where $h^{(l)}$ denotes the activation function associated with layer l, and $\mathbf{W}^{(l)}$ denotes the corresponding matrix of weight and bias parameters. Also, $\mathbf{z}^{(0)} = \mathbf{x}$ represents the input vector and $\mathbf{z}^{(L)} = \mathbf{y}$ represents the output vector.

Note that there has been some confusion in the literature regarding the terminology for counting the number of layers in such networks. Thus, the network in Figure 6.9 is sometimes described as a three-layer network (which counts the number of layers of units and treats the inputs as units) or sometimes as a single-hidden-layer network (which counts the number of layers of hidden units). We recommend a terminology in which Figure 6.9 is called a two-layer network, because it is the number of layers of learnable weights that is important for determining the network properties.

We have seen that a network of the form shown in Figure 6.9, having two layers of learnable parameters, has universal approximation capabilities. However, networks with more than two layers can sometimes represent a given function with far fewer parameters than a two-layer network. Montúfar *et al.* (2014) show that the network function divides the input space into a number of regions that is exponential in the depth of the network, but which is only polynomial in the width of the hidden layers. To represent the same function using a two-layer network would require an exponential number of hidden units.

6.3.1 Hierarchical representations

Although this is an interesting result, a more compelling reason to explore deep neural networks is that the network architecture encodes a particular form of inductive bias, namely that the outputs are related to the input space through a hierarchical representation. A good example is the task of recognizing objects in images. The relationship between the pixels of an image and a high-level concept such as 'cat' is highly complex and nonlinear, and would be an extremely challenging problem for a two-layer network. However, a deep neural network can learn to detect low-level features, such as edges, in the early layers, and can then combine these in subsequent layers to make higher-level features such as eyes or whiskers, which in turn can be combined in later layers to detect the presence of a cat. This can be viewed as a *compositional* inductive bias, in which higher-level objects, such as a cat, are composed of lower-level objects, such as eyes, which in turn have yet lower-level elements such as edges. We can also think of this in reverse by considering the process of generating an image starting with low-level features such as edges, then combining these to form simple shapes such as circles, and then combining those in turn to form higher-level objects such as cats. At each stage there are many ways to combine different components, giving an exponential gain in the number of possibilities with increasing depth.

Chapter 10

6.3.2 Distributed representations

Neural networks can take advantage of another form of compositionality called a *distributed representation*. Conceptually, each unit in a hidden layer can be thought of as representing a 'feature' at that level of the network, with a high value of the