

Search for optimization of a real-time neural trading system

Michal Pobucký

The Institute of Computer Science, Silesian University in Opava, Czech Republic

michal.pobucky@fpf.slu.cz

Abstract: We describe the development of a complex modular system that should be able to predict the movement of the currency market (FOREX) using real-time neural networks and then automatically issue instructions for buying or selling individual currencies. To access the currency market, we use the free MetaTrader 5 application, which is linked to the Thisifi Trading Terminal (TTT) application that we created. TTT first reads the data needed for network training and then selects the training sets. The samples serve as input for training the neural network. Once the network is trained, the user can switch to online mode, where the network responds in real time to a change in the currency market and passes to MetaTrader instructions for the transaction.

1 Introduction

Business trading tools aided with artificial intelligence methods, and artificial neural networks in particular, are a frequent research topic. We aim at a development of a complex modular trading system which should be fully autonomous. The article focuses on the development of that part of the system that should be able to trade automatically on the currency market - FOREX. We experiment with trading strategies using multilayer perceptrons with the *Backpropagation* training and we present results of this initial study. While this network model is standard, the novelty of our approach lies in the composition of the training dataset. Future extensions of our trading system are described in Conclusions. The reader is referred to [20] for a more detailed description of our trading system.

The strategy of the trading system should not be based on the fact that based on the input data we will predict the exact movement of the price by the predicted value, such as [27], where the input of the neural network is the set of last 25 High or Low values. Currently, currency market traders perform either fundamental or technical analysis. The vast majority of them use technical analysis based on a specific set of mathematical tools – indicators. Each broker uses an application that displays a graph on which the broker sees the current price of a currency pair and its movement up or down. The broker can also add indicators to this basic chart, which can indicate whether the price will rise or fall, and tries to predict future movement of the market on the basis of his experience and knowledge

of individual indicators. Various analytical tools and the choice of right methods can reduce the effect of errors and also increase profitability [1][17].

With the help of the MetaTrader application, we gain access to the online trading system on currency markets. First, we programmed an automatic expert system, the so-called ExpertAdvisor, which obtains the values of individual indicators from the market and stores them in a database. We programmed this automaton in a special language MQL. This ExpertAdvisor stores all types of indicators, i.e. trend indicators [4], [8], [14], [25], [26], oscillators [3], [5], [6], [7], [10], [18], [22], [28], volume indicators [2], [24] and Bill Williams indicators [9], [21].

The values of individual indicators are sent to the input of the neural network - indicators that take the values 1 (buy), -1 (sell) or 0 (no instruction). The broker, as a human person, himself evaluates the information from individual indicators and, based on his own experience, subsequently instructs the currency market. At this stage, our trading system will evaluate the values of the indicators, not the actual price position in the market. This procedure is different from previous research. For instance, [29] focuses on eight indicators, four of which are simple moving averages and four are exponential. [30] used LSTM model with 11 indicators, [23] then gives an overview of the neural networks that were used for prediction on financial time series and indicators are represented here rather marginally. So far, the research has not focused on a training set comprising several hundred market indicators.

We trade on the currency pair EUR/USD, because it has the greatest volatility. Trading is minute by minute, so we work with minute charts and data. Other authors use also 1-hour time frequencies [19], daily and monthly frequencies [11]. All fees (broker AdmiralMarkets), including slippage, are already included in the profit of individual trades, so the net financial profit is really the profit that comes to the client's account. The lot size is set to 0.1.

2 Network settings

We used a classical multilayer perceptron with the *Backpropagation* training [12] based on gradient descent method [16], while the activation function is similar to a hyperbolic tangent of the form:

$$f(x) = \frac{x}{\sqrt{1+x^2}}$$

and its derivation is

$$f'(x) = \frac{1}{(1+x^2)^{\frac{3}{2}}}$$

Within each adaptation cycle we check whether the network has already been trained by at least 95% of samples. When this threshold is reached, we let the unlearned samples be trained more often and regularly by a special rule, to ensure that the network learns the whole training set.

movement interval		signal
0.00140	$+\infty$	BUY
0.00120	0.00140	10th neuron
0.00100	0.00120	11th neuron
0.00075	0.00100	12th neuron
0.00050	0.00075	5th neuron
0.00030	0.00050	8th neuron
0.00005	0.00030	4th neuron
-0.00005	0.00005	ZERO
-0.00030	-0.00005	7th neuron
-0.00050	-0.00030	9th neuron
-0.00075	-0.00050	6th neuron
-0.00100	-0.00075	13th neuron
-0.00120	-0.00100	14th neuron
-0.00140	-0.00120	15th neuron
$-\infty$	-0.00140	SELL

Table 1: Signal table

For the gradual evaluation of the found patterns, we had to define the scope of individual business instructions. For each time T_0 , the price movement at time T_1 to T_{10} is evaluated, and based on the size of the movement, it is assigned to one of the categories. By optimizing the network topology, the output layer is extended to 15 neurons, three of which are *Buy*, *Sell* and *Zero*, the others are auxiliary (4th to 15th). If the maximum price value in period $\langle T_1; T_{10} \rangle$ is within ± 0.00005 , it is neither an increasing nor a decreasing trend. The price is holding almost unchanged and it is a signal called *Zero*. On the other hand, everything that is at least 0.00140 from the original value is the *Buy* signal and everything that is at least -0.00140 from the original value is then the *Sell* signal. See Table 1 for an overview.

