Image and Symbol

Continuous Computation and the Emergence of the Discrete*

Bruce J. MacLennan
Computer Science Department
University of Tennessee, Knoxville

August 5, 1993

Abstract

The development of connectionist models will be promoted by a theoretical construct analogous to the calculus in symbolic models. The *simulacrum* is proposed to fill this role. Whereas calculi implement discrete processes operating on formulas, simulacra implement continuous processes operating on abstract images. The theory of simulacra suggests a novel approach to many issues in cognitive modeling, including classification, invariants in behavior, constituent structure, intentions and approximately discrete processes, such as rule-like behavior, symbolic cognition, and language.

*An abbreviated version of this report will appear in *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration, Volume I: Basic Paradigms; Learning Representational Issues; and Integrated Architectures*, edited by Vasant Honavar and Leonard Uhr, New York, NY: Academic Press (in press).

There in the ring where name and image meet,
Inspire them with such a longing as will make his thought
Alive like patterns a murmuration of starlings,
Rising in joy over wolds, unwittingly weave.

— W. H. Auden (*Perhaps, Prologue* from *On the Island*)

1 Simulacra

1.1 Purpose

The concept of a calculus is the central theoretical construct underlying all traditional ("symbolic") approaches to knowledge representation and processing. By this I mean that the concept of a calculus captures the essential characteristics of discrete physical symbol manipulation in an idealized form suitable for mathematical analysis. To a large extent (digital) computer science is the theory and practice of using calculi to solve problems.

Unfortunately, connectionist models of knowledge representation and processing lack a central, unifying theoretical construct analogous to the calculus. This impedes progress in two ways. First, we have no clear idea of the range of possible connectionist approaches, so our imagination of possible representations and processes is restricted. (I think this is especially evident in connectionist language processing.) Second, without such a theoretical construct we are unable to prove theorems that apply in principle to all possible connectionist systems (e.g., results analogous to the uncomputability and undecidability results for calculi).

The representation of knowledge and inference by calculi is an old idea (it is inherent in the Greek word *logos*), and so it has been well explored over the centuries. Nevertheless, a clear understanding of the capabilities and limitations of calculi did not emerge before the first half of this century, in the work of Gödel, Turing, Church, Löwenheim, Skolem, Post and others. In comparison, theoretical understanding of connectionist knowledge representation is much less advanced.

Further, we are held back by the very pervasiveness of the idea of a calculus (strengthened by familiarity with the digital computer), so that now the ideas of knowledge and calculus seem inseparable and many researchers think it is inconceivable that there can be thought without a language of thought.[1] On the other hand, progress in connectionist theory will be accelerated by exploiting the duality and analogy between discrete and continuous systems.

Just as the idea of a calculus took many years to evolve into the definitive form it took in the first half of this century, so we may also expect that the analogous connectionist concept will require refinement; the idea presented here is just a step in that direction. As evidence that it fulfills the required role, I will show how it suggests new connectionist approaches to language and structured knowledge, and how it exposes theoretical limitations of any possible connectionist system.

[1]Gestalt psychology is a distinct exception to the ubiquity of discrete models in psychology, and connectionism can be expected to restore some of the viewpoints, if not the specific theories, of the Gestalt psychologists.

The theoretical construct proposed here is called a *simulacrum*, a term which reflects the central idea of connectionist representation.[2] Just as "calculus," which means "pebble," refers by metonymy to an entire system that computes by manipulating such physical tokens, so analogously "simulacrum," which means "likeness" or "image," refers by metonymy to an entire system that computes by manipulating physical images (continuous representations). Further, just as in computer science and formal logic we distinguish between uninterpreted and interpreted calculi, the former being purely "syntactic" (formal) processes whereas the latter have a specified "semantics" (i.e., they are meaningful), so also will I distinguish uninterpreted and interpreted simulacra.

1.2 Uninterpreted Simulacra

1.2.1 Introduction

The goal of the theory of simulacra is to capture the essentials of connectionist knowledge representation and processing. Therefore we begin with the observation that connectionist representations are frequently real vectors of high dimension. This observation is reinforced by the relation between connectionism and neuropsychology, since neural representations (whether in terms of spike density or graded potential) are predominantly continuous. Therefore we take continuity — of representation and process — as the essential feature distinguishing connectionism from traditional approaches to knowledge representation (which employ discrete representations and processes).

There are many objections that can be made to this characterization, which can, I hope, be disposed of quickly. For example, it may be objected that action potentials are all-or-nothing events, and therefore essentially discrete, and that even graded potentials are in fact discrete, since they result from an integral number of vesicle exocytoses or from an integral number of charge carriers (ions etc.). Furthermore, connectionist systems are usually implemented on digital computers, and connectionist researchers sometimes work with two-state units.

These objections are all irrelevant, at least in the absence of additional information. What is relevant is the behavior of a system at the appropriate level of analysis. Since a (mathematically) discrete system can approximate arbitrarily closely a (mathematically) continuous system, and conversely a (mathematically) continuous system can approximate arbitrarily closely a (mathematically) discrete system, we see that, for biological modeling, the relevant distinction is not mathematical. What is important is the behavior — discrete or continuous — of the system at the relevant level of analysis. This *Complementarity Principle* is offered as a *razor* (like Ockham's) for cutting off fruitless debate (MacLennan, in press-b, in press-c, in press-d, in press-e).

The properties of a calculus derive from its topology, rather than from any particular concrete representation of it (e.g. in terms of bits); likewise I expect the properties of a simulacrum to derive from its topology, rather than from its specific concrete representation (e.g. in terms of vectors of reals). Therefore, the theory of simulacra can be expected

[2] Previously (MacLennan, in press-a) I called simulacra "continuous symbol systems" to stress the analogy with the familiar discrete symbol systems, but "symbol" carries such a strong connotation of discreteness, that the phrase has proved more confusing than helpful.

to be primarily topological, and in this sense qualitative rather than quantitative. In this way it will address the essentials, rather than the details, of continuous representation and processing.

A *formal process* is one that depends only on the physical characteristics (e.g., shapes) and physical relations (e.g., positions) of signs, and not on any meanings that may be associated with them; that is, the processes are purely *syntactic* and independent of *semantics.* Formality is an important characteristic of any theory of cognition, since it permits "cashing out" meaning in terms of physical phenomena.[3] Just as it is useful in traditional knowledge representation to consider the properties of uninterpreted calculi, so also it will be useful to begin by considering uninterpreted simulacra. I will present briefly the basic postulates and propositions of the theory of simulacra; a more detailed justification and discussion can be found elsewhere (MacLennan, 1993a, in press-a).

1.2.2 Image Spaces

Calculi comprise (1) a *state space* of formulas, which is defined by *formation rules*, and (2) a *state-transition process*, which is defined by *transformation rules* (e.g., MacLennan, 1990b, pp. 423–425). Simulacra likewise comprise a state space and a process. Corresponding to the formulas and tokens of calculi, in simulacra we have *images*, which are the representational vehicles of simulacra, as formulas are of calculi. Simulacra differ from calculi in the manner of definition of these spaces, since formulas are defined in terms of their construction from atomic tokens, whereas images do not have atomic constituents. The definition of image spaces will be considered later.

In the context of simulacra, "image" has a precise definition, which will be given presently. However, it is helpful to begin with an informal description. The term is intended to include familiar visual and auditory images, but in addition to these *instantaneous* images (spread out in space), I also include more abstract *spatiotemporal* images, such as the signal received by the retina or cochlea over an interval of time. The term also applies to abstract representations, such as Gabor or other wavelet-like decompositions. Further, motor and memory images are included, and, at a more physical level, electrical or chemical distributions in smaller or larger regions of the brain. We turn now to a mathematical characterization of images.

The idea of *degrees* of similarity is fundamental to any notion of images, and it seems reasonable to require that these degrees be *quantifiable.* Mathematically, it is usually more convenient to work in terms of difference rather than similarity, so we postulate a difference measure δ, holding between any two images in an image space, with the following properties:

$$\begin{aligned}\delta(\varphi,\varphi) &= 0,\\ \delta(\varphi,\psi) &= \delta(\psi,\varphi),\\ \delta(\varphi,\psi) &\leq \delta(\varphi,\chi)+\delta(\chi,\psi).\end{aligned}$$

[3]Here we are concerned that the processes be independent of any meaning attributed to them by an outside observer, so that the processes do not depend on outside agents. That is, an uninterpreted calculus or formal system has no *attributed semantics.* Clearly, however, a system intended as a cognitive model must have *intrinsic semantics*, that is, the processes must have meaning to the cognitive agent constituted of those processes. Whether purely formal systems can have an intrinsic semantics is, of course, an important question.

In mathematical terms we have a distance *metric* and so an image space is a *metric space*. Needless to say, the metric does not have to be the Euclidean distance, and in most cases it will not be. However, we will generally leave the metric (and hence the notion of similarity) unspecified.[4]

A difference metric can be converted to a similarity metric in a variety of ways. For example, if $\delta = \delta(\varphi, \psi)$ is a difference metric that can be arbitrarily large, then we can define a similarity metric $\sigma(\varphi, \psi) = 1/(1+\delta)$ such that $\sigma = 1$ for identical images and $\sigma \to 0$ as the images become less similar (their difference approaches infinity). If, as is more common, the differences are bounded, $\delta \leq d$, then a similarity metric can be defined $\sigma = (d-\delta)/(d+\delta)$, $\sigma = 1 - \delta/d$, or in many other ways; the choice should be determined by the process to be modeled by the simulacrum.

The principal characteristic that distinguishes connectionist from "symbolic" knowledge representation is that images are drawn from a continuum, whereas formulas are discrete. There are many ways such a continuum can be defined topologically, but the definition proposed here seems to agree with our intuitions about continua, namely, that any image can be continuously transformed into any other in the same space. The fundamental intuition behind this is that if it doesn't hold, then we are dealing with two or more discrete image spaces. In the extreme case where none of the points can be continuously transformed into any other, we have a completely discrete space, and we are back to calculi. A continuous transformation of one image into another is just a continuous function of time that equals one image at one time and another image at another time. In mathematical terms, this is a *path*, and so we are assuming that an image space is *path-connected*. All the foregoing characteristics of image spaces are summarized in the first postulate:

Postulate 1 *All image spaces are path-connected metric spaces.*

The second postulate is more technical:

Postulate 2 *All image spaces are separable and complete.*[5]

Separability provides a link to the discrete, since it means there is a countable set of images that can be used to approximate, arbitrarily closely, any image in the space. Completeness is really the converse property, since it means that if we construct a sequence of increasingly similar images (i.e., a Cauchy-convergent sequence), then the limit will exist in the space. More precisely, if we construct a sequence of images $\varphi_1, \varphi_2, \varphi_3, \ldots$ so that $\delta(\varphi_j, \varphi_k) \to 0$, then there will be a limiting image φ to which this sequence converges, $\delta(\varphi, \varphi_k) \to 0$.

One of the reasons that calculi are interesting is that they are, at least potentially, implementable by purely mechanical means. In theory we can simulate any program by hand, though in practice it might exceed the capacities of the largest computer. The reader

[4]These assumptions are hardly unproblematic. It is not immediately apparent that all notions of difference are quantifiable, and it is certainly questionable whether all measures of difference satisfy the triangle inequality, $\delta(\varphi, \psi) \leq \delta(\varphi, \chi) + \delta(\chi, \psi)$. Nevertheless, the class of metric spaces is extremely large, and so the assumption of a metric seems permissible as a hypothesis, if for no other reason. See also MacLennan (1988, in press-a).

