Minds and Machines (2005) 15: 57–71

Cognition and the Power of Continuous Dynamical Systems

WHIT SCHONBEIN
Department of Philosophy, Mount Holyoke College, 50 College Street, South Hadley, MA 01075-6401, USA; E-mail: wschonbe@mtholyoke.edu

Abstract. Traditional approaches to modeling cognitive systems are computational, based on utilizing the standard tools and concepts of the theory of computation. More recently, a number of philosophers have argued that cognition is too 'subtle' or 'complex' for these tools to handle. These philosophers propose an alternative based on dynamical systems theory. Proponents of this view characterize dynamical systems as (i) utilizing continuous rather than discrete mathematics, and, as a result, (ii) being computationally more powerful than traditional computational automata. Indeed, the logical possibility of such 'super-powerful' systems has been demonstrated in the form of analog artificial neural networks. In this paper I consider three arguments against the nomological possibility of these automata. While the first two arguments fail, the third succeeds. In particular, the presence of noise reduces the computational power of analog networks to that of traditional computational automata, and noise is a pervasive feature of information processing in biological systems. Consequently, as an empirical thesis, the proposed dynamical alternative is under-motivated: What is required is an account of how continuously valued systems could be realized in physical systems despite the ubiquity of noise.

Key words: artificial neural networks, cognition, computational complexity, connectionism, dynamic systems theory

1. Introduction

Philosophers disagree over what makes cognition tick. Some claim that cognitive behavior must be understood computationally, in terms of the operation of language-like mechanisms: Complex mental representations are assembled from basic constituents (mental 'words') according to a finite set of rules (a 'grammar'). Cognition consists in the rule-based manipulation of these mental 'sentences' (Fodor, 1975). Other philosophers have recently rejected this 'classical' picture, arguing that such mechanisms are not sufficient to fully explain cognition. In its place, they offer a framework based on dynamical systems theory (Van Gelder, 1995; Horgan and Tienson, 1996).

While these 'dynamical' approaches share a number of features in common, this paper is concerned with only one, namely, the claim that a dynamical systems framework is 'more powerful' than that of classical computation. This assertion is apparently founded on the intuition that systems defined in terms of continuous rather than discrete values – e.g., artificial neural networks with real-valued activations – offer more 'ways of behaving' than are provided by classical systems (Fields, 1989, p. 172).

Here, I argue that this claim is not sufficiently motivated; indeed, there is reason to believe that dynamical systems theory, insofar as it relies on continuous values, does *not* offer a more powerful framework for understanding cognition than traditional computation.[1] I begin by sketching the standard computational hierarchy, and in the process set forth an important condition on a system's being computational in the traditional sense: It must be discrete. In Section 3 I argue that, by virtue of the use of continuous values, dynamical systems violate this condition, and therefore may be more powerful than traditional automata. Turning to literature on the relationship between continuity and computational power, one finds that there is a particularly relevant kind of dynamical system – the analog artificial neural network – that has been shown to possess computational power greater than that of traditional computational mechanisms. In Section 4 I consider three arguments against the nomological possibility of such networks. The first two arguments are rejected, while the third is defended. In particular, the presence of intrinsic noise lessens the power of analog artificial neural networks to the point that, in normal situations, they are not any more powerful than traditional computational automata. More generally, what is lacking in the dynamicist's account are reasons to believe that the additional computational power afforded by continuous values in idealized systems will be present in nomologically possible realizations of those mechanisms. To successfully motivate their claim, it must be shown how such proposed systems can be empirically viable, i.e., comport with what we know about information processing in actual physical systems. In conclusion, I indicate how endorsing the suitability of traditional computational automata does not, in and of itself, imply that we must therefore adopt a classical approach – there are systems that are simultaneously computational and non-classical.

2. The Power of Computation

The genesis of our contemporary understanding of computation occurred in 1936 with the introduction of several formal systems, most notably the Turing machine (henceforth, 'TM'; Turing, 1936; Hopcroft and Ullman, 1979). A TM consists of a small number of simple components: A controller, a tape head, and a tape of unbounded length. The controller can be in any one of a finite number of states, q_0–q_n. The tape is divided into discrete squares, each of which can contain one of a finite set of basic symbols (the 'alphabet' for the TM). At any given moment, the tape head is positioned over a single square.

