinds of rooms, such as kitchen, bathroom, living room, and bedroom. They select forty microfeatures corresponding to such entities as *sofa, oven, refrigerator, telephone, toilet, television, toaster, bathtub, computer* etc. Corresponding to each microfeature, there exists a node in their network. The nodes corresponding to entities which will be found in the same room have mutual excitatory connections between them; nodes corresponding to entities which are unlikely to be found in the same room have mutual inhibitory connections between them. Thus, for example, *oven* and *refrigerator* are likely to be found in kitchen. So, there will be mutual excitatory connections between the nodes corresponding to *oven* and *refrigerator*. On the other hand, *bathtub* and *television* are unlikely to be found in the same room; so there exist mutual inhibitory connections between them. Now, the idea is that if we clamp some of the nodes corresponding to items present in a room, the network will settle into a state where the nodes corresponding to the other items in that room will be active and the rest of the nodes will be inactive. Thus, for example, if we clamp the nodes corresponding to *oven* and *refrigerator*, then, in the stable state, the nodes corresponding to other items in the kitchen, such as *toaster* will be active and the nodes corresponding to entities which are unlikely to be in the kitchen, such as *bathtub* will be inactive.

One important similarity between the problem addressed in (Rumelhart et al., 1986) and the problem taken up in this paper must be obvious. In (Rumelhart et al., 1986), the input is a specification of some of the items present in a room. Given that input, the network has to determine what other items are likely to be present in that room. In our case, the input is a specification of some of the relation instances present in the "world"; the network has to determine what other relation instances will be present in that "world".

4 Connectionist Network: Encoding

In (Rumelhart et al., 1986), the building blocks of schemas are microfeatures; a schema is represented by representing which features are present in that schema and which features are absent. We take predicates and objects as building blocks of mental models (Microfeatures can be viewed as special cases corresponding to 0-ary predicates.). A model is represented by representing which relation instances hold between which objects. Particularly, the arguments of these relations can be dynamically bound to objects, to represent the relation instances that hold in a model.

Corresponding to every microfeature, there exists a unique node in the network of Rumelhart et al. (1986). Similarly, corresponding to the different "objects" of interest, there exist unique nodes (referred to as *constant nodes* since they correspond to the "constants" of predicate logic.) in our network. Thus, in the example network of Fig. 2, there are unique nodes (shown as circles) corresponding to the "objects" *mary*, *jack*, *hospiz*, and *super-fries*. Now, consider the other building block of models, i.e., predicates. Corresponding to an *n-ary* predicate, the network has *(n+1)* nodes. Thus, in Fig. 1, corresponding to the tertiary predicate P, there are four nodes. The nodes $a1$, $a2$, and $a3$ correspond to the three arguments of the predicate P. We refer to these nodes as *argument nodes* (shown as diamonds in figures). Also, corresponding to every predicate, there exists a *predicate node* (shown as squares in figures).

In (Rumelhart et al., 1986), the relations between the different microfeatures are indicated by having (excitatory/inhibitory) connections between the corresponding nodes. Similarly, the relationships between the different predicates in our system are represented by connecting the nodes corresponding to different predicates. However, in this case we need to represent the correspondences between the arguments of the predicates as well. Suppose that when $P(x, y, z)$ (for arbitrary x, y, and z) is known to be true, this lends some amount of evidence (say, C) for $Q(y, z, x)$ being true. That is, the knowledge we have here is of the form

$$P(x,y,z) \Rightarrow Q(y,z,x) \quad \text{(with likelihood } C\text{)}$$

As per this rule, the first argument of Q is bound to the same individual that binds the second argument of P. This is denoted by connecting the argument node $a2$ to the argument node $a4$ (Fig. 1). The connection between the other argument nodes are similar. Also, the predicate node of P is connected to the predicate node of Q via a link whose weight is C. The weights of the links connecting the argument nodes is not of significance in our current work; all those links can be assumed to be having the same weight w, where w is some positive constant.

