Submitted to *Encyclopedia of Cognitive Science*

Levels of analysis in neural modeling

Peter Dayan

Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR

Tel: +44 (0) 20 7679 1175
Fax: +44 (0) 20 7679 1173
Email: dayan@gatsby.ucl.ac.uk

Table of Contents

Levels of analysis in neural modeling

Abstract

Neural systems are analyzed by building multiple levels of descriptive and explanatory mathematical models. Parallel to these are interpretive, computational, models, showing how information is represented and manipulated.

1 Levels of Organization in the Brain

The construction and refinement of qualitative and quantitative models for observed phenomena is the standard practice in neuroscience, just as it is in all other scientific disciplines. What particularly distinguishes neural modeling, apart from the sheer complexity of the phenomena, is that brains are computational devices, transducing, representing, manipulating, and storing information, and using this information to control actions. The most comprehensive neural models must therefore play the dual role of accounting for experimental data *and* interpreting it in terms of underlying computations. Understanding what neural models are, what their connections to experimental data are, and how to build and use them in practice, is tricky because of the complexity and the multiple goals for modeling.

One key idea for structuring the enterprise of modeling that has many, and confusingly different, manifestations, is that of different levels of organization. The confusion comes because there are at least four different, though partially overlapping, threads to this idea. One thread is the standard one of scientific reduction, describing observable phenomena in qualitative and quantitative detail, and explaining them in terms of descriptions of their underlying substrates at lower and less abstract levels. A second thread, that is dual to this one, is that of the construction or synthesis of systems to execute some particular task. Here, the use of different levels (just like the use of subroutines and procedures in writing computer programs) is a conventional divide-and-conquer strategy in the face of design complexity. A third thread, originally suggested by David Marr, and widely attacked for being too rigid, is associated specifically with computational modeling and involves a *computational* level, which is an abstract description of the goal of the task, and the logic of the strategy by which it is to be satisfied, an *algorithmic* level, which is a description of the way that information should be represented and algorithmically manipulated in the service of carrying out the task, together with an *implementational* level, which is a description of the way that these representations and manipulations are instantiated in neural hardware. To avoid confusion, we will reserve the word *planes* for these three computational levels. Understanding these threads of levels of organization is critical to understanding the construction and critique of neural models.

The fourth and final thread is that of levels of processing as a strategy for manipulating and extracting information from input. This is best exemplified by the visual system of primates. In some primates, tens of different structurally- and functionally-defined areas of the brain are devoted wholly or partially to analyzing visual information. On the basis of a number of lines of evidence, including characteristic layers of termination of connections, these areas appear to be organized in a somewhat loose hierarchy, starting from the lowest level in primary visual cortex, where cells tend to have comparatively small receptive fields (*ie* are directly responsive only to a small area of visual space) and simple response properties (for instance, responding to small bars of light at particular orientations), all the way up to much higher levels involving areas such as inferotemporal cortex, where cells have large receptive fields and complex response properties (for instance, responding to pictures of particular classes of faces, but not other, apparently similar objects). Similarly, when building machines to process images for such tasks as object recognition, one can build a processing hierarchy, in which the visual input is subjected to sequences of manipulations leading to the answer.

This idea of a processing hierarchy is distinct from the idea of an analytical hierarchy. The latter, which is the main focus of this review, applies whatever form neural computations happen to take – it is about describing and explaining the behavior of systems of any sort. The former applies because this is how neural information processing often appears to be organized.

2 Types of Neural Model

Conventional reductive models

The first thread to the idea of levels concerns standard reductive modeling. For the practical (as opposed to the philosophical) aspects of this, models are useful in at least two ways: i) *describing* neural phenomena, and ii) providing a means for reductionist *explanations* of the phenomena, by appealing to the mechanisms that might actually be responsible for generating the phenomena in the first place. These mechanisms themselves are usually understood in terms of models, and so the modeling process is recursive. Although models need not be expressed as a set of mathematical equations, quantifying them in some way is essential to be able to check whether the mechanisms postulated (or rather the models of the mechanisms postulated) are really capable of capturing the phenomena to the desired degree of accuracy.

