Digitized neural networks: long-term stability from forgetful neurons

Alexandre Pouget and Peter Latham

Understanding how realistic networks integrate input signals over many seconds has eluded neuroscientists for decades. Koulakov and colleagues now propose a computational model to explain how bistable neurons might allow a network to integrate incoming signals.

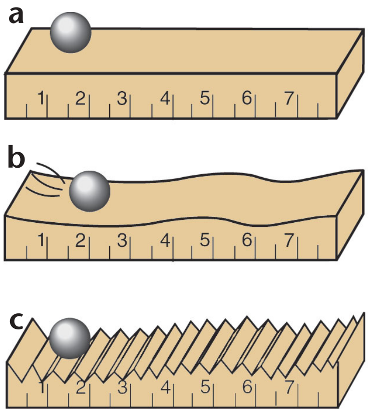
Temporal integration is relatively simple—it is nothing more than the accumulation (or sum) over time of some signal. This simple process, however, turns out to be a powerful form of computation that is believed to underlie a large variety of seemingly unrelated behaviors and cognitive functions, including eye movement[1], navigation[2,3], short-term memory of continuous variables[4,5] and mental rotation[6]. It might even be critical in the ability of neural circuits to perform statistical inferences[7], which is remarkable, considering that statistical inference represents one of the most promising computational theories of higher brain functions, such as perception, decision making, motor control and high-level reasoning[8–10]. Simple or not, understanding how the nervous system performs integration has been one of the most frustrating problems in computational neuroscience. In this issue, Koulakov *et al.*[11] propose a new theoretical model to explain how neural networks might integrate signals.

A classic example of a neural integrator is a circuit contained in the postsubiculum of the rat. Neurons in this circuit encode the direction of the rat's head in world-centered coordinates, acting much like an internal compass[2]. This compass is updated after each movement of the head, even when the animal is being moved by the experimenter in complete darkness. This is particularly remarkable because, in darkness, the main source of information regarding head movements is the discharge of neurons in the semicircular canals of the inner ear, which respond to head acceleration, not head direction. Head direction, however, can be recovered in two steps. First, head velocity can be computed by summing (that is, integrating) head acceleration over time. Then, head direction is obtained by repeating this operation, but, this time, on head velocity. This double integration is believed to be implemented by the circuitry linking the vestibular system to the postsubiculum.

As these examples illustrate, integration is emerging as a basic computational operation used by the brain, much like the switching of transistors is a basic computational operation used by digital computers. Unfortunately, despite numerous experimental and theoretical studies, its neural basis is still very poorly understood. But what is so complicated about it? After all, it is particularly easy to design an algorithm that integrates a signal over (discrete) time: on each time step, simply add the value of the input signal to the accumulated value, the latter being stored in memory (such integrators can easily be built from components bought for a few dollars at any electronics store). Two operations are required for this: addition and memory. The difficulty for the brain lies in the memory operation: 'remembering' the summed value is difficult to model in realistic neurons because such neurons integrate signals over a short time period (milliseconds), whereas biologically relevant signals occur over a longer period (seconds). In other words, neurons tend to forget what happened to them only 10 ms ago; given this intrinsic forgetfulness, it has been difficult to understand how networks of neurons remember things that happened seconds ago.

One solution that has been proposed by theorists is to build networks of neurons that have plenty of internal feedback[1,12]. This internal feedback allows the network to maintain a particular firing rate, even in the absence of input—much the same way a group of very forgetful people could remember a phone number by continually repeating it to each other. Such networks act as integrators in the following way: excitatory external input causes an increase in firing rate, inhibitory input causes a decrease, and, most importantly, without input, the firing rate of the network stays constant. This last fact is most important, because it allows the network to remember the accumulated input signal.

