

NeuroShell Trader

Version 6.5

WARD SYSTEMS GROUP

7820 B Wormans Mill Rd #383

Frederick, MD 21701

Phone: 301 662-7950

Email: sales@wardsystems.com;
support@wardsystems.com

Sales: www.neuroshell.com

Support: http://nstsupport.wardsystemsgroup.com

Product: Neural network trading software

Price: Trader Professional \$1,495; Trader Power User \$2,295; Day Trader Professional \$2,495; Day Trader Power User \$3,495.

by Jason K. Hutson

NeuroShell's programming interface can get any trader started down a path of building trading systems quickly. The point-and-click interface for the various types of trading systems, indicators, and neural nets that can be created and debugged reduces the learning curve. You get a grasp of the numerous capabilities of NeuroShell when using many of its built-in operations, indicators, and examples.

Ward Systems Group has for decades been considered a solid and well-thought-of company. NeuroShell Professional has won the top *Technical Analysis of STOCKS & COMMODITIES* Readers' Choice Award in the category of artificial intelligence software for 16 years in a row. The company produces software solutions for education, research, and forecasting outside the realm of financial markets as well, such as medical and scientific applications and process control. The company has software packages that are used in hundreds of other applications. Their software is stable, easy to use, well-documented, useful across many disciplines, and appreciated by its users.

TECH SUPPORT

NeuroShell has an exhaustive amount of help files and examples. Ward Systems has created support websites, forums, videos, tutorials, and examples to the tune of over 400 indexed pages of help within NeuroShell. It also has over 100 videos of examples and training. A new user has a gamut of information with which to get started, and Ward Systems has worked hard to help you understand what is going on within their software. Their tech support website at <http://nstsupport.wardsystemsgroup.com> contains almost 20 years of examples and hundreds of pages of documentation and support.

NeuroShell Trader combines traditional and artificial intelligence techniques to build trading systems that can be backtested and optimized. So, a trader can combine traditional indicators and neural networks to create their own trading system. Dreaming up logical or not-so-logical schemes to make profits in trading can take time and effort. Running complex models with massive amounts of variables, data, and ranges can put a

damper on testing and paper-trading. Walk-forward optimizations can further slow a trader's progress. But Neuroshell is able to perform all these tasks efficiently.

FEATURES

I have tested NeuroShell in earnest a couple of times over the last 12 years, once in 2005 and once in late 2017 as

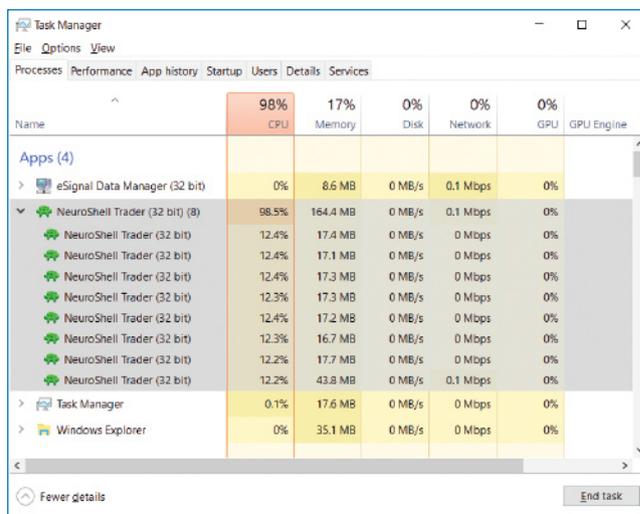


FIGURE 1: PROCESSOR CORES. Here you see an example of NeuroShell Trader using all eight processing cores.

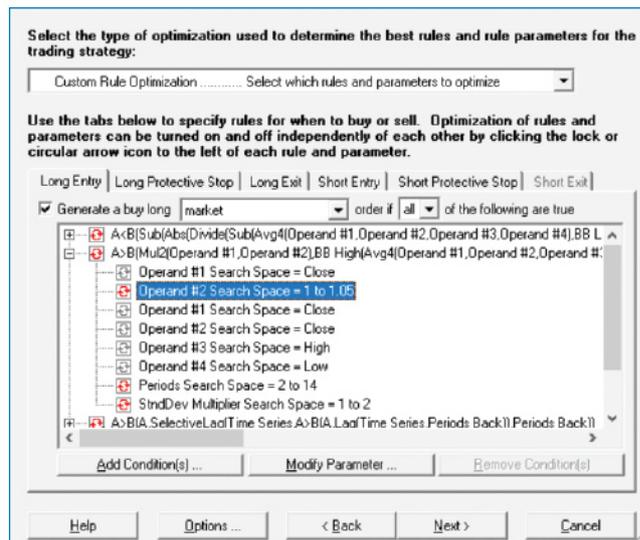


FIGURE 2: TO OPTIMIZE OR NOT TO OPTIMIZE. By clicking a datastream you can change whether NeuroShell uses a set parameter and whether it optimizes the parameter. You can also choose which parameter range to use in the optimization.

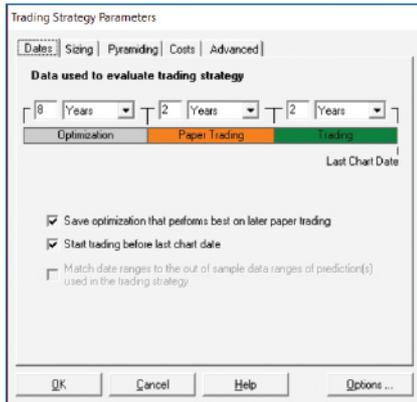


FIGURE 3: EXAMPLE OF THE INTERFACE FOR CHOOSING DATA. This strategy used an optimization that worked best during the eight-year optimization and two-year paper-trading amount of data. Then you can see how the model worked in real trading over the following two years.

