MetaNeural Manual

Dear Customer:

This is a guideline of how to collect data to, train, test, and create the structure for your neural network brain to be used in the MetaNeural EA. As well as a description of the features of the Collector, Indicator, and EA. If there is anything that is unclear please use the forum to ask detailed questions. Thank you.

Metaneural Engineering

TABLE OF CONTENTS

PREREQUISITES

Install Visual Studio 6

1. Run the Setup.exe file to launch the installation wizard. Click "Next" to begin the installation.
2. Check the "I Accept the Agreement" radio button to agree to the license agreement. Click "Next."
3. Enter your product number and user ID in the appropriate text boxes and press "Next."
4. You may be prompted to update Microsoft virtual machine for java, keep the box checked and click next – setup will reboot the machine.
5. The installation will resume upon login. Choose the custom setup from the Setup Options screen. Custom installation allows you to install only what you need. You can add or remove features from the installation.
6. press "Next" and choose where files should be stored. The default folder location works best for most users. If you want to install Visual Studio 6.0 on an external hard drive, be sure to click "Browse" to choose the correct drive. Press "Next", "continue", "ok"
7. Uncheck the following: Microsoft Visual Basic 6.0, Microsoft Visual FoxPro 6.0, Microsoft Visual SourceSafe 6.0, Enterprise Tools, Graphics, ActiveX. Click "continue".
8. Check the "Register Enviornment Variable" box, click "ok", Visual Studio will install. Click "ok" twice afterwards.
9. You will be asked to insall MSDN, uncheck the box and click "Next" and "Yes"
10. Click "Next" twice, uncheck the "Register Now" box and click "Finish", setup is complete.

Install Neurosolutions 6

1. Run Install file
2. Click "Next" and "Yes" to accept the license agreement
3. Choose a Destination Folder, the default works best for most users, click "Next"
4. In the Setup Type window choose "Typical" and click "Next"
5. Click "Next" three times and installation will begin
6. Click "Yes" to add a shortcut to your desktop, and "Finish" to complete the install.

COLLECTION

ALWAYS remember to download all history data for the pair you want to collect data from.

The first step in using this system is collecting data, there are a few things to consider before collection because once market data is collected these variables cannot change when used in the EA:

1. **Symbol and Timeframe**

2. **Main Data Point** (**Output Type** in the Data Collector)
   a. This is the data point that all the others will be compared to in order to judge the slight differences between them

   b. For example, if you choose close then the high, low, and open will be subtracted from the close price in order to determine their pip distance from the close.

   c. This process converts all price values to 1, 0, -1, which makes it easy to train and test with neurosolutions.

   d. "Data" output type designates Indicator values.

3. **Using an Indicator or Pure Price Action**
   a. The collector gives the option to use almost any indicator to filter the market prices. This means if you use an MA all the prices will be subtracted from the MA value (in order to find the distance patterns between them and covert to 1, 0,-1).

   b. Essentially when using an indicator we're trying to find the patterns that happen in the market price in relation to the indicator value. For example, if the MA is pointing down at a certain inclination and the open, high, low, close, prices are in a certain configuration then 90% of the time prices fall in the next few bars.

4. **Total Input**
   a. This represents the size of a group of bars to be collected in sequence. Basically, this is how many bars the EA will look backwards to determine a pattern to use for predicting the next future bar (which is also called output).

   b. For example, if you're collecting 5,000 bars and select a total input of 5, then 5 bars will be collected as one data group and the "output" or future bar will be collected first - ahead of this group and it will be the bar containing the values every other bar in the data group will be subtracted from.

c. Max Total Inputs are 20

d. The larger this value is the less accurate the prediction will be. However, if it is too small the accuracy will be decreased also because there will not be enough data to extrapolate from. You must collect, train, test, and forward test, to determine the combination of settings that produce the most profit.

5. Range Hour

a. This is simply range of time within every day that you want to gather data for. If you intend to trade only one session it may increase the accuracy of your trading if you collect, train and test only for that session.

6. Predicted Bars

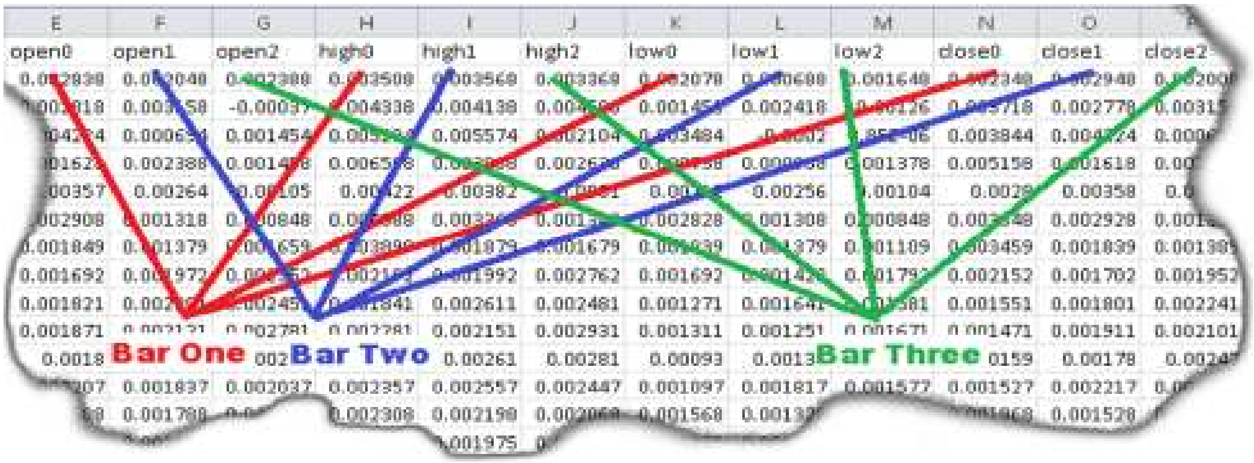
a. The amount of bars in the future to predict. This can be between 1 and 3 bars. For example, if you choose 3 then the output will be the next 3 bars in the future, the idea here is to maximize profits by getting a signal further in advance of market movement.

TIP: Use the CSV file name to remember your settings

EX: ma-14-eurh1-inp5-pred1-10k.csv

**The collector settings above must be the same in the EA or you will be effectively making your training irrelevant so take note of them before removing the indicator from the chart.

