The Resilience of Computationalism[1]

Gualtiero Piccinini

University of Missouri-St. Louis

This is a preprint of an article whose final and definitive form will be published in *Philosophy of Science*; *Philosophy of Science* is available online at http://www.journals.uchicago.edu/loi/phos.

Abstract: Computationalism—the view that cognition is computation—has always been controversial. It faces two types of objection. According to insufficiency objections, computation is insufficient for some cognitive phenomenon *X*. According to objections from neural realization, cognitive processes are realized by neural processes, but neural processes have feature *Y* and having *Y* is incompatible with being (or realizing) computations. In this paper, I explain why computationalism has survived these objections. Insufficiency objections are at best partial: for all they establish, computation may be sufficient for cognitive phenomena other than *X*, may be part of the explanation for *X*, or both. Objections from neural realization are based either on a false contrast between feature *Y* and computation or on an account of computation that is too vague to yield the desired conclusion. To adjudicate the dispute between computationalism and its foes, I will conclude that we need a better account of computation.

---

[1] This paper benefited from comments by Anibal Astobiza, Carl Craver, Tony Dardis, Mark Sprevak, and an anonymous referee, and from research support from the University of Missouri Research Board. Thanks also to my audience at the 2008 PSA meeting. Thanks to James Virtel for editorial assistance.

1. Introduction

Roughly speaking, computationalism says that cognition is (a kind of) computation, or that cognitive capacities are explained by the agent's computations. The cognitive processes and behavior of agents are the explanandum. The computations performed by the agents' cognitive system are the proposed explanans. Since the cognitive systems of biological organisms are more or less their nervous systems, we may say that according to computationalism, the cognitive processes and behavior of organisms are explained by neural computations.

Some people might prefer to say that cognitive systems are "realized" by nervous systems and that according to computationalism cognitive computations are "realized" by neural processes. In this paper, nothing hinges on the nature of the metaphysical relation between cognitive systems and nervous systems, or between computations and neural processes. For present purposes, if a neural process realizes a computation, then that neural process *is* a computation. Thus, I will couch much of my discussion in terms of nervous systems and neural computation.[2]

Before proceeding, we should dispense with a possible red herring. Contrary to a common assumption, computationalism does not stand in opposition to connectionism.

---

[2] See also Piccinini 2008a, b, 2009, forthcoming for why computationalism is to be tested by evidence about the nervous system.

Connectionism, in the most general and prevalent sense of the term, is the claim that cognitive phenomena are explained (at some level and at least in part) by the processes of neural networks. This is a truism supported by current neuroscientific evidence. Everybody ought to be a connectionist in this general sense.[3] The relevant question is, are neural processes computations? More precisely, are the cognitively relevant processes carried out by nervous systems computations? Computationalists say "yes", anti-computationalists say "no". This paper investigates the soundness of arguments against computationalism.

Computationalism has met a wide range of objections ever since Warren McCulloch and Walter Pitts (1943) first proposed it. Yet computationalism has become and remains the dominant theory of cognition in psychology, neuroscience, and philosophy of mind. This paper explains why traditional objections are inconclusive and suggests a more effective strategy for assessing computationalism.

There are two types of objection: (1) those based on alleged differences between computations and cognitive processes, and (2) those based on alleged differences between computations and neural processes. We will look at each type in turn.

[3] There are other, more specialized senses of 'connectionism'. For a more nuanced discussion of various senses of 'connectionism' and their relation to computationalism, see Piccinini (2008b, 2009).

Before proceeding, we ought to define 'computation' at least roughly. To a first approximation, computation is what computers and similar machines do. The paradigmatic computer—the electronic, programmable, stored-program, universal digital computer—is the kind we use every day. Of course, computationalism is not committed to the claim that the brain is like such a computer in every respect, but only to a *partial* analogy between brains and computers. We will see that in order to assess the objections to computationalism, it is crucial to distinguish the features of computers that are essential to computation from those that are not essential. I will argue that the shortcomings of current objections suggest the need for greater precision on what does and does not count as computation. A more precise account of computation is necessary to evaluate the plausibility of computationalism.

2. 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 mathematical insight (Lucas 1961; Penrose 1994), qualia (Block 1978), intentionality or "understanding" (Searle 1980), abduction (Fodor 2000), embodiment (Thompson 2007), and embeddedness (van Gelder 1998). All these insufficiency objections are unsound.

