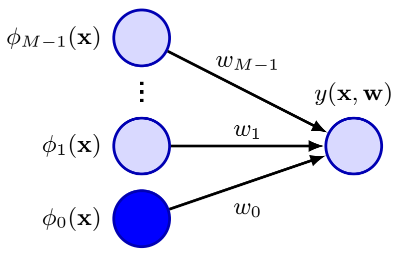

Figure 4.1 The linear regression model (4.3) can be expressed as a simple neural network diagram involving a single layer of parameters. Here each basis function $\phi_j(\mathbf{x})$ is represented by an input node, with the solid node representing the 'bias' basis function ϕ_0, and the function $y(\mathbf{x}, \mathbf{w})$ is represented by an output node. Each of the parameters w_j is shown by a line connecting the corresponding basis function to the output.

Before the advent of deep learning it was common practice in machine learning to use some form of fixed pre-processing of the input variables $\mathbf{x}$, also known as *feature extraction*, expressed in terms of a set of basis functions $\{\phi_j(\mathbf{x})\}$. The goal was to choose a sufficiently powerful set of basis functions that the resulting learning task could be solved using a simple network model. Unfortunately, it is very difficult to hand-craft suitable basis functions for anything but the simplest applications. Deep learning avoids this problem by learning the required nonlinear transformations of the data from the data set itself.

Chapter 1

We have already encountered an example of a regression problem when we discussed curve fitting using polynomials. The polynomial function (1.1) can be expressed in the form (4.3) if we consider a single input variable x and if we choose basis functions defined by $\phi_j(x) = x^j$. There are many other possible choices for the basis functions, for example

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\} \tag{4.4}$$

where the μ_j govern the locations of the basis functions in input space, and the parameter s governs their spatial scale. These are usually referred to as 'Gaussian' basis functions, although it should be noted that they are not required to have a probabilistic interpretation. In particular the normalization coefficient is unimportant because these basis functions will be multiplied by learnable parameters w_j.

Another possibility is the sigmoidal basis function of the form

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \tag{4.5}$$

where $\sigma(a)$ is the logistic sigmoid function defined by

$$\sigma(a) = \frac{1}{1+\exp(-a)}. \tag{4.6}$$

Exercise 4.3

Equivalently, we can use the $\tanh$ function because this is related to the logistic sigmoid by $\tanh(a) = 2\sigma(2a) - 1$, and so a general linear combination of logistic sigmoid functions is equivalent to a general linear combination of $\tanh$ functions in the sense that they can represent the same class of input–output functions. These various choices of basis function are illustrated in Figure 4.2.

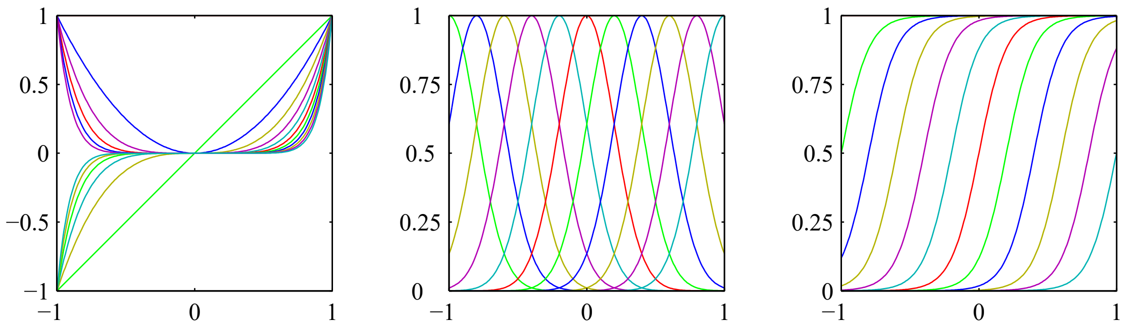

Figure 4.2 Examples of basis functions, showing polynomials on the left, Gaussians of the form (4.4) in the centre, and sigmoidal basis functions of the form (4.5) on the right.

Yet another possible choice of basis function is the Fourier basis, which leads to an expansion in sinusoidal functions. Each basis function represents a specific frequency and has infinite spatial extent. By contrast, basis functions that are localized to finite regions of input space necessarily comprise a spectrum of different spatial frequencies. In signal processing applications, it is often of interest to consider basis functions that are localized in both space and frequency, leading to a class of functions known as *wavelets* (Ogden, 1997; Mallat, 1999; Vidakovic, 1999). These are also defined to be mutually orthogonal, to simplify their application. Wavelets are most applicable when the input values live on a regular lattice, such as the successive time points in a temporal sequence or the pixels in an image.

Most of the discussion in this chapter, however, is independent of the choice of basis function set, and so we will not specify the particular form of the basis functions, except for numerical illustration. Furthermore, to keep the notation simple, we will focus on the case of a single target variable t, although we will briefly outline the modifications needed to deal with multiple target variables.

Section 4.1.7

4.1.2 Likelihood function

Section 1.2

We solved the problem of fitting a polynomial function to data by minimizing a sum-of-squares error function, and we also showed that this error function could be motivated as the maximum likelihood solution under an assumed Gaussian noise model. We now return to this discussion and consider the least-squares approach, and its relation to maximum likelihood, in more detail.

As before, we assume that the target variable t is given by a deterministic function $y(\mathbf{x}, \mathbf{w})$ with additive Gaussian noise so that

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \tag{4.7}$$

where ϵ is a zero-mean Gaussian random variable with variance σ^2. Thus, we can write

$$p(t|\mathbf{x}, \mathbf{w}, \sigma^2) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \sigma^2). \tag{4.8}$$