Market modelling · 12
Data mining models - 4

Chuck Chakrapani

The MLP models
In the April issue, we briefly touched upon MLP models. For the sake of convenience, I have reproduced here the figure that was used to illustrate the model (Figure 1 on the following page). In this example, we have two inputs, five hidden values and one output. Every neuron in every layer is connected to every neuron in the next layer. Processing by neurons is fairly straightforward. The neuron sums all inputs it receives and applies a function to this sum. There is no restriction on the nature of this function. It can be a straight linear function or a somewhat more complex non-linear 'squashing function'.

How they work - An illustration
In our example, we have two input values, x1 and x2. These are fed to the input neurons. The input layer processes the input values and transfers the connections to the hidden layer. The values are modified by the application of weights w1, w2, w3 ... at this stage. Where do these weights come from? As in many iterative models, these weights are simple random starting values to begin with. As the system gets 'trained', these weights are modified accordingly.

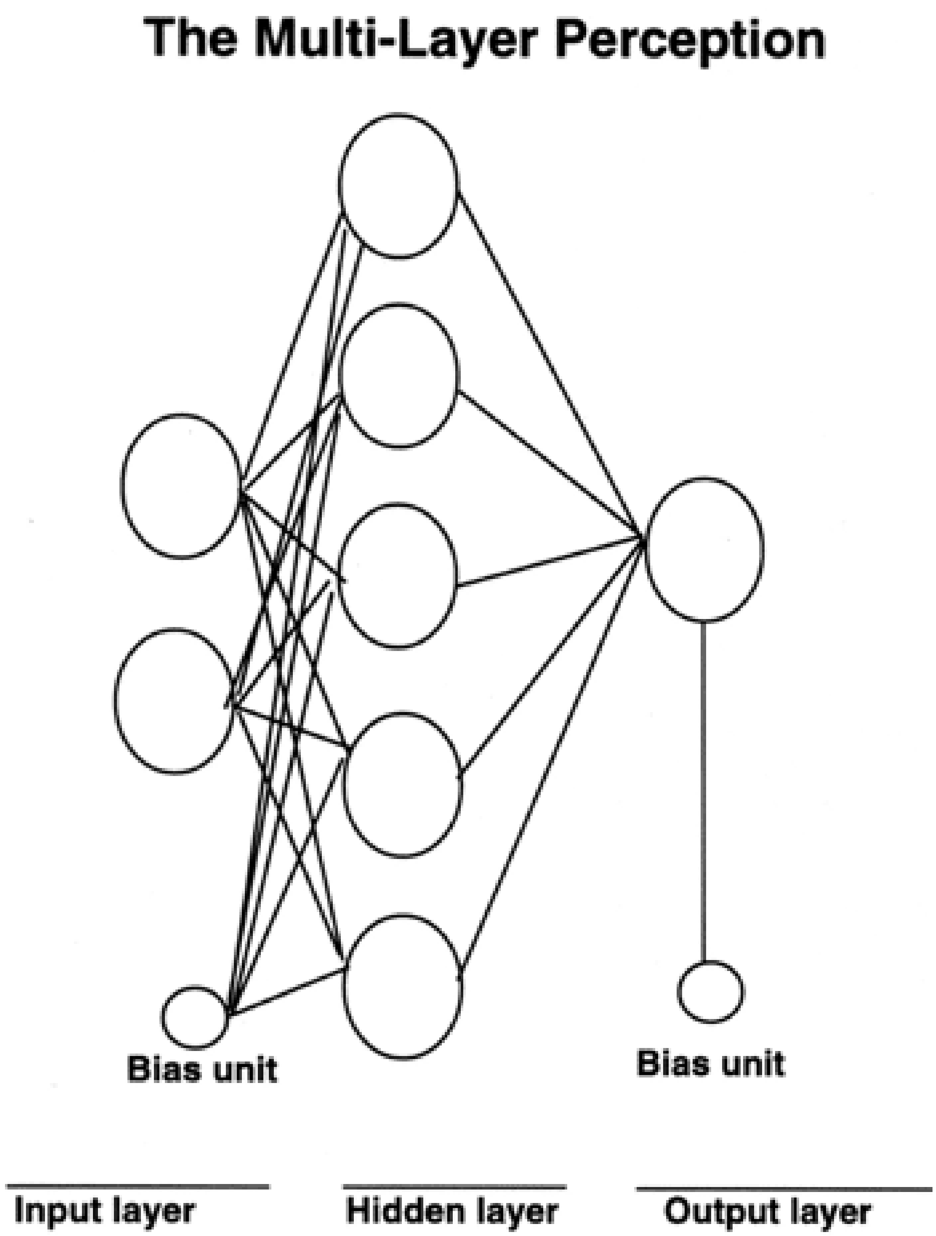
The hidden layer neurons process these modified weighted values. Once processed, these values are then transferred to the next set of layers. The processes of assigning (another set of) random values for initial weights followed by 'training' to modify the initial weights are carried out at this stage. We may call these weights u1, u2, u3 ... etc, to distinguish them from the weights w1, w2, w3 ... assigned at the previous stage. The weighted values are then transferred to the output neurons.