The machine runs through a series of discrete moves. Each move involves changing the state of the controller, writing a symbol to the tape, and moving the head one square left or right. The behavior of the machine can be described by a transition function from pairs of states and symbols to 3-tuples of controller states, symbols, and a direction of tape head movement. So, for example, the transition

$$\{q_2, w_1 \rightarrow q_6, w_2, \mathrm{R}\}$$

specifies that when the machine is in state q_2 and the input is the symbol w_1, it will change to state q_6, write w_2 on the tape, and move the head one square to the right. The transition table for a TM is the (finite) collection of all particular transition rules for that machine.

A TM computes the value of a function by beginning with some input finitely represented on the tape, and then running according to the deterministic procedure described by the transition table. The output of the calculation is the string remaining on the tape when the machine halts.[2]

An important feature of TMs, shared by all computational systems, is the fact that they are *finitely specifiable*. A TM uses a finite set of basic representations, and a finite set of basic operations or components, which means that any given TM can be fully described in a finite amount of space and time. Given this restriction, the representations utilized by TMs are necessarily *discrete*, for, if they differed by arbitrarily small amounts, the alphabet would not be finitely specifiable. Likewise, the transition table would contain an infinite number of transition rules to handle the infinite number of primitive symbols.[3]

Other computational automata possess finite specifiability and discreteness, although not all are as complex as TMs. For example, deterministic finite automata (DFA) lack a tape, while pushdown automata (PDA) have a memory store, but are restricted in the way they can access symbols stored there: Like a stack of trays in a cafeteria, they are only able to access the symbol (the tray) on the top.

Since different types of automata have distinct kinds of resources available for computing, we expect that they will differ in the classes of functions they can compute: The greater the resources, the more complex the functions computable. Indeed, this is the case, as automata define a hierarchy of functions depending on the resources required to calculate values of those functions. So, for example, DFA can compute certain sorts of functions, but not others, while TMs and Post systems can compute all functions computable by DFA, and many more besides.

This ordering reflects a hierarchy of computational power, with TMs and their equivalents being more computationally powerful than many other known forms of automata. In what follows, a system that possesses computational power equivalent to that of a TM is referred to as

'Turing-equivalent', while systems that are capable of computing functions not computable by TMs are 'super-Turing-equivalent'.

3. Beyond Traditional Computation

I've digressed into these rough details of the theory of computation because, whereas classicists (Jerry Fodor being the prime example) take TMs as paradigm cases of the kind of analysis that is applicable to cognition, some philosophers suggest that the explanation of cognition requires a framework *computationally* more powerful than the TM. For example, Van Gelder (1995, 1997) argues that explaining cognition requires a conceptual framework 'more powerful' (1997, p. 427) than that offered by traditional accounts of computation, since cognitive behavior may be 'much more subtle and complex than the standard concept of representation [and therefore computation] can handle' (p. 353).

Likewise, Horgan (1997) writes, 'at the mathematical level of description the alternative [dynamical systems] approach would invoke some form of mathematics more powerful than the discrete mathematics of computation theory' (p. 19), and that the 'dynamical cognition' framework 'invokes a potentially more powerful kind of mathematics' that can deal with a cognitive system that realizes 'a function so complex and subtle that it is not tractably computable' (p. 25; see also Horgan and Tienson, 1996).

In both cases, it is not completely unambiguous in what sense 'more powerful' should be understood. Nonetheless, it certainly seems reasonable to interpret the claim in terms of the computational hierarchy, given that these discussions are framed in terms of computation, the complexity of functions that must be computed, and the relative 'power' of competing information-processing accounts.[4] Furthermore, all of these authors endorse the use of continuous rather than discrete values in defining cognitive models. For example, Van Gelder notes that differential equations utilize continuous values (1995, p. 368). Horgan is more explicit, writing, 'the mathematics of dynamical systems is fundamentally continuous mathematics rather than discrete mathematics' (p. 19).

