Escaping the Building Block / Rule Dichotomy : A Case Study

Simon D. Levy
Computer Science Department
Washington and Lee University
Lexington, VA 24450, USA

Jordan B. Pollack
DEMO Laboratory,
Computer Science Department
Brandeis University
Waltham, MA 02453, USA

Abstract

The traditional approach to complex problems in science and engineering is to break down each problem into a set of primitive building blocks, which are then combined by rules to form structures. In turn, these structures can be taken apart systematically to recover the original building blocks that went into them. Connectionist models of such complex problems (especially in the realm of cognitive science) have often been criticized for their putative failure to support this sort of compositionality, systematicity, and recoverability of components. In this paper we discuss a connectionist model, Recursive Auto-Associative Memory (RAAM), designed to deal with these issues. Specifically, we show how an initial approach to RAAM involving arbitrary building-block representations placed severe constraints on the scalability of the model. We describe a re-analysis the building-block and "rule" components of the model as merely two aspects of a single underlying nonlinear dynamical system, allowing the model to represent an unbounded number of well-formed compositional structures. We conclude by speculating about the insight that such a "unified" view might contribute to our attempts to understand and model rule-governed, compositional behavior in a variety of AI domains.

Introduction

The traditional approach to complex problems in science and engineering is to break down each problem into a set of primitive building blocks, which are then combined by rules to form structures. In general, this approach has proved successful enough that it is rarely mentioned, let alone questioned, in contexts outside of introductory AI and engineering courses. Nevertheless, when applied to complex systems in nature, the approach tends to be an oversimplification.

At one end of the spectrum we find autocatalytic chemical reactions that generate the components required for their own synthesis (Zaikin & Zhabotinsky 1970). Such reactions provide perhaps the simplest example of a system in which the roles of building blocks and rules are entangled in a unconventional way. At the other end we have the full-blown complexity of human behavior, notably language. In this domain the Chomskyan "words and rules"

approach (Pinker 1999) has become the dominant analytical framework, despite the theoretical difficulties associated with the definition of "word" (Sciullo & Williams 1987), the practical difficulty of automatically isolating individual words in fluent speech (Rabiner & Juang 1993), and the everyday observation that real language seems to contain more exceptions than rules. In between these two extremes lies all the complexity of real biological systems, in which basic engineering concepts like modularity and reuse run up against the difficulty of knowing precisely what building blocks are being modularized and acted upon in rule-governed ways. Suggestions for these include genes (Dawkins 1990), species (Wynne-Edwards 1962), and mental faculties (Fodor 1983), among others.

Engineers and researchers using biologically inspired approaches to problem-solving have had to deal with various manifestations of this issue. In the field of genetic algorithms the notion of "building block" (Goldberg 1989) has received a good deal of attention and generated some controversy (Forrest & Mitchell 1993). It is only recently that thorough studies of crucial issues like compositionality in artificial and natural evolution have been undertaken (Watson 2002), and a great deal of work remains to be done.

In a somewhat parallel development, the literature on connectionist (a.k.a. neural) networks is fraught with attempts to address the issue of rule-governed compositionality. At one extreme, the distributed, "eliminativist" approach tries to do away with structures and rules entirely, seeing both as emergent epiphenomena (e.g., hidden-layer activations) in the behavior of networks trained on patterned data (Elman 1990). At the other extreme, so-called localist approaches choose to represent structure explicitly, either through direct connections among units representing primitive concepts, or through neuron-like synchronous firing of those units or clusters of units (Shastri & Ajjanagadde 1993).

The RAAM Model

An entirely different approach, discussed in the remainder of this paper, takes the traditional notion of compositional

structure very seriously, while still attempting to exploit the distributed representations that give neural network models so much of their appeal. In this approach, called Recursive Auto-Associative Memory, or RAAM (Pollack 1990), the basic building blocks are binary strings representing the various entities to be combined (words, concepts, etc.). As shown in Figure 1, the RAAM network consists of an *encoder*, which recursively composes the binary strings to produce fixed-size vector representations of structures over those strings, and a *decoder*, which unpacks the encoded vector representations into (an approximation of) their original forms. Together, the encoder and decoder are trained as an auto-associator (Ackley, Hinton, & Sejnowski 1985) using the standard neural net back-propagation algorithm (Rumelhart, Hinton, & Williams 1986).
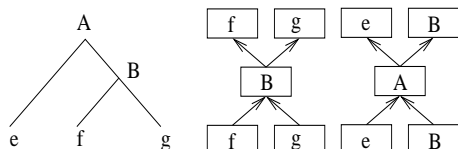