The problem with this approach is that it requires incredible fine-tuning of parameters. Trying to maintain a particular firing rate in a neural network is like trying to get a marble to sit still on a table (**Fig. 1a**). If the



Ivelisse Robles

Fig. 1. Storing values with a marble on a table. (**a**) The location of the marble can be used to store any analog value (1.5 in this example). Moreover, if there is enough drag on the surface that the marble stops quickly in the absence of any external force, the position of the marble codes for the integral of the force acting on it, that is, it acts as an integrator of force. (**b**) If the table is not flat, the marble will roll toward a local minimum, effectively forgetting the initial position of the marble, that is, the initial stored value. The characteristic time it takes to approach the minimum is the 'forgetting' time. For neurons, the forgetting time is around 10 ms, much too short to integrate input signals over seconds. Building a network with a long forgetting time—a nearly flat surface—out of such forgetful neurons is hard; this is the 'fine tuning' problem. (**c**) The solution proposed by Koulakov *et al.* is to put pits in the surface, so the marble does not roll even if the surface is slightly warped.

Alex Pouget is in the Department of Brain and Cognitive Sciences, University of Rochester, Rochester, New York 14627, USA. Peter Latham is in the Department of Neurobiology, UCLA, Los Angeles, California, 90095 USA.
e-mail: alex@bcs.rochester.edu; pel@ucla.edu

table is perfectly flat, the marble can 'remember' any value for arbitrarily long times—if you want it to remember the number '1.5', for example, simply put the marble at position 1.5 (**Fig. 1a**). The slightest slope will cause the marble to drift, however, and thus forget its stored value (**Fig. 1b**), and even if the table is perfectly flat, the system is still sensitive to noise, as the slightest tremor will cause the marble to wander aimlessly. (With a little more work, the marble can be turned into an integrator. But to do this, there needs to be enough drag on the marble so that it stops quickly in the absence of external forces. In the high-drag regime, the marble does more than simply remember where you put it: it integrates the forces that act on it. Again, however, the table needs to be perfectly flat, or the marble will drift; it also needs to be noise free, or the marble will wander.)

In digital computers, these problems—fine tuning of parameters and sensitivity to noise—are avoided thanks to two tricks. First, all numbers are discretized; they are represented as binary strings (for instance, 64 bits). Although this discretization prevents computers from integrating analog values, 64-bit strings are sufficiently long for most practical purposes. Second, each bit is in a very stable state, that is, its value cannot change because of the internal noise in the computer. This is because the voltage difference between the 0 and 1 that make up the binary code is much larger than the voltage fluctuations due to the electronic noise, and much larger than any applied forces, like those caused by stray electric fields. Once a bit is set, it does not change unless the CPU tells it to change. Integration, then, becomes trivial: simply add or subtract numbers from a counter. Because neither noise nor external forces cause bits to spontaneously flip, we are guaranteed that the counter will faithfully accumulate all additions and subtractions.

Applied to our table example, this approach would be equivalent to carving pits on the surface of the table (**Fig. 1c**). Provided the pits are deep enough, there is no need for fine tuning; the table could be slightly tilted without dire consequences. Furthermore, the sensitivity to noise would be greatly reduced: the marble would be unlikely to jump from one position to the next when the table shakes. Granted, the marble could only store a finite number of values, but this is not a problem if we make the pits closely spaced—just as it isn't a problem in digital computers that represent numbers using 64 bits.

In essence, this is the solution proposed by Koulakov *et al.*[11], but applied to a recurrent network of spiking neurons. It is easy to build a recurrent network that is stable at any one particular firing rate, but stability at a single rate does not solve the problem of integration. Koulakov *et al.*, however, took advantage of an important fact: the stable firing rate in a recurrent network depends on the number of neurons in the network. This is because the drive to each neuron depends on the number of neurons presynaptic to it. More neurons means more drive and a higher firing rate; fewer neurons means less drive and a lower firing rate. Importantly, for a fixed number of neurons, the network is stable. This leads to the following key observation: a network can have many stable firing rates if neurons can be effectively removed or added, and it takes a certain activation energy to do so. Koulakov *et al.* accomplished this by building a network out of interconnected bistable neurons. If external input is applied to such a network, it can change the relative numbers of quiet and active neurons: excitatory input switches some quiet neurons to the active state, increasing the effective number of neurons and thus increasing the average firing rate; inhibitory input switches some active neurons to the quiet state, decreasing the effective number of neurons and the rate; and no input does nothing to the effective number of neurons and thus nothing to the rate. The resulting network is a neural integrator that needs little fine tuning—in models, at least, network operation is robust to parameter changes of as much as ±20%. Just as important, it is stable with respect to noise, as it takes large fluctuations to cause the bistable neurons to switch states. This stability is consistent with previous models in which bistable neurons were used in models of working memory[5].

