UNIVERSITY OF TARTU
Faculty of Economics and Business Administration
Institute of Business Administration
Chair of Money

Anton Golovko

FOREIGN EXCHANGE RATE MOVEMENT PREDICTION USING TRIANGLE CHART PATTERNS AND ARTIFICIAL NEURAL NETWORKS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Business Administration

Supervisor: doctoral student Allan Teder, University of Tartu
Co-supervisor:Eduard Petlenkov, PhD, Tallinn University of Technology

Tartu 2013

Soovitan suunata kaitsmisele …………………………………..

(juhendaja allkiri)

Kaitsmisele lubatud “ “....................... 2013. a.

Rahanduse õppetooli juhataja …………………………

(õppetooli juhataja nimi ja allkiri)

Olen koostanud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, põhimõttelised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

…………………………………..

(töö autori allkiri)

TABLE OF CONTENTS

Introduction 5
1. Theoretical overview 11
   1.1. Fundamentals of Foreign Exchange 11
   1.2. Usage of triangle chart patterns for rate movement direction prediction 14
   1.3. Fundamentals of artificial neural networks 21
      1.3.1. Neural networks concepts and application domain 21
      1.3.2. Overview of neuron’s activation functions 26
      1.3.3. Neural network training and results analysis related concepts 29
   1.4. Overview of other researches 34
2. Investigation: currency rate movement prediction using triangle chart pattern and neural networks 38
   2.1. Data and methodology 38
   2.2. Finding triangles chart pattern samples 43
   2.3. Choosing neural network output parameters 49
   2.4. Choosing neural network input parameters 53
      2.4.1. Rates series descriptors 53
      2.4.2. Triangle descriptors 56
   2.5. Neural network training data preparation 61
   2.6. Choosing neural network architecture and training 63
   2.7. Results analysis 66
Conclusion 72
References 75
Appendix 1. Neural network experiments results summary 80
Appendix 2. Neural network experiments results details 82

Appendix 3. Description of implemented software applications 90
Appendix 4. MATLAB commands used to create neural network 91
Appendix 5. Correlations of triangle descriptors with the price move direction 92
Resümee 93

INTRODUCTION

Financial markets are developing and globalizing constantly offering more and more instruments that can be traded and also reaching wider and wider variety of participants. The instruments are not just used only for long-term and strategic investments, but also for achieving short-term speculative profits. Some markets with high volatility and liquidity also make possible so-called intra-day trading when positions are kept open only for few hours or even minutes or seconds (e.g. Foreign Exchange market). Participants in financial markets vary from big financial institutions and enterprises to small private traders.

Such development and popularization of trading on financial markets constantly increases a demand for new information sources and tools that could be used to facilitate trading and make it more profitable. Traders use the data and the tools for making their trading decisions and also for trading automation. The analysis methods used in the trading fall into two major categories: fundamental analysis and technical analysis. The fundamental analysis stands for finding relations between instrument's price movements and fundamental indicators. The main sources of information for such analysis are statistics of different economical parameters and news. The technical analysis, in contrast, is based on an assumption that instrument's price changes have regularities and that analysis of historical data of price changes can be used to predict future changes.

The technical analysis theory offers hypotheses that are based on a rich set of statistical indicators and chart patterns that can be used by traders to make their trading decisions. The chart patterns based approach usually involves a visual analysis of recent price movement charts by identifying certain shapes in the charts (such as so-called triangles, head and shoulders, wedges, channels etc.) that are supposed to predict the future price change direction. Chart pattern analysis is one of the simplest and cleanest forms of forecasting stock prices that can be applied without additional indicators or tools; and often there is little or no lag between signals and subsequent price movement (Bhandari 2013: 34). Chart formations are especially profitable for intra-day traders (Bulkowski 2005a: 7).

Triangles are seen as ones of the most reliable and simple patterns (Bhandari 2013: 36; Murphy 2012). The hypothesis regarding the triangles assumes that converging of price movements into a triangle can be a signal for a quick price movement breakout out of the triangle in a certain direction. Knowing the direction, a trader can make a trading decision. For example, if a rate should go up then a trader can get a profit opening a long position. However, it has to be said that there are also opposite opinions regarding triangle chart patterns reliability (Kamish 2009: 83).

Anyway, the major problem of any chart patterns in general is that the shapes cannot guarantee the correctness of a prediction and can suffer from low prediction accuracy (as is demonstrated in this paper, chapter 2.7). In addition to that, the chart patterns analysis is traditionally done mostly intuitively: traders do their pattern analysis based on their own understanding and experience taking into account very many different parameters such as shape recognition criteria, price movement details before the shape, price movement details within the shape, trades volume, day time, additional indicators etc. This is because of the fact that also the chart pattern definitions are often subjective (Anand *et al* 2001: 134) and there is no precise and really working guideline, algorithm or formula that would clearly define how to make a correct decision based on a found chart pattern. Instead, the regularities noticed by different technical analysis theorists have differences and their conclusions are sometimes contradicting. Also some academics and investors believe that the patterns don’t exist, but technical analysts swear they do, however the last are usually not mathematically sophisticated enough to demonstrate the validity of the chart patterns (Kirkpatrick, Dahlquist 2010: 303).

Another technique used for prediction of price movements is artificial neural networks. Financial markets volatility, complexity and their noisy environment make artificial neural networks a good candidate for that (Yao, Tan 2002: 191). Artificial neural networks have proven very helpful in solving prediction and classification tasks that involve finding complex non-linear relations between different parameters and processes. Artificial neural networks (further in the paper just “neural networks”) are studied by artificial intelligence and data mining subfields of computer science. Neural networks can be used to predict price movements based on different parameters that could include both fundamental and technical indicators. Neural network based model have proved working and credible for example for next currency rate prediction (Dunis, Williams 2002: 19-20). The advantage of neural network based models over traditional forecasting methods is because, as is often the case, the model best adapted to a particular problem cannot be identified. This means that it is better to resort to method that is a generalization of many models, than to rely on a priori models (Dunis, Williams 2002: 11).

This work aims to evaluate a currency rates movement prediction model built using both triangle chart patterns and a neural network that analyses the triangle chart patterns in order to find relationships between the pattern itself and the rate movement after the triangle. This work involves design and creation of the prediction model that then has to be evaluated from prediction accuracy perspective, and the most important indicator of the accuracy is the percentage of correctly forecasted rate movement directions. Also it is important to compare the achieved results with the prediction accuracy that could be achieved by just following triangle chart patterns instructions as per their description in the theoretical part of the given paper. Achieving the goal of the research means reaching conclusions regarding the best neural network models, their results and the applicability of the created models for currency rate movement direction prediction. If using a neural network can achieve a higher prediction accuracy than the usual use of triangle patterns then such prediction model definitely provides an added value. Having a better mechanism to predict correct price movement directions, a trader can improve quality of trading decisions.

The triangle chart pattern was chosen, because typically it is considered as the most reliable chart pattern compared to other chart patterns (Bhandari 2013: 36; Murphy 2012) and all its three sub-types were considered in the given research: ascending, descending and symmetrical triangles. This work uses neural networks to identify relations between the triangle chart pattern parameters defined by the hypotheses and the real price movements. An ability to identify any non-linear relationships between parameters was decisive for choosing neural networks for the analysis. The model is built and tested based on Foreign Exchange market which was chosen, because it is the most liquid and volatile financial market (Shamah 2003: 23) which in own turn means that triangle chart pattern can be more frequent for currency rates than any other financial instrument; and this is especially important as there is a need to build a prediction model which bases on triangles as a part of this work. Also, having a sufficient number of triangle samples in historical data is crucial for successful neural network modeling and training.

This topic was chosen, because a success to find such complex relations between the triangle chart patterns characteristics and the rate movement direction can make the chart pattern based trading more profitable than the traditional pattern based trading that involves only a visual and an intuitive analysis of the patterns. A successful prediction model can potentially be used by banks, funds, insurance companies and also retail foreign exchange traders to improve profitability of their trading. As a neural network implementation is usually a software, the neural network is able to produce answers automatically and this means that such models can be used not only for manual analysis, but also in automated trading systems which run on a computer and are able to operate on a market without human intervention and enter/exit trades based on a set on rules. In case of not reaching a model with high prediction accuracy, important conclusions can be made regarding applicability of neural network based analysis of triangle chart pattern properties for rate movement direction prediction. Also the work and conclusions reached in the work can serve as an input or a baseline for other future scientific investigations which involve applying of chart patterns and neural networks for prediction in financial markets. A similar approach that uses a neural network could be re-used also in researches regarding other chart patterns or technical indicators.

Once the goal of the paper was defined, the first task of the research was to study neural networks theory and to investigate works done by other researchers that involved using neural networks for financial trading. This was needed to evaluate applicability of neural networks for financial forecasting and to re-use the existing experience in defining the scope of neural network related experiments. The experiments involve trying very different neural network architectures, different parameters and methods. That's why it is important to limit the scope of the experiments with the most appropriate configurations and neural network architectures.

The second task done in the work was to analyze triangle chart patterns theory in order to implement a software application that can be used for automatic triangles search in the historical data of currency rates. This was needed in order to avoid a manual triangles search which cannot be acceptable for doing the research, because working with neural networks requires massive amounts of triangle examples.

The third task was to define what value the model has to predict. This task also involves providing the correct decision regarding rate change direction for every triangle example. This was needed to predefine the desired target outputs for neural network training.

The last task was to create appropriate neural network models suitable for currency rate movement direction prediction based on a triangle chart pattern and to make experiments with the models. This step involves choosing a neural network model, making experiments, analyzing results, constantly improving the approach and confirming the improvements by making additional experiments. The final step of the task is to conclude if the created neural networks are able to predict rate change direction after triangle breakout points, to compare the traditional method of using triangle chart patterns with the created neural network based method and to make the corresponding conclusions.

The paper consists of two major parts:

i. Theoretical introduction to foreign exchange trading, triangle chart patterns and artificial neural networks. This part also covers foreign exchange, triangles and neural networks related terminology that is used in the empirical part to describe the investigation. The part provides also an analysis of some differences, ambiguities, non-applicability and contradictions in triangle chart pattern definitions made by different authors. The first part also summarizes some previous researches done in the technical analysis domain that use artificial neural networks.

ii. Empirical part goes through all the steps and conclusions that were done in the research. As the goal of the research was to build and to evaluate a neural network based rate movement prediction model, the empirical part describes the process of creation of the model and decision flows required to define the model that would be appropriate for rates movement direction prediction. The first chapter in the part defines the problem that has to be solved by the created neural network model, model creation methodology and the data. The subsequent chapters cover the steps of neural network creation. The last chapter overviews the tested neural network models, covers in more details those models that achieved the highest prediction accuracy, analyzes the results and describes conclusions and achievements of this investigation. The part is followed by a conclusion chapter that summarizes the steps required to achieve the goal of the work and concludes the results.

Author’s previous education (MSc) and experience in computer science leveraged the research in technical analysis and artificial intelligence domains with software implementation of the ideas. Also supervisor’s practical experience in trading helped with the clear goal definition and with chart pattern’s related advices. Co-supervisor’s review of the work and advises were important to make sure that neural networks are modeled in an appropriate way for the defined problem.

Software programs implemented by the author as part of this work are described in the Appendix 3.

1. THEORETICAL OVERVIEW

1.1. Fundamentals of Foreign Exchange

Foreign Exchange market has existed since the advent of money. An operation on the Foreign Exchange market is buying one currency for another currency, i.e. trading takes place around currency pairs. The Foreign Exchange market enables international trade (e.g. goods exchange) and international investment through offering the currency conversion.

In the market, exchange rates are driven by supply and demand (in case of free-floating currencies) that determine their exchange rate. The Foreign Exchange market is not centralized and the main participants are larger international banks that trade between each other either directly (via e.g. SWIFT – Society of Worldwide Inter-bank Financial Telecommunication to settle their transactions) or through electronic matching platforms such as Reuters and EBS (Silvani 2008: xiv). Also other companies such as hedge funds, insurance companies and other financial institutions are involved. This forms a so called inter-bank Foreign Exchange market.

In addition to the interbank market, banks also provide a retail Foreign Exchange market for their clients (Wang 2010: 5) directly or via middlemen (brokers). In own turn, Retail Foreign Exchange Brokers (also referred as Futures Commission Merchants) open up the spot currency market for smaller retail traders. The brokers are middlemen between a market maker (a bank) and retail clients offering their trading platforms and charging a fee for their service (Silvani 2008: xiv).

Currency trading between institutions is not directly regulated by governments, although retail trading by individuals is usually regulated (Rockefeller, Schmelzer 2013) in some countries. The governments apply regulative requirements to the market makers and brokers. For example in the UK, retail Foreign Exchange is regulated by the Financial Conduct Authority (FCA); and in the USA, the retail trading is regulated by the governmental Commodity Futures Trading Commission (CFTC) and by the National Futures Association (NFA) in USA (O'Keefe 2010: 24-25). The last is an independent self-regulatory organization that oversees commodities and futures industry in the United States in order to protect investors or traders from fraudulent commodities and futures activities. Also some governments partially regulate the Foreign Exchange market by trying to stabilize the exchange rate for their domestic currency against other major currencies.

One of the most common types of a Foreign Exchange market from traded instruments perspective is a spot market that assumes that two parties immediately agree on an exchange rate between two currencies and then exchange money (Rosenstreich 2005: 76). A spot trade consists of two simultaneous transactions: a buying of one currency and a selling of another. In addition to spot exchange market, currencies are also traded on the currencies derivatives market in the form of forwards, currency futures, currency options, currency swaps etc. (Wang 2010: 14-15).

On the spot market, a market maker provides its rates to a trader. Traders can bet for both rate growth and decrease. A buy-position (also called a **long position**) is opened to get a profit from the rate growth and a sell-position (also called a **short position**) is opened to get a profit from the rate going down. Some trading platforms provide functionality to automatically close trading positions when the rate reaches some pre-configured levels. The most commonly used position closing levels are **stop loss** which is used to automatically exit from a potentially losing position when rate started to move in an undesirable direction and **take profit** which is used to automatically close a winning position keeping the profit.

The Foreign Exchange market represents the biggest asset (money) and this leads to its high liquidity. It operates 24 hours a day except weekends (Allen 2009: 152-153). When traded via electronic trading platform, usually orders on the market are performed almost instantly. A **slippage** is a difference between estimated transaction costs and the amount actually paid. A slippage takes place, because there is a time lag between the moment when a position opening/closing order is sent and the moment when the order completes; and the rate can change during that time.

Foreign exchange rates on the Foreign Exchange market are quoted as a number of units of one currency per unit of another currency. As trades frequency in the Foreign Exchange market is very high, the exchange rate changes all the time and quotations can be provided for even a very short period such as one second. A sequence of the rate values per period forms a rates time series. The rate series represents currency rates movement in time. Usually, a broker or a dealer provides the quotes feed that includes the following values for every time period:

- **Open** – starting rate of the period (the rate value at the period's start moment);
- **High** – maximum rate of the period;
- **Low** – minimum rate of the period;
- **Close** – close (ending) rate of the period;
- Date and time of the period start;
- **Volume** – number of trades completed during the period.

Visually the Open-High-Low-Close (**OHLC**) rates series are usually presented as on the figure below using so called OHLC bars. One vertical bar represents one period.

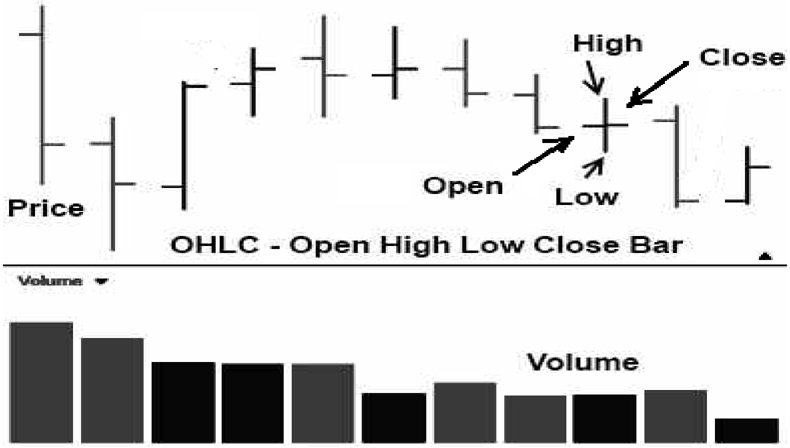


Figure 1. A typical graphical presentaion of OHLC rates used in charts. Source: (Stock Charts …).

The most of currencies are usually rated to four decimal places. This smallest unit of rate change is called a **pip** (percentage in point). As mentioned above, brokers take commissions for their services. The commissions are usually not paid upfront to the brokers, but are "hidden" into a spread (Rosenstreich 2005: 59). A **spread** is a difference between the bid and the asked rate (e.g. a few pips).

As the Foreign Exchange market is the most liquid and volatile financial market (Shamah 2003: 23) and quotes are available for even very short periods, this makes the market a suitable candidate for triangle chart pattern analysis, especially because this research needs a massive number of real triangle chart pattern examples.

1.2. Usage of triangle chart patterns for rate movement direction prediction

Triangle chart patterns offered by technical analysis methods are used in financial markets for price/rate change direction prediction and to produce corresponding position opening signals. Particularly, triangle chart patterns are used for trading in Foreign Exchange market also (Cheng 2007: 179; Bickford 2007; Person 2007: 135). There are many books and articles (including referenced right above) dedicated to Foreign Exchange that also shortly explain using triangle chart patterns specifically for that market. However, the most comprehensive overview of triangle chart patterns and observations related to triangles that are not specific to Foreign Exchange markets are done by Bulkowski, T. (2002; 2005a; 2005b).

The triangle chart pattern hypothesis is based on a certain behavior of traders that explains the reasons of the triangle pattern formation which according to the assumption serves as a trigger for the powerful and high momentum rate move in a certain direction (Schlossberg 2006: 128; Cheng 2007: 179). Triangle formation signals that the trend has gotten ahead of itself and needs to consolidate for a while (Murphy 2009: 65-66). Consolidation means that the trend stops and that the rate starts to fluctuate in a certain range where each next swing is smaller than the previous swing (Bhandari 2012:34).

Visually, a triangle is formed on a rate time series chart by two trend lines that converge toward to each other: a resistance and a support trend line. A **resistance line** limits the rate changes from the above and a **support line** limits the rate changes from the below. It can be said that the rate changes are "bouncing" between the resistance and the support lines. The moments when a rate bounces of a trend line are called **bouncing points**. The resistance line and the support line converge at a point called **apex** (Kirkpatrick, Dahlquist 2010: 314). According to the guideline, a triangle chart pattern is confirmed and can be used for rate movement direction prediction when the rate breaks out of the triangle formation (Classic patterns: 4). The period when the rate has broken out of a triangle is called a **breakout point**. Triangles can have many false breakouts (Kirkpatrick, Dahlquist 2010: 319) which can mean that the rate can return back into the triangle pattern and then can break out again also in the opposite direction.

The figure below depicts an example of a triangle chart pattern and its important attributes.

