Continuous Information Representation and Processing In Natural and Artificial Neural Networks

Technical Report UT-CS-03-508

Bruce J. MacLennan*

Department of Computer Science
University of Tennessee, Knoxville
maclennan@cs.utk.edu

September 2003
Revised

January 6, 2004

Abstract

ABSTRACT

1 Introduction

There has been a long-standing tradition in Western philosophy, with its roots in ancient Greek philosophy, that "genuine knowledge" (*epistêmê*) must be expressible in verbal formulas (*logoi*), and that "genuine thinking" is a process of logical deduction or discursive reasoning. There are many reasons for challenging these assumptions (and they have been questioned or denied since ancient times), but they are still prevalent in contemporary cognitive science and artificial intelligence. Contemporary attacks on discrete knowledge representation and processing come from two fronts.

*This report may be used for any non-profit purpose provided that the source is credited. Presentation of this research was supported by the Center for Information Technology Research, University of Tennessee, Knoxville. This report is based on a presentation at the Fifth International Conference on Computational Intelligence, March 8–13, 2002, Research triangle Park, NC.

On the one side, we have discovered the practical limitations of discrete representations of knowledge (e.g., semantic networks) and rule-based models of cognition. The practical evidence can be found is the rise and fall of expert systems in the '70s and '80s, but the fundamental problems were predicted by Dreyfus long ago [1]. The other challenge has come from neuroscience, which has demonstrated the importance of continuous representations and processes in natural intelligence. Information is represented by continuously variable firing rates and phase relationships; by synaptic efficacies which depend continuously on the distribution and arrangement of receptors for neurotransmitters; by graded electrical interactions in dendritic trees with complex geometries; and so forth. In addition we can observe the practical success of artificial neural networks, which usually use continuous representations (in emulation of continuously variable neuron impulse rates).

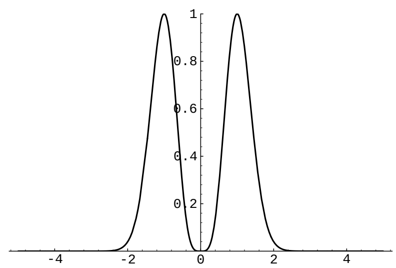
To better understand information representation and processing in natural and artificial neural networks, we need new, continuous models of knowledge and cognition. These models will make fundamentally different assumptions from the traditional theory of knowledge and, as a consequence, answer fundamentally different questions [13]. In our own investigations we have adopted a *principle of continuity*, which constrains all our models to be continuous. (This paper is primarily a summary of prior work, which is discussed in more detail in earlier publications [3, 6, 7, 9]. Most of these and others are available at `http://www.cs.utk.edu/~mclennan`.)

2 Representations

In nature, continuous information representation is much more common than discrete. Visual representations vary continuously in brightness and color, auditory in frequency and amplitude, tactile in pressure, and so forth. Motor output is continuously variable in force and direction. Finally, as previously noted, many neural representations are continuous in rate, phase, electrical potential, and so forth. Thus it is reasonable to take continuous (analog) representations as given, and to treat (approximately) discrete representations as a special case. In mathematical terms, we are dealing with continuous (e.g. real or complex) variables.

Of course, most representations comprise more than a single real variable, and so it is common to make use of vector spaces in the analysis of both natural and artificial neural networks. However, in many cases the dimension of the vector space is so large, that it is better to treat the vectors as continuous *fields* extended over some bounded continuum [2, 5, 10, 11]. This is especially the case for the numerous cortical maps found in the brain, which may be as small as several square millimeters, but still contain hundreds of thousands or millions of neurons.

In summary, traditional, discrete knowledge representation makes use of discrete structures, such as strings or graphs, but connectionist knowledge representation is more appropriately modeled by finite dimensional vectors or (approximately) infinite-dimensional continuous fields. We call continuous representations *images*.