3 Neural network optimization

3.1 Variants of the trading system

The autonomous trading system consists of four separate neural networks, which were trained separately on their own data sets. Depending on the user settings, it is possible to specify in more detail how these networks will cooperate with each other. At the same time, general conditions for trading on the currency market can be set. We will compare results of trading systems with various settings in the tables that follow. Setting of the trading system is denoted by numbers and letters as follow.

The numerals **2**, **3** and **4** denote a trading system where 2, 3 or 4 neural networks were required to issue an order to buy or sell.

Letter **K** means correction. This makes it possible to eliminate situations where neural networks generate, for example, a *Buy* signal and at least one network generates a *Sell*, then the output of the trading system will be the value 991 and not *Buy*. Similarly, if the networks generate a *Sell* signal and at least one network indicates a *Buy* signal, the system result will be 992 and not *Sell*.

The letter **M** indicates that we will only accept buy or sell signals if the same signal was generated in the previous time period. This means at least two identical instructions in a row.

The letter **N** indicates that we will not trade at night between 22:00 and 06:00. During this period, trades are the lowest and at the same time the most unusual signals appear here.

3.2 Changes in network topology

The search for the optimal network began with four indicators in the input layer, one output neuron and one or two hidden layers. The maximum allowed error is set to 0.1 (*Buy* will be detected in the range from 1 to 0.9 and *Zero* in the range from -1 to -0.9.), the total number of cycles to 100,000, with one cycle making 100 random selections from the training data set. Data was used from one trading day. None of the tested networks achieved 100% success, so none of the networks fully adapted to the training data. Similar results were obtained when expanding the number of indicators to 7 and 13.

When the input layer expands range from 14 to 26 neurons, some of the four networks gradually adapt. While in networks with 14 and 15 inputs only two out of four were successful, after 25 thousand to 50 thousand cycles, for example ad for 21 inputs is already enough 4 to 20 thousand cycles and in 26 input neurons all networks have already fully adapted and with one exception, 3 to 7 thousand cycles were enough. It was similar to up to 40 indicators in the input layer.

We tested a larger range of the input layer and programmed 151 indicators that will be on the input. The network topology is now $151 - X - 1$, $X \in \langle 10; 63 \rangle$, maximum allowed error at 0.25, total number of cycles 300,000, 146 training samples from 3 trading days. From 16 neurons in the inner layer, the networks are capable of full adaptation and 40 to 50 thousand iterations are needed. When the training set is extended to 5 trading days and 187 training samples, the error increases and none of the tested networks is able to fully adapt.

We programmed the maximum of indicators and sent them to the input layer, which has now expanded to 236 neurons. We also test in parallel on 1,180 input neurons, where information from five consecutive bars is input within one data set, with a clear tendency to overfitting. Neural networks show similar results of adaptation,

with the topology $1,180 - X - 1$ being able to learn the same percentage of training samples in the previous cycle. From 110 to 130 neurons in the inner layer, we get to a network success of about 80%, slightly increasing from 240 neurons onwards.

Adaptation on 537 training samples shows a success rate in the range of 61 to 63% in the $236 - X - 1$ network, from the 110th internal neuron to 280. In the $1,180 - X - 1$ network, in the range of 67 to 73%, from 210 internal neuron after 320. The average error is then in the range of 0.46 to 0.51 for the $236 - X - 1$ topology and in the range of 0.30 to 0.44 for the $1,180 - X - 1$ topology, which means that we already have about twice the average error of networks. These results indicate that the constant increase in the training set is already becoming counterproductive and the reliability of the trained neural network is declining. Therefore, next we examined the effect of changes in the initialization of the weights and the changes of the training rate coefficient α on the quality of results.

We generated various hidden layer sizes and initial weight settings. The best variant had 240 neurons in the hidden layer and the initial value of the weights and bias is in the range from $\langle -1.0; 1.0 \rangle$ to $\langle -2.0; 2.0 \rangle$. Subsequently, adaptation testing with a different magnitude of the α coefficient was performed, as well as was sent to the input layer from one to seven bars, so that the input layer was large from 236 to 1,652 neurons. The most important aspect of adapting these networks is that none of these twenty networks was able to fully learn the complete training set, so even resetting the input layer range, inner layer, initializing weights and biases or training coefficient did not lead to the goal. The best setting for the α training coefficient is 0.001, unlike for example [15], where it is 0.05.

3.3 Optimization of complex network activation

The optimization will now take place in terms of the width of the neural network output layer. We expanded the output layer from one output neuron, which recognizes three output values to a set of three neurons, with the first neuron, when excited, indicates the *Buy* signal, the second neuron the *Zero* signal, and the third neuron the *Sell* signal. We also standardized the training set to 637 samples, with *Buy* signals 207, *Zero* signals 209 and *Sell* signals 221.

The search for the optimal topology was performed on $X - 240 - 3$ networks, with the number of bars ranging from three to seven, so that the size of the input layer was 708, 944, 1,180, 1,416 and 1,652 neurons. There are two conclusions – all networks have learned to recognize a complete set of training samples, and less than 5,000 repetitions, in some cases even less than 3,000 repetitions, were almost always enough for the learning itself.

