Inference through Imagery: understanding simple sentence sequences in L_0

David Bailey
CS 288 Project
December 18, 1992
**** MAKING UP INCOMPLETE FROM FALL 1991 ****

Abstract

Until recently, the L_0 project has focused upon training connectionist nets to recognize, i.e. categorize, spatial relationships among pseudo-visual representations of geometric objects. This project is an initial effort in the reverse direction—that is, given the objects and prepositions describing their relationships, to conjure up a sort of "mental image" of a prototypical scene satisfying the relationships. This system works for multiple relationships, and thus serves as the beginnings of a system which can "understand" simple *sequences* of linguistic spatial relations, drawing inferences, creating defaults, etc. In this sense we have a simple "story understander."

1 Introduction

The stated goal of the L_0 project at ICSI is to build a learning system which can be trained, for any language, to accept or reject sentence–picture pairs on the basis of whether they agree [Feldman 90]. Towards this end, Terry Regier has build a system which is capable of categorizing idealized visual scenes consisting of geometric objects in terms of which spatial terms of the language (mostly prepositions) apply to the scene, given a labelling of the trajector and landmark of interest [Regier 92]. Andreas Stolcke is working on the parsing side of the project, whose goal is in part to extract from a sentence which visual object is the trajector and which is the landmark.

This project's focus is to begin to consider what extensions will be needed to the system in order to support simple "story understanding"; that is, the understanding of multiple sentences. The key task here is to carry the relevant information from sentence to sentence in order to generate defaults to fill in implicit information, and to adapt uncertain information from early sentences when more certain information in later sentences contradicts earlier assumptions.

Towards this end, this project investigates a simple scheme for "imagery"; that is, the generation of a *prototypical* image from a spatial term, an image capturing the essential characteristics of that term. We accomplish this task by first training specialized connectionist nets as Regier does in his thesis. Then, we "clamp" the output units of the net to a spatial term which we would like to "imagine", and then employ an unusual twist on the backpropagation training algorithm previously used by Thrun [Thrun 92], which adjusts the net's *input values* rather than its weights. In this way, the inputs converge to the prototypical image.

This report points out some limitations of what was done, some obvious extensions, and then discusses some more far-ranging ideas which may prove fruitful in the future.

2 L_0 Background

The full L_0 spatial relation system is quite complex and will only be quickly reviewed below (see [Regier 92] for details). For this project, a much-reduced system was used, and that explanation follows afterwards.

2.1 Regier's system

In its full generality, Regier's spatial term recognizer is able to handle both static and motion terms (as well as terms with both static and motion senses). The system began as just a static term recognizer, and that structured network serves as a module in the network which handles motion terms. Figure 1 shows an overview of the final structured net.

The lower module of the net is responsible for extracting salient *static* features from the input, a bitmap. Actually, the bitmap is used directly by only part of the net—the part responsible for detecting contact- and inclusion-based features (right-hand side of Figure 1). The rest of this module operates on angle-based features of the input bitmap rather than on the bitmap itself. These inputs include the orientation of the line connecting the centers of mass of the landmark and trajector (LM and TR henceforth), called the "center-of-mass angle," and the orientation of the line connecting the LM and TR where they are closest to each other, called the "proximal angle."

The bitmap is scanned by a grid of units which sit above it, and which are gated by whether the bitmap point below them is part of the outline of the TR. Where this is true, the units exhibit a center-surround behavior. Finally, two special units reduce the information in this grid into two values: the maximum and average valuea of the nodes in the grid. The angle inputs are processed by a collection of so-called *theta nodes*, which can operate in two ways. Some of them accept as input a single input angle (for example, the center-of-mass orientation), and maintain two state values: a target orientation, and a variance. These "learning" theta units, during training, will adapt so that their target orientations detect a useful feature for discriminating among the words of the training language, and the variance will indicate the degree to which the target orientation really is informative. The other type of theta unit accepts two input angles as inputs, and during training simply adapts its variance to indicate whether the co-incidence of the two input angles is important in the training language. The hope is that by connecting a bunch of such theta nodes to various input angles and pairs of input angles, we will end up making the "right" comparisons at at least some of them. Finally, these theta nodes as well as the "max" and "average" nodes from the bitmap are fully connected to the units in an output layer where there is one node for each spatial term (in some variations, a hidden layer is inserted first). Training proceeds basically

