way that is specific to each application. For many years, this was the mainstream approach in machine learning. Basis functions, often called *features*, would be determined by a combination of domain knowledge and trial-and-error. However, this approach met with limited success and was superseded by data-driven approaches in which basis functions are learned from the training data. Domain knowledge still plays a role in modern machine learning, but at a more qualitative level in designing network architectures where it can capture appropriate *inductive bias*, as we will see in later chapters.

Section 9.1

Since data in a high-dimensional space may be confined to a low-dimensional manifold, we do not need basis functions that densely fill the whole input space, but instead we can use basis functions that are themselves associated with the data manifold. One way to do this is to have one basis function associated with each data point in the training set, which ensures that the basis functions are automatically adapted to the underlying data manifold. An example of such a model is that of *radial basis functions* (Broomhead and Lowe, 1988), which have the property that each basis function depends only on the radial distance (typically Euclidean) from a central vector. If the basis centres are chosen to be the input data values $\{\mathbf{x}_n\}$ then there is one basis function $\phi_n(\mathbf{x})$ for each data point, which will therefore capture the whole of the data manifold. A typical choice for a radial basis function is

$$\phi_n(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{s^2}\right) \tag{6.6}$$

where s is a parameter controlling the width of the basis function. Although it can be quick to set up such a model, a major problem with this technique is that it becomes computationally unwieldy for large data sets. Moreover, the model needs careful regularization to avoid severe over-fitting.

A related approach, called a *support vector machine* or SVM (Vapnik, 1995; Schölkopf and Smola, 2002; Bishop, 2006), addresses this by again defining basis functions that are centred on each of the training data points and then selecting a subset of these automatically during training. As a result, the effective number of basis functions in the resulting models is generally much smaller than the number of training points, although it is often still relatively large and typically increases with the size of the training set. Support vector machines also do not produce probabilistic outputs, and they do not naturally generalize to more than two classes. Methods such as radial basis functions and support vector machines have been superseded by deep neural networks, which are much better at exploiting very large data sets efficiently. Moreover, as we will see later, neural networks are able to learn deep *hierarchical* representations, which are crucial to achieving high prediction accuracy in more complex applications.