Topology $708 - 240 - 3$, after activating a complex test data set that contained 14,382 samples, we received instead of 207 *Buy* signals 3,177 to 3,369 signals (on

four separate neural networks), instead of 221 *Sell* signals 3,077 to 3,564 signals and instead of 209 *Zero* signals 2,163 to 2,468 signals. Thus, instead of 637 recognized signals, the networks generated 8,798 to 8,928 signals.

Topology $708 - 240 - 4$, we decided to add another neuron to the output layer, which will be evaluated in all other cases where the samples will not be evaluated as *Buy*, *Sell* or *Zero*. Within the training set, two cases were tested – either 100 or 200 samples for the 4th neuron. The training set contains 737 and 837 samples.

A neural network with four neurons in the output layer has significantly better results, as we can see in Table 2. False signals (The neural network misinterpreted the *Buy* or *Sell* signal and there would be a financial loss in real trading.) for *Buy* fell by 36%, in the variant with 200 training samples even by 42 to 46%. There was a similar decrease in false signals for both *Zero* (by 28 to 30% and 42 to 44%, respectively) and *Sell* (by 32 to 36% and 40 to 46%, respectively). By creating the 4th neuron, we eliminated more than 2,000 erroneously detected signals, and even by simply increasing the set of training samples from 100 to 200 for the 4th neuron, we moved the nearly a thousand misrecognized samples that were divided between *Buy*, *Zero*, and *Sell* to the 4th neuron category.

Net 1	3N	4N 100	4N 200	Net 2	3N	4N 100	4N 200
1	3,369	2,125	1,812	1	3,230	2,147	1,942
0	2,352	1,574	1,328	0	2,290	1,612	1,287
-1	3,077	2,257	1,897	-1	3,324	2,074	1,851
4th sig.	-	784	1,667	4th sig.	-	690	1,499
Total	8,798	6,740	6,704	Total	8,844	6,523	6,579

Net 3	3N	4N 100	4N 200	Net 4	3N	4N 100	4N 200
1	3,201	2,070	1,743	1	3,177	2,023	1,709
0	2,163	1,557	1,237	0	2,468	1,716	1,372
-1	3,564	2,101	1,860	-1	3,220	2,071	1,825
4th sig.	-	793	1,858	4th sig.	-	687	1,549
Total	8,928	6,521	6,698	Total	8,865	6,497	6,455

Table 2: Results of complex activation of networks with three and four neurons in the output layer

Topology $708 - 240 - 6$, the training set contains 937 and 1,237 samples (for 100 and 200 samples on the 4th to 6th neurons). The neural network again has significantly better results, finding the *Buy* signal in 1,472 to 1,778 cases, and for the 200 variant in 1,067 to 1,117 cases. This is an improvement of 17-27% (respectively 37-42%) in removing *Buy*, *Sell* and *Zero* spurious signals over the four neurons in the output layer. The overall improvement over the three neurons in the output layer is then 47-53% (respectively 66%).

There has been a change in the ratio between the signals we require and the auxiliary signals (4th to 6th neurons). For networks with four neurons in the output layer, the ratio of generated signals was only 11% in favor of auxiliary neurons (738.5 out of 6,570.25) and 24% (1,643.25 out of 6,609), respectively, while for neural networks with six output neurons, this ratio is already 25% (1607.75 out of 6328.75) and 47% (2906.5 out of 6102.25), respectively. This condition is desirable because the adaptation of the network to *Buy*, *Zero* and *Sell* signals is being refined.

The adaptation rate with 708 neurons in the input layer begins to reach 9,384 to 9,911 cycles in 1,237 training

samples, so an additional input bar was added and the topology was changed to 1,180 – 240 – 6, where the adaptation rate is only 1,887 to 2,300 cycles.

Topology 1,180 – 240 – 7, the training set contains 1,437 samples (200 samples on the 4th to 7th neurons), the testing set contains 14,375 samples as usual. The 7th neuron was created by dividing the 4th neuron, so that only one new neuron was added to the network topology in the output layer, so it is not surprising that the improvement in network results is not so great. The improvement for *Buy* reaches a maximum of 5%, for *Zero* 10% and for *Sell* 15%.

Topology 1,652 – 240 – 9, the training set contains 1,837 samples. The improvement for *Buy* is 31-32%, for *Zero* 26-30% and for *Sell* 32-34%. The total number of identified signals then dropped below six thousand in three cases out of four for the first time and remained in the range from 5,894 to 6,008. The ratio of generated signals is now 66% in favor of auxiliary neurons (3,946.25 out of 5,954).

Topology 2,124 – 240 – 15 and topology 2,360 – 240 – 15, the training set contains 3,067 samples. We decided to add more output neurons at once – a total of 15, as shown in Table 1. The adaptation rate was 610 to 800 cycles and 560 to 880 cycles, respectively, indicating that it deteriorated in some cases. We got to the edge of the suitability of the topology, and therefore we chose the better one for further optimization, i.e. the topology 2,124 – 240 – 15 with nine bars at the input. In terms of improving the network’s results, we found that there was an improvement again, but it is no longer a leap. With each increasing neuron in the output layer, the network adaptation results will improve, however, this improvement will be smaller and smaller. That’s why we decided to change our optimization strategy.

