Application of High-Speed Algorithms for Training Neural Networks for Forecasting Financial Markets

Dmitry N. Karlov
Kuban State Technological University
Moskovskaya Ulitsa, 2, Krasnodar, Krasnodarskiy kray, Russia, 350042
Phone: +7 861 255-84-01
karlov-dmitry@mail.ru

Viktoria N. Zueva
Kuban State Technological University
Moskovskaya Ulitsa, 2, Krasnodar, Krasnodarskiy kray, Russia, 350042
Phone: +7 861 255-84-01
victoria_zueva@list.ru

Dmitry A. Trukhan
Kuban State Technological University
Moskovskaya Ulitsa, 2, Krasnodar, Krasnodarskiy kray, Russia, 350042
Phone: +7 861 255-84-01
neoasp@yandex.ru

Alexey A. Belykh
Kuban State Technological University
Moskovskaya Ulitsa, 2, Krasnodar, Krasnodarskiy kray, Russia, 350042
Phone: +7 861 255-84-01
aleksandr.belyh.75@mail.ru

Abstract

The article discusses the use of artificial neural networks in financial markets, due to the fact that this direction is very promising. Neural networks are well suited for tasks in which there are a large number of influencing factors, both known and unknown. The existing methods of neural networks training are considered, their pros and cons are noted. The use of neural networks in forecasting financial time series in real time is faced with the problem of the duration of the learning process of the neural network and the selection of significant inputs of the neural network. This problem can be solved by using the high-speed method of teaching the perceptron. The high-speed method of training allows for a much smaller number of iterations to train a multilayer perceptron on a given set of examples compared to the method of back propagation of the error. The high-speed method of training a multilayer perceptron allows us to assess whether it is possible to reach a given learning error of the neural network or not. This method of training is proposed to be used as an evaluation in the case of critical tasks before applying the method of back propagation of the error, in the case of non-critical tasks, the results of the neural network trained by this method can be used directly.

Keywords: Artificial Neural Networks; Methods of Neural Networks Training; Multi-Layer Perceptron Learning Rate; Financial Market.

1. Introduction

Currently, neural networks are the usual, common modeling tool in cases where it is impossible or difficult to create a mathematical model of the process being studied. Neural networks are well suited for tasks in which there are a large number of influencing factors, both known and unknown. The players of financial markets in terms of neural networks are interested in the following tasks: forecasting the rate of the instrument, classifying situations in the market, evaluating market instruments. Since financial time series are nonlinear series (Postarnak, 2012; Volkov, 2015; Gubaydullin, 2017; Malyshenko & Malyshenko, 2017), many researchers are looking for a solution using neural networks that are designed for the problems of forecasting the financial market.

2. Materials and methods

Consider the task of forecasting the exchange rate using a neural network. The inputs are usually the normalized values of Open, Low, High, and Close, possibly over the last several time periods, plus additional parameters, such as technical indicators, global indices, etc. The Close or High output. The number of inputs is 6–10, and the output is one, because in the forecast tasks, each output "grabs the biggest piece of the pie".In this case, the inputs are supplied with differences in the values of parameters, for example, Open [i] - Open [i-1], this is explained by the fact that the error on the difference of values is less than the error on the absolute value.

Given the high dynamics on the minute interval of the currency market, we consider it inappropriate to use standard neural networks for prediction (Malyshenko & Malyshenko, 2017). They should be used to forecast on five-minute timeframes and above, as well as to determine the trend reversal patterns. The serious disadvantages of neural networks for a rapidly changing financial series are the duration of training using gradient training methods, as well as frequent contact with local minima. The question of choosing a network structure sufficient for the task remains unresolved. There are also problems in choosing meaningful inputs.

The methods presented in Table 1 can be used as methods of teaching a multilayer perceptron methods for solving the problem posed (Volkov, 2015; Zueva, Trukhan & Karlov, 2017; Akobir, 2019; Karlov, 2018).