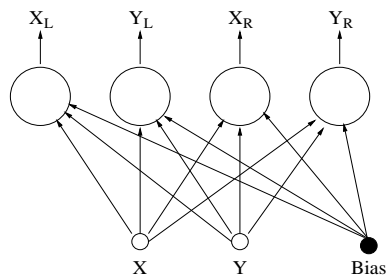Figure 1: RAAM encoding and decoding a tree structure.

The key insight of RAAM is that the hidden-layer activations representing compositional structure can be fed back into the encoder input, allowing structures (fixed-arity trees) of arbitrary size and complexity to be built. Similarly, the output of the decoder is fed back to the decoder input, after passing through a "terminal test" to see whether it is similar enough to a binary string (all zeros and ones) for decoding to terminate. If not, decoding continues. The decoder output is computed by a logistic sigmoid "squashing" function whose range is the interval (0,1). Therefore, the simplest terminal test is simply a hard threshold in which values below 0.2 are treated as 0, and values above 0.8 as 1.

Limitations of RAAM

Although RAAM answered the challenge of showing how neural networks could represent compositional structures in a systematic way, and led to lots of philosophical discussion, the model failed to scale up reliably to data sets of more than a few dozen different trees. This limitation arose from the terminal test, which created a variety of "halting problem": decoding either terminated too soon, treating a non-terminal as a terminal, or continued indefinitely, treating a terminal as a non-terminal. In addition, encodings of novel structures over existing terminals were typically decoded to already-learned structures over those terminals.

A number of different fixes were attempted, generally involving a more elaborate terminal test than the simple threshold. No significant progress was made, however, until a key insight about the decoder emerged.

New RAAM formulation

$$X_L = \frac{1}{1 + e^{-(w_{LXX}x + w_{LXY}y + w_{LX})}}$$

$$Y_L = \frac{1}{1 + e^{-(w_{LYX}x + w_{LYY}y + w_{LY})}}$$

$$X_R = \frac{1}{1 + e^{-(w_{RXX}x + w_{RXY}y + w_{RX})}}$$

$$Y_R = \frac{1}{1 + e^{-(w_{RYX}x + w_{RYY}y + w_{RY})}}$$

Figure 2: An example RAAM decoder that is a 4 neuron network, parameterized by 12 weights. Each application of the decoder converts an (X, Y) coordinate into two new coordinates.

Consider the RAAM decoder shown in figure 2. It consists of four neurons that each receive the same (X, Y) input. The output portion of the network is divided into a right and a left pair of neurons. In the operation of the decoder the output from each pair of neurons is recursively reapplied to the network. Using the RAAM interpretation, each such recursion implies a branching of a node of the binary tree represented by the decoder and initial starting point. However, this same network recurrence can also be evaluated in the context of dynamical systems. This network is a form of *iterated function system* or IFS (Barnsley 1993), consisting of two pseudo-contractive transforms which are iteratively applied to points in a two-dimensional space.

In the context of RAAMs the main interesting property of contractive IFSes lies in the trajectories of points in the space. For contractive IFSes the space is divided into two sets of points. The first set consists of points located on the underlying attractor (fractal attractor) of the IFS. The second set is the complement of the first, points that are not on the attractor. The trajectories of points in this second set are characterized by a gravitation towards the attractor. Finite, multiple iterations of the transforms have the effect of bringing the points in this second set arbitrarily close to the attractor. In this sense, the attractor points, rather than points above or below an arbitrary cutoff threshold, represent a guaranteed terminating condition for the RAAM decoder.

Using the points (vectors) of the attractor, rather than

bit strings, as a "natural" terminal test, allowed us to overcome the RAAM halting problem. The relevant threshold is not proximity to an arbitrary value like 0 or 1, but rather proximity to the (theoretically infinite) dust of points representing the attractor. With this approach, we were able to increase the number of decodable trees from a few dozen to tens of thousands. Ultimately the number of decodable trees was in fact proved to be unbounded, in (Melnik, Levy, & Pollack 2000). An equally promising result from that work was the observation that the unbounded set of decoded trees could be made to conform to a set of rules, contrary to traditional claims about the fundamental incompatibility of rule-based and connectionist approaches.