An example is modeling of the shape over time of the voltage inside an axon (relative to extra-cellular space) during the passage of an action potential or spike. The data can be characterized at some level of detail, such as the time course of the voltage averaged over a large number of spikes. The data can be summarized, at least within the approximation of experimental error, by any number of different descriptive quantitative models. For instance one could tune the parameters of various high order polynomials or piecewise linear functions to replicate the shape of the spike accurately. By contrast to these non-mechanistic descriptions, Alan Hodgkin and Andrew Huxley, in their classic study, were interested in explaining how action potentials arise, and therefore constructed a model on the basis of what they expected about the way that cells might manipulate the potential difference between inside and outside. They had evidence that the basic mechanism was some way for the axon to change its permeability to particular ions, and so postulated as a mechanism a form of channel or gate in the cell's membrane. They built an quantitative explanatory model of the action potential from a quantitative descriptive model of the gate, fitting the parameters of the model to make correct the form of the overall action potential it produced.

Following the advent of more powerful experimental techniques, we can now understand Hodgkin and Huxley's deterministic phenomenological channel model in terms of a large number of tiny individual channels in the membrane which open and close in a stochastic manner. The summed effect of all these individual channels matches Hodgkin and Huxley's model very closely. So we can now build an explanatory model of the action potential using descriptive models of these gates. Based on even more recent experimental evidence, we could go further, and try to build explanatory models of these gates from our knowledge of their molecular structure.

Another example is the phenomenon of the behavior of an interconnected network of neurons. This can be described in terms of a (relatively complicated) dynamical state description, but it might also be explained in terms of descriptions of the properties of the individual neurons and their connections. In turn, the descriptions of the individual neurons can be explained in terms of their detailed geometry and the membrane-bound channels they contain.

The general conclusions from such examples are that there are different *levels* of model, corresponding to different levels of reduction of a phenomenon, and, often in neuroscience, different levels of anatomical detail. There are *descriptive* models at a level, which capture the behavior without much regard to the substrate, and *explanatory* models, which capture the behavior by reducing it to models at lower levels. Quantitative models tie together the different levels because they allow proof, or at least numerical demonstration, that the behavior assumed at one level can truly account for the behavior that it is intended to explain. Almost always the models at a level are only approximate, with models at lower levels being faithful to more intricate details of the experimental observations. Indeed, there is typically a match between the degree of abstraction of the models at a level, and the degree of abstraction in the description of the data at that level. For instance, the gating model for spiking is inherently deterministic, whereas its single-channel explanation is inherently stochastic, and, thereby, is able to fit more detailed aspects of experimental data. Equally, we later consider models of neurons at different levels of abstraction, from firing-rate models, which abstract out even spikes, to compartmental models that capture even the detailed geometry of neurons and the effect of this on things like dendritic integration.

One can think of the models at different levels as *constraining* each other – the model of the stochastic channel has to behave correctly to produce the overall spike, if such gates are really to be responsible for generating action potentials. Similarly, explanatory models of the action potential are constrained by what is possible at lower levels. One can also think of the different levels as *liberating* each other – there are many different gating mechanisms that have the same qualitative behavior – so if one is only interested in the effect of gating and not its cause, then one need not be concerned with the explanatory details of the model. In practice, there is not really such a strong separation between explanatory and descriptive models. For instance, Hodgkin and Huxley did not just build any descriptive model of their phenomenological gate. Indeed, so close was their descriptive model to the true underlying mechanism, that we can now interpret in terms of the latter some of the parameter values that they found by their fitting process.

Computational interpretive models

Computational models start from the premise that many of the tasks for brains are best characterized as involving computations. For example, for a bee to forage optimally in a field containing two types of flowers with different characteristics of provision of nectar, it must continually compute the choice between sorts of flower to land on. Equally, to catch a flying ball in a hand, the brain has to take visual input about where the ball is and how it is moving, and transform this into a sequence of motor commands to move the hand to an appropriate place at an appropriate time, with the fingers arranged correctly. This transformation happens in various parts of the visual and motor systems of the brain.

