INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

NEURAL NETWORK TO RECOGNIZE THE MARKET VALUE OF THE PRODUCT FOR CUSTOMERS

Prof. Dr.SC. Sanja Bijaksic [1*], Prof. Dr.SC.Brano Markić [2] , Prof. Dr.SC. Arnela Bevanda [3]
[1*23] Faculty of Economics University of Mostar, Bosnia and Herzegowina
*Correspondence Author: sanja.bijaksic@sve-mo.ba

Keywords: Market value, customer behavior, brand, neural networks

Abstract

This paper analyzes the possibilities of application of artificial neural networks in discovering preferences of consumer into buying products according to their socio-demographic and economic characteristics. The objective of this paper is to improve customer satisfaction on product's and the revenues of the organizational system. Data set is the result of preprocessing activity followed by getting the fulfilled surveys from respondents. In the survey are the questions about the socio-demographic characteristics of the customers: gender, age, education and household revenues.

The artificial neural networks technique has been applied to classify the products according to the quality, taste, color of the packaging, shape of packaging, price and impression of less chemicals depending on the socio-demographic characteristics of the customers. Artificial neural networks assign the adequate product type to the customers and so help to discover their preferences, satisfaction and recognize the market value of the product for customers. Artificial neural network is built using R language which has shown satisfactory development strength and power.

Introduction

The customers are the most important part of assets of any organizational system that results in sale in the market. Nowadays, production power is enormous in comparison to sales opportunities. Therefore, every organizational system has to direct a special attention towards product market value since its development and survival in the market is directly dependent on perceived customer's benefit. It can be concluded that most of the customers behave rationally and purchase products with characteristics they prefer. Purchase of a certain product or service surely depends on a great number of factors among which socio-demographic ones have a special position.

There is an assumption that within the same group of products (the products that are mutually similar) the customer chooses the product according to his/her own preferences. The preferences are most commonly determined by socio-demographic characteristics of a customer such as age, gender, revenues, and education. Besides that this paper assumes that behaviour of a customer can be recognized by analysis of sales and direct data collection using questionnaires.

In other words a product verifies its market value in the moment of a purchase. The purchase is materialization of product market value. It gives social sense to production and services, and brings revenues and possibility of permanent production and service provision renewing. Therefore product market value is in customer's focus since it permanently supports, stimulates, and motivates him/her on a process of money exchange for a product or a service. In such a process the exchange is equivalent only at the first sight. Both a customer and a provider are simultaneously convinced that they have received more than they have given. The customer wants a product or a service more than the amount of money spent, and the provider wants money more than a product.

The paper assumes that it is possible to conceive customer behaviour with regards to their socio-demographic characteristics, and to develop a model of their behaviour using neural networks. An instrument to collect data is a questionnaire on bottled drinking water sales. The questionnaire has two sets of questions. In the first group there are socio-demographic characteristics of bottled drinking water sales such as age, gender, revenues, and education. The second group consists of questions related to product characteristics, brand, and purchase decision.

This paper shows a model of neural network that confirms market value of certain brand depending on listed socio-demographic characteristics. The model has been developed using appropriate packages and the R language functions.

Product market value

Organizational system can achieve business aims if it permanently analyses consumers' needs and behaviour, and business activities directed towards their satisfaction. Marketing is therefore one of the most important business functions since it tries to satisfy consumers' needs and wishes, and offer them those products or services with affordable market value. Marketing has to cooperate with anthropologists, psychologists, sociologists, and other experts who research consumers' needs and behaviour since in this way they can coordinate offer with target market requirements. Marketing has a role to create brand of a product or a service that will be reliable sign of expected market value.

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

Brand exists since there has been exchange of products and services in the market. American Marketing Association (AMA) defines brand as "name, term, sign, symbol or design, or a combination of them intended to identify the goods and services of one seller or group of sellers and to differentiate them from those of other sellers" (Keller, K.L. (2003), p. 3). The oldest function of a brand is "separation" or differentiation of some products from others. A customer has protected himself/herself from bad products by labelling them. Brand is not only a product but it also symbolizes relationships, attitudes, associations, and opinions that are developed between a product (service) and a consumer, and between a consumer and a company. Brand concept becomes complex and gets more manifestations such as trademark, service brand and corporative brand, and image. Ideas, information, actions, on-line products, cities, institutions, states, organisational systems, almost everything can be branded (Keller L., Aperia T., Georgson M., (2008), p. 9-24). Development of neural network model that describes and forecasts purchase of a certain brand of drinking water with regards to socio-demographic characteristics of consumers has been shown in this paper.
Brand equity is considered as a concept that was introduced in marketing literature in 1980s. Importance of brand equity is reflected in numerous benefits that it ensures to the companies that possess a brand. Brand equity is a positive differential impact that brand name has on a product or a service (Kotler P., Wong V., Saunders J. & Armstrong G., p. 556). Nowadays there are numerous different brand equity definitions, but most of the authors agree that it is value with which brand enriches a product. There are also attempts to classify different approaches to define brand equity so it is possible to identify three different avenues of approach to brand equity and they are the following:

1. Brand value – total brand value as nonmaterial assets, this approach is called financial approach;
2. Brand strength – strength of correlation between a consumer to certain brand, this approach is called behaviouristic approach;
3. Brand description – associations and beliefs that link consumers with certain brand, this approach is called cognitive approach.

