

А. А. Шабалов

ПРИМЕНЕНИЕ НЕЙРО-НЕЧЕТКИХ СИСТЕМ В ЗАДАЧАХ ПОДСЧЕТА КРЕДИТНОГО БАЛЛА БАНКОВСКОГО СЕКТОРА

В виду сложности и значительных временных затрат проектирования нейро-нечетких моделей вручную с нуля в рассмотрение вводятся эволюционные алгоритмы. С помощью генетических алгоритмов разработка нейро-нечетких систем упрощается и становится автоматической. Предложенная схема применяется к известным задачам о кредитах в Австралии и Германии. Приводится сравнение с другими подходами.

Ключевые слова: нейро-нечеткое моделирование, эволюционные вычисления, нечеткие системы, нейронные сети.

© Shabalov A. A., 2012

UDC 519.234

M. Yu. Sidorov, S. G. Zablotkiy, E. S. Semenkin, W. Minker

EVOLUTIONARY DESIGN OF NEURAL NETWORKS FOR FORECASTING OF FINANCIAL TIME SERIES

The problem of forecasting in various technical, economic, and other systems is an important problem of nowadays. The methods of artificial intelligence and machine learning analyze very effectively various data including financial ones. The main problem of such techniques is the choice of model structure and the configuration of its parameters. In this paper we propose an evolutionary method for the neural network designing that does not require any expert knowledge in the area of neural networks and optimization theory from the user. This algorithm has been applied to the FOREX forecasting task of 13 different currency pairs based on the historical data for 12,5 years. The performance of the proposed algorithm has been compared to the forecasting results of other 6 algorithms. The proposed algorithm has shown the best performance on more than half of the tasks. On remaining tasks the algorithm yields slightly to the multi-layer perceptron trained by the particle swarm optimization algorithm. However, the predominance of the proposed algorithm is more significant.

Keywords: neural networks, evolutionary algorithms, particle swarm optimization, FOREX forecasting.

One of the expressive and pragmatic applications of artificial intelligence and machine learning is the prediction of financial time series in various markets. The FOREX market is the largest (about \$4 trillion daily turnover) international currency market. According to the positive market theory there is a deterministic component in the stochastic price fluctuations on the FOREX market. Therefore, using a fairly accurate predictor it is possible to achieve some speculative success.

Recently, an increasing number of papers present the advantage of artificial intelligence methods and machine learning algorithms over the standard econometric methods for solving the problem of financial time series prediction. In particular, neural networks successfully cope with the challenges of the financial forecasting. Thus, the most popular econometric technology for the problem of time series forecasting is called ARIMA [1]. However, in [2] it was shown that the multi-layer perceptron trained by different algorithms outperforms the ARIMA (1, 0, 1) model for the problem of FOREX forecasting.

The main problem of artificial neural networks which prevents their widespread exploiting is the challenge of choosing their optimal parameters for a particular problem. There are many parameters to be set up by the user,

for example, the type of neural network, the learning algorithm, the number of hidden layers and neurons, activation functions, etc.

In addition, the most of modern artificial neural networks have a fixed structure with the predefined types of activation functions. What if the neural network with a more flexible structure will be able to solve a problem more accurately?

In this paper we propose a method for evolutionary forming of neural networks, which on the one hand does not require any expert knowledge in the fields of information technology and artificial intelligence from the user and on the other hand creates the neural network with the flexible architecture that could potentially increase the prediction quality.

The structure of this article is as follows. Section 2 describes the source data and its statistical characteristics. Section 3 provides the description of the proposed method. Section 4 describes others methods for solving the forecasting problem. The experimental setup is described in Section 5. Section 6 presents the forecasting results of the proposed neural network technology, as well as the comparison to the other forecasting models. The conclusions are done at the end of the paper.

Data sets. To test the efficiency of the suggested method the historical data of 13 FOREX currency pairs from 1 January 2000 to 20 July 2012 were used. Each value of the time series is the maximum price during the week. Statistical characteristics of data samples are listed in Table 1.

Table 1

Statistical characteristics of data sets

	Mean	Standard deviation
AUD/USD	0.771580	0.165808
CHF/JPY	86.539585	9.672273
EUR/CHF	1.501780	0.120486
EUR/GBP	0.736243	0.101301
EUR/JPY	129.387638	19.260813
EUR/USD	1.234870	0.196712
GBP/CHF	2.115349	0.368944
GBP/JPY	180.275463	33.396468
GBP/USD	1.692953	0.185273
NZD/USD	0.733674	0.075432
USD/CAD	1.247870	0.207525
USD/CHF	1.276138	0.256394
USD/JPY	107.203875	14.817511

The largest values of mean and deviation correspond to the currency pairs with Japanese Yen (JPY).