It is typical of practical examples that if knowing proposition A to be true lends positive (negative) evidence to another proposition B, then, knowing B to be true lends positive (negative) evidence to A. Fig. 1 depicts the connections representing that $P(x, y, z)$ lends evidence to $Q(y, z, x)$. To represent that $Q(y, z, x)$ lends evidence to $P(x, y, z)$, we need to also have connections in the direction opposite to that shown in Fig. 1. Thus, similar to the bidirectional connections in (Rumelhart et al., 1986), in our network also, there exist bidirectional connections between the nodes corresponding to different predicates. However, in addition to the difference of representing argument correspondences, there is another difference between the connections in our network and those in (Rumelhart et al., 1986). In the latter, the links running in opposite directions between two nodes have the same weight. In our network, this need not be the case. Consider the relationship between the predicates *have-dinner-at* and *eat-at* in Fig. 2. Knowing that x had dinner at y lends a very high evidence to the proposition that "x ate at y" (In fact, this is a certain implication.). On the other hand, though knowing that "x ate at y" lends a positive evidence to the proposition "x had dinner at y", the magnitude of this evidence is not as high as in the previous case. So, the weight of the link from the predicate node of *eat-at* to the predicate node of *have-dinner-at* is smaller than the weight of the link running in the opposite direction.

A feature of our network for which there is no strict conceptual parallel in (Rumelhart et al., 1986) corresponds to that of *background facts*. Background facts are specific facts present in the agent's memory. Examples of such facts may be "Jack is Mary's brother", "Jim is a computer scientist" etc. Such specific facts already present in the agent's memory significantly influence the model constructed in response to an input. For example, consider the processing of the following two sentences:

John went to the supermarket.
Mary went to the supermarket.

These two sentences contain similar information. However, due to the background facts present in the agent's memory, the model constructed in response to the first sentence could be significantly different from the model constructed in response to the second sentence. For example, suppose that the agent knew that "John has run out of groceries" and "Mary is an employee of the supermarket". In that case, the model constructed in response to the first sentence is likely to be the one of John going to the supermarket for shopping there. The model constructed in response to the second sentence is likely to be the one of Mary going to the supermarket to work there.

In the example network of Fig. 2, the encodings of three background facts, namely, *F1: hungry(jack)*, *F2: manager-of(mary,super-fries)* , and *F3: fast-food-shop(super-fries)* are shown (enclosed within hexagonal boxes). Let us skip the details of encoding background facts (Details can be found in (Ajjanagadde, 1991).)[2].

Another set of interconnections in our network for which there are no parallels in (Rumelhart et al., 1986) correspond to the representation of competition between alternative explanatory hypotheses. For example, two of the possible purposes behind one's going to a place are "to work at that place", and "to eat at that place". The competition between these two alternative possibilities is achieved in the network as follows. There is an inhibitory connection from the predicate node of *eat-at* onto the link from the predicate node of *goto* to the predicate node of *work-at* (Fig. 2). Similarly, there is an inhibitory connection from the predicate node of *work-at* onto the link from the predicate node of *goto* to the predicate node of *eat-at*. These inhibitory connections achieve the following winner-take-all kind of effect (Details in (Ajjanagadde, 1991).): If the activity level of the *eat-at* predicate node is higher than that of the *work-at* predicate node, the flow of activity along the link from the *goto* predicate node to the *work-at* predicate node gets cut-off. The reverse would be the situation if the activity level of the *work-at* predicate node is higher than that of the *eat-at* predicate. In effect, this mechanism results in the selection of that hypothesis which acquires maximum evidence.

[2]Actually, the network can be extended to encode background facts about classes of individuals instead of just individuals; space limitation precludes the discussion of that aspect here.

5 Representation of Mental Models as Patterns of Activity

Previous section discussed the encodings present in our network. In this section, we will discuss how models are represented as patterns of activity in this network.

As mentioned earlier, a model is taken to be an explicit representation of the various relation instances holding in the "world". We will first discuss the pattern of activity representing one relation instance. The overall pattern of activity representing the model is a combination of the individual patterns corresponding to the different relation instances that constitute the model.

Instances of relations are represented in the network by dynamically binding the "objects" to the arguments of relations. The dynamic argument bindings are represented using *phase locked oscillations* (Ajjanagadde & Shastri, 1991). Essentially, the idea is to represent the binding of an object to an argument by the synchrony of activation of the node corresponding to the object and the node corresponding to the argument. Thus, the bindings in the fact *drive-to(jack,super-fries)* will be represented as follows: The argument node $a14$ and the node corresponding to the object *jack* will be active in synchrony. Similarly, the argument node $a15$ and the node corresponding to the object *super-fries* will be active in synchrony. The activity level of the *drive-*

to predicate node represents the evidence for the fact *drive-to(jack,super-fries)*.

In order to suggest how models can be represented, the pattern of activity corresponding to the simultaneous representation of three relation instances is shown in Fig. 3.