Once collection is complete a CSV file containing the values will be placed in experts/files



E	F	G	H	I	J	K	L	M	N	O	P
open0	open1	open2	high0	high1	high2	low0	low1	low2	close0	close1	close2
[illegible]	[illegible]	0.002388	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.001648	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	0.004138	[illegible]	[illegible]	0.002418	[illegible]	[illegible]	0.002778	[illegible]
[illegible]	[illegible]	0.001454	[illegible]	0.005574	[illegible]	[illegible]	[illegible]	[illegible]	0.003844	[illegible]	[illegible]
[illegible]	0.002388	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.005158	[illegible]	[illegible]
[illegible]	0.00264	[illegible]	[illegible]	0.00382	[illegible]	[illegible]	0.00256	[illegible]	[illegible]	0.00358	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.002828	[illegible]	[illegible]	[illegible]	0.002928	[illegible]
0.001849	[illegible]	[illegible]	[illegible]	[illegible]	0.001679	[illegible]	[illegible]	[illegible]	[illegible]	0.001839	[illegible]
0.001692	[illegible]	[illegible]	[illegible]	[illegible]	0.002762	0.001692	[illegible]	[illegible]	0.002152	0.001702	0.001952
0.001821	[illegible]	[illegible]	[illegible]	0.002611	0.002481	0.001271	0.001641	[illegible]	0.001551	0.001801	0.002241
0.001871	[illegible]	0.002781	0.002281	0.002151	0.002931	0.001311	0.001251	[illegible]	0.001471	0.001911	0.002101
[illegible]	[illegible]	[illegible]	[illegible]	0.00261	0.00281	0.00093	[illegible]	[illegible]	[illegible]	0.00178	[illegible]
[illegible]	0.001837	0.002037	0.002357	0.002557	0.002447	0.001097	0.001817	0.001577	0.001527	0.002217	[illegible]
[illegible]	[illegible]	[illegible]	0.002308	0.002198	[illegible]	0.001568	[illegible]	[illegible]	[illegible]	0.001528	[illegible]

The collected data is extracted into a spreadsheet which displays price data as open, high, low, close (OHLC). The OHLC of each bar is collected separately and placed in its own column. In the example above each row represents 3 bars in total. Therefore, the columns represent hundreds or thousands of bars collected going back into history.

Use Indicator

If you want to use an indicator you must do so by inputting the MODE and SETTINGS from that indicator into the collector. MODE is easily determined by going to the color tab of that indicator and looking at the # to the left, that number is the MODE.

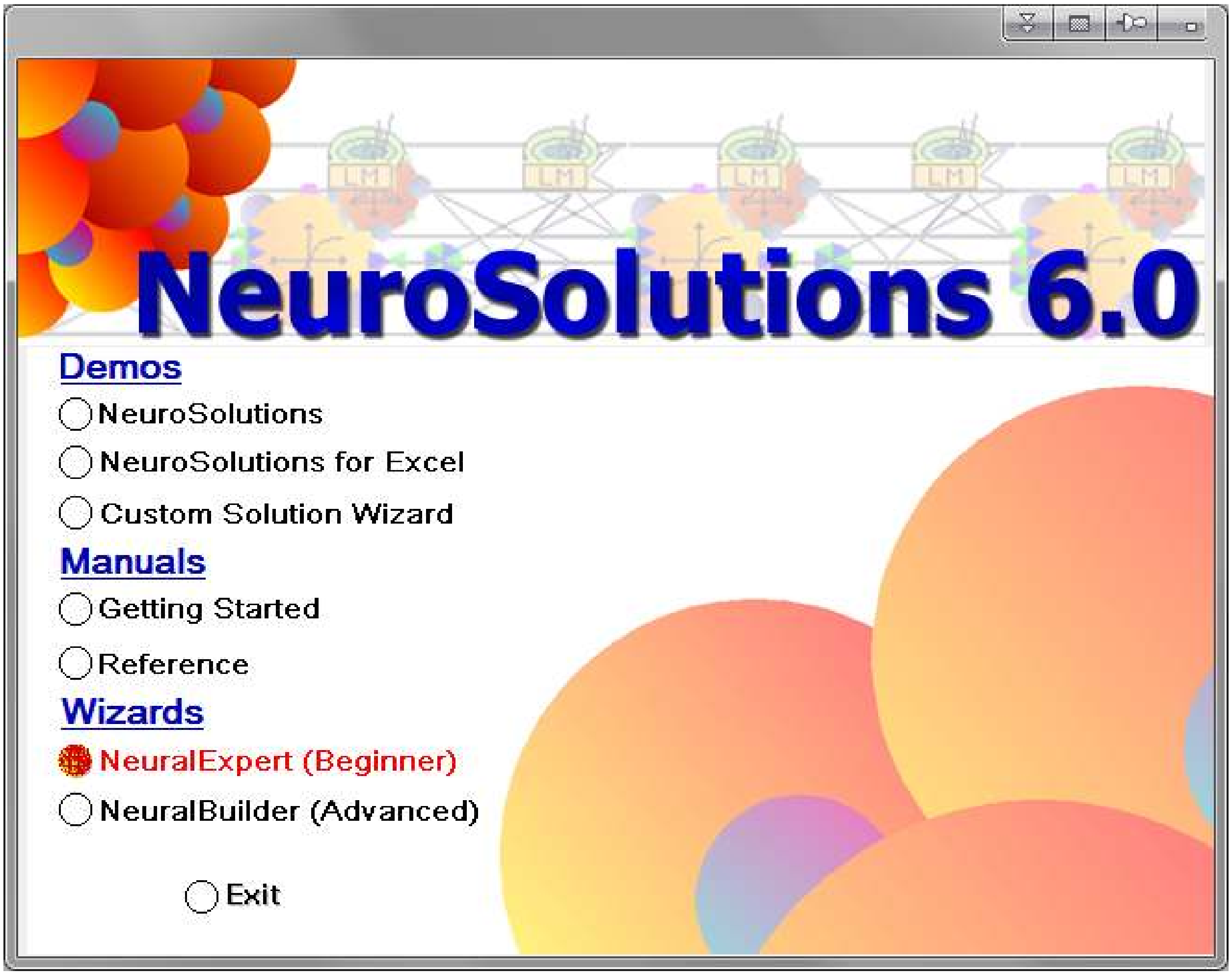
SETTINGS must be entered in order from top to bottom of the indicator setting window. So the first value in the indicator will be the leftmost value in the CustomIndicatorSettings section of the collector and EA.

Indicator settings must be separated by commas.

