

Simulation in computational neuroscience
Liz Irvine (Cardiff)
Word count: 5187

1. Introduction

In computational neuroscience the term ‘model’ is often used to capture simulations as well, but simulations have distinct features, and distinct epistemic uses. I follow Parker’s (2009) definition of a simulation as “a time-ordered sequence of states that serves as a representation of some other time-ordered sequence of states [e.g. of the target system]” (p. 486). From a starting state at t_0 , the program uses an algorithm to calculate the state of the system at t_1 , and given these values, the program can calculate the next state of the system at t_2 , and so on.

Simulations are used in cognitive neuroscience for a variety of reasons (see Connectionism Chapter for more detailed discussion). One of interest here is that simulation offers a scaffolded way to investigate the internal functioning of a complex system without necessarily needing a lot of experimental data about that internal functioning. That is, unlike some standard approaches to simulation in other disciplines (Winsberg, 2010), simulations in neuroscience are often built without much background knowledge about the system they are (sometimes not very closely) trying to simulate.

A description of this is Corrado and Doya’s ‘model-in-the-middle’ approach (Corrado & Doya, 2007; Corrado et al. 2009). They argue that investigating the process of decision making using standard methods in systems neuroscience is massively underconstrained. The standard method involves manipulating some variable of interest (e.g. visual input) and seeing what effect this has on other variables (e.g. motor action), and from there, trying to abstract away an account of decision making. With complex systems where it is not necessarily obvious what the relevant variables are or how one would expect them to interact, this approach is deeply problematic.

Instead, with the ‘model-in-the-middle’ approach researchers make a reasonable guess about how the system works, build a simulation based on this guess, and then see if the decision variables and outputs in the simulation correlate with internal signals and outputs experimental subjects. If the simulation ‘fits’ the experimental data in various ways, then this provides evidence for the hypotheses instantiated in the simulation. In computational neuroscience, these hypotheses can concern abstract computational processes as well as mechanistic hypotheses about how certain computations are implemented.

In Section 2 below, I evaluate different types and different approaches to simulation in computational neuroscience, including small and large scale simulations, and bottom-up and top-down approaches. Section 3 outlines what kind of explanations simulations can (and how), including a brief discussion of mechanistic vs. computational explanations. Section 4 briefly outlines some interesting benefits of computational simulations in this field to computer science, in the form of neuromorphic devices.

2. Types of Simulation: Large and small, bottom-up vs. top-down

2.1 Small scale simulation

Simulations are a key way to investigate brain function, and most simulations focus on small scale 'local circuits' to investigate how a specific function is realised. These include simulations of sensory processing, decision making, motor control, and different types of learning and memory.

For example, reinforcement learning algorithms have been used to simulate learning and decision making. Reinforcement learning describe how agents should act over time in an uncertain environment in order to maximize reward. Agents first recognize the present state of the environment, and then calculate and predict which action will deliver the highest expected reward using previous experience. The agent then performs the action, compares the predicted reward with the actual reward to get the 'reward prediction error', and then uses this to update its estimate of the values of particular actions. Reinforcement learning simulations can accurately predict real behavior of performing animals, and dopamine neurons in the basal ganglia circuit fire according to simulated reward prediction errors, suggesting that the simulations are fairly accurate depictions of computational processes involved in learning and decision making (Corrado et al. 2009).

A more widespread type of simulation in computational neuroscience comes from the connectionist tradition. Briefly, connectionist simulations use domain-general principles of parallel and recurrent processing and a set of learning rules to update connection weights between very simplified units, which represent neurons or groups of neurons. The same principles can be used to model a variety of specific behaviours, including word learning, perception, memory, concept formation and reasoning (see Connectionism Chapter for more detail, and other chapters in this section for alternative approaches).

In a review of the use of simulations in computational neuroscience, Sompolinsky (2014) states that most theories derived from simulation (particular obviously in the connectionist tradition) fall within a 'network paradigm'. Here, the functional capacity of a local circuit can be mostly explained by (relatively) simple rules that govern the activity of (relatively) simple and uniform simulated neurons connected via synaptic weights. That is, it looks like adding extra levels of complexity and biological realism to these highly simplified and idealized simulated local circuits adds little in the way of predictive or explanatory power.