Bistability is a key aspect of the Koulakov *et al.* model[11], so it is natural to ask how it might come about in real neurons. It is easy to imagine in principle: all one needs is a voltage-dependent inward current that is activated when a neuron is firing and inactivated when the neuron is silent. The existence of bistable neurons in cortex is more speculative, although such neurons have been observed in motor neurons in anesthetized rats[13]. In any case, Koulakov *et al.* use two models for bistability: one in which the voltage-dependent inward current is supplied by NMDA channels, the other in which there is an activity-dependent drive that is supplied by small groups of garden-variety neurons.

Whether or not their model applies to real neural integrators, however, is still far from decided. From a theoretical point of view, the model must be made more realistic in at least four ways if it is to describe integrators in mammalian networks. First, inhibitory neurons need to be included. So far, the model contains only excitatory neurons, which forces it to operate in an unrealistically weakly coupled regime (all-excitatory networks with realistic coupling are unstable, in the sense that they can only fire at high rates[12]). Second, a more realistic ratio of AMPA to NMDA receptors needs to be incorporated. Two conductance ratios were used in their models, 5 and infinity—both significantly higher than the experimentally observed ratio of 0.5–1 (ref. 14). Finally, the size of the network needs to be scaled up: in mammalian networks, each neuron receives ~5000 inputs, much larger than the 100–300 inputs used in the Koulakov *et al.* model. All of these effects increase the effective noise, which would make it more likely for the bistable neurons to spontaneously switch states and thus wreak havoc with stability. Whether or not the model is robust to these more realistic features needs to be investigated numerically.

Experimentally, the proposed model makes a very strong prediction: in areas where neural integrators exist, neurons should be bistable. Besides being a clever solution to a hard problem, the falsifiability of the model is one of its best features. We look forward to future experiments that either confirm or deny bistability as the mechanism underlying neural integrators.

1. Seung, H. *Proc. Natl. Acad. Sci. USA* **93**, 13339–13344 (1996).
2. Taube, J. S., Muller, R. U. & Ranck, J. B. J. *J. Neurosci.* **10**, 420–435 (1990).
3. Zhang, K. *J. Neurosci.* **16**, 2112–2126 (1996).
4. Droulez, J. & Berthoz, A. *Proc. Natl. Acad. Sci. USA* **88**, 9653–9657 (1991).
5. Camperi, M. & Wang, X.-J. *J. Comput. Neurosci.* **5**, 383–405 (1998).
6. Georgopoulos, A. P., Lurito, J. T., Petrides, M., Schwartz, A. B. & Massey, J. T. *Science* **243**, 234–236 (1989).
7. Deneve, S., Latham, P. & Pouget, A. *Nat. Neurosci.* **4**, 826–831 (2001).
8. Gold, J. I. & Shadlen, M. N. *Trends Cogn. Sci.* **5**, 10–16 (2001).
9. Knill, D. C. & Richards, W. *Perception as Bayesian Inference* (Cambridge Univ. Press, New York, 1996).
10. Wolpert, D. M. & Ghahramani, Z. *Nat. Neurosci.* **3**, 1212–1217 (2000).
11. Koulakov, A., Raghavachari, S., Kepecs, A. & Lisman, J. E. *Nat. Neurosci.* **5**, 775–782 (2002).
12. Amit, J. & Brunel, N. *Cereb. Cortex* **7**, 237–252 (1997).
13. Eken, T. & Kiehn, O. *Acta Physiol. Scand.* **136**, 383–394 (1989).
14. Andrasfalvy, B. K. & Magee, J. C. *J. Neurosci.* **21**, 9151–9159 (2000).