This endorsement amounts to a rejection of Turing's stipulation that the states of a computational system be discrete: If the boundaries of states (and/or symbols) of a system are defined in terms of infinitely precise values, these states (and/or symbols) can differ by an arbitrarily small degree. Since computation as defined in terms of TMs requires principled transitions between discrete states, continuous systems differ fundamentally from traditional automata.

Horgan (1997) goes on to suggest that suitably modified neural networks are the sorts of things that could realize such 'non-computational' systems.

He writes, 'dynamical systems whose transitions are computable are actually a relative rarity, and it is certainly possible for noncomputable dynamical systems to be subserved by neural networks – at least if the networks are made more analog in nature by letting the nodes take on a continuous range of activation values, and/or letting them update themselves instantaneously rather than by discrete time steps.' (p. 25).

This suggestion is interesting because there has in fact been considerable work done on the computational power of continuous artificial neural networks (Siegelmann and Sontag, 1994; Siegelmann, 1999; Siegelmann, 2000). What's more, these *analog* artificial neural networks (AANNs) possess computational power greater than that of TMs. The exact nature of this super-Turing-computability is not important in the current context, so I will not elaborate.[5] However, these networks are relatively simple: They are first-order, recurrent, and synchronously updated; they use saturated-linear activation functions; and have a finite number of nodes. If one restricts the activation and weight values to being rational (and hence specifiable as ratios between integers), these networks are Turing equivalent. However, if one allows for continuous weights and activations, AANNs are capable of computing functions not computable by TMs. Thus real values are central to their super-Turing capacity. So, just as classicists have the Turing machine, those who endorse dynamical systems have available an actual (if idealized) mechanism to serve as an example of the sort of system that is purportedly needed.

To sum, dynamicists (as represented by Van Gelder, Horgan, and Tienson) can be understood as claiming that the explanation of cognition requires models that possess computational power greater than that of TMs. Part of this claim revolves around the use of formalizations that make use of continuous values, which implies that these systems cannot be understood in terms of classical computation. Furthermore, certain sorts of neural networks, when provided with real-valued weights and activations (as suggested by Horgan) are in fact computationally more powerful than TMs, and therefore super-Turing-equivalent systems are at least *logically* possible. However, a number of arguments have been offered to the conclusion that AANNs (and other continuously valued automata) are *nomologically* impossible, i.e., not realizable by physical systems in our world.

4. Three Arguments against AANNs

There are at least three arguments against the empirical utility of analog systems for understanding information processing in physical systems. Here I address each in turn.

4.1. THE ARGUMENT FROM REPRESENTATION

On the use of infinite-precision values, Hadley (2000) writes:

> [I]t is known that biological 'neural networks' and even standard digital computers do not have the capacity to represent weights of infinite precision. Moreover, it appears likely that this limitation applies not only at the lowest (neural or circuit) level, but at more abstract levels of processing. Certainly, no one has discovered any means whereby, in a finite span of time, a finite precision substrate can operationally represent values requiring an infinite sequence of numerals to represent. (Hadley, 2000, p. 106)

While it is unclear whether the author intends the passage to be interpreted as an argument against continuously valued weights, it nonetheless suggests one: Such weights ought to be rejected on the grounds that the physical medium that is to realize them – electrochemical potentials, spiking frequencies, etc. – provides only a finite number of potential representational states, and therefore cannot represent values that require an infinite series of numerals to encode.[6] Schematized, the argument becomes:

1. No finite physical device has infinite representational capacity.
2. An infinite-precision value requires infinite representational capacity to represent that value.
3. Therefore, no physical device uses infinite-precision values in performing computations.

The first assumption is *prima facie* straightforward – no finite physical device has an infinite number of representational states. The motivation for the second assumption is that, in order to represent a number in a physical system, we need a sequence of digits (e.g., '0', '1', '2',. . .,'9' in decimal, '1' and '0' in binary, etc.). Yet each digit in an infinitely precise number must be (physically) present somewhere, and there are an infinite number of digits. Hence an infinite amount of 'representational media' is required. Since no physical device has an infinite representational capacity, none can make use of infinitely precise values.

