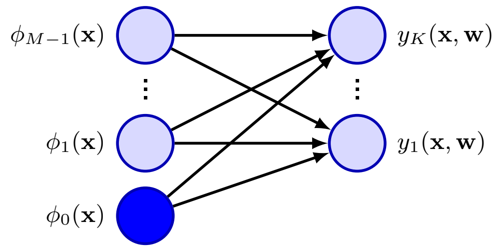

Figure 5.16 Representation of a multi-class linear classification model as a neural network having a single layer of connections. Each basis function is represented by a node, with the solid node representing the 'bias' basis function ϕ_0, whereas each output $y_1, \ldots, y_N$ is also represented by a node. The links between the nodes represent the corresponding weight and bias parameters.

where $y_{nk} = y_k(\boldsymbol{\phi}_n)$, and $\mathbf{T}$ is an $N \times K$ matrix of target variables with elements t_{nk}. Taking the negative logarithm then gives

$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}, \tag{5.80}$$

which is known as the *cross-entropy* error function for the multi-class classification problem.

We now take the gradient of the error function with respect to one of the parameter vectors $\mathbf{w}_j$. Making use of the result (5.78) for the derivatives of the softmax function, we obtain

Exercise 5.22

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\, \boldsymbol{\phi}_n \tag{5.81}$$

where we have made use of $\sum_k t_{nk} = 1$. Again, we could optimize the parameters through stochastic gradient descent.

Chapter 7

Once again, we see the same form arising for the gradient as was found for the sum-of-squares error function with the linear model and for the cross-entropy error with the logistic regression model, namely the product of the error $(y_{nj} - t_{nj})$ times the basis function activation $\boldsymbol{\phi}_n$. These are examples of a more general result that we will explore later.

Section 5.4.6

Linear classification models can be represented as single-layer neural networks as shown in Figure 5.16. If we consider the derivative of the error function with respect to a weight w_{ik}, which links basis function $\phi_i(\mathbf{x})$ to output unit t_k, we have from (5.81)

$$\frac{\partial E(\mathbf{w}_1, \ldots, \mathbf{w}_K)}{\partial w_{ij}} = \sum_{n=1}^{N} (y_{nk} - t_{nk})\, \phi_i(\mathbf{x}_n). \tag{5.82}$$

Comparing this with Figure 5.16, we see that, for each data point n this gradient takes the form of the output of the basis function at the input end of the weight link with the 'error' $(y_{nk} - t_{nk})$ at the output end.