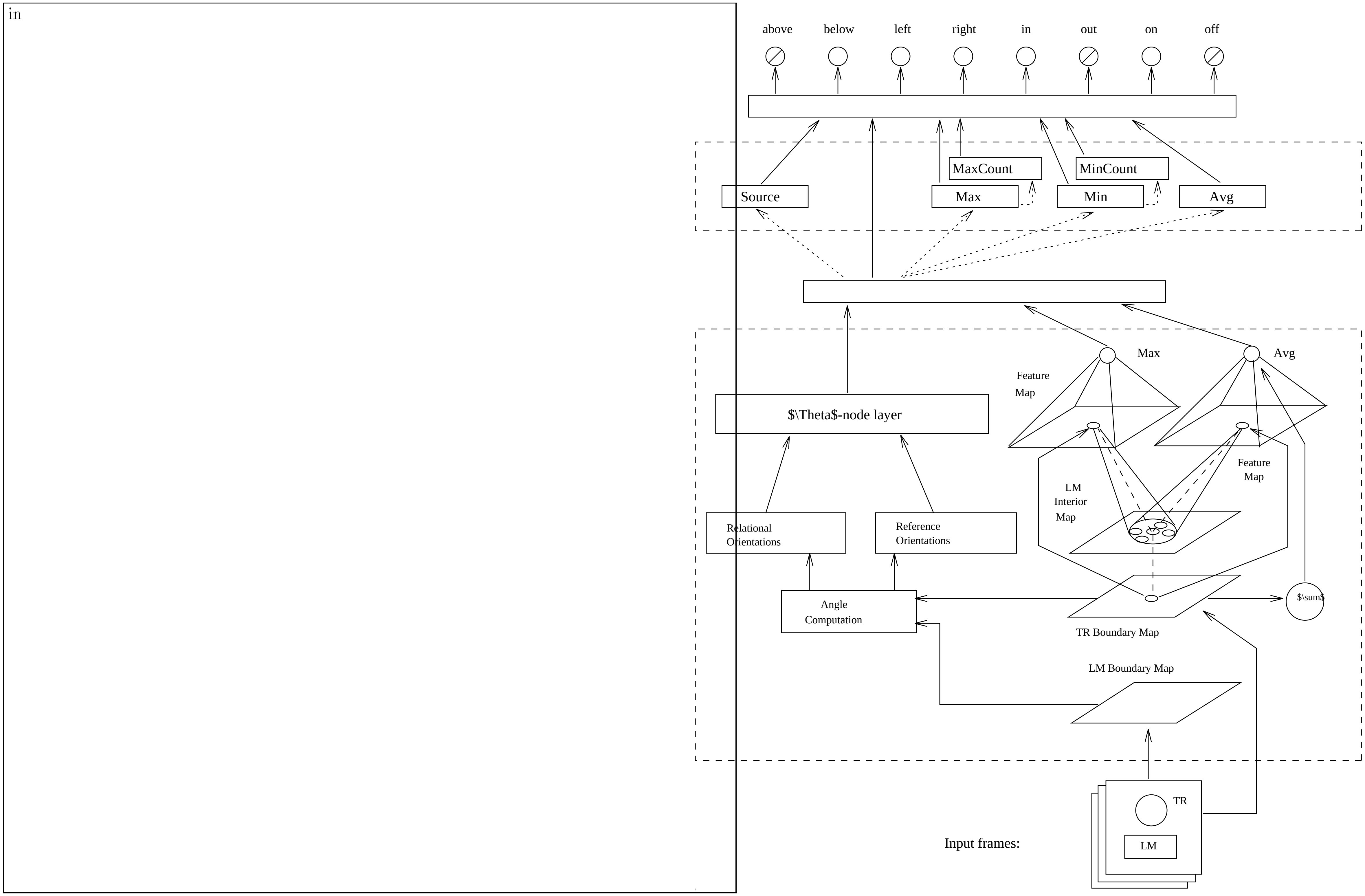


Figure 1: The complete static and motion spatial term recognizer.

by the standard backpropagation algorithm, with some efficiency enhancements, plus the obvious caveat that the derivative functions are non-trivial due to the unusual unit functionalities.

The motion net is an extension of this system. Instead of the layer of static term nodes, we feed the theta, "max" and "average" nodes into a series of *buffers*. During the presentation of each "movie," the "source buffer" keeps a record of the static module's outputs on the first frame of the movie (in order to ensure the significance of the all-important first frame in the net's final analysis of the movie). Three other buffers track the minimum, maximum and average values of each of the static output units during the course of the movie. Finally, one buffer always reflects the static module's ouputs on the current frame. Finally, these 5 buffers are fully connected to a vector of units representing all the static *and* motion terms (again, with optional intervening hidden layers). Weight updates to the nework occur only after the presentation of the final frame of the movie.