TRAINING AND TESTING

In Neurosolutions testing and training is combined into one easy operation. The program will set aside a percentage of your collected data so it can be used solely for testing, it will not be used to train. When training has begun testing will happen simultaneously. Below is the training, testing, and DLL creation (brain) in 10 easy steps.

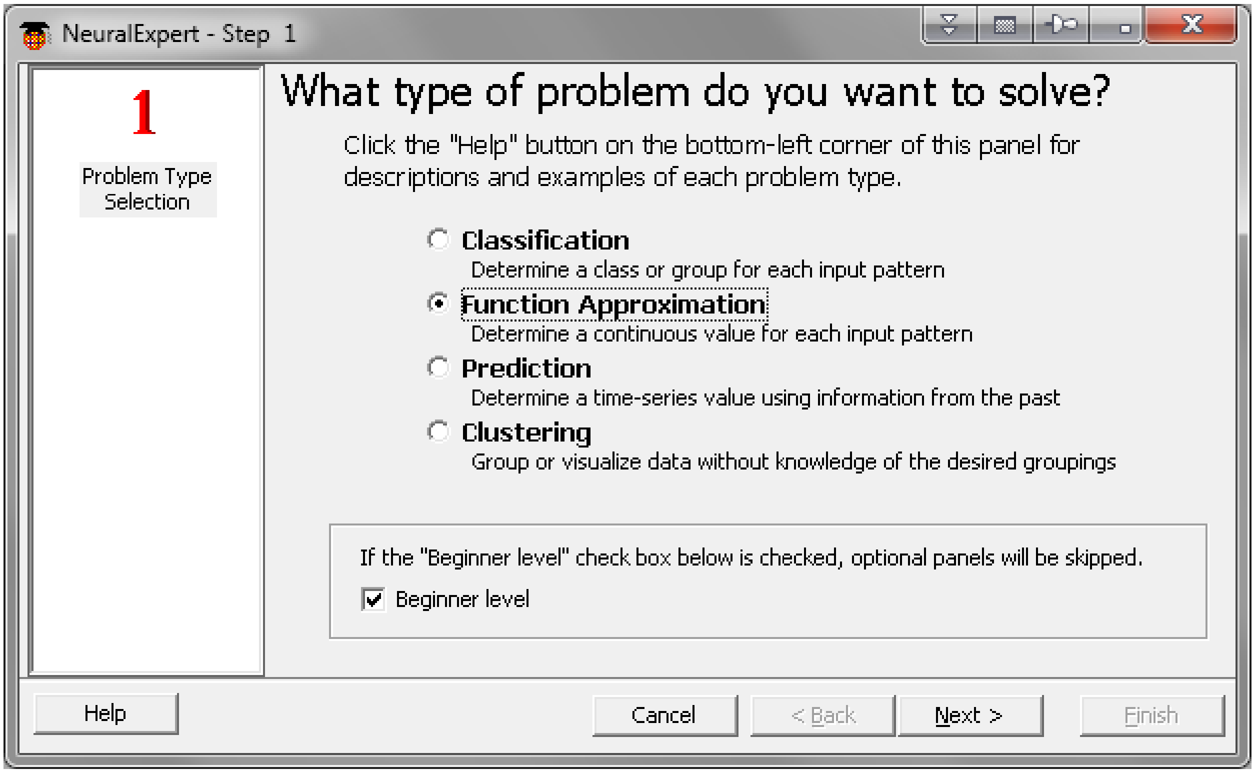
STEP 1



Choose NeuralExpert (Beginner) once NS opens.

STEP 2

Choose Function Approximation – Because we are trying to approximate they next leg in a pattern in order to predict the next bar. It is a very complex pattern of OHLC, indicator data etc but a pattern nonetheless.



What type of problem do you want to solve?

Click the "Help" button on the bottom-left corner of this panel for descriptions and examples of each problem type.

○ **Classification**
Determine a class or group for each input pattern

● **Function Approximation**
Determine a continuous value for each input pattern

○ **Prediction**
Determine a time-series value using information from the past

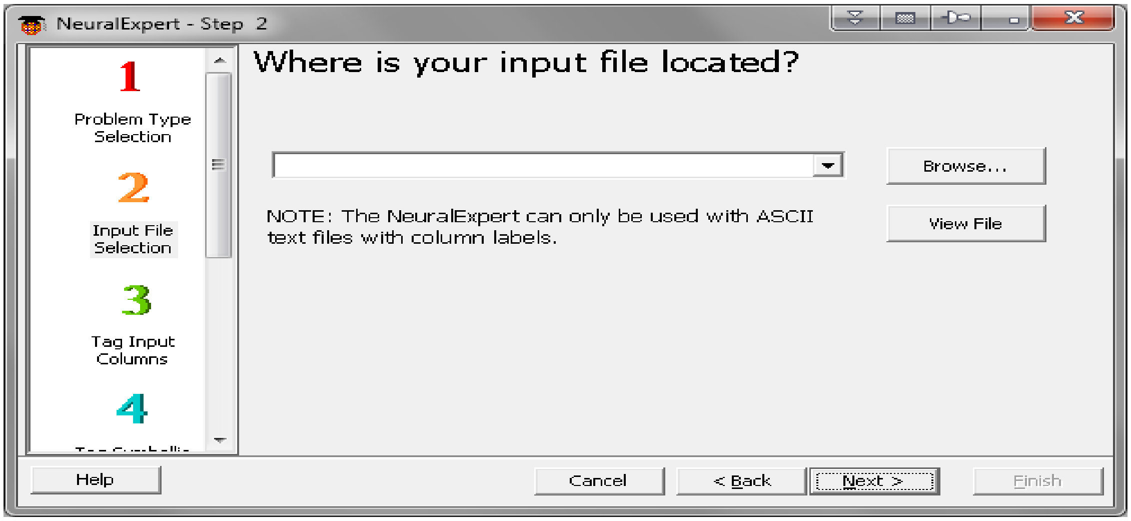
○ **Clustering**
Group or visualize data without knowledge of the desired groupings

If the "Beginner level" check box below is checked, optional panels will be skipped.