However, Sompolinsky argues that these simulations of local circuits are unlikely to scale up to offer explanations of higher level cognitive abilities for a number of reasons. One is that the network paradigm ignores the chemical soup that neurons function within which modulate and are modulated by electrical activity, and they also ignore the complexity of different types of neurons and their patterns of connectivity. It seems likely that these biological details do make some difference to brain function (in any case, it would be puzzling if they do not). There are two other related reasons. The network paradigm ignores the active nature of perception; of sensory and cognitive systems seeking out information rather than being passive receivers of it. This is related to the more general point that focusing on local circuits and specific functions fails to capture the massive interconnectivity

across the brain, and so the ways in which local circuits together contribute to the realization of many different functions.

Two different approaches to simulation that attempt to address some of these problems are reviewed below. One is a 'bottom-up' approach which attempts to address the problem of biological realism within simulations. The second is a 'top-down' approach which attempts to address the issue of whole-brain function across distributed local circuits.

2.2 Brain simulations: Bottom-up

The Blue Brain Project (<http://bluebrain.epfl.ch/>) is an example of a 'bottom up' or data-driven approach to large-scale brain simulation, which has so far focused on building a biologically realistic simulation of a juvenile rat somatosensory cortex (though the eventual aim is to construct a whole brain). The simulation is based on the structure provided by neocortical columns, and closely simulates neural type, structure, position, connectivity, and electrical activity. The simulation is generated using computationally demanding algorithms, based on large databases of experimental data. The only features of the simulation that are programmed in are low-level features of neurons, their activity, and their connections, so whatever the simulation does from then on emerges from the total activity of the connected simulated neurons. This is what makes it a bottom-up simulation; the simulation is constructed from experimental evidence concerning low-level features of neural systems only, with no constraints from higher level theory.

There are several reasons why one might take a bottom-up approach to brain simulation, and what one can do with such simulations. First, linked to the discussion above, such detailed simulations allow a better understanding of the roles of different types of neurons, layers, and connections between them, and so perhaps explain why there is such much complexity at this level. This is not possible with less biologically realistic models that use much simpler and more uniform simulated neurons.

Second, highly detailed models can make it possible to intervene on a (simulated) neural system in very specific ways, which can be used to corroborate and extend experimental findings. For example, Markram et al. (2015) varied tonic depolarization and calcium levels in a virtual brain slice (*in silico* experimentation), and found regular variations in neural firing synchrony. This was also found in *in vitro* experiments, and so this finding across both brain slices and simulated neocortical columns suggests that they had identified a "highly reproducible phenomenon, robust to biological and statistical variations in parameters [related to the microcircuit]" (p. 472). That is, the convergence of *in vitro* and *in silico* experimental data suggests that the findings in either condition were not artefactual, and hold over a wide range of neural microcircuits.

There are downsides however to such simulations. First, their very complexity makes it technologically difficult (to say the least) to simulate more than a tiny proportion of a brain at any one time. While this makes it possible to answer some questions about brain function, it misses out questions about overall brain function and control of behavior. Second, and relatedly, while clearly paying more attention to biological features of the brain, such data-driven models do not make contact with higher level structural and

functional characterisations of the brain, which again makes it difficult to connect these simulations to other levels of investigation. Third, their very complexity can also make it difficult to understand or explain exactly how neural phenomena unfold. The very practice of ‘observing’ the simulation and making sense of it is a massive technological and epistemic project in its own right.

2.3 Brain simulations: Top-down

Another approach is a ‘top-down’ or hypothesis driven strategy. Here, simulations are not used to replicate the brain in vast detail, which are then manipulated to see how different parameters affect neural activity. Instead, the simulation embodies hypotheses about brain function, so ‘running’ the simulation and comparing its outputs and internal behavior to experimental data directly tests those hypotheses. To change the hypothesis, one changes the simulation, which can then be tested again.

Spaun (Eliasmith et al., 2012) is a well-known example of a top-down approach to brain simulation. Spaun uses functional accounts of the roles of different brain areas (e.g. visual, motor, pre-frontal areas) to assign connection weights within simple simulated neural structures that represent these different brain areas. A key feature of Spaun is its three compression hierarchies over the domains of vision, action and working memory. Information is compressed throughout the visual hierarchy as image-based encoding is transformed to feature-based encoding. Conversely, motor decisions are gradually decompressed from the selection of an action plan into the kind of high-dimensional encoding that is required for control of a complex arm. Given the neurally realized functional characteristics of the different simulated brain areas, and these compression/decompression routes, Spaun can link visual input (via an eye) to external behavior (via a physically modelled arm), and is capable of performing eight different visual, motor and cognitive tasks. These features make Spaun an example of a ‘top down’ simulation; what drives the construction of Spaun are experimentally informed hypotheses concerning high level functional characterisations of brain areas.

