B1.1 Neurons and neural networks: the most abstract view

Michael A Arbib

Abstract

See the abstract for Chapter B1.

There are many types of artificial neuron, but most of them can be captured as formal objects of the kind shown in figure B1.1.1. There is a set X of signals which can be carried on the multiple input lines $x_1, \ldots, x_n$ and single output line y. In addition, the neuron has an internal state s belonging to some state set S.

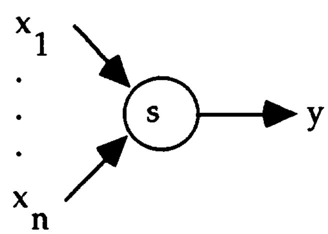

Figure B1.1.1. A 'generic' neuron, with inputs $x_1, \ldots, x_n$, output y, and internal state s.

A neuron may be either discrete-time or continuous-time. In other words, the input values, state and output may be given at discrete times $t \in \mathbb{Z} = \{0, 1, 2, 3, \ldots\}$, say, or may be given at all times t in some interval contained in the real line $\mathbb{R}$. A *discrete-time neuron* is then specified by two functions which specify (i) how the new state is determined by the immediately preceding inputs and (in some neuron models, but by no means all) the previous state, and (ii) how the current output is to be 'read out' from the current state:

The *next-state-function* $f : X^n \times S \to S$, $s(t) = f(x_1(t-1), \ldots, x_n(t-1), s(t-1))$; and
The *output function* $g : S \to Y$, $y(t) = g(s(t))$.

As we shall see in later sections, popular choices take the signal-set X to be either a binary set—$\{0, 1\}$ is the 'classical choice', though physicists, inspired by the 'spin-glass' analogy, often use the spin-down, spin-up set denoted by $\{-1, +1\}$—or an interval of the real line, such as $[0, 1]$; while the state-set is often taken to be $\mathbb{R}$ itself. A *continuous-time neuron* is also specified by two functions $f : X^n \times S \to S$, and $g : S \to Y$, $y(t) = g(s(t))$, but now f serves to define the *rate of change* of the state, that is, it provides the right-hand side of the differential equation which defines the state dynamics:

$$\frac{\mathrm{d}s(t)}{\mathrm{d}t} = f(x_1(t), \ldots, x_n(t), s(t)) \, .$$

Clearly, S at least can no longer be a discrete set. A popular choice is to take the signal-set X to be an interval of the real line, such as $[0, 1]$, and the state-set to be $\mathbb{R}$ itself.

The focus of this chapter will be on motivating and defining some of the best known forms for f and g. But first it is worth noting that the subject of neural computation is not interested in neurons as

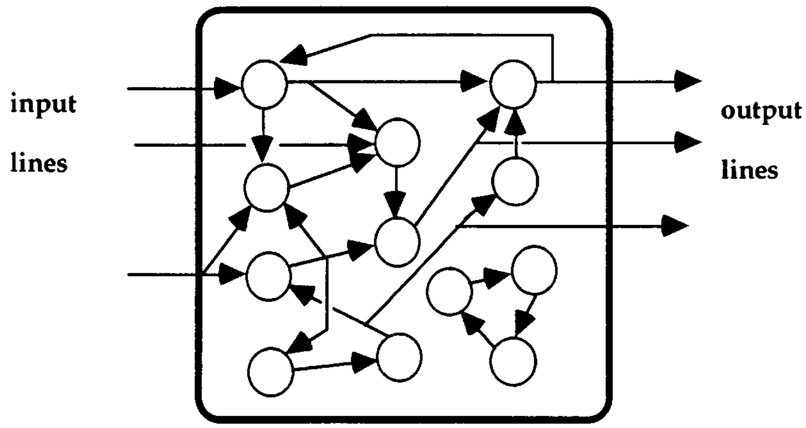

Figure B1.1.2. A neural network viewed as a system (continuous-time case) or automaton (discrete-time case). The input at time t is the pattern on the input lines, the output is the pattern on the output lines; and the internal state is the vector of states of all neurons of the network.

ends in themselves but rather in neurons as units which can be composed into networks. Thus, both as background for later chapters and as a framework for the focused discussion of individual neurons in this chapter, we briefly introduce the idea of a neural network.

We first show how a neural network comprised of continuous-time neurons can also be seen as a continuous-time system in this sense. As typified in figure B1.1.2, we characterize a neural network by selecting N neurons and by taking the output line of each neuron, which may be split into several branches carrying identical output signals, and either connecting each branch to a unique input line of another neuron or feeding it outside the network to provide one of the N_L network output lines. Then every input to a given neuron must be connected either to an output of another neuron or to one of the (possibly split) N_1 input lines of the network. Then the input set X of the entire network is $\mathbb{R}^{N_1}$, the state set $Q = \mathbb{R}^N$, and the output set $Y = \mathbb{R}^{N_L}$. If the ith output line comes from the jth neuron, then the *output function* is determined by the fact that the ith component of the output at time t is the output $g_j(s_j(t))$ of the jth neuron at time t. The *state transition function* for the neural network follows from the state transition functions of each of the N neurons

$$\frac{\mathrm{d}s_j(t)}{\mathrm{d}t} = f_j(x_{1_j}(t), \ldots, x_{n_j j}(t), s_j(t))$$

as soon as we specify whether $x_{i_j}(t)$ is the output of the kth neuron or the value currently being applied on the lth input line of the overall network.

