

The power of neural networks

Brian Cogan assesses NeuroSolutions from NeuroDimension, and Neural Networks, a Mathematica add-on

For centuries, the scientific approach to understanding physical processes has consisted mainly of constructing a mathematical model of the process, solving the equations of motion, and interpreting the behaviour of the process in terms of these solutions. The mathematical models themselves are expressions of physical laws that are usually written as differential equations. These laws can be used to describe the behaviour of the process, given the initial conditions. The attempt to understand nature by describing it in mathematical terms has been astonishingly successful. The giants of the subject, such as Newton, Maxwell, and Dirac, have encapsulated fundamental truths in the equations that bear their names. Not only can known phenomena be described, but also startling and unexpected effects can be predicted, such as Hamilton's prediction of Conical Refraction, or Einstein's prediction of the bending of a beam of light as it passes a very massive body.

Despite the success of the Newtonian approach, some processes cannot be described by a physical law. For example, what natural law could take a signature as an input and have as an output 'valid' or 'invalid'? Such a relationship is not described by a law, but by some non-linear mapping or function. To find such a function, it is necessary to go beyond the traditional approach.

Neural networks are universal function approximators. They represent a new computing paradigm based on the parallel architecture of the brain. They contain artificial neurons, linked by adaptive interconnections, arranged in a massively parallel structure. They produce a weighted sum of the inputs and can be 'trained' to produce an accurate output for a given input. This training consists of adjusting the weights applied by the network as it sums the inputs. The power of neural networks lies in their ability to represent general relationships, and in their ability to learn these relationships directly from the data being modelled. There are essentially four broad categories of problems to which neural networks have applications: classification of patterns; function approximation; behaviour prediction; and data mining.

Neural networks have applications in diverse areas where patterns need to be extracted from historical data, including financial forecasting, speech recognition, and process control. One prosaic application is to look at the pattern of returns from a test mailshot and use this to direct further mailshots. Another is in handwriting recognition, where one can write on a tablet and the writing will be translated into ASCII text. There is also a developing field called Computational Neuroscience that takes inspiration from both artificial neural networks and neurophysiology, and attempts to put the two together.

NeuroDimension, Inc. produces 'NeuroSolutions', a modular, icon-based, neural network design interface. Developed in association with the Computational Neural Engineering Lab at the University of Florida, it has a wonderful graphical user interface that is very intuitive and easy to use. There are 'wizards' for all critical phases and a 'drag and drop' method for constructing the networks. This makes the package easy to learn and work with. There is an obvious (and very successful) attempt to make their products as easy to use and to understand as possible.

NeuroSolutions has two separate wizards to build a neural

network automatically. 'NeuralExpert' will automatically choose and configure a neural network for you, based on the type of problem you select and the data that you give it. 'NeuralBuilder' builds a neural network based on your design specifications. Once you choose a neural architecture you can customise parameters such as the number of hidden layers, the number of processing elements and the learning algorithm, e.g. conjugate gradients or backpropagation.

If you don't know what a parameter should be set to, you can specify that a genetic algorithm comes into operation to optimise the setting for you. A genetic algorithm searches for the best combination of parameters and inputs. It will train your neural network, while adjusting the parameters and selecting inputs based on 'the survival of the fittest'. Once the genetic training is complete, you end up with neural network settings that produced the lowest error.

NeuroSolutions also provides a set of probing tools that offer real-time access to internal network parameters, and data such as inputs and outputs, gradients, hidden states, weights and sensitivities, among others. Once a probe is attached, you can view the numerical values, display the data graphically or stream the data to a file.

NeuroSolutions supports 14 architectures including: Multilayer Perceptron, Learning Vector Quantisation, and Self-Organising Map, although you are not limited to this set of architectures as you can build your own.

The Custom Solution Wizard is an optional add-in product that will take a neural network, designed within NeuroSolutions, and transform it into a dynamic link library (DLL). This DLL can then be embedded into your own C++, Visual Basic, Excel, Access or Internet (ASP) application.

NeuroSolutions is available in six different levels. The entry level is the 'Educator' and it is intended for those who want to learn about neural networks. This level uses the most common neural network architecture, the multilayer