Computational modeling is about *imputing* a computational task and *interpreting* the collective behavior of the neural components of the system in terms of this task. In these cases, the tasks involve generating the appropriate output, such as the choice of the better flower or the sequence of actions to catch the ball, from the input, such as the past qualities of the flowers or the visual impression of the ball over time. The issue is not whether brains actually *are* computers – it is about these imputations and interpretations.

The key aspects of computations are *representation*, *storage*, and *transformation* or *algorithmic manipulation*. Computational modeling is therefore about understanding (*ie* interpreting) how neural machinery can represent and store information about aspects of the outside world, and how it can transform information from one form to another in the satisfaction of tasks. As in a standard computer, the *semantics* of the computation are implemented by the *syntax* of the physical substrate. Computational models can themselves be decomposed into the three planes described above, from an abstract description of the underlying task, through a more concrete description of the representations and algorithms adopted to satisfy the task, to a description of how these representations and algorithms are actually implemented. There is not necessarily a single description in the computational and algorithmic planes – *eg* there can be many different algorithmic ways of expressing the same transformation.

Computational models satisfy many of the same properties as conventional models. First, in the same way that there are different levels of conventional models, there are different levels of computational models, paralleling a decomposition of the underlying computation. Second, there are both descriptive and explanatory computational models. For example, take the case of a single, spiking, neuron. One can build a descriptive model of what the output of a neuron represents by presenting all possible classes of input to the animal and recording what spikes it produces. However, one can also build an explanatory model by working out what inputs the neuron has, what these inputs represent, and what transformations the neuron performs on these inputs.

The third similarity between computational and conventional models is functionalism, that the same abstractly defined computation can, in principle, be carried out by many different representations and algorithms, and the same algorithm instantiated in many different forms of hardware. Since the overall tasks that need be solved (such as visual object or speech recognition) are extremely complex, it is quite likely that the neural substrate, as an evolved rather than a designed system, is only capable of instantiating an approximation to the sort of computational description of the tasks that we can write succinctly. Therefore, the system does not really possess functional equivalence between the planes – the implementational planes implement functions that might be slightly different from the idealised ones described in the computational planes, with differences that have observable consequences.

From the perspective of analyzing neural systems, computational and conventional modeling should work hand in hand. Consider a single level. At the implementational plane of the computational model lies exactly the experimental phenomena for which conventional reductive modeling provides an account. The explanatory reduction of these phenomena comprises the lower level of the conventional model. It must also comprise the implementational plane of the lower level of the computational model. The algorithmic and computational planes at multiple levels of the computational model, which go along with the implementational plane, will duly be forced to be consistent with the multiple levels of the conventional model.

The ultimate goal is to have conventional and computational models for neural function that are mutually consistent, extending from the lowest levels of molecular dynamics to the highest levels of ethology. We would then have a complete, quantitative, multi-level understanding of *how* the brain executes *which* tasks. Of course, such exhaustiveness is far off at the moment. The coverage of conventional and computational models of any sorts is very poor, and appropriate reductions and interpretations are few and far between.

In practice, computational models are often used synthetically rather than analytically. That is, networks of neurons, described down to some level of abstraction, are constructed to perform a computational task, and the behavior of the model neurons is compared with that observed in physiological (and other) experiments, justifying the model. In such synthetic treatments, the multiple levels of computational modeling can be made explicit, and the reduction between the levels can be shown to be exact. One popular synthetic technique is to start from an idea about optimal or normatively correct ways to perform a task, and to seek implementations involving biologically reasonable components.

3 Degrees of Modeling Detail

To a very coarse approximation, one can separate out three different classes of quantitative models in common use, namely conductance-based models, integrate-and-fire models, and firing-rate models. The different models are naturally couched at different levels of abstraction and, as described in general terms above, are used to account for data that is similarly collected at different levels of abstraction. Far from all the models that are built and used fit neatly into these categories, but they nevertheless give a flavor of the differing degrees of neural and analytical detail that are regularly employed.