2.2 My reduced system

First off, I address only static terms, so the buffering system just described is irrelevant to this project. The appropriate representation for prototypes of motion terms raises complicated issues and thus will not be addressed for a while yet.

Simplifications were also made to the static module itself at the outset of this project, some of which were subsequently removed; one major one remains and its removal is next on the agenda. Essentially, most of the work described here has been done on a static module consisting only of theta units. This is a direct result of the current imagery system's dependence upon gradient descent, which does not mix well with the discontinuous aspects of the bitmap system's "max" node, nor with the difficult credit assignment problem which results from the massive converence from the entire bitmap into just 2 units, "max" and "average". This is the simplification which remains. Initially, we started with an even simpler system, which contained only 4 "learning theta units" initially pointing up, down, left and right, each one connected to the center-of-mass orientation, plus another input unit for the center-of-mass distance, which was connected directly to the outputs. At this point, proximal angles, fixed theta units, and hidden layers have been added in order to learn more complex terms.

In addition, we have *added* a new unit type not found in Regier's system—"distance units," which like learning theta units have an adjustable target distance and variance. These make it possible to learn concepts such as *near* without the bitmaps.

See Figure 2 for a sample architecture used in this project.

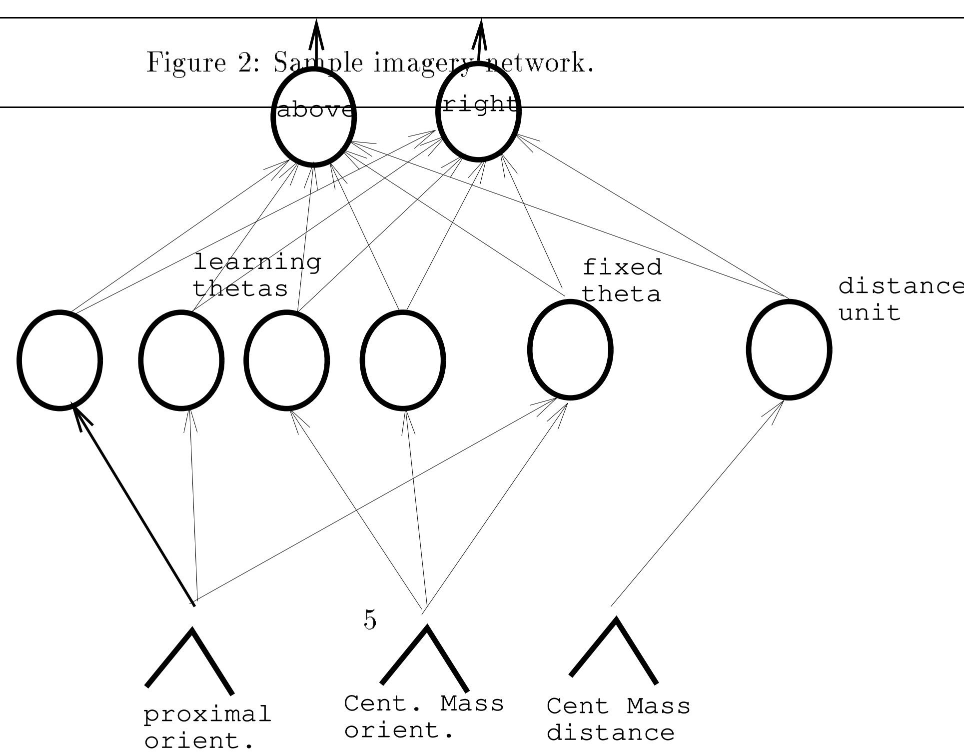
It is important to note that this reduced system is not capable of learning many of the spatial terms Regier's thesis system could learn, and so of course we have been limited in the variety of imageries we have been able to experiment with.

3 Focus and Simplifying Assumptions

In addition to the simplified network just described, two important aspects of a real imagery / inference system have been "faked" in this project (and in L_0 in general), and should be pointed out.

in

Figure 2: Sample imagery network.