3.4 Business system optimization

We decided to perform a simulation on a time series and observe what results of adaptation and complex activation we will get. We will now optimize the entire trading system based on 4 separate neural networks. The division of the training and test set is in a certain ratio (for example, [15] uses 70:30), we use a floating layout, where the training set includes data with a minimum size of two weeks, a maximum of eight and test data are always from one week. The training set is generated by an automaton, which gradually goes through the historical data after a minute, and assigns them the corresponding values of the output signal.

Two-week data Training set: 2,543 samples (October 1 – 14, 2018), *Buy* 208 times, *Zero* 56 times, *Sell* 115 times, duration: 503 - 702 cycles, test set: 14,371 data. As we can see in table 3, all trading systems show a weekly profit, the best ones almost \$ 900. The individual settings of trading systems are based on the names from the chapter 3.1.

The best weekly earnings results are highlighted in bold. Strikethrough are settings that show worse or similar results than their previous variation. We will no longer count on these settings. The underlined value of gross loss signals an excellent ratio of gross profit versus gross loss.

Settings	Transactions	Net profit	Gross profit	Gross loss	Weekly profit
4	55	265.36	310.20	-44.84	132.68
3	275	1,191.86	1,395.70	-203.84	595.93
2	729	1,777.43	2,561.38	-783.95	888.715
3K	274	1,189.96	1,393.80	-203.84	594.98
2K	713	1,784.93	2,540.78	-755.85	892.465
4M	2	17.30	17.30	0	8.65
3M	42	307.50	324.60	-17.10	153.75
2M	131	835.60	890.30	-54.70	417.80
4N	41	245.30	271.40	-26.10	122.65
3N	168	986.40	1,094.20	-107.80	493.2
2N	415	1,574.70	1,980.20	-405.50	787.35
3KM	42	307.50	324.60	-17.10	153.75
2KM	127	843.70	890.30	-46.60	421.85
3KN	167	984.50	1,092.30	-107.80	492.25
2KN	407	1,572.10	1,962.40	-390.30	786.05
4MN	2	17.30	17.30	0	8.65
3MN	26	223.10	228.90	-5.80	111.55
2MN	83	668.80	681.10	-12.30	334.4
3KMN	26	223.10	228.90	-5.80	111.55
2KMN	82	668.80	681.10	-12.30	334.4

Table 3: Results of a trading system with a two-week set of training data

In the figure 1 we see the balance on the trading account of *The Two system with a correction*.

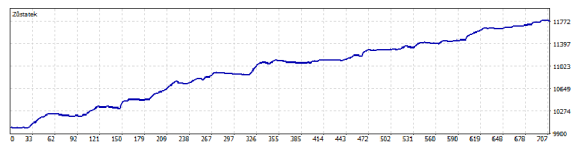


Figure 1: The Two system with a correction - **2K**

The validation of trading systems was performed on new data from the period from 15 to 21 October of the same year. The success of a data set that the network has never dealt with is significantly lower, and neither setting generates any gain but loss, as we can see in table 4. Although the network is able to predict movements in the currency market, it has not been able to generalize training to data that was not part of the training set.

Settings	Transactions	Net profit	Gross profit	Gross loss
4	19	-31.20	11.90	-43.10
3	81	-87.11	52.50	-139.61
2	298	-255.75	239.30	-495.05
2K	282	-258.25	218.70	-476.95
3M	3	-3.50	2.30	-5.80
3N	44	-31.20	47.80	-79.00
2N	145	-103.50	174.20	-277.70
2KM	19	-32.50	12.90	-45.40
2KN	135	-104.40	162.80	-267.20
2MN	14	-21.60	12.10	-33.70

Table 4: Results of the trading system with a two-week set of training data for the new week

Three-week data Training set: 2,888 samples (September 24 to October 14, 2018), *Buy* 299 times, *Zero* 72 times, *Sell* 229 times, duration: 1,116 - 1,316 cycles, test set: 21,565 data. The maximum weekly profit is around \$ 600

to \$ 700, see table 5. None of the settings used reached the same ratio between gross profit and gross loss as in the two-week data. For two-week data, this ratio was even equal to 55.37, for three-week data it ranges from 3 to 5.

Settings	Transactions	Net profit	Gross profit	Gross loss	Weekly profit
4	237	779.64	1,057.30	-277.66	259.88
3	823	1,816.95	2,779.45	-962.50	605.65
2	1,742	2,018.73	4,284.10	-2,265.37	672.91
2K	1,679	2,050.93	4,212.40	-2,161.47	683.64
3M	116	618.86	717.70	-98.84	206.28
3N	488	1,451.30	2,070.20	-618.90	483.76
2N	988	1,766.30	3,189.10	-1,422.80	588.76
2KM	260	972.95	1,247.69	-274.74	324.31
2KN	950	1,783.50	3,136.70	-1,353.20	594.5
2MN	169	633.30	851.10	-217.80	211.10

Table 5: Results of a trading system with a three-week set of training data

The validation of trading systems was performed on new data from the period from 15 to 21 October of the same year. Setting 4 is profitable, however \$ 10.69 per week is not a successful strategy. If we look at the profitability of trades by hour, then it is clear that this setting has the largest losses during the night hours. The 3M setup is also profitable, however, only 11 transactions per week and a net profit of less than \$ 30 is still low. There was an increase in the number of transactions for all trading systems, ranging from 1.76 times to 3.66 times, compared to trading systems with a two-week set of training data. There was an increase in net profit for the 4, 3M, 3N, 2KM and 2MN settings, gross profit for all settings and gross loss for all but 3M, where the gross loss decreased from \$ 5.80 to \$ 1.50. See table 6.