[5]In practice, most image spaces are (topological) *continua*, and so *compact*, which implies that they are both separable and complete. Compactness is not postulated because unbounded image spaces are sometimes useful.

might criticize simulacra for being too abstract and losing all touch with physical realizability. In this regard the following proposition is significant:

Proposition 1 *Every image space is topologically equivalent (homeomorphic) to a subset of a Hilbert space.*[6]

That is, there is a homeomorphism — a one-to-one relation that is continuous in both directions — between the image space and the subset of the Hilbert space. Although the metric of the image space is not preserved by the homeomorphism, it is recoverable from the elements of the Hilbert space.

This connection with Hilbert spaces is important for a number of reasons. First, elements of Hilbert spaces can be represented by (possibly infinite-dimensional) vectors of real numbers, so this proposition implies that images can be represented by the vectors manipulated by neural networks and thus are physically realizable.[7] Second, Hilbert space has proved to be a valuable framework in which to study visual and auditory perception, and it is beginning to show its value for modeling general neurodynamics, especially of the very-high-dimensional dendritic interactions (e.g., Pribram, 1991; MacLennan, 1990a, in press-b, in press-c). Finally, in a mathematically precise sense, the continuous meets the discrete in Hilbert space (thanks to the Riesz-Fischer theorem: the square-summable sequences are isomorphic and isometric to the square-integrable functions); therefore, if we want to understand the emergence of apparently discrete symbolic processes from continuous neurodynamics, then Hilbert space is probably the place to start.

1.2.3 Image Maps

In connectionist knowledge representation, most maps between image spaces are represented by neural nets that implement continuous functions. Furthermore, the continuous relation between input and output is one of the reasons neural nets avoid some of the brittleness characteristic of rule-based systems; we don't expect an infinitesimal change in the input to lead to more than an infinitesimal change in the output. Therefore we adopt:

Postulate 3 *Maps between image spaces are continuous.*

This postulate has a number of interesting consequences, which are especially significant for models of cognition. First, all categories recognized by simulacra are "fuzzy," since all maps between image spaces are required to be continuous, and therefore there must be a gradation of responses between the images that are in a class and those are not. To put it another way, categories in simulacra are ineluctably indeterminate, since as we approach

[6]This follows from a theorem of Urysohn, which shows that every metric space with a countable base is homeomorphic to a subset of the Hilbert space $\mathcal{L}_2(\mathbf{R})$ (Nemytskii & Stepanov, 1989, p. 324). This applies to image spaces because, by Postulate 2, they are separable, and so they have a countable base. The result is even stronger, since the proof shows that image spaces are homeomorphic to subsets of the space Q^∞ of fundamental parallelopipeds.

[7]In fact, because the image space is homeomorphic to a subset of the space of fundamental parallelopipeds (see previous footnote), we know that the infinite-dimensional vectors can be approximated arbitrarily well by finite-dimensional vectors, thus providing a direct link between physically realizable finite-dimensional vectors and mathematically simpler infinite-dimensional vectors.

the border of the category, the response to its members must come arbitrarily close to the response to its nonmembers. We cannot have a simulacrum that responds one way to avocados and a completely different way to non-avocados; there must be gradation of response from avocados to non-avocados. In summary:

Proposition 2 *An image-map cannot produce distinctly different responses to a set and its complement.*

It might be objected that this result is psychologically unrealistic, since we apparently make categorical discriminations all the time: from identifying phonemes to deciding whom to marry. This evidence does not contradict the proposition, since the Complementarity Principle tells us that we cannot distinguish a discrete category boundary from an arbitrarily sharp continuous boundary. Furthermore, it can be argued on physical grounds that between two definite responses there must always be a region which is neither (because a connected space cannot be the union of two basins of attraction; see Proposition 3, below).

We have seen that simulacra do not permit exact classification functions (i.e., functions that are 1 on the class and 0 outside the class), so we must ask what kinds of categories are permitted. Fortunately, a number of "separation theorems" from topology ensure us of the existence of classification functions that are useful and realistic, both psychologically and physically.

For example, Urysohn's Lemma tells us that for two disjoint, closed sets, A and B, we can have a classification function that is 0 on A and 1 on B. However, since a connected space, such as an image space, cannot be the union of two disjoint, closed sets, there must be other images in the space other than those in A and B, and for these continuity requires the classification function to give values between 0 and 1, that is, indeterminate classifications. Thus the theory of simulacra does not preclude a system that responds one way to definite avocados and another to definite bananas. It simply requires that things that are neither definite avocados nor definite bananas must elicit a response that grades into that for definite avocados on one side and that for definite bananas on the other.

We are also permitted a classification function that is 0 for definite nonmembers of the category and is equal to 1 for a finite number of images that are definite members (exemplars), but continuity requires that some other members have classifications arbitrarily close to 0. For example, we may be quite sure that the *Iliad*, the *Odyssey*, the *Nibelungenlied*, etc. are epic poems (since they effectively define the class), and there are countless things that we know are not epic poems (Shakespeare's sonnets, me, the moon, etc.), but there must also be epics, actual or possible, that are arbitrarily similar to non-epics, and about which we are thus unsure.

1.2.4 Image Processes

Although recurrent neural networks are often implemented by discrete-time simulations, they are usually treated as approximations of continuous-time gradient systems, and hardware implementations are often based on continuous-time relaxation processes. Therefore we adopt:

Postulate 4 *Formal processes in simulacra are continuous functions of time and process state.*

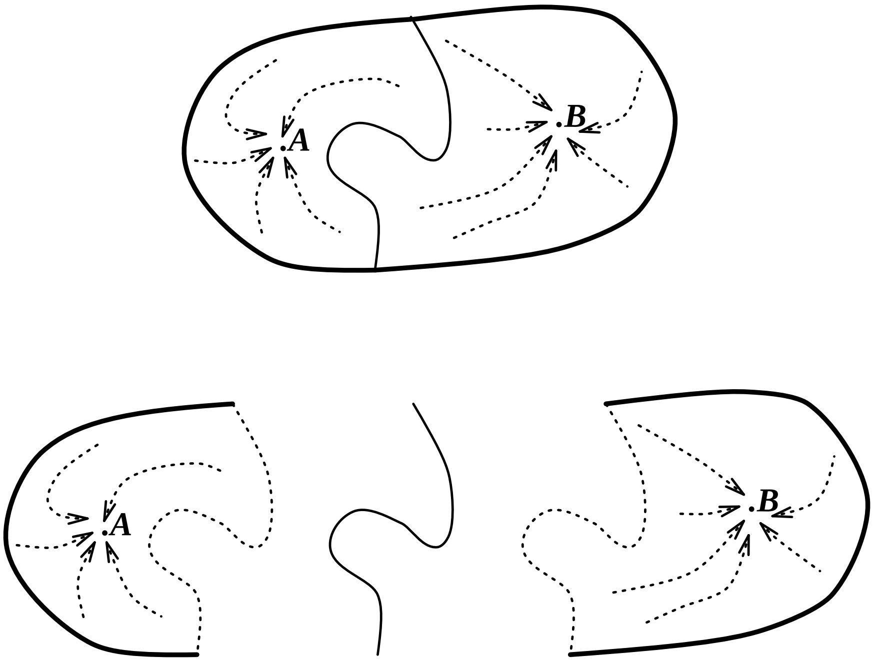

Figure 1: Basins of attraction are separated sets. The upper figure shows a connected state space for a process with two attractors, A and B. Since the basins of attraction of a continuous process are open sets, and a connected space cannot be the union of disjoint open sets, there must be a set of states not in any basin of attraction, such as the line of "metastable states" shown in the lower figure.

This is really a corollary of Postulate 3 if we take the differential equations defining the process to be a map between image spaces (a function of the process state and the differential of the process state).

Formally we adopt a more general notion of process than that provided by differential equations. A continuous function $P : S \times \mathbf{R} \to S$ is called a process over a state space S (an image space) if $P(s, 0) = s$ and $P[P(s,t), t'] = P(s, t + t')$. Intuitively, $P(s,t)$ represents the state of the process time t after starting in state s.[8]

The continuity of processes implies that their *basins of attraction* are open sets (roughly, they do not include their boundaries).[9] Since a connected space cannot be the union of two or more disjoint open sets, we have an important result:

Proposition 3 *If an image-process has multiple basins of attraction, then there are process states that are not in any basin of attraction.*

This means that any decision procedure implemented by a continuous process must leave certain problems undecided (Fig. 1). (This conclusion holds for any continuous process over a connected state space, not just for simulacra.)

[8]What we refer to here are deterministic simulacra; the notion of nondeterministic simulacra is also useful for some purposes. In these cases equations or inequalities define possible trajectories in state space.

[9]Suppose $P(s,t) \to \bar{s}$, that is, s is in the basin of attraction of $\bar{s}$. Then no matter how closely we wish a trajectory to stay to $P(s,t)$ and no matter for how long, the continuity of P implies that we can always find a neighborhood of p in which all the trajectories have this property. That is, points infinitesimally near to p are in the same basin of attraction, which is thus open (Hirsch & Smale, 1974, pp. 190–191).

1.3 Interpreted Simulacra

A calculus can be "interpreted" (assigned a meaning) by systematically associating meanings with its formulas and rules of derivation. Likewise, a simulacrum is interpreted by systematically associating meaning with its images and processes. For both calculi and simulacra, the basic requirement of systematicity is mathematical continuity. For simulacra this is an intuitive requirement, since we would expect infinitesimally small changes in an image to be associated with infinitesimally small changes in its meaning. For calculi the connection with continuity is less obvious; let it suffice here to observe that Scott has shown how to put a topology on a space of formulas in such a way that they are partially ordered in accord with their constituent structure (e.g., Scott, 1971, 1973).

Postulate 5 *Interpretations of simulacra are continuous.*

This postulate has a number of interesting implications. First recall that the formation rules of a calculus define a set of *well-formed formulas* (i.e., syntactically correct formulas), which is typically a proper subset of the set of all possible formulas (strings of atomic symbols). Meaning is attached to only the well-formed formulas; the ill-formed formulas remain uninterpreted.

The situation is somewhat different for simulacra. It is analogous in that normally only a subset of the images in an image space will be considered "well-formed." However, "well-formed" and "ill-formed" are categories, and so they must conform to the limitations on classification by simulacra. Thus there must be a continuous variation between the response to interpretable images and the response to uninterpretable images.

For example, we may have a closed set of definitely well-formed (interpretable) images and a definite set of ill-formed (uninterpretable) images, but between these must be a nonempty set of images that approximate the well- and ill-formed images on each side, and are to that extent more or less interpretable. One may think of the "confidence" of interpretation decreasing from 1 for the well-formed images to 0 for the ill-formed.

Actual (versus *idealized*) written or spoken language has exactly these properties. A signal can contain a certain amount of noise and still be interpreted correctly (within the error-correction capabilities of the receiver); also some signals are completely uninterpretable and must be considered pure noise. However, between these two extremes are many signals in which the noise interferes with interpretation, but doesn't preclude it.[10]

To account for the continuity of response from interpretable images to uninterpretable images, we require that the interpretation function be total, that is, defined for all images, whatever their degree of interpretability. Therefore, domains of interpretation for simulacra must include one or more "undefined objects," which represent the response to ill-formed images (Fig. 2).[11]

[10]For example, a digital code might provide 2-bit error correction. In this case transmissions with less than two bits wrong will be perfectly interpretable. Transmissions with more than two bits wrong may be detectable as incorrect but uninterpretable.