First off, there is a binding problem involved with assigning the roles of landmark and trajector to the two objects in an image. In Regier's system as well as mine, these are conceptually "pre-labelled," in that they feed into functionally distinct parts of the network. This assumes several important AI problems have been solved: on the vision side, that we are able to perform segmentation and object recognition to extract the two objects; and on the NLP side, that we are able to determine from context which objects are to serve which roles for a given spatial term occurring in a sentence. The vision problems are rather decoupled from the L_0 task, so it reasonable to take this tack. The NLP problems are directly related to L_0 but so far have been investigated independently by Andreas Stolcke.

The imagery system makes an even stronger assumption. The basic approach is to model a system which considers the various relationships in a scene in turn, satisfying constraints imposed by each until finally all constraints are met. Thus, the static module must be applied to various LM/TR bindings all within the same scene. The mechanism to switch between these bindings essentially is an *attention mechanism.* This is a very interesting topic which I would like to pursue but which is "hacked" in this project—i.e., regular code, not any sort of connectinist model, keeps track of the multiple objects and applies the right ones to the right input units at the right times.

4 Prototypes via Backprop

With those preliminaries, we are ready to look at the actual techniques and some results. I'll proceed in order of increasing complexity.

The key idea of this project is to derive prototype "images" by, in some sense, running Regier's network "backwards," effectively performing gradient descent in input space. We apply a naive initial image to the inputs of the trained network, then do a forward pass. The output units are then compared against the target outputs which represent the spatial relation we wish to "imagine." Errors are computed and backpropagated as if we were training, but weights are not updated. When the errors reach the first layer of computational nodes, though, we *backpropagate error once again* into the input units—a concept which makes little sense during normal network training, but which is here used to calculate an adjustment to the inputs in a direction which is more likely to be categorized so as to match the imagery target. This process is iterated until the inputs have migrated to values which, when the network performs a forward pass, induce the correct pattern across the spatial term units. Typically, this process converges, and does so in only a handful of iterations.

Note that since our simplified static module takes as input only a small collection of orientations, there is little information as to the actual shapes of the objects. In all the examples below, various objects have been randomly chosen *a priori*, and the details of the choices have little effect except for the size of the object and its corners which produce interesting proximal orientation effects.

Now we review the various cases which have been modelled, and the issues they gave rise to:

4.1 One Preposition, Imagine Two Objects & One Relation

This is the "base case." We have trained networks with single outputs units to recognize spatial terms such as *above* and *right-of.* Then these nets switch into "imagery mode" and in approximately

in

Figure 3: Imagining "Circle is above Square."

in

Figure 4: Imagining "Circle is above and right-of Square."

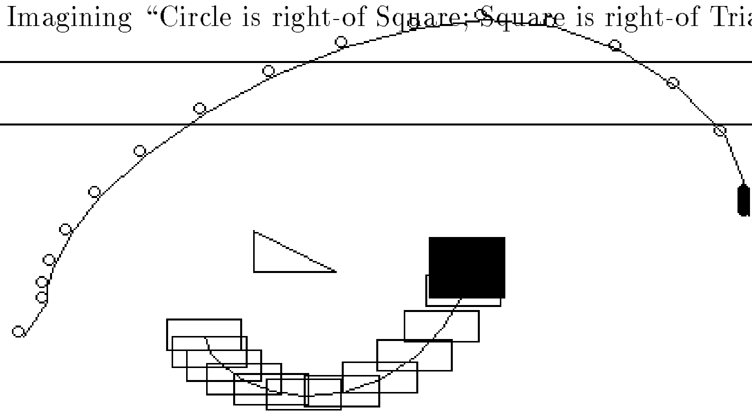
5-10 iterations converge on reasonable-looking prototypical examples of these concepts, for many initial positions of the trajector. See Figure 3. If we consider for a moment just the centers of mass of the objects, then the performance of this is neatly summed by the notion of the gradient vector field. To be successful, the legitimate region for a spatial term must include an attractor, and furthermore the basic of that attractor should extend over the entire legitimate region. If there are other attractors, these will be a danger to the system. For instance, in this simple system composed largely of theta nodes, there is a very flat spot in the derivative at 180 degrees away from the correct attractor. If the trajector starts the imagery process at exactly 180 degrees from where it should be, it will be stuck in a local minimum.

