NEURAL NETWORKS SOFTWARE CRUNCHES the BIG Numbers

This new technology for quantitative analysis gives little gray cells something to think about.

by Miroslaw W. Janusz, Ph.D.

With the breakthrough of a rapidly proliferating technique in desktop quantitative analysis known as neural networks, humans are now one step closer to leaving deskwork to computers. Based on a functional analogy to human brain cells called neurons, neural networks software adds artificial intelligence to data analysis by using highly interconnected processing units working in unison to solve problems.

Like its human counterpart, a computational neuron receives input from a number of sources and responds with a single output. To respond to diverse inputs from many neurons, neural networks use algorithms that imitate elementary brain-cell functions to "learn" the structure of data, i.e., to discern patterns in the data. Software based on neural networks can then use this new "knowledge" to predict the outcome of problems involving similar data.

Somewhat surprisingly, neural networks aren't new. The idea for computational neurons arrived long before the technology to deploy it capably. The first artificial neuron was proposed in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pitts. But the two men's theory didn't find its purpose until the advent of high-speed computing.

Driven by fully automated computational processes, neural networks have the capacity to extract patterns and detect trends too complex to be observed by humans or identified by other computational techniques. This makes them extremely useful in business, engineering and scientific enterprises that must analyze large sets of complicated or imprecise data.

Why use neural networks?

Neural networks don't do anything that can't be done by traditional computing techniques, but they can readily do some tasks that would otherwise be very difficult. Everyday computer systems are good at fast arithmetic, and they're very good at using linear mathematics to solve problems—i.e., at doing precisely what a programmer programs them to do. However, they're not so good at doing what neural networks do best: interacting with noisy data and adapting to circumstances.

Know & Go

- A neural network is a system of interconnected computational "neurons," which, like the neurons in our brains, receive multiple inputs and create a single output.
- Neural networks are adaptable and have the capacity to "learn" data, to extract patterns and detect trends too complex to be observed by humans or identified by other computational techniques.
- Once a neural network learns a dataset, it can then apply its "knowledge" to new data.
- Neural networks are ideal for making predictions from large bodies of complex data.
- They can be used to create two kinds of predictions: classification and specific numerical value.

Although techniques for optimizing linear models were well-known before neural networks and artificial neurons were invented, it took many years to develop effective algorithms for training neural networks. Today, a range of sophisticated algorithms exist for neural net training, making this technology an attractive alternative to more traditional statistical methods. Neural networks have the advantageous capability of modeling extremely complex functions.

Neural networks are useful when lots of examples must be analyzed, or when a structure in these data must be analyzed, but a single algorithmic solution would be too difficult to formulate. When these conditions are present, neural networks software can solve the following basic kinds of problems:

• Classification problems in which the investigator must determine what category an unknown entity belongs in, e.g., an unidentified medical condition or the likelihood a borrower will repay a loan

• Numeric problems in which the investigator must predict a specific numeric value such as the level of sales during some future period

How neural networks work

A neuron is a computational structure that has many inputs and one output. A neural network consists of connected neurons. Each neuron performs a portion of the computations inside the network. A neuron takes numbers as inputs, performs a relatively simple computation on them, and returns an output. A neuron's output value is passed on as one of the inputs for another neuron, although this isn't the case, of course, with neurons that generate the final output values of the entire system. (See figure 1.)

Neurons are arranged in layers. Input layer neurons receive inputs for the computations. Taking an example from the snack food industry, say a company wants to use its processing line to make uniform, consistent corn chips, and it inputs data on corn type, oil content, soak time, cooking temperature and cooking time. These values are passed to the neurons in the first hidden layer, which perform computations on their inputs and pass their outputs to another hidden layer, if there is one. Outputs from the neurons in the last hidden layer are passed to the neuron or neurons that generate the network's final outputs. In the example above, that output could be the predicted moisture content and hardness of the prepared corn, which can then be compared against the company's manufacturing standard.

Neural networks have two modes of operation: training and use. In training mode, algorithms based on simple 1/0 rules control what kinds of input will cause each neuron to fire, creating an output pattern. After a neuron "learns" a firing pattern from a limited data set, it can use this learned pattern to detect similar patterns in a larger, unexplored body of data. If a neuron recognizes a pattern, its own output pattern will be the one associated with the pat-

tern. If it doesn't recognize the pattern, its output pattern will be the taught input pattern that's the least different from it. This response to the unfamiliar is the source of the neuron's powerful adaptability.

Because a network consists of many adaptable neurons harnessed together, it can be trained to recognize very complex patterns.