Turning to the discrete-time case, we first note that, in computer science, an *automaton* is a discrete-time system with discrete input, output and state spaces. Formally, we describe an automaton by the sets X, Y and Q of inputs, outputs and states, respectively, together with the *next-state function* $\delta : Q \times X \to Q$ and the *output function* $\beta : Q \to Y$. If the automaton is in state q and receives input x at time t, then its next state will be $\delta(q, x)$ and its next output will be $\beta(q)$. It should be clear that a network like that shown in figure B1.1.2, but now a discrete-time network made up solely from discrete-time neurons, functions like a finite automaton, as each neuron changes state synchronously on each tick of the time-scale $t = 0, 1, 2, 3, \ldots$. Conversely, it can be shown (see e.g. Arbib 1987, Chapter 2—that the result was essentially, though inscrutably, due to McCulloch and Pitts 1943) that any finite automaton can be simulated by a suitable network of discrete-time neurons (even those of the 'McCulloch–Pitts type' defined below). Although we can define a neural network for the very general notion of 'neuron' shown in figure B1.1.1, most artificial neurons are of the kind shown in figure B1.1.3 in which the input lines are parametrized by real numbers. The parameter attached to an input line to neuron i that comes from the output of neuron j is often denoted by w_{ij}, and is referred to by such terms as the *strength* or *synaptic weight* for the *connection* from neuron j to neuron i. Much of the study of neural computation B3.3 is then devoted to finding settings for these weights which will get a given neural network to approximate some desired behavior. The weights may either be set on the basis of some explicit design principles, or 'discovered' through the use of *learning rules* whereby the weight settings are automatically adjusted B3.3 'on the basis of experience'. But all this is meat for later chapters, and we now return to our focal aim:

introducing a number of the basic models of single neurons which 'fill in the details' in figure B1.1.3. As described in Section A1.2, there are radically different types of neurons in the human brain, and further variations in neuron types of other species. A1.2

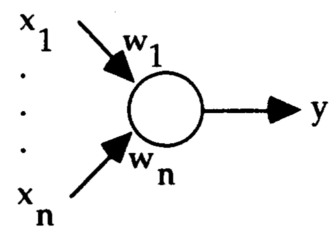

Figure B1.1.3. A neuron in which each input x_i passes through a 'synaptic weight' or 'connection strength' w_i.

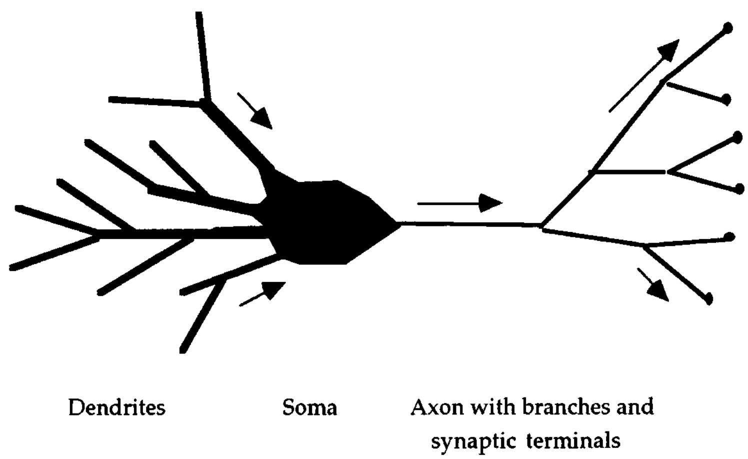

Figure B1.1.4. The 'basic' neuron. The soma and dendrites act as the input surface; the axon carries the output signals. The tips of the branches of the axon form synapses upon other neurons or upon effectors. The arrows indicate the direction of information flow from inputs to outputs.

In neural computation, the artificial neurons are designed as variations on the abstractions of brain theory and implemented in software, *VLSI*, or other media. Figure B1.1.4 indicates the main features needed to visualize biological neurons. We divide the neuron into three parts: the *dendrites*, the soma (cell body) and a long fiber called the axon whose branches form the *axonal arborization*. The soma and dendrites act as input surface for signals from other neurons and/or input devices (sensors). The axon carries 'spikes' from the neuron to other neurons and/or effectors (motors, etc). Towards a first approximation, we may think of a 'spike' as an all-or-none (binary) event; each neuron has a 'refractory period' such that at most one spike can be triggered per refractory period. The locus of interaction between an axon terminal and the cell upon which it impinges is called a *synapse*, and we say that the cell with the terminal synapses upon the cell with which the connection is made. E1.3, E1.4.3

References

Arbib M A 1987 *Brains, Machines and Mathematics* 2nd edn (Berlin: Springer)

McCulloch W S and Pitts W H 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33