Evolutionary forming of neural networks with flexible structure. Standard artificial neural networks have a fixed structure and predefined types of activation functions and connections between neurons. However, there is no fixed structure in the physiological analog of neural networks. What if the neural network with an arbitrary and flexible structure would be a better model describing available data? Since the structure and all the parameters of the neural network directly affect the final result, this could be a good idea to perform both structural and parametric optimization of the neural network according to the preselected quality criterion.

The Mean Absolute Error (Eq. 1) was chosen as the criterion for the quality estimation of the FOREX forecasting.

$$MAE = \frac{1}{N} \sum_i |x_i - x'_i| \quad (1)$$

This metric shows the closeness of the predicted currency rate time series (x'_i) to the historical values (x_i).

In terms of neural networks, the predicted value is a function of the structure and parameters of the neural network, respectively:

$$x'_i = x'_i(S, P).$$

The challenges for the optimization of the function (1) are the presence of many local minima, high dimensionality and undifferentiated structure in general. Therefore, many researchers prefer to use different heuristic algorithms of the direct search to minimize such functions.

Genetic algorithms have proved its high performance for the optimization of complex pseudo-boolean functions. In order to apply the genetic algorithm for solving

the problem of network structural optimization a mapping between a binary vector and a set of different neural network structures should be created. The mapping used in this study is depicted in Fig. 1.

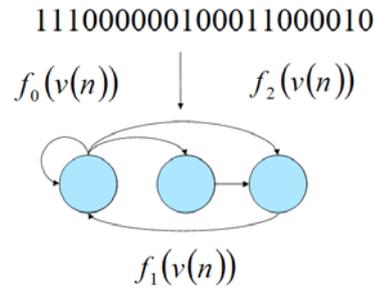


Fig. 1. Mapping between the boolean vector and the neural network structure

For each neuron there is a sequence of $n+4$ Boolean values, where n is the number of neurons in the network. The true value means the existence of the synaptic connection between the current neuron and the specified neuron, while false means the absence of such connection. The next 4 bits encode the ordinal number of the activation function. Thus, the number of different activation functions is equal to 16.

At each iteration of the structural optimization the synaptic weights of the current network are tuned by the particle swarm optimization algorithm which has shown the best performance. Thereafter the weights of the best found structure are tuned again with more resources to gain the finer model.

Alternative algorithms for solving the forecasting problem. To estimate the proposed algorithm the comprehensive comparison with other state-of-the-art techniques was carried out.

Multi-layer perceptron with standard back propagation. Multi-layer perceptron (MLP) [3] is a widely used type of the artificial neural network with a layer structure. The standard back propagation (SBP) is the first order algorithm which computes partial derivatives of the error function for all the synaptic weights. These weights are iteratively changed as follows:

$$\Delta w_i(n) = -\eta \frac{\partial E}{\partial w_i} + \alpha \Delta w_i(n-1)$$

where η – is the constant of learning speed, α – is the memory constant.

The disadvantage of this technology is the localized nature of the optimization process.

Multi-layer perceptron with genetic algorithm. The genetic algorithm (GA) [4] is an effective method for the optimization of pseudo-boolean functions which emulate the processes of natural evolution. Due to the binarization and the Gray code [5] the genetic algorithm is able to optimize the real-number functions. There are many examples in literature showing the advantage of the GA comparing to the classical optimization methods for MLP training, for example in [6].

Multi-layer perceptron with evolution strategy. The adaptation of the genetic algorithm for the direct optimization of the real value function without binarization is called evolution strategy (ES) [7]. The chromosome of the evolution strategy is the set of real values. ES can also be used for the task of MLP learning. The algorithm does not set any limitations on the optimized function.