[11]This is only an apparent difference from interpretations of calculi, in which only the well-formed formulas are taken to be interpreted. For if the topology of the formulas is taken into account, then, as Scott (e.g., 1971, 1973) has shown, it is necessary to adjoin an "undefined object" ($\perp$, the *bottom* of a complete lattice) to the domain of interpretation.

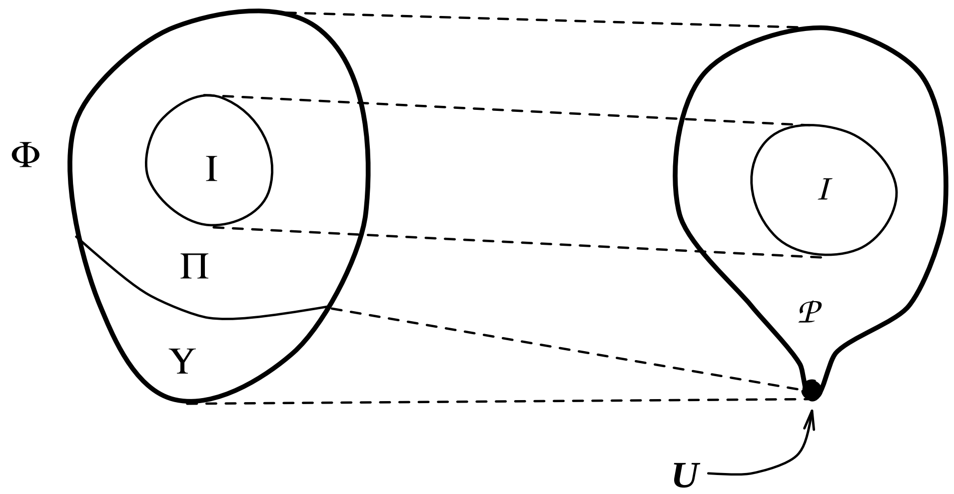

Figure 2: An interpretation must be total and continuous on the image space Φ, so it must assign a response that varies continuously from definitely interpretable images I through partially interpretable images Π to uninterpretable images Υ. This could be accomplished by adjoining a special "undefined object" U to the domain of interpretation. (For clarity the interpretable images are depicted as a single connected set I; more likely there are many "islands" of interpretable images.)

Since the domain of interpretation is a continuous image of an image space, the total domain of interpretation of a simulacrum must always be a continuum. This does not preclude there being a discrete set of full-fledged interpretations, these being infinitesimally close to partial interpretations. So, from a practical standpoint, discrete interpretations are indistinguishable from continuous interpretations. This characterization of the domain of interpretation has important logical implications (not considered here), but from the cognitive standpoint permits the same image may be interpreted or not depending on context (e.g., the need to find an interpretation).

2 Invariants

One obvious advantage of traditional, symbolic methods is in describing rule-like or regular behavior, so we must consider analogous phenomena in continuous computation. These are best characterized in terms of invariants in the behavior of a system. The following description is completely behavioral, with no discussion of how these invariants come to be (i.e., learning is not addressed).

2.1 Category Invariants

2.1.1 Definition

A category invariant is manifested by a system when it responds the same to a variety of stimuli, thus defining a category or class of stimuli.[12] However, saying that the system "responds the same" must include the possibility that the response depends on the individual stimulus, considered not as an individual but as a member of the category. For example, if Rover chases cats in his yard, without regard for the individual cat, then we say Rover is behaving invariantly with respect to cats in his yard. It is crucial however that the cat that Rover chases be the one that's in his yard. That is, the response is determined by the category and the context, but particularized to the category member.[13] More precisely, the context C determines an action A, so if the stimulus is $C(x)$, the context particularized to category member x, then the response will be $A(x)$, the categorical response particularized to that same category member. This is just the sort of situation that is described by a conventional "condition-action rule," $C(\mathrm{K}) \Rightarrow A(\mathrm{K})$, meaning, "if you see a K in context C, then perform action A on that K" (e.g., MacLennan, 1990b, pp. 424–425).[14] We can now put this definition in mathematical terms.

Definition (Category Behavior) Let Φ, X, Ψ be image spaces and let K be a proper subset of X. Then we say a system $S : \Phi \to \Psi$ *displays category behavior with respect to members of* K *in context* $C : \mathrm{X} \to \Phi$ *and for action* $A : \mathrm{X} \to \Psi$ if and only if for all $x \in \mathrm{K}$, $S(C(x)) = A(x)$. More concisely: $S \circ C|_{\mathrm{K}} = A|_{\mathrm{K}}$, where $F|_{\mathrm{K}}$ denotes the restriction of function F to K. (As usual, all maps are required to be continuous.) □

Intuitively, Φ is the stimulus space, Ψ the response space, and K a category with respect to a universe X of potential category members. The map C creates a stimulus by placing an object in a context, and the map A is the categorically determined action (Fig. 3).

2.1.2 Indefinite Boundaries

Consider the consequences of this definition. First, as would be expected from the postulates, the boundaries of categorical behavior are necessarily fuzzy. That is, because S and C are both continuous (and so also the composition $A|_{\mathrm{K}} = S \circ C|_{\mathrm{K}}$), the system must respond to nonmembers x just outside the category with actions close to $A(x)$. For example, as noncat images approximate cat images arbitrarily closely, in Rover's yard, so also will his behavior approximate chasing those things. Thus, the rule $C(\mathrm{K}) \Rightarrow A(\mathrm{K})$ is an incomplete description of the behavior of the system, since the rule must be automatically generalized to similar nonmembers of K.

[12] Although this definition and others to follow are phrased in terms of stimulus and response, they are intended to be more general than the stimulus/response behavior of a complete organism. Thus they may refer to the input and output of some cognitive faculty, and so, for example, the "stimulus" could be a mental image, or the "response" some other internal representation.

[13] We will see later that the case where the response is not particularized can be handled as a special case of the general formulation.

[14] Note, however, that we are dealing with continuous images, not discrete formulas. For example, the map C, which creates an image $C(x)$ from the image x, may not be simply expressible as a template.

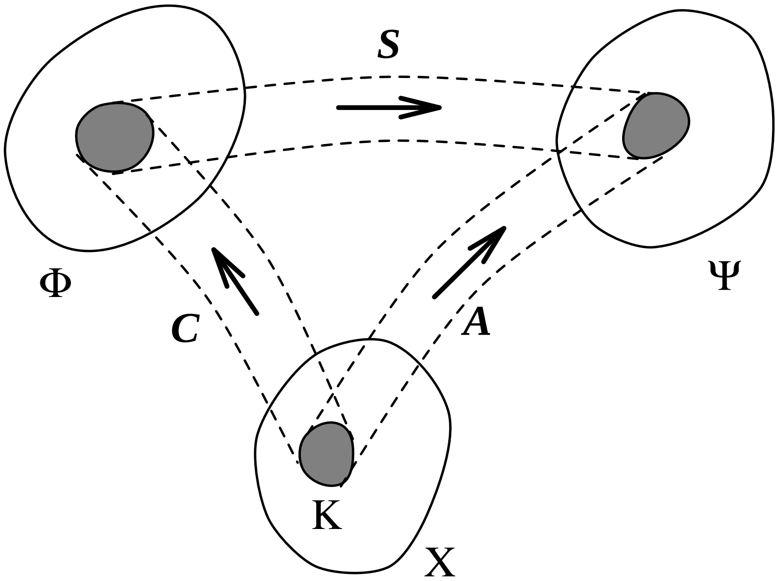

Figure 3: Category behavior. Φ is an image space of "stimuli," Ψ is an image space of "responses," and X is a space of images of potential category members. The system S maps stimuli in Φ into responses in Ψ. The function C maps a potential member x into a stimulus $C(x)$, and the function A maps a potential member x into a response $A(x)$. The functions S, A and C are required to be continuous. The system S displays category behavior with respect to members of K if for every member x of K the response to stimulus $C(x)$ is action $A(x)$, thus exhibiting behavior described by the rule $C(\mathrm{K}) \Rightarrow A(\mathrm{K})$.

Given the inherent indeterminacy or fuzziness in the border of K, it is somewhat misleading to speak of absolute category behavior. Since continuity requires that category behavior fade gradually as we move out of K, it is better to quantify the degree of invariance and hence the degree of membership in the category. To do this we can measure how much the actual response $S(C(x))$ differs from ideal category behavior $A(x)$. Specifically,

$$\check{\mathrm{K}}(x) = \delta[S(C(x)), A(x)].$$

The function $\check{\mathrm{K}}(x)$ is zero for those images for which the system exhibits perfect category behavior, and increases as the behavior is less characterizable in terms of a category behavior in terms of the rule $C(x) \Rightarrow A(x)$. If, as in fuzzy set theory, it is preferred to have a measure that equals one for perfect category members and decreases to zero as the responses differ from category behavior, then use a similarity rather than a difference metric:

$$\hat{\mathrm{K}}(x) = \sigma[S(C(x)), A(x)].$$

The σ and δ functions can be related in any of the ways previously described. Since $\hat{\mathrm{K}}(x)$ quantifies the degree to which behavior follows the rule $C(x) \Rightarrow A(x)$, it can be called the *regularity* of the system with respect to $C(\mathrm{X}) \Rightarrow A(\mathrm{X})$.

An especially important case occurs when the regularity is defined in terms of the response to a single exemplar α (the extension to multiple exemplars is trivial). The exemplar defines a *de facto* (indefinite) category $\hat{\mathrm{E}}_\alpha$, membership in which is measured by the similarity of the response to that evoked by the exemplar:

$$\hat{\mathrm{E}}_\alpha(x) = \sigma[S(C(x)), S(C(\alpha))].$$

2.1.3 Relation to Rules

I will consider briefly the effect that various properties of C and A can have on the form of category behavior. In many cases C will be a one-to-one function, which (because of continuity) implies that the space $C[\mathrm{K}]$ of images constructed from category K is topologically equivalent (homeomorphic) to the category.[15] In this case the construction is invertible, and the object can be unambiguously separated from its context; that is, if $\varphi = C(x)$, then $x = C^{-1}(\varphi)$. In this case the behavior of the system on $C[\mathrm{K}]$ is given by $S|_{C[\mathrm{K}]} = A \circ C^{-1}|_{C[\mathrm{K}]}$, that is, doing action A on the object extracted from context C. On the other hand, it is not necessary to assume that C is one-to-one, for in constructing an image $C(x)$ in Φ some distinguishing features of the category members may be lost. However, if this is the case, the equation $S \circ C|_{\mathrm{K}} = A|_{\mathrm{K}}$ requires that these features also have no role in determining the response image.

The special case where A is a constant function describes the situation in which the system's categorical response is *not* particularized to the category member. That is, if $A(x) = \psi$ for all $x \in \mathrm{K}$, then the system shows response ψ for any member of the category, $S[C[\mathrm{K}]] = \{\psi\}$. An example would be if Rover responds to a cat in his yard by barking, without the action being directed at that cat or any other.

A rule of category behavior $C(\mathrm{K}) \Rightarrow A(\mathrm{K})$ describes the behavior of a system over a restricted subset $C[\mathrm{K}]$ of its stimulus space. However, there is nothing preventing the behavior over this same subset from being described by a different rule, $C'(\mathrm{K}') \Rightarrow A'(\mathrm{K}')$. Figure 4 shows the simplest case, where $C[\mathrm{K}] = C'[\mathrm{K}']$. For example, a person who regularly separates a cat and dog that are fighting could be described as exhibiting category behavior either with repect to the class of dogs (in the context of fighting a cat), or with respect to the class of cats (in the context of fighting a dog).