Settings	Transactions	Net profit	Gross profit	Gross loss
4	44	10.69	64.30	-53.61
3	206	-113.91	224.40	-338.31
2	526	-382.96	486.40	-869.36
2K	510	-363.56	471.80	-835.36
3M	11	28.60	30.10	-1.50
3N	112	-28.20	179.70	-207.90
2N	264	-146.20	358.20	-504.40
2KM	51	5.00	72.20	-67.20
2KN	250	-129.00	343.60	-472.60
2MN	32	23.70	61.60	-37.90

Table 6: Results of the trading system with a three-week set of training data for the new week

Because the adaptation on all neural networks took more than two days, we decided to shorten the training data set, where in addition to the Buy, Sell and Zero signals, the signals of the 10th and 15th neurons remain, which are in the immediate vicinity of the Buy and Sell signals.

Three-week data truncated Training set: 888 samples (September 24 to October 14, 2018), Buy 299 times, Zero 72 times, Sell 229 times, duration: 106 - 151 cycles, test set: 21,565 data. The results of the three-week truncated data set are remarkable, see table 7. The maximum weekly profit is around \$ 700 to \$ 820, which is an improvement over the full data set by more than \$ 100. The number of transactions increased rapidly from a maximum of 1,600 to 1,700 to 7,000 and 9,000.

Settings	Transactions	Net profit	Gross profit	Gross loss	Weekly profit
4	1,164	1,500.13	2,995.16	-1,495.03	500.04
3	4,390	1,610.37	7,763.35	-6,152.98	536.79
2	9,275	-716.29	13,167.30	-13,883.59	-238.76
3K	4,050	1,785.29	7,424.67	-5,639.38	595.09
2K	7,378	418.79	11,157.34	-10,738.55	139.59
4M	187	573.48	757.78	-184.30	191.16
3M	1,226	1,914.46	3,423.11	-1,508.65	638.15
2M	3,604	1,901.36	6,931.41	-5,030.05	633.78
4N	645	1,447.70	2,295.60	-847.90	482.56
3N	2,339	2,390.50	5,907.90	-3,517.40	796.83
2N	4,762	1,696.00	9,700.10	-8,004.10	565.33
3KM	1,108	1,942.10	3,271.23	-1,329.13	647.36
2KM	2,615	2,165.52	5,563.73	-3,398.21	721.84
3KN	2,167	2,454.40	5,662.20	-3,207.80	818.13
2KN	3,793	2,246.00	8,331.20	-6,085.20	748.66
4MN	118	409.20	533.40	-124.20	136.4
3MN	674	1,692.20	2,547.80	-855.60	564.06
2MN	1,946	2,337.60	5,192.30	-2,854.70	792.53
3KMN	618	1,686.80	2,433.80	-747.00	562.26
2KMN	1,399	2,394.30	4,207.50	-1,813.20	798.1

Table 7: Results of a trading system with a three-week truncated set of training data

Although the results of the trading system are very satisfactory within the training period, the validation on the new week from 15 to 21 October of the same year proves the exact opposite. None of the tested trading systems shows a net profit.

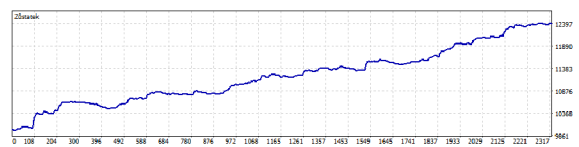


Figure 2: The Three and not night trading system - 3N

Four-week data truncated Training set: 1,179 samples (September 17 to October 14, 2018), Buy 381 times, Zero 95 times, Sell 313 times, duration: 184 - 197 cycles, test set: 28,758 data. Validation performed for a week from 15 to 21 October. Only one setting shows a net profit, the number of transactions has increased almost everywhere. See Table 8 for emergence of several categories:

1. improvement of net profit, increase of gross profit and decrease of gross loss
2. improvement of net profit, improvement of gross profit, but also increase of gross loss
3. a decrease in net profit, but at the same time there was an increase in gross profit, which increase is a percentage higher than the increase in gross loss
4. a decrease in net profit, but at the same time there was an increase in gross profit, which increase is already a percentage lower than the increase in gross loss
5. increase net loss, decrease gross profit and increase gross loss

Trading system *Three* falls into the best categories, i.e. 1st and 2nd, trading system *Two* then into the worst categories, i.e. 4th and 5th. The course of validation of the 3KMN trading system and the amount of the balance on the trading account can be seen in the figure 3.

Settings	Transactions	Net profit	Gross profit	Gross loss	Cat
4	406	-158.86	442.30	-601.16	1.
3	1,596	-1,048.69	1,430.30	-2,478.99	3.
3K	1,490	-862.48	1,382.70	-2,245.18	2.
3M	396	-167.12	428.10	-595.22	1.
2M	1,172	-891.84	1,008.10	-1,899.94	5.
3N	830	-386.30	1,104.70	-1,491.00	2.
2N	1,614	-1,164.90	2,029.00	-3,193.90	5.
3KM	358	-103.22	418.40	-521.62	1.
2KM	835	-576.83	773.50	-1,350.33	5.
3KN	784	-290.70	1,073.30	-1,364.00	2.
2KN	1,297	-820.10	1,659.60	-2,479.70	4.
3MN	236	-44.80	359.40	-404.20	2.
2MN	588	-459.60	707.30	-1,166.90	5.
3KMN	218	6.70	355.00	-348.30	2.
2KMN	437	-291.20	561.80	-853.00	5.

