What kind of machine is the mind?

Joscha Bach[1] and Mario Verdicchio[2*]
[1] Humboldt-University of Berlin, Berlin School of Mind and Brain, Berlin, Germany
[2] Università degli Studi di Bergamo, Dipartimento di Ingegneria dell'Informazione, Bergamo, Italy
joscha.bach@hu-berlin.de, mario.verdicchio@unibg.it

Abstract

Many computational modeling approaches of the mind seem to be characterized by an implicit strong physicalism, which frequently leads to confusion in philosophy of AI. This work aims at pointing out some fundamental aspects of this problem, both with respect to the relation between epistemological computationalism and physical realization, and the view of symbol manipulation as constrained computation.

1 Introduction

According to Alan Turing, the question of whether machines can think must begin by properly defining the notions of "machine" and "thinking" (Turing, 1950). Turing proceeded to address these questions with the famous Turing test. Here, we want to get closer to these definitions from a different direction. The possibility of Artificial Intelligence, in the strong version aimed at by its pioneers, hinges not only on technical feasibility and the right paradigm for its realization: we will address some of the epistemological issues involved in the "mind as machine" concept, namely, its stance with regard to computationalism, as opposed to non-functionalist physicalism. (We will not look at explicitly anti-mechanist positions here.) When Newell and Simon introduced their Physical Symbol System Hypothesis (PSSH), it seemingly implied three statements (Newell & Simon, 1976):

a) Cognition is information processing (computation), and is equivalent to symbol manipulation.
b) Its description can be given by specifying a possible implementation of an information processing system with a systemic boundary to an environment that supplies information.
c) The information processing system can and must be physically realized.

The PSSH has been (explicitly or implicitly) at the heart of most of the computational modeling approaches of the mind, but within the philosophical discussion, the different connotation of symbols, and the implied strong physicalism gave rise to frequent misunderstandings. Here, we want to point out some aspects of the root of this confusion, both with respect to physical realization versus epistemological computationalism, and symbol manipulation as constrained computation.

2 Troubles with physicalism

Here, we take strong physicalism to be the position of the epistemological primacy of physical interactions. There might be little doubt that minds are caused by matter and energy and the physical laws that govern them. According to the current theories of physics, then, minds are ultimately a consequence of quanta and their interactions. These interactions can be captured in computational

[*] partially supported by the Italian Ministry of Education, University, and Research in the framework of the PRIN project Gatecom.

theories and models, but, from the physicalist perspective, such computations (i.e. the complete formulation of regularities in the observables) are only a convenient way to describe what exists and happens in the universe.

Some philosophers maintain that it is possible to be a physicalist without accepting functionalism, i.e. the mind would be some kind of physical phenomenon or machine, but cannot be described computationally. Let us call this 'anti-computationalist physicalism'.

What we would call 'weak computationalism of the mind' is compatible with strong physicalism. It maintains that while minds are physical machines, they can be described in terms of functional properties with respect to information processing, using computational models.

Epistemological computationalism (Wolfram, 2002) (Margolus, 2003) denies strong physicalism in favor of a universal primacy of information, on the grounds that all possible observations of the universe do not impart matter or energy, but information (i.e. discernible differences). The description of all conceivable regularities in the observed data is necessarily and sufficiently computational. Epistemological computationalism does not deny the validity of physical descriptions, but takes an anti-realist stance and maintains that the physical universe is a possible computational theory that encodes the observed data. For minds and worlds to exist, computation is necessary and sufficient; the notion of an ultimate physical substrate is a superfluous metaphysical concept, since it can never be anchored in an immediate empirical observation. The exclusion of empirically inconvertible assumptions from physicalism leads with some inevitability to epistemological computationalism.

While the differences between strong physicalism and epistemological computationalism are only on a metaphysical domain, there seems to be no such thing as an epi-belief: In epistemological computationalism, it is impossible to derive a non-computational concept of the mind, while some physicalists see its locus either in a non-functional substance property of the physical universe (Putnam, 1983), or in an interactional relationship that surpasses a computational characterization (radical enactivism (Di Paolo, 2009)).

3 Constrained models of computation

Symbol manipulation in the sense of the PSSH does not carry the same connotations as symbol use in the context of semiotics, but refers to computation in the sense of a Turing Machine (TM), i.e. states that a computational system is a necessary and sufficient precondition for a mind. However, this statement requires further qualification.

The TM (Turing, 1965) is a mathematical principle that happens to define the class of Turing-computable functions of degree 0 (which, according to the Church-Turing thesis, encompasses all computable functions). The TM also happens to be a very intuitive description, so even though there are many other ways of capturing computable functions (for instance, the Lambda calculus, which was developed by Alonzo Church roughly at the same time (Church, 1936)), it also became a metaphor for computation within the context of philosophical discussion. This is somewhat unfortunate, because the TM evokes an intuition of a somewhat clunky, even if hypothetical contraption, consisting of tape, wheels, clicking levers, and sensor and actuator mechanisms. The idea that human brains might be equivalent realizations of TMs sounds implausible, even offensive to some philosophers (Searle, 1996).

Traditionally, mathematicians used to call Turing-computable functions 'effectively' computable functions, which does not mean the same as 'efficiently' computable: to be computable by a TM does not imply any constraint on the number of necessary steps, besides being finite. Any physical realization of a TM would have to deal with additional constraints (for instance, a limit on the number of execution steps—the execution time—and thereby, or on top of that, on the length of the tape—the available memory).

It is important to realize that other ways of defining computable functions might not only evoke other metaphors in the minds of philosophers, but, more importantly, their implementations may react differently to constraints, even though the set of functions that can effectively be computed by their unconstrained version is exactly the same as in the case of the TM.

