COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 34 (2013) 453–488
Copyright © 2012 Cognitive Science Society, Inc. All rights reserved.
ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/cogs.12012

Neural Computation and the Computational Theory of Cognition

Gualtiero Piccinini,[a] Sonya Bahar[b]

[a]*Center for Neurodynamics, Department of Philosophy, University of Missouri – St. Louis*
[b]*Center for Neurodynamics, Department of Physics and Astronomy, University of Missouri – St. Louis*

Received 29 August 2011; received in revised form 20 March 2012; accepted 24 May 2012

Abstract

We begin by distinguishing computationalism from a number of other theses that are sometimes conflated with it. We also distinguish between several important kinds of computation: computation in a generic sense, digital computation, and analog computation. Then, we defend a weak version of computationalism—neural processes are computations in the generic sense. After that, we reject on empirical grounds the common assimilation of neural computation to either analog or digital computation, concluding that neural computation is sui generis. Analog computation requires continuous signals; digital computation requires strings of digits. But current neuroscientific evidence indicates that typical neural signals, such as spike trains, are graded like continuous signals but are constituted by discrete functional elements (spikes); thus, typical neural signals are neither continuous signals nor strings of digits. It follows that neural computation is sui generis. Finally, we highlight three important consequences of a proper understanding of neural computation for the theory of cognition. First, understanding neural computation requires a specially designed mathematical theory (or theories) rather than the mathematical theories of analog or digital computation. Second, several popular views about neural computation turn out to be incorrect. Third, computational theories of cognition that rely on non-neural notions of computation ought to be replaced or reinterpreted in terms of neural computation.

Keywords: Analog computation; Computational theory of cognition; Digital computation; Neural computation; Spike trains

The brain is an analogical machine, not digital. (Lashley, 1958, p. 539)

[A]ny [psychological] theory that is incompatible with known facts of neurology must be, ipso facto, unacceptable. (Fodor, 1965, p. 176)

Correspondence should be sent to Gualtiero Piccinini, Center for Neurodynamics, Department of Philosophy, University of Missouri – St. Louis, St. Louis, MO 63121. E-mail: piccininig@umsl.edu

McCulloch and Pitts (1943) were the first to argue that neural activity is computation and that neural computation explains cognition (Piccinini, 2004). McCulloch and Pitts argued that neural computation is digital; Karl Lashley and others countered that it is analog. Their view is the origin of computationalism, or the computational theory of cognition.

Some authors (Fodor, 1975; Newell & Simon, 1976; Pylyshyn, 1984) formulate computationalism as the thesis that cognition is computation or that computation explains cognition—without reference to the nervous system. Since cognition in biological systems is a function of the nervous system, the computations that putatively explain biological cognition are carried out by the nervous system. Following the mainstream literature, we refer to them as *neural computations*. The topics of this article are the explanation of biological cognition and the nature of neural computation. Consequently, for present purposes, we define *computationalism* as the thesis that neural computation (simpliciter) explains cognition. This does not rule out that there may be artificial forms of cognition explained by other—non-neural—types of computation.

Over the last six decades, computationalism—in its various classicist, connectionist, and neurocomputational incarnations[1]—has been the mainstream theory of cognition. Many cognitive scientists consider it commonsensical to say that neural activity is computation and that computation explains cognition. However, they still disagree about *which type* of computation explains cognition. In addition, computationalism itself remains controversial.[2]

In this article, we propose to push the debate forward by establishing a sound neurocomputational basis for the explanation of cognition. In Section 1, we distinguish computationalism from a number of other theses that are sometimes conflated with it. In Section 2, we distinguish between several important kinds of computation. In Section 3, we defend a weak version of computationalism—neural processes that explain cognition are computations in a generic sense. Section 4 briefly rebuts some objections to computationalism. In Sections 5–7, we reject the common assimilation of neural computation to either analog or digital computation, concluding that neural computation is sui generis. Finally, in Section 8, we highlight how a proper understanding of neural computation affects the theory of cognition.

1. Five theses about the nervous system

To prepare the ground, let us distinguish computationalism properly so called from other related theses it is sometimes conflated with.

(1) The nervous system is an *input–output* system.

This thesis holds that neural activity generates responses by processing stimuli. This goes beyond simple ''input–output'' behaviorism because it allows reference to internal processes that mediate between inputs and outputs. Inputs come into the nervous system, are processed, and such processing affects the output. This is obviously true but says nothing about the nature of the inputs, outputs, and internal processes. In this sense, the nervous system is analogous to any system that generates outputs in response to inputs via internal processes. If the notion of input and output is liberal enough, any physical system belongs in the class of input–output systems. If so, then thesis (1) is equivalent to the trivial thesis that the brain is a physical system. But even if the notion of input and output is more restricted—for example, by specifying that inputs and outputs must be of a certain kind—thesis (1) remains weak.

(2) The nervous system is a *functionally organized* input–output system.

Thesis (2) strengthens (1) by adding a further condition: Neural activity has the *function* of generating specific responses from specific stimuli. In other words, the nervous system is functionally organized to exhibit certain capacities and not others. Under this stronger thesis, the nervous system belongs to the specific class of physical systems that are functionally organized. Systems of this type include computers as well as systems that intuitively do not perform computations, such as engines, refrigerators, and stomachs. This second thesis is stronger than the first one but remains weak. Again, there is no reason to doubt it.

(3) The nervous system is a *feedback control*, functionally organized, input–output system.

Thesis (3) is stronger than (2) because it specifies the function that is peculiar to neural activity: The nervous system controls the behavior of an organism in response to stimuli coming from the organism's body and environment that include the effects of the nervous system itself. The specific function of neural activity differentiates the nervous system from other functionally organized systems. Under thesis (3), the nervous system belongs in the class of feedback control systems together with, for instance, autopilot systems. Thesis (3) is not trivial and took serious scientific work to establish rigorously (Cannon, 1932). Nevertheless, it is by now uncontroversial.[3]

(4) The nervous system is an *information-processing*, feedback control, functionally organized, input–output system.

Thesis (4) is stronger than (3) because it specifies the means by which the nervous system exerts its feedback control function. Those means are the processing of information. The term ''information'' is ambiguous between several senses. We will briefly mention four.

In the thermodynamic sense, information is closely related to entropy. Everything carries information and every physical process is an instance of information processing in this sense. This notion of information is too general to be directly relevant to theories of cognition. We will leave it aside.

In a first relevant sense, ''information'' means mutual information in the sense of communication theory (Shannon & Weaver, 1949). Mutual information is a measure of the statistical dependency between a source and a receiver. It is experimentally well established that neural signals such as firing rates are statistically dependent on other variables inside and outside the nervous system. Thus, mutual information is often used to quantify the statistical dependence between neural signals and their sources and to estimate the coding efficiency of neural signals (Dayan & Abbott, 2001; Chapter 4; Baddeley, Hancock, & Földiák, 2000). It should be uncontroversial that the nervous system processes signals that carry information in this sense, although this is still a fairly weak notion of information *processing*.

In a second relevant sense, ''information'' means what we call ''natural semantic information,'' which is approximately the same as ''natural meaning'' (Grice, 1957). A signal carries natural semantic information *about* a source if and only if it causally, or at least reliably, correlates with the source (Dretske, 1981). This second relevant sense of ''information'' is different from the first because it is a kind of *semantic* content, whereas mutual information is not a measure of semantic content. Nevertheless, the two senses are conceptually related because whenever there is mutual information between a signal and a source, there is by definition a reliable correlation between source events and receiver events. Thus, the receiver acquires natural semantic information *about* the source.

Once again, it is both evident to naïve observation and experimentally well established that neural variables carry natural semantic information about other internal and external variables (Adrian, 1928; Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1999; cf. Garson, 2003). Much contemporary neurophysiology centers on the discovery and manipulation of neural variables that correlate reliably—and often usefully—with variables external to the nervous system.

Given (3) and the complexity of the nervous system's control functions, thesis (4) follows: In order to exert sufficiently complex control functions, the only efficient way to perform feedback control is to possess and process internal variables that correlate reliably with relevant external variables. These internal variables must be processed in a way that is explained by the natural information they carry. Thus, it should be uncontroversial that the nervous system processes information in the sense of processing signals based on the natural semantic information they carry.

Caveat: Many cognitive scientists appear to use the term ''representation'' for internal states or variables that carry natural semantic information. This is a weak sense of ''representation.'' Representation as usually understood by philosophers allows for the possibility of *misrepresentation* (Dretske, 1986; Millikan, 2004). For instance, the sentence ''this is an armadillo'' can be used either to accurately represent anything that is an armadillo or to misrepresent anything that is *not* an armadillo. However, natural semantic information cannot misrepresent. A variable that does not correlate with a source does not misrepresent; it simply fails to carry natural information about the source. Thus, natural semantic information represents only in a weak sense. In this weak sense, the nervous system is surely a representational system. Most philosophers of cognitive science, however, reserve the term ''representation'' for states or variables that can be ''wrong'' or misrepresent. From now on, we will follow this usage.[4]

The stronger notion of representation is related to a third relevant notion of information (Scarantino & Piccinini, 2010). In this sense, ''information'' means what we call ''non-natural semantic information,'' which is approximately the same as ''non-natural meaning'' (Grice, 1957). A signal carries non-natural semantic information *about* x if and only if it represents x a certain way, whether x is the way it is represented. In our example, the utterance ''this is an armadillo'' carries non-natural semantic information about whatever ''this'' refers to, and it represents it as an armadillo. The represented object may be a *source* of the non-natural information carried by a signal, but it need not be. In fact, a signal may represent objects that do not even exist.

Many theories in cognitive science assume that nervous systems possess and process representations (carrying non-natural semantic information). Specifically, it is difficult to explain language processing and other higher cognitive processes without postulating representations. Although some authors doubt that representations are needed in a theory of cognition (e.g., Ramsey, 2007), we shall follow the mainstream and assume that nervous systems process representations.

The upshot of our discussion of thesis (4) is that there is at least one important and uncontroversial sense of information processing—meaning the functionally organized processing of signals based on the (either natural or non-natural) semantic information they carry—in which the nervous system processes information.

(5) The nervous system is a computing system.

Thesis (5) is presupposed by any generic form of computationalism. To evaluate computationalism and assess its relation to (1–4) requires saying more about what counts as computation in a physical system such as the nervous system. We will get to that in the next section, where we will explicate computation independently of information processing.

We can already highlight some relationships between the five theses. As we already pointed out, each of (2), (3), and (4) is stronger than its predecessor.

While strictly speaking there can be computing systems without inputs, outputs, or both, we disregard them here. In addition, following Piccinini (2007a, 2008), we assume that any computing system that performs nontrivial computations is functionally organized to perform its computations. With these assumptions in place, computing (5) entails having inputs and outputs (1) and being functionally organized (2). There is no entailment in the other direction. For there are plenty of systems—including functionally organized systems such as the animals' locomotive, circulatory, and respiratory systems—that do not perform computations in any relevant sense. So (2) and a fortiori (1) do not entail (5).

Computing (5) entails neither feedback control (3) nor information processing (4). This is because computing systems need not perform any feedback control or information-processing functions—an example would be a computer programmed to manipulate[5] meaningless digits according to some arbitrary rule. We postpone discussing whether (3) and (4) entail (5) until after we say more about what counts as computing.

2. Computation: Generic, digital, and analog

Drawing from previous work on the nature of computation (Piccinini, 2007a, 2008; Piccinini & Scarantino, 2011), we now sketch what it takes for a physical system to perform (or implement) computations of a certain kind. We distinguish between computation in a generic sense and its two most relevant species: digital and analog computation.

2.1. Generic computation

Concrete computing systems are functionally organized mechanisms of a special kind—mechanisms that perform concrete computations. A functionally organized mechanism is a system of organized components, each of which has functions to perform (cf. Bechtel, 2008; Craver, 2007; Glennan, 2002; Wimsatt, 2002). When appropriate components and their functions are organized and functioning properly, their combined activities constitute the capacities of the mechanism. Conversely, when we look for an explanation of the capacities of a mechanism, we decompose the mechanism into its components and look for their functions and organization. The result is a mechanistic explanation of the mechanism's capacities—an explanatory strategy familiar to both biologists and engineers.

