Summary

Expert Advisor based on multi-layer neural network (BackPropagation Neural Network) with fully automated neural net training on the end of the week. Forex Neural Net will analyze all losing orders and inaccurate enters/exits for previous week and train based on new information. Expert Advisor will start trading on Monday with new re-trained neural net. This cycle will start again next weekend.

Theory

BackPropagation Neural Network

FeedForward BackPropagation architecture was created in the beginning of 1970s by several independent authors: Werbor, Parker, Rumelhart, Hinton and Williams. Now BackPropagation paradigm is the most popular, effective and easy learning model for complex multilayer networks. It is used in different types of applications and has generated the wide class of neural networks with different structures and training methods.

A typical BackPropagation networks has an input layer, an output layer, and at least one hidden layer. Theoretically, there are no restrictions concerning the number of hidden layers, but practically only one or two are used.

Neurons are organized in a level-by-level structure with a direct signal transmission. Every neuron of the networks produces the weighted sum of its inputs, runs this value through transfer function and delivers an output value. The network can model function of practically any complexity, and the number of layers and the number of neurons in each layer determine complexity of function. Determination of the number of intermediate layers and the number of neurons in them is important at network modeling. The majority of researchers and engineers, applying the architecture to certain problems, use general rules, in particular:

1. The number of network inputs and outputs are determined by the number of input and output parameters of the investigated object, phenomenon, process, etc. Unlike external layers, the number of neurons of the hidden layer is chosen empirically. In most cases sufficient number of neurons makes $n_{out} \times (\sum n_{in} + n_{out})$, where $n_{in}$, $n_{out}$ – the number of neurons in input and, accordingly, in the output layers.
2. If complexity in the relationship between the received data and the desired output data increases, the number of neurons in the hidden layer should also increase
3. If the process being modeled can be divided into several stages, the additional hidden layer (or layers) is required. If the process can’t be divided into stages, the additional layers may allow over-memorization and, accordingly, wrong general solution.
After the number of layers and the number of neurons in any of them are determined, it is necessary to find values for synaptic weights and network thresholds, capable to minimize an error of generated result. There are training algorithms exactly for this purpose, where an adjustment of a network model to available training data is made. The error for the concrete network model is determined by means of passing all training examples through the network and comparison of generated output values with the desired values. Multitude of errors creates an error function, and its value can be considered as a network error. As a rule, sum of error squares is used as the error function.

For better understanding of Back Propagation network training algorithm it is necessary to explain the concept of a surface of states. One dimension in multidimensional space meets each value of synaptic weights and network thresholds (free parameters of model, N in number). N+1th measurement meets a network error. The corresponding network error for different connection weights can be represented as a point in N+1-measured space, and all these points form a surface – the surface of states. The purpose of neural network training is to find the lowest point on the multidimensional surface. Actions, which occur in this case, can be referred to as controllable training: a “teacher” sends an input vector to the network input, and the desired value of computing result is sent to the network output. Controllable training of neural network can be considered as a solution of an optimization problem. Back Propagation can also be referred to as a network with a teacher.

The surface of states has complex structure and quite unpleasant properties, in particular, presence of local minima (points, which are placed too low in their certain region, but above a global minimum), flat regions, saddle points and long narrow ravines. It is impossible to find a position of the global minimum on the surfaces of states by means of analytical tools, that’s why neural network training, a matter of fact, includes analysis of this surface.

Basing on initial configuration of weights and thresholds (on randomly selected point on the surface), the training algorithm gradually finds the global minimum. Error surface gradient vector, which specifies the direction of the shortest descent along the surface from the given point, is calculated. If you move a little along the vector, the error will decrease. Eventually, the algorithm stops in the lowest point, which can prove to be only a local minimum (in an ideal case – the global minimum). Complexity here is in choice of step size. Convergence will be faster at a bigger step size, but there is a danger to jump over the decision, or to go in a wrong direction. The correct direction will be revealed at a smaller step size, but the quantity of iterations will increase. In practice the step size is proportional to the slope gradient with a certain constant – training speed. Correct choice of training speed depends on a specific task and is made on the basis of researches. This constant can also depend on time, decreasing in the process of algorithm movement.

The algorithm operates iteratively, and its steps are referred to as epochs. All training examples are sent one-by-one to the network input on each epoch, output network values are compared with the desirable values and the error is calculated. Error value and also gradient value of surface of states are used for weights correction, and then all actions are repeated. The training process will stop in several cases: if a certain number of epochs pass; if the error reaches a certain infinitesimal level; if the error ceases to decrease (as a rule, a user himself chooses the necessary criterion of stoppage).

**Operation principle of the advisor**

A core of the advisor is a scalping module, which trades during periods of quiet market. The sell signal is generated at achievement of maximum; the buy signal is generated at achievement of minimum. The signal is estimated by the neural network. A complex BackPropogation neural network architecture is used in the advisor. In our opinion, such network topology is perfectly suitable for prediction of time series behavior. Imperfection of neural networks as tools for prediction of financial markets is that accuracy of predictions decreases in time.
Thereby, we’ve developed «the auto-training module», which allows to retrain a neural network automatically. The training mode of the network is activated during weekend. After training the module deletes old files of the neural network and writes down the new ones. The advisor starts to trade with a new network on Monday. The advisor works in a completely automatic mode and does not require any actions to be done by the trader.

Installation

1. Run setup.exe

![Setup window](image)

2. Press ‘Next >’ button

![Risk Disclosure](image)

3. Check ‘I agree with Risk Disclosure’ and press ‘Next >’ button

5. Select currency symbol format for your brokerage company. For example: EURUSD, EURUSD_fx, fx_EURUSD, etc. You can check currency symbol for your brokerage company in Market Watch Window (Open Metatrader and press Ctrl+M).

6. Press ‘Next’ button
7. Select destination folder for Neural-AutoTraining module and press ‘Next >’

8. Select start menu folder and press ‘Next >’
9. Press ‘Install’ button

![Ready to Install window]

10. Press ‘Start’ button, select ‘All Programs’ click ‘Neural Auto Training’ and run Neural Auto Training module

![Start button and menu]

11. Registration window should appear

![Registration window]
12. Enter your client identifier, check ‘I agree with all the terms of license’ and press ‘Next >’ button

13. Confirmation window should appear


16. Open EURUSD chart timeframe m5, and add expert advisor to chart. Expert parameters window should appear.

17. Select ‘Inputs’ tab

**Parameters Explanation**

- **Lots** – Lot size. If you want to activate money management function, set Lots = 0
- **LotsPercent** – Money management function (percent by account balance)
- **OptimAuto** – should be 1 if you want to use Auto-training module, 0 – if you do not want to use this module
- **OptimEveryTime** – time when auto-training should start (24 hours format)
- **OptimEveryWeek** – day when auto-training should start (1 – Monday, 2 – Tuesday, etc.)
- **OptimPeriod** – historical period for training, days
OptimNeuroTraining- should be false

Recommended Parameters

We recommend to use Lots = 3% from account balance. For example if your account balance = 1000 usd Lots = 0.03

We recommend to use other parameters by default.

Contact Information

Tel: 1.347.329.4439 | Email: support@iticsoftware.com | skype: iticsoftware

Risk Warning

Before deciding to participate in the Forex market, you should carefully consider your investment objectives, level of experience and risk appetite. Most importantly, do not invest money you cannot afford to lose. There is considerable exposure to risk in any off-exchange foreign exchange transaction, including, but not limited to, leverage, creditworthiness, limited regulatory protection and market volatility that may substantially affect the price, or liquidity of a currency or currency pair.