To begin with, the reality or relevance of some *X*'s is controversial. For some *X*'s, it may be questioned whether *X* is real or is a genuine feature of cognition. For instance, someone might deny that qualia are real (Dennett 1988) or that the mind is embodied and embedded in a nontrivial sense. Consider that neural systems can exhibit many of their characteristic processes without being embedded, as the success of in vitro neurophysiology attests. Alternatively, someone might argue that qualia, although real, are not part of cognition proper. If so, then qualia fall outside the scope of computationalism.

Furthermore, even among people who agree that a candidate *X* is real, it may be controversial whether computation is sufficient for *X*. For instance, many have argued that the power of mathematical insight does not refute computationalism (e.g., Feferman 1995), or that qualia can be given a computational account (e.g., Lycan 1987), or that abduction can be done computationally. Each candidate *X* poses its own special challenges. There is no room here to review the complex issues associated with each *X*. Fortunately, for present purposes we can skip the details. For all objections of this type suffer from a more fundamental weakness: even if computation is insufficient for some cognitive phenomenon *X*, computationalism can be retained in its most plausible form. In its strongest formulation, computationalism says that all cognitive phenomena can be accounted for by computation alone. Few people, if any, hold this view. The reason is precisely that some aspects of cognition, such as the candidate *X*'s listed above, appear to involve more than computation.

A weaker and more plausible formulation of computationalism is that computation is a proper part of the explanation of all or most cognitive processes and behavior. This formulation is entirely adequate for the role computationalism plays in science. Scientists who carry out computationalist research programs offer computational explanations of cognitive tasks. Such explanations are usually formulated in terms of computations over representations, presupposing some account of how the representations get their representational content. In some cases, such explanations are formulated in terms of conscious states, presupposing some account of how the states get to be conscious. Finally, such explanations are often formulated in terms of states and processes that are coupled with the organism's body and environment. Again, this presupposes that neural computations are coupled with the body and environment in the relevant way.

Thus, a complete explanation of cognition may require, in addition to an appeal to computation, an account of the states' consciousness, representational content, embodiment, and embeddedness. Computation may be insufficient for intentionality, consciousness, etc., but it may still be an important part of the explanation of cognitive processes and behavior.

As Mark Sprevak has pointed out to me, this concessive strategy only goes so far. If insufficiency objections turned out to be not only correct, but correct to the point that computation only plays a minor role in the explanation of a few cognitive processes, then computationalism would lose most of its luster. In other words, if computation contributed

little to explaining cognition and solving the really hard problems (such as consciousness and intentionality), then computationalism would no longer be a very interesting and substantive view. But as they stand, the present objections do not undermine the most interesting version of computationalism: the claim that neural computations are a necessary and important part of the explanation of most cognitive processes and behaviors of organisms. To undermine such a view, we'll have to get our hands dirty and look at how nervous systems work.

In conclusion, insufficiency objections—to the effect that computation is insufficient for *X*, for some feature *X* that cognition has—are inconclusive. At best they show that a complete theory of cognition involves more than an appeal to computation—something that few if any people deny.

3. Objections from Neural Realization

There is also a long tradition of objections to computationalism based on alleged differences between neural processes and computations. Compared to insufficiency objections, these objections from neural realization have been neglected by philosophers. I will now discuss some important objections from neural realization and why they fail. Unlike the previous class of objections, each objection requires a separate treatment.

3.1 Non-electrical Processes

Like signals in electronic computers, neural signals include the propagation of electric charges. But unlike signals in electronic computers, neural signals also include the diffusion of chemical substances. More specifically, there are at least two classes of neural signals that have no analogue in electronic computers: neurotransmitters and hormones. Therefore, according to this objection, neural processes are not computations (cf. Perkel 1990).

As it stands, this objection is inconclusive. Computations are realizable by an indefinite number of physical substrates, so computing mechanisms can be built out of an indefinite number of technologies.[4] Today's main computing technology is electronic, but computers used to be built out of mechanical or electromechanical components, and there is active research on so called "unconventional computing" such as optical, DNA, and quantum computing. There is no principled reason why computations cannot be realized by processes of chemical diffusion.

Even if the chemical signals in question were essentially non-computational, pointing out that they occur in the brain would not show that neural processes are non-computational. Here, different considerations apply to different cases. Many hormones are released and absorbed at the periphery of the nervous system. So, the release and uptake of such

[4] This is not to suggest that every process is a computation—far from it (Piccinini 2007b).

hormones might be treated simply as part of the input-output interface of the neural computations, in the same way that sensory and motor transducers are.