In order to capture all uses of ''computation'' in cognitive science, we need a broad notion of computation. We call it *generic computation*. Digital computation and analog computation turn out to be species (among others) of generic computation.

Computation in the generic sense is the processing of vehicles (defined as entities or variables that can change state) in accordance with rules that are sensitive to certain vehicle properties and, specifically, to differences between different portions (i.e., spatiotemporal parts) of the vehicles. A rule in the present sense is just a map from inputs to outputs; it need not be represented within the computing system.[6] Processing is performed by a functionally organized mechanism, that is, a mechanism whose components are functionally organized to process their vehicles in accordance with the relevant rules. Thus, if the mechanism malfunctions, a miscomputation occurs.

When we define concrete computations and the vehicles that they manipulate, we may consider only the properties that are relevant to the computation. As concrete computations and their vehicles are defined independently of the physical media that implement them, we call them ''medium independent'' (cf. Garson, 2003).

In other words, vehicles are medium independent if and only if the rule (i.e., the input–output map) that defines a computation is sensitive only to differences between portions of the vehicles along specific dimensions of variation—it is insensitive to any more concrete physical properties of the vehicles. Put yet another way, the rules are functions of state variables associated with a set of functionally relevant degrees of freedom, which can be implemented differently in different physical media. Thus, a given computation can be implemented in multiple physical media (e.g., mechanical, electro-mechanical, electronic, and magnetic), provided that the media possess a sufficient number of dimensions of variation (or degrees of freedom) that can be appropriately accessed and manipulated and that the components of the mechanism are functionally organized in the appropriate way. (By contrast, many processes, such as cooking, cleaning, and exploding, are defined in terms of specific physical alterations of specific substances. They are not medium independent; hence, they are not computations.)

Given this account of computing, feedback control (3) does not entail computing (5); a system may exert feedback control functions without ever transducing the signals into an internal medium solely devoted to computing—that is, without any medium-independent processing. For instance, a Watt governor exerts feedback control by exploiting specific physical relations that obtain between its variables and the variables it controls.

2.2. Digital computation

The best-known kind of computation is digital computation, which is mathematically characterized by the classical theory of computation begun by Alan Turing (1936–7) and other logicians. Classical computability theory is a well-established branch of mathematics that lies at the foundation of computer science. Digital computation encompasses a widely used class of formalisms such as recursive functions and Turing machines. It gives us access to the fundamental notion of *universal computer*: a digital computer that can compute any digital computation that can be spelled out by an algorithm (Davis, Sigal, & Weyuker, 1994).

Digital computation may be defined both abstractly and concretely. Roughly speaking, abstract digital computation is the manipulation of strings of discrete elements, that is, strings of letters from a finite alphabet. Here, we are interested primarily in concrete computation or physical computation. Letters from a finite alphabet may be physically implemented by what we call ''digits.'' To a first approximation, concrete digital computation is the processing of sequences of digits according to general rules defined over the digits (Piccinini, 2007a). Let us briefly consider the main ingredients of digital computation.

The atomic vehicles of concrete digital computation are digits, where a digit is a macroscopic state (of a component of the system) whose type can be reliably and unambiguously distinguished by the system from other macroscopic types. To each (macroscopic) digit type, there correspond a large number of possible microscopic states. For instance, a huge number of distinct arrangements of electrons (microscopic states) correspond to the same charge stored in a capacitor (macroscopic state). Artificial digital systems are engineered so as to treat all those microscopic states in one way—the way that corresponds to their (macroscopic) digit type. To ensure reliable manipulation of digits based on their type, a physical system must manipulate at most a finite number of digit types. Digits need not mean or represent anything, but they can. For instance, numerals represent numbers, while other digits may not represent anything in particular.

Digits can be concatenated (i.e., ordered) to form sequences or *strings*, which serve as the vehicles of digital computations. A digital computation consists in the processing of strings of digits in accordance with a rule, which is simply a map from input strings, plus possibly internal states, to output strings. Examples of rules that may figure in a digital computation include addition, multiplication, identity, and sorting.

Digits are unambiguously distinguishable by the processing mechanism under normal operating conditions. Strings of digits are sequences of digits, that is, digits such that the system can distinguish different members of the set depending on where they lie along the string. The rules defining digital computations are, in turn, defined in terms of strings of digits and internal states of the system, which are simply states that the system can distinguish from one another. No further physical properties of a physical medium are relevant to whether they implement digital computations. Thus, digital computations can be implemented by any physical medium with the right degrees of freedom.

The notion of digital computation here defined is more general than three other commonly invoked but more restrictive notions of computation: classical computation in the sense of Fodor and Pylyshyn (1988), algorithmic computation, and computation of Turing-computable functions (see Fig. 1; Piccinini & Scarantino, 2011).

Briefly, classical computation is digital computation defined over language-like vehicles. Algorithmic computation is digital computation that follows an algorithm, regardless of whether the vehicles are language-like. Computation of Turing-computable functions is computation that may be carried out by a Turing machine, regardless of whether the process that actually computes it follows an algorithm (e.g., the internal process may be irreducibly continuous and hence not analyzable into discrete algorithmic steps).[7] Finally, digital computation may be Turing-computable or not, although it is doubtful that any physical computation can go beyond Turing-computability (Piccinini, 2011a).

Many other distinctions may be drawn within digital computation, such as hardwired versus programmable, special purpose versus general purpose, and serial versus parallel (cf. Piccinini, 2008). These distinctions play an important role in debates about the computational architecture of cognitive systems.

To summarize, a physical system is a digital computing system if and only if it is a system that is functionally organized to manipulate input strings of digits, depending on the digits' type and their location on the string, in accordance with a rule defined over the strings (and possibly certain internal states of the system).

For present purposes, two points about strings of digits are especially important: (a) whether a particular microscopic state belongs to a digit type is unambiguous relative to the behavior of the system; and (b) the output of a computation depends (either deterministically or probabilistically) only on the internal state of the system and on the number of input digits, their types, and the way they are concatenated within the string during a given time interval.

Systems that qualify as performing digital computations in this sense include physical implementations of standard computability theory formalisms such as Turing machines and finite state automata, ordinary computers and calculators, and physical implementations of neural networks that are characterized in terms of the processing of strings of discrete inputs and outputs. The latter category includes systems that are digital at every processing step,

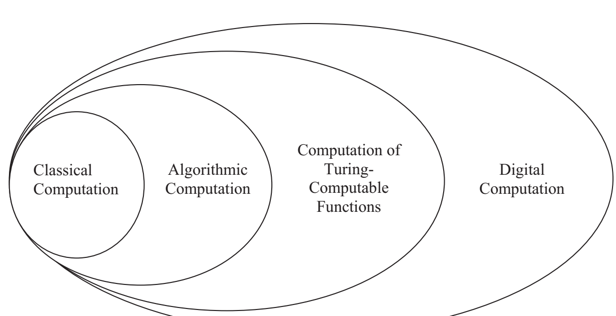

Fig. 1. Types of digital computation and their relations of class inclusion.

such as McCulloch–Pitts nets (McCulloch & Pitts, 1943) and perceptrons (Minsky & Papert, 1969). These are networks whose units take only a finite number of discrete states and update their state during fixed time intervals, so that during any time interval, each layer of the network may be characterized as a string of digits. Other neural networks are digital in their inputs and outputs but not in their internal dynamics. For instance, some Hopfield nets (Hopfield, 1982) and Parallel Distributed Processing networks (Rumelhart & McClelland, 1986), among others, are defined mathematically so that the activation patterns of their input and output layers may be characterized as strings of digits, but their internal dynamics are irreducibly continuous and hence may only be characterized in terms of systems of differential equations defined over continuous variables.

As digital computation is the notion that inspired the computational theory of cognition, it is the most relevant notion for present purposes.

2.3. Analog computation

Analog computation is often contrasted with digital computation, although it is a vaguer and more slippery concept. The clearest notion of analog computation is that of Pour-El, who provided a mathematical theory of the general-purpose analog computer (1974; see also Rubel, 1993; Mills, 2008). Roughly, abstract analog computers are systems whose function is to manipulate continuous variables—variables that can vary continuously over time and take any real values within certain intervals—specified by differential equations, so as to instantiate appropriate functional relationships between the variables. A major function of analog computers is solving certain systems of differential equations by instantiating the relations between continuous variables specified in the equations.[8]

Physically implemented continuous variables are quite different vehicles than strings of digits. While a digital computing system can always unambiguously distinguish digits from one another, a physically implemented analog computing system cannot do the same with the exact values of continuous variables, because these variables can only be measured within a margin of error. Primarily due to this, analog computations (in the present, strict sense) are a different kind of process than digital computations. Nevertheless, like digital computers, analog computers operate on medium-independent vehicles.

There are other kinds of computation besides digital and analog computation. One increasingly popular kind, quantum computation, is similar to digital computation except that digits are replaced with quantum states called qudits (most commonly, binary qudits, which are known as qubits). There is no room here for a complete survey of kinds of computation. Suffice it to conclude that generic computation includes digital computation, analog computation, quantum computation, and more (Fig. 2).

3. Two arguments for generic computationalism

We are now ready to give two arguments for generic computationalism—the thesis that the neural processes that explain cognition are computations in the generic sense.

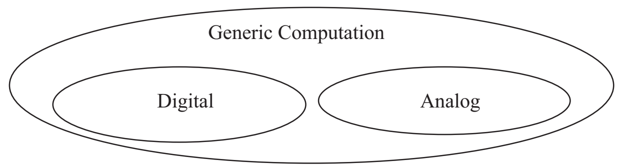

Fig. 2. Types of computation and their relations of class inclusion.

The argument from the functional organization of the nervous system

1. The neural processes that explain cognition are manipulations of medium-independent vehicles.
2. Manipulations of medium-independent vehicles are computations in the generic sense (as per Section 2.1).
3. Therefore, the neural processes that explain cognition are computations in the generic sense.

The vehicles manipulated by neural processes include voltage changes in the dendrites, neuronal spikes (action potentials), neurotransmitters, and hormones. Spike trains are the primary means by which neurons send long-distance signals to one another as well as to muscle fibers. Therefore, we will focus primarily on spike trains. Current evidence suggests that the functionally relevant aspects of neural processes depend on dynamical aspects of the vehicles—most relevantly, spike rates and spike timing. Early studies of neural signaling (Adrian, 1928; Adrian & Zotterman, 1926) demonstrated that nervous systems transduce different physical stimuli into a common internal medium. Spikes follow the *all-or-none* principle: Neurons transmit a signal of a certain amplitude—a spike—above a certain threshold of input intensity, whereas below that threshold no signal is transmitted. Information about the stimulus can be conveyed through changes in spike rates (*rate coding*). While it has been suggested that information is primarily encoded in spike rates, much recent evidence suggests that precise spike timing can also play a role in information transmission (*temporal coding*). Indeed, spike timing may be critically important when stimuli must be processed more quickly than would be possible if the brain relied on spike rates (VanRullen, Guyonneau, & Thorpe, 2005). Examples of temporal coding can be found in the visual (Bair & Koch, 1996; Gollisch, 2009; Meister & Berry, 1999) and somatosensory (Johansson & Birznieks, 2004) systems. The relative importance and prevalence of temporal versus rate coding remains an issue of intense debate (London, Roth, Beeren, Häusser, & Latham, 2010).

The functionally relevant aspects of spike trains, such as spike rates and spike timing, are similar throughout the nervous system regardless of the physical properties of the stimuli (i.e., auditory, visual, and somatosensory) and may be implemented either by neural tissue or by some other physical medium, such as a silicon-based circuit (cf. Craver, 2010). Thus, spike trains—sequences of spikes such as those produced by neurons in real time—appear to be medium-independent vehicles, thereby qualifying as proper vehicles for generic computation. Analogous considerations apply to other vehicles manipulated by neurons, such as voltage changes in dendrites, neurotransmitters, and hormones. Assuming that brains process medium-independent vehicles, it follows that brains perform computations in the generic sense.

The argument from semantic information processing

1. Some neural processes that explain cognition process semantic information.
2. Anything that processes semantic information is a computation (in the generic sense).
3. Therefore, some neural processes that explain cognition are computations (in the generic sense).