More generally the sets $C[\mathrm{K}]$ and $C'[\mathrm{K}']$ derived from these categories can overlap in various ways, so long as the system S behaves consistently in the area of overlap. Thus if a stimulus φ can be described either as a member of category K in context C or as a member of category K' in context C', then for any category members $x \in \mathrm{K}$ and $x' \in \mathrm{K}'$ yielding this stimulus, $C(x) = \varphi = C'(x')$, we must have the same response $A(x) = A'(x')$.

Of course, a system S can display categorical response to a number of alternative categories $\mathrm{K}_1, \ldots, \mathrm{K}_n$ in a given context C with respect to action A. All that is required is that the equations $S \circ C|_{\mathrm{K}_j} = A|_{\mathrm{K}_j}$ hold. However, the necessity of continuity places a number of constraints on such alternative categorical behavior. To understand these, consider again the case of just two categories, K and K' (Fig. 5).

Obviously, if these categories overlap, then the system must behave consistently on the area of overlap. What may not be obvious it that there are constraints on the nonoverlapping areas; indeed there are even constraints on disjoint categories. For example, response to elements of K is not independent of response to elements of K' unless the categories are

[15] As usual, we use $C[\mathrm{K}]$ for the image of the set K under the function C, i.e. $C[\mathrm{K}] = \{C(x) \mid x \in \mathrm{K}\}$.

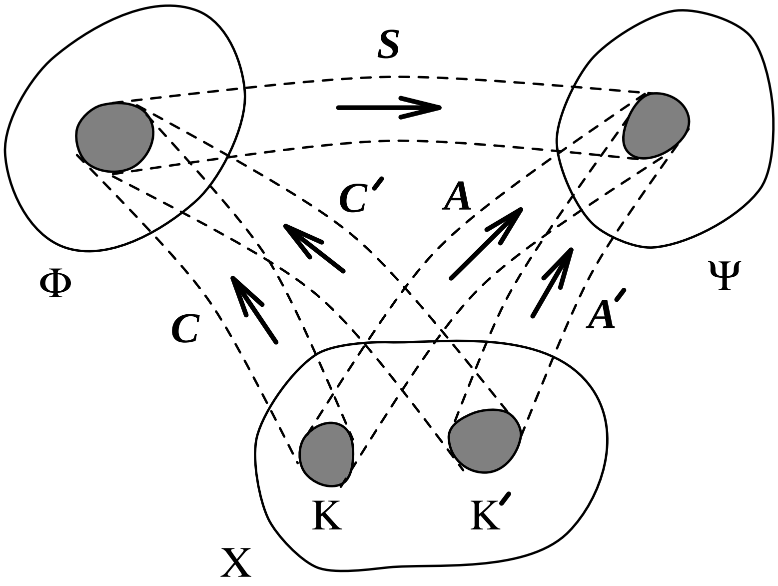

Figure 4: The same pattern of behavior may be described in different ways, for example, as category behavior with respect to category K or category behavior with respect to K′.

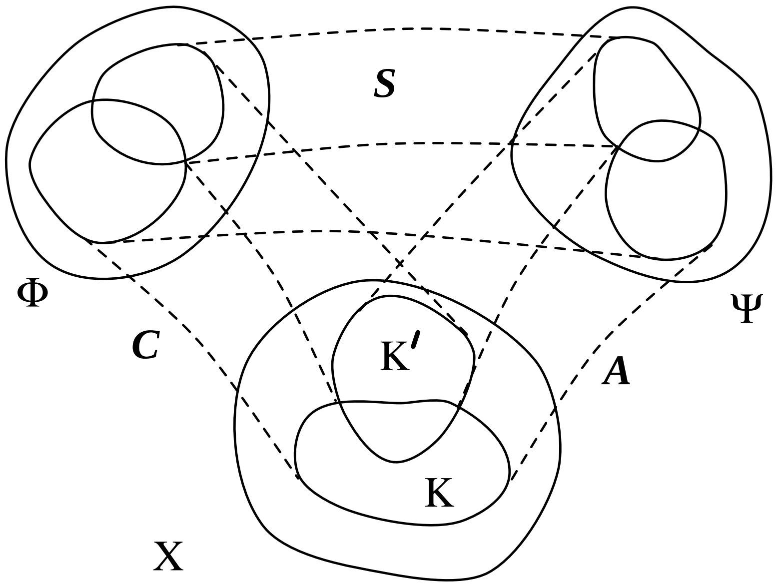

Figure 5: Multiple category behavior. There are constraints, dictated by continuity, on how a system may display category behavior with respect to more than one category in a single context.

separated. Informally, independence requires that there be some "space" between the categories. Continuity requires that members of one category that approach sufficiently closely to members of the other must yield responses that likewise approach the responses to those members of the other.

Another respect in which this "rule-like" behavior differs from traditional "rule-based" behavior, is that the rule $C(\mathrm{K}) \Rightarrow A(\mathrm{K})$ is descriptive of a limited region of the total behavior S of the system. That is, the total behavior S is the given, and the rule is just a partial accounting for it. In contrast, in a conventional rule-based system the behavior of the whole is the sum of the behaviors of the individual rules. We may say that whereas in a rule-based system the total behavior is a construction of the individual rules, here we may have many rule-like deconstructions of the total behavior, none of which may capture the whole.

An interesting question — with relevance for the translation of trained neural networks into rules — is when the behavior of a system can be completely described by rules.

One answer follows from the separability of any image space, which means that it has a countable, dense subset (comparable to the rationals being a countable, dense subset of the reals). To see how this allows the reduction to discrete rules, let Φ, Ψ and $S : \Phi \to \Psi$ be given. Then let $\{\varphi_1, \varphi_2, \ldots\}$ be such a countable, dense subset of Φ and, letting $\mathrm{X} = \Phi$, define a countably infinite list of categories to be the singleton sets $\mathrm{K}_j = \{\varphi_j\}$. Letting $C(x) = x$ and $A = S$, we have a countable infinity of rules $C(\mathrm{K}_j) \Rightarrow A(\mathrm{K}_j)$ defining S on the images φ_j. Since continuous functions preserve limits of convergent sequences, the behavior on the rest of the images is filled in automatically.

Let's be clear on what this result tells us. It says there is a countable infinity of rules of the form $\varphi_j \Rightarrow S(\varphi_j)$ that approximate arbitrarily closely to the behavior of S on any image. One problem is that an infinite number of rules are required, though it is not hard to see how finite sets of rules approximate this totality. The second problem is that these "rules" are expressed in terms of the system behavior, which may not be a discrete formula of the usual kind. The third problem, of course, is that singleton categories do not lead to a very interesting kind of category behavior.

2.2 Relational Invariants

The foregoing analysis is easily extended to cases in which the system exhibits category behavior with respect to two or more categories simultaneously. In this case the context is a relation between members of the categories, and the action, though determined by the relation, is particularized to those members. Such a system responds to an image putting members x and y of categories K and Λ in a relation $R(x, y)$ with an output image $A(x, y)$ that is determined by R and the category members. For example a person might respond to any cat and dog in a relation of proximity by separating the two. Such behavior is most simply expressed by a rule of the form $R(\mathrm{K}, \Lambda) \Rightarrow A(\mathrm{K}, \Lambda)$. Now we state this mathematically.

Definition Let $\Upsilon, \Phi, \mathrm{X}$, and Ψ be image spaces, and let K and Λ be proper subsets of X and Υ, respectively. Then we say a system $S : \Phi \to \Psi$ *displays category behavior* with respect to members of K and Λ in relation $R : \mathrm{X} \times \Upsilon \to \Phi$ and with respect to action $A : \mathrm{X} \times \Upsilon \to \Psi$ whenever for all $x \in \mathrm{K}$ and $y \in \Lambda$, $S(R(x, y)) = A(x, y)$. More concisely, $S \circ R|_{\mathrm{K} \times \Lambda} = A|_{\mathrm{K} \times \Lambda}$. □

The definition is easily extended to three or more categories. Everything we said about the effects of continuity in the single category case applies as well to multiple categories.

3 Structure

3.1 Introduction

The *structure* of a thing refers to the organization of its parts. For discrete knowledge representations, structure refers to the (usually hierarchical) syntactic relations between a formula's component parts. Further, in discrete representations the structure is usually manifest. In continuous representations, in contrast, it is not apparent what should be considered the "parts" of an image, or what sorts of relations should be considered "syntactic." Indeed, a common criticism of connectionist knowledge representations is that they have no constituent structure (e.g. Pinker & Prince, 1988). Thus we must investigate the structure of images. Since we are proceeding in unfamiliar territory, it will be worthwhile first to consider structure in discrete representations more closely.

Discrete representations are usually described generatively, that is, by giving formation rules that describe the construction of the allowable formulas from atomic components. The discrete atoms provide a natural starting point for synthesis and also an ending point for analysis. Further, we usually prefer unambiguous languages, so that each formula has a unique decomposition. (Though even for ambiguous languages the atoms are unambiguous.)

The situation is quite different for images, since there are in general many ways that an image can be decomposed into components, and different decompositions may be appropriate for different purposes. For example, a visual image of a face can be decomposed into anatomical features, pixels, edges, elementary polygons, wavelets, and so forth. Even a one-dimensional image can be decomposed into a set of points, or a set of straight line segments, or a superposition of powers (as in polynomial approximation), or a superposition of sinusoids (as in Fourier decomposition), etc. There is not, in general, a *privileged decomposition* as there is for discrete representations.

Though the preceding examples are infinite-dimensional, even a finite-dimensional image can usually be represented in terms of several different (possibly nonorthogonal) axes. Evidence that the appropriate decomposition is not apparent comes from the use of techniques such as principal components analysis and multidimensional scaling in the analysis of perceptual data (Shepard, 1980).

The multiple-decomposibility of images implies, in effect, that images are inherently ambiguous; they have no privileged structure. Given this ambiguity, we might hope that images can be reduced to a determinate set of atoms, but this is not the case. Since most images are infinite dimensional, they can always be further subdivided. Therefore, most image spaces cannot be defined bottom-up (generated from atoms), but must be specified top-down as a field of variation within a background. (We consider some of the methods later.) In summary: images are in general inherently ambiguous and atomless. This then is a key characteristic of connectionist knowledge representation.

3.2 Syntactic Invariants

Syntactic types and relations are special cases of the categorical and relational invariants we have already considered. For example, if the categories A and B represent the syntactic types 'A' and 'B' (that is, the set of all letter-A tokens and letter-B tokens, respectively), then the rule

$$C(\mathrm{A},\mathrm{B}) \Rightarrow C(\mathrm{B},\mathrm{A})$$

describes a process that exchanges an 'A' and a 'B' when they occur in the context $C(A, B)$, for example, when the 'A' precedes the 'B'. As expected, images close to 'A's and 'B's will evoke responses close to those described by the rule. But what about syntactic relations (compound, vs. atomic, types) that are close to C? Just as some variability is allowed in the shape of the token without changing its type, so also variation is allowed in syntactic relations without changing their type. For example, the syntactic type *'A' preceding 'B'* normally admits some variation in the vertical alignment and position of the A and the B. This suggests that we must consider the topology of the space $\mathcal{C}$ of contexts, so that we can describe the effect of variations in $C \in \mathcal{C}$.