Table 8: Results of the trading system with a four-week truncated set of training data for the new week



Figure 3: The Three with correction, minimal 2 and not night trading system - 3KMN

Five-week data truncated Training set: 1,386 samples (September 10 to October 14, 2018), *Buy* 464 times, *Zero* 121 times, *Sell* 401 times, duration: 253 - 283 cycles, test set: 35,951 data. Validation performed for a week from 15 to 21 October. As we can see in table 9, no setting shows a net gain. There has been an increase in transactions everywhere. Again, there was a division into five categories:

Settings	Transactions	Net profit	Gross profit	Gross loss	Cat
4	406	-158.86	442.30	-601.16	1.
4	533	-374.80	486.80	-861.60	4.
3	1,798	-1,188.91	1,694.60	-2,883.51	4.
3K	1,700	-1,049.11	1,640.20	-2,689.31	4.
3M	495	-303.42	527.30	-830.72	4.
2M	1,348	-787.61	1,374.00	-2,161.61	2.
3N	957	-455.10	1,293.10	-1,748.20	4.
2N	1,744	-665.60	2,486.60	-3,150.20	1.
3KM	445	-242.72	502.80	-745.52	4.
2KM	1,020	-586.12	1,081.50	-1,667.62	3.
3KN	912	-390.40	1,254.10	-1,644.50	4.
2KN	1,482	-620.00	2,076.30	-2,696.30	2.
3MN	281	-111.10	406.40	-517.50	4.
2MN	732	-211.50	1,090.80	-1,302.30	2.
3KMN	259	-83.40	391.00	-474.40	5.
2KMN	578	-127.60	887.80	-1,015.40	2.

Table 9: Results of the trading system with a five-week truncated set of training data for the new week

1. improvement of net profit, increase of gross profit and decrease of gross loss
2. improvement of net profit, improvement of gross profit, but also increase of gross loss
3. a decrease in net profit, but at the same time there was an increase in gross profit, which increase is a percentage higher than the increase in gross loss
4. a decrease in net profit, but at the same time there was an increase in gross profit, which increase is already a percentage lower than the increase in gross loss
5. increase net loss, decrease gross profit and increase gross loss

The best and worst trading systems have been reversed, with the *Three* no longer the best, but falling into the 4th and 5th categories. In contrast, the trading system *Two* shows the best results - it moved to the 1st and 2nd category.

Based on these results, we conclude that the adaptation of neural networks did not generalize in such a way as to create a self-sufficient business system. We will now test this hypothesis on an eight-week data set.

Eight-week data truncated Training set: 1,988 samples (August 20 to October 14, 2018), *Buy* 751 times, *Zero* 185 times, *Sell* 652 times, duration: 750 - 903 cycles, test set: 57,538 data. Validation performed for a week from 15 to 21 October. As we can see in Table 10, all tested trading systems show worse results with an eight-week set of training data than with a five-week or four-week set. We therefore confirmed our hypothesis that the neural network did not generalize sufficiently.

Settings	Transactions	Net profit	Gross profit	Gross loss
4	1,229	-998.11	1,094.00	-2,092.11
3	2,690	-2,004.60	2,410.26	-4,414.86
3K	2,614	-1,942.39	2,347.26	-4,289.65
3M	892	-732.91	771.00	-1,503.91
2M	1,650	-1,190.95	1,506.20	-2,697.15
3N	1,354	-783.70	1,803.60	-2,587.30
2N	1,963	-1,182.30	2,600.00	-3,782.30
3KM	853	-662.50	747.60	-1,410.10
2KM	1,379	-989.54	1,269.80	-2,259.34
3KN	1,313	-743.80	1,768.30	-2,512.10
2KN	1,763	-1,060.60	2,347.90	-3,408.50
3MN	444	-311.20	580.50	-891.70
2MN	818	-451.50	1,113.40	-1,564.90
3KMN	421	-252.60	565.40	-818.00
2KMN	687	-335.50	949.60	-1,285.10

Table 10: Results of the trading system with an eight-week truncated set of training data for the new week

4 The best setting for the trading system

As part of the search for the ideal trading system, we found only one that was able to generate a net profit even on unadapted data. It was a **3KMN** system with a four-week truncated data set – graphs in figures 4 and 5. It reported a net weekly profit of \$ 6.70 for the new week. In contrast, it generates a net profit of \$ 1,660.50 on the learned dataset, which is a weekly average of \$ 415.12. The weekly number of transactions is 218 in the first case and 216.75 in the second.

