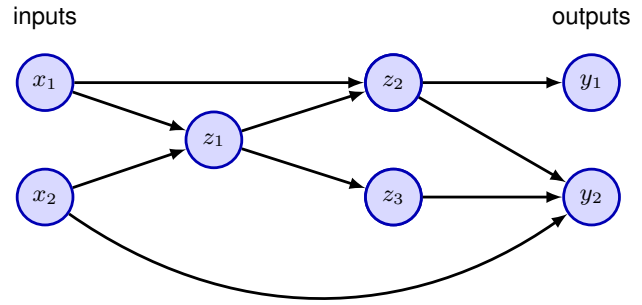


**Figure 6.15** Example of a neural network having a general feed-forward topology. Note that each hidden and output unit has an associated bias parameter (omitted for clarity).



### 6.3.7 Tensors

We see that linear algebra plays a central role in neural networks, with quantities such as data sets, activations, and network parameters represented as scalars, vectors, and matrices. However, we also encounter variables of higher dimensionality. Consider, for example, a data set of  $N$  colour images each of which is  $I$  pixels high and  $J$  pixels wide. Each pixel is indexed by its row and column within the image and has red, green, and blue values. We have one such value for each image in the data set, and so we can represent a particular intensity value by a four-dimensional array  $\mathbf{X}$  with elements  $x_{ijkn}$  where  $i \in \{1, \dots, I\}$  and  $j \in \{1, \dots, J\}$  index the row and column within the image,  $k \in \{1, 2, 3\}$  indexes the red, green, and blue intensities, and  $n \in \{1, \dots, N\}$  indexes the particular image within the data set. These higher-dimensional arrays are called *tensors* and include scalars, vectors, and matrices as special cases. We will see many examples of such tensors when we discuss more sophisticated neural network architectures later in the book. Massively parallel processors such as GPUs are especially well suited to processing tensors.

## 6.4. Error Functions

Chapter 4

Chapter 5

In earlier chapters, we explored linear models for regression and classification, and in the process we derived suitable forms for the error functions along with corresponding choices for the output-unit activation function. The same considerations for choosing an error function apply for multilayer neural networks, and so for convenience, we will summarize the key points here.

### 6.4.1 Regression

Section 2.3.4

We start by discussing regression problems, and for the moment we consider a single target variable  $t$  that can take any real value. Following the discussion of regression in single-layer networks, we assume that  $t$  has a Gaussian distribution with an  $\mathbf{x}$ -dependent mean, which is given by the output of the neural network, so that

$$p(t|\mathbf{x}, \mathbf{w}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \sigma^2) \quad (6.23)$$