a million or a billion steps? The answer is the brain doesn't "compute" the answers to problems; it retrieves the answers from memory. In essence, the answers were stored in memory a long time ago. It only takes a few steps to retrieve something from memory. Slow neurons are not only fast enough to do this, but they constitute the memory themselves. The entire cortex is a memory system. It isn't a computer at all.

Let me show, through an example, the difference between *computing* a solution to a problem and *using memory* to solve the same problem. Consider the task of catching a ball. Someone throws a ball to you, you see it traveling toward you, and in less than a second you snatch it out of the air. This doesn't seem too difficult—until you try to program a robot arm to do the same. As many a graduate student has found out the hard way, it seems nearly impossible. When engineers or computer scientists tackle this problem, they first try to calculate the flight of the ball to determine where it will be when it reaches the arm. This calculation requires solving a set of equations of the type you learn in high school physics. Next, all the joints of a robotic arm have to be adjusted in concert to move the hand into the proper position. This involves solving another set of mathematical equations more difficult than the first. Finally, this whole operation has to be repeated multiple times, for as the ball approaches, the robot gets better information about the ball's location and trajectory. If the robot waits to start moving until it knows exactly where the ball will arrive it will be too late to catch it. It has to start moving to catch the ball when it has only a poor sense of its location and it continually adjusts as the ball gets closer. A computer requires millions of steps to solve the numerous mathematical equations to catch the ball. And although a computer might be programmed to successfully solve this problem, the one hundred-step rule tells us that a brain solves it in a different way. It uses memory.

How do you catch the ball using memory? Your brain has a stored memory of the muscle commands required to catch a ball (along with many other learned behaviors). When a ball is thrown, three things happen. First, the appropriate memory is automatically recalled by the sight of the ball. Second, the memory actually recalls a temporal sequence of muscle commands. And third, the retrieved memory is adjusted as it is recalled to accommodate the particulars of the moment, such as the ball's actual path and the position of your body. The memory of how to catch a ball was not programmed into your brain; it was learned over years of repetitive practice, and it is stored, not calculated, in your neurons.

You might be thinking, "Wait a minute. Each catch is slightly different. You just said the recalled memory gets continually adjusted to accommodate the variations of where the ball is on any particular throw . . . Doesn't that require solving the same equations we were trying to avoid?" It may seem so, but nature solved the problem of variation in a different and very clever way. As we'll see later in this chapter, the cortex creates what are called *invariant representations,* which handle variations in the world automatically. A helpful analogy might be to imagine what happens when you sit down on a water bed: the pillows and any other people on the bed are all spontaneously pushed into a new configuration. The bed doesn't compute how high each object should be elevated; the physical properties of the water and the mattress's plastic skin take care of the adjustment automatically. As we'll see in the next chapter, the design of the six-layered cortex does something similar, loosely speaking, with the information that flows through it.

So the neocortex is not like a computer, parallel or otherwise. Instead of computing answers to problems the neocortex uses stored memories to solve problems and produce behavior.