To understand the problem, consider a simpler example. Suppose $n(x)$ represents the syntactic operation of putting a not-sign in front of the image x. For example,

$$n\left(\boxed{\text{mortal(Socrates)}}\right) = \boxed{\neg\text{mortal(Socrates)}}.$$

I will suppose this operation is defined so that $n(n(x))$ puts two not-signs in front of x.[16] One might expect that the process of removing double not-signs from any image in X could be described by a rule such as $n(n(\mathrm{X})) \Rightarrow \mathrm{X}$, but this is inadequate, since it guarantees the correct outcome only for the exact image constructed by n. More likely we want to allow a range $\mathcal{N} \subset \mathcal{C}$ of syntactic operations, all of which can be considered the prefixing of a not-sign. Thus a more realistic rule for removing double negations is $\mathcal{N}(\mathcal{N}(\mathrm{X})) \Rightarrow \mathrm{X}$, which applies to any image in X and any syntactic operation in $\mathcal{N}$.

To understand this idea better, suppose that the applicable syntactic operations in $\mathcal{C}$ are parameterized by $\gamma \in \Gamma$. Thus there is a one-to-one transformation $T : \Gamma \to \mathcal{C}$ such that T_γ is a syntactic operation. For example, the parameter γ might be a vector describing a relative position, and then $T(\gamma, x)$ would be the operation of attaching to x a not-sign at position γ relative to x. A certain set N of these γ will constitute *prefixing* a not-sign, and so the category of 'not'-prefixing operations will be an homeomorphic image of the allowed parameter values, which I'll write $\mathcal{N} \cong T[\mathrm{N}]$. Thus, the rule for eliminating double negation, $\mathcal{N}(\mathcal{N}(\mathrm{X})) \Rightarrow \mathrm{X}$, can also be expressed:

$$T(\mathrm{N}, T(\mathrm{N},\mathrm{X})) \Rightarrow \mathrm{X}.$$

We can see that T is a relational invariant on the product space $\Gamma\times\mathrm{X}$, which describes category behavior on the subspace $\mathrm{N}\times\mathrm{X}$.[17]

[16]For simplicity, assume $n : \mathrm{X} \to \Phi$ and $\mathrm{X} = \Phi$. This permits recursive construction of "not-trees," the mathematical requirements for which are taken up in the next section.

[17]We cannot, however, assume that the topology on $\mathrm{N}\times\mathrm{X}$ is the usual product topology, or that the metric is the Euclidean combination of the metrics on Γ and X (i.e., $\delta^2_{\Gamma\times\mathrm{X}} = \delta^2_\Gamma + \delta^2_\mathrm{X}$). However, the product topology is the weakest one that yields continuous projections from $\Gamma\times\mathrm{X}$ to Γ and X, and so any permissible topology on $\Gamma\times\mathrm{X}$ must be a refinement of the product topology.

I will consider briefly the topology of the parameter space Γ, or equivalently the topology of the space $\mathcal{C}$ of syntactic operators. First observe that for any fixed γ, the operator $C = T_\gamma : \mathrm{X} \to \Phi$ is required to be continuous (by Postulate 3). Furthermore, since we want $T_\gamma(x)$ to vary infinitesimally with infinitesimal variations of γ, it is clear that $T(\gamma, x)$ is continuous in both of its parameters. Therefore, any topology may be permitted on Γ so long as $T(\gamma, x)$ is continuous in γ.[18]

If N is the category of negation-type syntactic operations and A is the category of A-type images, then N$\times$A represents the category of syntactic relations that constitute prefixing not-signs to 'A' tokens.[19] It is thus a continuous analog to a parse tree. To see this, note that a discrete formula such as $n(n(A))$ corresponds to a tree with two nodes labeled n and a leaf labeled A; this *parse tree* represents the structure of the formula. The set of allowable formulas and the set of allowable parse trees are isomorphic (for an unambiguous language). The same holds in the continuous case, except that the expression N(N(A)) denotes a set of formula images that corresponds to the set of structural images N$\times$N$\times$A, which represent the structure of the formulas. Thus N$\times$N$\times$A represents a category of "double negations of A." As always, the boundaries of such a category are indefinite, since behavior must be continuous across the boundary. This provides a (quantifiably) indefinite structural description and a corresponding (quantifiably) indefinite notion of syntax. In this way "grammars" for indefinite languages can be defined.

3.3 Recursive Structure

Recursive structure has been claimed to be the hallmark of language and cognition (e.g., Fodor & Pylyshyn, 1988; Pinker & Prince, 1988), and therefore there has been some effort to show that neural networks can manipulate recursive structures (e.g., Pollack, 1988, 1989; Dolan & Smolensky, 1989). Although this claim is not beyond criticism, it nevertheless will be worthwhile to consider some of the topological constraints on continuous representations of recursive structure.

3.3.1 Mathematical Considerations

For simplicity, consider an image space B of binary trees. To allow the recursive construction of binary trees, we must have an operation $\mathsf{cons} : \mathrm{B}\times\mathrm{B} \to \mathrm{B}$ that takes two trees $\varphi, \psi \in \mathrm{B}$ and constructs a binary tree $\mathsf{cons}(\varphi, \psi) \in \mathrm{B}$ from them. Further, we must have means for extracting the left and right subtrees of a tree, that is, operations $\mathsf{left}, \mathsf{right} : \mathrm{B} \to \mathrm{B}$ satisfying the identities:

$$\mathsf{left}[\mathsf{cons}(\varphi, \psi)] = \varphi, \quad \mathsf{right}[\mathsf{cons}(\varphi, \psi)] = \psi.$$

This is no different from what we have in discrete computation. The postulates of simulacra, however, require that cons be a continuous map from $\mathrm{B}\times\mathrm{B} \to \mathrm{B}$, and likewise that left and right be continuous from B to B. The result is that cons is a continuous one-to-one map

[18] This will be the case, for example, if the metric on the operator space is the supremum metric, $\delta_{\mathcal{C}}(C, C') = \sup\{\delta_\Phi(C(x), C'(x)) \mid x \in \mathrm{X}\}$, where δ_Φ is, without loss of generality, a bounded metric on Φ. The same applies for a metric on the parameter space Γ.

[19] Strictly speaking, $\mathcal{N}$ is the category of negation-type syntactic operations and N is the corresponding set of parameters, but it is useful to treat the parameters $\gamma \in \mathrm{N}$ as equivalent to the operations $T_\gamma \in \mathcal{N}$.

from B×B to B with a continuous inverse. This means that the spaces B×B and B are topologically equivalent, B×B ≅ B.

In connectionist models we are usually dealing with finite-dimensional Euclidean spaces, $\mathrm{B} = \mathcal{E}^n$, therefore recursive tree construction requires $\mathcal{E}^{2n} \cong \mathcal{E}^n$, since $\mathcal{E}^n \times \mathcal{E}^n = \mathcal{E}^{2n}$. For example, if our trees are represented by 100-element vectors, then tree construction must create a 100-element vector from two 100-element vectors; likewise deconstruction must extract two 100-element vectors from a single 100-element vector. In addition, both operations must be continuous. Unfortunately, Brouwer's Theorem of the Invariance of Dimensionality shows that this is impossible; $\mathcal{E}^m \cong \mathcal{E}^n$ if and only if $m = n$ (Hausdorff, 1957, p. 232). That is, we cannot have continuous recursive construction of trees over a finite-dimensional Euclidean space.[20]

A way around Brouwer's theorem is suggested by the observation that $\mathcal{E}^{2n} = \mathcal{E}^n$ when $n = \infty$. Indeed, as will be shown presently, recursive tree construction is possible in Hilbert space (an infinite-dimensional Euclidean space), so it will be worthwhile to consider the practical significance of infinite-dimensional spaces, since actual neural nets are finite-dimensional. However, if the number of units in a net is sufficiently large, then it often can be treated mathematically as though it is infinite (MacLennan, 1987), and a neural net of such size will allow the construction of trees of effectively unlimited depth. Of course, so long as the net is finite, there must be some limit, but this is no different from the discrete case: the finite memory of a real digital computer places a limit on the size of trees; arbitrarily deep tree construction is possible only in the unlimited memory of imaginary computers, such as the Turing machine.

3.3.2 Fourier Interleaving

There are many ways that recursive structures can be represented in infinite-dimensional spaces (MacLennan, in press-a), but we will consider only one, "Fourier interleaving," which has some interesting properties.

Since image spaces are homeomorphic to subsets of Hilbert spaces, we can expand images in generalized Fourier series.[21] Two images φ and ψ in this space can be combined into a pair $\mathsf{cons}(\varphi, \psi)$, also in this space, by interleaving the Fourier coefficients of φ and ψ:

$$\begin{aligned} \mathsf{cons}(\varphi, \psi) &= \sum (c_k \upsilon_{2k} + d_k \upsilon_{2k+1}), \\ \text{where } \varphi &= \sum c_k \upsilon_k \quad \text{and} \quad \psi = \sum d_k \upsilon_k. \end{aligned}$$

This kind of mixing of images might seem questionable, but in fact the operation is mathematically very well behaved: it is continuous, linear and isometric (preserves the metric of the Hilbert space) on $\mathcal{L}_2 \times \mathcal{L}_2$. Since this operation returns an image in the same space as its arguments, it can be applied recursively to build binary trees of arbitrary depth (cons

[20] The reader familiar with Peano's space-filling curves may be confused by Brouwer's result. However, the Peano curve is a continuous map of the unit interval $[0, 1]$ into the square $[0, 1]^2$, but not vice versa. Observe that infinitesimal movements on the line map to infinitesimal movements in the square, but that infinitesimal movements in the square may map to finite movements on the line.

[21] A generalized Fourier series is a Fourier series with respect to an arbitrary orthonormal basis $\upsilon_0, \upsilon_1, \upsilon_2, \ldots$, i.e., not necessarily the familiar trigonometric or complex exponential bases.

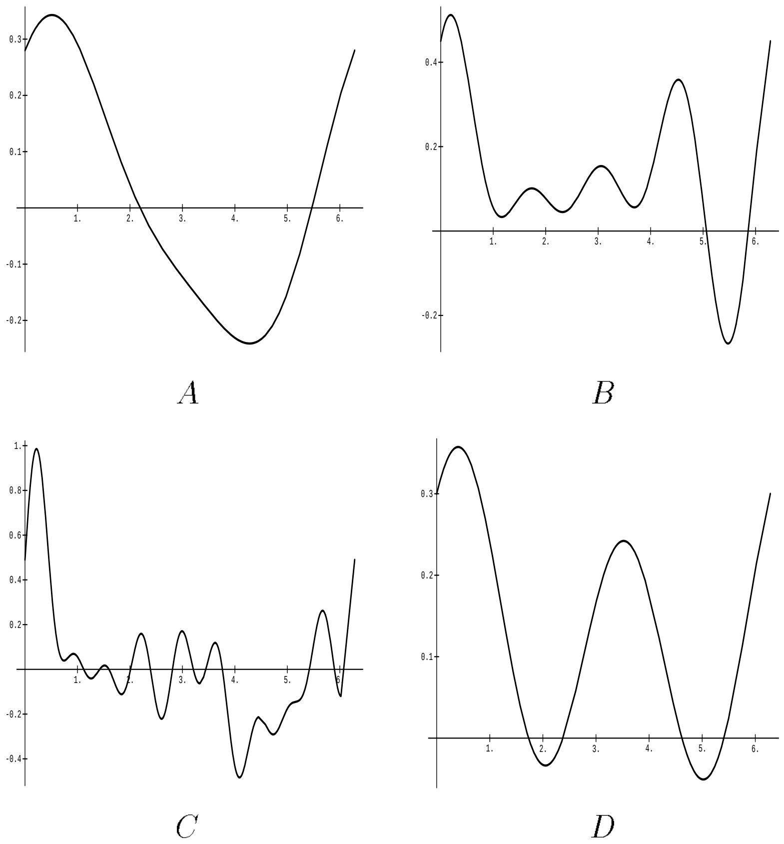

Figure 6: Fourier interleaving. The plots show examples of Fourier interleaving as a method of constructing recursive binary trees. Graphs A and B show two given images from which C is constructed by Fourier interleaving, $C = \mathsf{cons}(A, B)$. Conversely, images A and B can be extracted from image C by $A = \mathsf{left}(C)$ and $B = \mathsf{right}(C)$. Graph D demonstrates that left and right "parts" can be extracted from images that were not constructed by cons, in this case $D = \mathsf{left}(B)$.