The difference between the old and new formulations of RAAM is vividly illustrated in Figure 3, which maps the attractor and decoded trees for an attractor image derived by a visual "Blind Watchmaker" technique (Dawkins 1986). As the figure shows, the attractor (terminals) and its complement (decoded trees) are both part of a single, (theoretically) continuous space. Every point in this space decodes to a terminal or non-terminal tree, with the depth and variety of the trees being limited by the arbitrary pixel resolution at which the space is sampled. [1]
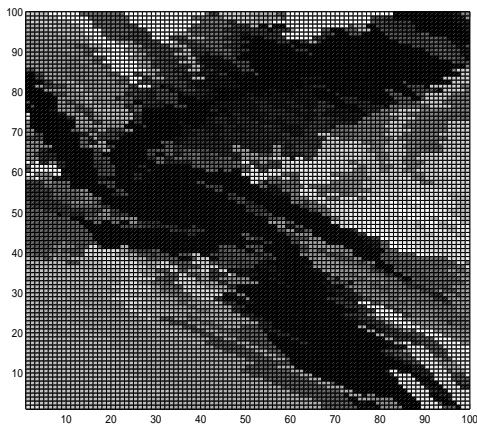

Figure 3: Map of unique trees in 2D fractal RAAM, sampled at 100x100 pixels. Each non-terminal tree is represented by a different grayscale value. Attractor (terminal set) is black spiral.

Re-grounding the Symbols

Eschewing the use of arbitrary bit-string representations as primitive building blocks overcomes the scalability problems with the older RAAM formulation, but leaves us with a bootstrapping problem: if the terminals are defined as the points on the attractor, and the attractor is a function of the decoder weights, then the symbolic "ground" is constantly changing underneath our feet as the weights are adapted via back-prop or some other learning algorithm. We currently see two ways out of this dilemma.

First, we might exploit some other principle to fix the set of terminal vectors, and then train the RAAM encoder/decoder network as in the older version of the model, using back-propagation. The criterion for evaluating such a principle would be the extent to which it produced terminal vectors yielding a rich, complex dynamics of the sort portrayed in Figure 3. A salient feature of this figure is that the terminals (attractor) take on real values distributed over a large portion of the unit square (whereas the original bit-string representations would be located only in the extreme corners of this space). This suggests that general (real-valued) terminal vectors in the space $(0,1)^D$ are the sort of representations we should be looking for.

Fortunately, there is a growing body of literature describing techniques for representing symbolic data (such as word meanings) in exactly such vector spaces. The Latent Semantic Analysis model (Landauer & Dumais 1997), which uses co-occurrence statistics on text corpora, is one such model that has enjoyed a good deal of success in a number of domains where traditional symbolic techniques have proven less adequate. Another possible candidate is Elman's Simple Recurrent Network (SRN) model of temporal structure.(Elman 1990). Elman has shown how training this sort of network on a temporal prediction task can result in hidden layer activations that embody meaningful categorical information about the item presented on the input layer (such as part of speech and more fine-grained semantic distinctions). It is intriguing to speculate how such representations might interact with a RAAM being trained to induce trees over the items in the temporal sequences.[2]

Second, we might extend the unified approach taken so far, by using the IFS decoder weights to label the attractor terminals as well. Barnsley (1993) notes that each point on the attractor is associated with an address which is simply the sequence of indices of the transforms used to ar-

---

[1] In this respect we may accused of returning to the sort of arbitrary discrete cutoff that plagued the terminal test of old RAAM. We hope by now to have conveyed the sense in which the fractal terminal test respects the underlying dynamics of the model, in a way that the thresholding cutoff terminal test did not. Further, our artificial neurons, implemented with double-precision floating-point numbers, likely have excessively *high* precision as compared to the biological neurons that inspire them.