This argument is sound under at least one interpretation. Consider what is required in order to build a computer. We begin by identifying physical parameters that behave in regular ways; for example, voltage passing through a wire can be held at two different stable states, high and low. With such reliable behavior guaranteed, a labeling is assigned to the states of the wire, e.g., a high-voltage state represents '1', and a low state represents '0', and hence the numbers 1 and 0. Given such a binary labeling system, an array of wires can be interpreted as representing additional numbers by providing a fixed mapping of the array into the number system. So, just as high and low voltages are mapped to 1 and 0, we establish an ordering of the wires in the array from most significant to least significant binary digit, and thus provide a principled mapping from the array of wires to integers: '000' $\rightarrow$ 0, '001' $\rightarrow$ 1, '010' $\rightarrow$ 2, '011' $\rightarrow$ 3, . . . In this way, the system is shown to be capable of representing a set of integers.

The essential ingredient in this construction is that states of the physical system are interpretable as representing numbers by way of being placed in correspondence with one or more of our numerical schemes, in this case binary. Given the finite amount of representational resources of the system, it is obvious that we cannot extend this process indefinitely, using more and more wires to represent larger and larger numbers, for at some point we will run out of wires and hence out of digits. Consequently, the represented number is always of finite precision, and infinite-precision values are thereby excluded.

Now, one possible response might be to argue that, by eliminating the possibility of an infinite number of representational states, the first premise begs the question against the use of continuous values. However, it seems to me that there is a more fundamental difficulty with the argument.

There is an important distinction between using physical states to represent numbers (as in the above example) and the task of accurately modeling the behavior of physical systems using mathematical formalisms. In the former case, a physical state is interpreted as a representation of a number, while in the latter a variable appearing in a model is intended to be a representation of a physical magnitude. Clearly, when constructing a model of a physical process, it is the mathematical model that must represent the physical process, not vice versa.

This observation is underscored through a consideration of a relatively standard notion of representation. A widely accepted feature of representation is that a representation 'stands in' for the thing it represents when that thing is absent (Haugeland, 1991). For example, some representation of the layout of objects upon my desk serves as a 'surrogate' in guiding my hand towards a glass of water when my eyes are fixated on the computer terminal. Moreover, we explain the movement of my hand by reference to the content of this representation: My hand makes such-and-such a swerve because somewhere in the motor-control circuit governing the movement of my limb is a representation of there being an object in the way. Similarly, it makes sense to conceive of the states of our homemade computer as standing in for the presence of numbers when performing some calculation, and to invoke the content of those representations in accounting for its behavior.

In contrast, it is counterintuitive to think of the states of some (naturally occurring) physical system as standing in for the variables of a model of that system, because (typically) those values (considered as the representational contents of the states) are not relevant to explaining its behavior. That is, a modeled physical system does not manifest its behavior because it tokens states that have values of the model as their representational contents. For example, we do not say that an automobile accelerates at such-and-such a rate because it tokens some state that has, as its representational content, some number of Newtons of force, even though the behavior of the automobile may be accounted for by Newton's second law.

The problem, then, is that the argument gets the representation relation the wrong way around. The homemade computer was designed to represent numbers according to our binary representational scheme, and realizing such a scheme places constraints on the organization of the physical system. In contrast, the possibility of a physical system realizing some process that requires a certain type of model to adequately capture does not depend on the features of the representational system of the model, because it is the model that must answer to the physical process.[7]

All of this has an obvious application to the case of AANNs: It does not follow from the fact AANNs utilize real-valued weights that a physical system modeled by an AANN represents the values of those weights in the sense required to justify the claim that no such physical systems are possible. As a result, the argument from representation fails to discount their potential utility for the proponent of dynamical systems theory.

4.2. THE ARGUMENT FROM MEASUREMENT

Fields (1989) offers a different argument against the relevance of AANNs for explaining behavior. His argument is based on the possibility of putting states of a physical system into correspondence with those of a formally defined automaton such as a TM. The argument is as follows:

1. If a virtual machine *M* (e.g., a TM) is to be realized in a physical system *S*, it must be possible to put states of *M* into correspondence with states of *S*.
2. In order to construct this mapping, the states of *S* must be measured.
3. Measuring a state of *S* involves adding energy to the system.
4. The more precise our measurements are – the 'smaller' the state being measured is – the more probable it is that our measurements will influence the behavior of the system. Thus measuring a state will result in *S* not making the transition to the state it would have entered had it not been perturbed.
5. Therefore, there is an upper bound on the precision we can achieve in measuring the states of *S*.
6. Therefore, the only correspondences that can be constructed are those that possess a finite number of states.
7. Therefore, a TM can compute any machine realized by *S*.