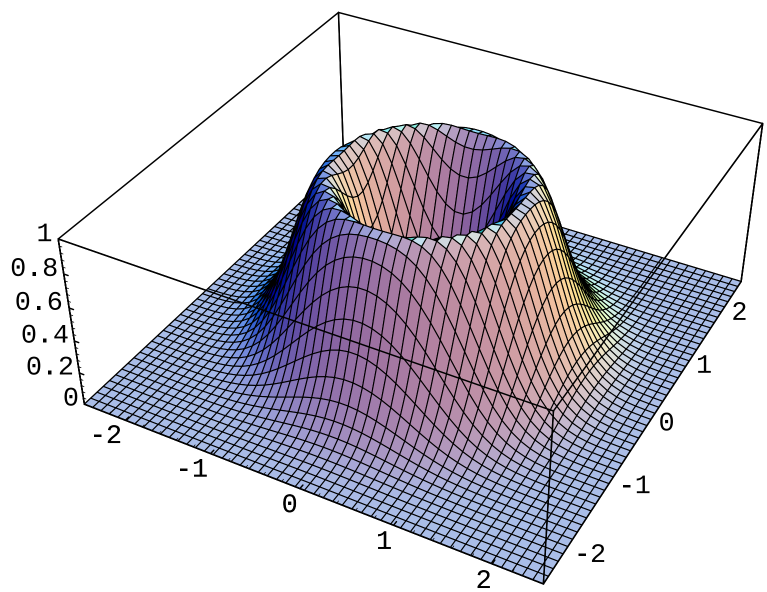


Figure 1: Example of degrees of well-formedness of two-dimensional vectors, wherein normalized vectors are considered perfectly well-formed. In this example the degree of well-formedness is measured by $f(\|\mathbf{v}\|^2)$, where $f(x) = x^2 e^{2(1-x)}$.

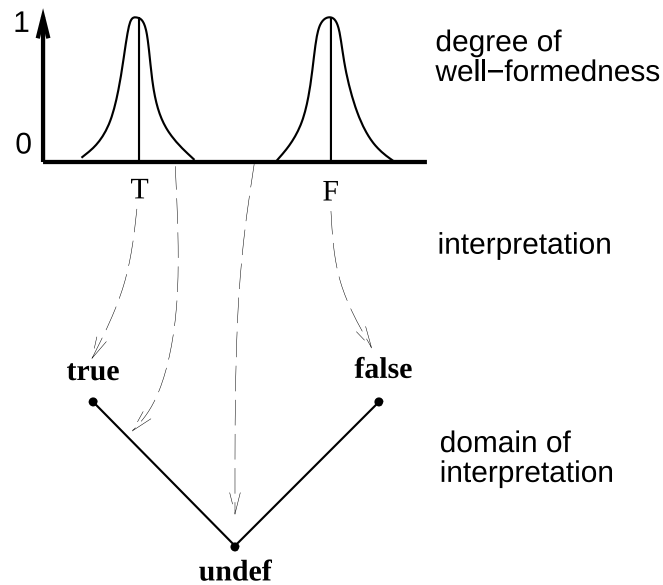


Figure 2: Inherent fuzziness of syntactic well-formedness. Just as the degree of well-formedness decreases continuously from 1 to 0, so also the interpretation must vary continuously from a defined interpretation to an undefined interpretation.

Traditional theories of discrete knowledge representation distinguish well-formed structures (such as “WFFs” — well-formed formulas), which can be interpreted, from ill-formed structures, which cannot. Similarly, in connectionist systems, certain images may be well-formed, in the sense that they are suitable for interpretation or further processing. For a simple example, in some system “well-formed vectors” might be required to be normalized. Clearly, however, no practical system can determine if a vector is exactly normalized, nor can the proper operation of such a system depend on exact normalization. Thus, the well-formedness of a vector must be continuous function of its length, or, in other words, the set of well-formed vectors is fuzzy (Fig. 1). This example illustrates the general situation: although the distinction between well- and ill-formed images may be as sharp as we please, it must be continuous, and therefore there will be images with intermediate degrees of well-formedness (and, hence, interpretability) [9].

In a natural computation system, images need not *represent* in the literal sense of referring to some actual or potential external state of affairs; that is, images need not have a semantic interpretation. Rather, the primary requirement of an image is its *pragmatic utility*, its ability to fulfil its function in a natural or artificial agent [13]. Nevertheless, images often *do* represent objects or situations in some *domain of interpretation*, and in these cases the interpretation must be a continuous function of the image [6, 9]. Often this dependence is natural, as when a cortical map represents a retinal image, the spectrum of a sound, or a location in three-dimensional space.

