ARTIFICIAL NEURAL NETWORKS

Boychenko I.V., *IN-43,*
Litvinenko G.I., *EL adviser*

Artificial neural networks(ANNs) have seen an explosion of interest overthe last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics. Indeed, anywhere that tereare problems of prediction, classification or control, neural networks are being introduced.

Neural networks are also intuitively appealing, based as they are on a crude low-level model of biological neural systems. In the future, the development of this neurobiological modeling may lead to genuinely intelligent computers.

Neural networks are applicable in virtually every situation in which a relationship between the predictor variables (independents, inputs) and predicted variables (dependents, outputs) exists, even when that relationship is very complex and not easy to articulate in the usual terms of "correlations" or "differences between groups."

Neural networks grew out of research in Artificial Intelligence; specifically, attempts to mimic the fault-tolerance and capacity to learn of biological neural systems by modeling the low-level structure of the brain. A typical feedfoward network has neurons (abstract analogs of biological neural cells) arranged in a distinct layered topology. The input layer is not really neural at all: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks that are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better.

When the network is executed (used), the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer, and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.

So neuron networks can therefore be used where you have some known information, and would like to infer some unknown information. Some examples are:

Stock market prediction. You know last week's stock prices and today's DOW, NASDAQ, or FTSE index; you want to know tomorrow's stock prices.

Credit assignment. You want to know whether an applicant for a loan is a good or bad credit risk. You usually know applicants' income, previous credit history, etc. (because you ask them these things).

Control. You want to know whether a robot should turn left, turn right, or move forwards in order to reach a target; you know the scene that the robot's camera is currently observing.

Needless to say, not every problem can be solved by a neural network. Many financial institutions use, or have experimented with, neural networks for stock market prediction, so it is likely that any trends predictable by neural techniques are already discounted by the market, and (unfortunately), unless you have a sophisticated understanding of that problem domain, you are unlikely to have any success there either!

Therefore, another important requirement for the use of a neural network therefore is that you know (or at least strongly suspect) that there is a relationship between the proposed known inputs and unknown outputs. This relationship may be noisy (you certainly would not expect that the factors given in the stock market prediction example above could give an exact prediction and there may be an element of pure randomness) but it must exist.

In general, if you use a neural network you won't know the exact nature of the relationship between inputs and outputs. The other key feature of neural networks is that they learn the input/output relationship through training. There are two types of training used in neural networks, with different types of networks using different types of training variables, and can subsequently be used to make predictions where the output is *not* known.