Fields concludes that, provided we accept premises 3 and 4, 'a continuous dynamical system cannot, even in principle, exhibit behavior that cannot be simulated by a universal Turing machine.' (p. 171)

There are a number of problems with this argument. For example, it does not follow from the conclusion stated in line 7 that a continuous system cannot in principle be computationally more powerful than a TM – AANNs are just such continuous systems. More fundamentally, it is fallacious to infer from our lack of ability to measure the states of a system to the conclusion that those unmeasured (or unmeasureable) states are not relevant to the behavior of the system – to do so is to confuse our metaphysics with our epistemology. For example, it may be the case that the arrangement of quarks in the atoms comprising my desktop happen to be realizing a virtual machine calculating π to the nth decimal place, but we have no way of measuring those states short of blasting the desk to smithereens. Yet this does not imply my desktop is not in fact realizing that virtual machine.

Hence the conclusion stated in line 7 does not follow from the premises. What follows is a weaker claim: A TM can compute any machine determined through known methods of measurement to be realized in S. However, we are not interested in what it is possible to determine through measurement, but rather what is really going on in the relevant physical systems.

4.3. THE ARGUMENT FROM NOISE

Despite the failure of the previous two arguments, they suggest a third. Take, for example, Fields' observation that measurement can perturb the system being measured to the point where it ceases to behave in the same way it would have had it been left undisturbed. This situation is in fact commonplace – only it has nothing to do with measurement, but rather merely with *noise*.[8]

It is well known that the transmission of information between neurons in biological neural networks involves an element of intrinsic noise (e.g., Rieke, 1997). If we conceive of the ideal case of inter-node communication in an AANN as involving infinite-precision weights, then successful communication in real-world contexts (i.e., despite noise) indicates that not all the precision provided by the posited real values is required for the system to function normally. This is because noise renders the lesser-significant bits useless (i.e., unreliable) for the purposes of carrying out the relevant computation, and therefore these bits can be ignored.

If the presence of noise removes the utility of lesser-significant digits for carrying information, we should expect the computational power of systems that rely on those digits to be reduced in the presence of noise. Indeed, AANNs subjected to noise are reduced in computational power to finite automata, and often to a power less than that of finite automata (Maas and Orponen, 1998; Casey, 1996; Maas and Sontag, 1998). When subjected to typical noise, AANNs do not exhibit computational power greater than TMs, and will probably be reduced to levels below that of TMs. In short, AANNs only enjoy super-Turing-computability under ideal circumstances, and such circumstances are not what we find in actual cognitive systems.[9]

One response to this line of argument is to grant the presence of noise but deny that it is relevant to the current argument. For example, in order for an AANN to reduce to a TM (or less), it must be the case that we can divide the activation space of the system into a set of discrete states, the boundaries of which are defined by real values. Applying noise to these values effectively renders all bits less significant than a certain point useless for the purpose of computation. However, noise is random. Therefore, we cannot determine *a priori* which bits will be ineffectual: At one time it may be at the nth bit, while at another it may be at the $n + m$th bit. So (the response goes) we cannot segment the space of the network in such a way as to yield a set of discrete states, as required by computation, traditionally defined, since the boundaries of the desired state are constantly shifting.

The flaw in this response lies in assuming that any bit not occluded by noise is thereby relevant to computation. Whether or not a bit will be used depends not only on the information it carries, but also on whether or not that bit is a *reliable* carrier of information. A bit that carries useful information only very infrequently will not be used by the system, since the use of that bit would usually generate incorrect results. In other words, the same randomness that renders the number of significant bits uncertain also forces the system not to rely on those bits that typically are subject to error.[10]

A second response to the suggestion that noise reduces the computational power of continuous systems is to insist that, in contrast to traditional computational approaches, the very point of invoking analog systems such as AANNs in the explanation of intelligent behavior is to take noise into account in the infinite precision of variables such as weights and activations. In short, this response argues that noise is built into the system.