Table 1. The main teaching methods of multilayer perceptron

No	Methodname	Principleofoperation	Disadvantages
1	RPROP		Not accurate enough
2	Newton'smethod	Coincides with the well-known Newton method for solving systems of equations	Requires calculation of the second derivative
3	Newton's method with step adjustment	$x^{k+1} = x^k + \alpha_k h^k$, when $\alpha_k = 1$, coincides with the classical Newton method. Sometimes gives better results than the classical method.	Requires calculation of the second derivative
4	Coordinatedescent	Construct a vector with approximately the same properties as the gradient. A positive increment is given to the first coordinate of the vector. If the score is increased, then give a negative increment. It is necessary to do the same with other components of the vector. Theresultis a vectorwith a decreasingestimate.	MorecalculationsthanBackProp
5	RandomSearch	The initial parameter vector is set. The new vector is searched for as the initial plus random multiplied by the radius. If the error has not decreased after a certain number of iterations, then it is necessary to narrow the radius, etc.	High probability not to train neural networks
6	Shootingmethod	Almost the same as the random search method.	High probability not to train neural networks
7	TheNelder–Mead	A random point is generated in the parameter space, then a simplex is constructed with the center at this point and the side length *k*. At each vertex of the simplex, an estimate is considered. The vertex with the highest estimate is selected; a new simplex is built from it, and so on.	Experiments have shown that it does not exceed BackProp
8	Antigully	The random starting point is remembered, then *n* steps are performed for optimization using the steepest descent method, then optimization is done.	High probability of escaping from the global minimum
9	Quasi-Newtonian	The calculation is carried out using the second derivative.	A large number of calculations

3. Results and discussion

Further, the task of increasing the learning speed of a multilayer perceptron is considered in detail (Postarnak, 2012; Akobir, 2019; Karlov, 2018). One way is to use well-known structured method KBANN - knowledge based artificial neural networks, which was developed by J. Towell. The concept of this method is to generate a neural network topology based on well-known structured knowledge. The initial data for building a network using the KBANN method is a set of conjunctive rules of the type "If ... Then ..." relatedtothetaskbeingmodeled. This set of rules is transformed into a decision tree, which is transformed by a special algorithm into a multilayer perceptron. For this, the input and output variables are subjected to the scaling procedure. The scale values are matched with the corresponding input and output neurons. Intermediate nodes of the decision tree are distributed by levels of nesting. The number of nesting levels determines the number of hidden layers in the network. Each node of the decision tree that is not an input or an output is assigned a hidden neuron. Then connections are made in the direction from the inputs to the outputs; according to the principle, each neuron must be connected with all the neurons of the next layer. The result is a conceptually labeled perceptron-type architecture (Postarnak, 2012; Gubaydullin, 2017; Malyshenko & Malyshenko, 2017).

M - networks in which a definite concept is put in one-to-one correspondence to each neuron of an untrained network are widely spread. The conceptual connection of neurons greatly simplifies the meaningfulness and integrity of the understanding of the work of a neural network.

The ALN - Adaptive Logic Network method proposed by Dendronic Decisions is also known. This method is similar to the KBANN method; there is a fundamental difference at the initialization stage of the initial values of link weights. Each neuron of the hidden and output layer is assigned a logical operation. The calculation of the initial values of the links weights is carried out according to the following rules. If the operation of logical multiplication is implemented, then the sum of the weights of connections with the neurons of the previous layer, the corresponding positive variables of the logical expression, must exceed the threshold of neuron activation. If it is necessary to implement the operation of logical multiplication, then each variable of the disjunctive rule corresponds to the weighted active input of the neuron, the connection weight of which has a value greater than the activation function threshold. The resulting neural network structure already implements calculations prior to learning, similar to the decision tree on the basis of which it is derived.

The methods of KBANN, M-networks and ALM have a number of advantages, but require a priori knowledge of the problems to be solved in terms of formulas or rules, which is not always known.

Another, radically different approach to accelerating neural network learning is the RProp (ResilentPropogation) algorithm (Karlov, 2018). This algorithm uses the signs of partial derivatives when adjusting the weighting coefficients of the correction. When calculating the correction value, the rule is applied:

$$\Delta_{ij}^{(t)} = \left\{ \begin{array}{ll} \eta^{+}\Delta_{ij}^{(t)}, & \frac{\partial E^{(t)}}{\partial w_{ij}} \frac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \eta^{-}\Delta_{ij}^{(t)}, & \frac{\partial E^{(t)}}{\partial w_{ij}} \frac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \end{array} \right\}$$

$$0 < \eta^{-} < 1 < \eta^{+} \tag{1}$$

where

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \tag{2}$$

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)} \quad (3)$$

A change in the sign of the partial derivative of the weight w_{ij} at the current step indicates that the last change w_{ij} was large and the calculation algorithm slipped the local minimum. The magnitude of the change η^- must be reduced by and return the previous value of the weighting factor.