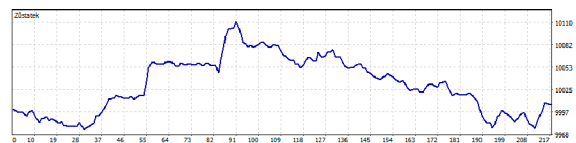


Figure 4: 3KMN with a four-week truncated set of training data for the new week

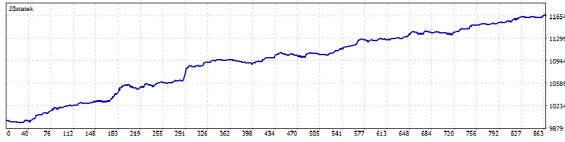


Figure 5: **3KMN** with a four-week truncated set of training data

The second best system was **3MN** also on the four-week data set (graphs in figures 4 and 5), which generated a net loss of \$ -44.80 on the new week and then reported a net gain of \$ 1,724.80 on the learned data set, the weekly average then \$ 431.20. The weekly number of transactions is 236 in the first case and 242.5 in the second.

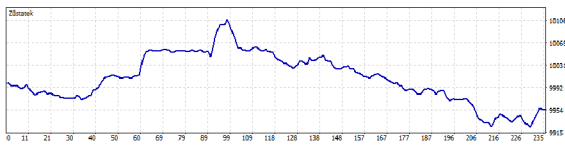


Figure 6: **3MN** with a four-week truncated set of training data for the new week

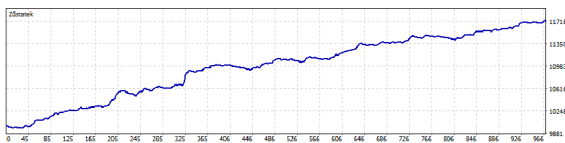


Figure 7: **3MN** with a four-week truncated set of training data

The third best trading system was the **3KMN** variant on the five-week dataset (graphs in figures 8 and 9), which reported a net profit of \$ -83.40 on the new week, with a net profit of \$ 1,910.80 and a weekly average of \$ 382.16 on the adapted dataset. The weekly number of transactions is 259 in the first case and 266.8 in the second.

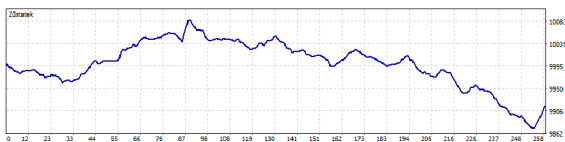


Figure 8: **3KMN** with a five-week truncated set of training data for the new week

There is a clear difference between the application of the trading system on adapted and non-adapted data. It is obvious that the neural network could not be generalized enough to be able to generate profit even on unknown data.

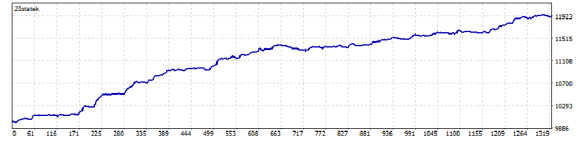


Figure 9: **3KMN** with a five-week truncated set of training data

5 Conclusion

Several conclusions emerged from the search for the optimal setting of the trading system. The first is the fact that the neural network was able to adapt to both unabridged and abbreviated data sets, and this did not have a significant effect on trading in the currency market. On the training dataset, the trading system worked relatively well. Net profit for the best systems ranged from \$ 382.16 to \$ 431.20 per week.

Another conclusion can be made that the constant expansion of the size of the time period for the training data set is not the most suitable solution, because the two most successful trading systems are from a four-week period, one from a five-week period. The eight-week period did not generate a single successful trading system, and even resulted in worse than five and four weeks. We also performed a test on a **2KMN** trading system with nine bars in the input layer, topology 2,124 – 420 – 15. When we extended the training set to an entire calendar year, the result was a net profit of \$ 22.2 on unknown data.

The neural network, although doing relatively well on the trained data, was unable to perform well with the untrained test set due to overfitting. Of all the settings and trading systems tested, only one variant reported a net profit on the new unlearned week, with this net weekly gain being only \$ 6.70. This indicates that raw trading data series may not have enough regularities that could be successfully learned by the network. It will be necessary to optimize various types of input aggregated from the input dataset, and to include external information possibly influencing the market. It will be necessary to use a larger training data set, which the computing power of the GPU will allow.

Another improvement will be the use of deep learning methods [13] with various types of layers suitable for time series prediction. Therefore, in the next phase, the application will be migrated from the *C#* programming language to *Python*, and the *Keras* and *Tensorflow* libraries will be used to find the optimal topology of the trading system. We will, e.g., compare results of convolutional neural networks and recurrent neural networks, including LSTM or GRU networks, multimodal networks and fuzzy-neural networks. The transition to this technology using GPU will also make it possible to work with a larger amount of data. We tested that switching to *Keras* libraries would allow to work with a ten-fold larger data set, so the adap-

tation set can include data from the entire calendar year.

The complex modular system will be completed by a separate group of fuzzy-neural networks, which will form a set of rules, which will then be applied in online trading in the currency market. By combining all these separate modules, we will obtain an artificial intelligence tool that will incorporate many possible approaches to time series prediction.

Acknowledgements

This work was supported by the Ministry of Education, Youth and Sports Of the Czech Republic from the National Programme of Sustainability (NPU II) project IT4Innovations Excellence in Science - LQ1602, and by the Silesian University in Opava under the Student Funding Scheme, project SGS/11/2019.