With respect to neurotransmitters and neuropeptides, we need to remember that computational explanations apply at some mechanistic levels and not others (cf. Piccinini 2007a). For example, at one mechanistic level, the activities of ordinary electronic computers are explained by the programs they execute; at another level, they are explained by electronic activity in their circuits; at yet another level, by quantum mechanical processes. We should also remember that not all processes that occur in a computing system are computations. As a counterexample, consider temperature regulation: it occurs in all artificial computers, but it is not part of their computations. So, anyone who wishes to appeal to chemical signals as an objection to computationalism needs to show both that those chemical processes are non-computational and that they occur at the mechanistic level at which neural systems are purported to perform computations. This cannot be done without clear criteria for what does and does not count as computation. And such criteria cannot be based simply on whether the signals being used are chemical or electrical. The presence of chemical signals in nervous systems, by itself, does not threaten computationalism.

3.2 Temporal Constraints

Neural processes are temporally constrained in real time, whereas computations are not; hence, neural processes are not computations (cf. Globus 1992, van Gelder 1998). This

objection trades on an ambiguity between mathematical representation of time and real time. True, computations are temporally unconstrained in the sense that they can be defined and individuated in terms of computational steps, independently of how much time it takes to complete a step. More accurately, this point applies to algorithmic (i.e., step by step) computations. Many neural networks that perform computations do so without their computations being decomposable into algorithmic steps (Piccinini 2008b), so the premise of this objection does not apply to them. But the deeper flaw in this objection is that 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. Consider differential equations—a type of description favored by many anti-computationalists. Differential equations contain time variables, but these do not correspond to real time any more than the time steps of a Turing machine correspond to any particular real time interval. In order for the time variables of differential equations to correspond to real time, a temporal scale must be specified (e.g., whether time is being measured in seconds, nanoseconds, years, etc.). By the same token, the time steps of a computer can be made to correspond to real time by specifying an appropriate time scale. When this is done, computations are no less temporally constrained than any other process. As any computer designer knows, you can't build a well-functioning computer without taking the speed of components and the timing of their operations into account (in real time). In fact, many computationalists have been concerned with temporal constraints on the

computations that, according to their theory, explain cognitive phenomena (Pylyshyn 1984; Newell 1990; Vera and Simon 1993).[5]

The two objections above share a common problem: the properties they canvass (electrical signals, temporal constraints) are irrelevant to whether a process is computational. It is easy to show that they miss their target. The following objections are more promising, because they are based on properties that are relevant to whether neural processes are computations.

3.3 Analog vs. Digital

Neural processes are analog whereas computations are digital; hence, neural processes are not computations (cf. Dreyfus 1979; Perkel 1990). This is one of the oldest and most often repeated objections to computationalism. Unfortunately, it is formulated using an ambiguous terminology that comes with conceptual traps.

A common but superficial reply to this objection is that computations may be analog as well as digital. After all, most people who work in this area have heard of machines called 'analog computers'. Analog computers were invented before digital computers and were used widely in the 1950s and 60s. If neural processes are analog, they might be analog

[5] Another rebuttal of the objection from temporal constraints—developed independently of the present one—is given by Weiskopf (2004). Some of his considerations go in the same direction as mine.

computations, and if so, then computationalism remains in place in an analog version. In fact, the original proponents of the analog-versus-digital objection did not offer it as a refutation of all forms of computationalism, but only of McCulloch and Pitts's digital version of computationalism (Gerard 1951). This reply is superficial, however, because it employs the ambiguous terms 'analog' and 'computation'. Depending on what is meant by these terms, an analog process may or may not be an analog computation, and an analog computation may or may not be a computation in the sense relevant to computationalism.

In a loose sense, 'analog' refers to the processes of any system that at some mechanistic level can be characterized as the temporal evolution of real (i.e., continuous, or analog) variables. Some proponents of the analog-vs.-digital objection seem to employ some variant of this broad notion (cf. van Gelder 1998). It is easy to see that neural processes fall in this class, but this does not help us refute computationalism. On one hand, most processes that are analog in this sense, such as the weather, planetary motion, and digestion, are paradigmatic examples of processes that are not computations in any interesting sense. Neural processes may fall into this class. On the other hand, it is also easy to see that virtually all systems, including computing mechanisms such as computers and computing neural networks, fall within this class. Continuous variables have more expressive power than discrete ones, so continuous variables can be used to express the same information and more. But more importantly, most of our physics and engineering of midsize objects, including much of the physics and engineering of digital computers, is couched in terms of differential equations,

which employ continuous variables. So, the trivial fact that neural mechanisms are analog in this loose sense does nothing to either refute computationalism or establish an analog version of it.