In AI, and in cognitive science, we are looking at a class of systems that is not well described by an unconstrained TM (Sloman, 2002), but at very limited approximations. These limits prevent human minds from performing arbitrary calculations and explain why there are problems that can be solved by a personal computer, but not by an unaugmented human. In applied computer science, we are not looking at unconstrained TMs either, but at differently limited approximations, which explains why even many efficiently computable functions cannot be performed by a practically realizable computer (for instance, because they might require too much memory and more execution time than afforded by the duration of the experiment). Computation can be defined in arbitrarily many ways by using a practical implementation of a personal computer and removing its constraints on execution speed and memory. (In a fundamental sense, this extends to quantum computers (Lloyd, 2006); any function that can be calculated by a quantum computer can be effectively, even if not always efficiently calculated by a classical computer.) For the human brain, this is much less clear, because it is not obvious which constraints must be lifted to turn it into a universal model for computation.

One of the many ways to define computability consists in the use of a recurrent neural network, an Activation Spreading Machine (ASM). Such a network consists of a set of activation bearing nodes that are connected by weighted directional links, along which the activation is propagated via an effectively computable transfer function.

All Turing-computable functions can be performed by an ASM (Hyötyniemi, 1996). It can also be shown that for rational weights and activations, the ASM can be computed by a TM and is thus exactly equivalent in its computational power. As an aside, if the link weights were real-valued (i.e. if they had infinite precision), ASMs would be more powerful than TMs, i.e., hypercomputational (Cabessa & Siegelmann, 2012). This is of no relevance, however, because the transfer function would no longer be efficiently computable (just as the physical universe does not afford an infinite resolution to which computational systems can be realized). Intuitively put, real-valued activation spreading would require the transfer of infinite amounts of information and can be specified (in the same way as we can specify a procedure that squares a circle), but not performed (Davis, 2006).

Again, in connectionist modeling, we are not concerned with unconstrained (Turing-computational) recurrent networks, but with constraints on the number of nodes, links and transition operations. This does not only limit the set of functions that can be efficiently computed by such networks, but it limits them in a different way than limiting the execution speed and memory in a given personal computer: the set of functions that can be computed by a constrained computational system differs in accordance with the effectively imposed constraints, that is, with the structure of the system itself.

The metaphor of the constrained ASM might be of greater utility than a constrained TM to capture the information processing performed by a cognitive system, but the actual constraints of a human nervous system go much further than that. For instance, information is not stored in the brain in an ubiquitously available form, as reflected in the difference between procedural and declarative knowledge (Ryle, 1946) (Devitt, 2011). Nervous systems (among other things) are additionally characterized by a very specific architecture, which marks the primary field of research for unified theories of cognition (Newell, 1987) (Duch, Oentaryo, & Pasquier, 2008). Thus, while the PSSH's symbol manipulation requirement is appropriate and sufficient, it does not specify the necessary limitations.

Practical research in Artificial Intelligence is and will likely remain unconcerned with the aforementioned troubles with physicalism, and the underspecification of the PSSH's symbol requirement, since its object is the discovery and exploration of the right sets of architectural and structural constraints for suitably defined computational systems. The philosophy of AI, however,

reflects explicit and implicit assumptions within these domains in a variety of positions that are difficult or even impossible to reconcile.

References

Cabessa, J., & Siegelmann, H. (2012). The Computational Power of Interactive Recurrent Neural Networks. *Neural Computation , 24* (4), 996-1019.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics , 58*, 345-363.

Davis, M. (2006). Why There Is No Such Discipline As Hypercomputation. *Applied Mathematics and Computation , 178* (1), 4-7.

Devitt, M. (2011). Methodology and the Nature of Knowing How. *Journal of Philosophy , 108* (4), 205-218.

Di Paolo, E. A. (2009). Extended Life. *Topoi , 28*, 9-21.

Duch, W., Oentaryo, R., & Pasquier, M. (2008). Cognitive Architectures: Where do we go from here? In P. Wang, B. Goertzel, & S. Franklin (Ed.), *Proceedings of the 2008 conference on Artificial General Intelligence* (pp. 122-136). Amsterdam: IOS Press.

Hyötyniemi, H. (1996). Turing Machines are Recurrent Neural Networks. In A. Jalander, T. Honkela, & M. Jakobsson (Ed.), *Proceedings of STeP '96* (pp. 13-24). Vaasa: Finnish Artificial Intellgence Society.

Lloyd, S. (2006). *Programming the Unvierse: From the Big Bang to Quantum Computers.* London, UK: Jonathan Cape.

Margolus, N. (2003). Looking at Nature as a Computer. *International Journal of Theoretical Physics , 42* (2), 309-327.

Newell, A. (1987). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. (1976). Computer Science ad Empirical Enquiry: Symbols and Search. *Communications of the ACM , 19*, 113-126.

Putnam, H. (1983). Why Reason Can't Be Naturalized. *Synthese , 52* (1), 3-23.

Ryle, G. (1946). Knowing How and Knowing That. *Proceedings of the Aristotelian Society , 46*, 1-16.

Searle, J. R. (1996). *The Ridiscovery of the Mind.* Cambridge, MA: MIT Press.

Sloman, A. (2002). The Irrelevance of Turing Machines to AI. In M. Scheutz, *Computationalism: New Directions.* Cambridge, MA: MIT Press.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind , 59*, 433-460.

Turing, A. M. (1965). On Computable Numbers, with an Application to the Entscheidungsproblem. In M. Davis, *The Undecisable: Basic Papers on Undecidable Propositions, Undecidable Problems and Computable Functions.* New York, NY: Raven Press.

Wolfram, S. (2002). *A New Kind of Science.* Champaign, IL: Wolfram Media.