value t has been flipped to the wrong value (Opper and Winther, 2000). Here ϵ may be set in advance, or it may be treated as a hyperparameter whose value is inferred from the data.

If we have K separate binary classifications to perform, then we can use a network having K outputs each of which has a logistic-sigmoid activation function. Associated with each output is a binary class label $t_k \in \{0, 1\}$, where $k = 1, \ldots, K$. If we assume that the class labels are independent, given the input vector, then the conditional distribution of the targets is

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \prod_{k=1}^{K} y_k(\mathbf{x}, \mathbf{w})^{t_k} \left[1 - y_k(\mathbf{x}, \mathbf{w})\right]^{1-t_k}. \tag{6.34}$$

Taking the negative logarithm of the corresponding likelihood function then gives the following error function:

Exercise 6.13

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \{t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln(1 - y_{nk})\} \tag{6.35}$$

where y_{nk} denotes $y_k(\mathbf{x}_n, \mathbf{w})$. Again, the derivative of the error function with respect to the pre-activation for a particular output unit takes the form (6.31), just as in the regression case.

Exercise 6.14

6.4.3 multiclass classification

Finally, we consider the standard multiclass classification problem in which each input is assigned to one of K mutually exclusive classes. The binary target variables $t_k \in \{0, 1\}$ have a 1-of-K coding scheme indicating the class, and the network outputs are interpreted as $y_k(\mathbf{x}, \mathbf{w}) = p(t_k = 1|\mathbf{x})$, leading to the error function (5.80), which we reproduce here:

Section 5.1.3

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}). \tag{6.36}$$

The output-unit activation function, which corresponds to the canonical link, is given by the softmax function:

Section 5.4.4

$$y_k(\mathbf{x}, \mathbf{w}) = \frac{\exp(a_k(\mathbf{x}, \mathbf{w}))}{\sum_j \exp(a_j(\mathbf{x}, \mathbf{w}))}, \tag{6.37}$$

which satisfies $0 \leqslant y_k \leqslant 1$ and $\sum_k y_k = 1$. Note that the $y_k(\mathbf{x}, \mathbf{w})$ are unchanged if a constant is added to all of the $a_k(\mathbf{x}, \mathbf{w})$, causing the error function to be constant for some directions in weight space. This degeneracy is removed if an appropriate regularization term is added to the error function. Once again, the derivative of the error function with respect to the pre-activation for a particular output unit takes the familiar form (6.31).

Chapter 9

Exercise 6.15