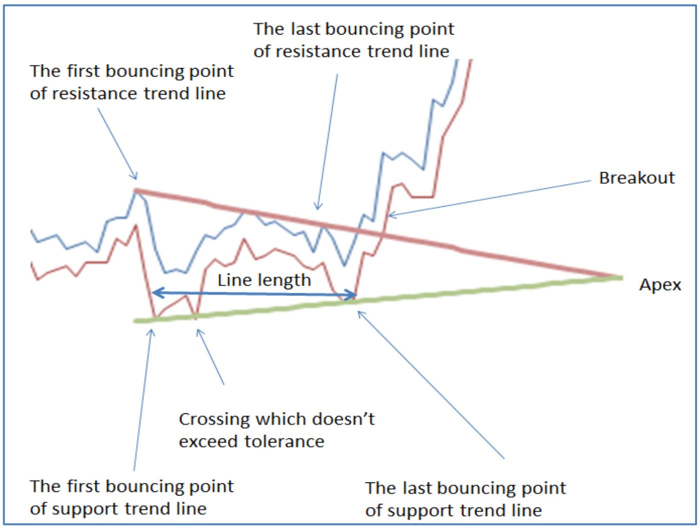


Figure 2. Main attributes of a triangle[1] (created by author).

Throwbacks/pullbacks are minor temporary rate declines in a direction opposite to the predicted rate movement direction in which case the rate can return to the breakout level. A pullback occurs after a downward breakout and throwback after the upward breakout. When a rate breaks out of a pattern, throwbacks/pullbacks are quite common and have to be taken into the consideration (Bulkowski 2005a; Kirkpatrick, Dahlquist 2010: 318). The figure below shows an example of a throwback.

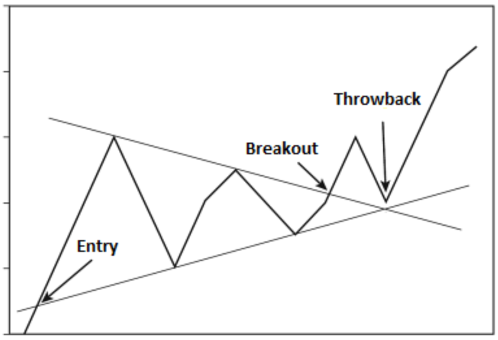


Figure 3 A throwback example. Source: (Kirkpatrick, Dahlquist 2010: 318).

There are three types of triangle patterns as per the description below:

Ascending triangle (see Figure 4) – in an ascending triangle, the upper resistance trend line is flat (horizontal) and the support trend line slopes upward. According to the

[1] Higher rates series are shown in blue and lower rates series are shown in brown on the figure.

hypothesis, the guideline is that an ascending triangle generally predicts that the price will continue to move upward after breaking out of the triangle. The pattern with increasingly higher lows and constant highs indicates that buyers are more aggressive than sellers. The pattern appears because a supply of the instrument is available at a fixed price (Classic patterns: 4). Every time when the currency rate goes up to the certain level (resistance line), there are sellers which hold up the level with the fixed asking, thus pushing the rate down each time when the particular level is tested (Cheng 2007: 182). Buyers, however, are becoming more aggressive to buy as they feel that the rate must go up over time. Thus, when the rate bounces of the resistance line, buyers take their opportunity to buy again with each offer higher and higher than the previous one. This is reflected in the support line sloping up (Cheng 2007: 182). When the supply depletes, the prices quickly break out from the top resistance trend line and move higher (Classic patterns: 4). Regardless of the theory, obviously, a breakout in ascending triangles can happen in both directions, and Bulkowski (2002: 307) advises to trade ascending triangles when its breakout direction agrees with the triangle's inbound trend.

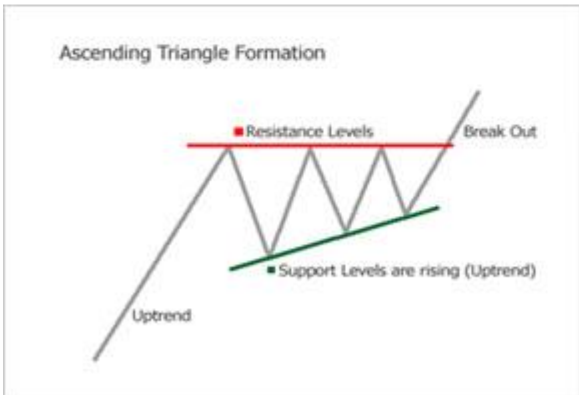


Figure 4 Ascending triangle[2]. Source: (Technical analysis).

Descending triangle (see Figure 5) - in a descending triangle, the bottom support trend line is flat (horizontal) and the resistance trend line slopes downward. The pattern is a kind of opposite of ascending triangle; its increasingly lower highs and constant lows indicate that sellers are more aggressive than buyers (Classic patterns: 4). According to

[2] The curve on the diagram and the diagrams below shows the rate change in time. Horizontal red line is a resistance trend line and the green line is a support trend line.

the hypothesis, the guideline is that a descending triangle generally predicts that the price will continue to move downward after breaking out of the triangle. The typical definition however contradicts to an opinion of Bulkowski (2002: 323) that says that descending triangles performance is good if to follow his recommendation to trade descending triangles in the direction of the breakout that can be both upward and downward.

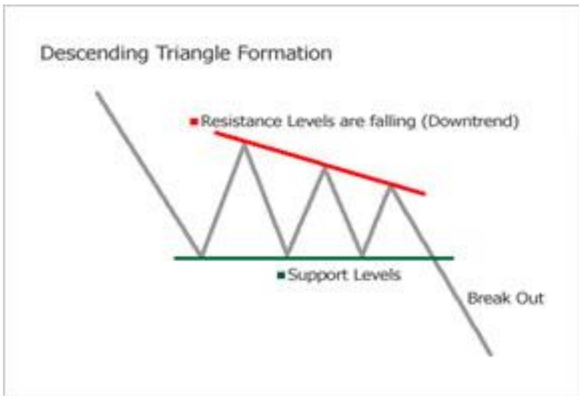


Figure 5 Descending triangle. Source: (Technical analysis).

Symmetrical triangle (see Figure 6) – in this case, the resistance line slopes downward and the support line upward. The guideline regarding symmetrical triangles is that the price after the breakout will continue its movement in the same direction according to the trend that existed before entering the triangle (Classic patterns: 31). However, symmetrical triangles also can serve as a trend reversal pattern, they have an identical look as trend continuation triangles and that's why it is important to use other parameters in order to predict the correct price movement direction, such as fundamental parameters, technical indicators, but especially the breakout direction (Schabacker 2005: 181). Anyway, an important parameter in determining rate movement direction after the breakout is the direction of the inbound trend, it is recommended to consider inbound trends which are longer than the triangle pattern formation itself (Classic patterns: 9).

The pattern appears, because the trading action gets tighter, but buyers and sellers are not sure whether the trend will continue. The uncertainty is expressed in their actions of buying and selling sooner; and that makes the range of price movements increasingly tight (Classic patterns: 32). The uncertainty can be also expressed in the decrease of volume as the pattern develops toward its apex (Bhandari 2012: 34). During the consolidation, neither buyers nor sellers extend their ranges (Schlossberg 2006: 128). When a consensus is reached, the price breaks out of the triangle and starts to move in a certain direction. The spike of volume at the breakout confirms the strengths of the consensus.

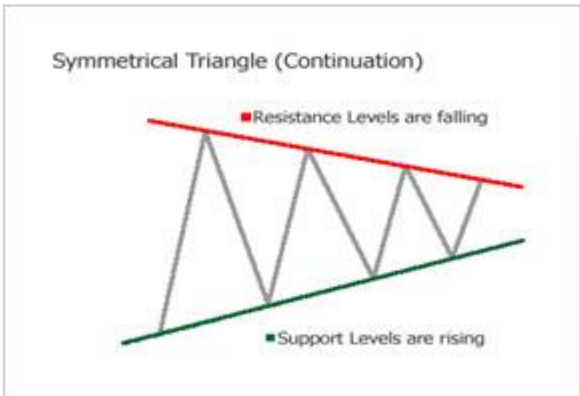


Figure 6 Symmetrical triangle. Source: (Technical analysis).

As opposed to many other authors, Bulkowski describes some additional triangle pattern criteria, for example how to distinct a good triangle from a wrong triangle. According to him, prices must cross a triangle chart pattern from side to side bouncing and reversing several times (minimum four reversals) and leave little whitespace within the body of the pattern (Bulkowski 2005b). Such definition however contradicts with triangle understanding of some other authors, e.g. with a symmetrical triangle example by Schlossberg (2006: 129) that has a big whitespace area as per the figure below:

Figure 7 A symmetrical triangle example. Source: (Schlossberg 2006: 129).

There is another pattern that is similar to a triangle (called a pennant) which can be considered as a pause in a trend (Kamich 2009: 121). Pennant looks like a triangle, but is usually short-term. In that case, the price has moved in a certain direction, but because of decreased activity, it takes a break before continuing the trend. The pause is usually reflected in the decreased trading volume. When the pause is over, the trend should continue (Classic patterns: 30-32).

As can be seen, definitions of triangle chart patterns sometimes are quite ambiguous and also there can be additional and sometimes contradictory opinions coming from different authors. For example, it is said that a pennant is usually shorter than a triangle, but obviously the term “shorter” depends a lot on a considered time horizon. Another problem is that one of the most important parameters emphasized by triangle chart pattern theorists is a volume of trades, but unfortunately the volume value in the currency rates series provided by a certain market maker reflects just the volume of this particular market maker (e.g. bank’s own clients). And consequently the local volume doesn’t reflect the overall volume of the Foreign Exchange market (Cheng 2007: 54). Some authors, e.g. Schabacker (2005: 181) recommend the usage of fundamental indicators that of course are applicable in general in financial markets for triangle chart pattern based trading, but are not very applicable in case of intra-day Foreign Exchange trading when the periods are very short, because the news are not frequent enough to combine them with relatively short-term triangle formations and their arriving and processing takes time. So, it can be summarized that the triangle chart patterns based trading principles are not defined precisely enough to apply them for intra-day trading in Foreign Exchange and basically the only data that is surely available to support triangles based trading in intra-day Foreign Exchange Markets are the rate changes themselves.

This means that an additional study is still needed to understand which parameters are really important and which are less important for predicting of a rate change direction when the rate breaks out of a triangle. And the work will evaluate applicability of neural networks to find the relationships between the parameters and the resulting rate change direction.

1.3. Fundamentals of artificial neural networks

1.3.1. Neural networks concepts and application domain

Artificial neural networks are used to solve function approximation, classification and other problems. Financial forecasting tasks can also be defined as function approximation or classification problems. That's why neural networks find their active application in financial forecasting (Gurney 2009: 5).

There are many other widely used methods of function approximation such as the linear regression analysis, splines and others. However, the methods are not as universal as neural networks, because approximated functions assume that there is a certain type of dependency between the input and the target parameter (Bhadeshia 1999: 967) (for example linear in case linear regression analysis, polynomial for splines). An artificial neural network based analysis is not as restricted as the methods where the form of the function has to be specified before the analysis. Artificial neural networks can capture any relations between the parameters and the greatest advantage of artificial neural networks is their ability to be used as an arbitrary function approximation mechanism (Beale *et al* 2013: 1-14) that learns from observed data. For example, a neural network can take observations from currency rates historical data in order to attempt to predict future rates.

An artificial neural network is a computational structure whose creation was inspired by investigation of biological processes of the human brain and which is built based on the similar principles as human brain's nerve net or a network of neurons. The human brain gets information through receptors, analyzes the data in the nerve net, makes decision and learns by getting feedback to the decisions (Gurney 2009: 2). Learning means that a neural network adapts itself based on a positive and a negative feedback that it has received in the past experience. Neurons in the human brain are interconnected between each other into a network using connections (synapses) that transmit signals from one neuron to another. Thus, neurons and their synapses form a biological neural network.

An artificial neural network is a massively parallel distributed processor that is made up of many simple processing units (neurons) which are able to memorize experiential knowledge and make it available for use. The knowledge is acquired through a so called learning process that adjusts neurons' state through getting a feedback. Thanks to its massively parallel distributed structure, a neural network has an ability to learn and consequently to generalize. **Generalization** means that a network is able to produce reasonable outputs for inputs that it has never encountered during its training (Haykin 1999: 24).

Neural networks are utilized for speech recognition, textual characters recognition, medical diagnosis, financial forecasting etc. (Gurney 2009: 5). The most of the tasks solved using neural networks fall into two categories:

- **Function approximation** (or regression analysis). The goal of the task is to find a function that would closely match ("approximate") a target function. For example, in order to check if the next currency rate change somehow depends on rate changes in previous periods, it can be said that the next rate change is the value of a function which takes previous rate changes as input parameters. And then a neural network can be trained in order to attempt to approximate such function.

- **Classification** (or pattern recognition). Classification addresses a problem of identifying a corresponding category where the given observation belongs to. Assuming that there are regularities in currency rates historical data, it is possible to build a neural network which has to recognize one of three currency rate change classes (rate goes up, rate goes down or rate doesn't change) for the next period based on data from previous periods.

A classical example of a neural network usage is hand-written characters (such as "a" and "b") recognition. A scanned image with a hand-written character is represented as an array of pixels on a computer. The same character can be written by hand in very many different ways (for example by different people) and this means that there can be a huge number of different valid images which represent the same character. Creation of an algorithm that would use some logical rules to analyze the image and recognize the characters is extremely complex and may not give the good results. Instead, the character recognition problem is successfully solved using neural networks which take pixels array as an input and produce an output which identifies the character (Bishop 1996: 3-5). As mentioned above, the problem of currency rate change direction prediction can be also defined as a pattern recognition task. This is why the methodology was chosen for the given research.

In the similar way to biological neurons, an **artificial neuron** (or a "formal neuron") is a fundamental unit of an artificial neural network. Every neuron receives information, processes it and produces output according to the following (Haykin 1999: 32-33):

1. Each incoming synapse's importance is characterized by its **weight**. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by synaptic weight w_{kj}. Weights can have positive and negative values.

2. A neuron sums all its weighed incoming signals and adds a bias b_k to the sum. The **bias** is used to lower or increase the aggregated input. The total sum is also called a **net input**.

3. Then the neuron applies a so called **activation function** that converts the net input into output usually by limiting its amplitude in order to "squash" the output into a certain range. Typically, (depending on the neuron's activation function) the output falls into the range of [0, 1] or [-1, 1] (Kamruzzaman 2006: 4). The most typical activation functions are listed in the chapter 1.3.2.

The following figure describes the neuron's model:

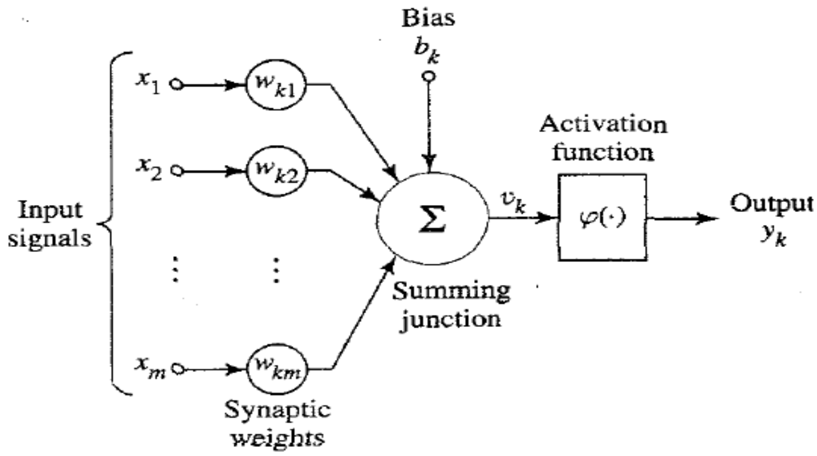


Figure 8. Neuron's model. Source: (Haykin 1999: 32-33).

Mathematically, a neuron's **output** y_k can be described as:

(1) $$y_k = \varphi_k(v_k), \quad v_k = \sum_{j=1}^{m} w_{kj}x_j + b_k$$

where $x_1, x_2, \ldots, x_m$ are input signals;

$w_{k1}, w_{k1}, \ldots, w_{km}$ are their corresponding weights in neuron *k*;

b_k is the bias of neuron *k*;

v_k is the net input of neuron *k*;

φ_k is the activation function of the neuron *k*;

y_k is the output of neuron *k*.

An artificial neural network is a set of arbitrarily interconnected neurons. The most frequently used type of artificial neural networks is a multilayer perceptron. A **multilayer perceptron** consists of several layers of neurons: one input layer, one or more hidden layers and one output layer (see example on Figure 9). Each layer can contain any number of neurons, every input represents an independent variable and every output neuron is a dependent variable (Kamruzzaman 2006: 5). In a multilayer perceptron, the signals are transmitted within the network in one direction: from the input neurons through the hidden layers to the output neurons, basically mapping a set of input data to a set of output data. There is no loop and the output of each neuron does not affect the neuron itself. Such architecture without loops is called a **feed-forward network** (Gallant 1993: 11).

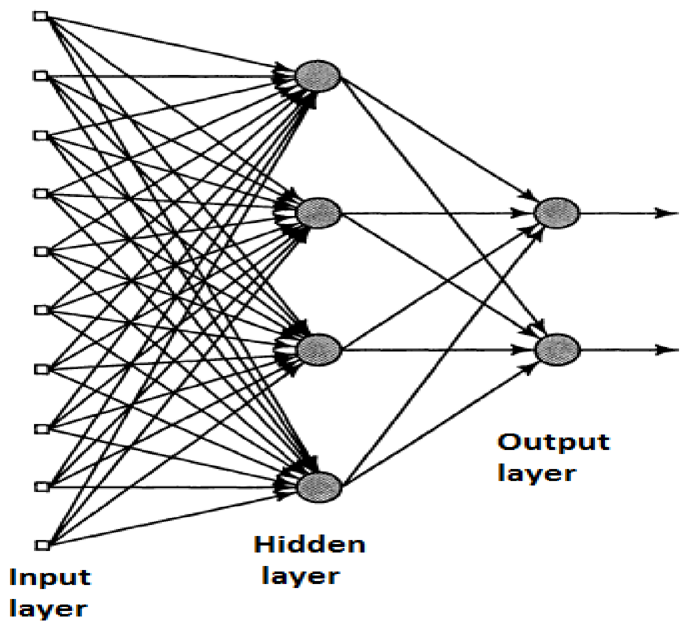


Figure 9. A feed-forward neural network example. Source: (Haykin 1999: 44).

In the architecture, each neuron of one layer is connected to each neuron of the next layer. This means that each neuron's output serves as an input for the next layer's neurons. Input parameter values in the multilayer perceptron are propagated to the input layer neurons (each neuron gets one value). The role of an input layer is just to transmit input data to the next layer; no processing of the input values is done in the layer (Gallant 1993: 11). All neuron's except for the input layer calculate a transformation of all their inputs and pass their outputs to each neuron of the next layer. Neurons in a hidden layer fulfill a role on middlemen between the previous layer and the next layer. Output layer neurons make processing of all their inputs and produce the network's outputs: one output neuron produces one network's output value.

Universal approximation theorem states that an arbitrary continuous function can be approximated in any precision by only one hidden layer feed-forward neural network (perceptron with two[3] layers: hidden and output) with a finite number of neurons (Cybenko 1989: 303-314).