Conductance-based models

Conductance-based models place their emphasis on describing single or just a few neurons with a high degree of detail. They typically approximate the structure of an individual neuron by multiple, interconnected, compartments, each of which is treated as being electrically compact. The whole set of compartments is designed to be more or less faithful to the geometry of the neuron, including such facets as branching points of dendrites and the diameters and lengths of different parts. There is an elaborate art to representing cells that have complex geometrical structures in terms of just a few compartments, or even just a single compartment, a reduction that is often necessary to make computations involving the models adequately quick. In standard conductance-based models, each compartment is given an assortment of active channels, such as voltage sensitive or synaptic channels.

Conductance-based models of single cells are ideal for explaining phenomena to do with spikes and the thresholds for initiating spikes, the precise effects of synaptic input, bursting, spike adaptation, spikes that propagate backwards up the dendritic tree, and the like. One problem with these models is that there is rarely good experimental data on the actual locations on the dendritic tree or the strengths of the active channels. Such data are obviously critical to making the models faithful to the neural substrate. More generally, conductance-based models involve a very large number of parameters, and the values of only a few of these can be determined from experiments. A second problem with these models is that they are so complex that they cannot readily be analyzed, and yet can exhibit a huge range of behaviors depending on the exact values of the parameters.

There are various extensions of this single-cell use of conductance-based models. One is that it is common to model networks of neurons by connecting together simple (*eg* single compartment) conductance-based models through model synapses. This allows simple network properties to be explored; more complicated networks are typically modeled using integrate-and-fire models. A second extension is to modeling important internal states, such as intracellular calcium levels, using the same compartmental structure. This is particularly important for things like calcium-sensitive potassium channels, which can lead to phenomena such as spike-rate adaptation. Although compartmental models capture the electrical geometry of single cells, they rarely capture the three dimensional milieu in which the cells live.

Integrate-and-fire models

Integrate-and-fire models lie at a level of abstraction above conductance-based models. Rather than using active channels to implement action potentials, they make the approximation of using a symbolic model of spike generation coupled with a leaky integrator model of a cell when its voltage is below the threshold for spike initiation. They also radically simplify the geometry of cells, eliminating the compartmentalization and including, at best, a stereotyped time-course for synaptic input and for other time-dependent factors such as those allowing spike-rate adaptation.

Integrate-and-fire models are good for simulating large, recurrently connected, networks of neurons. Many mathematical issues about networks, such as the synchronization and desynchronization of spiking across the whole population, and the effects of different sorts and sources of noise have been explored through using them. Furthermore, recent evidence suggests that the details of such phenomena as synaptic plasticity are dependent on such phenomena as precise time differences between pre-synaptic and post-synaptic activity. The integrate-and-fire model is the simplest form that still outputs spikes, and so can be used to address such issues. However, integrate-and-fire models pose substantial analytical difficulties themselves, and sometimes end up being an unhappy medium between more realistic, but analytically intractable, compartmental models, and highly abstract, but tractable, firing-rate models.

Firing-rate models

The most abstract level of characterization of neurons abandons spiking altogether, and instead treats the output of cells as being continuous-valued, time-varying, firing rates. This can be derived as an abstract approximation to integrate-and-fire neurons, under some assumptions about the time-constants of processes inside cells. Networks of firing-rate models can be constructed, in which the influence of one cell on another is given by the product of the pre-synaptic cell's firing rate and the synaptic strength for the connection.

The main advantages of the firing-rate models are their empirical and analytical tractability. Firing-rate models usually involve a mild non-linearity, turning an internal continuous variable like somatic voltage or current into a firing rate (which must be positive). Therefore, networks of neurons described using firing-rate models can be treated as coupled, non-linear differential equations that can be shown to exhibit dynamical behaviors such as attraction to one of a set of fixed points, oscillations or chaos. By contrast with conductance-based models, these often evolve to relatively simple fixed-point or limit-cycle attractors. The regularities implied by attractor and oscillatory dynamical behaviors make them ideal as substrates on which to hang analyses of network computation.