Real-world applications

The current commercial and scientific environments in which computer systems harvest enormous bodies of data are ideal for deploying neural networks. The technique is especially useful for predicting and forecasting. Because it's now accessible to many professionals who don't have high-level computing expertise, neural networks are being adopted by diverse enterprises for which sophisticated statistical processing is a critical function. These include industrial process control, risk management, data validation, customer research and income prediction.

More specific examples, grouped by field, include the following:
• Industry
• Quality control
• Six Sigma
• Beer and wine flavor prediction
• Highway maintenance programs
• Telecommunication line fault detection
• Government
• Missile targeting
• Criminal behavior prediction
• Finance
• Loan underwriting
• Credit scoring
• Credit card fraud detection
• Reserves estimation
• Investment prediction
• Energy price prediction
• Real-estate appraisal
• Science and medicine
• Specimen identification
• Protein sequencing
• Tumor and tissue diagnosis
• Heart attack diagnosis
• New drug effectiveness
• Prediction of air and sea currents
• Archaeological artifact identification and dating
• Air and water quality

Current commercial and scientific environments are ideal for deploying neural networks.

Figure 1: Diagram of a Simple Neuron
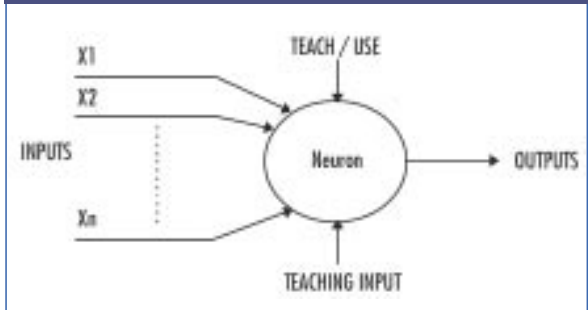
Source: Christos Stergiou and Dimitrios Siganos, Department of Computing, Imperial College of London

Neural networks at work

Palisade Corp. recently released the new neural networks software Neural-Tools, which is an add-in for Microsoft Excel. Even before NeuralTools was commercially available, the beta version was snapped up by customers and put to work on problems that eluded solution. The following two examples, one from medicine and another from the health care industry, demonstrate how the software can be used to address problems in quality control, industrial processing and operations research.

Predicting tumor diagnosis

Researchers at the Katharinenhospital in Stuttgart, Germany, use advanced data-analysis tools to diagnose tumors. Dr. José R. Iglesias-Rozas, associate professor at the Universität Tübingen and a researcher at the Laboratory of Neuro-pathology in the Institute for Pathology at Katharinenhospital, is using NeuralTools for histological classification and grading of tumors. Histological classification of tumors is based on microscopic tissue study. Tumor grading is an important aspect of diagnosis because the treatment and outcome of each case depends greatly on the assigned grade.

Quantitative diagnostic assessments in histopathology (i.e., microscopic changes in diseased tissue) frequently deal with uncertain information and vague linguistic terms. Final decisions are rarely based on the evaluation of a single diagnostic clue; rather, multiple pieces of evidence are routinely observed, and the certainty of combined evidence supports the final diagnosis. Neural networks analysis, which intelligently predicts outcomes based on multiple pieces of input data, is a natural fit for such medical diagnostic applications.

The aim of the study was to predict the degree of malignancy of tumors based on ten discrete characteristics in 786 patients. Histological secti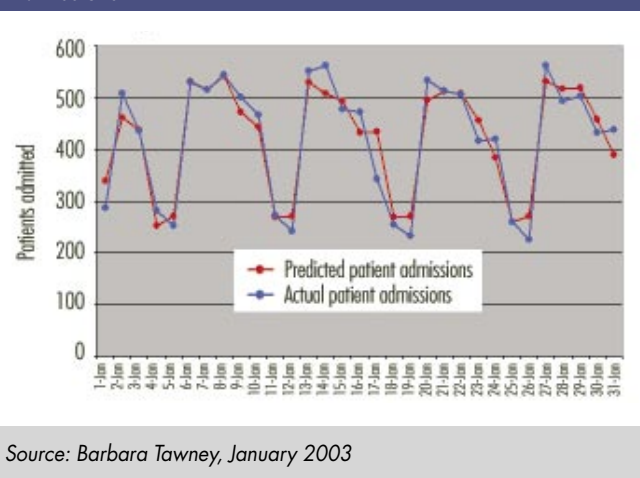ons of 786 different human brain tumors were collected. Ten histological characteristics were assessed in each case, describing the presence of a specific histological feature on a scale of zero to three, with zero being the absence of the feature, and three representing an abundant presence of the feature. NeuralTools was then used to predict a malignity coefficient between one and four.