Spaun is therefore rather different to the simulations provided by the Blue Brain project: Spaun is more coarse-grained, leaving out much biological detail, but this makes it possible to simulate the interactions between a range of brain areas and how they together to perform a range of different tasks. Spaun is also aimed at answering rather different questions. Instead of trying to get an understanding of how fine-grained details of neural structures affect their functionality, the idea is to see how interconnected and functionally specified groups of neurons can generate flexible behaviour. That is, the main aim of developing and testing a simulation like Spaun is to “propose a unified set of neural mechanisms able to perform [a range of tasks].” (Eliasmith et al. 2010, p. 1204).

This highlights the advantages of top-down simulations. First, they form complex and powerful tests of high level hypotheses about overall brain function. Second, existing bodies of behavioral data and theory can be used to inform and test these hypotheses, and so they do not require extensive knowledge of neural structures and activity (which in some cases, we do not yet have). Third, their (relative) simplicity makes it possible to track the sources of inconsistencies between simulated and real data, so that it is possible to understand how

and why the simulation is working (or failing to). Having epistemic access to the workings of the simulation is a major advantage of such an approach.

However, there are also some downsides. First, there is the obvious fact that Spaun is not biologically complete or particularly biologically realistic. There may be some features of neural activity or connectivity that are important for enabling flexible behavior that the simulation fails to capture. Second, using Spaun it is not possible to investigate how the brain develops over time (which the Blue Brain project can, in principle), and although there is some task flexibility Spaun is by no means as flexible as a 'real' human brain. Its findings then can only be partial.

2.4 Different approaches, different aims

There has been a vigorous debate on the appropriateness of both of these approaches to brain simulation (see e.g. Chi, 2016 for summary), but *if* the aim is to understand how brains generate complex behavior, then there are arguments that using (relatively) simple simulations to test hypotheses directly is an efficient way of generating knowledge.

First, Spaun exemplifies the tendency in cognitive neuroscience toward top-down simulation, in which simpler and less biologically realistic simulations are used to test bundles of hypotheses about brain function. This is arguably an approach that works well within the constraints of cognitive neuroscience. These constraints include on the one hand fairly easily accessible and plentiful behavioural data, and on the other hand, fairly inaccessible and less common neural data/recordings, all concerning an incredibly complex system. Getting the experimental data to even start building a simulation as complex as those developed in the Blue Brain project is a massive and on-going undertaking. Top-down simulation makes it possible to start answering questions and developing theories given what is available.

Second, hypotheses concerning the generation of complex behavior will themselves be fairly complex. Building complex hypotheses into a simulation makes it possible to test them in a way that no other method can offer. In particular, it makes it possible to test bundles of hypotheses together, and (somewhat easily) to change and re-test updated hypotheses. It is also possible to use large scale simulations to investigate the potential power of general computational principles or general organizational structures in explaining a range of behaviours.

Relatedly, it is (relatively) easier to understand a simple, top-down simulation than a complex bottom-up simulation, both when it seems to work, and perhaps more importantly when it fails to generate realistic looking outputs. Understanding a simulation like Spaun is easier both because the simulation contains simplifications (so there are fewer variables and activities to record and analyse), and because of its top-down nature. Researchers building the simulation have a good idea of how it fits together at the functional level precisely because this is a central feature of the simulation. Much of its behavior is emergent and unpredictable, but the general principles that are supposed to make it work are understood.

Testing hypotheses in a complex bottom-up simulation however is harder because both it is more complex (creating huge amounts of data to analyse) and because it is constructed 'blind' to functional or system-level properties, which themselves emerge from the low-level activity of the programmed parts. In this case, if the simulation fails to generate realistic looking outputs, it is far from obvious how to pinpoint the most causally relevant factors involved. There are techniques for getting a better understanding of some kinds of neural networks (e.g. deep visualization of deep neural nets Yosinski et al. 2015), but techniques for visualizing Blue Brain are still under construction (Abdellah et al. 2015).