4.2 Multiple Prepositions, Imagine Two Objects & One Complex Relation

The next step was to train a single net with multiple output units to recognize multiple spatial terms (e.g., *above*, *right-of* and *near*). Then, we tested the networks's ability to *combine* prototypes by specifying an imagery target of, say, *above-and-right-of*. Results are basically successful; an example is shown in Figure 4. Note that in this case, the trajector will move according to the *sum* of the vector fields for the multiple concepts. While theta units provide a decent error surface individually, there are more local minima (that is, false attractants) when we start to add vector fields. We found a number of configurations in which the trajector would veer off and not converge. A program to plot these vector fields would have been useful here, but alas time is short.

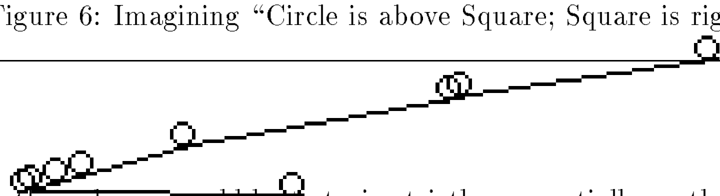
in

Figure 5: Imagining "Circle is right-of Square; Square is right-of Triangle."



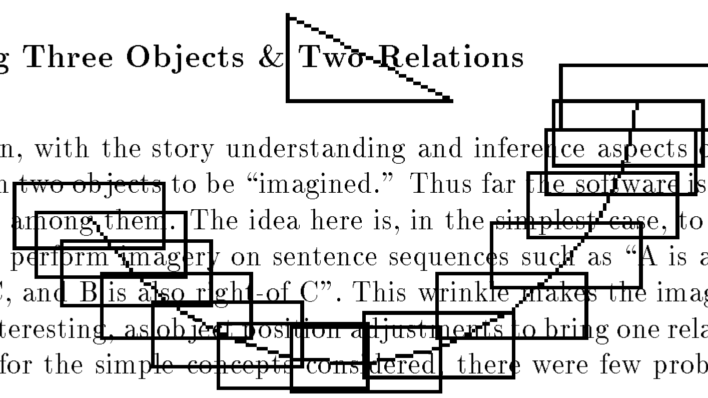
in

Figure 6: Imagining "Circle is above Square; Square is right-of Triangle."



A twist on this scheme would be to train strictly sequentially on the two targets. This is similar to Shepard's work on, e.g., mental rotation, wherein an internal model must be updated in discrete steps, as opposed to having all the information available at once. This possibility was not explored.

4.3 Imagining Three Objects & Two Relations



The next extension, with the story understanding and inference aspects of the project in mind, is to allow more than two objects to be "imagined." Thus far the software is limited to three objects, with two relations among them. The idea here is, in the simplest case, to train the net on a single spatial term, then perform imagery on sentence sequences such as "A is above B, B is right-of C" or "A is right of C, and B is also right-of C". This wrinkle makes the imagery convergence process somewhat more interesting, as object position adjustments to bring one relation into register disturb others. However, for the simple concepts considered, there were few problems. See Figures 5 and 6 for examples.

While it is difficult to see any new inferences which could be drawn from this particular imagined scene, it is clear that with 4 objects, we could create a situation where two of the objects would be in a relation to each other which directly corresponds to a spatial term, and hence, if we input those to objects to the net as TR and LM, the appropriate output node would fire.

We suspect that local minima will prove troublesome with more sophisticated nets that can handle larger vocabularies. See the next Section for more complex ideas for handling this case. See Figure for an example of the system "imagining" "Circle is right-of Square and Square is right-of Triangle."

4.4 Code

This system is implemented on top of ICSIM [Gomes 92], a structured connectionist simulator written primarily by Ben Gomes at ICSI. ICSIM is written in the language Sather [Omohundro 91]. Due to the complexity and relative obscurity of both the language and simulator, I don't think it is productive to spend time reviewing the code for the project, which is rather straightforward in any case. To give a "flavor" of the code, I have attached the main control loop source file, as well as a typical parameter file, at the end of this report.

5 Extensions to this Approach

5.1 Restoring proximal orientations

Originally the proximal orientation were dropped from the system because it is non-obvious how to derive a bitmap version of the imagined scene from just proximal and center-of-mass orientations and distances—that is, this information underspecifies the scene. Thus we opted to use random objects, focussing only on "imagining" the proper center of mass orientation and separation distance. It would be instructive to attempt to restore the proximals—while still employing our random objects—and use them to *rotate* the objects into certain prototypical positions.

5.2 Including Bitmaps

Some concepts, such as *in* or *on*, currently depend critically upon the bitmap half of the static net, since they involve inclusion and contact, respectively, which theta units cannot compute—at least in a decisive way. Thus an important step is to find a reasonable heuristic by which we may make use of the bitmap apparatus.