Multi-layer perceptron with particle swarm optimization. Particle swarm optimization (PSO) [8] is the heuristic algorithm of direct search emulating the behavior of bird flocks, fish shoals, etc. Each particle (the solution for optimized problem) is characterized by the vector of speed $V(t)$ and the vector of position $X(t)$. The j -th speed component of the i -th particle is evaluated as follows:

$$V_i^j = wV_i^j + c_1 \text{rand}1_i^j (pbest_i^j - X_i^j) + c_2 \text{rand}2_i^j (gbest^j - X_i^j),$$

where c_1, c_2 – the acceleration coefficients, $pbest_i$ – the best position of i -th particle, $gbest$ – the best position of all particles, $\text{rand}1, \text{rand}2$ – uniformly distributed random variables from $[0, 1]$, w is the coefficient of insertion. The coordinates of the new position are calculated iteratively:

$$X_i = X_{i-1} + V.$$

Thus, each particle approaches both global and local optima. Random variables $\text{rand}1, \text{rand}2$ are responsible for the search in the small area of the found optimum points.

Multi-layer perceptron with numerical computing of partial derivatives. The optimization of functions with undifferentiated parts can be done using numerical values of partial derivatives. The derivatives can be estimated as follows:

$$F'(x_0) = \frac{F(x_0 - \varepsilon) - F(x_0 + \varepsilon)}{2\varepsilon}.$$

These estimations can be used by the optimization algorithm of the first order. In this study, the algorithm of gradient descent was used.

Non-parametric Parsen-Rosenblatt's estimation with genetic algorithm. Non-parametric methods play a special role among the algorithms for the data analysis. Non-parametric Parsen-Rosenblatt's estimator (PR) [9] has been successfully applied for the tasks of modeling, identification and control of complex systems.

Among the drawbacks of this method the following ones should be noted: the necessity to configure the smoothing parameters and processing of all training data at each iteration. The optimal smoothing parameters can determine which input arguments have the significant effect on the model, and which ones can be eliminated from the model without serious loss of accuracy.

For the data $\{u_{ij}, x_i\}$, where s is the number of samples and n is the dimension of the input space, the non-parametric Parsen-Rosenblatt's estimator is calculated as follows:

$$x_s(v) = \frac{\sum_{i=1}^s x_i \prod_{j=1}^n F\left(\frac{v_j - u_{ij}}{c_s^j}\right)}{\sum_{i=1}^s \prod_{j=1}^n F\left(\frac{v_j - u_{ij}}{c_s^j}\right)},$$

where v – the input vector, F – the bell function, for example Gaussian curve, c_s^j – smoothing parameters (width of the bell function). The quality of non-parametric model strongly depends on the values of the smoothing parameters. Moreover, the smaller value of the smoothing parameter indicates the larger importance of the current variable to the whole model.

Optimal smoothing parameters correspond to the minimum value of the average square error criterion:

$$Error(c_s) = \frac{1}{s} \sum_{k=1}^s \left(x_k - \frac{\sum_{i=1}^s x_i \prod_{j=1}^n F\left(\frac{u_{jk} - u_{ji}}{c_s^j}\right)}{\sum_{i=1}^s \prod_{j=1}^n F\left(\frac{u_{jk} - u_{ji}}{c_s^j}\right)} \right)_{k \neq i}^2.$$

The minimization of this criterion is performed by the genetic algorithm.

Experimental setup. The prediction was carried out by a sliding window method with a window length from 1 till 10 previous currency rate data with the unit step. For all neural networks the number of neurons on the hidden layer ranged from 1 till 20. For a particular type of neural network the optimal prediction window size and the number of neurons were determined experimentally. The prediction quality was determined by the MAE metric.

There were 200 neural networks (from 1 to 10 neurons at the input layer and from 1 to 20 neurons at the hidden layer) in each experiment. The activation function for the multi-layer perceptron was the hyperbolic tangent $f(w) = th(w)$. Also, there was a neuron bias [3] in MLP.

All available data were separated into training and test data in proportion $\frac{3}{4}$ to $\frac{1}{4}$ respectively. The best results of MSE metric on the test data for each problem are presented in Table 2.

Algorithm SBP has been restarted 10 times with the number of epochs $n = 7317$. The speed parameter was $\eta = 0.1$, the value of the memory parameter was chosen $\alpha = 0.25$.

There were the following setups and resources in GA for the MLP learning: the number of populations was 271, the number of individuals in each population was 271, the maximum step of binarization (with Gray's code) was 0.1 and each synopsis weight ranged in $[-3, 3]$. The type of selection was the tournament with the size of the tournament equal to 3, the type of the population forming was elite, the type of recombination was uniform, the type of mutation was normal (the probability of each gin mutation is $p = \frac{1}{n}$, where n is the chromosome size).