1.3.2. Overview of neuron's activation functions

Creation of a neural network's architecture involves choosing activation functions for the neurons. Different activation functions can be used to serve different purposes. This chapter describes and compares the most commonly used activation functions such as linear, logistic sigmoid, hyperbolic tangent sigmoid and SOFTMAX:

Logistic sigmoid activation function (Filimonov 2004: 8-9):

(2) $$y_k = \frac{1}{1+e^{-v_k}}$$

where *e* is Euler's number;

y_k is the output of the neuron *k*;

v_k is the net input of the neuron *k*.

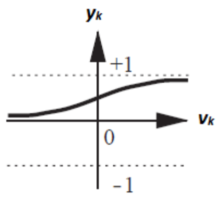


Figure 10. Logistic sigmoid function. Source: (Hagan *et al* 1996: 2-4, 2-5).

Utilization: multilayer perceptron with continuous signals.

Benefits: Logistic sigmoid is a continuous function and its first derivative function is also continuous that makes possible usage of gradient learning methods like error's back-propagation (Filimonov 2004: 8-9).

[3] Input layer is usually not counted into the number of perceptron's layers.

Disadvantages: function's range is not symmetrical in relation to 0 and this makes the learning slow (Filimonov 2004: 8-9).

Hyperbolic tangent sigmoid activation function (Filimonov 2004: 8-9):

(3) $$y_k = th(v_k) = \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}$$

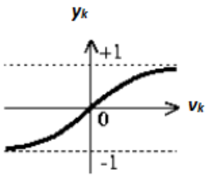


Figure 11. Hyperbolic tangent sigmoid function (created by author).

Utilization: multilayer perceptron with continuous signals (Filimonov 2004: 8-9).

Benefits: Hyperbolic tangent sigmoid is a continuous function and symmetrical relatively to point (0, 0). The activation function has proven better performing than logistics sigmoid and other continuous and differentiable activation functions (Karlik, Olgac 2010: 121).

Linear activation function (Filimonov 2004: 8-9):

(4) $$y_k = v_k$$

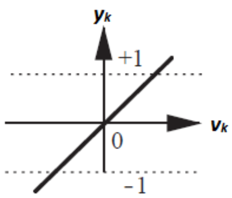


Figure 12. Linear function. Source: (Hagan *et al* 1996: 2-4, 2-5).

Utilization: to find a linear approximation of non-linear functions (Filimonov 2004: 8-9).

Benefits: fast and have only one minimum in error surface, because error surface is a multidimensional parabola (Filimonov 2004: 8-9).

Disadvantage: A linear network cannot perform a nonlinear computation (Filimonov 2004: 8-9).

SOFTMAX activation function (Filimonov 2004: 8-9)**:**

(5) $$y_k = \frac{e^{v_k}}{\sum_i e^{v_i}}$$

where y_k is the output of the neuron k;

v_k is the net input of the neuron k;

v_i is the net input of the neuron i (from the layer of the neuron k);

Utilization: classification tasks where expected output are probabilities. SOFTMAX function makes sure that sum of all outputs always equals to 1 and every output neuron's value is a probability of that the sample belongs to the given class. Neural network with such behavior are called **classification neural networks**. Classification neural network works based on the principle: a winner gets everything (the neuron with higher probability value in the output is decided to be the right class for the data sample).

The negative side of classification neural networks for Foreign Exchange rate movement direction prediction is that such neural networks answer will always fall into few strict classes (e.g. rate grows, rate decreases and rate doesn't decrease). Of course, the strict classes may be even good if they can be mapped directly to trading signals (for example a rate growth always means opening a long position), but applying of the classes is sometimes problematic if there is a need for a more sophisticated approach. For example, such neural network will never answer a question, how quick will be the growth (function approximation neural networks with sigmoid activation functions may be more suitable to get such answer). But the important positive side of a classification network and SOFTMAX activation function is that it returns a probability of that the observation belongs to the given class. This can be important if a trader wants to open positions only when the neural network is quite sure regarding the rate movement direction.

As described in the second part of the paper, neural network models created in this work, used hyperbolic, linear and SOFTMAX activation functions.

1.3.3. Neural network training and results analysis related concepts

Neural network training involves a systematic and sequential feeding of training samples to a neural network input, comparing of the produced output with the desired output and the adapting of the neural network to better match the input samples to the desired output samples. Such learning when the desired output is known is also called **supervised learning** (Mehrotra *et al* 1996: 43).

When a training starts, neuron's inputs are multiplied by random weights, the products are summed together with a constant bias and then the activation function is applied (Han, Kamber 2000: 305). This gives some output. Since the weights and the constant bias were chosen randomly, the value of the output will not match with the desired target data. Then, by processing training samples, the weights are systematically changed until a best-fit description of the output is obtained as a function of the inputs.

When a neural network calculates output for the given training sample then the desired output values are used to calculate the difference (delta) between the desired output and the resulting output. And then the weights of the neurons are corrected according to the delta. Such method of supervised learning is called a **back-propagation method** (Han, Kamber 2000: 305).

For the sake of simplicity of the explanation of the error back-propagation principle, let's assume there is only one output neuron k. And let's assume that a neural network has provided output $y_k(n)$ for the sample n and the desired output was $d_k(n)$. Then the delta (error signal) of the output is calculated as (Haykin 1999: 73-74):

(6) $$e_k(n) = d_k(n) - y_k(n)$$

If $w_{jk}(n)$ is the weight of the connection j to the neuron k in the moment of passing of the sample n, the weight adjustment for the weight $w_{jk}(n)$ is calculated using the following formula (Haykin 1999: 73-74):

(7) $$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$

where η is some positive constant that is called a **learning speed** (the greater the learning speed, the faster the neuron trains; the lower the speed, the more accurate the training is). It is clear that the bigger error will cause bigger adaptations done to the neuron connections weights (Haykin 1999: 73-74).

The goal of the training is to minimize the deltas for every output neuron and for every sample used in the training (the set of sample is called a **training set**). As there can be several output neurons, the goal of the training is actually not to minimize a delta of one particular neuron, but to minimize some value that measures an error for all output neurons. Such value is usually calculated using a so called **error function**. The typical error function (also called a **performance function**) and the most widely used for classification tasks is **Mean Squared Error (MSE)** (Gallant 1993: 123):

(8) $$MSE(n) = \frac{1}{m} \sum_i^m e_k^2(n)$$

where *m* is the number of output neurons.

During the training the error value constantly decreases until it reaches some steady state (a minimum) and then the training can be stopped.

The main goal of the training is to decrease the value of the error for the training data set. However, the minimization of the error only for the training set doesn’t mean that the trained network will work well also for other sets (out-of-sample sets). To ensure that, training methods use an additional **validation set** of samples in order to make sure that the error decreases also for that set. If the validation error doesn’t decrease then the error optimization for the training set doesn’t make sense (network **over-fits** the training data) and the training can be stopped. There is one more additional set of samples called a **test set** that is not used during training at all, but is used after the training to compare the errors for test set with errors for training and validation sets.

Due to non-linearity of activation functions of neurons, there can be more than one minimum in the error surface, because the surface can have numerous 'pits' and 'hills'. So, if training means following down a local slope of the error surface then the training can end up in reaching a **local minimum** that is not the best overall solution and there could be another the best minimum (**global minimum**) in case of which the error value would be really minimal (Gupta 2003: 162). The local minimum problem can be visualized using the following figure:

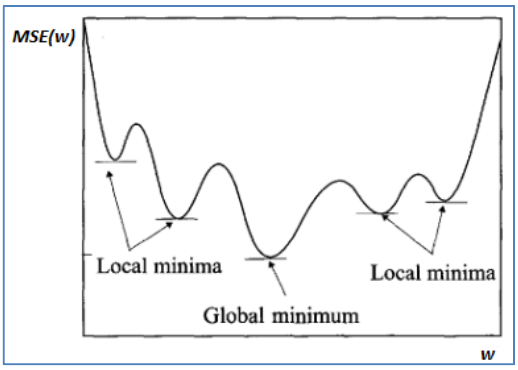


Figure 13 Global minimum and local minimums, one-dimentional example. Source: (Gupta 2003: 162).

The chapter described shortly a basic back-propagation algorithm, but there are more sophisticated back-propagation training algorithms (such as Levenberg-Marquardt, Resilent Propagation, BFGS Quasi-Newton and Scaled Conjugate Gradient) which implement optimizations in order to make training faster and better cope with the local minimum problem. Such methods are used in the work, because they enable faster training and also different training methods can give different results. Describing these training methods requires much more details, so it was decided to leave their description out of scope of the given paper.

This work used MATLAB software product and its Neural Network Toolbox for artificial neural networks creation and training. MATLAB provides functionality for defining and training of feed-forward neural networks, it provides user-friendly interface which is helpful in training results analysis and is sufficient tool to reach the goal of the research.

When training completes, the achieved result can be seen and evaluated using a performance plot and the confusion plot which are available in MATLAB software (see examples on Figure 14 and Figure 15). The performance plot shows how much the training error value (e.g. MSE) has decreased during the training for three given sets: a training, a validation and a test set.

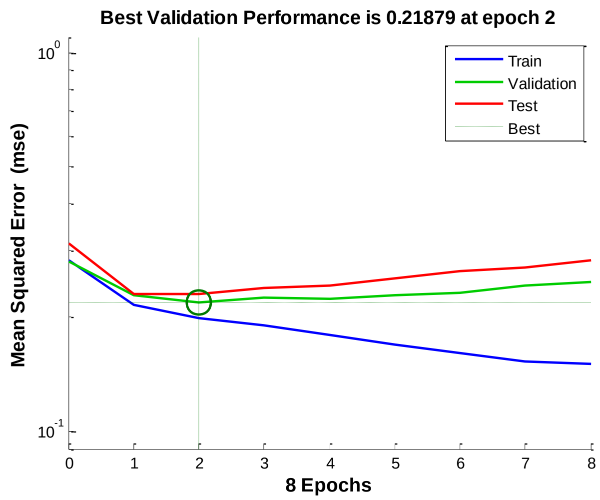


Figure 14. Training performance plot example, MATLAB (created by author).

Continuing a training if the error decreases only for the training set, but not for validation set doesn't make sense and the training has to be stopped, because otherwise the neural network will just over-fit the training set instead of generalizing. An over-fit network will have a bad predictive performance, because it has remembered the noise and the specifics details of the training set instead of the underlying relationship between the inputs and the desired outputs. Also it is possible that a neural network performed well for the training set and the validation set, but not for the test set. This situation is also caused by over-fitting. For example, it can be seen on the Figure 14 above that the test curve had increased significantly before the validation curve increased, then it is possible that some over-fitting might have occurred. There are several ways to avoid over-fitting:

1. Providing more samples data. This can help in a situation when a neural network had a big number of neurons which was sufficient to start memorizing training sample details. Number of triangle samples shouldn't be a big problem for currency rate series with a short period (seconds or a few minutes), as there can be many triangle chart pattern examples in the rate series. But this cannot be said regarding currency rate series with a longer period (several hours or days).

2. An alternative approach is to make a neural network simpler (decrease the number of hidden neurons). The larger network is used, the more complex functions it can create. The smaller network will not be able to over-fit the data (Beale *et al* 2013: 8-34). Such approach can be tried when training a neural network for currency rate series with a longer period (several hours or days).

3. Another way is the early stopping that is implemented out of the box by MATLAB. The training stops automatically if decreasing of training error will not decrease validation error a certain number of times in a row.

Another important plot that describes training results of a classification neural network is a confusion plot (see example of the figure below).

Figure 15. A confusion plot example, MATLAB (created by author).

The confusion plot shows how many samples were classified correctly and how many incorrectly into each class for each of the sample data sets. Each data set (training, validation and test) is presented by a separate matrix and there is one more matrix that describes totals. The matrix maps expected results into results produced by the neural network's output. Green cell show a number of samples that were related to the correct class. The important characteristics shown in the plot are percentages of correct answers which describes neural network model’s prediction accuracy. And the most important criteria which assess the model are prediction accuracy percentages for the test set, because the test set wasn’t used in the neural network training, and that’s why the numbers are reliable. Accuracy percentages for training and validation sets are usually higher, because training and validation sets were used in the training and there might be some over-fitting. Prediction accuracy of neural network models created in this work will be assessed using confusion plots and based on results for test sets.

1.4. Overview of other researches

There is a wide variety of publications regarding the use of triangle chart patterns for the prediction of currency rates or shares prices. Also many researches were done regarding usage of neural networks in currency rates prediction. However, there was no scientific works found that would combine the chart patterns with the neural networks.

Kirkpatrick, C. and Dahlquist, J. in their book claim that validity of chart patterns is still questionable, because the patterns are not proven mathematically. And they also think that methods such as neural networks may prove useful in chart patterns validity demonstration, but only sometime in the future (Kirkpatrick, Dahlquist 2010: 303). This statement gives an additional motivation for the given investigation.

Bhandari, B. (2013: 36) in his article opines that the triangle chart patterns are ones of the most reliable and simple patterns, describes different types of the triangles and reaches to the conclusion that still the triangles are not perfect and that’s why triangle breakouts have to be confirmed using volume and other indicators as moving average and stochastic oscillators etc.; and also the opened position have to be protected by a stop-loss (Bhandari 2013: 36). This clearly means that the fact of a triangle chart patterns formation by itself is insufficient for a reliable rate movement predication and involving of other additional parameters is required for making a right prediction. However, the author doesn’t suggest a clear algorithm that would cover all required parameters and conditions. And this is where neural networks can be introduced which can help in identifying relationships between a triangle pattern’s parameters and the future rate movement.

Vyklyuk, Y., Vukovic, D. and Jovanovic, A. (2013: 261-273) in their article describe creation of a multilayer perceptron with two hidden layers and reach to a conclusion that an exchange rate of EUR-USD for the next period depends linearly on the previous rates and can be modeled using a neural network which reaches prediction accuracy $R > 0.8$ i.e. correlation between neural networks output and desired output is more than 0.8 even without using chart patterns or any other indicators (Vyklyuk *et al* 2013: 261-273). A big problem with such conclusion can be seen, because authors use absolute values of currency exchange rates in the input and in the output of a neural network. Non-applicability of such approach is explained in details in chapter 2.4. This means that the conclusions reached by the article's authors stay very questionable.

In a similar way, Dunis, C. and Williams, M. (2002: 19-20) have built a neural network for next day's EUR-USD rate change forecasting, but using a non-typical approach for neural network's performance measurements, measuring it financially and not statistically. This means that instead of using a usual statistical mean square error as a performance function, a total profit function was used; and the goal of training was to maximize the received profit. They built a neural network that uses current rate changes and fundamental indicators as the input. Their neural network regression models reached a 57.2% accuracy of winning trades prediction. Nevertheless, their overall results confirm the credibility and potential of neural networks usage in currency rates forecasting. However, as the one of the major disadvantages that they see is the inability of a neural network to explain its reasoning (Dunis, Williams 2002: 19-20). It has to be said that the accuracy achieved by the authors looks quite high, but unfortunately their results cannot be reproduces as part of the given research because of the lack of details in description of parameters used by the authors to create such a neural network model. Also it cannot be seen from the article if applying their model for prediction of several periods ahead (as in case of triangle chart patterns) could give the same good results. Anyway, the results described in the article demonstrate that neural networks are a proper choice for currency rates behavior prediction.

Zhang, G. and Hu, M. (1997: 495-506) analyzed foreign exchange rates forecasting using neural networks and emphasized that an appropriate selection of neural network inputs and a neural network architecture is critical for the predictive accuracy of the rates. They investigate dependency between the number of neurons (input and hidden) and performance of the corresponding models. And as the input nodes represented the rate series of the several previous periods in their model, the authors reached to a conclusion that the number of input nodes is determining for the prediction accuracy of the model, because it determines the underlying autocorrelation structure. Also authors admit that their investigation of a one-step-ahead forecasting doesn't mean that the prediction results for several periods ahead will be same good (Zhang, Hu 1997: 504-505). Their work demonstrated that different neural network architectures can give different prediction accuracy and that's why the aspect also received a high attention in the given research and many combinations of architectures and inputs were tried. However, it has to be said that their investigated next period's rate prediction model differs from the goals of this work which will need to predict rate movement direction and consequently rate behavior for several periods ahead have to be predicted, not for one period.

Mechgoug, R. and Titaouine, N. (2012: 87-92) addressed a known issue that a typical problem with using a neural network for price forecasting is a large variety of possible input parameters that could be potentially used in the input for the model and numerous options of neural network architectures. The authors of the article have used genetic algorithms in order to optimize the parameters of a multilayer perceptron, the number of neurons in the hidden layers, the activation functions and other properties of a neural network. They have evaluated their approach in predicting the next rate in the series of different currency pairs' rates. The reached conclusion was that neuro-genetic model has a better performance than empirically constructed models (Mechgoug, Titaouine 2012: 92). Usage of genetic algorithms is out of scope of the given research, because incorporating also genetic algorithms would require additional efforts exceeding the size meant for this thesis, but it could be a right direction for future optimizations of the neural network model.

Chen, A. and Leung, T. (2005: 403-418) have evaluated performance of different neural network architectures for foreign exchange prediction. They found that neural networks created based on different architectures can capture useful relations not found in each other; and that the useful information sets captured by the networks are independent one of the other. This means that the prediction results can be improved if to combine several neural networks of different architectures (Chen, Leung 2005: 418). Such approach is called a **neural networks committee**; and their investigation confirms the theory that combining multiple neural networks output into a single response can give a superior result than the separate results of the committee members. Usage of neural networks committee is out of scope of the given research, but it could be a right direction for future optimizations of the neural network model.

2. INVESTIGATION: CURRENCY RATE MOVEMENT PREDICTION USING TRIANGLE CHART PATTERN AND NEURAL NETWORKS

2.1. Data and methodology

The final goal of making any analysis is taking decisions. The main decisions that a trader has to take are targeted to profit maximization and are related to opening and closing trading positions. The basis for taking such decisions in Foreign Exchange market trading are trader's assumptions regarding the future rate movement direction.

As was described in the chapter 1.2, triangle chart patterns are used to predict quick rate change movements (up or down). However, a visual analysis of rate charts of EUR-USD pair for example (as can be seen on Figure 19 for instance) clearly demonstrates that the triangle patterns don't work always; and it is very hard or impossible to find a good explanation when the regularity suggested by the triangle chart pattern hypotheses will work and when won't. Rates series and the charts have a lot of unpredictable, subjective, stochastic components that can be considered as a noise. It is well known that neural networks have very good dependencies capturing and noise filtering properties, i.e. are capable to generalize (as mentioned in chapter 1.3.1). This is the reason why they can be applied to such kind of analysis.

This work aims to design and build a currency rates movement prediction model using both triangle chart patterns and a neural network that uses the triangle chart patterns for finding relationships between the pattern itself and a rate movement after the breakout from the triangle. Consequently, the goal is to verify if the regularities in a rate movement after a triangle breakout can be captured by a neural network, i.e. if a neural network can provide good accuracy in answering the question: when a rate breaks out of a triangle, whether the rate will quickly go up, will quickly go down, or there will be no clear move is any of these two directions. The rate movement direction signal will be neural network's output. The Figure 16 below shows an example of three possible scenarios after the breakout point during the given short time interval: rate will either reach minimum upper level without going down during the move, or it will reach minimum lower level without going up during the move or none of these possibilities realize i.e. there will be no clear quick move in one of the directions. In case if a triangle pattern works, then the rate should move in the direction of the breakout. It is also possible that the breakout is a false breakout (see chapter 1.2) in which case the rate can return back to the triangle and later can break out and start moving quickly in the opposite direction (again assuming that the triangle works). Both predictions are valuable for a trader, because knowing a clear rate movement direction a trader can open a corresponding trading position. The rest of cases fall under a category of "no clear movement direction". Such cases are not predicted using triangles patterns, but will be predicted by created neural network based model. Receiving such prediction should signal about uncertainty regarding rate movement direction.