5.3 Heuristic constraints

By far the most interesting issue raised by this initial work is how to insert appropriate constraints into the system to handle a large vocabulary of words and large numbers of concurrent relationships. Our simple approach of independent gradient descents interacting only to the extend that relations perturb each others' object positions, will not stand up to many concepts. For instance, if we attempt imagery on the 2 sentences "Circle above Square. Triangle above Square," we currently have no mechanism to prevent the Circle and Triangle from overlapping. Thus non-overlapping objects is the first constraint which should be added. There will be others constraints; for example, English *over* requires (loosely at least) that no objects fill the space between the TR and the LM below it.

6 Other Approaches

While the ultimate goal is to find a vaguely neurally plausible model for the L_0 task, "non-neural" algorithms which might actually solve the task are nonetheless of interest, in the hope that they may in turn be implementable in a connectionist style. A Bayesian approach to the imagery task suggests itself as a possibility here, especially if we consider that it may frequently be advantageous to allow *several* prototypes for a given spatial term, rather than attempting to mush several senses into a single "average" prototype. The Bayesian model can easily accomodate this paradigm.

For instance, perhaps the best *a priori* belief distribution after encountering an "A *over* B" relation is 0.5 on the "flying over" sense and 0.5 on the "hovering directly above" sense. Then, say we encounter a relation whose only interpretation is that another object C is sitting directly on B, covering it completely. Then, a conditional probability will cause our belief in the "hovering directly over" sense of the original *over* relation to drop significantly.

7 Conclusions

This project has explored a simple scheme for generating prototype images from, essentially, symbols representing spatial terms in a natural language. These prototypes can be used as defaults, to aid in understanding a sentence sequence, and can also be used for inference by assigning new TR / LM pairs to the objects in the "imagined" scene and applying a forward pass to the network to see which spatial terms apply to that binding.

An important question is how "low" the representation for the "imagined" scene should go. That is, is it really useful to go all the way back to a bitmap as would appear on the retina? Obvious the brain does not do this. There must be a somewhat higher level of feature nodes which can be manipulated in such a way as to cause "hallucinations", or imagery. What characteristics might this representation have? Would it merely be some layer of our L_0 net? We propose that a reasonable representation would be one which is able to specify some aspects of the scene while leaving others unspecified. A bitmap cannot do this, since every pixel must carry some light level. But a representation composed of feature vectors, where each vector element is a *pair* containing a value as well as a "degree of committment", might work. For example, a prototype for *near* might represent its constraints by setting the distance feature's value to 5 (a small distance) and its degree of committment relativley high, while it would set its center-of-mass orientation feature's committment value very low since this feature is irrelevant to nearness. In this way, feature vectors for multiple relations might be combined by a process that we might describe as "quantified unification"—where two feature vectors are combined as long as there are no high-committment, conflicting feature values, and if combined, high-committment values override low-committment counterparts.

There are some problems to the current backprop-oriented approach to imagery—problems which are common to neural network algorithms. Results of our experiments have been annoyingly, if not excessivley, sensitive to parameters such as learning rates and initial variances on theta nodes, leading to the "black art" syndrome which leaves one wondering whether it is the system or the researcher who has succeeded at the learning task. And further, we have not had time to come to any true understanding of the strengths or weaknesses of the approach; we have only collected a bit of emperical evidence.

Nevertheless, it is hoped that this project will steer the $L0$ project in a direction which pays close attention to the problem of gradual building up of understanding during the parsing of sentence sequences, as opposed to merely recognizing single sentences. The need for a connectionist attention mechanism, as mentioned earlier, is expected to be a key component in this endeavor.

References

[Feldman 90] Feldman, J., Lakoff, G., Stolcke A. and Weber, S. "Miniature Language Acquisition:: A Touchstone for Cognitive Science." Technical Report TR-90-009, International Computer Science Institute, Berkeley, CA, 1990.

[Gomes 92] Gomes, B. "The Design of ICSIM 1.9." Master's Thesis, UC Berkeley, December 1992.

[Omohundro 91] Omohundro, S. "The Sather Language". ICSI working document (no TR number), June 1991.

[Regier 92] Regier, T. "The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization." Ph.D. Thesis, UC Berkeley, September 1992.

[Thrun 92] Thrun, S. "The Role of Exploration in Learning Control." In *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*. White, D and Sofge, D., eds. Van Nostrand Reinhold, 1992.