omitted for clarity):

$$(a, b), \quad (a, (b, c)), \quad ((a, b), (c, d)), \quad ((a, (b, c)), (d, e)), \quad \text{etc.}$$

It is also apparent that the left and right elements can be easily extracted from the pair $\chi = \mathsf{cons}(\varphi, \psi)$: simply take either the odd or the even elements of its Fourier series:

$$\mathsf{left}(\chi) = \sum c_{2k} \upsilon_k \qquad \text{and} \qquad \mathsf{right}(\chi) = \sum c_{2k+1} \upsilon_k,$$
$$\text{where} \qquad \chi = \sum c_k \upsilon_k.$$

One of the interesting properties of Fourier interleaving is that we can extract the left and right elements of *any* image, regardless of whether it resulted from Fourier interleaving

(Fig. 6). This is because any image can be expanded as a Fourier series (even if many of its coefficients are zero), from which the odd or even elements can be selected. In this sense, images behave like infinite trees with no leaves; we can always go further out from the root. Of course, if an image has finite bandwidth — as would be the case in practical situations — then eventually we will reach images that are either the zero image or images proportional to the first element of the basis sequence. The left and right parts of these can still be extracted, but they will return the same images:

$$\mathsf{left}(\mathbf{0}) = \mathbf{0}, \quad \mathsf{right}(\mathbf{0}) = \mathbf{0}, \quad \mathsf{left}(kv_0) = kv_0, \quad \mathsf{right}(kv_0) = \mathbf{0}.$$

Of course $\mathbf{0} = 0v_0$, so the images kv_0 (with k a real number) function in some ways like the atoms of the image space.

We have seen before that the Riesz-Fischer theorem, which states that $\mathcal{L}_2$ (the space of finite-energy images) is topologically equivalent to ℓ_2 (the space of square-summable real sequences), suggests that the interface between the continuous and the discrete lies in Hilbert spaces. In this sense the real numbers can be considered the "elementary images" from which all images are constructed. These form a continuum and do not fall into discrete types.

Engineers may object that Fourier interleaving is impractical since, with increasing tree-depth, information gets pushed into higher and higher frequency bands; this is valid objection.[22] Noise often increases with frequency, and high-frequency representations may be limited by physical inertia (for temporal frequency) or atomic structure (for spatial frequency). Therefore, although in the mathematically ideal sense arbitrarily deep trees (or long sequences) can be represented by Fourier interleaving, any physical instantiation of the process will have the property that as trees become deeper (or sequences longer), the leaves become more degraded until they are completely inaccessible. This is an appealing model of the competence/performance distinction, since the physical medium causes performance to degrade gradually with increasing depth, rather than facing a hard limit (such as would be implied by a finite stack or sequence buffer). Competence corresponds to the ideal provided by the infinite-dimensional space.

Although the preceding construction shows that recursive tree construction is possible in connectionist systems, we do not attach as much significance to this as might those who think that trees are essential to linguistics (e.g., Fodor & Pylyshyn, 1988; Pinker & Prince, 1988). I have argued elsewhere that *the path to a post-Chomskian linguistics is hidden by our general inability to see language in any terms other than trees* (MacLennan, 1993a). Although I cannot offer a specific alternative (since I wear the same blinders), the preceding approach to category invariants and rule-like behavior is one possibility.

4 Formal Languages

4.1 Specification of Image Spaces

The formation rules of a calculus define a formal language, namely, the well-formed formulas. In this section I will consider analogous procedures for specifying image spaces and (fuzzy)

[22] We assume here the trigonometric basis, though the same arguments apply to any basis sequence characterized by increasing frequency.

classes of well-formed images, that is, continuous formal "languages."

4.1.1 Delimiting the Space

Specification of a (discrete) formal language usually begins by defining an *alphabet* or *vocabulary* of atomic types; this is usually defined *ostensively*, that is, by exhibiting (in a special context) a token of each type. For example,

$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}.$$

Certain well-known alphabets may be taken as given (e.g., the digits, the letters).

In exactly the same ways, we may specify the range of variation of a continuous formal language. We define it ostensively, by exhibiting (in a special context) the limits of the range. For example, we may exhibit a range of lengths, or a range of lightness/darkness:

$$R = [\text{-}, \rule{4em}{0.4pt}], \qquad R = [\circ, \bullet].$$

We may also define it by use of a predefined sets or values. For example, we may take as given spaces such as $\mathbf{R} = (-\infty, \infty) =$ the real line, and $\mathbf{U} = [0, 1] =$ the unit interval. Also we may take as given the constant π in a definition such a $R = [0, 2\pi]$.

In defining a discrete formal language the next step is to specify the possible places a token can be put, which we call the domain of locations. This is almost always a linear array of locations of arbitrary but finite length; thus the domain of possible locations corresponds to the natural numbers. In this case the formula space is V^*. Occasionally the characters will be put into a two-dimensional array (rather than a one dimensional string), or some other arrangement; in some cases the strings are restricted to a fixed length n, so the space is V^n.

The situation is analogous for a continuous formal language. For example, the space can be given by a Cartesian product, for example $R_1 \times R_2 \times \cdots$, where $R_1, R_2, \ldots$ are the ranges of the individual dimensions. It is also necessary to specify the metric on the image space; it cannot be assumed to be the Euclidean metric.

If the domain of locations is a continuum, then it can be defined by the same mechanisms used for ranges: ostension or predefined sets. In this case, for a domain of locations D and a range of variation R, the images are continuous functions from D into R. As with finite-dimensional image spaces, it is necessary to specify the metric, which we may do by appeal to a named function space. For example, we might say that the image space is the space of $\mathcal{L}_2$ functions from $\mathbf{U}$ into $\mathbf{R}$.

4.1.2 Well-Formedness

The usual way of specifying a subset of *well-formed formulas* in the formula space of a discrete language is to exhibit a nondeterministic generative process (a *generative grammar*) that produces just the well-formed formulas. Reversing this process — determining whether a given string is well-formed — is nontrivial, and a large body of computer-science theory (parsing theory) is devoted to the efficient inversion of generative grammars.

The well-formed images of a continuous language may be defined in many of the same ways as discrete languages, for example, by specifying a generative process. In many cases other methods are more appropriate, such as definition by example. In any case, since the category of well-formed images is a subset of an image space, by Postulate 3 we know the category must be fuzzy.

For a concrete example, suppose $\omega_1, \omega_2, \ldots, \omega_n$ are exemplars of well-formed images. Then the continuous map

$$W(\omega) = \epsilon - \min[\delta(\omega, \omega_1), \delta(\omega, \omega_2), \ldots, \delta(\omega, \omega_n)]$$

is positive just when ω is closer than ϵ to at least one exemplar. Thus W defines an indeterminate category of well-formed images. More complex classes of well-formed images can be defined by specifying category invariants that must hold.

4.2 Descriptions of Formal Systems

In writing a program, or otherwise constructing a calculus, we use a language of discrete symbols to describe a discrete process over a discrete state space. When we broaden our awareness to include continuous languages and simulacra, then we see that there are four possibilities to consider:

- Discrete description of discrete systems.
- Discrete description of continuous systems.
- Continuous description of continuous systems.
- Continuous description of discrete systems.

The first has already been considered, so I'll briefly address the others.

The description of the continuous by the discrete is what we do whenever we use mathematical notation to describe image spaces (as discussed above). For example, given certain primitive spaces (such as the real line), we can use the discrete language of set theory to construct higher-dimensional Euclidean spaces. Describing the continuous by the discrete is also what we do when use differential equations to describe a continuous process.

There is a natural match between discrete languages and discrete systems, and so we might expect that the best vehicle for describing a continuous system is a continuous language. The difficulty is that continuous languages are much less familiar than discrete languages, so it isn't obvious how to proceed. I'll mention a few possibilities.

I have indicated above how image spaces can be defined ostensively (as are formula spaces), and have suggested means for defining classes by example. Learning procedures permit the definition of maps and processes by continuous manipulation of the input and output, thus providing a continuum of examples. In other cases we may use gestures or other continuous operations to directly mold a potential surface to determine the dynamics of a continuous process.

Our investigation of continuous languages has just begun, but some opportunities are already apparent. In addition to providing a framework in which to understand ritualized motor activities, including dance and speech, we expect that continuous languages will have applications in computer system design, for example, in gestural interfaces.

The last combination, the description of the discrete by the continuous, is also important, for this is what happens when people talk *informally* about a calculus, for example, when mathematicians discuss formal logic or axiomatics, when computer scientists discuss program organization, or when chess players discuss strategy. Continuous languages may prove a powerful tool for conquering the overwhelming minutiae of complex calculi.

5 Intentions

5.1 Definition

Another advantage of discrete knowledge representations is the ease with which one part of a structure may refer to ("point at") another, which is especially useful in the representation of propositional information. In this section I will discuss *intentional fields*, a mechanism for accomplishing the same ends with continuous representations (MacLennan, in press-b).

Traditionally, an intention (*intentio*) is a mental construct that is directed at something else (either internal or external to the mind), selecting it for further processing. The Latin verb *intendere* means to *stretch toward* or *point at*, and includes the sense of *focusing one's attention on* (*Oxford Latin Dict.*, s.vv. *intendere, intentio*). In Medieval Scholastic theories of the mind, an *intentio* was "something in the soul capable of signifying something else" (Ockham, *Summa Logicae* I §12). Use of *intentionality* by Brentano, Husserl and Heidegger retains the sense of a mental "vector" directing awareness at its object. (Runes, 1942, s.v., defines it as "the property of consciousness whereby it refers to or intends an object.") In short, *intentions are the basic mental mechanisms for pointing and directing.*

Intentions are concerned with selection of parts from a whole, and so we will have to treat the parts of images. Therefore it will be convenient to consider them elements of Hilbert spaces, specifically, as *fields*, quantities defined over a spatial continuum (MacLennan, 1987, in press-b).

Formally, a field ν is defined to be an *intention of type F toward φ* if $\nu : D \to [0,1]$, where $\varphi : D \to R$, and F is an operator defined on $\mathcal{L}_2(D) \times \mathcal{L}_2(D)$. The intention is *resolved* by applying F to the pair (ν, φ), that is, $F(\nu, \varphi)$. The implication is that ν focuses F's processing of φ on the region of D where ν is largest. Thus an intentional field is similar to a generalized focus of attention — generalized in that it need not be present to consciousness.