The argument for generic computationalism from semantic information processing begins with the observation made in Section 1 that cognition involves the processing of information, at least in three important senses. First, cognition involves processing stochastic signals; this is information in the non-semantic sense of mutual information between a source and a receiver. Second, cognition involves the processing of signals based on their causal correlations with their sources (Adrian, 1928; Rieke et al., 1999); we called this natural semantic information. Third, at least some forms of cognition, such as language processing, appear to require manipulating full-blown representations, that is, internal states that can represent either correctly or incorrectly; we called this non-natural semantic information. As natural cognition is carried out by neural processes, those neural processes that are involved in cognition process semantic information.

For those who identify computation and information processing, the fact that cognition involves information processing is enough to conclude that cognition involves computation. But, as we pointed out in Section 1, computation and information processing are conceptually distinct (Piccinini & Scarantino, 2011). Nevertheless, the fact that cognition involves information processing does entail that cognition involves computation. This is because information (in the three above-mentioned senses of the term) is a medium-independent notion.

Processing information means processing vehicles on the basis of the information they carry rather than their specific physical properties. And the information carried by a signal is defined independently of the physical medium that implements the signal. Thus, processing information requires processing vehicles in accordance with rules that are defined in a medium-independent way. Thus, per Section 2.1, information processing entails generic computation. (The converse does not hold because medium-independent vehicles need not carry any information.)

We conclude that cognition involves computation in the generic sense and nervous systems perform computations in the generic sense. As we shall see, it is much harder to establish that cognition involves a more specific kind of computation.

4. Traditional objections to computationalism

Before discussing digital and analog versions of computationalism, we need to briefly see why computationalism is not refuted by existing objections. There are two types of objection: (a) insufficiency objections, based on alleged differences between computations and cognitive processes; and (b) objections from neural realization, based on alleged differences between computations and neural processes (Piccinini, 2010).

4.1. Insufficiency objections

There is a long tradition of arguments to the effect that computation is insufficient for genuine cognition. These are the most discussed objections to computationalism. Their general form is as follows: Cognitive phenomena include *X*, but computation is insufficient for *X*. Therefore, cognition is not computation. Candidates for *X* include, but are not limited to, the human ability to grasp mathematical truth (Lucas, 1961; Penrose, 1994), conscious experience (Block, 1978), the human ability to represent or understand (Searle, 1980), the human ability to perform explanatory inferences such as abduction (Fodor, 2000), embodiment or the role played by the body in cognition (Thompson, 2007), and embeddedness or the role played by the environment in cognition (van Gelder, 1998).

Each insufficiency objection has met resistance. Whether computation is sufficient for any proposed *X* remains controversial. But the most relevant observation for present purposes is that insufficiency objections do not refute computationalism because they are at best partial: Even if it is granted that computation is insufficient for *X*, it does not follow that neural processes are not computations or that computation plays no role in explaining cognition. All that follows is that explaining *X* requires something else *in addition to computation*. For all that insufficiency objections establish, computation may well be sufficient for cognitive phenomena other than *X*, part of the explanation for *X*, or both.

4.2. Objections from neural realization

There are also objections to computationalism based on alleged differences between neural processes and computations.

4.2.1. Non-electrical processes

Similarly to signals in electronic computers, neural signals include the propagation of voltage changes through axons. But neural signals also include the diffusion of chemical substances such as neurotransmitters and hormones, which mediate or modulate the electrical signals. Therefore, according to this objection, neural processes are not computations (cf. Perkel, 1990).

While this objection correctly points out a disanalogy between nervous systems and electronic computers, the objection does not impugn computation in general. Computation may be realized using chemical signals, as it is a medium-independent process.

4.2.2. Temporal constraints

Neural processes are temporally constrained, whereas computations are not; hence, neural processes are not computations (cf. van Gelder, 1998; Globus, 1992). This objection is based on the observation that *algorithmic digital* computations can be defined and individuated in terms of computational steps, independently of how much time it takes to complete a step.

This abstraction from temporal constraints is a feature not of computations themselves, but of (some of) their descriptions. The same abstraction can be performed on any dynamical process, whether computational or not. When a temporal scale is specified, computational descriptions must take temporal constraints into account just as much as any other descriptions. In fact, even defenders of classical computationalism have been concerned with temporal constraints on the computations that, according to their theory, explain cognitive phenomena (Newell, 1990; Pylyshyn, 1984; Vera & Simon, 1993). And temporal constraints are a main concern, of course, in the neurocomputational literature—especially on motor control (VanRullen et al., 2005).

4.2.3. Spontaneous activity

Neural processes are not the effect of inputs alone because they also include a large amount of spontaneous activity; hence, neural processes are not computations (cf. Perkel, 1990). The obvious problem with this objection is that computations may certainly be the effect of inputs *together with* internal states. In fact, only relatively simple computations are the direct effect of inputs; in the more interesting cases, internal states also play a role.

4.2.4. Analog versus digital

Neural processes are analog, whereas computations are digital; hence, neural processes are not computations (e.g., Dreyfus, 1979; Perkel, 1990). This is one of the oldest and most repeated objections from neural realization.

This objection presupposes the mistaken assumption that all computation is digital. (There is also a problem with the assumption that neural processes are analog, more on that presently.) While digital computation appears to be the most versatile type of computation, and while analogies between neural processes and digital computations were the original source of computationalism (McCulloch & Pitts, 1943), the notion of computation is broader than that of digital computation. This observation leads directly to analog computationalism: The view that neural processes are analog computations.

So far, we have defended a generic version of computationalism and replied to some common objections. In the rest of this article, we will argue that neural computation is neither analog nor digital; it is sui generis.

5. Why neural processes are not analog computations

Some authors have argued that neural processes are analog computations. In fact, the original proponents of the analog versus digital objection did not offer it as a refutation of computationalism simpliciter, but only of McCulloch and Pitts's *digital* computationalism (Gerard, 1951).

But analog computationalism does not immediately follow from the claim that neural processes are analog. Depending on what is meant by ''analog,'' an analog process may or may not be an analog computation.

In a broad sense, ''analog'' refers to processes that can be characterized as the dynamical evolution of real (i.e., continuous or analog) variables in real time. Some authors who discuss whether the nervous system is analog seem to employ this broad notion (e.g., Churchland & Sejnowski, 1992; van Gelder, 1998). Surely neural processes are analog in this sense, but so are all or most physical processes according to current physical theory.[9] Yet most systems that are analog in this sense, such as the weather, planetary motion, or digestion, are not analog computers in any interesting sense. Even more important, given a sufficiently fine-grained description, ordinary digital computers are analog in this sense, too! Thus, the trivial fact that neural mechanisms are analog in this broad sense does nothing to either refute computationalism or establish an analog version of it.

In another sense, ''analog'' refers to representations that bear some positive analogy to what they represent. There is evidence that nervous systems contain and manipulate analog models of their environment (Grush, 2004; Waskan, 2003). Examples include visual (Hubel & Wiesel, 1962) and auditory (Schreiner & Winer, 2007) receptive fields, as well as the somatosensory representation of the world touched by a rodent's whiskers in the ''barrel cortex'' (Inan & Crair, 2007). But analog *models* need not be represented and manipulated by analog *computers*.

Finally, in the most relevant sense, ''analog'' refers to analog computers properly so called (Pour-El, 1974; Rubel, 1993; Mills, 2008; and see Section 2.3). Claiming that neural systems are analog computers is a strong empirical hypothesis. Analog computationalism is committed to the view that the functionally significant signals manipulated by the nervous system are irreducibly continuous variables.

To be sure, both brains and analog computers appear to have an essentially continuous dynamic, that is, their vehicles must be assumed to vary over real time in order to construct successful scientific theories of them (for the case of brains, see Dayan & Abbott, 2001; Ermentrout & Terman, 2010). Neuronal inputs—that is, neurotransmitters and hormones—are most usefully modeled as continuous variables; in addition, their release and uptake is modulated by chemical receptors that operate continuously in real time. Also, dendrites and *some* axons transmit graded potentials—that is, more or less continuously varying voltage changes. In these respects, brains appear to be more similar to analog computers than to digital ones. Finally, as Rubel (1985) points out, neural integrators do play an important role in at least some processes, such as oculomotor control. (Notice that integrators are the most important components of classical, general-purpose analog computers.)

However, there are crucial disanalogies between brains and analog computers. To begin with, while the firing threshold is subject to modulation by continuous variables (e.g., ion concentrations), graded potentials vary continuously, and spike timing (in real time) may be functionally significant, none of these aspects of neural signals are similar enough to the processes used by traditional artificial analog computers for the mathematics of analog computers to be directly applied in understanding brain processes. New mathematical theory—specific to neural processes—had to be invented for this purpose.

Even more important, spikes are the most functionally significant signals transmitted by neurons, and what is functionally significant is not the absolute value of the voltage or the exact shape of the voltage curve but simply whether a spike is present or absent. If the total input signal received by a neuron reaches a certain threshold, the neuron fires a spike. If the total signal received by a neuron is below the threshold, no spike is fired. As a consequence of the all-or-none character of the spike, computational neuroscience focuses on firing rates and spike timing as the most significant functional variables at the level of circuit and network. There is nothing similar to firing rates and spike timing that plays a similar role in analog computers (in the present strict sense).

Finally, neural integrators are specialized components that operate on spike trains (as opposed to continuous variables), which means they operate on a different type of vehicle than the integrators of analog computers. Thus, in spite of some similarities, neural systems are not analog computers in the strict sense.

6. How spike trains *might have been* strings of digits

The mathematical modeling of neural processes can be traced back to the mathematical biophysics pioneered by Nicolas Rashevsky and his associates (Householder & Landahl, 1945; Rashevsky, 1938). They employed integral and differential equations to describe and analyze processes that might explain neural and cognitive phenomena. In their explanations, they used neither the notion of computation nor the mathematical tools of computability theory and computer science. Those tools were introduced into theoretical neuroscience by the youngest member of Rashevsky's group, Walter Pitts, in collaboration with Warren McCulloch.

Like Rashevsky's mathematical biophysicists, McCulloch and Pitts (1943) attempted to explain cognitive phenomena in terms of putative neural processes. However, McCulloch and Pitts introduced concepts and techniques from logic and computability theory to model what is functionally relevant to the activity of neurons and neural nets, in such a way that (in our terminology) a neural spike can be treated mathematically as a digit, and a set of spikes can be treated as a string of digits.

McCulloch and Pitts's empirical justification for their theory was the similarity between digits and spikes—spikes *appear* to be discrete or digital, that is, to a first approximation, their occurrence is an unambiguous event relative to the functioning of the system. This was the main motivation behind McCulloch's identification between spikes and atomic mental events, leading to his formulation of digital computationalism (Piccinini, 2004). We will leave aside whether spikes correspond to atomic mental events and limit our discussion to whether spikes are digits and whether sets of spikes are strings of digits.

The assumption that sets of spikes are strings of digits requires that spikes be concatenated. In the case of spike trains from a single neuron, the temporal ordering of spikes is a natural candidate for the concatenation relation. In the case of sets of spikes from different neurons occurring within well-defined time intervals, a concatenation relation might be defined by first identifying a relevant set of neurons and then taking all spikes occurring within the same time interval as belonging to the same string. The set of neurons might be defined purely anatomically or by a combination of anatomical and functional criteria. For example, one could try to identify, electrophysiologically or histologically, which neurons are presynaptic and postsynaptic to which.

The suggestive analogy between spikes and digits—based on the all-or-none character of spikes—is far from sufficient to treat a spike as a digit, and even less sufficient to treat a set of spikes as a string of digits. In order to build their theory, according to which neural activity is digital computation, McCulloch and Pitts made additional assumptions.

A crucial assumption was that a fixed number of neural stimuli are always necessary and sufficient to generate a neuron's pulse. They knew this is false: They explicitly mentioned that the ''excitability'' of a neuron does vary over time (McCulloch & Pitts, 1943, pp. 20–21). In fact, neurons exhibit both absolute and relative refractory periods as well as varying degrees of plasticity ranging from short- to long-term potentiation and depression. To this, it should be added that spiking is not a deterministic but a probabilistic function of neuronal input. Indeed, the flow of ions through transmembrane channels has a stochastic component (Hille, 2001; White, Klink, Alonso, & Kay, 1998), as does neurotransmitter release (Bennett & Kearns, 2000; Hille, 2001). Even when neurons fire with almost pacemaker-like regularity, their natural frequencies retain some variability (Le Bon-Jego & Yuste 2007; Yassin et al., 2010) due to natural fluctuations in physiological parameters which, in turn, affect the time at which the transmembrane potential reaches the firing threshold.