For the NeuralTools training set, 629 tumors were available, and 157 independent cases were used as the test set. NeuralTools accurately predicted 98.58 percent of the training set cases and 95 percent of the testing set cases.

What's next for the study? "We have over 30 years of data and more than 8,000 patients with different brain tumors to assess next," Iglesias-Rozas says.

Predicting patient load

Health care industry consultant Barbara Tawney faced a tough task. She needed to forecast patient loads for the entire metropolitan hospital system of Richmond, Virginia. Every hospital has a finite number of beds and therefore a maximum capacity. But unpredictable patient demand throughout the system had resulted in two occasions when all nine hospitals in the system had reached capacity, and patients had to be diverted to facilities outside the area. To figure out how to anticipate and prepare for surges in patient load, Tawney turned to StatTools and NeuralTools.

Tawney maintains an active consulting practice and is a Ph.D. candidate in the systems and information engineering department of the University of Virginia's School of Engineering and Applied Science. She specializes in data analysis—particularly large data sets—and could have easily written her own software for the patient load project. Instead, she used StatTools to tackle the mountain of data.

With the cooperation of Virginia Health Information, a nonprofit organization that collects and warehouses all health care data statewide, she was granted limited access to metropolitan Richmond patient data for the four years from 2000 to 2003. Time-series data were derived from hospital billing information for about 600,000 patients being treated at area hospitals during 2000–2003. The patient-level data (PLD) were detailed in 78 different fields, including dates of admission and discharge, diagnosis and length of stay.

"I was looking for a user-friendly way to do autocorrelation, and a colleague recommended StatTools to me," says Tawney. She created time series for the data by "binning" the PLD according to the dates and times of activity for each case. The time-series data were analyzed for daily, weekly and event trends.

After she analyzed the historical data, Tawney predicted future patient loads, as seen in figure 2. She began by training NeuralTools on the existing data. Additional daily, weekly and event trends, along with unusual days, stood out during the analyses. For example, Tawney determined that patient load

Figure 2: Actual Patient Admissions vs. Predicted Patient Admissions

Source: Barbara Tawney, January 2003

Neural networks software can address problems in quality control, industrial processing and operations research.

It's not just what we do, it's what we do for you.

Certification can be a challenge.
SGS completes the puzzle when
choosing your registrar.

SGS IS THE WORLD'S LEADING VERIFICATION, TESTING AND CERTIFICATION COMPANY

Contact us for information on ISO 9001:2000, ISO 14001,
AS 9100, TL 9000, QS 9000, TS 16949, CE Mark, ISO 13485/88,
ISO 9000 for Healthcare, Qualicert, Second party audits,
OHSAS 18001, and our full suite of training courses.

In US, call: (800) 747-9047
In Canada, call: (800) 636-0847
www.us.sgs.com

WHEN YOU NEED TO BE SURE

SGS

peaked at midweek during most weeks of the year. Holiday periods also have a different, distinctive pattern. The number of patients entering the hospitals just before and during the Thanksgiving holiday was lower than normal but was followed on Monday by an influx of patients that stretched the facility's resources. Similarly, patient load dropped markedly throughout the double holidays of Christmas and New Year's. But each year there was a significant surge in demand on the Monday and Tuesday of the first full week of the year.

For hospital planners and administrators, Tawney's findings provide the basis for predicting patient load throughout the Richmond hospital system. These predictions range from a few days to several months. The ability to predict patient demand allows for more efficient allocation of system resources, including scheduling of services.

The software market

The number of neural networks software products available has kept pace with a growing recognition of the technique's many capabilities. Although prospective neural networks users will want to assess these products in light of their own needs and priorities, two important considerations for any customer are precision of results and ease of use. It's important to keep in mind that what a neural networks package "learns" is only as accurate as its computational processes are sensitive. Two different programs can analyze the same body of data and produce very different results. This shouldn't intimidate prospective purchasers, however. The differences between the two sets of results aren't subtle and can be interpreted even by an inexperienced analyst. It's a good idea to test trial versions of the packages under consideration and compare the results.

Ease of use has obvious benefits for the uninitiated. Although transparency of computational processes might be of interest to analysts specializing in neural networks, it's far more efficient for the average neural networks user to work without having to consider what the computer is doing behind the scenes.

About the author

Miroslaw W. Janusz, Ph.D., is a software engineer for Palisade Corp. in Ithaca, New York. Janusz holds a master's degree in computer science and a Ph.D. in philosophy from Cornell University, as well as a master's degree in advanced mathematical logic and philosophy from the University of Southern California. He has helped develop a number of applications in decision analysis, statistical analysis and optimization, and he was the primary developer for Palisade's new Excel add-in for neural networks, NeuralTools.

QD