The essence of this response is the assertion that noise is equally relevant for the production of cognitive behavior as is the content of those states that entered into the production of that behavior. This is a rather radical proposal, for it rejects the standard methodology of distinguishing between the informational content of states mediating between perception and action and random perturbations in those states. Suppose, for example, that states of the network are to be interpreted as entities in some scientific belief/desire psychology, e.g., data structures comprised of concepts standing in various relations.[11] Then, the proposal is that noise is always relevant to understanding action, inference, or thought as such. But this position is extremely counterintuitive, since standard belief/desire explanations assume that noise is not the cause of behavior. For example, the reason Bob went to the store and returned with milk is not because such-and-such a random perturbation occurred at some point in the processing of information by the neural networks constituting his brain, but rather because he desired milk and believed he could acquire it at the store. By collapsing the distinction between noise and content, this proposal fails to honor a basic commitment of belief/desire psychology.

This commitment arises from the fact that belief/desire psychology explains behavior by showing how particular events fall under generalizations. In Bob's case, the relevant generalization is roughly: anyone who desires something and believes it can be obtained at a certain location will take steps to reach that location, *ceteris paribus*. The difference between Bob's being subsumed or not subsumed by the generalization is precisely whether or not his behavior is the result of tokening representational states with certain contents in certain relations. However, the proposed response renders this question unintelligible. The problem is that the possibility of establishing and using generalizations involving states with representational content depends on abstracting away from noise, e.g., by invoking *ceteris paribus* clauses or utilizing statistical methods. Moreover, because the goal of establishing such generalizations is not unique to belief/desire psychology, but is rather a feature of cognitive science in general, it follows that maintaining the distinction between noise and content is of paramount importance. Consequently, the proposal that noise should be treated as equally relevant for the production of cognitive behavior ought to be rejected.

To sum, even when continuously valued systems are computationally more powerful than TMs, they may not be so when realized in physically possible mechanisms. The case of AANNs illustrates this situation, as their super-Turing-equivalence disappears under the effects of noise characteristic of information processing in naturally occurring systems. As a result, the dynamicist's appeal to continuous systems requires further motivation, for, whereas there is evidence for both the presence of noise in biological information processing systems and for its adverse effects on the computational power of continuous systems, there is no well-formulated example of a continuously valued, super-Turing-equivalent system whose computational power is not affected by noise.

5. Conclusion

I have argued that, while representation and measurement pose no problems for the possible relevance of super-Turing computability for explaining cognition, noise constitutes a serious obstacle. More specifically, despite there being various abstract automata that possess super-Turing computational power – most notably analog artificial neural networks – their performance under normal circumstances may not exhibit this feature. This detracts from the dynamical systems perspective, since what is lacking from appeals to these sorts of systems is any indication that they are the kinds of things that, when physically realized, retain their additional computational power.

If super-Turing-computable systems are nomologically suspect, does this provide support for classicism? In the introduction, I contrasted the dynamical systems approach with that of 'classical' accounts, such as that view espoused by Jerry Fodor (1975, 1987). Turing machines are considered prototypical examples of classical systems, and, since continuous automata such as AANNs are non-discrete, they constitute a distinct sort of information processing framework from classical mechanisms. Yet I've argued that nomologically possible information processing systems cannot take advantage of the computational benefits of continuous values, and as a result, some form of traditional automata will have to do the job. Nonetheless, we are not thereby forced to endorse a classical approach.

Being interpretable as the operation of a traditional automaton cannot be *sufficient* for being classical, on the classicist's own account (although it is necessary). For, if it were the case that being so describable were sufficient, then, if a physical system computes a Turing-computable function, it is, as a matter logical implication, classical. But the classical hypotheses – that cognition consists in the rule-based manipulation of language-like representations – is an *empirical* one (McLaughlin, 1993). Therefore, it must be (logically) possible on the classical account for there to be Turing-equivalent systems that nonetheless are non-classical. In short, mere (traditional) computation is not sufficient for classicality.