Another crucial assumption was that all neurons within a network are synchronized so that all relevant events—conduction of the impulses along nerve fibers, refractory periods, and synaptic delays—occur within temporal intervals of fixed and uniform length, which are equal to the time of synaptic delay. In addition, according to McCulloch and Pitts, all events within one temporal interval only affect the relevant events within the following temporal interval. This assumption makes the *dynamics* of the net discrete: It allows the use of logical functions of discrete inputs or states to fully describe the transition between neural events.

As McCulloch and Pitts pointed out, these assumptions are not based on empirical evidence. The reason for these assumptions was, presumably, that they allowed the mathematics of digital computation to describe, at least to a first approximation, what they presumed to be most functionally relevant to neural activity.

The similarity between digits and spikes can hardly be doubted. However, there are significant dissimilarities as well, and there are further dissimilarities between spike sets—either spike trains from the same neuron or sets of more or less synchronous spikes from different neurons—and strings of digits. In 1943, relatively little was empirically known about the functional properties of neural spikes and spike trains. Since then, neurophysiology has made great progress, and it is past time to bring what is currently known about neural processes to bear on the question of whether they are digital computations.

Although McCulloch and Pitts's theory became fundamental to computer science and computability theory (Piccinini, 2004), as a theory of neural activity, it was soon abandoned in favor of more sophisticated and neurally plausible mathematical models. These include the following, in order of decreasing biological detail and increasing computational tractability: conductance-based models, which go back to Hogdkin and Huxley's (1952) seminal analysis of the action potential based on conductance changes; networks of integrate-and-fire neurons, which fire simply when the input current reaches a certain threshold (Caianiello, 1961; Knight, 1972; Lapicque, 2007; Stein, 1965); and firing rate models, in which there are no individual action potentials—instead, the continuous output of each network unit represents the firing rate of a neuron or neuronal population (Wilson & Cowan, 1972).

From a mathematical point of view, the models employed by theoretical neuroscientists after McCulloch and Pitts resemble those of Rashevsky's original mathematical biophysics, in that they do not employ concepts and techniques from computability theory. Recall that by ''computability theory'' we mean the theory of recursive functions and their computation (Davis et al., 1994).[10] In the models currently employed by theoretical neuroscientists, spikes are not treated as digits, and spike sets are not treated as strings of digits. The rest of this article identifies principled empirical reasons for why this is so.

The point of the exercise is not to beat a long-dead horse—McCulloch and Pitts's theory. The point is that if we reject McCulloch and Pitts's assumptions, which were the original motivation for calling neural activity digital computation, it remains to be seen whether there is any way to replace McCulloch and Pitts's assumptions with assumptions that are both empirically plausible and allow us to retain the view that neural activity is digital computation in the relevant sense. If no such assumptions are forthcoming, digital computationalism needs to be either abandoned or grounded in (nondigital) neural computation.

7. Why spikes are not digits and spike sets are not strings

We will now argue that current neuroscientific evidence suggests that typical neural processes are not digital computations. The structure of the argument will be as follows:

1. Digital computation requires the manipulation of strings of digits (as per Section 2.2);
2. Neural spikes are not digits, and even if they were digits, spike sets would not be strings;
3. Therefore, the manipulation of spike sets is not digital computation.

Our argument challenges not only classical computationalism, which is explicitly committed to digital computation, but also any form of connectionist or neurocomputational theory that is either explicitly or implicitly committed to the thesis that neural activity is digital computation.[11]

Caveat 1: A defender of classicism might argue that while classicism is committed to digital computation at the ''cognitive'' level, it is not committed to digital computation occurring at the ''implementation'' level we are considering here. It may be that non-digital neural computations give rise to digital computations at the ''cognitive'' level. This speculative proposal faces two problems. First, biological cognitive systems—that is, nervous systems—cannot be analyzed into two levels, one of which is cognitive and one of which is implementational (cf. Lycan, 1990). Nervous systems contain many mechanistic levels (Craver, 2007), which may be more or less directly involved in explaining cognitive phenomena. The activities of neurons and networks thereof are routinely invoked by neuroscientists to explain cognitive phenomena. Thus, the levels of neurons and networks are as cognitive as any higher level. Second, the idea that higher level digital computation emerges from lower level non-digital computation requires both an explicit account of how this occurs and some empirical support, both of which are lacking to date. We will come back to this at the end.[12]

Caveat 2: The present argument is not formulated in terms of the notion of representation. It rests on the notion of digits as discrete entities that can be concatenated into strings. The digits discussed in this articles may be vehicles for many types of representation. Our argument does not depend on which if any notion of representation is adopted. Specifically, it is indifferent to the distinctions between digital versus analog (Maley, 2011), localized versus distributed, and ''symbolic'' versus ''subsymbolic'' representation (Rumelhart & McClelland, 1986; Smolensky, 1989).[13] In fact, our argument is consistent both with representationalism and its denial. The truth value of representationalism is simply not affected by our argument.

For a class of particular events to constitute digits within a computing mechanism, it must be possible to type them in an appropriate way. In the next subsection, we discuss the difficulties in doing so for spikes. For a set of digits to constitute a string, it must be possible to determine, at the very least, which digits belong to which strings. Later, we will discuss the difficulties in doing so for sets of spikes.

7.1. Typing spikes as digits

Digits belong to finitely many types, which must be unambiguously distinguishable by the system that manipulates them. So for spikes and their absence to be digits, it must be possible to type them into finitely many types that are unambiguously distinguishable by neural mechanisms.

Spikes are all-or-none events; either they occur or not. So there might be two types of digits: the *presence* or *absence* of a spike at some time. This was McCulloch and Pitts's proposal. For this proposal to be adequate, more needs to be said about the time during which the presence or absence of spikes is to be counted.

The relevant time cannot be instantaneous. Otherwise, there would be uncountably many digits in any finite amount of time, which is incompatible with the notion of digit. Besides that, spikes do take some time (about 1 ms) to occur. McCulloch and Pitts's proposal was to divide time into intervals whose length was equal to synaptic delay. Their proposal suffers from a number of shortcomings. First, it assumes that synaptic delay is constant; recent studies show it to be variable (e.g., Lin & Faber, 2002). Second, any time interval separating two spikes is assumed be equal to an exact multiple of the master time interval, whereas there is no empirical evidence of any such master time interval. Third, to count the presence or absence of spikes from different neurons as belonging to the same computation, it assumes that neurons are perfectly synchronized, in the sense that their relevant time intervals begin and end in unison. The third assumption is also relevant to the concatenation of digits into strings, so we will discuss it in subsequent subsections. In what follows, we will focus on the first two assumptions.

The lack of evidence for the first two assumptions could be remedied by choosing a fixed time interval of unambiguous physiological significance. But, even though spikes may occur fairly reliably within 1 ms of neural stimuli in vitro (Mainen & Sejnowski, 1995), actual neural firing in vivo (and in many in vitro preparations as well) often occurs with high variability, due to a large number of factors in the fluctuating cellular environment, which includes thousands of neuronal connections and dozens of different cell types. Thus, spikes are often *highly stochastic* functions of a large number of highly variable inputs. While some neural firing can be quite temporally precise, it cannot be claimed that spikes occur only within fixed time intervals of physiological significance. To summarize, since spiking is affected in varying degrees by events (such as input spikes) that occur at different times in the neuron's history (e.g., Shepherd, 1999, p. 126), there is no known principled way of typing both relevant causes and the time intervals during which they occur into finitely many types.

By itself, that spikes are probabilistic need not undermine the analogy between spikes and digits, as digital computations may be probabilistic as well as deterministic. In a probabilistic digital computation, there are a finite number of possible outcomes, each of which has a certain probability of occurring. In other words, the sample space of a probabilistic digital computation is finite. This seems superficially analogous to the neural spiking, which occurs with a certain probability at a time (given a certain input).

But there is a crucial disanalogy. Whereas digital computational events are temporally discrete, spike times are values of continuous variables. In the case of a probabilistic digital computation, events occur within fixed functionally significant time intervals; this makes it possible to specify the probabilities of the finitely many outcomes effectively, outcome by outcome, so that each constitutes a digit. In contrast, there appear to be no functionally significant time intervals during which spikes are to be counted, and as a result, spikes must be counted over real time. Because of this, the probability that a spike occurs at any instant of real time is zero. In other words, the sample space for spiking events during a time interval is uncountably infinite. Spike probabilities need to be given by probability densities, which specify the probability that a spike occurs within an infinitesimal time interval (Dayan & Abbott, 2001, p. 24). In conclusion, as there can only be finitely many digit types but there are an uncountable number of possible spiking events, spiking events cannot be digits.

There is another independent and perhaps deeper reason for denying that the presence or absence of a spike can constitute a digit. Until now, the discussion has been based on the implicit assumption that the physiological significance of the presence or absence of *individual* spikes is functionally relevant to the processing of neural signals. But this assumption is unwarranted in many cases. Although the presence or absence of spikes does affect neural systems differentially, individual spikes rarely seem to have unambiguous functional significance for the processing of neural signals (though see VanRullen et al., 2005).

First, neurons and neural systems can exhibit spontaneous activity, not triggered by any sensory stimulus; indeed, spontaneous neural activity—including postsynaptic activity (cf. Alle, Roth, & Geiger, 2009)—has been suggested to account for at least 75% of the brain's energy consumption (Raichle & Mintun, 2006). (This spontaneous activity need not be stochastic: Spontaneous firing of neocortical neurons in cortical microcircuits has been shown to have regular, pacemaker-like activity [Le Bon-Jego & Yuste, 2007], and pacemaker cells in the suprachiasmatic nuclei can fire regularly even in the absence of any neuronal input.) Some have suggested that neural signals consist in relatively small deviations from a background level of activity. This is what makes the hypothesis of a ''default network,'' suggested by BOLD fMRI imaging studies, so compelling (Raichle et al., 2001; Zhang & Raichle, 2010), although still controversial (Morcom & Fletcher, 2007). Finally, even a noisy, stochastic spontaneous firing state can have some function; witness the role of noise in enhancing weak subthreshold signals in the nervous system via stochastic resonance (Douglass, Wilkens, Pantazelou, & Moss, 1993; Levin & Miller, 1996; Moss, Ward, & Sannita, 2004).

Functional, noisy, or neither, spontaneous activity per se is not an objection to digital computationalism. For digital computations can depend on both inputs and internal states. In principle, spontaneous activity could be analogous to the internal state of a digital computing mechanism. But the internal states of (fully) digital computing mechanisms are such that their contribution to a computation can be described as digits (discrete states) and specified independently of the contribution given by the input. In contrast, spontaneous neural activity has all the characteristics already noted for spikes in general, which led to the conclusion that spikes are not digits.

Second, regardless of whether a spike is caused by spontaneous activity, a sensory input, or both, what appears to have functional significance is often not any individual spike, but rather the firing *rate* of a neuron. Consider the simple case of the crayfish photoreceptor, which fires at a rate of about 5 Hz in the dark and speeds up to 30 Hz when exposed to visible light (Bruno & Kennedy, 1962). In such clear cases of rate coding, individual spikes have no functional significance on their own. For that matter, many individual spikes might in principle be added to or removed from a spike train without appreciably altering its functional significance (Dayan & Abbott, 2001; Chapter 1). In fact, neuroscientists typically assess the functional significance of neural activity by computing average firing rates, namely by averaging the firing rates exhibited by a neuron over many similar trials.[14]

Thus, there are principled reasons to doubt that spikes and their absence can be treated mathematically as digits, to be concatenated into strings. But individual spikes are not the only candidates to the role of digits. Before abandoning the issue of digits, we should briefly look at two other proposals.

