Because of the artistic nature of neural net modeling, it is difficult for novice data miners to use many implementations of neural nets successfully. But there are some implementations that are highly automated, permitting even novice data miners to use them effectively.

The type of neural net described here is sometimes called a Multilayer Perceptron (MLP).

**Additional Types of Neural Networks**

Following are some additional types of neural nets:

- **Linear Networks**: These networks have two layers: input and output layers. They do not handle complexities well but can be considered as a “baseline model.”

- **Bayesian Networks**: Networks that employ Bayesian probability theory which can be used to control model complexity, and can be used to optimize weight decay rates, and to automatically find the most important input variables.

- **Probabilistic Networks**: These networks consist of three to four layers.

- **Generalized Regression**: These networks train quickly but execute slowly. Probabilistic (PNN) and Generalized Regression (GRNN) neural networks operate in a manner similar to that of Nearest-Neighbor algorithms (see Chapter 12), except the PNN operates only with categorical target variables and the GRNN operates only with numerical target variables. PNN and GRNN networks have advantages and disadvantages compared to MLP networks (adapted from http://www.dtreg.com/pnn.htm):
  - It is usually much faster to train a PNN/GRNN network than an MLP network.
  - PNN/GRNN networks often are more accurate than MLP networks.
  - PNN/GRNN networks are relatively insensitive to outliers (wild points).
  - PNN networks generate accurate predicted target probability scores.
  - PNN networks approach Bayes optimal classification.
  - PNN/GRNN networks are slower than MLP networks at classifying new cases.
  - PNN/GRNN networks require more memory space to store the model.

- **Kohonen**: This type of neural network is used for classification. It is sometimes called a “self-organizing” neural net. It iteratively classifies inputs, until the combined difference between classes is maximized. This algorithm can be used as a simple way to cluster data, if the number of cases or categories is not particularly large. For data sets with a large number of categories, training the network can take a very long time.

MLPs can be used to solve most logical problems, but only those in which the classes are *linearly separable*. Figure 7.14 shows a classification problem in which it is possible to separate the classes with a straight line in the space defined by their dimensions.

Figure 7.15 shows two classes that cannot be separated with a straight line (i.e., are not linearly separable).