Neural Smithing

1 Introduction

Artificial neural networks are nonlinear mapping systems whose structure is loosely based on principles observed in the nervous systems of humans and animals. The major parts of a real neuron include (see figure 1.1) a branching dendritic tree that contacts and collects signals from other neurons, a cell body that integrates the signals and generates a response, and a branching axon that distributes the response to other neurons. The response of each neuron is a nonlinear function of its inputs and internal state. It is thought to be largely determined by the input connection strengths. Real neurons are more complex than this, of course, but still simple in comparison to the entire brain. The interesting idea is that massive systems of simple units linked together in appropriate ways can generate many complex and interesting behaviors.

Artificial neural networks are loosely based on these ideas. In general terms, an artificial neural network consists of a large number of simple processors linked by weighted connections. By analogy, the processing units may be called neurons. Each unit receives inputs from many other nodes and generates a single scalar output that depends only on locally available information, either stored internally or arriving via the weighted connections. The output is distributed to and acts as an input to other processing nodes.

By itself, a single processing element is not very powerful; the power of the system emerges from the combination of many units in a network. A network is specialized to implement different functions by varying the connection topology and values of the connecting weights. Depending on the connections, many complex functions can be realized. In fact, it has been shown that a sufficiently large network with an appropriate structure and properly chosen weights can approximate with arbitrary accuracy any function satisfying certain broad constraints.

In many networks, the processing units have responses like (see figure 1.1b)

$$y = f\left(\sum_k w_k x_k\right) \tag{1.1}$$

where x_k are the output signals of other nodes or external system inputs, w_k are the weights of the connecting links, and $f(\cdot)$ is a simple nonlinear function. Here, the unit computes a weighted linear combination of its inputs and passes this through the nonlinearity f to produce a scalar output. In general, f is a bounded nondecreasing nonlinear function such as the *sigmoid* function (figure 1.2)

$$f(\mu) = \frac{1}{1+e^{-\mu}}. \qquad \text{(the sigmoid)}$$

The tanh and step functions are other common choices. f is sometimes called the *squashing function* because it limits very large positive or negative values. The term *perceptron*

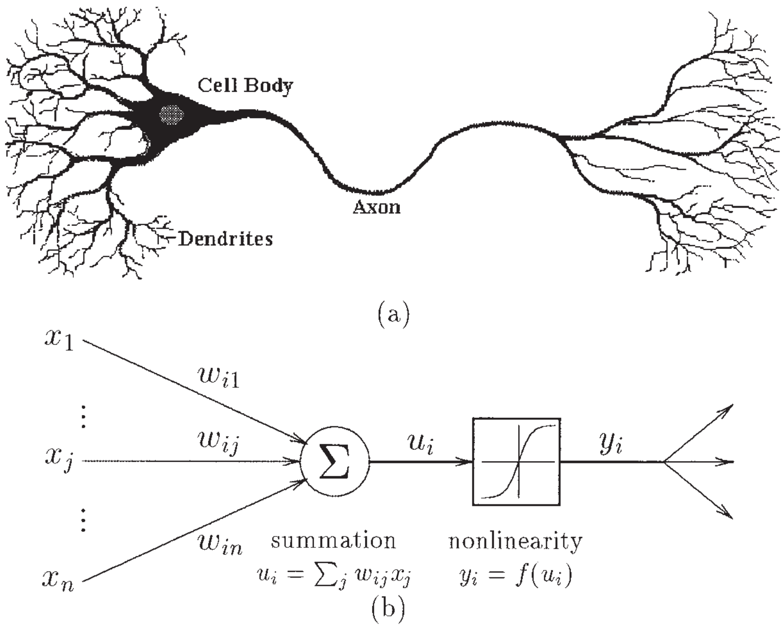

Figure 1.1
(a) A real neuron collects input signals from other neurons through a dendritic tree, integrates the information, and distributes its response to other neurons via the axon. (b) An artificial neuron model.

is commonly used to refer to any feedforward network of nodes with responses like equation 1.1.

A network can have arbitrary structure, but layered architectures are very popular. The multilayer perceptron (figure 1.3) is widely used. In this structure, units are arranged in layers and connected so that units in layer L receive inputs from the preceding layer $L-1$ and send outputs to the following layer $L+1$. External inputs are applied at the first layer and system outputs are taken at the last layer. Internal layers not observable from the inputs or outputs are called *hidden* layers. The simplest networks have just one active layer, the output units. (By convention, inputs are not counted as an active layer since they do no processing.) Single-layer networks are less powerful than multilayer circuits so applications are relatively limited.

The network in figure 1.3 has a *feedforward* structure, meaning there are no connection loops that would allow outputs to feed back to their inputs (perhaps indirectly) and change the output at a later time. The network implements a static mapping that depends only on its present inputs and is independent of previous system states. Although recurrent networks

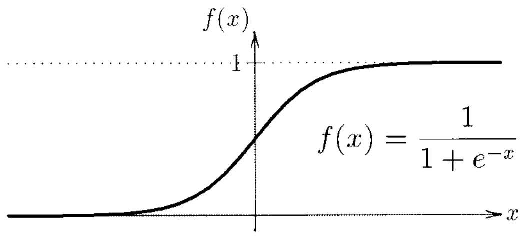

Figure 1.2
The sigmoid is a continuous, bounded, monotonic function of its input x. It saturates at 0 for large negative inputs and at 1 for large positive inputs. Near zero, it is approximately linear.

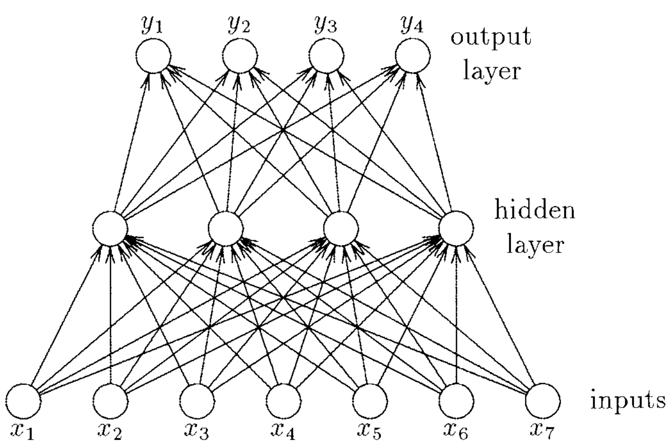

Figure 1.3
A layered feedforward network.

with feedback have a wider range of possible behaviors, analysis and training are more difficult. This book focuses on feedforward models, which form an important subclass in themselves.

The network in figure 1.3 is also *fully connected*. That is, every node in layer L receives inputs from every node in the preceding layer $L-1$ and projects outputs to every node in the following layer $L+1$.

This model is, of course, a drastically simplified approximation of real nervous systems. This isn't the place to dwell on its limitations; the model captures certain major characteristics which are thought to be important and undoubtedly ignores others. A few differences between multilayer perceptrons and real nervous systems are listed in table 1.1. More