One possibility is that digits are to be found in precise patterns of several spikes. There have been serious attempts to demonstrate that some repeating triplets or quadruplets of precisely timed ($\approx$1 ms) spikes from single neurons have functional significance. However, these attempts have failed: Rigorous statistical analysis indicates that both the number and types of precisely timed spike patterns are those that should be expected by chance (Oram, Wiener, Lestienne, & Richmond, 1999). In fact, in many cases, signals coming from individual neurons are too noisy to constitute the minimal units of neural processing; according to some estimates, the minimal signaling units in the human brain are of the order of 50–100 neurons (Shadlen & Newsome, 1998).

Another possibility is that spike rates themselves should be taken to be digits of sorts. This was von Neumann's proposal (von Neumann, 1958). Unlike the previous proposal, this one is based on variables that are functionally significant for the processing of neural signals. But this proposal raises more problems than it solves. Again, there is the problem of finding a significant time interval. Spike rates vary continuously, so there seems to be no non-arbitrary way to parse them into time intervals and count the resulting units as digits. It is also unclear how spike trains from individual neurons, understood as digits, could be concatenated with other spike trains from other neurons to form strings. Perhaps most important, spike rates cannot be digits in the standard sense, because they cannot be typed into finitely many types of unambiguous physiological significance. As von Neumann noted, there is a limit to the precision of spike rates, so if we attempt to treat them as digits, there will be a significant margin of error as to which type any putative digit belongs to; the same problem occurs with individual spike timings. As a consequence, computability theory as normally formulated does not apply to spike rates—a different mathematical theory is needed. Von Neumann pointed out that he did not know how to do logic or arithmetic over such entities as spike rates, and hence, he did not know how the brain could perform logic or arithmetic using such a system. As far as we know, no one after von Neumann has addressed this problem. And without a solution to this problem, von Neumann's proposal remains an empty shell.

7.2. Fitting spikes into strings

In the previous section, we saw that there is no reason to believe that spikes, or sets thereof, can be treated as the digits of a computation. But even if they could, it would not follow that neural mechanisms perform computations in any useful sense. For although some trivial computations are defined over single digits or pairs of them (i.e., those performed by ordinary logic gates), nontrivial computations are defined over strings of digits. Strings of digits are entities with finitely many digits concatenated together, in which there is no ambiguity as to which digits belong to a string. If sets of spikes were to count as strings of digits, there should be a way to unambiguously determine which spikes belong to which string. But there is no reason to believe that this can be done. The two natural candidates for strings are sets of spikes from synchronous neurons and spike trains from individual neurons.

7.2.1. Strings and synchronous spikes

In the case of sets of synchronous spikes, the concatenation relation might be given by spatial and temporal contiguity within a structure defined by some combination of anatomical and functional properties. For instance, synchronous spikes from a cortical column may be seen as belonging to the same string. Implicitly, this is how inputs and outputs of many connectionist systems are defined.

A first difficulty with this proposal is the looseness of the boundaries between neural structures. It does not appear that neural structures, such as cortical columns, are defined precisely enough that one can determine, neuron-by-neuron, whether each neuron belongs to one structure or another. The boundaries are fuzzy, so that neurons can be recruited to perform different functions as they are needed, and the functional organization of neural structures is reshaped as a result (Elbert & Rockstroh, 2004; Recanzone, Merzenich, & Schreiner, 1992). This first problem may not be fatal. Perhaps the boundaries between neural structures are fuzzy, and yet at any time, there is a way to assign each spike to one and only one string of spikes synchronous to it.

But there is a fatal problem: Synchrony between spikes is a matter of degree. Just as there is no meaningful way to divide the activity of a neuron into discrete functionally significant time intervals (see Section 7.1), there is no meaningful way to divide the activity of neural ensembles into discrete functionally significant time intervals to determine in an absolute manner whether their spikes are synchronous. Even if such absolute time determinations were possible, the inherent noisiness in neural firing times would result in a varying degree of synchrony over time. Synchrony between spikes has been studied intensely by the nonlinear dynamics community over the past few decades in the context of the synchronization of coupled oscillators (Pikovsky, Rosenblum, & Kurths, 2003); it may play a role in normal brain functions such as attention (Fries, Reynolds, Rorie, & Desimone, 2001; Fries, Womelsdorf, Oostenveld, & Desimone, 2008; Roy, Steinmetz, Hsiao, Johnson, & Niebur, 2007; Steinmetz et al., 2000); it has been applied to the analysis of neural pathologies such as epilepsy (Uhlhaas & Singer, 2006; Wong, Traub, & Miles, 1986) and Parkinson's disease (Tass et al., 2003). But the interesting question about neural synchrony is not *whether*, in an absolute sense, there is synchrony in a neural process, but rather *how much* synchrony there is. The typical definition of synchrony used within the neurophysiological community is that of *stochastic phase synchronization*, which measures the degree to which the phase difference between neurons remains constant; thus, neurons can be synchronized even with a (relatively) constant phase lag (Pikovsky et al., 2003).

As a result, even if neural spikes were digits (which they are not), sets of synchronous spikes would not be strings of digits.

7.2.2. Strings and spike trains

In a spike train from a single neuron, the concatenation relation may be straightforwardly identified with the temporal order of the spikes. Nevertheless, in order to treat a spike train as a string, its beginning (first spike) and end (end spike) must be identified unambiguously. Again, two independent problems undermine this proposal.

The first problem is that the question of which spikes belong to a string appears to be ill defined. One reason for this is the inherent stochasticity present in much neuronal firing, discussed above. While spike timing can be exquisitely precise, much neural activity is subject to various fluctuations, not least the variability in neurotransmitter release at the synaptic terminal. This can result in a noisy background of ''spontaneous'' (i.e., not driven by external sensory input) activity. This can be seen, for example, in the typically broad power spectrum of spiking frequencies from the unstimulated crayfish mechanoreceptor–photoreceptor system. Under external periodic stimuli, the stimulus frequency appears ''encoded'' in the power spectrum, superimposed on the background of variable spontaneous activity (Bahar, 2003; Bahar & Moss, 2003). There does not seem to be any functionally significant way to parse *which* spikes are the ones encoding the stimulus, however, or to divide the neural responses to sensory inputs into subsets with unambiguous beginnings and ends.

The second problem is that given the amount of noisiness in neural activity, individual spike trains are not useful units of functional significance. What has functional significance is not a specific train of spikes, each of which occurs during a particular time interval, but something held in common by many different spike trains that are generated by the same neuron in response to the same sensory stimulus. The only known way to find useful units of functional significance is to average spike trains from single neurons over many trials and use average spike trains as units of functional significance.

For these reasons, even if neural spikes could be treated as digits (which they cannot), spike trains could not be meaningfully treated as strings of digits.

7.3. What about the neural code?

A final relevant issue is the relationship between neural codes and (digital) computational codes. In both cases, the same term is used, which suggests an analogy between the two. Is there such an analogy?

The expression ''neural code'' usually refers to the functional relationship between properties of neural activity and variables external to the nervous system (e.g., see Dayan & Abbott, 2001). Roughly speaking, the functional dependence of some property of a neural response on some property of an environmental stimulus is called neural encoding, whereas the inference from some property of a neural response to some property of the stimulus is called neural decoding. Neural encoding is a measure of how accurately a neural response reflects a stimulus; neural decoding is a measure of how accurately a neural mechanism can respond to a stimulus.

In contrast, in the theory of (digital) computation, ''coding'' refers to one-to-one (sometimes, many-to-one) mappings between classes of strings of digits (e.g., see Rogers, 1967, pp. 27–29). Given each member of a (countably infinite) class of strings over some alphabet, any algorithm that produces a corresponding member of a different class of strings (possibly over a different alphabet) is called an encoding. This kind of encoding is usually reversible: For any encoding algorithm, there is a decoding algorithm, which recovers the inputs of the encoding algorithm from its outputs.

Neural coding is only loosely analogous to computational coding. Both are relationships between two classes of entities, and both give procedures for retrieving one class of entities from the other. But the disanalogies are profound. Whereas neural encoding is only approximate and approximately reversible, (digital) computational coding is (by definition) exact and is usually exactly reversible. Most important, whereas neural encoding relates physical quantities in the environment to neural responses, neither one of which is likely to be accurately described as a class of digital strings, computational coding is a relationship between classes of digital strings. For these reasons, neural coding should not be confused with (digital) computational coding. The existence of a neural code does not entail that neural processes are digital computations.

7.4. Against digital computationalism

Given current evidence, the most functionally significant variables for the purpose of understanding the processing of neural signals are properties of neural spike trains such as firing rate and spike timing. We have argued that neither spike trains from single neurons nor sets of synchronous spikes from multiple neurons are viable candidates for strings of digits. Without strings, we cannot identify operations defined over strings. And without operations defined over strings, we cannot have digital computations.

The discrete nature of spikes was the main empirical motivation for digital computationalism. Once the original assumptions about spikes that supported digital computationalism are rejected, the burden is on supporters of digital computationalism to find evidence that some neural variables are suited to being treated as strings of digits, and that this has something to do with explaining cognitive phenomena.

8. Some consequences

While few neuroscientists take typical neural processes to be digital computations, many psychologists and philosophers are still building their theories of cognition using constructs that rely either implicitly or explicitly on digital computation, and some of them still defend digital computationalism (e.g., Fodor, 2008; Gallistel & King, 2009; Schneider, 2011). We will reiterate that understanding neural computation requires specially designed mathematical tools (Dayan & Abbott, 2001; Ermentrout & Terman, 2010) rather than the mathematics of analog or digital computation. Moreover, there remains considerable confusion in the literature—including the neuroscience literature—about what computationalism does and does not entail. We will examine some widespread fallacies. We will conclude with a plea to psychologists who rely on non-neural notions of computation to replace or reinterpret their theories in terms of neural computation.

8.1. Neural computation is sui generis

Given the present state of the art, a defense of digital computationalism (or analog computationalism, for that matter) as a general account of neural computation is not only implausible—it is unmotivated. Unlike in 1943, when digital computationalism was initially proposed, today we have an advanced science of neural mechanisms and processes, based on sophisticated mathematical concepts and techniques that are, in turn, grounded in sophisticated experimental concepts and techniques. We are referring to what is often called theoretical or computational neuroscience (Dayan & Abbott, 2001; Ermentrout & Terman, 2010). Neurocomputational models can include, for example, spiking outputs from model cells, but also terms that rely on graded changes in parameters and on spike rates rather than on precise spike timing.

Theoretical neuroscientists build mathematical models of neural mechanisms and processes and use them to explain neural and cognitive phenomena. Many philosophers and psychologists have unduly neglected the development of this discipline. Specifically, they have rarely noticed that in realistic mathematical models of neural processes, which can be tested against experiments, the explanatory role of (digital) computability theory and (digital) computer design is nil.

When a process is explained by digital computation, at a minimum the explanation includes a stream of digital inputs, internal states, and outputs, a set of rules for manipulating the digits, and a system of digital circuits that perform the manipulations. This is how computational explanation works in digital computers (Piccinini, 2007a, 2008). But nothing of this sort has been found to explain neural processes. If the present argument is right, none is forthcoming.

In a nutshell, current evidence indicates that typical neural signals, such as spike trains, are graded like continuous signals but are constituted by discrete functional elements (spikes). Therefore, typical neural signals are neither continuous signals nor strings of digits; neural computation is sui generis.

8.2. Avoiding some fallacies

Computationalism properly so called is often conflated with other theses that are weaker than or logically independent of it. At least two issues are relevant here.

The first issue has to do with the relationship between computationalism and representationalism. Many authors believe that computation is the same as information processing, so that all computational systems process information (or, more strongly, process representations; e.g., Churchland, Koch, & Sejnowski, 1990; Fodor, 2008; O'Brien & Opie, 2006; Shagrir, 2010). We argued that processing information does entail computing, but the reverse does not hold. Equating computation with information processing obscures the structural (non-semantic) differences between different types of computing, such as the differences between digital, analog, and neural computation that we have discussed in this article. To understand the nature of neural computation, it is essential to define computation independently of information processing (and, a fortiori, representation) so as to highlight its structural aspects.