Certainly there are situations in which the natural interpretation of an image is discrete. For example, an image may represent a proposition which is either true or false (i.e., fuzzy truth is not permitted), or an image might represent the choice between two mutually exclusive actions (e.g., fight or flight). In such a situation, the continuity principle requires that the discrete interpretations be embedded in a continuous domain of interpretation (Fig. 2). For example, in the two preceding examples, the dichotomous choices could be represented by $+1$ and -1, which could be embedded in the continuum $[-1, +1]$. This does not mean that we now have degrees of truth, or a spectrum of actions between fight and flight. Rather, the absolute value $|I|$ of an interpretation $I \in [-1, +1]$ could represent the interpretability of the image, or the system's confidence in the image's interpretation. If $|I|$ is zero or near zero, then the image is effectively uninterpretable (i.e., ill-formed). Regardless of confidence or interpretability, the image represents a discrete meaning, determined by its sign, $\operatorname{sgn} I \in \{-1, 0, 1\}$, where 0 ("undefined") must be added to the range of semantics. The sign might be extracted by a continuous sigmoid function, as is commonly used in artificial neural networks. (Additional discussion of the representation of semantics and pragmatics can be found in prior articles [11].)

Discrete representations, such as those used in traditional knowledge representation, are constructed from atomic components by the application of specific syntactic constructions. The models, of course, are written language, formal logic, and mathematics as finite strings of atomic symbols. In contrast, images need not be constructed from atomic components. Even vectors, which are commonly thought of as comprising their coordinate components, may be more naturally decomposed into other components (such as their principal components). Further, infinite-dimensional images (fields) may not have nontrivial components; certainly they cannot be constructed finitely from the infinity of their point values. In practice, fields and many finite-dimensional images are given as wholes, without any unique "natural" or "preferred" decomposition; indeed, finding an appropriate decomposition may be an important problem for an image processing system to solve [6, 7, 13].

Nevertheless, there are various decompositions that can be applied to a field to reduce it to a discrete set of continuous quantities; examples are generalized Fourier, Gabor, and wavelet decompositions [4, 9]. In effect, what these accomplish is to represent a given image (e.g., a 2-dimensional image) as a (linear) combination of fixed elementary images of the same dimension, weighted by a discrete set of continuous coefficients ("zero-dimensional fields") [9, 11].

3 Processes

The principle of continuity, as well as our knowledge of neural processes, suggests that information processing be considered a continuous-time process, such as a system of differential equations [8, 9, 12]. (I will consider discrete-time processes shortly.) In the simplest case, the differential change of state is a (possibly nonlinear) function

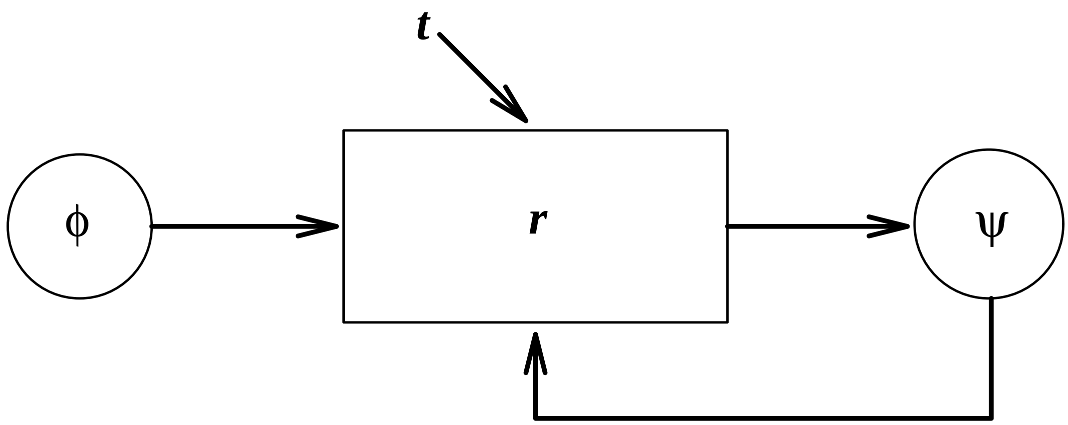


Figure 3: A detministic process.

of time and the current state, $\dot{\psi} = F(t, \psi)$; the state comprises one or more images (fields or finite-dimensional vectors). (Without loss of generality we can take the output of the process to be a continuous *projection* of the state.) More typically, the change of state also depends on one or more time-varying input images ϕ, that is, $\dot{\psi} = F(t, \psi, \phi)$. The process function F is required to be a continuous function of time, state, and input (Fig. 3).