$$\Delta w_{ij}(t) = \Delta w_{ij}(t) - \Delta_{ij}^{(t-1)} \quad (4)$$

If the sign of the partial derivative has not changed, then the correction value needs to be increased by η^+ to achieve faster convergence. Parameter values are $\eta^+ = 1.2$ and $\eta^- = 0.5$. Initial values Δ_{ij} are usually set to 0.1. The scale correction value is calculated according to the rule:

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, \dfrac{\partial E^{(t)}}{\partial w_{ij}} = 0 \end{cases} \quad (5)$$

If, the derivative is positive as a result of the calculations, then the weighting factor must be reduced by the correction value, because the error increases. If the derivative is negative, then the weighting factor needs to be increased.

Further, it is necessary to calculate the weight correction using the formula:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (6)$$

Neural network speed learning algorithm is as follows:

1. Initialization of the correction value Δ_{ij}.
2. Presentation of examples of the training sample and the calculation of partial derivatives.
3. The value calculation Δ_{ij} according to the formula (5).
4. Weight correction by formula(6).
5. Check break condition. If it is not done, go to step 2.2.

This neural network learning algorithm has convergence 4-5 times faster than the back-propagation algorithm of the error.

Consider testing the speed learning method of a multilayer perceptron. During the experiments, the back-propagation error algorithm and the RProp algorithm were used for comparison. Before testing, standard data preprocessing was performed to predict:

- obtaining a time series with a given timeframe;
- filling in gaps in history;
- obtaining a series of relative changes in quotes;
- scaling.

Experiments on testing neural networks were conducted to predict EUR / USD exchange rate quotes, company stocks and the RTS index. During the experiments, in addition to the parameters of neural networks, financial instruments and timeframes were moved. The designations of the columns in the tables should be understood as: B - the back-propagation error algorithm (BackProp), R - the RProp algorithm, F - the speed learning method (Fast learning). The average learning time is understood as the number of iterations of learning. At the same time, one iteration of learning B, R, and F corresponds on average to 1: 0.8: 0.75. The training was stopped if the accuracy in the validation sample reached a certain number of percentages depending on the timeframe (as experiments showed, the higher the timeframe, the more difficult it is to predict). Table 2 shows the results of the work of neural networks in predicting EUR / USD exchange rates.

Table 2. The results of the work of neural networks in forecasting EUR / USD quotes

TF	Num. of examples	Wed number of neurons			Wed studying time			Accuracy on valid. sampling			Accuracy with new examples		
		B	R	F	B	R	F	B	R	F	B	R	F
1 min	100	25	29	22	2802	1011	562	97	97	97	85	79	82
	200	37	35	31	3599	1222	883	97	97	97	82	78	76
	300	58	52	47	4283	2152	1083	97	97	97	80	69	74
	400	64	62	52	4891	3152	1511	97	97	97	77	64	68
	500	73	68	65	5863	3208	2118	97	97	97	74	51	60
5 min	100	27	24	27	3413	829	575	95	95	95	82	78	79
	200	34	37	37	3720	1207	899	95	95	95	77	72	71
	300	42	42	41	4482	2135	1002	95	95	95	72	71	68
	400	55	53	55	6120	2865	1455	95	95	95	69	59	62
	500	73	69	72	8462	4251	2102	95	95	95	62	52	55
15 min	100	27	25	26	2240	956	433	92	92	92	86	80	82
	200	35	36	36	3652	1235	802	92	92	92	82	78	79
	300	44	42	43	4125	2154	1107	92	92	92	75	72	71
	400	56	53	57	4852	2944	1596	92	92	92	70	65	63
	500	85	77	78	7726	3461	2256	92	92	92	68	54	60
30 min	100	29	30	29	2451	951	563	90	90	90	85	79	82
	200	39	34	35	3752	1235	822	90	90	90	83	74	78
	300	44	45	44	4223	2154	1007	90	90	90	80	69	72
	400	56	54	59	4822	2944	1576	90	90	90	76	63	68
	500	72	70	68	7946	3961	2536	90	90	90	72	54	61
1 hour	100	31	31	29	2654	1051	525	90	90	90	85	77	80
	200	42	48	30	3652	1235	955	90	90	90	83	74	78
	300	53	57	39	4125	2134	1217	90	90	90	80	70	71
	400	66	62	57	4822	2984	1476	90	90	90	75	61	68
	500	81	77	82	9742	4461	2258	90	90	90	72	58	61
4 hours	100	33	35	32	2621	951	605	85	85	85	85	79	82
	200	44	47	39	3782	1235	916	85	85	85	82	74	79
	300	57	54	54	4165	2154	1217	85	85	85	77	69	72
	400	66	64	61	4222	2944	1596	85	85	85	76	64	68
	500	89	82	71	8741	3461	2596	85	85	85	72	60	61
1 day	100	36	32	33	2251	951	453	85	85	85	84	74	80
	200	42	39	41	3342	1235	812	85	85	85	81	72	78
	300	53	49	50	5125	2156	1121	85	85	85	77	64	71
	400	73	68	68	5822	2054	1433	85	85	85	74	61	64
	500	94	84	80	9745	4341	2153	85	85	85	68	52	61