The second issue has to do with the relationship between computationalism and mechanistic, or functional, explanation. Computationalism and mechanism have been run together in the literature. For example, Marr (1982) dubbed the mathematical analysis of the functions of neural mechanisms ''computational theory,'' by which he meant theory of neural computations. From this, it follows that neural processes are computations. But thus understood, computationalism ceases to be an empirical hypothesis about the specific functional nature of neural processes, to become a trivial consequence of the definition of mechanistic explanation. Suppose, instead, that we begin our investigation with a rich notion of mechanistic explanation (Craver, 2007). Under such a notion, computation is one kind of mechanistic process among others (Section 2; also Piccinini, 2007a; Piccinini & Scarantino, 2011). Then, we may ask—with McCulloch and Pitts and other computationalists—whether neural processes are computations and in which sense. As we argued, we can find empirical evidence that neural processes are computations (in the generic sense) and that neural computation is sui generis.

Computationalism is often used to derive various consequences, which may or may not follow from digital computationalism but certainly do not follow from generic computationalism: (a) that in principle we can capture what is functionally relevant to neural processes in terms of some formalism taken from (digital) computability theory (such as Turing machines); (b) that it is possible to design computer programs that are functionally equivalent to neural processes in the sense in which computer programs can be computationally equivalent to each other; (c) that the study of neural (or mental) computation is independent of the study of neural implementation; and (d) that the Church-Turing thesis applies to neural activity in the sense in which it applies to digital computers. None of these alleged consequences of computationalism follow from the view that neural activity is computation in the generic sense.

If neural activity is not digital computation, as we have argued, then there is no reason to suppose that formalisms that manipulate strings of digits can capture what is functionally relevant to neural processes. They might be used to build computational *models* of neural processes, but the explanation of neural processes will be given by the constructs of theoretical neuroscience, using ''computational models'' in the sense of computer programs which perform numerical integration of coupled nonlinear ordinary or partial differential equations to simulate neural activity, not the formalisms of computability theory.

The relationship between computer programs that model neural processes and the processes they simulate is significantly different from the relationship between computer programs that simulate each other. When a computer program simulates another program, (some aspect of) the computations of the simulating program encode (in the sense of computer science) the computations of the simulated program. In contrast, when a computer program implements a mathematical model of a neural process, the computations of the simulating program are a means for generating approximate representations of the states that the mathematical model attributes to the neural process. This kind of model attempts to capture something essential to a neural process based on what is known about the process by employing suitable abstractions and idealizations. There is no sense in which such modeling work can proceed independently of the empirical study of neural processes. There is no room here for an in-depth treatment of the methodology of computational modeling (Craver, 2006; Humphreys, 2004; Piccinini, 2007b; Winsberg, 2010). For present purposes, what matters is that computational modeling is a different activity, with different rules, from the simulation of one computer program by another program.

Finally, the Church-Turing thesis is a very complex issue, and there is no room here to do it justice. For present purposes, the following brief remark will suffice. To a first approximation, the Church-Turing thesis states that any function that is computable in an intuitive sense is computable by some Turing machine. Many authors have argued that the Church-Turing thesis proves computationalism to be true (e.g., Churchland, 2007). But as recent scholarship has shown, this argument commits the fallacy of conflating mechanism and a form of computationalism restricted to the computations that fall under the Church-Turing thesis (Copeland, 2000; Piccinini, 2007c).

When mechanism and computationalism are kept distinct, as they should be, two separate questions about neural computation arise. One question—the most discussed in the literature—is whether neural computations are computable by Turing machines. If McCulloch and Pitts's theory of the brain (or a similar theory) were correct, then the answer would be positive. Another possibility is that neural computations are digital but more powerful than those of Turing machines—neural mechanisms might be hypercomputers (Copeland, 2000; Siegelmann, 2003).

The more fundamental question, however, is whether neural activity is digital computation in the first place. If, as we have argued here, neural activity is *not* digital computation, then the question of whether neural computation is computable by Turing machines does not even arise. Of course, there remains the question of whether computing humans can do more than Turing machines. Human beings can certainly perform digital computations, as anyone who has learned basic arithmetic can attest. And presumably there are neural processes that explain human digital computing. However, there is no evidence that human beings are computationally more powerful than Turing machines. No human being has ever been able to compute functions, such as the halting function, that are not computable by Turing machines. In this limited sense, the Church-Turing thesis appears to be true.

8.3. From cognitive science to cognitive neuroscience

If neural computation is sui generis, what happens to computational theories in psychology, many of which rely on digital computation? Mental models (e.g., Johnson-Laird, 1983, 2010), production systems (e.g., Newell, 1990), and ''theories'' (e.g., Murphy, 2002) are examples of computational constructs employed by psychologists to explain cognitive phenomena. For over 60 years, various forms of *digital* computationalism—either classical or connectionist—have provided the background assumption against which many computational constructs in psychology have been interpreted and justified. Computational psychological theories are legitimate, the assumption goes, because the digital computations they postulate can be realized by neural computations. But if neural computations are not digital (or if digital neural computations are an exception rather than a rule), what should we do with computational psychological theories that rely on digital computation?

One tempting response is that psychological explanations remain unaffected because they are *autonomous* from neuroscientific explanations. According to a traditional view, psychological theories are not directly constrained by neural structures; psychology offers functional analyses, not mechanistic explanations like neuroscience (Cummins, 1983; Fodor, 1968). But on closer examination, functional analysis turns out to be an elliptical or partial mechanistic explanation (Piccinini & Craver, 2011). Thus, psychological explanation is a type of mechanistic explanation after all. Psychological explanation is not autonomous from neuroscientific explanation; instead, psychological explanation is directly constrained by neural structures.

Another tempting response is to eliminate psychological explanations. That would be premature. Computational theories in psychology are often our current best way of capturing and explaining psychological phenomena.

What needs to be done, instead, is to take psychological constructs that rely on digital computation and gradually reinterpret or replace them with theoretical constructs that can be realized by known neural processes, such as the spike trains of neuronal ensembles. The shift that psychology is currently undergoing, from classical cognitive psychology to cognitive neuroscience, goes in this direction. Much groundbreaking work in psychology over the past two decades may be seen as contributing to this project (e.g., Anderson, 2007; Barsalou, 1999; Gazzaniga, 2009; Kalat, 2008; Kosslyn, Thompson, & Ganis, 2006; O'Reilly & Munakata, 2000; Posner, 2004). We are on our way to explaining more and more neural and cognitive phenomena in terms of neural mechanisms and processes. There is no reason to restrict our rich discipline of theoretical neuroscience and psychology to the narrow and incorrect framework of digital computationalism.

We are calling for abandoning the classical approach of just searching for computational explanations of human behavior without worrying much, if at all, about neural computation. It does not follow that we should consider only algorithms that can be implemented using spiking neurons and abandon immediately any other research program. But it does follow that anyone seriously interested in explaining cognition should strive to show how the computations he or she postulates may be carried out by neural processes, to the extent that this can be made plausible based on current neuroscience. The better an explanation of cognition is grounded in neural computation, the better the explanation.

We cannot rule out that some cognitive phenomena are explained by digital computation. If so, then digital neural computation must somehow emerge from non-digital neural computation. The next frontier for serious digital computationalists is to develop an explicit account of how this occurs and find neuroscientific evidence to back it up.

A final objection may be raised. If computational theories of cognition must be grounded in neural computation, as we have argued, should not neural computation be grounded in turn in molecular processes, and those in atomic theory, and those in quantum mechanics, all the way down to the fundamental physical level? Is not our approach a form of brute reductionism?

Ideally, every mechanistic level should be grounded in the level below it. In an ideally complete mechanistic explanation of a phenomenon, the capacities of entities at each level are explained by the organized subcapacities of those entities' components. But no brute reductionism follows because the converse is also true. In an ideally complete mechanistic explanation, *a specific subset* of the capacities of entities at one level must be shown to contribute something essential to the capacities of entities at a higher level, such that they contribute to explaining the phenomenon. In other words, an ideally complete mechanistic explanation of a phenomenon integrates all and only the information that is needed to explain the phenomenon at each level of a mechanistic hierarchy. This is why no level of a mechanistic hierarchy (e.g., the ''cognitive'' level, if there were a unique cognitive level) can be considered in isolation from other levels.

This is also why each level of description of a mechanism allows for specific predictions that cannot be made at other levels, because each level articulates essential information that is at best implicit at lower levels. The information most relevant to explaining and predicting cognitive phenomena in detail is likely found at the levels of neural systems, neural networks, and neurons—where neural computation takes place. So any explanation of cognition worthy of that name needs to take neural computation into account.

Notes

1. Roughly speaking, classicism is the view that the type of computation that explains cognition is closely analogous to digital computation as it occurs in digital computers (Fodor & Pylyshyn, 1988); both connectionism and computational neuroscience

maintain that cognition is explained by (nonclassical) neural network activity, but they differ in whether they constrain their models primarily using behavioral evidence (connectionism) or also emphasizing neuroanatomical and neurophysiological evidence (computational neuroscience).

2. The literature is enormous. Classic and influential defenses of computationalism include McCulloch and Pitts (1943), Wiener (1948), von Neumann (1958), Marr (1982), and Churchland and Sejnowski (1992); a similar thesis, with less emphasis on the nervous system, is defended by Fodor (1975), Newell and Simon (1976), Pylyshyn (1984), and Newell (1990); more recent defenses of computationalism include Edelman (2008), Fodor (2008), Gallistel and King (2009), and Schneider (2011); influential critiques on computationalism include Lucas (1961), Dreyfus (1979), Searle (1980), Perkel (1990), Globus (1992), Penrose (1994), and van Gelder (1998).
3. Of course, there are different kinds of feedback control mechanism, some of which are more powerful than others. Therefore, it remains to be determined which feedback control mechanisms are present or absent within the nervous system. For some options, see Grush (2003).
4. Sometimes neurons themselves are said to constitute a representation. For instance, the motor cortex is said to carry a representation of the hand regardless of whether the motor cortex is firing. This notion of representation is analogous to the sense in which a map is a representation regardless of whether someone is looking at it and using it as such.
5. Terminological note: We use the terms ''to manipulate'' and ''to process'' interchangeably.
6. Therefore, the present account covers all systems that are generally deemed to compute, regardless of whether they are ''rule-based'' using more restrictive notions of rule (e.g., Horgan & Tiensen, 1989; O'Brien & Opie, 2006) .
7. Some cognitively oriented computer scientists have suggested a version of computationalism according to which cognition is computable (e.g., Rapaport, 1998; Shapiro, 1995). This view appears to be a specific version of digital computationalism according to which the computations performed by cognitive systems fall in the class of Turing-computable functions.
8. The class of functions of a real variable whose values can be generated by a general-purpose analog computer are the differentially algebraic functions, that is, solutions to algebraic differential equations (Pour-El, 1974; see also Lipshitz & Rubel, 1987; Rubel & Singer, 1985). Algebraic differential equations have the form $P(y, y', y'', \cdots y^{(n)}) = 0$. Here, P is a polynomial function with integer coefficients and y is a function of x. Some algebraic differential equations can be classified as ''universal'' to the extent that a solution of the equation can be used to approximate any continuous function of a real variable over the interval $0 \leq t < \infty$, up to an arbitrary degree of accuracy. General-purpose analog computers with as few as four integrators can generate outputs which, corresponding to this type of ''universal'' equation, can approximate any continuous function of a real variable arbitrarily well (see Boshernitzan, 1986; Duffin, 1981). Like analog computation, neural firing can be described using coupled nonlinear ordinary differential equations. But whether neural firing constitutes analog

computation depends on more precise similarities between the properties of neural firing and those of analog computation. As we shall argue in Section 5, there are important disanalogies between the two.