☑ Beginner level

STEP 3

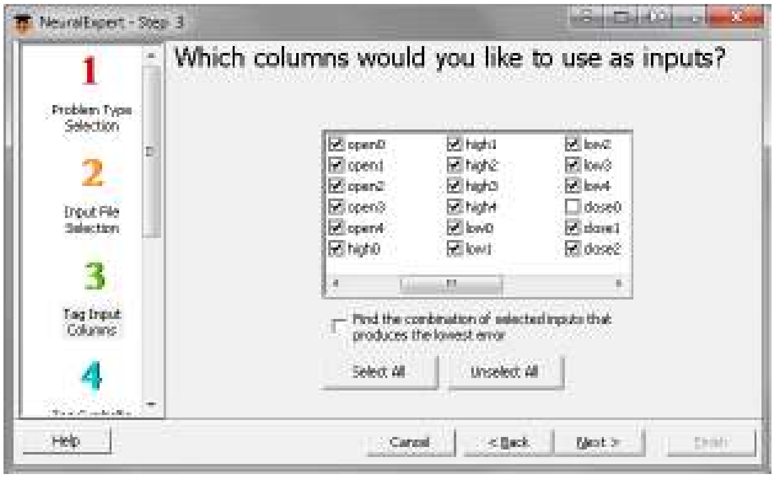
Choose the csv file you created from the collector



Where is your input file located?

NOTE: The NeuralExpert can only be used with ASCII text files with column labels.

STEP 4



Which columns would you like to use as inputs?

☑ open0
☑ open1
☑ open2
☑ open3
☑ open4
☑ high0
☑ high1
☑ high2
☑ high3
☑ high4
☑ low0
☑ low1
☑ low2
☑ low3
☑ low4
□ close0
☑ close1
☑ close2

□ Find the combination of selected inputs that produces the lowest error

CLOSE 0 was the main data point “Output Type” in this picture example

There are many different was to **choose your inputs,** the data you’re going to use to train your neural network, but there are two guidelines that never change:

1. The Main Data Point (Output Type) you selected in the collector for the ‘future’ bar must be unselected.
    a. The ‘future’ bar is the first bar collected before the data groups (Total Input # in collector) that all the group values are subtracted from, because of this the main output type in the ‘future’ bar will always be zero – it is subtracted from itself and is the baseline of 0.

2. The output must also be unselected, it will be the first checkbox to the left labeled Output1,2, or 3.

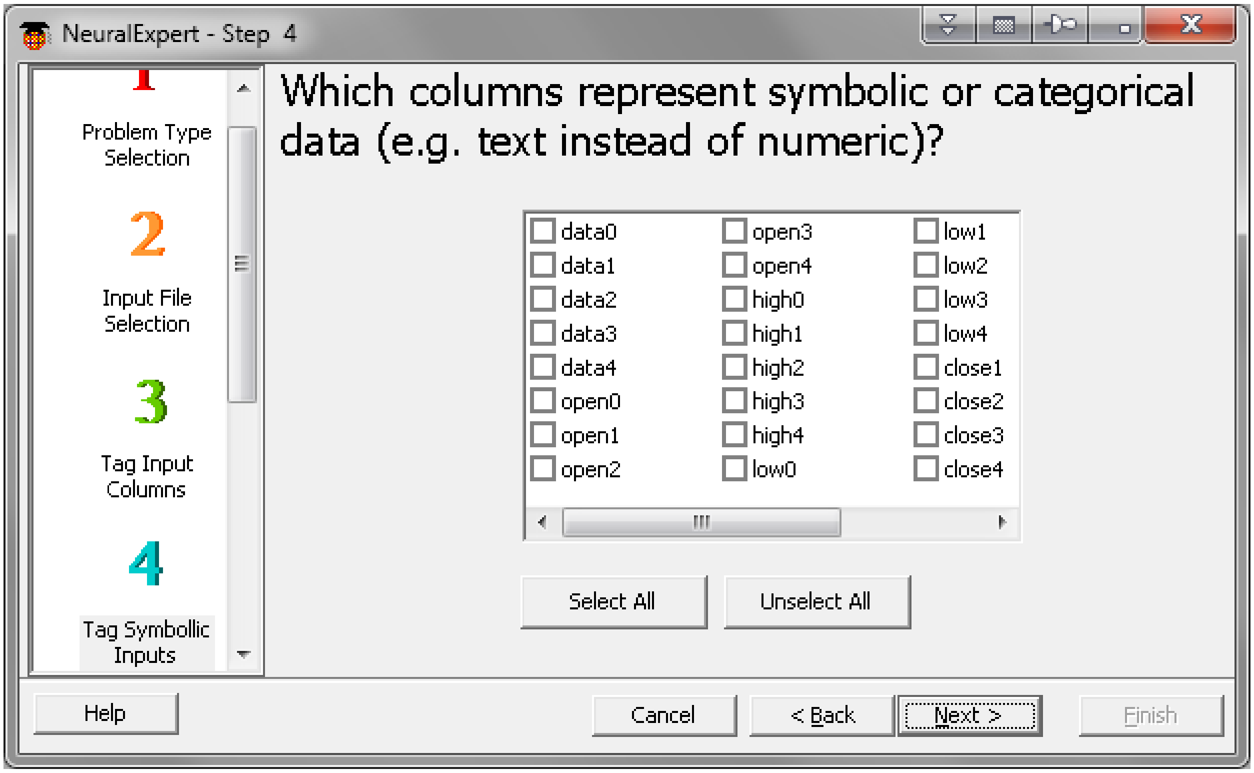
Variable Configuration

1. You can vary the data point you’d like to use as output. You can predict only, highs, lows, closes, even if the main data point is the Open.
    a. The main data point is simply a point of reference from which the distance of all the other data points are judged, it need not be your output.

    b. For example: I select “Close” as my Output Type or Main Data Point in the collector

STEP 5

There are no symbolic columns, click 'Next'

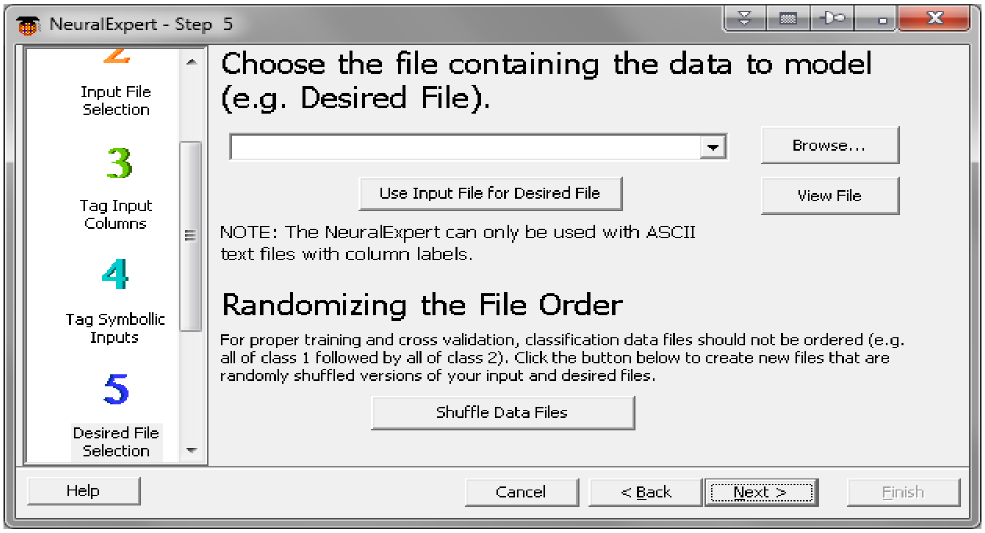


Which columns represent symbolic or categorical data (e.g. text instead of numeric)?