At this stage, the output neuron applies an additional process to compute the output value Y1.

The role of bias units
In our figure, there were additional items - bias units. Bias units output a constant value. This is conceptually equal to the constant term obtained in the equations describing the processes carried out by the network. Another way of looking at the bias term is to consider it the equivalent of the error term in equations.

Validating the network
The process of network training involves adjusting the connection weights in order to make the network reproduce the known outputs from the known inputs. A criterion can then be set to measure the success of training. For instance, one can consider the difference between the observed and the expected output. If we use this as a criterion, then we would consider the error mean square (the squared difference between the observed and expected values) to be the central measure. Our purpose here would be to create weights such that the mean squared errors obtained at the end is a minimum.

Figure 1: Multi-layer perception



The above example consists of three layers of perception. Every neuron in each layer is connected to every neuron in the next layer. This is the common pattern in MLP models, which are becoming increasingly popular in commercial applications.

Training

When networks are large, the computations involved can be considerable. The main reason is that when we 'train' the network, we need to iterate to obtain weights at different levels. This is akin to our acquiring a motor skill (such as walking or bicycling) which involves doing something, learning from our mistakes, redoing it and, if successful, learning it, repeating it to see if we get the desired results and if the behaviour continues to be successful, making it a part of our learned behaviour. The way we learn motor skills is very different from the way we follow instructions. Motor skills take longer to learn since such learning comes mostly from doing rather than simply from following instructions. Similarly, neural networks can consume considerably more computing time compared to computations that are non-iterative and do not have heuristic rules.

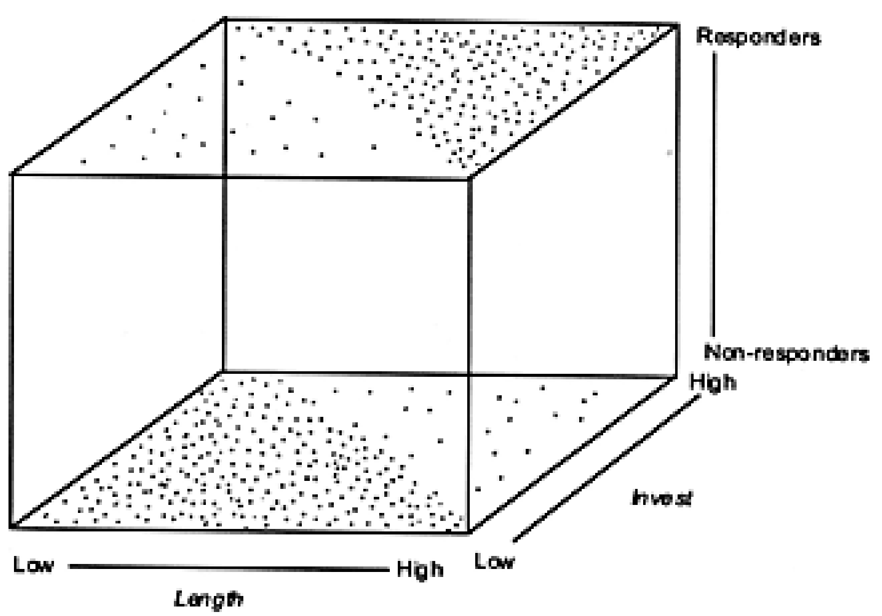
Just as learned motor skills can be reproduced at will with minimal effort, once a network is trained, we can easily and quickly calculate the output value corresponding to any pair of inputs.

What makes neural nets powerful?
The flexibility of the model is what makes it powerful. More specifically, the nonlinear, hidden layer neurons provide strong inputs for modelling the patterns hidden in data.

Another useful feature of these models is that there is no limit on the number of inputs and outputs. While in the past, computing power was a potential limitation, the increased availability of high-powered computers in recent years has practically eliminated this limitation.

Other statistical techniques
As we discussed earlier, neural nets are tackling problems that used to be tackled by other techniques. CHAID is one such technique (see the earlier articles in this series). CHAID repeatedly splits the data base to identify the connection between the input and the output variables (such as consumer demographics and purchase behaviour). Neural nets, on the other hand, tackle the same problem by estimating the connections and modifying the estimates. Some practitioners use CHAID as a preliminary tool to create an input coding of categorical variables. These coded variables are then input into a neural net.