9. It has been argued that at the fundamental physical level, everything is discrete (e.g., Wolfram, 2002). This remains a very speculative hypothesis for which there is no direct evidence. Moreover, it would still be true that our best science of midsize objects, including digital computers at certain levels of organization, is carried out in terms of real variables.
10. Specifically, we are *not* questioning the relevance to neuroscience of Shannon's information and entropy measures, which have been used extensively by computational neuroscientists over recent years (Borst & Theunissen, 1999; Rieke et al., 1999; Dayan & Abbott, 2001; Pereda, Quiroga, & Bhattacharya, 2005).
11. For example, Eliasmith (2000) argues that neural signals are digital because spikes are all-or-none and spike timing is affected by noise. Contra Eliasmith, we argue that neural computations are not digital in the relevant sense, although we agree with Eliasmith (2003, 2007) that the theory of cognition needs to be grounded on neural computation. While Eliasmith relies on a specific approach to theoretical neuroscience (Eliasmith & Anderson, 2003), our argument is based on a general account of computation plus mainstream neuroscience.
12. The proposal just discussed should not be confused with the view defended by Aydede (1997). Aydede argues that classicism is not committed to concatenation between language-of-thought symbols because the syntactically complex symbolic structures may be encoded implicitly (by patterns of activation of neural network units) rather than explicitly (by concatenating primitive symbols). The implicit encoding schemes discussed by Aydede still involve concatenated digits in the form of patterns of active units, so Aydede's twist on classicism is yet another variant of digital computationalism.
13. Maley (2011) gives a helpful account of digital versus analog representation. Digital and analog vehicles in our sense need not be representations, although they can be. When they are, our notions are more encompassing than Maley's. Maley's notions of digital and analog representations are species of digital and analog representational vehicles in our sense.
14. Individual animals must react quickly to external stimuli and do not have the luxury of averaging trials; this is one argument used for the importance of temporal coding (VanRullen et al., 2005).

Acknowledgments

Thanks to Neal Anderson, Ken Aizawa, Glenn Carruthers, Carl Craver, Ilke Ercan, Nir Fresco, Corey Maley, Sam Scott, Oron Shagrir, Terrence Stewart, Brandon Towl, Mark Zeise, the anonymous referees, and especially Brendan Ritchie for helpful comments. Thanks to our audiences at North American Computing and Philosophy, Chicago, IL, July

2007; at Modeling, Computation and Computational Science: Perspectives from Different Sciences, Helsinki, Finland, November 2007; the Institute for Advanced Studies, Hebrew University, Jerusalem, Israel, May 2011; and Philosophy and Computation, Lund, Sweden, May 2012. Thanks to Frank Fairies for editorial assistance. Sections 2 and 3 partially overlap with Sections 4 and 5 of Piccinini (2011b). This material is based on work supported by the National Science Foundation under grant no. SES-0924527 to Gualtiero Piccinini.

References

Adrian, E. D. (1928). *The basis of sensation: The action of the sense organs*. New York: Norton.

Adrian, E. D., & Zotterman, Y. (1926). The impulses produced by sensory nerve-endings. Part 2. The response of a single end-organ. *Journal of Physiology*, *61*(2), 151–171.

Alle, H., Roth, A., & Geiger, J. R. P. (2009). Energy-efficient action potentials in hippocampal mossy fibers. *Science*, *325*, 1405–1408.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford, England: Oxford University Press.

Aydede, M. (1997). Language of thought: The connectionist contribution. *Minds and Machines*, *7*(1), 57–101.

Baddeley, R., Hancock, P., & Földiák, P. (Eds.) (2000). *Information theory and the brain*. Cambridge, England: Cambridge University Press.

Bahar, S. (2003). Effect of light on stochastic phase synchronization in the crayfish caudal photoreceptor. *Biological Cybernetics*, *89*(3), 200–213.

Bahar, S., & Moss, F. (2003). Stochastic phase synchronization in the crayfish mechanoreceptor/photoreceptor system. *Chaos*, *13*(1), 138–144.

Bair, W., & Koch, C. (1996). Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. *Neural Computation*, *8*, 1185–1202.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, *22*(4), 577–660.

Bechtel, W. (2008). *Mental mechanisms: Philosophical perspectives on cognitive neuroscience*. London: Routledge.

Bennett, M. R., & Kearns, J. L. (2000). Statistics of transmitter release at nerve terminals. *Progress in Neurobiology*, *60*(6), 545–606.

Block, N. (1978). Troubles with Functionalism. Perception and Cognition: Issues in the Foundations of Psychology. C. W. Savage. Minneapolis, University of Minnesota Press. *6*, 261–325.

Borst, A., & Theunissen, F. E. (1999). Information theory and neural coding. *Nature Neuroscience*, *2*(11), 947–957.

Boshernitzan, M. (1986). Universal formulae and universal differential equations. *The Annals of Mathematics*, 2nd Series, *124*(2), 273–291.

Bruno, M. S., & Kennedy, D. (1962). Spectral sensitivity of photoreceptor neurons in the sixth ganglion of the crayfish. *Comparative Biochemistry and Physiology*, *6*, 41–46.

Caianiello, E. R. (1961). Outline of a theory of thought processes and thinking machines. *Journal of Theoretical Biology*, *1*(2), 204–235.

Cannon, W. B. (1932). *The wisdom of the body*. New York: Norton.

Churchland, P. M. (2007). *Neurophilosophy at work*. Cambridge, MA: Cambridge University Press.

Churchland, P. S., Koch, C., & Sejnowski, T. J. (1990). What is computational neuroscience? In E. L. Schwartz (Ed.), *Computational neuroscience* (pp. 46–55). Cambridge, MA: MIT Press.

Churchland, P. S., & Sejnowski, T. J. (1992). *The computational brain*. Cambridge, MA: MIT Press.

Copeland, B. J. (2000). Narrow versus wide mechanism: Including a re-examination of Turing's views on the mind-machine issue. *The Journal of Philosophy*, *XCVI*(1), 5–32.

Craver, C. F. (2006). When mechanistic models explain. *Synthese*, *153*, 355–376.

Craver, C. F. (2007). *Explaining the brain*. Oxford, England: Oxford University Press.
Craver, C. F. (2010). Prosthetic models. *Philosophy of Science*, *77*(5), 840–851.
Cummins, R. (1983). *The nature of psychological explanation*. Cambridge, MA: MIT Press.
Davis, M., Sigal, R., & Weyuker, E. J. (1994). *Computability, complexity, and languages*. Boston: Academic Press.
Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Cambridge, MA: MIT Press.
Douglass, J. K., Wilkens, L., Pantazelou, E., & Moss, F. (1993). Noise enhancement of information transfer in crayfish mechanoreceptors by stochastic resonance. *Nature*, *365*(6444), 337–340.
Dretske, F. I. (1981). *Knowledge and the flow of information*. Cambridge, MA: MIT Press.
Dretske, F. I. (1986). Misrepresentation. In R. Bogdan (Ed.), *Belief: Form, content, and function* (pp. 17–36). New York: Oxford University Press.
Dreyfus, H. L. (1979). *What computers can't do*. New York: Harper & Row.
Duffin, R. J. (1981). Rubel's universal differential equation. *Proceedings of the National Academy of Sciences USA*, *78*(8 [Part 1: Physical Sciences]), 4661–4662.
Edelman, S. (2008). *Computing the mind: How the mind really works*. Oxford, England: Oxford University Press.
Elbert, T., & Rockstroh, B. (2004). Reorganization of human cerebral cortex: The range of changes following use and injury. *Neuroscientist*, *10*(2), 129–141.
Eliasmith, C. (2000). Is the brain analog or digital? The solution and its consequences for cognitive science. *Cognitive Science Quarterly*, *1*(2), 147–170.
Eliasmith, C. (2003). Moving beyond metaphors: Understanding the mind for what it is. *Journal of Philosophy*, *100*(10), 493–520.
Eliasmith, C. (2007). How to build a brain: From function to implementation. *Synthese*, *159*(3), 373–388.
Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation and dynamics in neurobiological systems*. Cambridge, MA: MIT Press.
Ermentrout, G. B., & Terman, D. H. (2010). *Mathematical foundations of neuroscience*. New York: Springer.
Fodor, J. A. (1965). Explanations in psychology. In M. Black (Ed.), *Philosophy in America*. London: Routledge and Kegan Paul.
Fodor, J. A. (1968). *Psychological explanation*. New York: Random House.
Fodor, J. A. (1975). *The language of thought*. Cambridge, MA: Harvard University Press.
Fodor, J. A., & Pylyshyn, Z. W. (1988). *Connectionism and Cognitive Architecture*. *Cognition*, *28*, 3–71.
Fodor, J. A. (2000). *The Mind Doesn't Work That Way*. Cambridge, MA: MIT Press.
Fodor, J. A. (2008). *LOT 2: The language of thought revisited*. Oxford, England: Oxford University Press.
Fries, P., Reynolds, J. H., Rorie, A. E., & Desimone, R. (2001). Modulation of oscillatory neuronal synchronization by selective visual attention. *Science*, *291*(5508), 1560–1563.
Fries, P., Womelsdorf, T., Oostenveld, R., & Desimone, R. (2008). The effects of visual stimulation and selective visual attention on rhythmic neuronal synchronization in macaque area V4. *Journal of Neuroscience*, *28*(18), 4823–4835.
Gallistel, C. R., & King, A. P. (2009). *Memory and the computational brain: Why cognitive science will transform neuroscience*. Malden, MA: Wiley-Blackwell.
Garson, J. (2003). The introduction of information into neurobiology. *Philosophy of Science*, *70*, 926–936.
Gazzaniga, M. (Ed.) (2009). *The cognitive neurosciences*. Cambridge, MA: MIT Press.
van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, *XXI*, 615–665.
Gerard, R. W. (1951). Some of the problems concerning digital notions in the nentral nervous system. Cybernetics: Circular causal and feedback mechanisms in biological and social systems. In H. V. Foerster, M. Mead & H. L. Teuber (Eds.), *Transactions of the seventh conference* (pp. 11–57). New York: Macy Foundation.
Glennan, S. (2002). Rethinking mechanistic explanation. *Philosophy of Science*, *69*, S342–S353.
Globus, G. G. (1992). Towards a noncomputational cognitive neuroscience. *Journal of Cognitive Neuroscience*, *4*(4), 299–310.
Gollisch, T. (2009). Throwing a glance at the neural code: Rapid information transmission in the visual system. *HFSP Journal*, *3*(1), 36–46.

Grice, H. P. (1957). Meaning. *The Philosophical Review*, *66*(3), 377–388.

Grush, R. (2003). In defense of some 'Cartesian' assumptions concerning the brain and its operation. *Biology and Philosophy*, *18*, 53–93.

Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, *27*, 377–442.

Hille, B. (2001). *Ion channels of excitable membranes*. Sunderland, MA: Sinauer.

Hogdkin, A., & Huxley, A. (1952). Currents carried by sodium and potassium ions through the membrane of the giant axon of *Loligo*. *Journal of Physiology*, *116*, 449–472.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, *79*, 2554–2558.

Horgan, T., & Tiensen, J. (1989). Representation without rules. *Philosophical Topics*, *XVII*(1), 27–43.

Householder, A. S., & Landahl, H. D. (1945). *Mathematical biophysics of the central nervous system*. Bloomington, IN: Principia.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, *160*, 106–154.

Humphreys, P. (2004). *Extending ourselves: Computational science, empiricism, and scientific method*. Oxford, England: Oxford University Press.

Inan, M., & Crair, M. C. (2007). Development of cortical maps: Perspectives from the barrel cortex. *Neuroscientist*, *13*(1), 49–61.

Johansson, R. S., & Birznieks, I. (2004). First spikes in ensembles of human tactile afferents code complex spatial fingertip events. *Nature Neuroscience*, *7*, 170–177.

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference and consciousness*. New York: Cambridge University Press.

Johnson-Laird, P. N. (2010). Mental models and human reasoning. *Proceedings of the National Academy of Sciences USA*, *107*(43), 18243–18250.

Kalat, J. W. (2008). *Biological psychology*. Belmont, CA: Wadsworth Publishing.

Knight, B. W. (1972). Dynamics of encoding in a population of neurons. *Journal of General Physiology*, *59*(6), 734–766.

Kosslyn, S. M., Thompson, W. I., & Ganis, G. (2006). *The case for mental imagery*. New York: Oxford University Press.

Lapicque, L. (2007). Quantitative investigation of electrical nerve excitation treated as polarization (N. Brunel & M. van Rossum, Trans.). *Biological Cybernetics*, *97*, 341–349. (Original work published 1907).