One common example is a gradient descent process, which reduces, by steepest descent, some potential function V, which might represent error or cost: $\dot{\psi} = -r\nabla V(\psi)$. The best-known example is the back-propagation algorithm, but it is more commonly implemented as a discrete-time approximation of the gradient process, $\Delta\mathbf{w} = -\eta\nabla E(\mathbf{w})$, where $E(\mathbf{w})$ is the error for weight vector $\mathbf{w}$.

Indeed, traditional recurrent artificial neural net algorithms are discrete-time processes, in which the state (and output) at one time step is a function of the input and the previous state, $\psi_n = F(\psi_{n-1}, \phi_n)$. So, for back-propagation, $\mathbf{w}_n = \mathbf{w}_{n-1} - \eta\nabla E(\mathbf{w}_{n-1})$. Discrete-time processes of this kind are useful on conventional computers, but they don't seem to be found in natural intelligence, or in artificial systems modeled on it. Therefore we suggest that future research should focus on continuous-time information processing [13].

In the traditional theory of discrete computation, we know that information processes can be represented as finite sets of discrete rules, that is, by *programs*. This allows us to describe these processes, by reducing them to rules, but it also allows us to create *universal machines* (e.g., programmable general-purpose computers) capable of obeying the rules. Similarly, continuous processes can be described by systems of differential equations, which can be used to program general purpose analog computers [14, 15, 16, 17]. Nevertheless, differential equations are not necessarily finite rules, for they may incorporate real or complex coefficients that cannot be expressed in discrete, finite terms [12, 13]. More interestingly, and more relevantly to natural and artificial neural networks, many important continuous processes will be described or governed by *guiding images* [9]. The simplest example of a guiding image is a potential surface, mentioned above. Given a potential surface V and an initial state ψ_0, the subsequent trajectory is determined by a gradient process, $\dot{\psi} = -r\nabla V(\psi)$.

The concept of a guiding image suggests an intriguing new idea of programming; the metaphor is different: instead of a program being *written*, a guiding image is *painted* or *sculpted* [9].

Nondeterministic processes, that is, processes whose trajectories are not completely determined by the initial state and inputs, are often useful for describing information processing in natural and artificial neural networks. We may use a nondeterministic process to describe a natural system when we don't know all the factors governing the trajectory, or when it's preferable to omit their description. Alternately, we may design a nondeterministic system when we don't wish to prescribe precise trajectories, either because it's unimportant or because we want to leave the possibilities open.

Nevertheless, nondeterministic processes are subject to constraints, which determine the possible trajectories. Further, the principle of continuity dictates that there be a continuum between possible and impossible trajectories, and so the constraints must be "soft." For example, soft constraints can be defined by a function $P(\psi, \dot{\psi}) \in [0, 1]$, which defines the "facility" of a change $\dot{\psi}$ in a state ψ [9]. If $P(\psi, \dot{\psi}) = 0$ then the change is impossible; if $P(\psi, \dot{\psi}) = 1$ then it is as easy as it can be. Alternately, each potential trajectory has a cumulative "cost" or "difficulty" $D(\psi, \dot{\psi}) \in [0, -\infty)$, defined by the integral of $D(\psi, \dot{\psi}) = -\log P(\psi, \dot{\psi})$, that is,

$$C(\psi) = \int_0^1 D[\psi(t), \dot{\psi}(t)]\mathrm{d}t,$$

but trajectories with infinite cost are not allowed.

Guiding images can be used to define the soft constraints of a nondeterministic process. For example, we may allow the process to take any path down a potential surface without constraining it to the path of steepest descent [9]. That is, state changes $\mathrm{d}\psi$ for which $-\nabla V(\psi) \cdot \mathrm{d}\psi > 0$ are allowed, whereas those for which it is ≤ 0 are impossible (Fig. 4). Since the facility function must be continuous, we may define

$$P(\psi, \dot{\psi}) = \frac{[-\nabla V(\psi) \cdot \dot{\psi}]^+}{\|\nabla V(\psi)\| \, \|\dot{\psi}\|},$$

that is, the positive part of the inner product of the normalized vectors.