□ data0
□ data1
□ data2
□ data3
□ data4
□ open0
□ open1
□ open2
□ open3
□ open4
□ high0
□ high1
□ high2
□ high3
□ high4
□ low0
□ low1
□ low2
□ low3
□ low4
□ close1
□ close2
□ close3
□ close4

STEP 6

Click the 'Use Input File for Desired File' to use the same collected data file as the model data



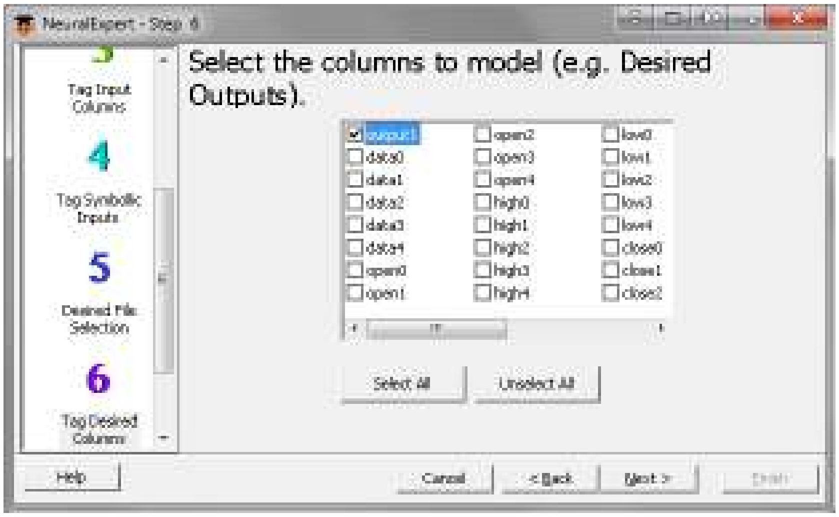
Choose the file containing the data to model (e.g. Desired File).

NOTE: The NeuralExpert can only be used with ASCII text files with column labels.

Randomizing the File Order

For proper training and cross validation, classification data files should not be ordered (e.g. all of class 1 followed by all of class 2). Click the button below to create new files that are randomly shuffled versions of your input and desired files.

STEP 7



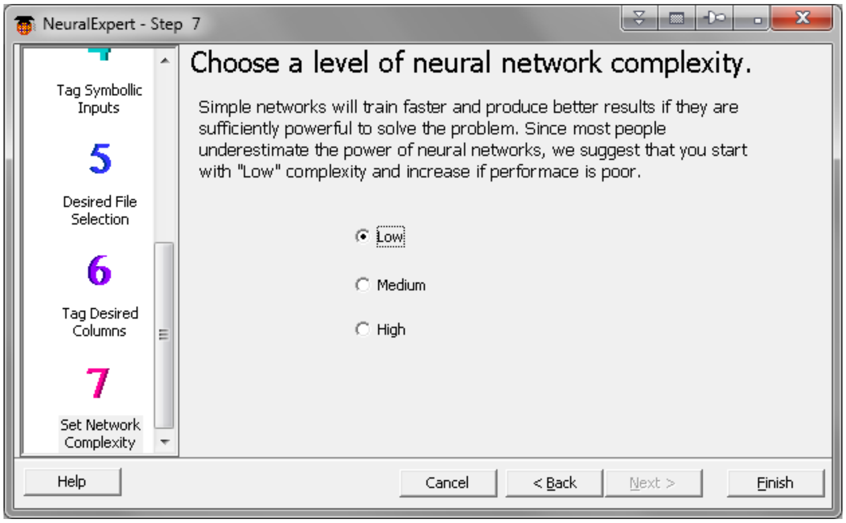
Select the columns to model (e.g. Desired Outputs).

☑ output1
□ data0
□ data1
□ data2
□ data3
□ data4
□ open0
□ open1
□ open2
□ open3
□ open4
□ high0
□ high1
□ high2
□ high3
□ high4
□ low0
□ low1
□ low2
□ low3
□ low4
□ close0
□ close1
□ close2

Select your desired output, this is the data point value you want to predict. It can be any data point except for the main data point selected as the output type in the collector for the future bar.

For example, if you selected your Output type as close, then close0 (main data point in future bar) would be all zeroes and could not be used as desired output.

STEP 8



Choose a level of neural network complexity.

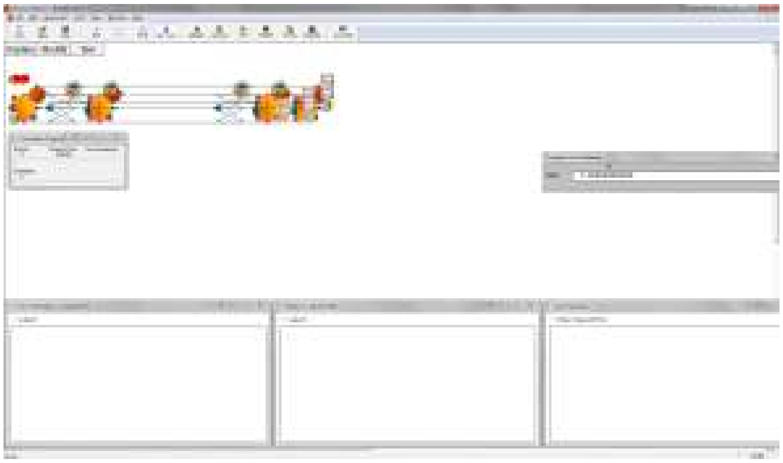
Simple networks will train faster and produce better results if they are sufficiently powerful to solve the problem. Since most people underestimate the power of neural networks, we suggest that you start with "Low" complexity and increase if performace is poor.