In a more restricted sense, 'analog' refers to analog computers—a class of machines that used to be employed for solving certain differential equations.[6] Analog computers are either explicitly or implicitly invoked by many of those who formulate the analog-vs.-digital objection. Claiming that neural systems are analog in the sense that they are analog computers is a strong empirical hypothesis. Since this hypothesis claims that neural mechanisms are analog computing mechanisms, it may be considered a form of computationalism. Nevertheless, this analog computationalism is incompatible with the mainstream (digital) form of computationalism, which is the one under discussion in this paper.

But in spite of some valiant attempts to argue for something like this analog version of computationalism (Rubel 1985), neural mechanisms and analog computers are quite different. The principal difference is that the vehicles of analog computation are real variables, namely, variables that vary continuously over time, whereas the main vehicles of

[6] The class of equations solved by analog computers is that of the differentially algebraic functions, i.e., functions that arise as solutions to algebraic differential equations. Algebraic differential equations are equations of the form $P(y, y', y'', \ldots y^{(n)}) = 0$, where P is a polynomial with integer coefficients and y is a function of x (Pour-El 1974).

neural processes appear to be neuronal spike trains, which are sequences of all-or-none events. Of course, the *rate* at which a neuron fires can increase or decrease in a graded way within certain limits. For this reason, firing rates are sometimes called 'analog'. But even firing rates are not real variables, which can literally take any real value within the relevant interval. Once again, the term 'analog' proves ambiguous.

Although it's implausible that nervous systems are analog computers in a strict sense, they may still share some interesting properties with them while failing to share some interesting properties with digital computers. For instance, it is plausible that the temporal evolution of neural processes, like that of analog computers but unlike that of digital computers, cannot be analyzed into functionally relevant discrete time intervals. (Evidence: none of the mainstream studies of neural systems have found any such functionally relevant discrete time intervals.) Whether this point establishes that neural processes are not computations, however, depends on whether discrete temporal evolution is essential to (digital) computation. As I already pointed out, there are digital computing devices (such as many neural networks) whose internal dynamics are not discrete. Thus, even this disanalogy between brains and computers fails to establish that neural processes are not (digital) computations. To determine whether they are, more needs to be said about what counts as (digital) computation.

3.4 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 need not be the effect of inputs alone either. Computations may 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. Perhaps the spontaneous activity of neurons is not quite analogous to the internal states that affect computations. To say why, more needs to be said about spontaneous neural activity and the role it plays vis-à-vis internal states of computations and the roles they play.

While the first two objections appealed to properties that are simply irrelevant to whether something is computational, these last two objections fail primarily because they lack a precise enough account of what does and does not count as computation (in the relevant sense). To make progress on whether computationalism holds, we need a precise account of computation.

4. How to Test Computationalism

As we have seen, objections to computationalism fall into two main classes: insufficiency objections and objections from neural realization. According to insufficiency objections,

computation is insufficient for some cognitive phenomenon *X*. According to objections from neural realization, neural processes have feature *Y* and having *Y* is incompatible with being a computation. In this paper, I explained why computationalism has survived these objections. Insufficiency objections are at best partial: for all they establish, computation may be sufficient for cognitive phenomena other than *X*, may be part of the explanation for *X*, or both. Objections from neural realization are based either on a false contrast between feature *Y* and computation or on an account of computation that is too vague to yield the desired conclusion.

To date, objections from neural realization have been ineffective for lack of a precise enough account of computation. But their method is sound. The method is to consider relevant functional properties of computations, consider the evidence about neural processes, and see whether neural processes appear to possess the relevant functional properties. In order to improve on past debates over computationalism, we need to identify general functional properties of computations and compare them with relevant functional properties of neural processes.