There are also questions about the epistemic value of adding more complexity or realism to simulations. As Winsberg (2010) discusses in depth, building simulations is a skill, and the epistemic value of a simulation depends in large part on the knowledge and skill sets of those building it. A researcher will (ideally) have a good idea what can be left out of a simulation, what is crucial to include in detail, what short cuts can be taken in the programming, how to represent parts and activities in the simulation, and so on, given the specific epistemic aims of the simulation. The simplifications, idealisations, and so on in Spaun are directed at specific goals (i.e. testing specific hypotheses within certain boundaries), and given these goals, it is reasonable to make certain widespread and sometimes extreme simplifications. These will affect the sort of conclusions can be drawn from the simulation, but this in no way renders it epistemically inert. As McClelland (2009) writes: "Simplification is essential [for understanding], but comes at a cost, and real understanding depends in part on understanding the effects of the simplification" (p. 18). This is echoed in Trappenberg's (2009) textbook on computational neuroscience: "Modelling the brain is not a contest in recreating the brain in all its details on a computer. It is questionable how much more we would comprehend the functionality of the brain with such complex and detailed models" (p. 7). If understanding is the aim of simulation, then hypothesis-driven simulations should be favoured.

Clearly though, there are idealisations, simplifications, and some things just left out of the Blue Brain simulations too. But as the purpose of this type of simulation is more of a multi-purpose exploratory and experimental tool, it appears to be less reasonable to make widespread simplifications, because it is not clear what specific properties might be relevant to a given behavior, and the generators of such a simulation presumably do not want to limit the kinds of conclusions that can be drawn from it in advance. Yet this may be the problem: a multi-purpose near replica of a complex system is almost as epistemically opaque as the original system. Constructing hypothesis-driven simulations makes it possible to simplify and idealise in specific and well-reasoned ways, and this in turn makes the performance of the simulation easier to understand, even if this limits the kinds of conclusions that can be drawn from it.

All this though is only relevant *if* the intended goal is to understand how brains generate complex behavior. If the goal is to get a better understanding of questions in cellular/molecular neuroscience, or to generate an *in silico* experimental suite, then very different standards apply. There are other epistemic outputs from the construction of the Blue Brain simulations too: they are dependent on vast amounts of recorded information about neural structure and activity, which demands new experimental work.

Building on this, the next section assesses the kinds of explanations that can be constructed by using computer simulations.

3. Simulations and computational explanations

It is natural to think that computer simulations in computational neuroscience generate computational explanations of the brain. However, things are not always so straightforward. First, some philosophers working within the mechanistic approach to explanation have argued that computational explanations, as (somewhat) autonomous computational or functional level explanations, are not explanatory at all unless they fit into the mechanistic framework (Kaplan, 2011; Piccinini & Craver, 2011).

Briefly, a mechanistic explanation of a phenomenon P is an explanation of how a system of spatially and temporally organized parts and activities together produce P (for review see Illari & Williamson, 2012). In particular, Kaplan (2011) has argued that to be explanatory at all, computational explanations must fit into the mechanistic framework by satisfying a 'model-to-mechanism-mapping' (3M) requirement:

“(3M) A model [or explanatory simulation] of a target phenomenon explains that phenomenon to the extent that (a) the variables in the model correspond to identifiable components, activities, and organizational features of the target mechanism that produces, maintains, or underlies the phenomenon, and (b) the (perhaps mathematical) dependencies posited among these (perhaps mathematical) variables in the model correspond to causal relations among the components of the target mechanism.” (Kaplan 2011, p. 347)

In this case, a simulation of purely computational processes that does not include any detail of how those processes are implemented in the brain may be predictive, but it is not explanatory. Further, a model or simulation that is more complete is better than one that is less complete.

This obviously conflicts with the points about simplification and understanding raised above in discussion of Spaun, and blocks the possibility of non-mechanistically focused computational explanations of neural phenomena. These are taken in turn below.

3.1 Completeness

In contrast with Kaplan's suggestion that a more detailed mechanistic model is better than a less detailed one, and in line with the statements from neuroscientists themselves, Milkowski (2016) has argued that idealization is an important feature of explanations gained from simulations. This is true not merely for practical reasons, but for reasons core to the mechanistic framework.

A core part of the mechanistic program is that mechanisms are mechanisms *for* generating a particular phenomenon, and so are individuated by functional capacities and/or explanatory targets. As a consequence of this, the only parts (and levels) that are necessary to include in a mechanism, and a mechanistic explanation are those that are causally relevant to the

explanatory target. In this case, idealisations and simplifications can be mechanistically appropriate, and indeed are regularly found in mechanistic explanations¹.