● Low
○ Medium
○ High

Select ‘Low’ complexity if you have only one desired output, for three you can use medium or High and test to determine which gives the best results.

Press ‘Finish’ to build the network.

STEP 9



Press the ‘Start’ button at the upper left-hand menu to begin training.

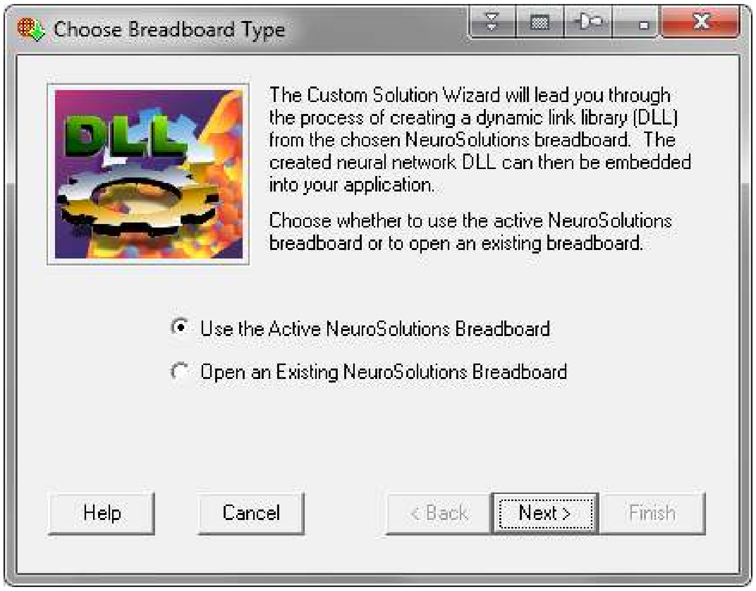
You will see graphs appear that depict the training process but most important is the ‘Average Cost of Criterion’ this will tell you how error prone your network is and the lower the value to better your results will be.

Keep an eye on the graphs also, if the desired output and actual output are wildly out of sync there will most likely be a problem creating the DLL or getting any useful results.

TIP: Create a folder for each set of breadboard files and name the folder the same as the collection csv and the DLL itself.

STEP 10

To Create the DLL (Brain) Click on CSW (Custom Solution Wizard) in the top menu to create the DLL which will house your neural network. Once you click ‘next’ it will prompt you to save your data as a ‘breadboard’ which is basically a set of project files which save all your settings and neural structure.



Choose Breadboard Type

The Custom Solution Wizard will lead you through the process of creating a dynamic link library (DLL) from the chosen NeuroSolutions breadboard. The created neural network DLL can then be embedded into your application.

Choose whether to use the active NeuroSolutions breadboard or to open an existing breadboard.

● Use the Active NeuroSolutions Breadboard

○ Open an Existing NeuroSolutions Breadboard

Once you save the breadboard select ‘visual basic 6’ from the project types, uncheck ‘launch project after DLL generation’ and click ‘Next’.

Select the breadboards folder name you created and ‘Finish’.

When DLL creation is finished it will say ‘DLL creation succeeded!’.

Now it is time to import the DLL into the EA, choose your trading settings and start trading.

EA OPERATION

THREE NEURAL NETWORK BRAINS

Each neural network brain that you create in neurosolutions can be used individually or with others that you create. You can use the DLLs containing your neural network structure for either trading, intelligent stop loss or intelligent trailing stop.

NN1 – NN2 – NN3

1. These sections are for using your NN DLLs for trading, when multiple DLLs are used they EA will wait until their signals are going in the same direction before opening a trade.

2. Each NN trading section can be used with different timeframes, indicators, output types and thresholds.

NN For Trailing

1. You can use one of your NN DLLs to determine the amount of pips to trail. For example, if your trained neural network predicts that the next bar will travel a large distance from the current bar you can set the trailing stop smaller or larger in that situation.

2. **NNOutputTrailingBuy** and Sell – When the NN ouput value is larger than these thresholds the **NNTrailingpips2** trailing amount is used, when below **NNTrailingpips1** is used.

NN For Stop Loss

1. You can use one of your NN DLLs to determine the amount of pips the SL is moved depending on the predicted value of your output. For example, if you neural network predicts that the next bar will travel a large distance from the current bar the EA will either increase or decrease your SL depending on the options you set.

2. **NNOutputNearSLBuy or Sell** – If the predicted value output from your DLL is larger than this threshold the **NearSL** will be used; it can be set to be smaller than Far or Larger depending on how you want the SL to react to such a situation.

3. **NNOutputFarSLBuy or Sell** – If the predicted value output from your DLL is smaller than this threshold the **FarSL** will be used; it can be set to be smaller than Far or Larger depending on how you want the SL to react to such a situation.

Thresholds

The thresholds are setup to be as variable as possible; you can set just about any max or min threshold for the by or sell side that you require.

You'll notice that the thresholds are setup for both the buy and sell side separately.

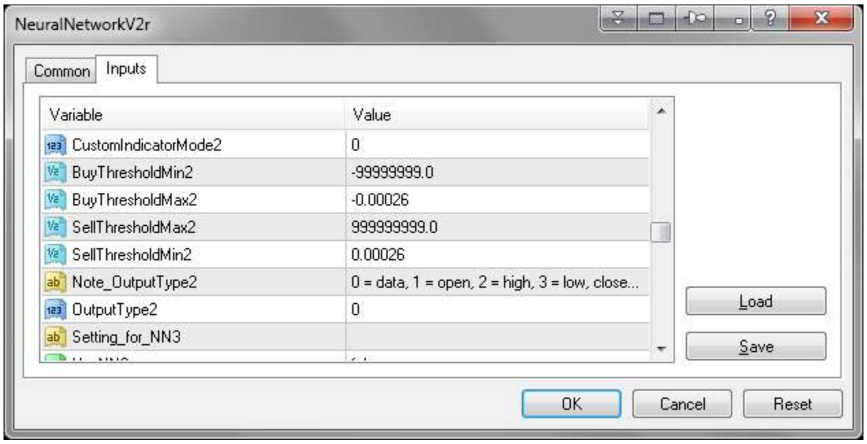
- ✓ This is necessary because all buy values are NEGATIVE numbers (because remember we're predicting the next bar's difference to the current bar so a NEGATIVE value would be the result of subtracting the furture bar's higher price to the lower current price)

To turn the thresholds off

- ✓ If you want to set the threshold option to accept all values (basically turning it off) simply set the BuyThresholdMin to -99999999 [this is the max or highest value for Buys] and the SellThresholdMax to 99999999 [this is the max or highest value for Sells] – then set the BuythresholdMax to zero and the SellThresholdMin to zero.