In some of my recent work (Piccinini 2007a, 2008a, b), I have attempted to identify precisely such general functional properties of (digital) computations. Very briefly put, computations are processes whose function is to produce output strings of "digits" (i.e., sequences of discrete states) from input strings and (possibly) internal states, in accordance with a general

rule defined over the strings. I have also argued that my account of computation captures the notion of computation relevant to (digital) computationalism. To see whether computationalism holds, what remains to be done is to analyze the neuroscientific evidence.

References

Block, Ned. 1978. “Troubles with Functionalism.” In *Perception and Cognition: Issues in the Foundations of Psychology*, ed. C. Wade Savage, 261-325. Minneapolis: University of Minnesota Press.

Churchland, Paul S., and Terrence J. Sejnowski. 1992. *The Computational Brain*. Cambridge, MA: MIT Press.

Dayan, Peter, and Larry F. Abbott. 2001. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press.

Dennett, Daniel C. 1988. “Quining Qualia.” In *Consciousness in Contemporary Science*, eds Anthony J. Marcel, and Edoardo Bisiach, 42-77. Oxford: Clarendon Press.

Dreyfus, Hubert L. 1979. *What Computers Can't Do*. New York: Harper & Row.

Feferman, Solomon. 1995. "Penrose's Gödelian Argument." *Psyche* 2:21-32.

Fodor, Jerry A. 2000. *The Mind Doesn't Work That Way*. Cambridge, MA: MIT Press.

Gerard, Ralph W. 1951. “Some of the Problems Concerning Digital Notions in the Central Nervous System.” *Cybernetics: Circular Causal and Feedback Mechanisms in Biological and Social Systems. Transactions of the Seventh Conference March 23-24, 1950*, ed. Heinz Von Foerster, Margaret Mead, and Hans Lukas Teuber, 11-57. New York: Macy Foundation.

Globus, Gordon G. 1992. "Towards a Noncomputational Cognitive Neuroscience." *Journal of Cognitive Neuroscience* 4:299-310.

Lucas, John Randolph. 1961. "Minds, Machines, and Gödel." *Philosophy* 36:112-137.

Lycan, William. 1987. *Consciousness*. Cambridge, MA: MIT Press.

McCulloch, Warren S., and William H. Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 7:115-133.

Minsky, Marvin, and Seymour Papert. 1969. *Perceptrons*. Cambridge, MA: MIT Press.

Nagel, Thomas. 1974. "What Is It Like to Be a Bat?." *Philosophical Review* 83:435-450.

Newell, Allen. 1990. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Penrose, Roger. 1994. *Shadows of the Mind*. Oxford: Oxford University Press.

Perkel, Donald H. 1990. “Computational Neuroscience: Scope and Structure.” *Computational Neuroscience*, ed. Eric L. Schwartz, 38-45. Cambridge, MA: MIT Press.

Piccinini, Gualtiero. 2007a. "Computing Mechanisms." *Philosophy of Science* 74:501-526.

———. 2007b. "Computational Modeling vs. Computational Explanation: Is Everything a Turing Machine, and Does It Matter to the Philosophy of Mind?." *Australasian Journal of Philosophy* 85:93-115.

———. 2008a. "Computers." *Pacific Philosophical Quarterly* 89:32-73.

———. 2008b. "Some Neural Networks Compute, Others Don't." *Neural Networks* 21:311-321.

———. 2009. "Computationalism in the Philosophy of Mind." *Philosophy Compass* 4: 1-18.

———. Forthcoming. "The Mind as Neural Software? Understanding Functionalism, Computationalism, and Computational Functionalism." *Philosophy and Phenomenological Research*.

Pour-El, Marian 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, Zenon W. 1984. *Computation and Cognition*. Cambridge, MA: MIT Press.

Rubel, Lee A. 1985. "The Brain as an Analog Computer." *Journal of Theoretical Neurobiology* 4:73-81.

Searle, John R. 1980. "Minds, Brains, and Programs." *The Behavioral and Brain Sciences* 3: 417-457.

Thompson, Evan. 2007. *Mind in Life: Biology, Phenomenology, and the Sciences of Mind*. Cambridge, MA: Harvard University Press.

van Gelder, Tim. 1998. "The Dynamical Hypothesis in Cognitive Science." *Behavioral and Brain Sciences* 21:615-665.

Vera, Alonso H., and Herbert A. Simon. 1993. "Situated Action: A Symbolic Interpretation." *Cognitive Science* 17:7-48.

Weiskopf, Daniel. 2004. "The Place of Time in Cognition." *British Journal for the Philosophy of Science* 55:87-105.