References

- [1] ABRAHAM, A., CHOWDHURY, M. U. *An Intelligent FOREX Monitoring System*. In Proceedings of IEEE International Conference on Info-tech and Infonet, Beijing, China, page 523-528, 2001.
- [2] *Accumulation/Distribution (A/D)* [online]. Copyright © 2009 - 2019 FXstreet.cz s.r.o. [cit. 17. 6. 2019]. <https://www.fxstreet.cz/accumulationdistribution-ad.html>
- [3] *CMF – Chaikin Money Flow* [online]. Copyright © eTradink.sk [cit. 17. 6. 2019] <https://www.etrading.sk/cz/technicka-analyza/116-indikator/649-cmf-chaikin-money-flow>
- [4] *Discover the Advantages of the MT4 Moving Average Indicator* [online]. [cit. 16. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/moving-average-indicator>
- [5] *Everything You Need to Know About the Oscillator of Moving Average (OsMA Indicator)* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/osma-indicator>
- [6] *Find the Truth Behind the Trend with the MT4 Momentum Indicator* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/momentum-indicator-mt4-explained-in-detail>
- [7] *Force Index - FRC* [online]. [cit. 17. 6. 2019]. https://www.instaforex.eu/cz/forex_technical_indicators/frc
- [8] *Forex Trading Strategies With Envelopes Indicator* [online]. Copyright © 2019 Dolphintrader.com. [cit. 17. 6. 2019]. <https://www.dolphintrader.com/forex-trading-strategies-with-envelopes-indicator/>
- [9] *Get a Feel for Market Momentum with the Awesome Oscillator Indicator* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/awesome-oscillator-indicator>
- [10] *Get Ahead of the Curve with the MT4 Commodity Channel Index Indicator* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/commodity-channel-index-indicator>
- [11] GALESCHUK, S. *Neural networks performance in exchange rate prediction*. Neurocomputing, 172, pages 446-452, 2016.
- [12] GOH, A. T. C. *Back-propagation neural networks for modeling complex systems* Artificial Intelligence in Engineering, vol. 9, num. 3, pages 143-151, 1995, issn 0954-1810.
- [13] GOODFELLOW, I., BENGIO Y., COURVILLE A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>
- [14] *How to Use the Parabolic SAR Indicator to Trade the Trend* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/parabolic-sar-indicator>
- [15] KIPRUTO, G., MUNG'ATU, J., ORWA, G., GAITHIMBA, N. *Application of Artificial Neural Network (Ann)*. In Modeling Foreign Currency Exchange Rates. International Journal of Scientific Research and Management. vol 6, issue 10, pages 111-123, 2018. ISSN: 2321-3418
- [16] LAM, M. *Neural network techniques for financial performance prediction: integrating fundamental and technical analysis*. In *Decision Support Systems*, vol. 37, Issue 4, pages 567-581, 2004.
- [17] LAVANYA, V., PARVEENTAJ, M. *Foreign Currency Exchange Rate (FOREX) using Neural Network*. In International Journal of Science and Research (IJSR 2013) ISSN (Online): 2319-7064.
- [18] *The MACD Indicator In Depth* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/macd-indicator-in-depth>
- [19] MANSOUR, F., AKAY, M. *Predicting Exchange Rate by Using Time Series Multilayer Perceptron*. In Science and engineering congress (IMSEC), 2019.
- [20] POBUCKÝ, M. *Neuronový obchodní systém pracující v reálném čase*. Opava: Silesian University, The Institute of Computer Science, Rigorous thesis, 2020. 118 s.
- [21] *Put the Pedal to the Metal With the Accelerator Oscillator* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/accelerator-oscillator>
- [22] *See Beneath the Surface of the Market With the Bears and Bulls Power Indicators* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/bears-and-bulls-power-indicator>
- [23] SEZER, O. B., GUDELEK, M. U., OZBAYOGLU, A. M. *Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019*. In Applied Soft Computing, vol 90, 2020.
- [24] *Track the Pulse of the Market with the Money Flow Index Indicator* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/money-flow-index-indicator>
- [25] *Trading With the Cloud: Using the Ichimoku Kinko Hyo Indicator in MetaTrader 4* [online]. [cit. 17. 6. 2019]. <https://admiralmarkets.com/education/articles/forex-indicators/ichimoku-kinko-hyo-indicator>
- [26] *Vidya – Variable Index Dynamic Average* [online]. Copyright © eTradink.sk [cit. 16. 6. 2019] <https://etrading.sk/cz/technicka-analyza/116>

indikatory/639-vidya-variable-index-dynamic-average

- [27] VUKOVIC, D., VYKLYUK, V. *Forex prediction with neural network: usd/eur currency pair*. In Actual Problems of Economics. vol 10, pages 251-261, 2013.
- [28] %R: Williams Percent Range [online]. Copyright © 2009 - 2019 FXstreet.cz s.r.o. [cit. 17. 6. 2019]. <https://www.fxstreet.cz/r-williams-percent-range.html>
- [29] YONG, Y. L., NGO, D. C. L., LEE, Y. *Technical Indicators for Forex Forecasting: A Preliminary Study*. In Advances in Swarm and Computational Intelligence, 6th International Conference, ICSI, Lecture Notes in Computer Science book series (LNCS, volume 9142), pages 87 - 97, 2015.
- [30] ZHANG, X., TAN, Y. *Deep stock ranker: A LSTM neural network model for stock selection*. In Data Mining and Big Data, pages 614–623. Springer International Publishing, 2018.