For the algorithm of the evolution strategy the following options were used: the number of populations was 271, the size of intermediate population was $\mu = 41$,

the number of individuals in each population was $\lambda = 271$, the size of parents pool was $\rho = 30$, the interval for each synaptic weight $[-3, 3]$, the type of recombination was intermediate [7], the type of population forming was $(\mu + \lambda)$.

The following options and resources were chosen for the MLP learning algorithm with PSO: $c_1 = c_2 = 2$, interval for each neural network weight was $[-3, 3]$, the number of swarms was 271, the number of particles in each swarm was 271, the speed limitation was 4 and the constant of the moment was $w = 0.81$.

For the MLP learning by gradient descent the following options were used: the coefficient of learning speed was $n = 0.1$, the number of steps was 7313, $\varepsilon = 10^{-8}$. This algorithms was run 10 times on each neural network.

The proposed method of neural network design was set up as follows. For structural optimization the GA with the following options was used: the number of populations was 15, the number of individuals in each population was also 15. The type of selection was tournament with the size of tournament equal to 3, the type of the population forming was elite, the type of the recombination was uniform and strong mutation level (the probability of each gin mutation was $p = \frac{3}{n}$, where n – the chromosome size). For the parametrical optimization of each neural network structure the particle swarm optimization algorithm with the following parameters was used: $c_1 = c_2 = 2$, the interval for each neural network weight was $[-3, 3]$, the number of swarms was 15, the number of particles in each swarm was 15, the speed limitation was 4 and the constant of moment was $w = 0.81$.

For the best neural network structure the particle swarm optimization algorithm with the following parameters was used: $c_1 = c_2 = 2$, the interval for each neural network weight was $[-3, 3]$, the number of swarms was 150, the number of particles in each swarm was 150, the speed limitation was 4 and the constant of moment was $w = 0.81$.

For the task of smoothing parameter optimization by the GA the following settings were chosen. The number of populations was 271, the number of individuals in each population was also 271. The type of selection was tournament with the size of tournament equal to 3, the type of the population forming was elite, the type of the recombination was uniform and the mutation level was strong (the probability of each gin mutation was $p = \frac{1}{n}$, where n – the chromosome size).

All the described algorithms were implemented from scratch in C++ language by the authors.

The proposed algorithm has shown the best performance on the 7 from 13 forecasting problems. On the other 6 time series the algorithm was only slightly outperformed by the MLP with PSO. The advantage of the suggested method is the automatic determination of all the important aspects, such as the number of neurons on the hidden layer, types of activation functions, connections between neurons, etc.

References

1. Box G. E. P., Jenkins G. M. Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, CA.
2. Kamruzzaman J., Sarker R. A. Forecasting of Currency Exchange Rates using ANN: A Case Study.
3. Wasserman P. D. Neural Computing: Theory and Practice.
4. Holland J. H. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.
5. Gray F. Pulse code communication. U.S. Patent 2,632,058.
6. Montana D., Davis L. Training feedforward neural networks using genetic algorithms. International joint conference on artificial intelligence.
7. Beyer H., Schwefel H. Evolution strategies, a comprehensive introduction. Natural computing 1. P. 3–52.
8. Eberhart R., Kennedy J. A new optimizer using particle swarm theory. Proc. of sixth int. symposium on micromachine and human science, Nagoya, Japan. P. 39–43.
9. Parzen E. On estimation of a probability density, function and mode // IEEE transactions on information theory. Vol. Pami-4, № 6. 1982. P. 663–666.

Table 2

The MAE criterion values for the best results of forecasting

MAE	MLP SBP	MLP GA	MLP ES	MLP PSO	MLP NUM	PR GA	NET PSO
AUD/USD	0.034148	0.011428	0.013778	0.010307	0.011815	0.052470	0.010188
CHF/JPY	3.704241	1.509978	1.258708	1.107943	3.448210	1.157921	1.140523
EUR/CHF	0.065937	0.011636	0.010799	0.009248	0.014545	0.155853	0.011724
EUR/GBP	0.009163	0.006831	0.007001	0.006625	0.009069	0.009890	0.006564
EUR/JPY	8.812449	11.847508	2.202127	1.401458	8.936750	1.571704	1.405196
EUR/USD	0.015725	0.013085	0.013660	0.012808	0.014412	0.014564	0.013166
GBP/CHF	0.053686	0.016296	0.017031	0.017330	0.049498	0.145116	0.013725
GBP/JPY	10.901556	10.895421	5.700924	1.654690	9.612234	6.182739	1.535645
GBP/USD	0.016016	0.012953	0.014562	0.012567	0.013086	0.012738	0.012851
NZD/USD	0.014358	0.010184	0.009745	0.009690	0.010118	0.022841	0.009472
USD/CAD	0.013623	0.010509	0.011513	0.010049	0.010824	0.014293	0.010237
USD/CHF	0.020761	0.010645	0.011908	0.010576	0.010980	0.085298	0.010454
USD/JPY	5.263673	4.674802	1.957348	1.706362	5.210588	7.796211	0.814446