In this case, Spaun can be seen as an idealized ‘how-plausibly’ mechanistic model of (some aspects of) cognitive flexibility. That is, it contains organized parts and activities that plausibly generate some degree of cognitive flexibility (at least the degree involved in shifting across the eight tasks that it can perform). This echoes the ideas outlined above: given the particular epistemic goal that Spaun is aimed at, its lack of biological realism is entirely acceptable.² If Spaun were aimed at a different explanatory target, it may not be plausible or sufficiently mechanistic, but this is the nature of mechanistic explanations in general.

With respect to the Blue Brain project, Milkowski (2016) there is no obvious explanatory target, but rather a series of questions that the Blue Brain project as a whole targets. What the Blue Brain amounts to therefore is not a mechanistic model capable of providing a mechanistic explanation, but a system for generating mechanistic simulations to test hypotheses. This echoes the points made above; that the Blue Brain project seems aimed at rather different epistemic aims than Spaun and other top-down simulations. Perhaps more interestingly here, the lack of constraints from higher levels of description in generating the Blue Brain simulations make them mechanistically deficient: mechanisms are comprised of multiple levels of parts and activities, and the strict bottom-up approach taken in the Blue Brain means that relevant levels are missed out.

One aspect of this kind of simulation that a Kaplan-style approach gets right however is that while a simulation may generate outputs comparable to experimental data, it is not possible to claim that cognition actually works in the way suggested in the simulation; the simulation may or may not capture internal processes accurately. Only by specifying how the simulation is implemented, and then checking whether these details are in fact realized can this kind of simulation be said to accurately describe cognitive processes. This limits Spaun to giving ‘how-plausibly’ explanations, but simulationists are generally open about this limitation, as well as it’s explanatory limits.

3.2 Non-mechanistic computational explanations

There is clearly not enough space here to fully evaluate the mechanistic framework with respect to computational neuroscience (see other Chapters in this section, and Chirimuuta, 2014; Irvine, 2015; Milkowski, 2013; Piccinini, 2015; Rusanen & Lappi, 2007; Serban, 2015; Shagrir, 2010). However, looking to the construction of at least top-down simulations in computational neuroscience can shift the balance towards accepting computational explanations as at least somewhat distinct from mechanistic explanation.

¹ As a special case of this, Levy and Bechtel (2013) outline how focusing on causal connectivity and leaving out aspects of mechanistic structure can also generate appropriate abstract mechanistic explanations.

² It also appears that adding detail leads to no greater predictive power either; replacing simple neurons in Spaun’s frontal cortex with more complex and biological realistic ones does not improve Spaun’s performance (Chi 2016).

An initial move in this direction is to look to the kind of explanations provided by early connectionist simulations. Here, the simulations themselves were not deemed to be explanatory. Instead, simulations were used to test the application of general computational principles to cognitive processes, and it was these general principles that were seen as explanatory (McClelland, Rumelhart, Group, 1987). A possible response to this is that the simulations provided sketches of possible mechanisms, so were perhaps minimally mechanistically explanatory, but this sharply contrasts with the expressed aims and conclusions from researchers at the time: “We are striving for theories that explain phenomena in specific domains...in terms of more general principles concerning the representation of knowledge, its acquisition and use....Modelling [simulating], on this view, is a way to both identify general principles and assess their validity” (Mark Seidenberg & James McClelland, manus., pp. 3-4).

Looking more closely to the overall practice of using and constructing top-down simulations in cognitive neuroscience, it is apparent that the hypotheses instantiated in simulations are often based on computational templates borrowed from other disciplines (e.g. from statistics, economics, thermodynamics, AI). Computational templates are sets of equations or computational methods that are interpreted to generate models of particular phenomena, which here are often learning algorithms or general computational structures (Humphreys, 2002, 2004). These templates provide the ‘general principles’ mentioned above.

According to Humphreys, in order to use a template to construct a model or simulation, a ‘correction set’ is generated that de-idealizes and de-abstracts parts of the template, and changes the constraints and approximations of the template for use in a particular domain. The correction set is based on relevant background empirical, theoretical, and practical knowledge, which can include single-cell recordings, patterns of behavioural data, functional analyses of brain areas, as well as analyses of time and energy constraints that limit what computational processes can be implemented. In generating the correction set, researchers become aware of which parts of the simulation are likely to represent relevant parts of a target phenomenon for a specific aim, and which are not (Humphreys 2002, pp. 76–81). Simulations, either in part or in whole, are tested in multiple ways throughout an on-going process of construction and refinement, which in turn updates the correction set.