Classical mechanisms such as Turing machines should therefore be viewed as constituting a *subset* of possible Turing-equivalent computational systems. Dynamical systems such as AANNs, while not offering greater computational power, may nonetheless offer something else – something not provided by classical mechanisms, yet still computational in the traditional sense. Exploring this possibility is left as a future exercise.

Acknowledgements

For valuable comments on earlier drafts of this paper, gratitude is extended to Chase Wrenn, Joe Cruz, Chris Eliasmith, and an anonymous reviewer of an earlier draft of this paper, as well as attendees of the 2001 Society for Philosophy and Psychology annual meeting.

Notes

[1] There may be some other feature or features of dynamical systems that may contribute to their potentially being 'more powerful' than traditional computational automata; the current paper is concerned only with the endorsement of continuous values.

[2] There are other ways to conceive of TMs besides as calculating the values of integer functions. TMs can be understood as language recognizers, i.e., mechanisms for deciding whether a given input is a member of a class of expressions, or as enumerators of members of a language. In the present context, nothing of substance turns on these distinctions.

[3] Note that the unbounded size of a Turing machine's tape is not an obstacle to finite specificity. Since all unspecified tape squares are merely blank, they do not require explicit description.

[4] An interpretation of Horgan and Tienson's position is complicated by the fact they use the term 'intractable' in a non-standard way (i.e., not in accordance with its use in the theory of computation). For Horgan and Tienson, 'tractable' means roughly 'doable by a physical system' (Horgan, 1997, p. 5, fn. 4), rather than 'computable in non-polynomial time'. This makes it difficult to determine which of the following claims they wish to endorse (if either): (I) The relevant cognitive function *F*, when computed in terms of a TM, is intractable (in Horgan and Tienson's sense of the term). When rendered in terms of dynamical systems theory, the function is rendered tractable. In short, using a dynamical systems approach gets us the same result but does so much faster. (II) The relevant cognitive function *F* is not computable in terms of a TM, but is nonetheless 'dynamical-systems-computable', that is, within the power of some dynamical system to accommodate. In this case, dynamical systems are super-Turing-equivalent. One problem with (I) is that, by assumption, *F* is computable in terms of a TM, and therefore can be understood as the operation of a system that manipulates discrete symbols. The proposed dynamical system makes no essential use of its continuous nature (except insofar as it speeds things up). Given this consideration, (II) is preferable to (I), as only it makes sense of claims concerning the 'power' of dynamical systems while also guaranteeing the proffered mechanisms are intrinsically different than traditional computational automata. Further motivation for this interpretation is given in the main text, in discussing the use of continuous values.

[5] In fact, justifying that the results concerning AANNs as defined by Siegelmann have any relevance for actual physical systems is no easy task. For example, super-Turing computability only emerges when one allows the system unbounded time in which to compute. In this paper, I eschew any discussion of these particulars in favor of pursuing more general problems with the use of continuous values.

[6] Because it is possible to interpret this passage as not containing any argument, but rather as expressing the assumption that physical substrata cannot realize continuously valued models, the argument from representation should not be attributed to Hadley without further confirmation. In any event, Hadley's project as presented in his (2000) does not rely to any great extent on the argument from representation.

[7] In making this argument I am not claiming that it is not justifiable to infer from the necessary use of a real value in a model to the conclusion that there is some feature of the physical system that varies with that value. If a model must use a real value, and that value tracks some physical property, then there is something about that property that distinguishes it from those that do not require the use of such a value. The question with which I am concerned, however, is whether the antecedent of this conditional will ever be satisfied – i.e., whether we currently have reason to conclude that there will be no such necessary uses – and this question is independent of concerns over the justification of the above inference.

[8] Chris Eliasmith (2001) has also parlayed the intrinsic noisiness of physical instantiations of cognitive systems into an argument against the utility of continuously valued dynamic systems

models for understanding cognitive processes. While we reach the same or similar conclusion, our respective paths to that end differ, and the present paper should be viewed as complementary.

[9] These results all involve noise distributions where the noise is almost always non-zero. However, when noise distributions are sparse, i.e., only non-zero in a small number of cases, the presence of noise can actually boost the computational power of an AANN (Siegelmann, 1999). Similarly, work on modeling biological neurons has shown that the effect of noise on the ability of a network to process information depends on where noise is applied in the network. However, naturally occurring noise distributions are of the first sort: They are usually non-zero.