М. Ю. Сидоров, С. Г. Заблотский., Е. С. Семенкин., В. Минкер

ЭВОЛЮЦИОННОЕ ФОРМИРОВАНИЕ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ ПРОГНОЗИРОВАНИЯ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ

Прогнозирование в различных технических, экономических и др. системах является важнейшей задачей современности. Методы искусственного интеллекта и машинного обучения являются эффективными средствами анализа в том числе и финансовых данных. Основной проблемой использования таких методов остается сложность настройки параметров моделей. Предлагается эволюционный способ формирования нейросетевых технологий, не требующий экспертных знаний в области нейронных сетей и теории оптимизации от конечного пользователя. Произведен сравнительный анализ показателей качества прогнозирования предложенной модели с другими методами искусственного интеллекта на исторических данных 13 валютных пар рынка FOREX, более чем за 12 лет. Предложенный алгоритм показал наилучшую результативность более чем на половине задач. На остальных задачах, алгоритм незначительно уступил многослойному перцептрону, обученному стайным алгоритмом.

Ключевые слова: нейронные сети, эволюционные алгоритмы, стайный алгоритм оптимизации, прогнозирование на FOREX.

© Sidorov M. Yu., Zablotkiy S. G., Semenkin E. S., Minker W., 2012

UDK 004.89 (681.3)

Yu. V. Smeshko, T. O. Gasanova

MODIFICATION OF FUZZY C-MEANS ALGORITHM WITH AUTOMATIC SELECTION OF THE NUMBER OF CLUSTERS FOR SPEECH UTTERANCE CATEGORIZATION

In this paper we propose a fuzzy clustering algorithm, which is able to find the clusters in a data set without the number of clusters as a user input parameter. The algorithm is based on the standard fuzzy c-means method and consists of two parts: 1) detecting the number of clusters c^ ; 2) calculating the cluster partition with the obtained c^* . We apply this method to the preprocessed database which was provided by Speech Cycle Company. The proposed algorithm has been tested with optimal parameters which we have calculated on the test data.*

Keywords: unsupervised fuzzy classification.

Cluster analysis consists of methods used to find the group structure in a certain data set. These algorithms can be applied to many different problems, such as image segmentation, data mining, the analysis of genomic and sensorial data, among others. A lot of different clustering techniques have been appeared in the last few years. Clustering algorithms do not need knowledge about the object's group labels, thus they use unsupervised machine learning models.

However, many clustering algorithms require parameters which should be selected by user, i. e., the obtained clustering depends on some user input parameters which should be chosen for the certain dataset. The target number of clusters or equivalents of it (such as density indicators in density models) are usually required. In this respect, we should consider and develop approaches, which can automatically detect the true number of clusters in a given database with no prior information about the structure and group labels.

With the appearance of the classical clustering approaches in the 1970s, researchers like J. A. Hartigan (k-means) were very conscious about the problem of detecting the correct number of clusters and proposed some

metrics for automatically determining this value. The general approach is to evaluate the quality of solutions which were obtained with the different number of clusters and select the value of number of clusters that originates the optimum partition according to a quality criterion.

In this work we considered the fuzzy c-means clustering algorithm and developed its modification which is able to discover the number of clusters automatically. In contrast to hard clustering methods where each object can be unequivocally assigned to only one cluster, in soft clustering (fuzzy clustering), objects are associated to all possible clusters. There is a so-called membership matrix, whose elements are degrees of certain object's membership to each cluster. Fuzzy approaches are more appropriate to deal with the existence of polysemous words and phrases.

We have chosen fuzzy c-means algorithm and developed its modification in order to solve the clustering problem on the database provided by Speech Cycle. This database consists of utterances in text form and some phrases and words have different meaning in another context.

This paper is organized as follows: Section 2 and Section 3 introduce the standard fuzzy c-means algorithm and