Axone_Icon.tiff ¬

Axone™

General purpose neural network application

Axone_SG-Half.tiff ¬

Axone is a general purpose Neural Network application for the NeXTSTEP platform. It was conceived to work with the MESA spreadsheet application from Athena Design, Inc. and allows the user to build neural networks and spreadsheet models in an interactive manner.

Axone was designed to take full advantage of the NeXTSTEP environment with its Mach messaging and its easy to use graphical user interface. Axone is thus able to offer several advantages over products on other platforms.

Features:

- **Axone and MESA are designed to run simultaneously**, and changes in Mesa are instantaneously reflected in Axone with the push of a "Refresh" button. This ease of use offers tremendous flexibility and makes neural network development very fast. Most neural network programs on other software platforms access their data from closed spreadsheet files, whereas Axone directly gets its data from the opened and running MESA spreadsheet. Any mistakes can thus be corrected instantaneously.

- **Axone's graphical user interface allows easy selection and normalization of Input and Output variables.** Further, it allows the user to visually inspect his Neural Networks, and, with the `click' of a button, control almost every aspect of Network construction. You can connect and disconnect connections, add and delete neurons, add and delete hidden layers, and you can even set the learning rate for each neuron individually, all with the simple click of a button. The user also has control over the colors with which the Neuron Threshold values and the Connection Strengths are displayed. Important connections and data relationships can thus be recognized very easily.

- **Axone accesses its data through the use of Mesa LABELS** (you can also access data by Column or by Row, or from ASCII files). A Mesa Label is a Mesa facility that lets you name a data range on your spreadsheet. Each Label thus represents an INPUT or OUTPUT variable. For each Input or Output variable, Axone gives you a choice of four normalization functions: LINEAR1(0,1), LINEAR2(-1,1), LOGISTIC(0,1), and the TANH(-1,1) normalization functions. You also have the option of not normalizing your data.

- **Axone was initially designed to allow for more sophisticated Financial Market Forecasting.** During the Training phase, the user can thus specify a TRAINING period, but also a TESTING period. For example, if you have ten years of price history, you could tell Axone to train your network on only eight years of daily historical data, and, at the same time, test the network on the last two years of out of sample data. This is an excellent way of avoiding 'overlearning' (learning by heart).

- **All calculations necessary for neural network training are performed by a platform independent calculation server.** Should only one machine be available, then this machine may naturally also serve as its own calculation server. However, if learning speed beyond that offered by the host computer is required, then the calculation server may transparently run on any other machine of a network. Axone.app and Axone_server are multitasking, which means that you can simultaneously train one or more neural network files, while at the same time work on another.

- Once you are satisfied with the performance of your neural network, you may chose to **output the neural network as either a C-function, which you can then integrate into your own custom applications, or, you may tell Axone to create a Mesa spreadsheet AddIn for you.** By choosing the AddIn option, you may transparently use your neural network just like any other innate Mesa function. For example, should you have trained a networks that predicts the Deutsche Mark market 5 and 10 days from now, then your network has two output variables. Assuming the network has 5 input variables located in columns A,B,C, D, and E, then your Mesa formula would look like this (assuming you named the network "DeutscheMark"):

  =DeutscheMark_OUT1(A1,B1,C1,D1,E1)

  and for the second output:

  =DeutscheMark_OUT2(A1,B1,C1,D1,E1)

Note: You need the NeXTSTEP developers kit to compile the networks. If you don't have the developers kit, you can just send the network files back to us for compilation.

- **Version 1.0 of Axone is based on the BackProp learning algorithm**. This algorithm offers excellent learning behavior on a wide spectrum of problems and is today thought to be in use in almost all neural networks. Nonetheless, future versions of Axone, currently in development, will offer the possibility to use several other learning algorithms.

To place orders:
1.800.452.7608
303.799.1435 fax
info@alembic.com