This model of intentions is analogous to Peirce's explanation of signification as a triadic relation between a subject and an object with a particular significance (meaning) for that subject.[23] That is, an intention is *someone's* consciousness of *something* in *some regard.* So, for a fixed subject, the intention indicates an *attitude* toward an *object.* We may also say it has *form* and *matter.*

If we fix the subject and object, then what distinguishes different kinds of intentions is their manner of being processed, which corresponds mathematically to the function to which

[23]For example, Peirce said a *representamen* (representation) is "something which stands to somebody for something in some respect or capacity" (2.228; Goudge, 1969, p. 139). The triadic relation is between sign, object and interpretant, though Peirce takes the latter to be an equivalent sign in the mind of the interpreter, whereas I cash it out as a function. Roughly, the triadic relation is manifest in the subject, verb and object of sentences such as "Bill notices the bear," "Bill fears the bear," etc.

they are passed. These functions may correspond to different brain areas, or perhaps to orthogonal subspaces within a single area.

Many behaviors can be explained in terms of the generation of intentions and their translation from one modality to another. For example, motion in the visual field could cause the generation of a *novelty intention* focusing attention on a region of the visual field. This could cause the object in that region to be classified, perhaps generating a *fear intention* toward that same region of the visual field, and to a corresponding region in an egocentric or allocentric spatial field. This in turn could generate an *avoidance intention*, which would be translated into a *motor intention* toward another place, which would cause the generation of appropriate motor images (i.e. activity in motor areas).

In the following sections I'll consider some examples of intentions, gradually working our way toward the representation of propositional information.

5.2 Perceptual Intentions[24]

The easiest intentions to understand are perceptual intentions, especially visual intentions. Perhaps the simplest visual intention is a *motion intention*, which results from the detection of motion; it could be computed as the square of the time derivative of the visual image, no doubt spatially smoothed. For example, $\nu_{xyt} = [\mathrm{d}(G_{xy} * \varphi_{xyt})/\mathrm{d}t]^2$, where G_{xy} is a 2D Gaussian and $*$ means 2D convolution. A motion intention could also be computed from a more abstract image, such as the Gabor coefficients of the retinal image.

Although some perceptual intentions are computed bottom-up; more typically they result from a combination of top-down and bottom-up processing. Examples of such intentions include one directed at the presence of an unexpected object, or one directed at the absence of an expected object. This latter demonstrates that intentions need not be directed at things present. These are both "surprise intentions," but there may also be intentions noting fulfilled expectations (e.g., the presence of an expected object, or absence of one expected to be absent).

Though an intention directs processing toward some region of an image, there is no requirement that that image be a representation of reality (as Meinong observed). That is, the image may be perceptual in form (e.g., visual or auditory), but not perceptual in content (i.e., imagined rather than perceived). This is the foundation on which is built our ability to reason propositionally, as will be discussed next (see also MacLennan, 1993b).

5.3 Propositional Intentions

Johnson-Laird and Byrne (Johnson-Laird & Byrne, 1991, 1993) argue convincingly that deduction is implemented by the manipulation of models, but their theory requires certain elements of a model to be treated in special ways. For example, a object may be "tagged" to indicate that a model does not contain that object, or that this model is the only one that can contain that object. I have argued (MacLennan, 1993b) that Johnson-Laird & Byrne's models can be explained as combinations of images and propositional intentions. For example a model of the absence of a circle and the presence of a triangle, which they

[24]This section will be omitted from the published version.

abbreviate

$$\neg \bigcirc \qquad \triangle$$

can be represented by an image of a circle and a triangle and an intentional field negating the circle and asserting the triangle, which I abbreviate:[25]

$$\begin{array}{|cc|}\hline \bigcirc & \triangle \\ \hline \ominus & \oplus \\ \hline \end{array}$$

What gives this intentional field the aspect of propositional representation is the manner in which it is processed (which is no doubt correlated with cortical region carrying the field). For example, since a model comprises both an image and an intentional field, the integration of new information with a model necessitates updating the intentional field. Therefore, if the new information asserts the presence of a circle:

$$\begin{array}{|cc|}\hline \bigcirc & \\ \hline \oplus & \\ \hline \end{array}$$

then the model negating the circle must be rejected. Such a contradiction is indicated by the two intentional fields — for the model and for the new information — having opposite signs in a common region.

The rejection of an entire model can be indicated by a intentional field that negates the region holding that model, while leaving the rest unnegated. Thus, negation of m_2 out of $m_1, \ldots, m_n$ could be represented:

$$\begin{array}{|ccccc|}\hline m_1 & m_2 & m_3 & \cdots & m_n \\ \hline & \ominus & & & \\ \hline \end{array}$$

This is not the place for additional detail, but the general pattern should be clear.

6 Emergence of Discrete from Continuous[26]

6.1 Introduction

Discrete knowledge representation and processing systems are (superficially, at least) very good at manipulating mathematical and logical formulas, and similar discrete structures, according to precise rules; they have been less successful at subsymbolic processes, such as perception, recognition, association, control and sensorimotor coordination. In contrast, connectionist approaches are well-suited to subsymbolic tasks, but there has been doubt about how well they can operate at the symbolic level. This naturally suggests hybrid architectures, with symbolic tasks accomplished by discrete (digital) computation and subsymbolic tasks by continuous (analog) computation.

This is not the way the brain works, however, for in the brain discrete, symbolic processes emerge from underlying continuous, subsymbolic processes. There is reason to believe that

[25] Although I defined intentional fields to be $[0, 1]$-valued, in this case it is more convenient to allow them to be $[-1, 1]$-valued, so that assertion and negation can be represented in the same intentional field.

[26] This section omitted from published version.

this is not just an accident of biological intelligence, but that this underlying continuity imparts to the emergent symbolic processes the flexibility characteristic of human symbol use (MacLennan, 1991, 1992). The problem is then to understand how discrete-looking representations and processes can emerge from continuous representations and processes.

It is an oversimplification to treat language as a discrete system. For example, although we accept the space between written words without question, anyone who has done continuous speech recognition knows that we cannot depend on spaces in the sound stream. Furthermore, in ancient Greek, when written language was not considered an autonomous means of expression, but was viewed as a visual representation of the sound stream, we find words run together without intervening spaces, just the way we speak. Seneca claimed that Latin writers sometimes separated words because there was a *difference in speech rhythm* between Greek and Latin speakers, namely, that Latin speakers left a pause after each word (Small, 1992). Havelock (1982, p. 8) points out that the alphabet was in use in Greece for 300 years before Greek had a word for 'word'. Apparently the concept of a word, as a discrete, indivisible unit of the sound stream is not so obvious as we now take it.

In summary, the phenomenological salience of the word is partially a result of our use of an alphabetic writing that separates words, and of the cultural practices that go along with it, such as dictionaries, indices, and word-oriented reading instruction (Small, 1992). Nevertheless, the concept of a word is neither illusory nor arbitrary, since as a matter of fact speakers tend to treat certain segments of the sound stream as units, and it is the recurrence and semi-independence of these segments that form the basis of the 'word' idea. Therefore, it seems that the emergence of approximately-discrete, symbolic processes from the underlying continuous, subsymbolic processes will be illuminated by considering the self-organization of processes for recognizing recurring parts of images (such as the sound stream).

6.2 Metric Correlation

The familiar correlation attempts to match one signal to all possible translations of another signal, and returns a signal showing how well these matched. The correlation of images φ and ψ is defined:

$$[\varphi \otimes \psi]_\alpha = \int_\Omega \varphi(t-\alpha)\psi(t)\mathrm{d}t.$$

The structure of this is more apparent when we realize that $\varphi(t-\alpha)$ is φ translated to the right by an amount α. Therefore we introduce the operator T_α to mean a rightward translation by α, and rewrite the correlation:

$$[\varphi \otimes \psi]_\alpha = \int_\Omega T_\alpha\varphi(t)\psi(t)\mathrm{d}t.$$

It is then apparent that the correlation is the inner product of the translated image $T_\alpha\varphi$ and the image ψ. So we write it that way:

$$[\varphi \otimes \psi]_\alpha = \langle T_\alpha\varphi, \psi\rangle.$$

The significance of the inner product is that it measures the similarity of normalized images. Therefore we write $\sigma(\varphi, \psi) = \langle\varphi, \psi\rangle$ to emphasize that its purpose is to measure similarity:

$$[\varphi \otimes \psi]_\alpha = \sigma(T_\alpha\varphi, \psi).$$

Thus, in general terms, the value of the field $\chi = \varphi \otimes \psi$ at a point α is the similarity of the α-translate of φ to ψ.

The common correlation can be generalized to other classes of transformation (rotation, scaling, perspective distortion, etc.) as well as to other measures of similarity or dissimilarity. If T is any parameterized class of transforms and σ is any similarity metric, then define $\sigma(\varphi \lhd_T \psi)$ the *metric correlation* with ψ of all T-transforms of φ by:

$$[\sigma(\varphi \lhd_T \psi)]_\alpha = \sigma(T_\alpha\varphi, \psi).$$

Sometimes it is easier to work in terms of difference rather than similarity, in which case we write:

$$[\delta(\varphi \lhd_T \psi)]_\alpha = \delta(T_\alpha\varphi, \psi).$$

where δ is a (difference) metric. When the metric or transform is clear from context it will be omitted; thus in general $\varphi \lhd \psi$ is the metric correlation of all transforms of φ with ψ. Sometimes it is more convenient to consider the metric correlation of φ with all T-transforms of ψ, so we write:

$$[\rho(\varphi \rhd_T \psi)]_\alpha = \rho(\varphi, T_\alpha\psi),$$

where ρ is either a similarity or difference metric. Clearly, $\varphi \lhd \psi = \psi \rhd \varphi$. Finally we define the operator which correlates all T-transforms of φ with all U-transforms of ψ:

$$[\rho(\varphi \diamond_{T,U} \psi)]_{\alpha,\beta} = \rho(T_\alpha\varphi, U_\beta\psi).$$

Notice that the correlation field resulting from this operation is indexed over two parameter spaces. These operations satisfy many simple identities, most of which are obvious, and so omitted.

It is often useful to consider metric correlations under multiple transformations. For example, if X_α for $\alpha \in \mathcal{E}^2$ (2D Euclidean space) is a translation by α, and R_β for $\beta \in S^1$ (a topological circle) is a rotation through an angle β, then $T_{\alpha\beta} = X_\alpha R_\beta$ is a rotation followed by a translation (the order doesn't matter; they commute). Thus $(\varphi \lhd \psi)_{\alpha\beta}$ measures the correlation between ψ and the α-translation, β-rotation of φ.

Often the transformations applied to images have algebraic structure; frequently they form a topological group. In these cases the metric correlations inherit the structure; for example, if the transformations are an abelian group:

$$(T_\alpha\varphi \lhd \psi)_\beta = (\varphi \lhd \psi)_{\alpha+\beta} = (\varphi \lhd \psi)_{\beta+\alpha} = (T_\beta\varphi \lhd \psi)_\alpha.$$

The reader may suppose that metric correlations are computationally too expensive to have much significance to cognitive processing. First observe that they are not much more expensive than the usual (inner product) correlations. Second, the limited precision of neural computation (one or two digits) will limit the number of transforms to be computed to a dozen or so, for each real parameter. A combination of two transforms might require 100 to be computed (in parallel).

As discussed above, the speech stream is essentially continuous, and, to a first approximation, words are segments of the sound stream that can be treated as discrete units, that is, relocated as wholes. Conversely, recurrence of the same signal in a variety of contexts is evidence that it is a meaning unit, often a word. Therefore, we suspect that one component of language learning is the detection of such recurrence. However, a word will rarely recur exactly; it will be transformed in duration, pitch, amplitude, etc., or by the context of surrounding sounds. This suggests metric correlation as a mechanism for the self-organization of a word-recognition system.