[2] With respect to language learning, a supervised algorithm like back-prop seems more appropriate to the sequential tasks given to Elman's SRNs, than it does to a structure-learning model like RAAM: whereas the sequence of words is explicitly available as input to the language-learner, the putative syntactic structure of of that sequence is much less obvious. We might conceptualize the language-learner's task as somehow deducing a set of decoder weights that will yield the appropriate sequences, given an SRN representation of the sequential items.

rive on that point from other points on the attractor. The address is essentially an infinite sequence of digits. Therefore to achieve a labeling for a specific alphabet we need only consider a sufficient number of significant digits from this address. With a two-transform decoder and one significant digit, this scheme gives us binary addresses (0 and 1, a and b) sufficient for representing a large variety of formal languages. This approach allows us to derive the decoder weights without the need for an encoder, using a number of evolutionary algorithms instead of back-prop. The first author is currently working with an undergraduate student in an attempt to co-evolve IFS RAAM decoders for a pursuer-evader communication game of the sort described in (Ficici & Pollack 1998), where the "messages" consist of zeros and ones. Our hope is that simultaneous, conflicting constraints of this game (be predictable to friends, confusing to enemies) will yield a rich compositional dynamics, based solely on the string representations available to the communicating agents.

Relevance to Computational Synthesis

From the perspective of the current symposium, the main insight of the IFS RAAM work is that *treating the building blocks and compositional rules of a system as two separate components may be an inherently limiting approach*. It was only after understanding the terminals (attractor points) and non-terminals (transients to attractor) as merely different aspects of a single underlying system (IFS) that we were able to overcome the severe limitations of the original RAAM formulation. Although this result was obtained in the context of a particular type of model (auto-associative neural net) applied to a particular type of compositional structure (fixed-arity tree), we believe that the insight gained from the result may have profound implications for the field of computational synthesis as a whole. In facing the challenge of scaling to high complexities, we may well need to turn away from the traditional dichotomy between building block and rule, and seek out substrates in which such distinctions emerge as artifacts of human observation, rather than being stipulated design principles.

References

Ackley, D. H.; Hinton, G.; and Sejnowski, T. 1985. A learning algorithm for boltzmann machines. *Cognitive Science* 9:147–169.

Barnsley, M. F. 1993. *Fractals everywhere*. New York: Academic Press.

Dawkins, R. 1986. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*. New York: W.W. Norton and Co.

Dawkins, R. 1990. *The Selfish Gene*. Oxford University Press.

Elman, J. 1990. Finding structure in time. *Cognitive Science* 14:179–211.

Ficici, S., and Pollack, J. 1998. Coevolving communicative behavior in a linear pursuer-evader game. In Pfeifer; Blumberg; and Kobayashi., eds., *Proceedings of the Fifth International Conference of the Society for Adaptive Behavior*.

Fodor, J. 1983. *The Modularity of Mind*. MIT Press.

Forrest, S., and Mitchell, M. 1993. What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *Machine Learning*.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Landauer, T. K., and Dumais, S. T. 1997. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104:211–240.

Melnik, O.; Levy, S.; and Pollack, J. 2000. RAAM for an infinite context-free language. In *Proceedings of the International Joint Conference on Neural Networks*. Como, Italy: IEEE Press.

Pinker, S. 1999. *Words and Rules: The Ingredients of Language*. Basic Books.

Pollack, J. 1990. Recursive distributed representations. *Artifical Intelligence* 36:77–105.

Rabiner, L., and Juang, B.-H. 1993. *Fundamentals of Speech Recognition*. Prentice Hall.

Rumelhart, D.; Hinton, G.; and Williams, R. 1986. Learning internal representation by error propagation. In Rumelhart, D., and McClelland, J., eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press.

Sciullo, A.-M. D., and Williams, E. 1987. *On the Definition of Word (Linguistic Inquiry Monographs, No. 14)*. MIT Press.

Shastri, L., and Ajjanagadde, V. 1993. From simple associations to systematic reasoning. *Behavioral and Brain Sciences* 16(3):417–494.

Watson, R. A. 2002. *Compositional Evolution: Interdisciplinary Investigations in Evolvability, Modularity, and Symbiosis.* Ph.D. Dissertation, Brandeis University.

Wynne-Edwards, V. C. 1962. *Animal Dispersion in Relation to Social Behavior*. New York: Hafner.

Zaikin, A., and Zhabotinsky, A. 1970. Concentration wave propagation in two-dimensional liquid-phase self-oscillating system. *Nature* 535–537.