Table 3. Processing the results of Table 2

Number of neurons			Wed studying time			Accuracy on valid. sampling			Accuracy with new examples		
F - B	F - R	R - B	F - B	F - R	R - B	F - B	F - R	R - B	F - B	F - R	R - B
0,09	0,06	0,03	3,03	0,75	1,35	0,00	0,00	0,00	-6,37	3,29	-9,66

From Tables 2 and 3 of the experiments, it can be seen that the speed learning method of the multilayer perceptron for the task of predicting financial instruments exceeds the learning time of the back propagation error and the RProp algorithm, while using fewer neurons of the hidden layer. At the same time, in those examples that the neural network did not see during the training, the high-speed algorithm is inferior to the back-propagation error algorithm. The numbers of the difference in the number of neurons in the hidden layer, speed and accuracy of learning vary slightly depending on the chosen prediction tool - it is most difficult to work with the highly dynamic Forex market. On average, the figures are as follows:

- according to the speed of learning, the speed learning algorithm exceeds the back-propagation error algorithm by 305%, and the RProp method by 77%;
- according to the number of neurons in the hidden layer, the speed learning algorithm uses fewer neurons than the back-propagation error algorithm by 9.5% and less than the RProp by 6.5%;
- in terms of accuracy on real-life examples after training, the speed learning method loses by 6.37% backtracking, but gains 3.3% from RProp.

4. Conclusion

The analysis of the training methods of multilayer perceptron suggests that the application of neural networks to the task of forecasting financial time series in real time is faced with the problem of the duration of the neural network learning process and the selection of meaningful inputs. Increasing the speed of learning through the use of high-speed learning algorithms for multi-layer perceptron eliminates one of the main problems of using neural networks to predict the financial market.The speed learning method allows reducing the number of iterations when training a multilayer perceptron compared to the back propagation method of an error when solving problems of financial market forecasting. This method will make it possible to assess whether it is possible to reach the specified neural network training error. The speed learning method can be used as an evaluation method for solving important problems before directly applying the back propagation method of error. When solving critically unimportant problems, the results of the neural network trained by this method can be used directly.

References

Akobir,Sh. Algorithm of learning RProp - mathematical apparatus.

Gubaydullin, A.V. (2017). The use of artificial neural networks in the development of the stock market forecasting algorithm.New information technologies in scientific research.Materials of the XXII All-Russian Scientific and Technical Conference of students, young scientists and specialists.Ryazan State University of Radio Engineering. 175-176.

Karlov, D.N. (2018). The algorithm of high-speed learning of a multi-layer perceptron.Scientific works of the Kuban State Technological University. (3), 167-173.

Malyshenko, V., & Malyshenko, K. (2017). The impact of the development of methods for financial stability analysis on the transformation of methods for stock market fundamental analysis. Экономический Анализ: Теория И Практика, 16(2), 364-378.

Postarnak, D.V. (2012). Critical analysis of models of neural networks. Bulletin of the Tyumen State University. No. 4. pp. 12-33.

Volkov,M.V. (2015). Simulation based on neural networks. Stages and examples. Innov: electronic. Journal. 33-34.

Zueva, V.N., Trukhan, D.A., Karlov, D.N. (2017). Neural network module of electricity consumption forecasting. Polytematic network electronic scientific journal of the Kuban State Agrarian University. (132), 1322-1331.