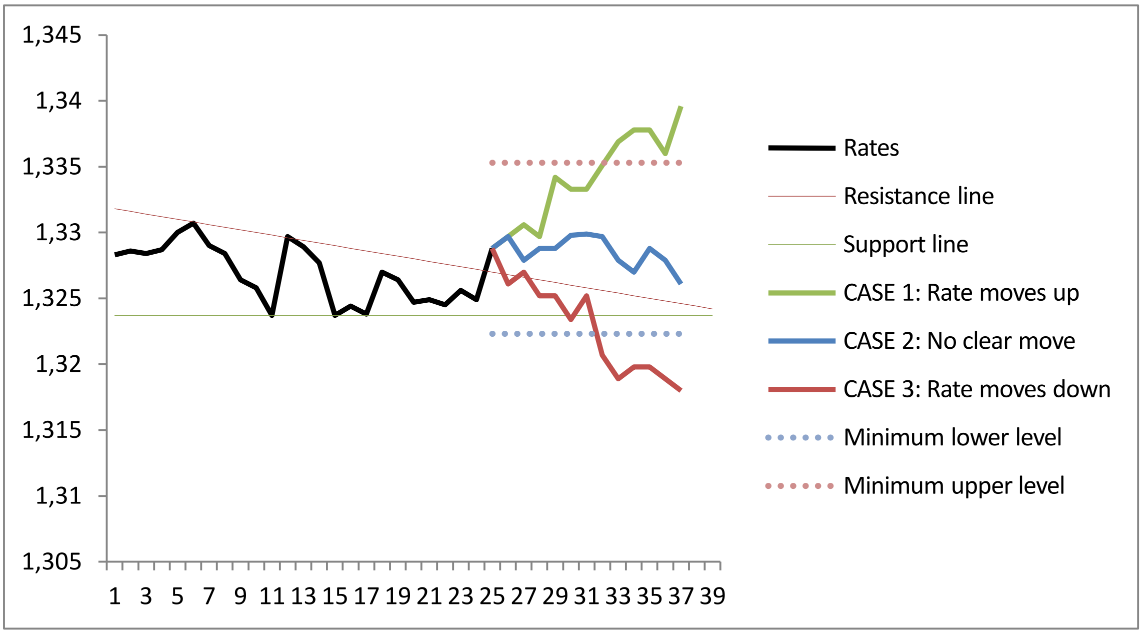


Figure 16 Possible currency rate movement scenarios after the triangle's breakout point (created by author).

Of course it is also possible that the rate can reach both levels during the given short time interval, but the most important is the level which was reached first. Knowing the level which will be reached by the rate first is important input for a trader and can be used to make trading decisions, for example opening a position at a breakout and closing the position once the level is reached etc.

When a rate breaks out of a pattern formation, throwbacks or pullbacks are possible. This means that the rate can temporarily return to the breakout level (as mentioned in chapter 1.2) or a bit further before continuing the movement. Consequently, the created prediction model must not assume that the rate will move only in a certain direction without the temporary reversals after a breakout, but take into account throwbacks/pullbacks which are natural for triangle chart patterns. But it is also possible that the declining in the direction isn't a throwback/pullback, but is a change in the rate movement direction. Two make difference between these two, a maximum acceptable throwback/pullback levels must be defined. If the created prediction model takes the levels into account then a trader can use the levels for example to exit from potentially losing positions if the predicted rate movement direction was incorrect.

Modeling an appropriate neural network for solving a certain task involves many steps and most of the steps can be done only by a trial and error method rather than referring to a precise architectural blueprint (Kirkpatrick, Dahlquist 2010: 603). Anyway, the process of modeling a neural network for rate movement prediction involves the following major steps (which are elaborated further in the paper):

1. **Definition of the goal of the modeled neural network, what data it uses and what output it produces.** As mentioned above, the goal is to build a neural network that would process a triangle pattern data and produce a signal regarding the rate movement after the breakout. This step also involves choosing the currency rates data and finding of triangle chart pattern samples in the rate series that could be used for neural network training.

2. **Choosing output parameters.** It must be decided how exactly the rate movement direction will be predicted using a neural network i.e. neural network's output parameters meaning has to be defined. It has to be decided

how the desired output values required for neural network training will be calculated.

3. **Choosing input parameters.** As a part of the step, it has to be analyzed what data can be available for the neural network input and what data can be helpful in determining the expected output. As every data sample will represent a triangle breakout period, the aim is to use a triangle chart pattern characteristics in order to predict the signal. This step involves analysis of possible triangle characteristics which could be suitable for neural network input.

4. **Preparation of training data.** This step involves preparation of input samples and desired output samples which can be used for the training. As a part of the step, the data has to be cleaned from invalid values, normalized if needed and split into three separate training sets: training data, verification data and test data.

5. **Choosing of a neural network architecture and creation of neural networks.** A neural network can be built in numerous different ways: it can have different number of layers and neurons; it can use different activation functions, can be trained using different learning methods and use different parameters to measure its performance, etc. The most appropriate architecture is usually determined empirically (Han, Kamber 2000: 303). One particular neural network's architecture can work well for one particular one task, but be absolutely not suitable for another. And there is no perfect architecture that would be appropriate for all scenarios. For this reason, the step involves experimenting with many architectures, subsequent training and assessment of their performance.

6. **Neural network training and analysis of results.** When a neural network training completes, the training results can be seen and analyzed. The most important question is whether the network has been trained successfully, has achieved sufficient prediction accuracy and can be used for trading.

In order to do the analysis and complete the listed steps of modeling a neural network, a certain currency pair has to be chosen. The most important requirements for such pair are liquidity and volatility, because the most frequently traded pair will mean more changes in the rates and consequently more triangle patterns that could be identified. Foreign Exchange offers a wide variety of currency pairs that are traded on the market. The traded currencies include EUR, USD, GBP, AUD, CAD, CHF, JPY and others. The most traded currency pair is EUR-USD (Triennial ... 2013: 11), so the pair was chosen for the investigation.

Analysis of chart patterns requires historical rate series data and there are many sources of the currency rates history available in the Internet. The rates history data is available for different time periods: 1 or several seconds, 1 minute, 5 minutes, 15 minutes, 30 minutes, 1 hour, 4 hours, 1 day etc. Each period in the rate series is described in the following values: Open, High, Low, Close, date/time and volume.

As a remark, the classical OHLC bars notation as referred in chapter 1.1 will not be used in this paper, because triangle chart patterns are built using only two rate value series (e.g. only using High and Lows series or only on highest and lowest of Open and Close series as per description in the chapter 2.2). The paper will use a line-based notation as on the example in the figure below. Vertical axis of the chart measures rates and horizontal axis shows a sequential number of a time period.

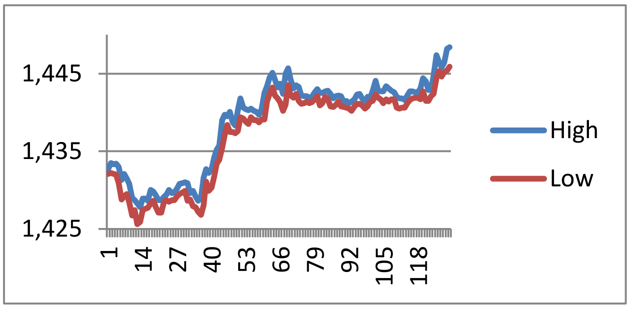


Figure 17. Highs and lows lines chart example (created by author).

As the currency rates behavior properties can change over the time (because of continuous appearing of new analysis methods, new technologies, new strategies etc.), it makes sense to use the most recent rate series for neural network training. But as neural network training requires many training samples (as mentioned in chapter 1.3.3), quite longer time intervals of EUR-USD rate series had to be used for this research, such as 01.01.2010 – 01.11.2013. Currency exchange rates used in this work were downloaded from FIBO Group's historical rates data page (History Center) in a CSV-file format.

It was decided not to consider rate periods of one or few seconds in the investigation, because applying triangle pattern for such a short period is not practical. The reason is that the formation of such very short-term triangle has a smaller chance to be caused only by decisions made by market participants (buyers and sellers). Rates of that short period are affected by delays in communication between different participants of the foreign exchange market which in turn cause delays in quotes delivery and orders completion. Also trading such triangles is problematic, because triangle's signal is a single moment (breakout) and order completion after a few seconds of delay will be too late already. Using daily or longer periods are also not suitable for the investigation, because there will be not enough triangle situations in order to train a neural network. So, only the following periods were considered: 5 minutes, 15 minutes, 30 minutes, 1 hour and 4 hours.

2.2. Finding triangles chart pattern samples

Investigated neural networks are supposed to produce rate change direction signals based on characteristics of triangle chart patterns. Training of such neural networks requires a huge number of triangle samples taken from historical data. This means that there was a need to automate triangle chart patterns finding in a currency exchange rates time series, because otherwise preparing of the training data for the neural networks manually would take ages.

In order to automate triangles finding, the author has created a software program as part of this work. The program was implemented in Java programming language. The algorithm goes through a rate time series analyzing every given moment and checking if the moment is a breakout point of some triangle.

When a neural network is trained and is used for trading, same algorithm could be used to automatically identify if the current rate is some triangle's breakout point and if yes, the neural network would have to take a decision regarding the coming rate change direction.

The principle of the algorithm is to attempt to build a triangle for the given time moment using two trend lines: a resistance line and a support line. A resistance line lays on top of the rate movements highs. And a support line lays below the rate movements lows (supports the rates). As rates data is represented by several values for every period (OHLC – Open/High/Low/Close) there are two ways to build the triangles that are used in trading practice and are implemented by the triangles searching software:

1. Triangle's resistance line is build based on High rates and a support line is built on the Low rates.
2. In the similar way the program also searches for triangles based on the following series: MAX(Open, Close) used as a higher rate and MIN(Open, Close) used as a lower rate. In this case, the program tolerates moments when High or Low rates are getting out of a triangle.

The two different approaches were tried because according to the chart patterns theory the patterns can be built based on different combinations of Open, High, Low and Close (Kirkpatrick, Dahlquist 2010: 301).

Identification of a set of triangles search criteria is not straightforward because of the fact that chart patterns definitions are often subjective (Anand, Wei-Ngan, Siau-Cheng 2001: 134) and different authors describe the patterns differently as shown in chapter 1.2. As a part of the work, the following criteria were chosen to find the triangles:

1. A support trend line must be ascending or horizontal. This criterion is based on ascending, descending and symmetrical triangle definitions which state that symmetrical triangles are formed by rising support line and a falling resistance line; descending triangle has a falling resistance line and a horizontal support line; ascending triangle is composed of a rising support line and a horizontal resistance line (Thomsett: 67-68). However, it is also allowed for a support trend line to be slightly descending which is regulated by a tolerance parameter which permits a trend line to be sloped down, but not more than a pre-defined number

of pips per period (tolerance 0.05 pips was used in this work). This is needed in order to avoid exclusion from the observation of the descending triangles which are slightly sloped down as on the figure below.

Figure 18 Descending triangle pattern example. Source: (Bhandari : 36).

2. In the same way, a resistance line must be descending or horizontal. However, it is also can be slightly ascending according to the tolerance parameter (0.05 pips was used in this work).

3. Support and resistance trend lines must converge in their near future. If a triangle's converging point (**apex**) distance from the breakout points was more than two times longer than the shortest trend line's length then such triangle was excluded from the consideration. Such criterion is chosen, because usually prices break out at about two thirds to three quarters of the way into the pattern (Murphy 2009: 66; Kamish 2009: 89). The condition is also important in order to distinct a triangle from a channel chart pattern which assumes parallel (or almost parallel) resistance and support trend lines.

4. A rate movement must bounce of each line at least two times (Bulkowski 2005b). This means that a trend line is basically formed by at least two bouncing points.

5. There must be at least one support line's bouncing point between resistance line's bouncing points. And vice versa. This is because prices must cross a triangle chart pattern from side to side and leave little whitespace within the body of pattern; and there must be minimum four reversals (Bulkowski 2005b). Bulkowski (2002: 347) sees cutting the rounding rate turns and calling it a triangle as a typical mistake.

6. The minimum and maximum accepted trend line lengths are defined using configuration parameters in the program (this importance of the parameters is covered below in this chapter).

7. The first bouncing point of a trend line must not be too far away from the first bouncing point of another trend line. The maximum accepted distance can be defined using a configuration parameter in the program. In this work, if the distance between the first bouncing points were longer that the shortest trend line length then such triangle wasn't taken into the consideration. The rule is important in order to avoid triangles with already mentioned large "whitespaces".

8. One trend line must not be more than two times shorter than the other trend line. This is also important in order to avoid the whitespaces, because the correct pattern should look filled with the rate; the rate shouldn't walk along one of trend line, but should cross the triangle from one side to the other (Bulkowski 2005a). If one trend line is much shorter than another then the rate crosses the pattern only during a certain small interval.

9. A trend line is not used for forming a triangle if the rate movement crosses (doesn't bounce of) the trend line, because this would contradict a triangle chart pattern definition that the price has to bounce of a trend line (see chapter 1.2). However, it was decided to introduce a tolerance parameter in pips that defines how far a rate can go after crossing the trend line before getting back into the triangle. The tolerance parameter can be configured in the program. The parameter is important in order to avoid exclusion of such triangles where price bounces of each trend line more than two times, but some bouncing points are very slightly out of the triangle's border. Such insignificant anomalies can be ignored (Bulkowski 2002: 303).

The implemented program finds all triangles that satisfy the criteria described above and saves the triangles data into an output CSV-file which can be used to visualize the found triangles in Excel software. The following figure shows triangles found by the program based on High and Low rates of EUR-USD.

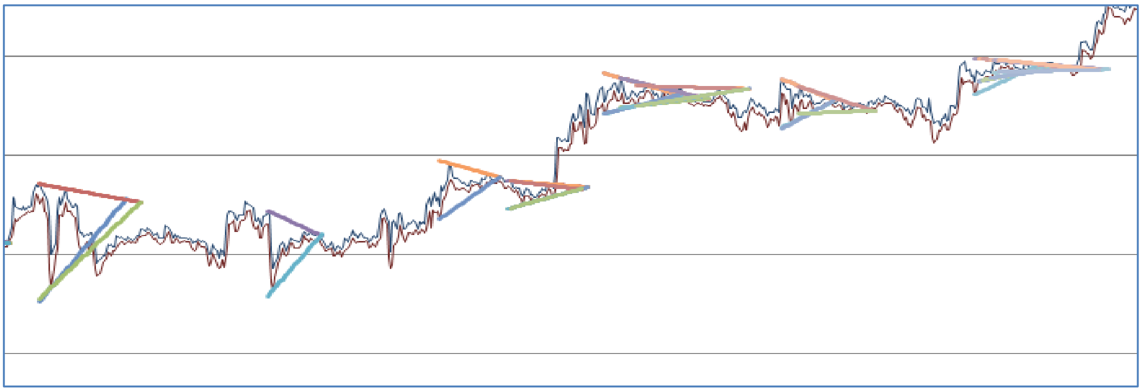


Figure 19. Program's output example opened in Excel and visualized using a line chart (created by author).

The triangle searching program takes into account several additional parameters: minimum and maximum trend line length. These parameters were needed to limit durations of the searched triangles. A triangle duration can be understood as a distance between the first bouncing point of the pattern and the breakout point. A triangle duration is considered to be in a relationship with trading horizons (Classic patterns: 5) and to be an indicator of the duration of the influence of the pattern. The longer pattern, the longer it will take for the rate to the target level. Hence, it was decided not to mix together long triangles with the short triangles in one particular experiment.

Thus, it was decided to limit the trend lines length (and consequently the triangle chart pattern length) between 10 and 30 periods. If the number of periods between a trend line's bouncing points is out of the defined limits then such trend line is not considered for triangle building. Another reason why it was decided to find only relatively shorter triangles is because searching for longer triangles would lead to a negative effect: a huge number of very similar triangles (as on the Figure 20 below). Presenting the big number of similar triangle to a neural network would force it to memorize specifics of these similar triangles instead of generalization.

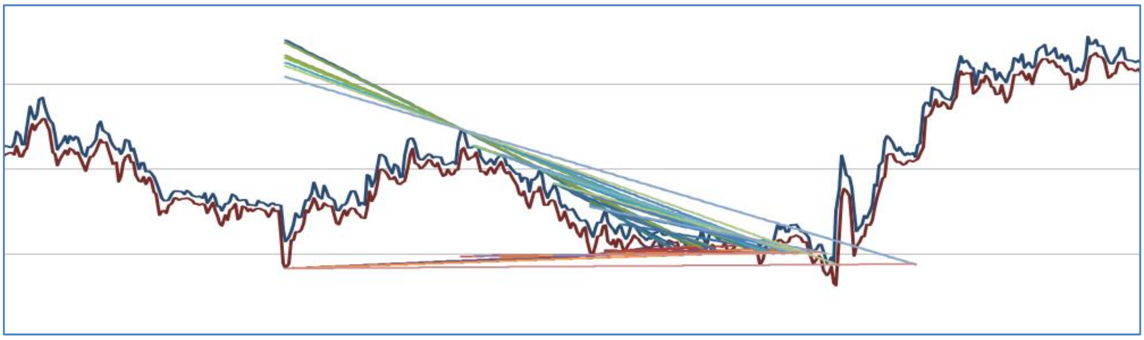


Figure 20. Many similar long triangles concentrated in one place (created by author).

In order to cover all sorts of triangle durations in the experiments without having the described problems, it was decided to make separate independent experiments for currency exchange rates for different periods such as 5 minutes, 15 minutes, 30 minutes, 1 hour and 4 hours.

The table below shows the list of investigated time series and corresponding numbers of triangles found for the rate series by the triangle searching software implemented by the author. As mentioned above in the chapter 2.2, triangles were searched based on High/Low series and on Open/Close series.

Table 1. Numbers of triangles found for different rate series

Series	**History range**	**Number of periods**	**Number of triangles (High/Low)**	**Number of triangles (Open/Close)**
EUR-USD 5 minutes	01.01.2010 - 01.01.2013	222985	21715	19869
EUR-USD 15 minutes	01.01.2010 - 01.01.2013	74851	4586	3347
EUR-USD 30 minutes	01.01.2004 - 01.10.2013	117587	5453	4930
EUR-USD 1 hour	01.01.2006 - 01.01.2013	43334	1233	876

EUR-USD 4 hours	01.01.2004 - 01.10.2013	15641	259	96

2.3. Choosing neural network output parameters

As mentioned above, it was decided that neural networks will have to produce one of three rate direction signals: rate will grow, rate will decrease or rate has no clear direction.

