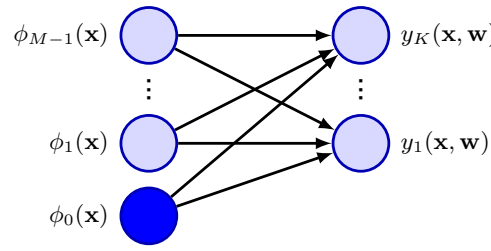

Figure 4.4 Representation of a linear regression model as a neural network having a single layer of connections. Each basis function is represented by a node, with the solid node representing the 'bias' basis function ϕ_0. Likewise each output $y_1, \ldots, y_K$ is represented by a node. The links between the nodes represent the corresponding weight and bias parameters.

4.1.7 Multiple outputs

So far, we have considered situations with a single target variable t. In some applications, we may wish to predict $K > 1$ target variables, which we denote collectively by the target vector $\mathbf{t} = (t_1, \ldots, t_K)^{\mathrm{T}}$. This could be done by introducing a different set of basis functions for each component of $\mathbf{t}$, leading to multiple, independent regression problems. However, a more common approach is to use the same set of basis functions to model all of the components the target vector so that

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) \tag{4.28}$$

where $\mathbf{y}$ is a K-dimensional column vector, $\mathbf{W}$ is an $M \times K$ matrix of parameters, and $\boldsymbol{\phi}(\mathbf{x})$ is an M-dimensional column vector with elements $\phi_j(\mathbf{x})$ with $\phi_0(\mathbf{x}) = 1$ as before. Again, this can be represented as a neural network having a single layer of parameters, as shown in Figure 4.4.

Suppose we take the conditional distribution of the target vector to be an isotropic Gaussian of the form

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \sigma^2) = \mathcal{N}(\mathbf{t}|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \sigma^2\mathbf{I}). \tag{4.29}$$

If we have a set of observations $\mathbf{t}_1, \ldots, \mathbf{t}_N$, we can combine these into a matrix $\mathbf{T}$ of size $N \times K$ such that the nth row is given by $\mathbf{t}_n^{\mathrm{T}}$. Similarly, we can combine the input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ into a matrix $\mathbf{X}$. The log likelihood function is then given by

$$\begin{aligned}\ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \sigma^2) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \sigma^2\mathbf{I}) \\ &= -\frac{NK}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left\|\mathbf{t}_n - \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\|^2. \end{aligned} \tag{4.30}$$

As before, we can maximize this function with respect to $\mathbf{W}$, giving

$$\mathbf{W}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T} \tag{4.31}$$

where we have combined the input feature vectors $\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)$ into a matrix $\boldsymbol{\Phi}$. If we examine this result for each target variable t_k, we have

$$\mathbf{w}_k = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}_k = \boldsymbol{\Phi}^{\dagger}\mathbf{t}_k \tag{4.32}$$

where $\mathbf{t}_k$ is an N-dimensional column vector with components t_{nk} for $n = 1, \ldots N$. Thus, the solution to the regression problem decouples between the different target variables, and we need compute only a single pseudo-inverse matrix $\mathbf{\Phi}^\dagger$, which is shared by all the vectors $\mathbf{w}_k$.

The extension to general Gaussian noise distributions having arbitrary covariance matrices is straightforward. Again, this leads to a decoupling into K independent regression problems. This result is unsurprising because the parameters $\mathbf{W}$ define only the mean of the Gaussian noise distribution, and we know that the maximum likelihood solution for the mean of a multivariate Gaussian is independent of the covariance. From now on, we will therefore consider a single target variable t for simplicity.

Exercise 4.7

Section 3.2.7

4.2. Decision theory

We have formulated the regression task as one of modelling a conditional probability distribution $p(t|\mathbf{x})$, and we have chosen a specific form for the conditional probability, namely a Gaussian (4.8) with an $\mathbf{x}$-dependent mean $y(\mathbf{x}, \mathbf{w})$ governed by parameters $\mathbf{w}$ and with variance given by the parameter σ^2. Both $\mathbf{w}$ and σ^2 can be learned from data using maximum likelihood. The result is a *predictive distribution* given by

$$p(t|\mathbf{x}, \mathbf{w}_{\mathrm{ML}}, \sigma^2_{\mathrm{ML}}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}_{\mathrm{ML}}), \sigma^2_{\mathrm{ML}}). \tag{4.33}$$

The predictive distribution expresses our uncertainty over the value of t for some new input $\mathbf{x}$. However, for many practical applications we need to predict a specific value for t rather than returning an entire distribution, particularly where we must take a specific action. For example, if our goal is to determine the optimal level of radiation to use for treating a tumour and our model predicts a probability distribution over radiation dose, then we must use that distribution to decide the specific dose to be administered. Our task therefore breaks down into two stages. In the first stage, called the *inference* stage, we use the training data to determine a predictive distribution $p(t|\mathbf{x})$. In the second stage, known as the *decision* stage, we use this predictive distribution to determine a specific value $f(\mathbf{x})$, which will be dependent on the input vector $\mathbf{x}$, that is optimal according to some criterion. We can do this by minimizing a *loss function* that depends on both the predictive distribution $p(t|\mathbf{x})$ and on f.

Intuitively we might choose the mean of the conditional distribution, so that we would use $f(\mathbf{x}) = y(\mathbf{x}, \mathbf{w}_{\mathrm{ML}})$. In some cases this intuition will be correct, but in other situations it can give very poor results. It is therefore useful to formalize this so that we can understand when it applies and under what assumptions, and the framework for doing this is called *decision theory*.

Suppose that we choose a value $f(\mathbf{x})$ for our prediction when the true value is t. In doing so, we incur some form of penalty or cost. This is determined by a *loss*, which we denote $L(t, f(\mathbf{x}))$. Of course, we do not know the true value of t, so instead of minimizing L itself, we minimize the average, or expected, loss which is