Figure 2. The preliminary 3-D box model



Example of a neural net
Consider a bank which has a database of 2 million customers. It creates an initial mailing for a new product and mails it to some 30,000 customers. Nine hundred customers (3%) respond to this offer. Can we increase the response by a minimum of 67%? In other words, who should we target, among our customers such that we get a response rate of at least 5% instead of just 3%? The purpose is to increase the profitability by decreasing the cost of mailing to less responsive prospects. If we can accomplish this goal, the savings are obvious. For example:

Total intended mailing	200,000
Cost at $0.75/mailing	($150,000)
Response at 3%	6,000
Profit $40 per response	$240,000
Net Profit	
($240,000?$150,000)	$90,000

In the above example, net profits almost triples (from $90,000 to $250,000) because targeted mailing brought in more profit for the same cash outlay.

To continue with our example, let us assume that our bank's database is rather sparse and the only two variables that are available for all respondents are:

INVEST: The total amount a person has in the bank (all product combined).

LENGTH: The length of period in years that the customer has been with the bank.

Obviously, this is an oversimplified example. In reality there are likely to be many variables. Neural net programs do not have any difficulty in handling a large number of variables.

We note that we have 900 responders and some 29,100 non-responders. To make matters simple, we choose 900 from the non responders. We further divide the groups as follows:

	Training	Test
Responders	450	450
Non-responders	450	450

(The training set is used to create the model, while the test set is used to validate the model.)

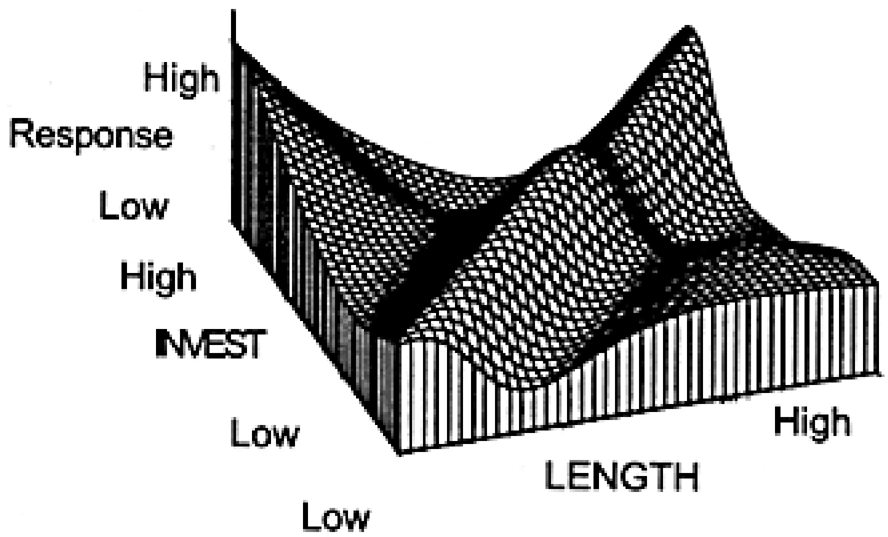
Suppose we plot the responders and non-responders on a 3-dimensional box model. It might look like Figure 2. Visual inspection shows that responders tend to be long-term customers with higher balances while non-responders tend to be relatively newer customers with lower bank balances.

The neural net can be set up such that it provides an index for each person that relates to his or her probability of responding. The index can now be plotted as a response surface as shown in Figure 3. The figure visually shows how the response level to our offer varies depending on a person's score on INVEST and LENGTH.

Note how complex the response surface is. This means that traditional linear statistical techniques such as multiple regression analysis, discriminant analysis or logistic regression will be unable to capture the complexity involved in the response pattern.

Techniques like CHAID and KnowledgeSeeker (KS) are better in this respect because they can deal with non-linearity. Neural net analysis tends to detect even subtler patterns compared to CHAID or KS.

Figure 3. Response surface



At this stage we use the test sample to assess the validity of our analysis. Let us assume that our test sample confirms the validity of our initial model. Then we can calculate the 'leverage index' which will indicate how much we gained by using a segmented mailing as opposed to an undifferentiated mailing to our customer. As we noted earlier, although both CHAID/KS type of techniques as well as neural nets can tackle the problem at hand, for many problems, the results are not equivalent. The leverage index, more often than not, for various groups is likely to be higher for the neural net solution compared to solutions generated by other, more common, methods.

Dr. Chuck Chakrapani of Standard Research Systems is a Toronto-based consultant, author and seminar leader. He works internationally.