Two different ways to present the rate direction were investigated in this work:

1. The rate direction presented by a single output neuron with value in a range of [-1, 1]. The following sub-ranges are interpreted in this way:

    a. [-1, -0.5] means that the rate will decrease. A stronger decreasing will drive the value closer to -1.

    b. [-0.5, 0.5] means that the rate will not change significantly. If a triangle sample wasn't followed by a clear rate movement direction, the desired output value was set 0. This was done in order to better isolate such signals from signals that are used in cases when the rate has a clear direction.

    c. [0.5, 1] means that the will increase. A stronger increasing will drive the value closer to -1. Such ranges [-1, -0.5], [-0.5, 0.5] and [0.5, 1] were chosen in order to equally separate three possible answers in one range of [-1, 1]. For example, a value which is close to 1 means that the rate should definitely increase and a value which is close to 0 means that there is no clear rate direction. That's why it was decided that the threshold can be 0.5 which is the middle between the area of rate growth and the area of no clear direction. In addition, if a neural network is trained according to the ranges [-1, -0.5], [-0.5, 0.5] and [0.5, 1] then in theory it should learn to produce signals according to the ranges that were used for the training. However, it is possible that the threshold can shift after neural network is trained because of non-linearities.

2. The signal is presented using three neurons for each of three classes: rate will grow, rate will decrease or there will be no clear rate movement direction.

The first output is suitable for a function approximation neural network. However, the approach has a problem. The problem of this approach is that an assumption is made that the rate growth signal value is more distant from rate decreasing signal than from the no clear move signal, i.e. no clear move area is located between rate growth and rate decreasing ranges. In spite of that it may sound logical, such approach can have a bad side-effect, for example if a neural network will conclude there will be a clear move and the signal looks a bit more like “growth” than “decreasing” then the neural network can calculate the outcome as an average with a slight shift to the “growth”-side and this can fall into “no clear move” area. As the result, such neural network will prefer to doubt. The conclusion was confirmed by experiments that were done as a part of this work.

The second output is good for using it in a pattern recognition (classification) neural network. This approach doesn’t have the problem that was found for the first approach, because all three classes are presented by separate output neurons. For this reason, it was decided to proceed only with classification networks with three output neurons.

The training of such neural network requires samples that include the desired target outputs. In other words, there was a need to calculate the expected output for every input sample i.e. for every found triangle. To do that, another software program was implemented by author in order to calculate desired rate change direction signals for neural network training. Author used Java programming language to implement the software program.

The program takes future *N* periods (10 periods were used in this work) for the given moment and provides a rate change direction signal (rate moves up, rate moves down or no clear move). Taking into account that the chosen triangle pattern length is from 10 to 30 periods, 10 periods for tracking the following rate move must be sufficient, because after the period of consolidation the rate should make a powerful and quick move in a certain direction (Schlossberg 2006: 128, Cheng 2007: 179) as mentioned in the chapter 1.2. In order to calculate the signal the following configurable parameters were taken into account:

1. **Minimum considerable rate change level**. This parameter defines minimum lower and upper level of a change (as described in 2.1). If a rate change doesn't reach any of these levels, such case is treated as an unclear rate direction.

2. Maximum acceptable throwback/pullback. When a rate goes up, but makes a throwback which exceeds the maximum acceptable level, such case is not treated as a move in the upward direction (same with rate going down and pullbacks).

The algorithm produces a signal that the rate goes up/down if the minimum upper/lower level is reached during the given number of period and there was no limit exceeding throwbacks/pullbacks before reaching the upper/lower level. The classification neural network is presented by three classes (three output neurons). The desired output for the classification neural network is set in the following way depending on which signal is expected as per the table below:

Table 2. Desired output values for the classification neural network

Signal	**Output 1**	**Output 2**	**Output 3**
Rate moves up	1	0	0
No clear rate movement direction	0	1	0
Rate moves down	0	0	1

As described, calculated rate change direction depends on a selected minimum change level parameter and minimum acceptable throwback/pullback. Triangle hypotheses don't provide a clear guideline how to choose the values. In other words, a triangle chart pattern is only an indication of a direction of the rate change, but not of the timing or of the value (Lucas 2013: 18). So, there was a need to find the parameters values experimentally.

When choosing the minimum change level, the following costs have to be taken into account: spread, slippage risk and other costs. Of course, the change level has to cover all the costs, otherwise for example opening a position which aims to reach such level will not give a profit.

Obviously, a maximum acceptable throwback/pullback must be significantly smaller than the minimum considerable change level, because a neural network is not supposed to be trained so that it would predict rate move in one direction when the move actually is going to happen in the opposite direction. Also, it is clear that the maximum acceptable throwback/pullback must not be strictly zero, because this would exclude from the consideration cases when a rate goes a bit back after a breakout and before doing a strong move. So, it was concluded that the maximum acceptable throwback/pullback could be around three times smaller than the minimum considerable change level. Such ratio was chosen, because there is opinion that the minimum requirement for the successful Foreign Exchange trading must be at least 3:1 reward/risk ratio meaning that the possible gain must be at least three times more that the possible loss (Gilbert 2012: 37). This ratio determines parameters when to exit from a losing position in case of a throwback/pullback (Gilbert 2012: 37).

The author has implemented another algorithm that helps in picking a value for the minimum considerable change level. The algorithm analyses all found triangles and their breakout points and then produces a maximum change achieved during 10 periods assuming that the throwback/pullback will not exceed its limit of being at least three times smaller. The output of the function is a chart that visually shows the maximum rate changes for every breakout point (the values are sorted by the maximum rate change; positive numbers on the vertical axis mean that the rate goes up; negative numbers mean that the rate goes down). The Figure 21 below shows a sample output of the algorithm and it visualizes rate change directions of 5700 training samples. Considering that the significant part of triangles must predict a strong rate change (according to the hypotheses described in chapter 1.2), looking at the figure below helps in choosing the minimum considerable change level (which could be 0.002 for example in this particular displayed case). But anyway, it makes sense to experimentally try different values of minimum considerable rate change in order to find best performing prediction model.

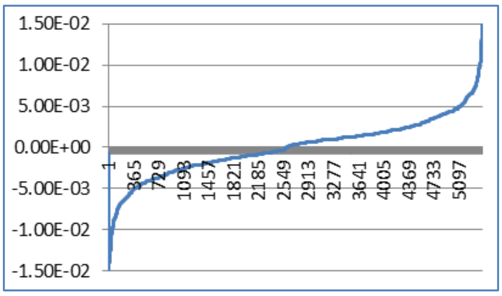


Figure 21. Maxumum rate changes of breakout points from a training set (created by author).

2.4. Choosing neural network input parameters

2.4.1. Rates series descriptors

According to the chosen model, information about a triangle has to be passed to the input of a neural network in order to get the rate movement direction in the output. So, it had to be decided which parameters describe a triangle pattern in the best way and could be suitable for neural network input. Triangle chart pattern hypotheses take into account the following data: rate behavior before and within a triangle pattern (before the breakout point moment) and also other different criteria related to the rate behavior such as triangle type (ascending, descending or symmetrical), breakout direction etc.

The best thing that describes rate behavior before and within a triangle pattern are obviously the **absolute values of the rates** for the periods of the triangle formation and for the periods before the triangle. The absolute values of rates fully cover rate behavior within and before the triangle and all characteristics of the triangle can be deducted from the rates data.

So, a typical temptation in choosing neural network inputs would be just to take the absolute values of the rates, namely: a sequence of Open rates for the breakout point and a certain number of periods before the breakout point period; same for High, Low and

Close rates. However, a quite obvious but important conclusion regarding choosing the input parameters is that absolute rates are not suitable because of the following reason: rates series is not stationary time series (Mehta 1995: 191), meaning that its joint probability distribution changes when shifted in time. Therefore the rates series are not suitable neural network training (Matignon 2005: 509), for example because the absolute rate ranges are different for different time intervals. And this means that if a current tested breakout point's rate will not be in a range of training data rates then such neural network will not be able to predict for the current breakout point.

So, it is clear enough that absolute rates are not suitable for neural network input and that's why it was decided to consider **rate changes** instead of the absolute rates. For example instead of period's *High(i)*, more appropriate would be using *High(i) – High(i – 1)* where *i* is a period's index. Rate changes are stationary, normally distributed and their mean is usually close to 0 and consequently are suitable for neural network trainings (Matignon 2005: 509).

Thus, the following parameters were considered as candidates for a neural network input:

1. A sequence of High rate changes and Low rate changes for the breakout point period and a certain number of periods before the breakout point period. 40 periods were selected as an appropriate time window for the sequence of rates, because the chosen maximum duration of a triangle formation was 30 periods and there was also a need for information before the triangle took place. Alternatively a sequence of 40 last *MAX(Open(i), Close(i)) – MAX(Open(i – 1), Close(i – 1))* and 40 last *MIN(Open(i), Close(i)) – MIN(Open(i – 1), Close(i – 1))* rate changes was used where *i* is a period's index. These parameters are used in case if triangles were built on top of Open and Close series (as explained in the chapter 2.2).

2. Current date and time of the breakout point. The parameters are presented by four values: a year, a month, a day and a day time; and the parameters were considered for a neural network input, because if there is any seasonality in the

currency rates movement (Yao, Tan 2002: 189) then a neural network could potentially capture them.

3. Volume (number of trading operations completed during the period) sequence. According to the theory, the increase of volume at a breakout is very important (Murphy 2009: 66).

After making experiments with the considered neural network inputs, it was found that providing rate changes to the input creates another problem: the number of input parameters becomes too big. For example, using a time window of 40 periods will mean that there will be at least $40 + 40 = 80$ inputs in the neural network (highs series + low series). Learning of non-linear dependencies of at least 80 input neurons will require having neural network with many hidden neurons and consequently a huge number of weighed parameters (connections and biases). Such approach however is extremely impractical, because achieving a good performance for such network would require overcoming a local minimum problem in a space with a huge number of dimensions. Also neural networks with a large number of weights are undesirable because of the danger over over-fitting (Duda *et al* 2000: 299). One dimension corresponds to one weighed parameter (a connection weight or a bias). For example, if the number of hidden neurons will be 20 and there will be three output neurons then the total number of weighed parameters will become at least $80 \times 20 + 20 + 20 \times 3 + 3 = 1683$ (number of connection between input and hidden layer + hidden layer biases + number of connection between hidden and output layer + output layer biases). Training of such network would require an enormous amount of training samples.

The number of samples required for neural network training frequently presents difficulties. There are some heuristic guidelines, which relate the number of cases needed to the size of the network. The simplest of these says that there should be ten times as many training samples as connections in the network. But in reality, the number of required samples also depends on complexity of the approximated function and on the quality of the input data. Ideally the samples have to fully cover the whole input space. Anyway, as the number of variables increases, the number of samples required increases nonlinearly, so that with even a fairly small number of variables a huge number of cases are required. This problem is known as "the curse of dimensionality" (Neural Networks… 2002). An attempt to train a neural network with rates change sequences in the input didn't give good results and demonstrated that the trained network prediction accuracy was low (around 33%).

As the result, it become clear that there is a need to find a more compact way to represent the rates data than just a sequence of rate changes. More compact way would require a smaller number of input neurons which would make training of such neural network more realistic, because the number of weighed parameters would reduce many times. Chapter 2.4.2 below covers the alternative approach in more details.

In addition, it was also decided not to use trades volume series provided in the historical EUR-USD rates data, because volume is local to a given market maker, describes just a tiny part of all operations executed on the whole market and doesn't reflect the overall activity of the whole Foreign Exchange market in regard EUR-USD exchange operations (as mentioned in the chapter 1.2).

Also, it was decided not to use date and time in the neural network's input. Testing a network with date and time in the input didn't give good results, because in this case, the neural network tends to prefer learning certain moments in time instead of learning based on the rest of parameters. This lead to a situation when testing of a trained neural network against test samples resulted in a low accuracy if the test samples belonged to a different time interval.

2.4.2. Triangle descriptors

As rates and rate changes were found not suitable for neural network input, there was a need to find another way to describe a triangle chart pattern. One of the ways to represent rates data in more compact way is to use a zig-zag style time series that breaks time series into linear segments preserving all important maximums and minimums (bouncing points in case of a triangle). Such representation is able to concisely encode price behavior data (Ge 1998: 7). However, this method is hard to use in a neural network input, because the number of linear segments can vary for different triangle samples, but the number of input neurons must be fixed in a neural network. This means that there is a need for another way to describe a triangle chart pattern using some fixed and preferably a small number of parameters in order to avoid impractically big number of input neurons.

One of the most important triangles classifiers is its type: ascending, descending or symmetrical (as per description in chapter 1.2). But it is not good to use these three mentioned types of the triangles as an input for the network, because classification of a triangle into a certain type assumes some very fuzzy logic. Instead, it was decided to express triangle's type in a more general way using a numeric value that describes the triangle direction. The benefit of the approach is that the parameter is universal enough to present a type of any triangle; it describes the triangle's shape as a numeric value which is more precise comparatively to three triangle types (ascending, descending and symmetrical). For example, the following triangle may be called a descending, in spite of that the support line is clearly not horizontal.

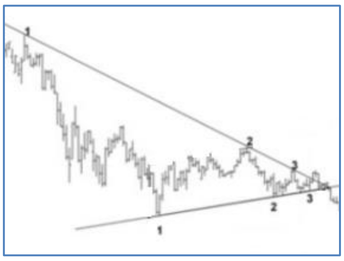


Figure 22 Descending triangle example in (Fischer 2011) with a clearly not horizontal support line.

So, as a part of this work, the author has defined a new parameter called **triangle direction** that is calculated via trend lines slopes ratio using the following formula:

(9) $$direction = \begin{cases} 1, resistanceLineSlope \geq 0 \\ -1, supportLineSlope \leq 0 \\ \begin{cases} 1 - slopesRatio, \; slopesRatio \leq 1 \\ \frac{1}{slopesRatio} - 1, \; slopesRatio > 1 \end{cases} \end{cases}$$

where *slopesRatio* is calculated in the following way:

(10) $$slopesRatio = \left|\frac{resistanceLineSlope}{supportLineSlope}\right|$$

A triangle direction falls into a range of [-1…1]. The value of -1 means that the triangle is descending, 1 means that the triangle is ascending and 0 means that the triangle is symmetrical. All intermediate values mean that the triangle is somewhere between the three types. A triangle direction can be provided as one of inputs to a neural network; and it will be a responsibility of the neural network to decide upon a direction of the future rate movement.

However, the triangle direction is not enough to describe triangle's shape fully. And the following parameters were suggested to describe the shape in the neural network input:

1. triangle's direction (as per description above);
2. triangle's apex distance from the breakout point period in a number of periods;
3. triangle's support line and the breakout point's low rates difference at the breakout point's period;
4. triangle's resistance line and the breakout point's high rates difference at the breakout point's period;
5. triangle's support line's first bouncing point distance till the breakout period;
6. triangle's resistance line's first bouncing point distance till the breakout period;
7. triangle's support line's last bouncing point distance till the breakout period;
8. triangle's resistance line's last bouncing point distance till the breakout period.

Triangle hypotheses also state that the breakout direction and strength can depend on inbound trend's slope which should be longer than a triangle formation itself (see chapter 1.2). However, there is no precise guideline regarding how many periods exactly have to be considered. For this reason and as the considered inbound trend length can depend on a triangle formation length, it was decided to express the incoming trend using several parameters:

1. 40 periods long inbound trend's slope;

2. 20 periods long inbound trend's slope;

3. 10 periods long inbound trend's slope.

If one of them turns out more important for some triangle then the neural network should discover this importance and create relationship between the parameters and the desired output accordingly. The numbers of periods 10, 20 and 40 were chosen in order to consider also trends which are same long as the shortest possible triangle chart patter and also trends which are longer than the longest possible triangle chart pattern.

The Figure 23 below visualizes the values selected as neural network inputs.

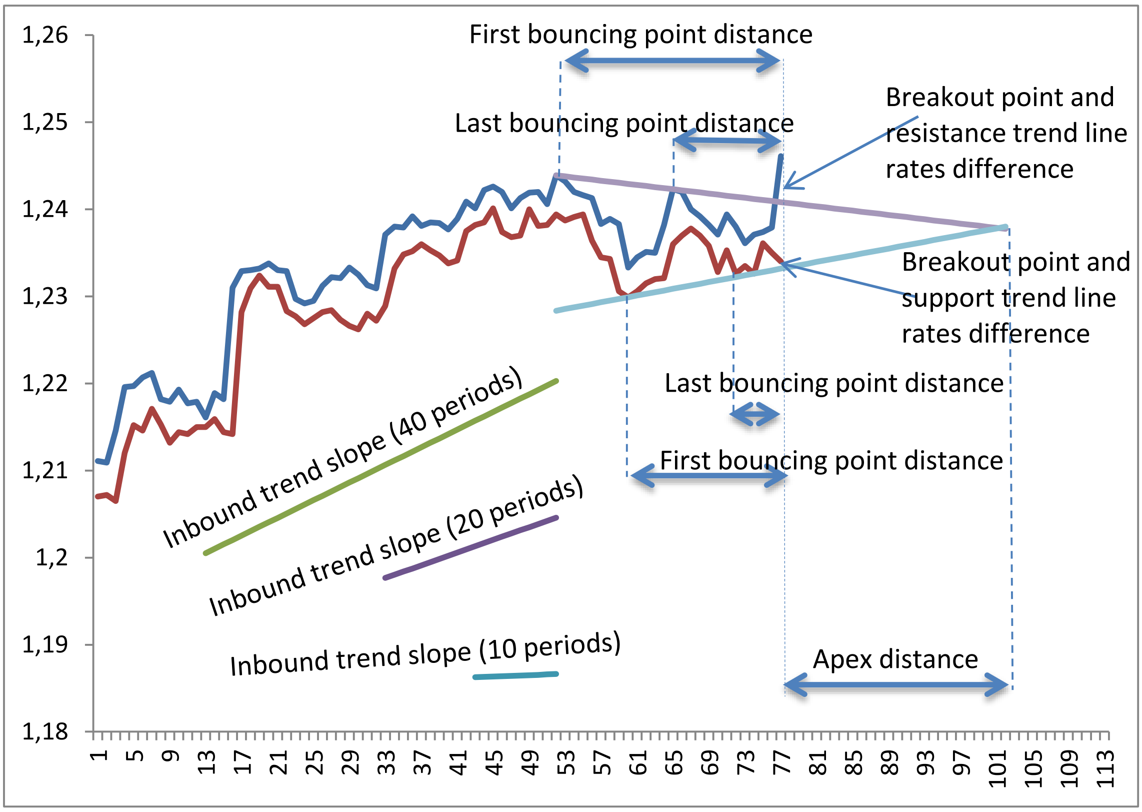


Figure 23 Neural network input parameters (created by author).

The listed input parameters fully describe a triangle's shape, its type, its starting points, main bouncing points, apex, angle, line's slopes and breakout point details. But of course the completeness of such parameters is not as good as completeness of rate changes sequence. For example, the parameters don't cover all the details of price movements within the triangle, instead they cover only the main properties considered by a triangle chart pattern hypotheses. But the most important benefit of the approach is the achieved compactness of the input.

It was decided to check a correlation between the chosen input parameters and desired output for the sample triangles. Existence of a correlation between inputs and output would mean that relation between input and output parameter is found and consequently there is a good chance to build a successful prediction model. Also input parameters that correlate with the desired output could be more important than those that don't correlate; and this can be a hint for a possible neural network model optimization by excluding less important parameters.