There are several related reasons why templates are commonly used in computational neuroscience. The first is that applying general computational templates to explain (neural) computational processes is epistemically efficient. Computational templates provide a ready made set of algorithms and methods from which to build specific hypotheses about brain function. Relatedly, templates are tractable and usually well understood, which both makes it easier to apply them in new cases, and to tell whether their outputs will likely match behavioural data (Knuuttila, 2011). It also seems reasonable to assume that the brain is regulated by a set of common computational processes (e.g. compression algorithms, learning rules, etc), in which case a template used to develop a simulation of visual perception may well be useful in developing a simulation of action or audition. Tying all these together, if the brain is seen as solving computational problems, then pre-existing computational templates developed in other fields that are known to solve similar problems

can be imported in and tested in a fairly straightforward way (for more discussion see Irvine, 2014).

At least some simulations function as ‘proof of concept’ that something like a particular template and the computational principles that it describes, suitably adapted by a correction set, captures the core computational principles that make the system able to do what it does. A simulation can test whether low-level decisions can be made in short time frames because they are governed by algorithms of sequential analysis (Gold & Shadlen, 2007), or that the brain can maximize reward in uncertain environments because it relies on reinforcement learning (Corrado et al. 2009), or remain fairly functional in the face of damage because it uses parallel processing over distributed representations (McClelland, Rumelhart, Group, 1987). That is, these simulations explain how a cognitive system solves a computational problem (Shagrir 2010).

Importantly, these explanations need not include any detail about how computational principles are implemented. The principles need to be implemented in the simulation in order to test whether they apply to a cognitive system, but having shown that the simulation fits with experimental data, it is the principles that are explanatory. That they are used at all in a cognitive system can answer some important questions about cognition; how they are implemented answers others.

It is also possible that a simulation can offer both types of explanation at the same time (computational and mechanistic), and the more realistic the simulation, the more ways there are of checking that it does in fact describe real processes or merely has high predictive power. Yet this does not mean that one type of explanation can be reduced to another (for more discussion, see references at the start of this section).

4. Contributions to computer science

Computer simulations, particularly those that are developed from pre-existing computational templates, make use of expertise from computer science in order to explain cognitive phenomena. However, the development of computer simulations can also contribute positively to computer science in several ways, reviewed below.

The Blue Brain project is a particularly good example of neuroscientists working with computer scientists and computer engineers (e.g. IBM, who supply the BlueGene supercomputer used in the project). Within the Blue Brain project, it is not only running the simulations that requires BlueGene, but also constructing the simulations in the first place. For example, Markram (2006) notes that the algorithm that determines the structural positions of neurons and connections between axons and dendrites is more computationally intense than running the resulting simulation. Complex software programs are required just to get the simulation to run across different processors. Writing from 2006, Markram stated that a million-fold increase of computation power was needed to simulate a whole brain at the level of detail then used to simulate a single neocortical column, but that algorithmic and hardware efficiencies could bring that down to manageable levels.

Colombo (2016) suggests that one possible way of developing efficient enough computers

to run a detailed whole-brain simulation is actually to learn from existing simulations, and how they trade off features like execution time, processing capacity and energy consumption, compared to the brains they are trying to simulate. This is significant as energy consumption and heat production are pressing problems in moving beyond current levels of computational power. Taking inspiration from the difficulties posed by large scale computer simulations of brain, and the properties of real brains, can be used to generate neuromorphic devices; computational devices that are modelled on brains and how they manage high computational efficiency. For example, the SpiNNaker chip and system (<http://apt.cs.manchester.ac.uk/projects/SpiNNaker/project/>) were developed in tandem with the Blue Brain project, and which constitute a massively parallel architecture composed of billions of spiking (neuron-like) units. This can be used both to simulate neural networks but also has applications in robotics and computer science more generally. In addition to the differences noted above, large-scale bottom-up simulations therefore can have significant impacts outside testing hypotheses about specific cognitive processes.