To isolate a particular range of neural network output values that correspond to the most profitable values you received after backtesting

- ✓ Say you noticed after extensive back and forward testing that the neural network output values you're getting for a particular pair are most accurate when between .00026 and up for Sell (sell values always positive) and -.00026 and up (really down, because buys are always positive). Then you would set the threshold to the following example picture:



NeuralNetworkV2r

Variable	Value
CustomIndicatorMode2	0
BuyThresholdMin2	-99999999.0
BuyThresholdMax2	-0.00026
SellThresholdMax2	999999999.0
SellThresholdMin2	0.00026
Note_OutputType2	0 = data, 1 = open, 2 = high, 3 = low, close...
OutputType2	0
Setting_for_NN3	

Other Variables

3. **SLpips** and **TPpips** – Set the fixed TP and SL amount

4. **isECN** – if your broker is an ECN type this must be set or TP and SL will not be set correctly

5. **MagicNumber** – used to differentiate EAs running on other charts and pairs, there must be a unique magic number for every chart this EA is placed on.

6. **FixedLots** – the fixed lot amount for each trade

7. **Risk Tolerance** – used with **ExitHighestLoss** to determine when to close trades when the losses are too high

8. **ExitHighestLoss** – when losses get to the Risk Tolerance amount the order producing the highest loss will be closed

9. **ExitOpposite** – When a trading signal that is opposite to the one of the current open order is triggered by the NNs the current trade is closed and a new one opened in the new direction – Even if the current trade is not in profit.

10. **TradeDuration** and **MaxDurationMinutes** – When **TradeDuration** is set to TRUE trades will only stay open to the **MaxDurationMinutes** limit then they will be closed, either in profit or not.

11. **TotalOrdersPerBar** – The amount of trades allowed within each bar.

12. **Slippage** – if the pip distance of the current prices exceeds this number from the moment the trade signal is triggered to the moment of actually opening to order then the trade will not open.

13. **TrailingPips** – the amount of pips that must be gained before the normal trailing begins; not used with **NNTrailing**.

14. **Hedging** – **HedgingPercentLots** – **HedgeTPpips** – when Hedging is set to true two orders will be opened for every signal, one in the direction of the signal and another in the opposite direction.

    - **HedgingPercentLots** is the amount of lots the opposite order is in relation to main signal order. If set to 1 then the lots used for the opposite order is 100% of the lot amount of the main signal (which is set by **FixedLots**)
    - **HedgeTpPips** is the amount of pips that must be gained before both sides of the hedge are closed.

15. **TakePicture** – is a diagnostic tool that is used when backtesting your strategy. When a backtest is run in visual mode it will take a snapshot of the chart at the exact moment a trade is opened in order to determine the exact NN value that opened that trade.

A Note About Backtesting

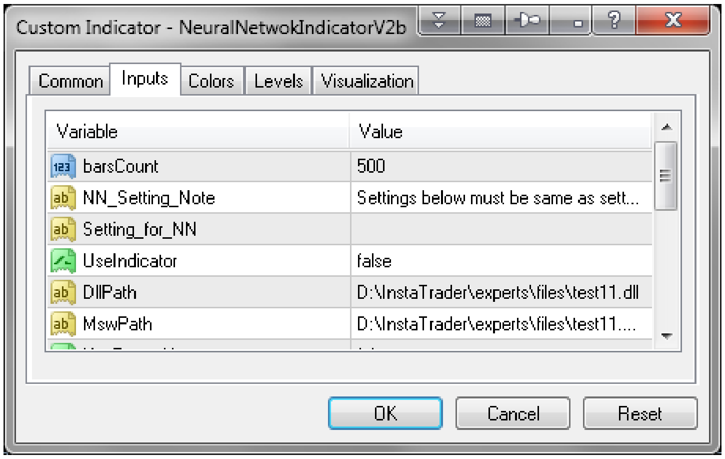
Keep in mind, the results in backtests are only approximate and do not represent 100% accuracy. Forward testing is the only true way to effectively test results.

NEURAL NETWORK INDICATOR

The main purpose of this indicator is to determine the approximate values your neural network DLL (brain) will produce in certain market conditions and therefore set your thresholds accordingly.

The reason it is approximate is because you'd be looking at past data and the EA processes patterns and predictions in real-time – it's the difference between using the static patterns of already fully formed closed bars to live bars that are being created.

The settings are similar to the collector and EA because they are mostly concerned with loading the neural network DLL correctly and passing the correct market data and indicator values to it.



Variable	Value
barsCount	500
NN_Setting_Note	Settings below must be same as sett...
Setting_for_NN	
UseIndicator	false
DllPath	D:\InstaTrader\experts\files\test11.dll
MswPath	D:\InstaTrader\experts\files\test11....

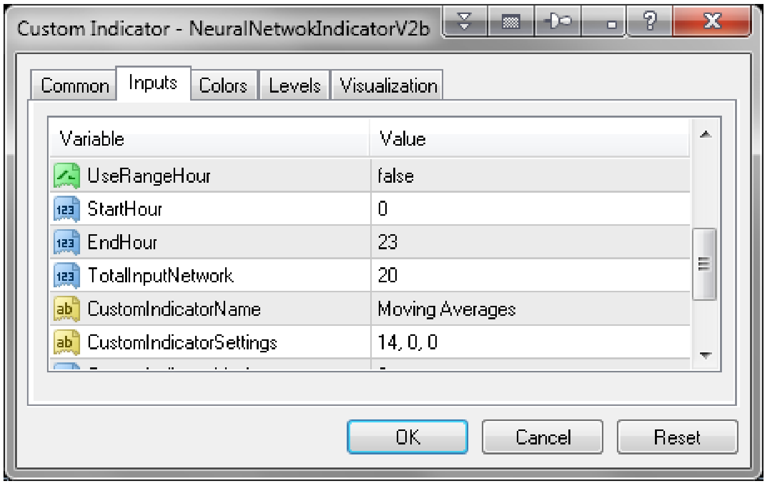
BarsCount – The amount of bars back you want the indicator histogram to display on.

**Keep in mind the more you use the more time it will take for your computer to process all the values so give it some time.

Use Indicator – Did you elect to use and indicator to associate your price data with in the collector? If not then you should not use an indicator here because it will skew your results and make all your training useless. **TRUE** to use an indicator – **FALSE** if you only used price data.

