activation indicating that the corresponding feature is present and a low value indicating its absence. With M units in a given layer, such a network can represent M different features. However, the network could potentially learn a different representation in which *combinations* of hidden units represent features, thereby potentially allowing a hidden layer with M units to represent 2^M different features, growing exponentially with the number of units. Consider, for example, a network designed to process images of faces. Each particular face image may or may not have glasses, it may or may not have a hat, and it may or may not have a beard, leading to eight different combinations. Although this could be represented by eight units each of which 'turns on' when it detects the corresponding combination, it could also be represented more compactly by just three units, one for each attribute. These can be present independently of each other (although statistically their presence is likely to be correlated to some degree). Later, we will explore in detail the kinds of internal representations that deep learning networks discover for themselves during training.

Chapter 10

6.3.3 Representation learning

We can view the successive layers of a deep neural network as performing transformations of the data, that make it easier to solve the desired task or tasks. For example, a neural network that successfully learns to classify skin lesions as benign or malignant must have learned to transform the original image data into a new space, represented by the outputs of the final layer of hidden units, such that the final layer of the network can distinguish the two classes. This final layer can be viewed as a simple linear classifier, and so in the representation of the last hidden layer, the two classes must be well separated by a linear surface. This ability to discover a nonlinear transformation of the data that makes subsequent tasks easier to solve is called *representation learning* (Bengio, Courville, and Vincent, 2012). The learned representation, sometimes called the *embedding space*, is given by the outputs of one of the hidden layers of the network, so that any input vector, either from the training set or from some new data set, can be transformed into this representation by forward propagation through the network.

Section 1.1.1

Representation learning is especially powerful because it allows us to exploit unlabelled data. Often it is easy to collect a large quantity of unlabelled data, but acquiring the associated labels may be more difficult. For example, a video camera on a vehicle can gather large numbers of images of urban scenes as the vehicle is driven around a city, but taking those images and identifying relevant objects, such as pedestrians and road signs, would require expensive and time-consuming human labelling.

Learning from unlabelled data is called *unsupervised learning*, and many different algorithms have been developed to do this. For example, a neural network can be trained to take images as input and to create the same images as the output. To make this into a non-trivial task, the network may use hidden layers with fewer units than the number of pixels in the image, thereby forcing the network to learn some kind of compression of the images. Only unlabelled data is needed because each image in the training set acts as both the input vector and the target vector. Such networks are known as *autoencoders*. The goal is that this type of training will force the network

Section 19.1