Although it is possible to study the effects of synaptic plasticity in the context of conductance-based or integrate-and-fire models, by far the bulk of the work on computational analyses has been performed using firing-rate models. Here, with the exception of studies on recurrent attractor networks, a majority of the work has focused on non-recurrent, feedforward network models, which are analytically much simpler to handle. These models take information represented in one way at one level of a processing hierarchy by the firing rates of a population of neurons, and transform and manipulate it to represent it in a different way by the firing rates of a population of neurons at a higher level of the hierarchy. Rules for plasticity have been seductive for computational modeling, since they offer an obvious way for large networks of fairly simple processing units to come to perform computationally sophisticated tasks, apparently without requiring sophisticated programming.

Although the boundaries between them are somewhat blurred, there are three main classes of learning model. Unsupervised learning models act in a self-organizing manner, extracting statistical structure from input data. They are often used to capture activity-dependent adaptive processes that are assumed to be operational during development. Reinforcement learning models use evaluative information, *ie* rewards and punishments, in temporally complex and controllable environments (such as mazes) and specify how to predict future returns and choose actions in order to maximize these returns. Supervised learning models are rather special in that adaptation is based on information both about the inputs to the network and the desired outputs. Supervised learning is very extensively used outside the context of neural modeling. However, in two incarnations, it is important for neural modeling too. One is when the output of one network is used to train another network. This requires an intricate dance by which neurons in the trained network must have two sets of input, one of which controls synaptic plasticity in the other. It has been suggested that this might take place under the control of neuromodulators. The other case for supervised learning is when the key question is whether a particular design of network of neurons is capable of executing a particular computational task or exhibiting some particular behavior, or whether the activity of some neurons in a network is consistent with their playing a role in the solution of a task. In this case, one can attempt to train the network (for instance by minimizing the discrepancy between the target and the actual outputs), using procedures that need have no relation to the rules governing neural adaptation.

4 Summary

Partnering conventional, multi-level, reductive, models of experimental observations on neural mechanisms, are models offering computational interpretations, couched at exactly the same multiple levels. These models indicate how the neural mechanisms implement computations, in the sense of representing information and performing algorithmic manipulations which are appropriate for solving a computationally well-specified task. The computational and conventional models should ideally mesh completely at all the levels. Few existing conventional and computational models actually achieve this degree of mutual coherence. Most current modeling is either conventional, at the compartmental and integrate-and-fire level, or computational, using sophisticated models of synaptic adaptation in the face of extremely simplified models of individual neurons, but rarely both conventional and computational.

There are various suggestions that modeling should really proceed top-down, from the definition and computational decomposition of abstract tasks to the reductive neural implementation. However, such a strict policy is not productive as a strategy for analyzing neural systems, because it means throwing away constraints from the experimental data, and because it relies on us having adequate, multi-level accounts of the underlying computational tasks, which we presently lack. Instead, both conventional and computational modeling at multiple levels should progress together.

Further Reading

Churchland PS (1986) *Neurophilosophy*. Cambridge, MA: MIT Press.

Churchland PS and Sejnowski TJ (1992) *The Computational Brain*. Cambridge, MA: MIT Press.

Foster CL (1992) *Algorithms, Abstraction and Implementation: Levels of Detail in Cognitive Science.* London: Academic Press.

Haugeland J (1978) The nature and plausibility of cognitivism. *Behavioral and Brain Sciences* **1**:215-226.

Marr D (1982) *Vision*. New York: Freeman.

Montague PR and Dayan P (1998). Neurobiological modeling: Squeezing top down to meet bottom up. In: Bechtel W and Graham G (eds) *A Companion to Cognitive Science,* pp. 526-542. Oxford: Basil Blackwell.

Pylyshyn ZW (1984) *Computation and Cognition.* Cambridge, MA: MIT Press.