Lashley, K. S. (1958). Research Publications of the Association for Research in Nervous & Mental Disease. Cerebral organization and behavior. In F. A. Beach, D. O. Hebb, C. T. Morgan & H. V. Nissen (Eds.), *The neuropsychology of Lashley; Selected papers of K. S. Lashley* (pp. 529–543). New York: McGraw-Hill. *The Brain and Human Behavior, Proceedings of the Association for Research of Nervous and Mental Disorders*, *36*, 1–18. (1960) (reprint)

Le Bon-Jego, M., & Yuste, R. (2007). Persistently active, pacemaker-like neurons in neocortex. *Frontiers in Neuroscience*, *1*(1), 123–129.

Levin, J. E., & Miller, J. P. (1996). Broadband neural encoding in the cricket cercal sensory system enhanced by stochastic resonance. *Nature*, *380*(6570), 165–168.

Lin, J.-W., & Faber, D. S. (2002). Modulation of synaptic delay during synaptic plasticity. *Trends in Neurosciences*, *25*(9), 449–455.

Lipshitz, L., & Rubel, L. A. (1987). A differentially algebraic replacement theorem, and analog computability. *Proceedings of the American Mathematical Society*, *99*(2), 367–372.

London, M., Roth, A., Beeren, L., Häusser, M., & Latham, P. E. (2010). Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, *466*, 123–127.

Lucas, J. R. (1961). Minds, machines, and Gödel. *Philosophy*, *36*, 112–137.

Lycan, W. (1990). The continuity of levels of nature. In W. Lycan (Ed.), *Mind and cognition* (pp. 77–96). Malden, MA: Blackwell.

Mainen, Z. F., & Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science*, *268*(5216), 1503–1506.
Maley, C. J. (2011). Analog and digital, continuous and discrete. *Philosophical Studies*, *115*, 117–131.
Marr, D. (1982). *Vision*. New York: Freeman.
McCulloch, W. S., & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *7*, 115–133.
Meister, M., & Berry, M. J. (1999). The neural code of the retina. *Neuron*, *22*, 435–450.
Millikan, R. G. (2004). *Varieties of meaning*. Cambridge, MA: MIT Press.
Mills, J. W. (2008). The nature of the extended analog computer. *Physica D: Nonlinear Phenomena*, *237*(9), 1235–1256.
Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.
Morcom, A. M., & Fletcher, P. C. (2007). Does the brain have a baseline? Why we should be resisting a rest. *NeuroImage*, *37*(4), 1073–1082.
Moss, F., Ward, L. M., & Sannita, W. G. (2004). Stochastic resonance and sensory information processing: A tutorial and review of application. *Clinical Neurophysiology*, *115*(2), 267–281.
Murphy, G. L. (2002). *The big book of concepts*. Cambridge, MA: MIT Press.
von Neumann, J. (1958). *The computer and the brain*. New Haven, CT: Yale University Press.
Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
Newell, A., & Simon, H. A. (1976). Computer science as an empirical enquiry: Symbols and search. *Communications of the Association for Computing Machinery*, *19*, 113–126.
O'Brien, G., & Opie, J. (2006). How do connectionist networks compute? *Cognitive Processing*, *7*, 30–41.
Oram, M. W., Wiener, M. C., Lestienne, R., & Richmond, B. J. (1999). Stochastic nature of precisely timed spike patterns in visual system neuronal responses. *Journal of Neurophysiology*, *81*, 3021–3033.
O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain*. Cambridge, MA: MIT Press.
Penrose, R. (1994). *Shadows of the mind*. Oxford, England: Oxford University Press.
Pereda, E., Quiroga, R. Q., & Bhattacharya, J. (2005). Nonlinear multivariate analysis of neurophysiological signals. *Progress in Neurobiology*, *77*(1–2), 1–37.
Perkel, D. H. (1990). Computational neuroscience: Scope and structure. In E. L. Schwartz (Ed.), *Computational neuroscience* (pp. 38–45). Cambridge, MA: MIT Press.
Piccinini, G. (2004). The first computational theory of mind and brain: A close look at McCulloch and Pitts's ''Logical calculus of ideas immanent in nervous activity.'' *Synthese*, *141*(2), 175–215.
Piccinini, G. (2007a). Computing mechanisms. *Philosophy of Science*, *74*(4), 501–526.
Piccinini, G. (2007b). Computational modeling vs. computational explanation: Is everything a Turing machine, and does it matter to the philosophy of mind? *Australasian Journal of Philosophy*, *85*(1), 93–115.
Piccinini, G. (2007c). Computationalism, the Church-Turing thesis, and the Church-Turing fallacy. *Synthese*, *154*(1), 97–120.
Piccinini, G. (2008). Computers. *Pacific Philosophical Quarterly*, *89*(1), 32–73.
Piccinini, G. (2010). The resilience of computationalism. *Philosophy of Science*, *77*(5), 852–861.
Piccinini, G. (2011a). The physical Church-Turing thesis: Modest or bold? *British Journal for the Philosophy of Science*, *62*(4), 733–769.
Piccinini, G. (2011b). Computationalism. In E. Margolis, R. Samuels & S. P. Stich (Eds.), *Oxford handbook of philosophy and cognitive science* (pp. 222–229). Oxford, England: Oxford University Press.
Piccinini, G., & Craver, C. (2011). Integrating psychology and neuroscience: Functional analyses as mechanism sketches. *Synthese*, *183*(3), 283–311.
Piccinini, G., & Scarantino, A. (2011). Information processing, computation, and cognition. *Journal of Biological Physics*, *37*(1), 1–38.
Pikovsky, A., Rosenblum, M., & Kurths, J. (2003). *Synchronization: A universal concept in nonlinear sciences*. Cambridge, MA: Cambridge University Press.
Posner, M. I. (2004). *Cognitive neuroscience of attention*. New York: Guilford Press.

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). *Transactions of the American Mathematical Society*, *199*, 1–28.

Pylyshyn, Z. W. (1984). *Computation and cognition*. Cambridge, MA: MIT Press.

Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., & Shulman, G. L. (2001). A default mode of brain function. *Proceedings of the National Academy of Sciences USA*, *92*(1), 676–682.

Raichle, M. E., & Mintun, M. A. (2006). Brain work and brain imaging. *Annual Review of Neuroscience*, *29*, 449–476.

Ramsey, W. M. (2007). *Representation reconsidered*. Cambridge, MA: Cambridge University Press.

Rapaport, W. J. (1998). How minds can be computational systems. *Journal of Experimental and Theoretical Artificial Intelligence*, *10*, 403–419.

Rashevsky, N. (1938). *Mathematical biophysics: Physicomathematical foundations of biology*. Chicago, IL: University of Chicago Press.

Recanzone, G. H., Merzenich, M. M., & Schreiner, C. E. (1992). Changes in the distributed temporal response properties of SI cortical neurons reflect improvements in performance on a temporally based tactile discrimination task. *Journal of Neurophysiology*, *67*(5), 1071–1091.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1999). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press.

Rogers, H. (1967). *Theory of recursive functions and effective computability*. New York: McGraw-Hill.

Roy, A., Steinmetz, P. N., Hsiao, S. S., Johnson, K. O., & Niebur, E. (2007). Synchrony: A neural correlate of somatosensory attention. *Journal of Neurophysiology*, *98*(3), 1645–1661.

Rubel, L. A. (1985). The brain as an analog computer. *Journal of Theoretical Neurobiology*, *4*, 73–81.

Rubel, L. A. (1993). The extended analog computer. *Advances in Applied Mathematics*, *14*(1), 39–50.

Rubel, L. A., & Singer, M. F. (1985). A differentially algebraic elimination theorem with application to analog computability in the calculus of variations. *Proceedings of the American Mathematical Society*, *94*(4), 653–658.

Rumelhart, D. E., & McClelland, J. M. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press.

Scarantino, A., & Piccinini, G. (2010). Information without truth. *Metaphilosophy*, *43*(3), 313–330.

Schneider, S. (2011). *Language of thought: A new philosophical direction*. Cambridge, MA: MIT Press.

Schreiner, C. E., & Winer, J. A. (2007). Auditory cortex mapmaking: Principles, projections, and plasticity. *Neuron*, *56*(2), 356–365.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, *3*, 417–457.

Shadlen, M. N., & Newsome, W. T. (1998). The variable discharge of cortical neurons: Implications for connectivity, computation and information coding. *Journal of Neuroscience*, *18*, 3870–3896.

Shagrir, O. (2010). Brains as analog-model computers. *Studies in History and Philosophy of Science*, *41*, 271–279.

Shannon, C. E., & Weaver, W. (1949). The Mathematical Theory of Communication. Urbana, IL, University of Illinois Press.

Shapiro, S. C. (1995). Computationalism. *Minds and Machines*, *5*(4), 517–524.

Shepherd, G. (1999). Electronic properties of axons and dendrites. In M. J. Zigmond, F. E. Bloom, S. C. Landys, J. L. Roberts, & L. R. Squire (Eds.), *Fundamental neuroscience* (pp. 115–117). Amsterdam: Academic Press.

Siegelmann, H. T. (2003). Neural and super-Turing computing. *Minds and Machines*, *13*(1), 103–114.

Smolensky, P. (1989). Connectionist modeling: Neural computation/mental connection. In L. A. C. L. Nadel, P. Culicover & R. M. Harnish (Eds.), *Neural connections, mental computation*. Cambridge, MA: MIT Press.

Stein, R. (1965). A theoretical analysis of neuronal variability. *Biophysical Journal*, *5*(2), 173–194.

Steinmetz, P. N., Roy, A., Fitzgerald, P. J., Hsiao, S. S., Johnson, K. O., & Niebur, E. (2000). Attention modulates synchronized neuronal firing in primate somatosensory cortex. *Nature*, *404*(6774), 187–190.

Tass, P. A., Fieseler, T., Dammers, J., Dolan, K., Morosan, P., Majtanik, M., Boers, F., Muren, A., Zilles, K., & Fink, G. R. (2003). Synchronization tomography: A method for three-dimensional localization of phase synchronized neuronal populations in the human brain using magnetoencephalography. *Physical Review Letters*, *90*(8), 088101.

Thompson, E. (2007). *Mind in life: Biology, phenomenology, and the sciences of mind*. Cambridge, MA: Harvard University Press.

Turing, A. M. (1936–7 [1965]). On computable numbers, with an application to the Entscheidungsproblem. In M. Davis (Ed.), *The undecidable* (pp. 116–154). Hewlett, NY: Raven Press.

Uhlhaas, P. J., & Singer, W. (2006). Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron*, *52*(1), 155–168.

VanRullen, R., Guyonneau, R., & Thorpe, S. J. (2005). Spike times make sense. *Trends in Neuroscience*, *28*(1), 1–4.

Vera, A. H., & Simon, H. A. (1993). Situated action: A symbolic interpretation. *Cognitive Science*, *17*, 7–48.

Waskan, J. A. (2003). Intrinsic cognitive models. *Cognitive Science*, *27*(2), 259–283.

White, J. A., Klink, R., Alonso, A., & Kay, A. R. (1998). Noise from voltage-gated ion channels may influence neuronal dynamics in the entorhinal cortex. *Journal of Neurophysiology*, *80*(1), 262–269.

Wiener, N. (1948). *Cybernetics or control and communication in the animal and the machine*. Cambridge, MA: MIT Press.

Wilson, H. R., & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, *12*(1), 1–24.

Wimsatt, W. C. (2002). Functional organization, analogy, and inference. in A. Ariew, R. Cummins & M. Perlman (Eds.), *Functions: New essays in the philosophy of psychology and biology* (pp. 173–221). Oxford, England: Oxford University Press.

Winsberg, E. (2010). *Science in the age of computer simulation*. Chicago, IL: University of Chicago Press.

Wolfram, S. (2002). *A new kind of science*. Champaign, IL: Wolfram Media.

Wong, R. K. S., Traub, R. D., & Miles, R. (1986). Cellular basis of neural synchrony in epilepsy. In A. V. Delgado-Escueta, A. A. Ward, D. M. Woodbury, & R. J. Porter (Eds.), *Advances in neurology*. New York: Raven Press.

Yassin, L., Benedetti, B. L., Jouhanneau, J. S., Wen, J. A., Poulet, J. F. A., & Barth, A. L. (2010). An embedded subnetwork of highly active neurons in the neocortex. *Neuron*, *68*, 1043–1050.

Zhang, D., & Raichle, M. E. (2010). Disease and the brain's dark energy. *Nature Reviews Neurology*, *6*, 15–28.