Natural and artificial intelligence in the real world are continually faced with noise, error, inaccuracy and other sources of uncertainty. In practice, all images and processes must be assumed to have an element of uncertainty. One way to accommodate this is to replace images by probability density functions (PDFs) of images, and to replace trajectories of images by trajectories of these PDFs. For example, the instantaneous state $\psi(t)$ is replaced by the PDF $\Psi(\psi, t)$, and the instantaneous input $\phi(t)$ by $\Phi(\phi, t)$. The process is then defined on the PDFs, $\dot{\Psi} = F(t, \Psi, \Phi)$. The approach has much in common with the treatment of the wave-function in quantum mechanics.

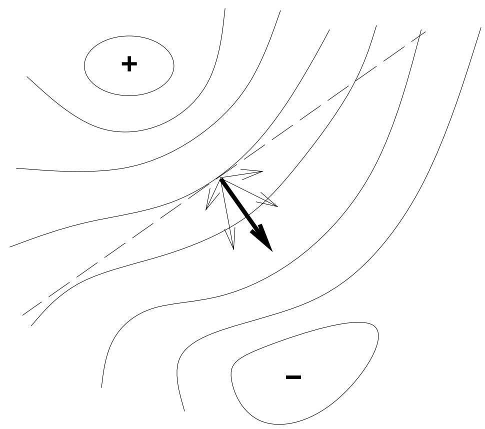


Figure 4: Nondeterministic computation by descent on potential surface. The facility of change in a given direction is shown by the length of the arrow (which is proportional to the cosine of its angle with the negative gradient. The dotted line separates impossible changes to the left from possible changes t o the right.

4 Emergent Rules

Certainly many cognitive processes can be described, at least approximately, by discrete rules; these include language and discursive reasoning. How can continuous neural network processes account for rule-like behavior? This is a large and important topic, about which I can make only few remarks.

Rule-based information processing can be described in the following terms [9]: (1) The state, or a part of the state, is classified into one of a finite number of distinct *situations*. (2) From the classified situation certain limited *index information*, appropriate to the situation, is extracted, which refers to the particulars of the situation; traditionally, we refer to the rule's *variables* being *bound* to components of the situation. (3) The classification determines the rule to be applied, which creates a new situation (from a finite set of possibilities) incorporating some or all of the particulars indexed in (2). Thus, rule-like behavior can be described as the composition of two functions. The first analyzes the state and projects it into a low-dimensional space representing the classification of the situation and the index information. The second function synthesizes the new state by means of the index information.

Now, the interesting observation is that a system will exhibit rule-like behavior whenever it *could* be factored into two functions in this way, regardless of whether the system actually projects the state into a low-dimensional space. In particular, a system might be using only a low-dimensional subspace of what is in reality a higher-dimensional intermediate representation. When this possibility is seen, ones realizes that there is a whole spectrum of degrees of rule-like behavior [9]. For example, intermediate states might be confined to the subspace most of the time, but occasionally go outside of it. Further, the intermediate representation might only *appear* to be