As the output is presented by three values, the values were recalculated into a single value by subtracting the 3rd output parameter's value from the 1st output parameter's value according to the following table:

Table 3. Single-valued expected rate moevement direction signal used for the correlation analysis

Signal	**Output neuron 1**	**Output neuron 2**	**Output neuron 3**	**Resulting single value**
Rate moves up	1	0	0	1
Rates movement direction is not clear	0	1	0	0
Rate moves down	0	0	1	-1

Then a correlation between the resulting value and every input parameter was calculated for 21715 triangles found in EUR-USD with periods of 5 minutes (01.01.2010 - 01.01.2013); and the resulting table with correlation coefficients can be seen in Appendix 5. As the table shows, there is no correlation between the input parameters and the expected output, because the correlation coefficient is close to zero. The correlation was also measured for other rates series (e.g. for 15 minutes periods, etc.), but the results were more or less same.

However, it is important to understand that correlation coefficient indicates the strength of a linear relationship between two variables. This means that low correlation is an indicator of no linear relationship, but it doesn't necessarily mean that there is no any other (non-linear) relationship between the parameters that still could be found by neural networks. A neural network with non-linear activation functions of neurons (such as logistic sigmoid or hyperbolic tangent) are able to find such relationships if they exist (Beale *et al* 2013: 1-14).

2.5. Neural network training data preparation

Training a neural network requires a preparation of training samples: sample inputs and their corresponding sample outputs. One more software program was implemented in order to facilitate neural network training data preparation by automating it (the author used Java programing language to implement the program).

The main goal of the program is to read rate time series data from a file, find triangle samples and their breakout points, calculate input and the desired target output values; and then to create two CSV files: one with neural network input training data samples and one with desired output data samples. Later, the CSV files can be used by MATLAB to read the sample input and output data for neural network training. MATLAB scripts used in this work can be found in the Appendix 4.

Preparation of sample data involves input values normalization. Data normalization means transformation of an input space into a certain range (Han, Kamber 2000: 114), e.g. [-1, 1] in case of a feed-forward neural network which uses a hyperbolic tangent activation function in the hidden layer (the activation function is described in chapter 1.3.2). It is also important to keep data set's mean value close to 0. Input normalization is needed in order to obtain good results and to significantly reduce calculation time (Haykin 1999: 200-206).

MATLAB makes automatically input data linear normalization if the input parameter values are not in the range of [-1, 1] using its MAPMINMAX function. That's why there was no need to care about normalization of all input parameters.

However, it was decided to do an additional preliminary normalization of some input parameters, such as monetary values (rate changes) in order to increase their dispersion in the range of [-1, 1] and to preserve their mean as 0 (MAPMINMAX is not doing that). The software program implemented by author automatically applied logistic sigmoid normalization for rate change input values (see sigmoid function formula in chapter 1.3.2).

As a certain period can be a breakout point for several triangles. In order to avoid adding several samples for the same breakout point to the training data, it was decided to make sure that every period can enter training samples data only once. If a period was a breakout point for several triangles then only the longest triangle was used in the samples. Not doing this could lead to a situation when a neural network would start to memorize and prefer some parameter values just because they are repeating in the training set. Longer patterns are considered more reliable than shorter (Kirkpatrick, Dahlquist 2010: 301). The probability of a random formation of a shorter triangle that wasn’t caused by demand and supply is higher than the probability of a random formation of a longer triangle.

As part of the training data preparation, the data was cleaned from invalid values. For example, some periods contained wrong rates for EUR-USD (such as 0 and 0.6) and some periods had constant rate values that didn’t change during long time. Such periods were removed from the training data.

Experimenting with neural network training has demonstrated that it is very important to split the samples data of triangle breakout points into training, validation and test sets using independent non-intersecting time intervals. Random splitting of the samples led to nice-looking results: low error and high prediction accuracy on the training, validation and test sets, but a much higher error for a new data set. This happened because there are many places in the rates series with several similar triangles in the same place. Random splitting led to a situation that training, validation and testing data included very similar samples. So, it was decided to split the samples into three sequential from time perspective sets: 60% for the training data, 20% for the validation data and 20% for the test data. The default splitting in MATLAB is 70%-15%-15%, but as there were a sufficient number of samples, it was possible to use more data for validation and test samples in order to get more reliable results.

2.6. Choosing neural network architecture and training

Defining an architecture for neural networks involves making decisions regarding neuron layers, number of neurons in the layers, neurons connections and activation functions of neurons. A feed-forward neural network (multilayer perceptron) architecture with hidden neuron layers that use a hyperbolic tangent activation function (described in 1.3.2) was chosen for the experiments. According to MATLAB guidelines (Beale *et al* 2013: 1-14) and universal approximation theorem mentioned in chapter 1.3.1, such networks can be trained to approximate any function arbitrarily well. Also multilayer perceptrons were used by other researchers for currency rates forecasting as referred in the chapter 1.4. Hyperbolic tangent activation function has been chosen, because it is proved more efficient than logistic sigmoid function as mentioned in the chapter 1.3.2.

As it was decided to use a classification neural network, output neurons use the SOFTMAX activation function (see chapter 1.3.2); so that each output neuron calculates a probability of that the sample belongs to the given class. The author also has tested another approach using function approximation neural network with the linear activation function (see chapter 1.3.2) in the output, but as described in chapter 2.3, classification network turned out to be more appropriate for the rate movement direction prediction.

So, the multilayer perceptron is used as a classification neural network with three output neurons (one for each class/signal). And there is an input neuron per each used input parameter. As mentioned above in several places, the author has tried to train a neural network using different combinations of input parameters.

A bigger open question in regard of a feed-forward neural network architecture was the number of hidden layers and the number of neurons in the layers. Universal approximation theorem says that a neural network with a single hidden layer is able to approximate any continuous function in any precision. However, the theorem doesn't say anything about the sufficient number of neurons in the hidden layer and it doesn't say if a single hidden layer network can be trained quicker than a network with multiple hidden layers. For this reason, it was decided to do experiments with both single hidden layer and two hidden layers. Also different experiments with different numbers of hidden neurons were needed.

Neural network training involves feeding of input and output samples and applying of a training method. Several different methods are applicable for multilayer perceptron as described in 1.3.3. The following training methods were used in this work in order check if any of them can provide better performance: Levenberg-Marquardt, Resilent Propagation, BFGS Quasi-Newton and Scaled Conjugate Gradient (see chapter 1.3.3). The training methods utilized in this work were chosen based on guidelines and comparison of the algorithms provided by MATLAB (Beale *et al* 2013: 2-29, 8-18). According to the guidelines, Levenberg-Marquardt, Resilent Propagation and Scaled Conjugate Gradient methods have faster convergence. In addition to being fast, Levenberg-Marquardt provides higher accuracy of training. However, the advantages of the method decrease when there is a bigger number of weighted parameters especially in case of classification tasks.

When samples are provided to a neural network in MATLAB, it automatically divides the samples into three sets: training, validation and test. For the reason mentioned in 2.5, the samples have to be split sequentially, not randomly.

As mentioned in chapter 1.3.3, a confusion plot was used to assess the result of training experiments. As was found during experiments, the higher percentage of correct answers don't necessarily mean good results. The important conclusion was that in order to get adequate confusion percentage parameter, there must be an equal number of samples for every class. Otherwise, if some class is presented by more samples then it is possible to get into a situation when a neural network will just learn to prefer the class, because the total average number of correct answers will look good. Especially it is important to have the equal number of samples of different classes for training and validation sets, because these sets are used to train a neural network. In order to have the equal number of samples belonging to different classes the author has adjusted the data preparation software to randomly remove samples until all classes are represented equally.

As described above, a neural network with too many neurons can learn more complex relationships, but such network introduces another problem: a local minimum problem. In order to try to overcome the local minimum and to achieve better results, trainings with the same architecture and same training data were done several times and the weights were randomly reset after each repetition. Another tried way to solve the local minimum was to prolong training by increasing maximum allowed validation fail count (MATLAB defaults the validation fail count to 6, however increasing it helped to achieve a lower MSE, especially for Resilent Propagation learning method). And one more way to avoid the local minimum problem is to use a smaller network with a smaller number of neurons.

Various network architectures were investigated in order to determine the optimal multilayer perceptron neural network architecture (i.e. lowest confusion rate and the lowest root mean square error) by combining together different training algorithms, different numbers of hidden layers and different numbers of neurons. The best results were produced by Levenberg-Marquardt (LMA) and Resilent Propagation (RProp) training methods.

Separate experiments were made for different types of ascending, descending and symmetrical networks independently. This was needed to check if any of these types of triangle patterns have a stronger relationship with the right rate movement direction, because there is an opinion that ascending and descending triangles have a more decisive predictive quality (Murphy 2009: 66; Schabacker 2005: 88-89). However, the experiments didn't lead to the higher prediction accuracy than accuracy achieved by experiments which took all three types of triangles together.

Separate experiments were made for different combinations of input parameters. For the reasons described in the chapter 2.3, the better results were achieved when rate changes sequences and date-time parameters were excluded from the training samples. However, it is still an open question if there is any better combination of inputs in order to predict the chosen output parameters. Unfortunately, neural networks are black-box methods meaning that there is no explicit form to explain and analyze the relationship between inputs and outputs (Zhang *et al* 1998: 55). This means that there is no other way to check the importance of any input parameter rather than doing a separate experiment with a different combination of inputs.

Every neural network with a certain configuration was re-trained several times with resetting weights to random values in order to try to avoid a local minimum.

2.7. Results analysis

The table in Appendix 1 summarizes the best results achieved during the experiments. However, it is important to mention that the total number of experiments done during this work was several hundred; but the weaker results were not included into the table, because they don't present any interest as non-successful training may have several reasons, like improper input, improper neural network architecture, over-fitting, reaching local minimum instead of a global minimum. Weak results don't prove that there is no relationship between the input and output. The weak results prove only that the neural network training didn't give the expected results.

The best results table can be seen in Appendix 1 and it presents results of eight experiments that used different input data and different configurations of neural networks. The details of the results (confusion plots and performance plots) for these network experiments can be found in the Appendix 2. The most important criteria that assess the model achieved in a training experiment are prediction accuracy percentages for the test set. The test set wasn't used for a neural network training, that's why the numbers are reliable.

As can be seen from the best results summary table in Appendix 1, the overall achieved prediction accuracy is around 40%. This means that a neural network is able to predict one of three rate movement direction with accuracy 40%. Totally random guessing picking one of three possible direction would give the result of 33.(3)% (because there were equal number of samples from every class representing three possible rate change directions). So, this means that the experiments have proven that neural networks are able to identify some relations between triangle chart patterns and the rate movement behavior after the breakout point, but the found relationship is not very strong. The best results table presents experiments that cover all considered rate series periods (5 minutes, 15 minutes, 30 minutes, 1 hour and 4 hours) and this means that neural networks succeed to find regularities in triangle chart patterns of different durations.

But in order to reach the main conclusion whether neural networks based approach's prediction accuracy is higher than the prediction accuracy that can be achieved by a straightforward following of the triangle pattern rules, there is a need to calculate how many right answers would give the straightforward following of the triangle pattern rules (the classical guidelines are described in chapter 1.2). Taking an example of 21715 triangles found for EUR-USD with a period of 5 minutes (see Table 1 in chapter 2.2), it is possible to calculate how many of the found triangles match the criteria of the classical guidelines and how many of them predict the currency rate movement direction correctly. The expected currency rate movement directions were calculated using minimum considerable change level 0.0006 and maximum throwback/pullback 0.0002, because maximum prediction accuracy was achieved for these specific levels. Also combinations of minimum considerable change level and maximum acceptable throwback/pullback (0.0003, 0.0001), (0.0009, 0.0003), (0.0012, 0.0004), (0.0015, 0.0005) and (0.0018, 0.0006) were tried, but gave lower prediction accuracy. The calculations were done in Microsoft Excel by filtering those triangles which match all the criteria of triangle chart pattern and then counting those which make the correct prediction. The result of the calculations is presented in the table below.

Table 4 Percentage of correct successful prediction following triangle chart patterns hypotheses

Type of an indicator and its criteria	**Number of triangles which match the criteria**	**Number of triangles which correctly predicted rate movement direction**	**Percentage of correctly predicted rate movement directions**
Ascending triangle confirmed by a breakout to the upside. Such	1428 of 21715	389 of 1428	27.2%

triangles assume that the rate after the breakout will go up.			
Descending triangle confirmed by a breakout to the downside. Such triangles assume that the rate after the breakout will go down.	1387 of 21715	440 of 1387	31.7%
Symmetrical triangles with a growing inbound trend and confirmed by a breakout to the upside. Such triangles assume that the rate after the breakout will go up.	1040 of 21715	380 of 1040	36.5%
Symmetrical triangles with a decreasing inbound trend and confirmed by a breakout to the downside. Such triangles assume that the rate after the breakout will go down.	1994 of 21715	572 of 1994	28.7%
TOTAL	5849 of 21715	1781 of 5849	30.4%

This shows that less than a third of triangles were qualified to give rate movement (increasing or decreasing) prediction; and the achieved average accuracy was 30.4%. Now taking as an example a neural network based model built in an experiment 1 (see Appendix 2, confusion matrix, test set), the similar calculations can be done as per the Table 5 below. The neural network used 1186 triangles which is a subset (test set) of the exactly same 21715 triangles. Only the test set was considered since the results calculated for the training and validation set are usually better, because of over-fitting during training. The expected currency rate movement directions were calculated using minimum considerable change level 0.0015 and maximum throwback/pullback 0.0005, because maximum prediction accuracy was achieved for these specific levels.

Table 5 Percentage of correct successful prediction by the neural network

Type of an indicator	Number of such triangles	Number of cases when the rate went in the assumed direction	Percentage of correctly predicted rate movement directions
Neural network decides that the rate grows.	359 of 1186	160 of 359	44.6%
Neural network decides that the rate decreases.	382 of 1186	97 of 382	25.4%
TOTAL	641 of 1186	257 of 641	40.1%

The table shows that around a half of triangles was qualified to give rate movement direction (increasing or decreasing[4]) prediction; and the achieved total accuracy was 40.1%, which is significantly higher than the result achieved in case of following triangle chart pattern instructions (30.4%). It can be noticed that the model was less successful in prediction of rate decreasing than in prediction of rate increasing. But it is important to understand that using of the neural network for forecasting only rate growths can give even 44.6% accuracy and that cannot be reached by using triangle chart pattern instructions. Also a big difference was that the neural network based model was able to predict a minimum rate change 0.0015 versus 0.0006 which was predicted using the traditional approach. Predicting larger rate changes can produce higher profits. Unfortunately, there is no way to know what additional information the neural network was able to extract from a triangle chart pattern in order to achieve such result, because a neural network is a black box as also noticed by Dunis, C. and Williams, M. (2002: 19-20).

[4] Forecasting of cases when a rate movement direction in not clear is intentionally not included into the table for the sake of comparability of the results with the result which can be achieved by following triangle chart pattern instructions, because triangles are meant to predict only rate increasing and decreasing.

Anyway, this comparison demonstrates that a neural network is definitely able to provide an added value by identifying some additional relations between a triangle chart pattern and the following rate movement direction. And this in own turn means that neural networks can be helpful in analyzing of chart patterns and a trader which uses such neural network based approach is in a more beneficial position than traders who follow only triangle chart pattern instructions. In addition, using of a neural network based approach can be combined with other technical indicators in order to increase a confidence in a produced signal.

Also, it has to be said that if a neural network will predict that the rate will go up during given 10 periods, but in the reality the rate will not go up and will just balance between the minimum considerable upper level and the maximum acceptable throwback level during the given number of periods, then there still be a chance of that the rate will move in the upside direction after the 10 periods. For example in the experiment 1 (test set), in 34.8% of cases the neural network predicted that the rate will go up, but actual there was no clear movement direction during 10 periods; and still some significant part of the cases may reach the upper level later. This means that if for example a trader has opened on long position on a breakout, decided to take profit on the minimum upper level, but the pre price didn't reach the maximum throwback in 10 periods then there is still a change that the upper level will be reached later and the trader can get a profit.

Comparing to other rate change prediction models, such as investigated by Dunis, C. and Williams, M. (2002: 19-20) or Zhang, G. and Hu, M. (1997: 495-506) which used neural networks to predict next period's rate, this work investigates a model which predicts rate behavior for several periods which provides more specific details regarding the rate behavior, because the rate is predicted regarding reaching or not reaching of some certain levels (minimum considerable rate change and maximum acceptable throwback/pullback) and regarding the order of reaching the levels during the given number of periods. The benefit of the approach is that getting a prediction, a trader has a possibility to use the levels to set stop loss and take profit orders. For example, a stop loss can be set right behind the maximum acceptable throwback/pullback level and a take profit can be set to the minimum considerable change level. One-step ahead prediction models predict only rate values for the next period (e.g. High, Low, Close rate), but don’t provide other details regarding rate behavior within the period and this can be insufficient for making a decision regarding opening a position. The benefit of the next period prediction models is that a prediction can be produced at any moment; and the model created in this work is able to make a prediction only for triangle breakout moments. In addition, the model created in this work as opposed to next rate prediction modes is not able to predict situations when the rate will not move at all or the movement will be insignificant (such predictions could be used for currency options trading for example). Also, Dunis, C. and Williams, M. (2002: 19-20) reached quite high prediction accuracy 57.2% for next day rate prediction which is significantly higher than the result reached in this work. The difference in the results can be explained by the fact that intra-day rates are very noisy as opposed to the day rates (Kaastra, Boyd 1996: 220), and this could impact the accuracy achieved in this work. Also Dunis, C. and Williams, M. (2002: 19-20) used fundamental indicators for their forecasting.

The neural network based models created in this work can be used to create different trading strategies depending on a certain traded currency pair, acceptable risk, etc. Stop loss and take profit levels can be adjusted analyzing rates historical data statistically. The further investigations are needed in order to build working trading strategy based on the results of the given research. Creation of a trading strategy is out of scope of the research, but the paper can serve as a base for the creation of a trading strategy.

Anyway, the achieved models have same application domain as traditional triangle chart patterns. Traders can open long positions when rate has to grow according to the prediction. And traders can open short positions when rate has to decrease based to the prediction. As the neural network’s desired outputs were chosen according to minimum considerable rate change levels (upper and lower), the expected targets to take profit could be set up to these levels. As the neural network prediction can be wrong, it makes sense to exit from potentially losing positions cutting losses shortly.

CONCLUSION

As part of the research a currency rates movement direction prediction model was designed and created. The prediction model is based on an artificial feed-forward neural network which analyses a triangle chart pattern at its breakout moment and produces an answer whether the currency rate will go down, will go up or there will be no significant rate change after the breakout. All possible types of triangle i.e. ascending triangle, descending triangle and symmetrical triangles were considered.