Bibliography

- Abdellah, M., Bilgili, A., Eilemann, S., Markram, H., & Schürmann, F. (2015). Physically-based in silico light sheet microscopy for visualizing fluorescent brain models. *BMC Bioinformatics*, 16(11), 1.
- Chi, K. R. (2016). Neural modelling: Abstractions of the mind. *Nature*, 531(7592), S16–S17.
- Chirimuuta, M. (2014). Minimal models and canonical neural computations: The distinctness of computational explanation in neuroscience. *Synthese*, 191(2), 127–153.
- Colombo, M. (2016). Why build a virtual brain? Large-scale neural simulations as jump start for cognitive computing. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–10.
- Corrado, G., & Doya, K. (2007). Understanding neural coding through the model-based analysis of decision making. *The Journal of Neuroscience*: 27(31), 8178–8180.
- Corrado, G. S., Sugrue, L. P., Brown, J. R., & Newsome, W. T. (2009). The trouble with choice: studying decision variables in the brain. *Neuroeconomics: Decision Making and the Brain*, 463–480.
- Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, 338(6111), 1202–1205.
- Gold, J. I., & Shadlen, M. N. (2007). The neural basis of decision making. *Annu. Rev. Neurosci.*, 30, 535–574.
- Humphreys, P. (2002). Computational models. *Philosophy of Science*, 69(S3), S1–S11.
- Humphreys, P. (2004). *Extending ourselves: Computational science, empiricism, and scientific method*. Oxford University Press.
- Illari, P. M., & Williamson, J. (2012). What is a mechanism? Thinking about mechanisms across the sciences. *European Journal for Philosophy of Science*, 2(1), 119–135.
- Irvine, E. (2014). Model-based theorizing in cognitive neuroscience. *The British Journal for the Philosophy of Science*, axu034.
- Irvine, E. (2015). Models, robustness, and non-causal explanation: a foray into cognitive science and biology. *Synthese*, 192(12), 3943–3959.
- Kaplan, D. M. (2011). Explanation and description in computational neuroscience. *Synthese*, 183(3), 339–373.

- Knuuttila, T. (2011). Modelling and representing: An artefactual approach to model-based representation. *Studies in History and Philosophy of Science Part A*, 42(2), 262–271.
- Levy, A., & Bechtel, W. (2013). Abstraction and the organization of mechanisms. *Philosophy of Science*, 80(2), 241–261.
- Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, 7(2), 153–160.
- Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2), 456–492.
- Mark Seidenberg, & James McClelland. (manus., 1992). *Connectionist models and explanatory theories in cognition*. Available from: http://stanford.edu/~jlmcc/papers/PublicationFiles/MISC_manuscripts_Dates_Unknown/SeidenbergMcClellandXXConnectionistModelsandExplanatoryTheoriesinCognition.pdf
- McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, 1(1), 11–38.
- McClelland, J. L., Rumelhart, D. E., Group, P. R., & others. (1987). *Parallel distributed processing* (Vol. 2). MIT press Cambridge, MA.
- Milkowski, M. (2013). *Explaining the computational mind*. MIT Press.
- Milkowski, M. (2016). Explanatory completeness and idealization in large brain simulations: a mechanistic perspective. *Synthese*, 193(5), 1457–1478.
- Parker, W. S. (2009). Does matter really matter? Computer simulations, experiments, and materiality. *Synthese*, 169(3), 483–496.
- Piccinini, G. (2015). *Physical computation: A mechanistic account*. OUP Oxford.
- Piccinini, G., & Craver, C. (2011). Integrating psychology and neuroscience: Functional analyses as mechanism sketches. *Synthese*, 183(3), 283–311.
- Rusanen, A.-M., & Lappi, O. (2007). The limits of mechanistic explanation in neurocognitive sciences. In *Proceedings of the European cognitive science conference*. London: Francis and Taylor.
- Serban, M. (2015). The scope and limits of a mechanistic view of computational explanation. *Synthese*, 192(10), 3371–3396.
- Shagrir, O. (2010). Marr on computational-level theories. *Philosophy of Science*, 77(4), 477–500.
- Sompolinsky, H. (2014). Computational neuroscience: beyond the local circuit. *Current Opinion in Neurobiology*, 25, xiii–xviii.
- Trappenberg, T. (2009). *Fundamentals of computational neuroscience*. OUP Oxford.
- Winsberg, E. (2010). *Science in the age of computer simulation*. University of Chicago Press.
- Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv Preprint arXiv:1506.06579*. Retrieved from <http://arxiv.org/abs/1506.06579>