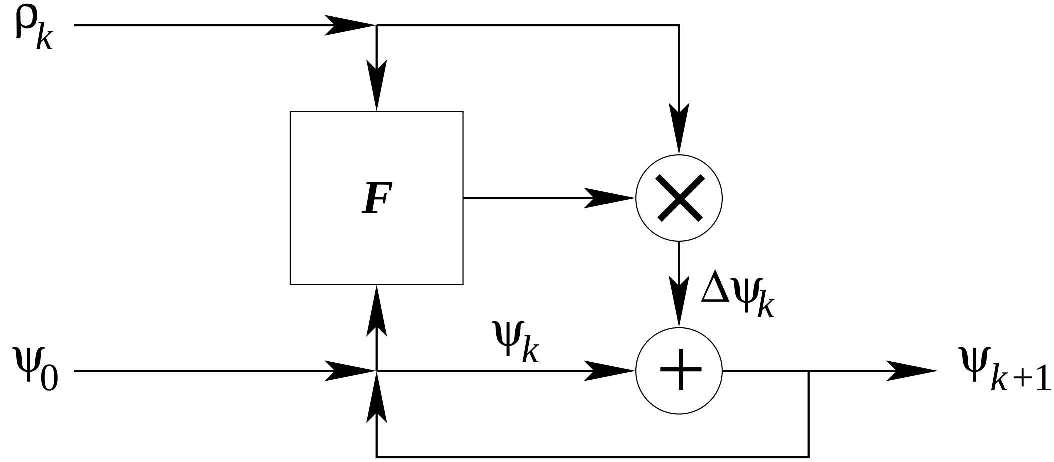


Figure 5: Implementation of Discrete-time Nondeterministic Process

lower dimensional, since most of the variation might be in a few dimensions, while still having some variation in the other dimensions. Such possibilities may explain the exceptions, subtlety and sensitivity of rule-like behavior in humans and other animals.

The foregoing account of rule-like behavior provides the basis for an explanation of how rule-like behavior may emerge and how (apparent) rules may be continuously adaptive [9]. For example, if through ordinary neural network learning, a system adapts to use only a low-dimensional subspace of the sort described, then the system will exhibit rule-like behavior, even though it makes no use of rules. This may occur if the system *could* be so factored (or approximately factored), that is, even if there is no explicit intermediate representation.

Furthermore, such a set of "virtual rules" may reorganize by continuous adaptation into a partially or completely different set of rules [9]. For example, through adaptation the intermediate subspace might expand to occupy a larger part of the whole intermediate space, and then contract again into a completely different subspace. Because the rules do not actually exist, and are only a descriptive device used by observers of the system, the reorganization is more in our discrete description than in the actual continuous system. At the beginning and end of the adaptive processes the system might appear to be using different rule sets, but in fact no discrete rules have been added or deleted (because there are no discrete rules).

5 Generative Specification of Well-formedness

In the theory of discrete formal systems it is usual to specify the set of WFFs by means of a *generative grammar*, that is, a nondeterministic computational process that is capable of generating all the WFFs from a fixed starting symbol. Since continuous images are not typically constructed from atomic images, the generative approach is less useful with continuous formal systems than with discrete, but it is still interesting to consider how we might go about it.

Similarly to WFFs, we can define a (fuzzy!) set of well-formed continuous images by specifying a nondeterministic process that generates them from a fixed initial image. For a very simple example, if we wanted to generate normalized two-dimensional vectors, we could start with the $(0,0)$ vector and nondeterministically increase its length until it's approximately 1. For the guiding image of such a nondeterministic process we could use a potential surface, $V(\mathbf{x}) = -f(1 - \|\mathbf{x}\|)$, where f increases monotonically from 0 (Fig. 6). Note: Instead of expressing grammars by discrete rules, we use guiding images.

For more complex, structured images, such as those found in visual perception and language, we need ways of describing recursively-structured images. This is easily accomplished for fields, or images extended over some continuum (typically some Euclidean space). In such a case the domain of one field can be continuously embedded in the domain of the other, and so the one image can be embedded in another. (Since

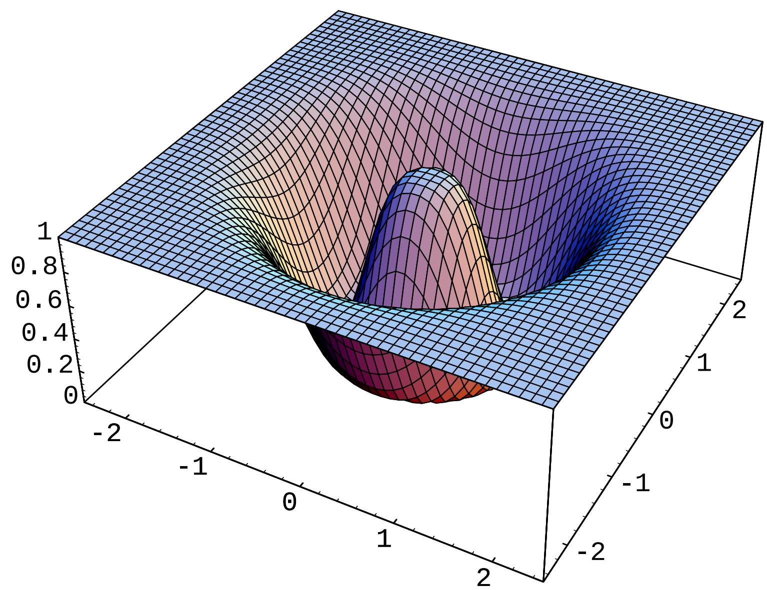


Figure 6: A simple continuous grammar. Nondeterministic descent on this potential surface from $(0, 0)$ generates normalized two-dimensional vectors.

the result is required to be continuous, we must specify some “blending function” to smooth the transition between the embedded and embedding images.) By such a procedure an image may even be embedded in itself (at a smaller scale, of course, for finite images).

