Studies on machine learning have mainly been concerned with automatic learning from examples to develop the knowledge describing these examples. This is clearly different from the kind of learning as learning to ride a bicycle. In supervised learning, each example used is typically described by a number of attributes. The attributes are divided into inputs and outputs, and the learning process is to develop a model mapping the multiple inputs and outputs. The model is gradually refined during learning to minimise the errors between the predictions and real values of outputs, i.e., so-called supervised learning. The most widely studied supervised learning approach is the feedforward neural network (FFNN). The FFNN model and its application to process operational support will be introduced in this Chapter. The discussion on FFNN will be focused on many of the practical issues that have to be considered in applying FFNN. While the focus will be on FFNN, other supervised models will also be described and compared with FFNN. These include fuzzy FFNN, fuzzy set covering approach and fuzzy signed digraph.

5.1 Feedforward Neural Networks

5.5.1 FFNN Architecture

There are already a large number of textbooks on FFNNs. Here it is introduced less technically. Simply speaking, a FFNN neural network is an algorithm or computer software that can learn to identify the complex nonlinear relationship between multiple inputs and outputs. The learning process has a number of characteristics.
Firstly, FFNN does not need fundamental domain problem models and is easy to be set up and trained. This is different from conventional statistical methods that usually require the user to specify the functions over which the data is to be regressed. In order to specify the function, the user has to know the forms of the equations governing the correlations between the data. If these functions are incorrectly specified, the data will not be satisfactorily regressed. Furthermore, considerable mathematics and numerical experience is required to obtain convergence if these equations are highly nonlinear. FFNN does not need to specify the forms of the correlations as well as any mathematical and numerical expertise requirements. Secondly, data examples used for training are allowed to be imprecise or noisy, in some cases even incomplete. Thirdly it mimics the human learning process: learning from examples through repeatedly updating the performance.

A FFNN neural network consists of a number of processing elements called neurons. These neurons are divided into layers. Figure 5.1 shows a three layered FFNN architecture including an input, a hidden and an output layer. Typically the input layer nodes correspond to input variables and the output layer to output variables. Hidden neurons do not have physical meanings. Neurons between two adjacent layers are fully connected by branches.

Each neuron in the hidden and output layer is described by a transfer function (or activation function). Usually a sigmoidal function is used,

\[ f(z) = \frac{1}{1 + e^{-az}} \]  

(5.1)

\( f(z) \) transforms an input \( z \) to the neuron to the range of \([0.0, 1.0]\) as shown in Figure 5.2(a). The parameter \( a \) in Equation 5.1 is used to change the shape of the