DLL Path – The location on your computer where the created DLL resides, you must put the full path and filename with extension (.dll)

NswPath – The location on your computer where the created Nsw (Neurosolutions settings file) resides, you must put the full path and filename with extension (.nsw)



Variable	Value
UseRangeHour	false
StartHour	0
EndHour	23
TotalInputNetwork	20
CustomIndicatorName	Moving Averages
CustomIndicatorSettings	14, 0, 0

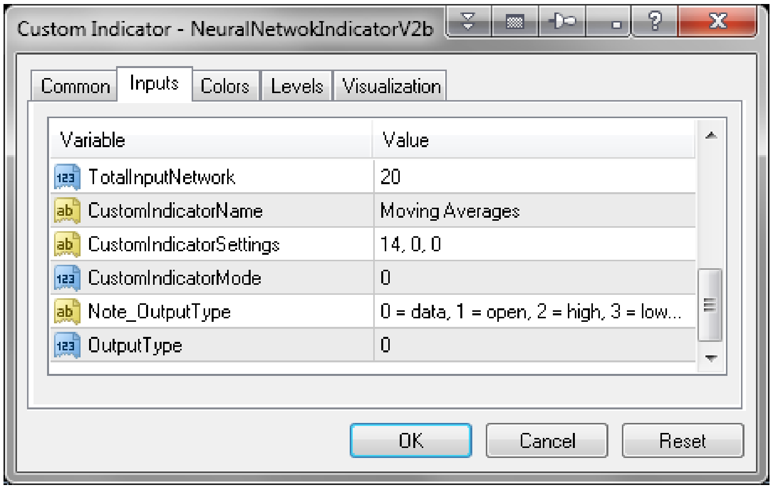
UseRangeHour – Enable ONLY if you collected and trained a specific block of time within each day.

StartHour and **EndHour** – Integers are used. Ex. 11:00 is entered as 11.

TotalInputNetwork – MUST be the same amount used in collector or the neural network structure you created will become confused.

CustomIndicatorName – Enter the name of the indicator you're using EXACTLY as it appears in the indicator filename (except for the extension at the end).

CustomIndicatorSettings - settings must be entered in order from top to bottom of the indicator setting window. So the first value in the indicator will be the leftmost value in the this section. Indicator settings must be separated by commas.



Variable	Value
TotalInputNetwork	20
CustomIndicatorName	Moving Averages
CustomIndicatorSettings	14, 0, 0
CustomIndicatorMode	0
Note_OutputType	0 = data, 1 = open, 2 = high, 3 = low...
OutputType	0

CustomIndicatorMode – The mode of the indicator you're using. Go to your indicator settings and the colors tab to find the mode you want.

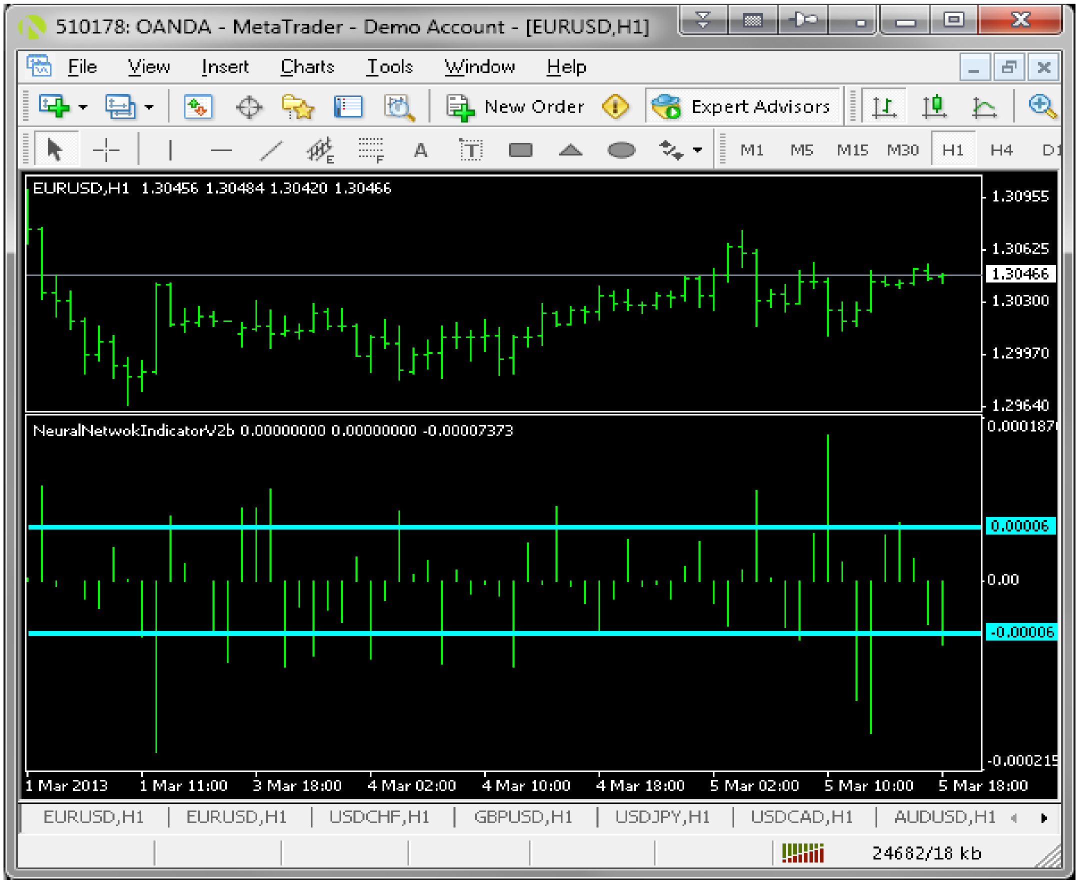
OutputType – This is the output type you used first for collection. If this is not the same as your collection output type all values will be incorrect.

Using the neural network indicator – Be careful not to tax your CPU

The indicator is meant for two things:

1. Determining the best threshold values to use to filter out the undesirable, least profitable, or most losing signals that are produced from the neural network output.

    a. In this case you'll probably want to set the BarsCount option to a thousand or so to get a good example of market movement to decide on the best range of values to use. [the higher the bar history the indicator is displayed on the more taxed the Metatrader application and your computer will be.



2. Aiding in trading by using it to 'predict' the next bar for manual trading.

    a. In this case it is best to set the BarsCount option very low, perhaps 100 or less because you only need the new, real-time values to determine you next trade and taking up valuable system resources is unnecessary.