To use these techniques for the generative description of well-formed images, we must of course have some method of specifying where such embeddings are permitted. Therefore, in general, certain qualities of the embedding image will constrain where embeddings of various sorts are permitted, or we will have a separate “embedding map,” generated in parallel to the embedding image, which provides this information. Of course, by the continuity principle, there must be some fuzziness in the permissibility of embedding. Further we will usually have a finite set of generative guiding images for the embeddable images of various sorts, which may in turn permit embeddings into themselves. In summary, let it be observed again that the “grammar” for the well-formed images comprises a set of guiding images.

6 Conclusions

In conclusion, I have shown, I hope, that by pursuing consistently continuous models of information representation and processing, we are led to new perspectives that will help us to understand information representation and processing in natural and artificial recurrent neural networks. There is much work yet to be done, which should be interesting as well as challenging, since we have centuries of experience in discrete knowledge representation, but very little in its continuous analogues.

References

[1] Dreyfus, H. L. (1979). *What Computers Can't Do: The Limits of Artificial Intelligence*. Harper & Row, rev. ed.

[2] MacLennan, B. J. (1987). Technology-independent design of neurocomputers: The universal field computer. In M. Caudill & C. Butler (eds.), *Proc. IEEE First Intl. Conf. on Neural Networks*, Vol. 3, pp. 39–49.

[3] MacLennan, B. J. (1988). Logic for the new AI. In James H. Fetzer (ed.), *Aspects of Artificial Intelligence*, Kluwer, pp. 163–192.

[4] MacLennan, B. J. (1991). Gabor representations of spatiotemporal visual images. Tech. Rep. CS-91-144. Comp. Sci. Dept., Univ. of Tennessee, Knoxville.

[5] MacLennan, B. J. (1993). Field computation in the brain. In K. Pribram (ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data*, Lawrence Erlbaum, pp. 199–232.

[6] MacLennan, B. J. (1993). Characteristics of connectionist knowledge representation. *Inf. Sci.*, **70**, 119–143.

[7] MacLennan, B. J. (1994). Continuous symbol systems: The logic of connectionism. In D. S. Levine & M. Aparicio IV (Eds.), *Neural Networks for Knowledge Representation and Inference*. Lawrence Erlbaum, pp. 83–120.

[8] MacLennan, B. J. (1994) "Words lie in our way." *Minds and Machines*, **4** (4), 421–437.

[9] MacLennan, B. J. (1995). Continuous formal systems: A unifying model in language and cognition. In proc. of *IEEE Workshop on Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems*, Monterey CA, Aug. 27–9, 1995.

[10] MacLennan, B. J. (1997). Field computation in motor control. In P. Morasso & V. Sanguineti (eds.), *Self-Organization, Computational Maps, and Motor Control*, Elsevier, pp. 37–73.

[11] MacLennan, B. J. (1999). Field computation in natural and artificial intelligence. *Inf. Sci.* **119** (1–2), 73–89.

[12] MacLennan, B. J. (2001). Can differential equations compute? Tech. Rep. UT-CS-01-459. Comp. Sci. Dept. Univ. of Tennessee, Knoxville.

[13] MacLennan, B. J. (in press). Transcending Turing computability. *Minds and Machines.*

[14] Pour-El, M. B. (1974). Abstract computability and its relation to the general purpose analog computer (Some connections between logic, differential equations and analog computers). *Trans. Am. Math. Soc.* **199**, 1–29.

[15] Rubel, L. A. (1981). A universal differential equation. *Bull. (N.S.) Am. Math. Soc.* **4**(3), 345–349.

[16] Rubel, L. A. (1993). The extended analog computer. *Adv. Appl. Math.* **14**, 39–50.

[17] Shannon, C. E. (1941). Mathematical theory of the differential analyzer. *J. Math. & Physics MIT* **20**, 337–354.