7 Learning

A detailed discussion of learning in simulacra is beyond the scope of this chapter, so a few general observations must suffice. First observe that since finite-dimensional Euclidean spaces are image spaces, all the usual learning algorithms, such as back-propagation, apply to finite-dimensional image spaces with the inner-product norm. Second, I will assert without proof that these same algorithms generalize with little change to the infinite-dimensional Euclidean case, that is, to Hilbert spaces. (To see this, just observe how little of the usual derivation depends on the dimension of the space; convergence of the infinite sums follows from the square-summable property of the Hilbert space.) Finally, recall that any image space is topologically equivalent to a subset of a Hilbert space. Therefore learning in image spaces can be accomplished by gradient descent in the corresponding Hilbert space. Since the image space and the Hilbert space are topologically equivalent, convergence in one is equivalent to convergence in the other. (The homeomorphism preserves limits.)

8 Decidability & Computability

The question naturally arises of whether the well-known undecidability and uncomputability results hold for simulacra and continuous computational systems, and the beginnings of a theory of continuous computation has already appeared. For example Blum and her colleagues (Blum, 1989; Blum, Shub, & Smale, 1988) have shown that many classical computability, decidability and complexity results generalize to computation over the real numbers. On the other hand, Stannett (1990) has published a proof that the halting problem for Turing machines can be decided by certain machines with continuous dynamics, which thus have super-Turing power, and Pour-El and Richards (1979, 1981, 1982) have shown that a Turing-computable wave equation with Turing-computable initial conditions can have non-Turing-computable solutions.

Interesting as these results are, they do not address the central issue, for the traditional theory of computation asks what *calculi* can decide about *calculi*, and analogously a theory of computation that takes a consistently continuous perspective should ask what *simulacra* can decide about *simulacra*. However, such a perspective forces us to ask different questions about simulacra than we ask about calculi. First observe that *a simulacrum cannot answer a yes-or-no question.* This follows simply from continuity, since if the answer is 0 for some questions and 1 for others, then there must also be questions yielding neither answer; this is the case for both image maps and image processes (Props. 2 and 3). It may be objected that *we* can draw an absolute distinction (such as '$< 1/2$' vs. '$\geq 1/2$'), but that just places the calculus in *us*, and presumes that *we* can instantiate a perfectly discrete system. This is the familiar situation of a calculus deciding a property of a simulacrum, and doesn't address the question of what simualcra can decide about simulacra. The latter question is the relevant one if, as connectionist research suggests, continuous models of cognition are more accurate than discrete ones.

A concrete illustration is provided by a continuous analog of Turing's famous proof of the undecidability of the halting problem. Let $\mathcal{F}$ be a space of continuous processes $P : S \times \mathbf{R} \to S$. The *stability problem* for a process P and initial state s is to determine if the process $P(s,t)$ is asymptotically stable (i.e., in some basin of attraction); if it isn't then we say that the process is unstable. What we would like to know is whether there is a *process* in $\mathcal{F}$ that can decide the stability problem for all processes in $\mathcal{F}$ and initial states in S. In order to have a process make decisions about other processes, the latter must be representable in the state space (this is what Gödel numbering accomplishes in the discrete case). Therefore, assume that there is a space $F \subseteq S$ such that for each process $P \in \mathcal{F}$ there is at least one corresponding "program" (an image) $p \in F$. (That is, we assume there is a continuous *universal function* from F onto $\mathcal{F}$.) Then, a *decision process for the stability problem* is a continuous function $D : (F \times S) \times \mathbf{R} \to (F \times S)$ such that:

$$\begin{aligned} D[(p,s),t] &\to \sigma \quad \text{iff} \quad P(s,t) \text{ is stable}, \\ D[(p,s),t] &\to \upsilon \quad \text{iff} \quad P(s,t) \text{ is unstable}, \end{aligned}$$

where P is the process corresponding to the "program" p and σ and υ are any distinct points in the state space $F \times S$, which are used to represent the answers 'stable' and 'unstable'.

The impossibility of such a decision procedure is now apparent, since by Prop. 3 no such process can exist; there must be pairs (p,s) that are outside all basins of attraction, including those for σ and υ. For these pairs the decision process gives no answer. In general there can be no decision procedure that always gives an answer, so decision problems have to be reformulated to be consistent with the postulates of simulacra.

With their assumption of discreteness, calculi have by definition the ability to perform exact classification, but also the liability of responding discontinuously to infinitesmal variations in the input (brittleness). Conversely, with their assumption of continuity, simulacra have by definition the ability to respond continuously to infinitesmal changes in input (flexibility), but are unable to classify exactly. These abilities and inabilities are consequences of the idealizing assumptions (perfect discreteness or perfect continuity) in each case. In reality we know that a calculus can take arbitrarily small steps, and that a simulacrum can make arbitrarily fine discriminations. Therefore the conclusions we draw from the theoretical models are relevant to the real system only to the extent that the real approximates the ideal at the relevant level of abstraction. This, again, is the Complementarity Principle.

9 Summary

The calculus is a theoretical construct that captures the essence of traditional, discrete, symbolic models of cognition. The development of new, connectionist models, which are characterized by continuity, will be promoted by the development of an analogous theoretical construct, and the simulacrum has been proposed to fill this role. Whereas calculi implement discrete processes operating on formulas, simulacra implement continuous processes operating on abstract images. The theory of simulacra suggests a novel approach to many issues in cognitive modeling, including classification, invariants in behavior, constituent structure, intentions and approximately discrete processes, such as rule-like behavior, symbolic cognition, and language.

10 References

Blum, L. (1989). *Lectures on a theory of computation and complexity over the reals (or an arbitrary ring)* (Report No. TR-89-065). Berkeley CA: International Computer Science Institute.

Blum, L., Shub, M., & Smale, S. (1988). On a theory of computation and complexity over the real numbers: NP completeness, recursive functions and universal machines. *The Bulletin of the American Mathematical Society* **21**: 1–46.

Dolan, C. P., & Smolensky, P. (1989) Implementing a connectionist production system using tensor products. In: *Proceedings of the 1988 connectionist models summer school.* San Mateo, CA: Morgan Kaufmann.

Fodor, J. A., & Pylyshyn, Z. W. (1988) Connectionism and cognitive architecture: A critical analysis. In: S. Pinker & J. Mehler (Eds.), *Connections and symbols* (pp. 3–71); reprinted from *Cognition* **28**.

Goudge, T. A. (1969) *The thought of C. S. Peirce.* New York NY: Dover.

Havelock, E. A. (1982) *The literate revolution in Greece and its cultural consequences.* Princeton NJ: Princeton University Press.

Hausdorff, F. (1957) *Set theory,* transl. J. R. Aumann & al. Oxford UK: Oxford University Press.

Hirsch, M. W., & Smale, S. (1974) *Differential equations, dynamical systems, and linear algebra.* San Diego CA: Academic Press.

Johnson-Laird, P. N., & Byrne, R. M. J. (1991) *Deduction.* Hillsdale NJ: Lawrence Erlbaum.

Johnson-Laird, P. N., & Byrne, R. M. J. (1993) Precis of *Deduction. Behavioral and Brain Sciences* **16**, 2: 323–333, 368–380.

MacLennan, B.J. (1987) Technology-independent design of neurocomputers: The universal field computer. In: M. Caudill & C. Butler (Eds.), *Proceedings, IEEE first international conference on neural networks* (Vol 3, pp. 39–49). New York NY: Institute of Electrical and Electronic Engineers.

MacLennan, B. J. (1988). Logic for the new AI. In: J. H. Fetzer (Ed.), *Aspects of artificial intelligence* (pp. 163–192). Dordrecht, NL: Kluwer Academic Publishers.

MacLennan, B. J. (1990a). *Field computation: A theoretical framework for massively parallel analog computation; parts I – IV* (report CS-90-100). Knoxville TN: University of Tennessee, Computer Science Department.

MacLennan, B. J. (1990b). *Functional programming: Practice and theory.* Reading MA: Addison-Wesley.

MacLennan, B. J. (1991) Flexible computing in the 21st century. *Vector Register* **4**, 3: 8–12.

MacLennan, B. J. (1992) *Research issues in flexible computing: Two presentations in Japan* (Report No. CS-92-172). Knoxville TN: University of Tennessee, Computer Science Department.

MacLennan, B. J. (1993a). Characteristics of connectionist knowledge representation. *Information Sciences* **70**: 119–143.

MacLennan, B. J. (1993b) Visualizing the possibilities (review of Johnson-Laird & Byrne's *Deduction*). *Behavioral and Brain Sciences* **16**, 2: 356–357.

MacLennan, B. J. (in press-a). Continuous symbol systems: The logic of connectionism. In: D. S. Levine and M. Aparicio IV (Eds.), *Neural Networks for Knowledge Representation and Inference*, pp. 83–120. Hillsdale NJ: Lawrence Erlbaum.

MacLennan, B. J. (in press-b) Field computation in the brain. In: Karl Pribram (Ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data.* Hillsdale, NJ: Lawrence Erlbaum.

MacLennan, B. J. (in press-c) Information processing in the dendritic net. In: Karl Pribram (Ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data.* Hillsdale, NJ: Lawrence Erlbaum.

MacLennan, B. J. (in press-d) Grounding analog computers. *Think.*

MacLennan, B. J. (in press-e) "Words lie in our way." *Minds and Machines.*

Nemytskii, V. V., & Stepanov, V. V. (1989) *Qualitative theory of differential equations.* New York NY: Dover, 1989; previously published, Princeton NJ: Princeton University Press, 1960.

Pinker, S., & Prince, A. (1988) On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition* **28**: 73–193.

Pollack, J. B. (1988) Recursive auto-associative memory: Devising compositional distributed representations. In: *Proceedings of the tenth annual conference of the cognitive science society* (pp. 33–39).

Pollack, J. B. (1989) Implications of recursive distributed representations. In: D. S. Touretzky (Ed.), *Advances in neural information processing systems* (pp. 527–536). San Mateo, CA: Morgan Kaufmann.

Pour-El, M. B., & Richards, I. (1979). A computable ordinary differential equation which possesses no computable solution. *Annals of Mathematical Logic* **17**, 61–90.

Pour-El, M. B., & Richards, I. (1981). The wave equation with computable initial data such that its unique solution is not computable. *Advances in Mathematics* **39**, 215–239.

Pour-El, M. B., & Richards, I. (1982). Noncomputability in models of physical phenomena. *International Journal of Theoretical Physics* **21**, 553–555.

Pribram, K. H. (1991). *Brain and perception: Holonomy and structure in figural processing.* Hillsdale NJ: Lawrence Erlbaum.

Runes, D. O. (1942) *The dictionary of philosophy.* New York NY: Philosophical Library.

Scott, D. S. (1971). The lattice of flow diagrams. In: E. Engeler (Ed.), *Symposium on the Semantics of Algorithmic Languages* (pp. 311–366). Berlin: Springer-Verlag.

Scott, D. S. (1973). Lattice-theoretic models for various type-free calculi. In: P. Suppes, L. Henkin, A. Joja & G. C. Moisil (Eds.), *Logic, Methodology and Philosophy of Science IV* (pp. 157–187). Amsterdam NL: North-Holland.

Shepard, R. N. (1980) Multidimensional scaling, tree-fitting, and clustering. *Science* **210**: 390–398.

Small, J. P. (1992) Historical development of writing and reading. *Psycholoquy.* `92.3.61.reading.10.small`.

Stannett, M. (1990). X-machines and the halting problem: Building a super-Turing machine. *Formal Aspects of Computing* **2**: 331–341.