6 Dynamic Construction of Models

Previous section described how models are represented as patterns of activity in the network. In this section, let us briefly examine how the network constructs a model when the input clue is specified. That is, the process we will be examining is the following: The input proposition(s) (e.g., say, *drive-to(jack,super-fries)*) will be specified to the network by clamping the pattern of activity representing the input proposition(s) onto the network. Now, the network has to generate the patterns of activity corresponding to the other relation instances which hold in the "world". The process is quite similar to the one described in (Rumelhart et al., 1986) wherein some of the nodes in the network are externally clamped and the rest of the nodes in the network settle into appropriate levels of activity. Let us discuss some major differences between the process in (Rumelhart et al., 1986) and in our network.

One main difference is that in our network the additional task of propagating variable bindings has to be done. This aspect has been described in detail in (Ajjanagadde & Shastri, 1991; Ajjanagadde, 1991).

The second main difference between the network of (Rumelhart et al., 1986) and ours is that in addition to the external clamping of input (similar to the external clamping of nodes in (Rumelhart et al., 1986)), there is also what can be viewed as *internal clamping* in our network. This is due to the presence of background facts. In order to clarify this, note that clamping of a node reflects the belief that the proposition represented by that node is true. Thus, in (Rumelhart et al., 1986), clamping of the nodes represents that the items represented by the clamped nodes are *known* to be present in a particular room. The background facts encoded in our connectionist network correspond to the facts the agent already believes to be true. Such background facts should constrain model construction in a fashion similar to the way in which externally clamped nodes do. However, note that not all background facts residing in the agent's memory will be relevant in any given context. Thus, for example, suppose that the natural language sentence being processed is "Mary went to Super-Fries". In this context, a background fact such as *employee-of(mary,super-fries)* will be relevant. That piece of background knowledge makes the agent to conclude that the most likely purpose behind Mary's visit must be the one of working at Super-Fries (rather than, say, eating there). But, when the input proposition is *goto(mary,super-fries)*, a background fact such as *employee-of(lisa,mcdonalds)* will not be relevant. In our network, upon the specification of the input, encodings of those background facts which are relevant in that context automatically get activated. Once activated, they constrain the model construction process in a way similar to externally clamped nodes do. For details of these, please refer to (Ajjanagadde, 1991).

Another difference between the network of Rumelhart et al. (1986) and our system is that the former uses energy minimization technique for reasoning while our network does distributed evidential reasoning by spreading activation. The approach we use is quite similar to that of Pearl (1986). The main reason for our choice is one of reasoning speed. It does not appear that the currently known energy minimization techniques can achieve the kind of reasoning speed we desire. For example, Derthik (1990) reports that with a rather small knowledge base, energy minimization took about 40,000 time steps to settle into the most plausible model. On the other hand, human beings are able to construct and manipulate mental models within fractions of a second. Taking into account the slowness of neurons (Feldman & Ballard, 1982), this means that model construction take place within a few tens to a few hundred time steps. By following the distributed evidential reasoning approach (quite similar to (Pearl, 1986)), it is possible to meet such tight constraints on the number of time steps.

7 Conclusion

The process of dynamic model construction underlies a large number of cognitive tasks. The paper outlined how mental models can be represented as patterns of activity in a massively parallel connectionist network and how can fast, dynamic construction of mental models be efficiently achieved.

Acknowledgments. I would like to thank Seppo Keronen, Vipin Kumar, Uwe Oestermeier and Peter Schroeder-Heister for their comments and suggestions.

References

Ajjanagadde,V. 1991. Abductive reasoning in connectionist networks: Incorporating variables, background knowledge, and structured explananda, Technical Report, WSI 91-7, Wilhelm-Schickard-Institut, Universitaet Tuebingen, Germany.

Ajjanagadde, V., and Shastri, L. 1989. Efficient inference with multi-place predicates and variables in a connectionist system. In Proceedings of the Eleventh Annual Conference of the Cognitive Science Society, 396-403. Hillsdale, NJ: Lawrence Erlbaum.

Ajjanagadde, V., and Shastri, L. 1991. Rules and variables in neural nets. *Neural Computation* 3:121-134.

Charniak, E., and McDermott, D. 1985. *Introduction to Artificial Intelligence*. Reading, MA: Addison Wesley.

Derthik, M. 1990. Mundane reasoning by settling on a plausible model. *Artificial Intelligence* 46:107-157.

Feldman, J.A., and Ballard, D.H. 1982. Connectionist models and their properties. *Cognitive Science* 6:205-254.

Johnson-Laird, P.N. 1983. *Mental Models*. Cambridge, London: Cambridge University Press.

Mannes, S.M., and Kintsch, W. 1991. Routine computing tasks: Planning as understanding. *Cognitive Science* 15:305-342.