[10] I do not mean to deny that there may be instances in which bits that do not reliably carry information do in fact cause changes in behavior, or that in these cases the resulting behavior might not have some utility. For example, in robotics it is sometimes useful to utilize noise to lessen the chances of a robot becoming trapped in local minima during navigation. However, in most cases noise will not be the relevant factor in determining behavior – instead, informational content will be, and tracking the processing of information will not require super-Turing computational power.

[11] Nothing in the argument turns on the choice of belief/desire psychology. This example was chosen on the basis of philosophic tradition in discussing connectionist networks.

References

Casey, M.P. (1996), 'The Dynamics of Discrete-Time Computation with Application to Recurrent Neural Networks and Finite State Machine Extraction', *Neural Computation* 8(6), pp. 1135–1178.

Davis, M. (1965), *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, Hewlett, NY: Raven Press.

Eliasmith, C. (2001), 'Attractive and In-discrete', *Minds and Machines* 11, pp. 417–426.

Fields, C.A. (1989), 'Consequences of Nonclassical Measurement for the Algorithmic Description of Continuous Dynamical Systems', *Journal of Experimental and Theoretical Artificial Intelligence* 1, pp. 171–189.

Fodor, J. (1975), *The Language of Thought*, Cambridge, MA: MIT Press.

Fodor, J. (1998), *Concepts: Where Cognitive Science Went Wrong*, Oxford, UK: Oxford University Press.

Fodor, J. and Lepore, E. (1992), *Holism: A Shopper's Guide*, Oxford, UK: Blackwell Publishers.

Hadley, R.F. (2000), 'Cognition and the Computational Power of Connectionist Networks', *Connection Science* 12(2), pp. 95–110.

Haugeland, J. (1991), 'Representational genera', in W. Ramsey, S.P. Stich and D.E. Rumelhart, eds., *Philosophy and Connectionist Theory*, Hillsdale, NJ: Erlbaum.

Hopcroft, J.E. and Ullman, J.D. (1979), *Introduction to Automata Theory, Languages and Computation*, Reading, MA: Addison-Wesley.

Horgan, T. and Tienson, J. (1996), *Connectionism and the Philosophy of Psychology*, Cambridge, MA: MIT Press.

Horgan, T. (1997), 'Connectionism and the Philosophical Foundations of Cognitive Science', *Metaphilosophy* 28, pp. 1–30.

Maass, W. and Orponen, P. (1998), 'On the Effect of Analog Noise on Discrete Time Analog Computations', *Neural Computation* 10(5), pp. 1071–1095.

Maass, W. and Sontag, E. (1999), 'Analog Neural Nets with Gaussian or Other Common Noise Distribution Cannot Recognize Arbitrary Regular Languages', *Neural Computation* 11, pp. 771–782.

McLaughlin, B. (1993), 'The Connectionism/Classicism Battle to Win Souls', *Philosophical Studies* 71, pp. 163–190.

Rieke, F. (1997), *Spikes*: *Exploring the Neural Code*, Cambridge, MA: MIT Press.

Siegelmann, H.T. and Sontag, E.D. (1994), 'Analog Computation via Neural Networks', *Theoretical Computer Science* 131, pp. 331–360.

Siegelmann, H.T. (1999), *Neural Networks and Analog Computation*: *Beyond the Turing Limit*, Boston: Birkhäuser.

Siegelmann, H.T. (2000), 'Finite versus infinite neural computation', in C.S. Calude and G. Paun, eds., *Finite vs Infinite*: *Contributions to an Eternal Dileitmma*, London: Springer.

Turing, A.M. (1936), 'On Computable Numbers, with an Application to the Entscheidungsproblem', *Proceedings of the London Mathematical Society* 42(series 2), pp. 230–256.

Van Gelder, T. (1995), 'What Might Cognition be, if not Computation?', *Journal of Philosophy* 91(7), pp. 345–381.

Van Gelder, T. (1997), 'Dynamics and Cognition', in J. Haugeland, ed., *Mind Design II*, Cambridge, MA: MIT Press, pp. 421–476.