First of all, in order to reach the goal of the research, the Foreign Exchange market specifics were studied and its suitability for chart pattern based analysis was evaluated. Also triangle chart pattern definitions made by different authors were investigated and compared; triangle chart pattern main characteristics were identified and their applicability for Foreign Exchange trading was analyzed. It was found that triangle chart pattern definitions details described by different authors do not match. Also some definitions cannot work well for the Foreign Exchange market. For example, the most comprehensive triangles definition assumes analysis of trades' volume which usage is not possible in case of Foreign Exchange trading. Visual analysis of triangles based on historical data demonstrated that triangles just didn't work always. These facts gave reasons to assume that use of triangles for intra-day Foreign Exchange trading still requires an additional study in order to understand the importance of different triangle chart pattern characteristics combinations for rate movement direction prediction. Artificial neural networks are typically used to find such complex relations between different parameters. During training, artificial neural networks are able to internally understand which of the provided input parameters are the most important in order to produce the desired output. Thus, the artificial neural networks theory was studied in order to apply the neural network based methodology for rate movement direction prediction in the practical part of the given work in the most appropriate way.

The neural network based prediction model was designed, built and tested on an example of triangle chart patterns found in EUR-USD currency pair rate series (5 minutes, 15 minutes, 30 minutes, 1 hour and 4 hours). Thus, this work has combined the usage of the hypotheses of triangle chart patterns with the ability of neural networks to find regularities in the currency rates historical data expressed via triangles pattern shape descriptors in order to predict a currency rate movement direction. Neural networks performance in rate movement prediction was measured for the moments when one of classical triangle chart patterns (ascending triangle, descending triangle and symmetrical triangle) was identified.

As part of this work, possible input and output parameters for the neural network were analyzed (rates series, date, time, volume and a set of triangle descriptors suggested by the author). Some of the possible input parameters such as rate series, rate change series, date, time and volume were found not suitable and finally only the most proper set of input parameters was selected.

As the result of analysis of triangles hypotheses, a set of triangles search criteria was identified and then a software program was implemented which automatically searched for triangle chart patterns for a given historical currency rates time series. Another software program was implemented in order to automate neural network training data preparation based on the found triangle chart patterns. This included also a definition and implementation of a rate movement direction signal calculation for every triangle breakout point. The calculated rate movement direction signals had to be used as the expected output during the neural networks training.

A big number of neural network training experiments was done combining different input parameters, different rates series, different neural network architectures and training methods. The experiments aimed to find neural network models with the highest prediction accuracy for the test sets. As the result of this investigation, the most appropriate network configurations were identified that were able to predict correctly the rate movement direction with the average accuracy of around 40%. Such neural networks solved the currency rate direction problem as a classification task mapping triangle shape's properties into one of three classes (rate will grow, rate will decreases or no clear rate movement direction).

The measured prediction accuracy means that the model is able to pick one right answer of three possible with probability 40% which is better than random guessing. In spite of that the predictably level is quite low, it was demonstrated that this result is still significantly higher than the prediction accuracy which can be achieved by straightforward following of triangle chart patterns instructions. This means that the created prediction models provide an added value and can be used instead of traditional ways of trading triangle chart patterns in order to improve quality of trading decisions. Trading strategies that relied on triangle chart patterns as indicators can start using neural network based model in order to predict the currency rate movement direction. It can be assumed that the same practice can be applied not only for currency markets, but also for trading other financial instruments; however additional tests have to be done in order to confirm that. A disadvantage of the neural network approach is that a neural network is not able to provide an explanation regarding which of the input parameters are most important for determining of the predicted rate movement direction.

The results achieved in this work leave a space for a further development. There still can be more possible input parameters and also other neural network architectures can be tried in order to improve prediction accuracy. Finding the best combination can be automated and optimized via genetic algorithms. Another direction is to try using Kohonen's Self Organizing Maps in order to map triangles into several types and then use the types as input for another neural network that would produce a signal based on the types. The benefit of such approach is that it would require less complex neural network architecture and consequently the training could converge better. Also other chart patterns can be investigated based the same approach in order to understand if the use of neural network can improve currency rate movement direction prediction accuracy for the patterns.

REFERENCES

1. **Allen, L.** The Encyclopedia of Money by Larry Allen. ABC-CLIO, 2009, 520 p.

2. Analyzing Chart Patterns: Triangles. Investopedia, [http://www.investopedia.com/university/charts/charts5.asp]. 01.11.2013.

3. **Anand, S., Wei-Ngan, C., Siau-Cheng, K.** Charting patterns on price history. - Proceedings of the sixth ACM SIGPLAN international conference on Functional programming, 2001, pp. 134-145.

4. **Bhandari, B.** Trading stocks with triangles. - Futures Magazine Contributors, 2013, pp. 34-36.

5. **Beale, M., Hagan, M., Demuth, H.** Neural Network Toolbox User's Guide. MATLAB, 2013, 430 p.

6. **Bhadeshia, H. K. D. H.** Neural Networks in Materials Science. - ISIJ International, 1990, Vol. 39, No. 10, pp. 966-979.

7. **Bickford, J.** Forex Shockwave Analysis. McGraw-Hill, 2007, 368 p.

8. **Bishop, C.** Neural Networks for Pattern Recognition. United States: Oxford University Press, 1996, 504 p.

9. **Bulkowski, T.** Encyclopedia of Chart Patterns. Wiley Trading, 2005a, 1032 p.

10. **Bulkowski, T.** Getting Started in Chart Patterns. Wiley, 2005b, 320 p.

11. **Bulkowski, T.** Trading Classic Chart Patterns. Wiley, 2002, 439 p.

12. Classic patterns. Kotak Securities, [http://www.kotaksecurities.com/pdf/tradesmartclassic.pdf], 92 p. 01.11.2013.

13. **Chen, A., Leung, T.** Performance Evaluation of Neural Network Architectures: The Case of Predicting Foreign Exchange Correlations. - Journal of Forecasting. Wiley InterScience, 2005, pp. 403-420.

14. **Cheng, G.** 7 Winning Strategies for Trading Forex: Real and Actionable Techniques for Profiting from the Currency Markets. Harriman House, 2007, 262 p.

15. **Cybenko, G.** Approximations by superpositions of sigmoidal functions, Mathematics of Control, Signals, and Systems. New York: Springer-Verlag, 1989, 12 p.

16. **Duda, R., Hart, P., Stork, D.** Pattern classification. Wiley-Interscience, 2000, 680 p.

17. **Dunis, C., Williams, M.** Modeling and trading the euro/US dollar exchange rate: Do neural network models perform better? Liverpool Business School, 2002, 24 p.

18. **Filimonov, A**. Структурирование и обучение нейронных сетей с применениями к задачам физической химии и медицины. Ivanovo State University, 2004, 80 p.

19. **Fischer, R.** Trading With Charts for Absolute Returns. Wiley, 2011, 226 p.

20. **Gallant, S.** Neural Network Learning and Expert Systems. A Bradford Book, 1993, 364 p.

21. **Ge, X.** Pattern Matching Financial Time Series Data. University of California Irvine, 1998, 12 p.

22. **Gilbert, B.** Fx Insider: Investment Bank Chief Foreign Exchange Trader With More Than 20 Years' Experience As A Marketmaker. BalboaPressAU, 2012, 188 p.

23. **Gurney, K.** An Introduction to Neural Networks. UCL Press, 2009, 148 p.

24. **Hagan, M., Demuth, H., Beale, M.** Neural Network Design. Boston, MA: PWS Publishing, 1996, 736 p.

25. **Han, J., Kamber, M.** Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000, 550 p.

26. **Haykin, S.** Neural Networks: A Comprehensive Foundation. Upper Saddle River, New Jersey: Prentice-Hall, 1999, 823 p.

27. History Center. FIBO Group Ltd. [http://fibogroup.com/traders/download_quotes.html] 01.11.2013.

28. **Jagerson, J.** Profiting With Forex: The Most Effective Tools and Techniques for Trading Currencies Hardcover. McGraw-Hill, 2006, 267 p.

29. **Kaastra, I., Boyd, M.** Designing a Neural Network for Forecasting Financial and Economic Time Series. – Neurocomputing, 1996, Vol. 10, pp. 215-236.

30. **Kamish, B.** Chart Patterns. - Bloomberg Financial (Book 41). Bloomberg Press, 2009, 208 p.

31. **Karlik, B., Olgac, V.** Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks. - International Journal of Artificial Intelligence And Expert Systems (IJAE), 2010, Vol. 1, Issue 4, pp. 111-122.

32. **Kirkpatrick, C., Dahlquist, J.** Technical Analysis: The Complete Resource for Financial Market Technicians. FT Press, 2010, 704 p.

33. **Kamruzzaman, J.** Artificial Neural Networks in Finance and Manufacturing. Idea Group Pub, 2006, 287 p.

34. **Lucas, K., Lai, J., Liu, N.** Chart Pattern and Least Square Support Vector Machine in Stock Forecasting. - International Journal of Advanced Engineering Applications, Vol. 2, Issue 4, 2013, pp. 18-36.

35. **Matignon, R.** Neural Network Modeling Using SAS Enterprise Miner. AuthorHouse, 2005, 604 p.

36. **Mechgoug, R., Titaouine, N.** Exchange rate prediction using Neural-Genetic Model. University of Mohamed Kheidre, Algeria, 6 p.

37. **Mehrotra, K., Mohan, C., Ranka, S.** Elements of Artificial Neural Networks. A Bradford Book, 1996, 320 p.

38. **Mehta, M.** Foreign Exchange Markets. - Refenes A. Neural Networks in the Capital Markets. Chichester: John Wiley, 1995, pp. 176-198.

39. **Murphy, J.** Charting Made Easy. Wiley Trading, 2012, 75 p.

40. **Murphy, J.** The Visual Investor: How to Spot Market Trends. Wiley Trading, 2009, 336 p.

41. Neural Networks. StatSoft, [http://www.obgyn.cam.ac.uk/cam-only/statsbook/stneunet.html], 2002.

42. **O'Keefe, R.** Making Money in Forex: Trade Like a Pro Without Giving Up Your Day Job. Wiley Trading, 2010, 222 p.

43. **Person, J.** Forex Conquered: High Probability Systems and Strategies for Active Traders. Wiley Trading, 2007, 304 p.

44. **Rockefeller, B., Schmelzer, V.** The Foreign Exchange Matrix: A new framework for understanding currency movements. Harriman House, 2013, 250 p.

45. **Rosenstreich, P.** Forex Revolution: An Insider's Guide to the Real World of Foreign Exchange Trading. FT Press, 2005, 278 p.

46. **Schabacker, R.**, Technical Analysis and Stock Market Profits. Harriman House, 2005, 472 p.

47. **Schlossberg, B.** Technical Analysis of the Currency Market: Classic Techniques for Profiting from Market Swings and Trader Sentiment. Wiley, 2006.

48. **Shamah, S.** A Foreign Exchange Primer. Wiley, 2003, 182 p.

49. **Silvani, A.** Beat the FOREX Dealer. Wiley Trading, 2008, 219 p.

50. Stock Charts – Investors Guide to Stock Charts. Liberated Stock Trader, [http://www.liberatedstocktrader.com/stock-charts/] 01.11.2013.

51. Technical analysis. AlgoSysFx - Forex Education. [http://www.algosysfx.com/index.php/education] 01.11.2013.

52. **Thomsett, M.** Support and Resistance Simplified. Columbia, Maryland: Marketplace books, 127 p.

53. Triennial Central Bank Survey, Foreign exchange turnover in April 2013: preliminary global results. Triennial Central Bank, 2013, 22 p.

54. **Vyklyuk, Y., Vukovic, D., Jovanovic, A.** FOREX prediction with neural network: USD/EUR currency pair. - Actual Problems of Economics, 2013, Vol. 148, Issue 10, pp. 261-273.

55. **Wang, P.**, The Economics of Foreign Exchange and Global Finance. Springer, 2010, 446 p.

56. **Yao, J., Tan, C.** Neural Networks in Technical Forecasting of Foreign Exchange Rates. - Neural Network in Business: Techniques and Applications. Idea Group Publishing, 2002, pp. 189-203.

57. **Zhang, G., Hu, M.** Neural Network Forecasting of the British Pound/US Dollar Exchange Rate. - Omega, Int. J. Mgmt Sci, 1997, Vol. 26, pp. 495-506.

58. **Zhang, G., Patuwo, B., Hu, M.** Forecasting with artificial neural networks: The state of the art. - International Journal of Forecasting, 1998, Vol. 14, pp. 35–62.

Appendix 1. Neural network experiments results summary

The table presents results of eight experiments which produced the higher prediction accuracy.

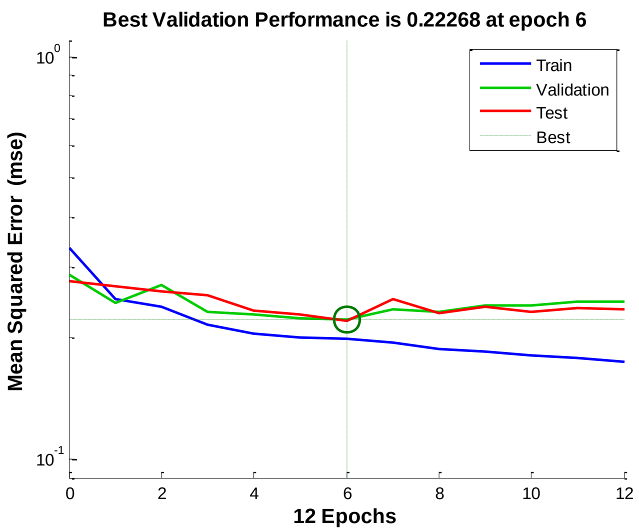
Experiment (model) number	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**	**Model 8**
Configuration:								
Rates data (pair, period, interval)	EUR-USD, 5M, 01.01.2010-01.01.2013	EUR-USD, 5M, 01.01.2010-01.01.2013	EUR-USD, 5M, 01.01.2010-01.01.2013	EUR-USD, 5M, 01.01.2010-01.01.2013	EUR-USD, 15M, 01.01.2010-01.01.2013	EUR-USD, 30M, 01.01.2004-01.10.2013	EUR-USD, 1H, 01.01.2006-01.01.2013	EUR-USD, 4H, 01.01.2004-01.10.2013
Number of samples	5931	5931	5931	5931	1170	3744	990	204
Training-validation-test split ratios, %	60-20-20	60-20-20	60-20-20	60-20-20	60-20-20	60-20-20	60-20-20	60-20-20
Number of hidden layers	1	1	1	1	1	1	1	1
Neurons in a hidden layer	30	30	30	50	50	30	30	5
Learning method	LMA	RProp	LMA	RProp	LMA	LMA	LMA	LMA
Performance	MSE	MSE	MSE	MSE	MSE	MSE	MSE	MSE
Min considerable change	0.0015	0.0015	0.002	0.002	0.003	0.003	0.003	0.005
Max acceptable throwback/pullback	0.0005	0.0005	0.0007	0.0007	0.001	0.001	0.001	0.0015
Triangles min-max length	10-30	10-30	10-30	10-30	10-30	10-30	10-30	10-30
Max validation fail count	6	30	6	30	6	6	6	30
Results:								

Training MSE	0.1993	0.2027	0.2252	0.1938	0.1863	0.2000	0.1997	0.1964
Validation MSE	0.2227	0.2169	0.2149	0.2229	0.2315	0.2268	0.2188	0.2369
Test MSE	0.2199	0.2219	0.2172	0.2140	0.2111	0.2216	0.2300	0.2322
Correct answers on the test set	**43.2%**	**40.1%**	**42.2%**	**44.5%**	**46.2%**	**40.4%**	**40.4%**	**45%**

Appendix 2. Neural network experiments results details

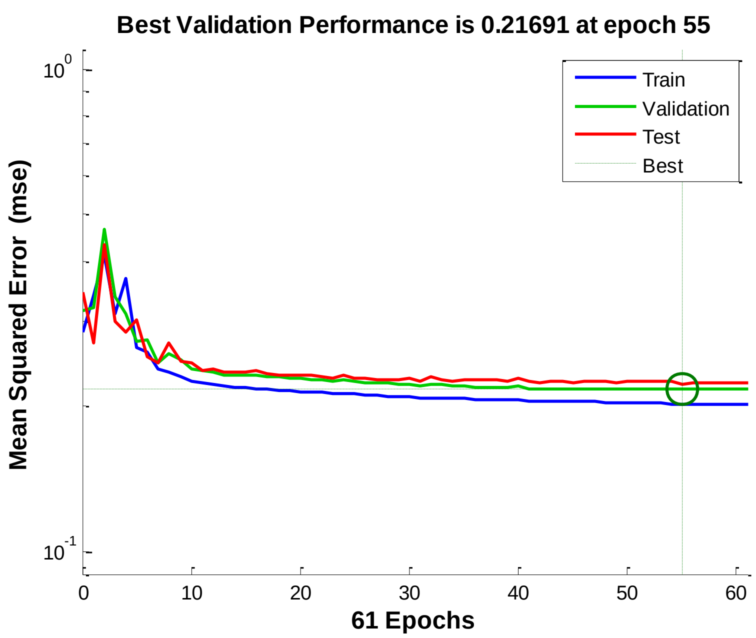
The appendix contains a confusion plot and a performance plot for each of the experiments.