Pearl, J. 1986. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence* 29:241-288.

Rumelhart, D., Smolensky, P., McClelland, J., Hinton, G. 1986. Schemata and sequential thought processes in PDP models. In McClelland, J., Rumelhart, D., and the PDP Research Group (eds.), *Parallel Distributed Processing: Psychological and Biological Models*. Cambridge, MA: MIT Press.

Shastri, L., and Ajjanagadde, V. 1990. From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings, Technical Report, MS-CIS-90-05, Department of Computer and Information Science, University of Pennsylvania.

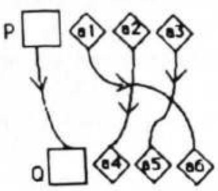

Fig.1 Encoding P(x,y,z) => Q(y,z,x).

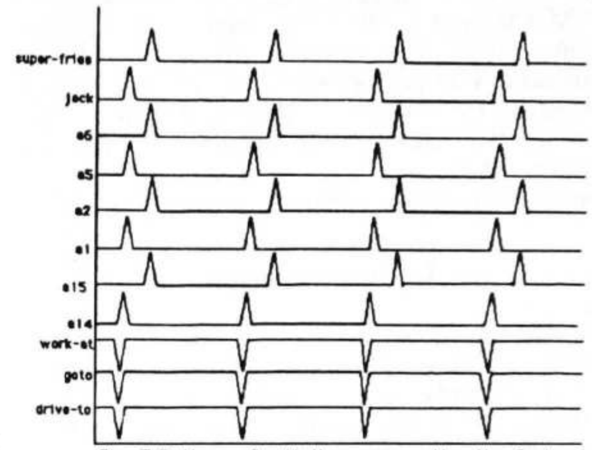

Fig. 3 Pattern of activity representing the facts drive-to(jack,super-fries), goto(jack,super-fries) and work-at(jack,super-fries).

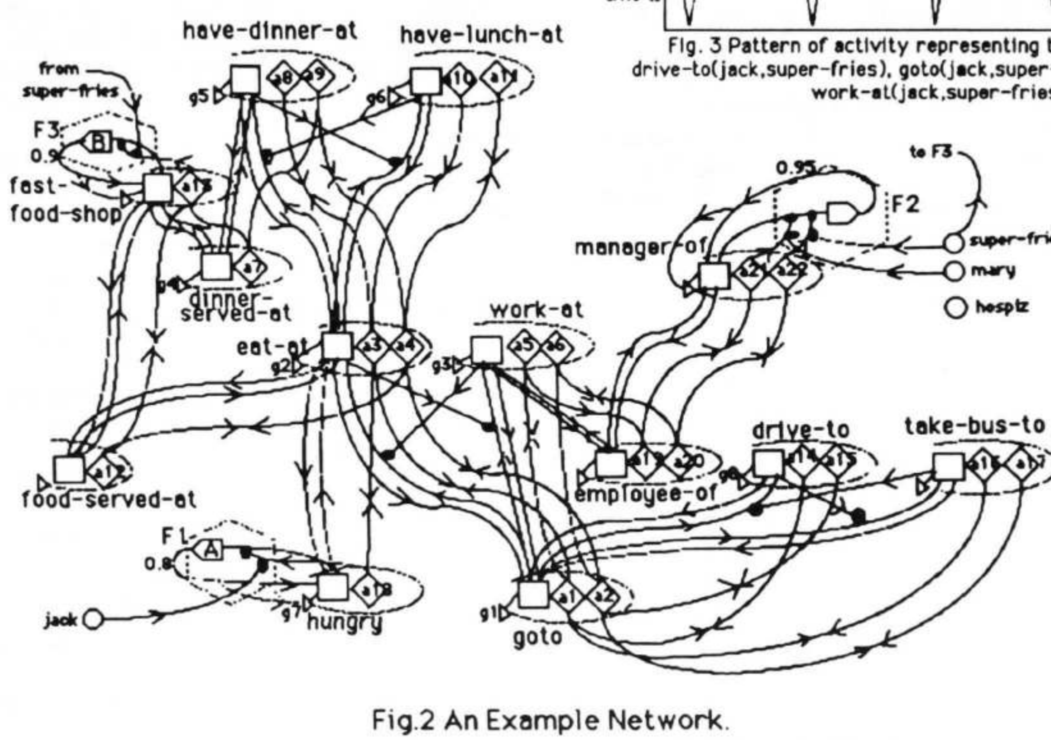

Fig.2 An Example Network.