I write this. While recently running optimizations on my home computer, I was pleasantly surprised with how NeuroShell took advantage of all my computer's processor cores (Figure 1). This greatly increased the speed at which I could debug and test new ideas. While using the program in 2005, I would just leave my computer be and let it work through a complex amount of data. Today, my fairly old 2013 eight-core AMD processor whipped through the same tasks. This means I can now dream up new ideas and test even larger data-crunching ideas. This software is fun to use.

The new interface in this version allows you to control parameter optimization on a case-by-case basis and to choose or change the range. I found this to be a great feature since it can be easy to curve-fit and over-optimize. Being able to turn off parts or change the optimization ranges helped me speed up my thought process and debug my systems and models (Figure 2).

You can backtest, optimize, and then test on different amounts of new data. You can also backtest, optimize, and use best parameters from paper-trading results on a forward data range. You can test that best model on amounts of new data moving forward to simulate what could happen if you traded your idea. Thinking about trading systems in terms of performance can be useful because you can curve-fit any system to look

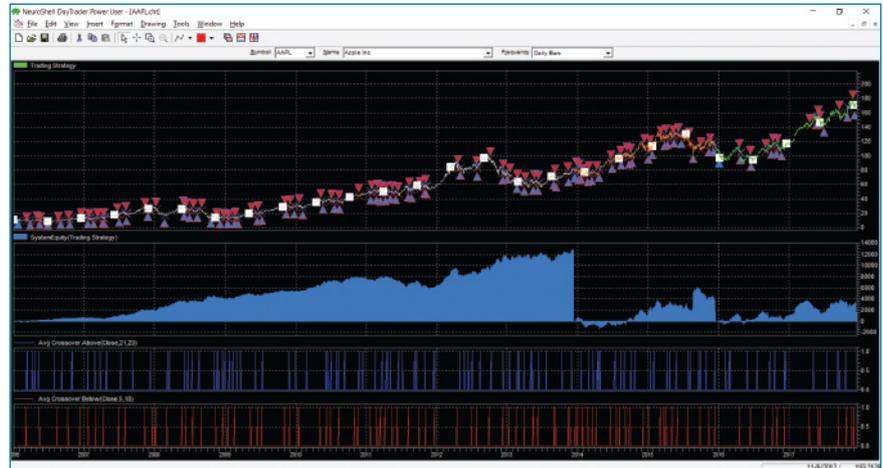


FIGURE 4: OPTIMIZATION EXAMPLE. The equity curve is seen in the subchart below the price chart.

spectacular during backtesting.

The trading system can move forward through future unknown data while still providing correct signals. This could show how profits could be made when trading your account (Figure 3).

There are precise backtesting inputs for approximating cost and trades associated with different-size orders, pyramiding, and brokerage fees. To simulate real trading, everything has to be accounted

Having the system move forward through future unknown data while still providing correct signals can show how profits could be made when trading your account.

for, and NeuroShell gives you the opportunity to input every cost. There is also the ability to optimize for many different objectives. Some examples are *Maximize #Winners - #Losers*, *Minimize Max Drawdown*, *Maximize Return on Account*, *Maximize ReturnOnAccount*EquityCurveCorrelation*, among others. Each objective pushes your model closer to matching up with your real trading goals. Keep in mind that trading costs and slippage can change your model and real-life trading scenarios.

NeuroShell provides the ability to test trading systems directly on the price chart. All you do is open up one or several charts for a specific time period and insert a trading system. Then you run an optimization and analyze at the system equity indicator. If you're satisfied, see how the system performs in an out of sample period by expanding the chart. In Figure 4 you see out-of-the-box example optimization for Apple stock (AAPL) while showing an equity curve in the second blue section. The data used went back to 2006 with eight years of backtesting, two years of paper-testing, and two years of actual trading.

You can use hundreds of built-in technical analysis indicators, create custom indicators within the NeuroShell interface, or pass data from NeuroShell to an external .dll. And if you'd like to manipulate data or provide inputs from other applications, you can do that by calling a .dll. The external .dll can return an array or value to NeuroShell. This ability extends the use of combining programming that may be complex or you already have from other programs or languages. You can create an interface between your work and NeuroShell by creating a .dll in a C, C++, or PowerBasic compiler and work directly with data to and from NeuroShell. You're not limited to stocks. You can test your models on forex, commodities, indexes, and options. As with everything else about Ward Systems and NeuroShell Trader, there

are numerous help files and example programs to get you on your way.

STRETCH YOUR IMAGINATION

If you have the data, NeuroShell will do the work. You can feed different inputs, outputs, and various data into a model to come up with a prediction based on different patterns found using neural networks. You choose the data and NeuroShell Trader can look for patterns in all your datastreams to make predictions a future price. Predictions work tick by tick using any timeframe. They can be adjusted out into the future based on how far into the future you'd like to try and predict.

I loaded a few neural net prediction models that I created several years ago based on fundamental and technical data. One of my more complex

neural nets showed extremely good performance for the out-of-optimization two-year “real-life” trading period with zero adjustments. Real-life trading versus system-testing, or in my case, following up on previous work, is an entirely different subject, but had I persevered, it would have certainly made significant profits.

In that context, I found the price to be well worth it and for that you get robust, solid, and well-documented software. The time you spend learning, practicing, and paper-trading your systems with it can give you a trading edge.

With NeuroShell Trader, the only limits to your success are your imagination and ability to follow through while testing all of your strategies.

NeuroShell Trader combines traditional and artificial intelligence techniques to build trading systems that can be backtested and optimized.



Jason Hutson can be reached at JasonHutson@traders.com.

‡NeuroShell Trader (Ward Systems Group)
‡See Editorial Resource Index

Find similar articles online at Traders.com 