Experiment 1



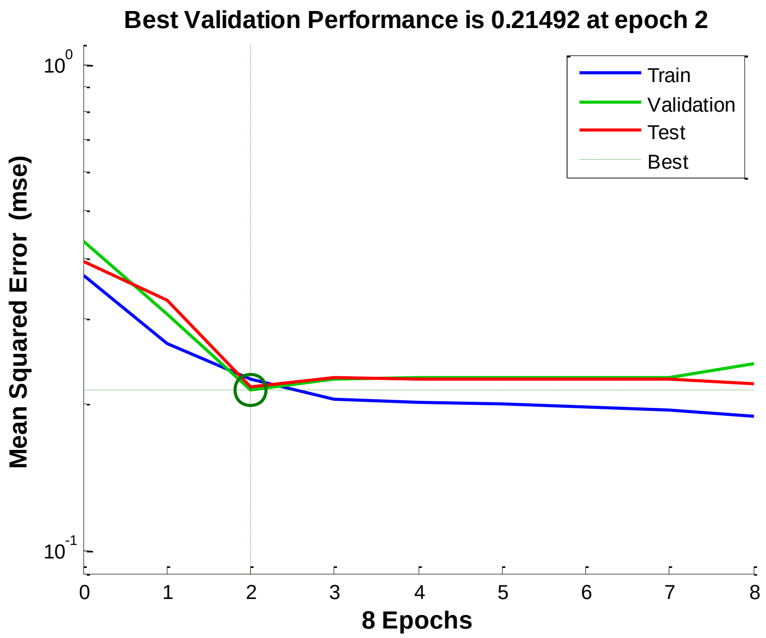
Appendix 2 (continued)

Experiment 2



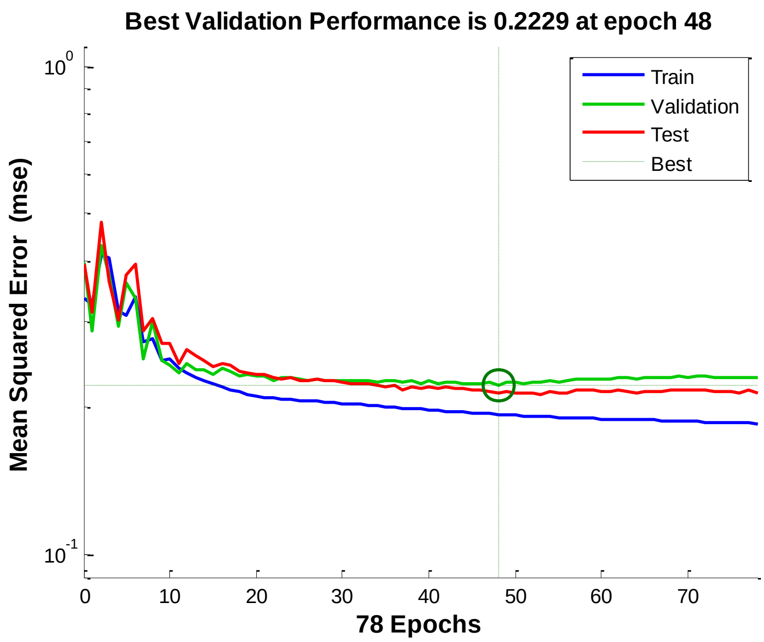
Appendix 2 (continued)

Experiment 3



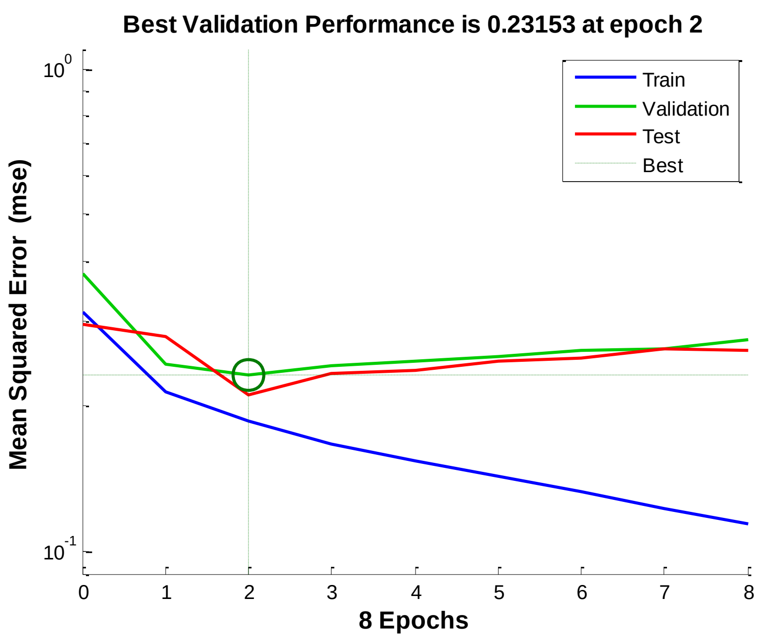
Appendix 2 (continued)

Experiment 4



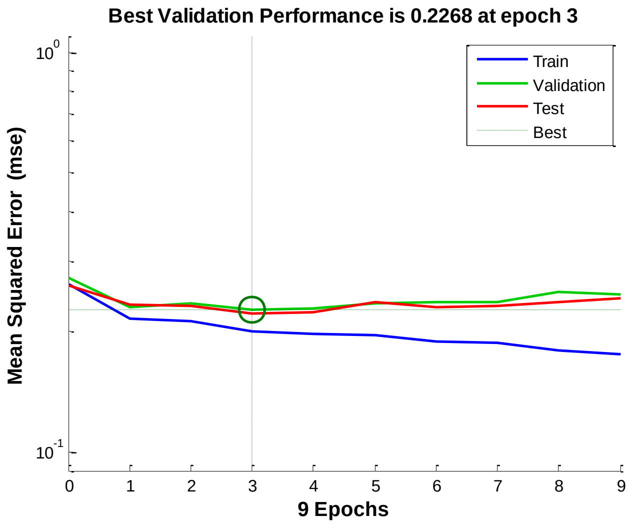
Appendix 2 (continued)

Experiment 5



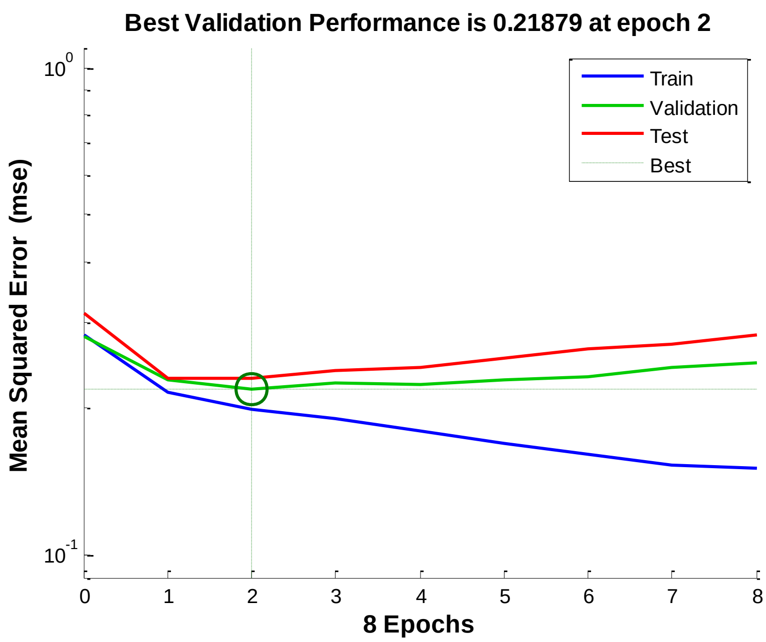
Appendix 2 (continued)

Experiment 6



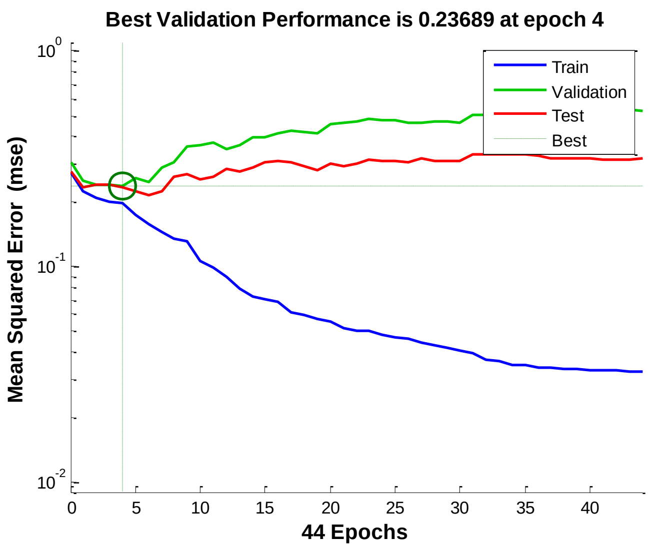
Appendix 2 (continued)

Experiment 7



Appendix 2 (continued)

Experiment 8



Appendix 3. Description of implemented software applications

The section outlines some software modules implemented by the author as part of the research:

Triangles finder. The program implements an algorithm which goes through currency exchange OHLC rates time series and finds triangle chart patterns and their breakout points according to the given criteria.

Rate movement direction calculation function. The program analyses the future rates for the given time moment and produces the rate movement direction signal.

Neural network data preparation. The program uses triangle finder module and rate movement direction calculation function module in order to find all triangle chart patterns, their breakout points, to calculate rate movement directions for the breakout points and to generate data files which can be loaded into MATLAB to provide input and output samples for neural network training.

Appendix 4. MATLAB commands used to create neural network

Reading the sample input and output CSV files into the workspace:

```
input_file = importdata(<input-file-location>);

input = transpose(input_file);

target_file = importdata(<output-file-location>);

target = transpose(target_file);
```

Creating a classification neural network with 30 hidden neurons:

```
net = patternnet(30);
```

Creating a classification neural network with 2 hidden layers with 30 neurons in the first hidden layer and 10 neurons in the second hidden layer.

```
net = patternnet([30 10]);
```

Setting a training method and a performance function (error function) used by the training method to measure neural network performance (this chooses Levenberg-Marquardt and MSE as the performance function):

```
net.trainFcn = 'trainlm';

net.performFcn = 'mse';
```

Sequential splitting of sample data into training, validation and test sets is done using the following script:

```
net.divideFcn = 'divideblock';

net.divideParam.trainRatio = 0.6;

net.divideParam.valRatio = 0.2;

net.divideParam.testRatio = 0.2;
```

Training a neural network:

```
[net,tr] = train(net,input, target);
```

Appendix 5. Correlations between neural network input parameters and expected rate movement direction

The following table shows correlation between triangle descriptors and the price move direction for 21715 triangles found in EUR-USD with periods of 5 minutes (01.01.2010 - 01.01.2013).

Compared parameter	**Correlation with the expected result**
Triangle's direction	0.0635
Apex distance from a breakout	-0.0019
Breakout point's low and support line's rate difference.	-0.0886
Breakout point's high and resistance line's rate difference.	-0.1342
Support line's first bouncing point distance to breakout	0.0081
Resistance line's first bouncing point distance to breakout	-0.0188
Support line's last bouncing point distance to breakout	0.02
Resistance line's last bouncing point distance to breakout	-0.0199
Inbound trend slope (40 periods)	0.0548
Inbound trend slope (20 periods)	0.0311
Inbound trend slope (10 periods)	0.0115

RESÜMEE

VALUUTAKURSI LIIKUMISE PROGNOOSIMINE VALUUTATURUL KOLMNURGA MUSTRI JA TEHISNÄRVIVÕRKUDE ABIL

Anton Golovko

Finantsturgude pidev areng ja kauplejate arvu pidev kasv nõuab kogu aeg uusi lahendusi, mis võimaldaksid kauplemist lihtsustada ning seda finantsturgudel kasumlikumaks teha. Üks metodoloogia, mida tihti rakendatakse hinna liikumise prognoosimiseks, on tehniline analüüs ning selle käsitletud niinimetatu graafikumustrid. Tehnilist analüüsi kasutatakse eriti tihti ka valuutaturul. Arvatakse, et tõusvad, langevad ja sümmeetrilised kolmnurgad on üldiselt usaldusväärsemad graafikumustrid ja neid on ka lihtne kasutada. Kolmnurkade hüpoteesi põhjal on valuutakursi väljamurdmine kolmnurga mustrist selline indikaator, mis näitab, et valuutakurss hakkab tugevalt liikuma kindlas suunas (ehk siis kas ülespoole või allapoole), ning seetõttu saab neid mustreid kasutada kauplemisotsuste langetamiseks. Näiteks kui valuutakurss tõuseb, siis kaupleja võib avada ostupositsiooni. Tehnilise analüüsi teooria seletab sellist kolmnurkade mustrites leitud seaduspärasust lähtuvalt kauplejate (ehk ostjate ja müüjate) käitumisest. Samas pole kolmnurkade usaldusväärsust kunagi matemaatiliselt tõestatud, kolmnurkade definitsioonid on tihti subjektiivsed, umbmäärased ning kohati ka vasturääkivad. Kõige suurem probleem seisneb aga selles, et kolmnurkade baasil tehtud prognoosid väga tihti lihtsalt ei toimi, kuigi paljudes raamatutes ja artiklites, mis käsitlevad tehnilist analüüsi ning valuutaturul kauplemist, on palju kolmnurga graafikumustrite teooriat kinnitavaid seisukohti.

Üks teine meetod, mida finantsturgudel prognoosimiseks kasutatakse, on tehisnärvivõrgud. Tehisnärvivõrgud kujutavad endast matemaatilist struktuuri, mis on võimeline funktsiooni lähendama. Närvivõrke kasutatakse näiteks järgmise perioodi

valuutakursi prognoosimiseks lähtuvalt valuutakursi ajaloolistest andmetest ning teistest indikaatoritest.

Antud töö eesmärk oli uurida, kas tehisnärvivõrkude kasutamine, et otsida seoseid kolmnurga parameetrite ning selle vahel, kuidas muutub valuutakurss tulevikus, võib suurendada kolmnurkade põhjal tehtud järelduste prognoosimisvõimet. Närvivõrke kasutatakse finantsturul prognoosimiseks, kuna nad suudavad leida keerulisi seoseid erinevate parameetrite või nähtuste vahel. Seoste leidmine kolmnurga mustrite ja valuutakursi liikumise vahel ning nende baasil eduka prognoosimismudeli ehitamine annab võimaluse kasutada kolmnurga mustreid efektiivsemalt võrreldes sellise lähenemisega, mille puhul kasutatakse kolmnurkade mustreid puhtalt tehnilise analüüsi teoorias toodud instruktsioonide põhjal. Antud uurimisteema sai valitud sellepärast, et efektiivsema kolmnurkade kasutamise meetodi leidmine võib anda kauplejatele võimaluse teenida kauplemisel suuremat kasumit. Sellise meetodi võimalikeks kasutajateks võivad olla pangad, fondid, kindlustusseltsid ning teised valuutakauplemisega tegelevad osapooled.

Selles töös läbiviidud uuring on tehtud valuutaturu ning valuutapaari EUR-USD näitel, kuna valuutaturg on kõige volatiilsem ja likviidsem finantsturg ning EUR-USD kõige rohkem kaubeldav valuutapaar. Vastavalt töö eesmärgile pidid loodud närvivõrgud prognoosima valuutakursi liikumise suunda sellistel ajahetkedel, kus toimub valuutakursi väljamurdmine klassikalise kolmnurga (tõusva, langeva või sümmeetrilise kolmnurga) mustrist. Seega on selles töös kombineeritud teoreetilised hüpoteesid kolmnurkade kohta koos närvivõrkude oskusega otsida seaduspärasusi valuutakursside ajaloolistes andmetes selleks, et teada saada, kas valuutakurss tõuseb, langeb või olulist kursimuutust ei toimu.

Töö tegemisel on analüüsitud valuutaturu omapärasid, tehnilise analüüsi teooriat kolmnurga mustri kohta, närvivõrkude teooriat ning teiste teadlaste töid, milles on analüüsitud närvivõrkude kasutamist finantsturgudel hinna prognoosimiseks tulevikus. See teoreetiline osa uurimusest oli vajalik selleks, et kontrollida kolmnurga mustrite rakendatavust valuutaturul, ning veel selleks, et selgitada, millised närvivõrkudele tuginevad meetodid on rakendatavad prognoosimiseks finantsturgudel.

Kõige enam finantsprognoosimiseks kasutatavaks närvivõrgu tüübiks on mitmekihiline pertsepron. Töö käigus oli vaja leida niisugune närvivõrgu (ehk antud juhul pertseptroni) arhitektuur, mis oleks kõige sobivam valuutakursi hinnamuutuse suuna prognoosimiseks lähtuvalt kolmnurga mustri andmetest. Kuna erinevate ülesannete lahendamiseks kasutatakse erinevat närvivõrgu mudelit, mis tähendab erinevaid närvivõrgu sisend- ja väljundparameetreid, neuronite kihtide arvu, neuronite arvu, neuronite struktuuri, sisendandmeid jne, oli töö käigus vaja langetada otsuseid erinevate mudelite sobimise või mittesobimise kohta.

Töö tegemisel olid läbi analüüsitud võimalikud närvivõrkude väljund- ja sisendparameetrid (kursimuutused, kuupäev, kellaaeg, maht ning autori pakutud kolmnurga mustrit kirjeldavad omadused). Selgus, et osade parameetrite kasutamine ei olnud otstarbekas, mistõttu lõpuks valiti välja ainult sobivate parameetrite hulk. Kolmnurkade hüpoteeside analüüsimisel identifitseeris autor kriteeriume kolmnurkade otsimiseks ning realiseeris tarkvaraprogrammi, mis otsib kolmnurki etteantud valuutakursi aegreal. Lisaks realiseeriti veel üks rakendus, mis valmistab ajalooliste andmete põhjal ette sisendandmed närvivõrkude treenimiseks. Närvivõrgu treenimise jaoks oli tarvis ka defineerida ning realiseerida algoritm, mis arvutas ajalooliste andmete põhjal iga leitud kolmnurga väljamurdmise punkti jaoks järgneva kursi liikumise suuna.

Töö käigus on tehtud arvukalt eksperimente närvivõrkudega, kus olid kombineeritud erinevad sisend- ja väljundparameetrid, erinevad valuutakursside aegread, erinevad närvivõrkude arhitektuurid ning erinevad närvivõrkude treenimise meetodid. Selle tulemusena on leitud kõige sobivamad närvivõrkude arhitektuurid, mis olid suutelised prognoosima õiget valuutakursi suunda keskmiselt 40% täpsusega. Loodud närvivõrgud lahendasid valuutakursi liikumissuuna prognoosimise ülesannet nii, et klassifitseerisid kolmnurga kuju omadusi kolmest klassist ühte (kurss tõuseb, kurss langeb või kurss ei muutu oluliselt).

Täpsus 40% on suurem võrreldes selle olukorraga, kus üks võimalikust valuutakursi muutuse suunast (kurss tõuseb, kurss langeb või kurss oluliselt ei muutu) oleks prognoositud juhuslikult. Samas, vaatamata sellele, et saadud täpsus on ikkagi suhteliselt madal, on näidatud, et see närvivõrkude põhjal tehtud prognoosi täpsus on oluliselt suurem võrreldes selliste meetodite täpsusega, kus kasutatakse kolmnurki puhtalt tehnilise analüüsi teoorias toodud instruktsioonide baasil. Seega on jõutud järeldusele, et tehisnärvivõrgud on võimelised leidma lisaseosed kolmnurkade ning järgneva kursi liikumise vahel, ehk närvivõrkude alusel loodud mudelite kasutamine lubab tõsta kolmnurga mustrite baasil tehtavate järelduste prognoosimisvõimet. Seda järeldust võib kasutada kauplemisstrateegia kujundamiseks, mis otsustab tehtud signaalide põhjal, milliste instrumentidega kaubelda, millal positsiooni avada, või millal seda sulgeda jne. Kuna tehisnärvivõrk on oma olemuselt tarkvara, tähendab see seda, et närvivõrgu signaale saab kasutada ka automaatse kauplemissüsteemi ehitamiseks. Lisaks oskusele prognoosida kursi tõusu või langust, võib närvivõrgu põhjal loodud mudel olla indikaatoriks ka siis, kui valuutakurss oluliselt ei muutu. Sellist indikatsiooni traditsiooniline teooria kolmnurkade mustrite kohta ei paku. Teoreetiliselt võib saadud tulemust kasutada lisaks valuutaturule ka kauplemiseks teiste finantsinstrumentidega, kuid sellist oletust oleks vaja kinnitada lisaeksperimentidega.

Saadud uuringutulemusi on võimalik edasi arendada. Erinevate parameetrite arv, mida saab kasutada kursi liikumise prognoosimiseks, on suur. Kõige sobivama parameetrite hulga ning närvivõrgu arhitektuuri otsimiseks saab kasutada geneetilisi algoritme, mis on mõeldud sellise kombinatsiooni otsimise optimeerimiseks ning automatiseerimiseks. Teisel juhul on võimalik uurida Kohoneni võrke, mis võimaldavad lihtsustada närvivõrkude arhitektuuri ning seega teha treenimist efektiivsemaks. Lisaks saab kasutada sarnast närvivõrgu baasil loodud mudelit ka teiste tehnilise analüüsi mustrite uurimiseks.

Non-exclusive licence to reproduce thesis and make thesis public

I, Anton Golovko (date of birth: 20.08.1978),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

FOREIGN EXCHANGE RATE MOVEMENT PREDICTION USING TRIANGLE CHART PATTERNS AND ARTIFICIAL NEURAL NETWORKS,

supervised by Allan Teder,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **14.01.2014**