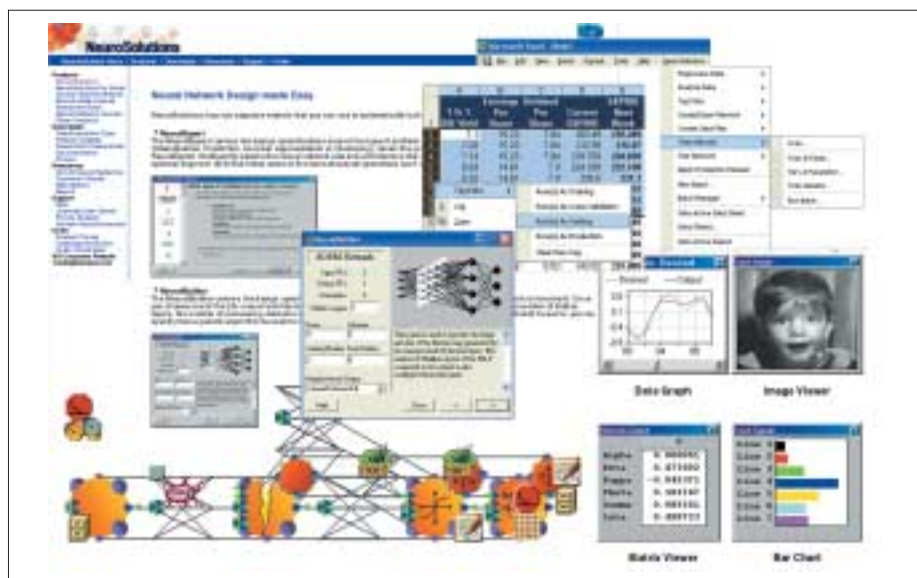
perceptron. It allows up to 50 inputs/neurons per layer and two hidden layers. The learning paradigm is backpropagation, and it also has probing capabilities and allows neural network creation with 'NeuralExpert' and 'NeuralBuilder'. The most advanced level is the 'Developers' level. Here there is no restriction on the number of inputs, outputs or hidden neurons. You can optimise neural network parameters using a genetic algorithm, and produce ANSI-compatible C++ code, allowing you to embed NeuroSolutions' algorithms into your own applications. Also, you can implement your own algorithms as DLLs and generate C++ source code for your neural networks.

Another very useful product is NeuroSolutions for Excel. It is an easy to use add-on that allows you to use NeuroSolutions to develop your neural network models entirely within the Excel environment. It creates a NeuroSolutions sub-menu that simplifies the process of getting data into and out of a neural network. You highlight portions of your data as training, cross validation, or testing, step through a few configuration panels, and you have a working neural network.

NeuroSolutions for Excel is organised into seven modules, each of which can be extended with user-defined functions in Visual Basic. You can apply various preprocessing techniques to the raw data and put it in a form that is most useful to the neural network. You can create a neural network from scratch through the use of the NeuralBuilder utility, or by opening an existing neural network.

The 'Train Network' module trains a network either once, multiple times with different random initial conditions, or multiple times while varying one or more network parameters. This permits the user to find the optimum network for a particular problem.

The 'Test Network' module can be used to test a network after training has been completed. Once you have trained the



Neurosolutions produces a modular neural network with a graphical user interface that is easy to use.

neural network and tested its performance to make sure it meets your requirements, the final step is to put the neural network to use. To do this, you simply enter in one or more rows of new input data and let the trained neural network generate an output.

Another NeuroDimension product is the interactive book *Neural and Adaptive Systems: Fundamentals Through Simulations* (ISBN: 0471351679) by Principe, Euliano, and Lefebvre, published by John Wiley and Sons in 2000. Its eight chapters bring the reader (using NeuroSolutions) almost painlessly from simple linear models, through Multilayer Perceptrons and Hebbian learning, to Kohonen Networks, Digital Signal Processing, and Adaptive Filter design. It describes the fundamentals of neural networks and adaptive systems. An immense amount of care has been put into the text, and into the preparation of the 200 interactive experiments and dynamic examples. The reader is led through the details of creating and applying neural networks by means of these worked examples. Playing with the experiments is a stimulating, entertaining, and efficient way to become a neural networks adept.

Wolfram Research produces an add-on for Mathematica called Neural Networks. It provides the tools to train, visualise, and validate neural network models. It supports neural network structures such as radial basis function,

‘Neural networks represent a new computing paradigm based on the parallel architecture of the brain. They can be “trained” to produce an accurate output for a given input’

feedforward, dynamic, Hopfield, perceptron, vector quantisation, unsupervised, and Kohonen networks. There is support for neural networks with any number of hidden layers and any number of neurons in each layer. It implements training algorithms like Levenberg-Marquardt, Gauss-Newton, and steepest descent. Neural Networks also includes

special facilities for function approximation, classification and detection, clustering, non-linear time series, and non-linear system identification.

More than 50 functions are made available on built-in palettes. These palettes are well organised with command templates, options, and links to online documentation. They facilitate the input of any parameter for the analysis, evaluation, and training of the network. Neural Networks also provides options to modify the training algorithms. The default values have been set to give good results for a large variety of problems, allowing you to get started quickly using only a few commands. Later, you will be able to customise the algorithms to improve the performance, speed, and accuracy of your neural network models. A special training record allows you to keep intermediate information.

The online documentation contains an introduction to neural network theory and detailed examples that demonstrate different neural network models. These are representative of typical problems so they can be adapted to your own data.

Neural Networks offers a sophisticated, easy, and convenient, environment in which to analyse neural network models. There is access to all of the capabilities of Mathematica to prototype new algorithms or to perform further manipulations on neural network structures.