Strength and description of a brand are considered as the elements of brand equity from consumers' point of view. Brand equity is that intangible brand value which determines relationship between a brand and a client. Market and intangible brand value are very tangible in financial result of a company because they assume future business success of a brand. Therefore it is understandable that more attention is paid to brand equity in relation to financial brand value. (Vranešević, T., (2007), p.177).
There are factors that influence brand equity and they can be grouped into five categories or dimensions. They are as follows: brand loyalty, brand notoriety, experienced quality, associations related to brand and other advantages related to brand ownership (Aaker A., David, (1991), str. 19.).
Every product and service is made of numerous "tangible" and "intangible" characteristics as well as of "basic" and "additional" characteristics that make them complete. It is possible to observe and manage those characteristic based on clients' needs and wishes.

Neural networks in forecasting brand equity

Neural networks attempt to imitate operational manner and structure of human brain in data processing. Basic principle is mass parallel data processing. Natural (biological) neural network presents human or animal brain (Markić, B., (2014), str. 339).
In natural neural network inputs from other neurons are accepted by dendrite, axon transfers impulses, and links with dendrites of other neurons are called synapses. Neural is the most important functional unit of a human brain. Figure 1 shows nervous cell structure. One can notice the following: dendrites i.e. parts of nervous cell that receive information; cell body processes information; axon enables transmission of processed information to other neurons; and synapses are connections between the end of axon, dendrite, and other neurons.
Neural networks are used in numerous and different areas. Some of them are as follows:

1. Economy
    1.1. Financial markets. Classical methods are almost inapplicable since nonlinear functions are involved.
    1.2. Problems of forecasting and clustering (e.g. a phase in which decisions are brought based on following great number of changeable factors or supply and demand analysis; in marketing to make customer behavior model, market segmentation etc.)
    1.3. Evaluation of credit capability of bank clients while approving loans
2. Medicine
    2.1. Diagnosing illnesses, brain functioning observation.
3. Weather forecast
4. Identification of samples (in criminology, identification of aims for military purposes, etc.)

Artificial neural network is an attempt to imitate natural neural networks by means of digital computers of general purpose. Neuron is basic element of neural network. It accepts inputs, processes them and produces one output.
The simplest artificial neural network is called perceptron. Such neural network is presented in the following figure:

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

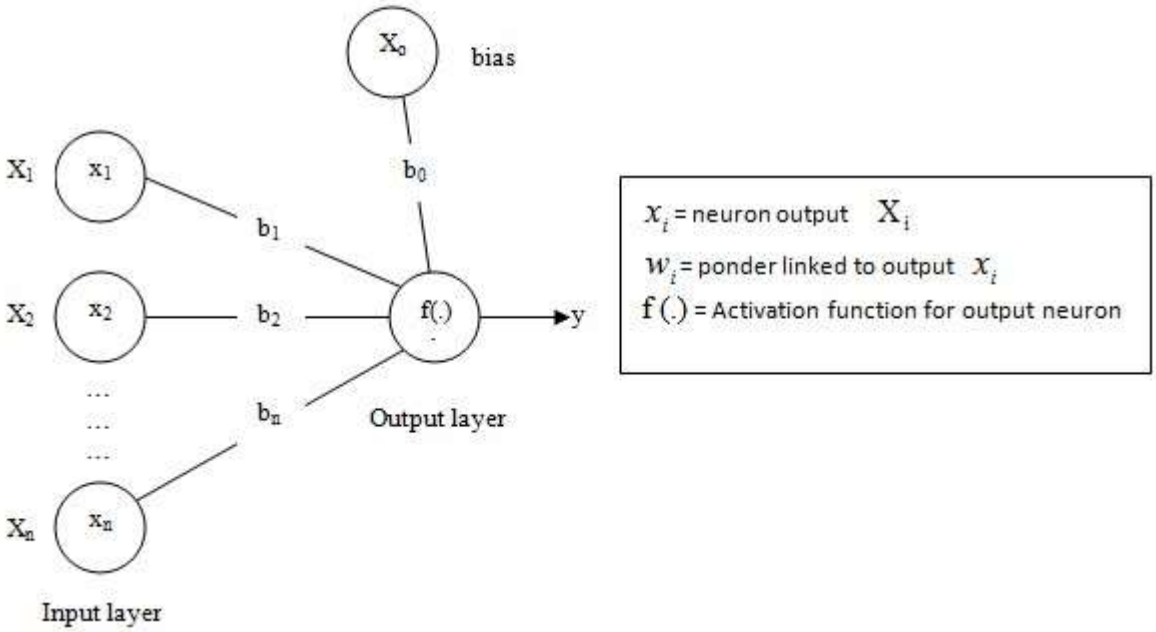


Figure 1: Perceptron

Total input for output neuron is the following:

$$s_{in} = \sum_{i=1}^{n} x_i b_i = x_0 b_0 + x_1 b_1 + x_2 b_2 + \ldots + x_n b_n$$

Neuron output at output layer is **y**. Input in output neuron from j-neuron transforms activation function ***f(.)***. Input sin_j is transformed into output $y=f(sin_j)$.

Input x_0 is special and is marked as biased. It is normal to put its value on 1. Weighting factor w_0 is also marked as biased.

Activation function accepts inputs in neuron and transforms them into outputs. There are many activation functions. The following two are especially interesting:

Bipolar function that converts every input lower or equal to 0 into 0, and converts every output higher than 0 to 1:

$f(x)= 0$ for $x\leq 0$

$f(x)=1$ for $x>0$

Second activation function is sigmoid function $y(x,\sigma) = \dfrac{1}{1+e^{-\sigma x}}$

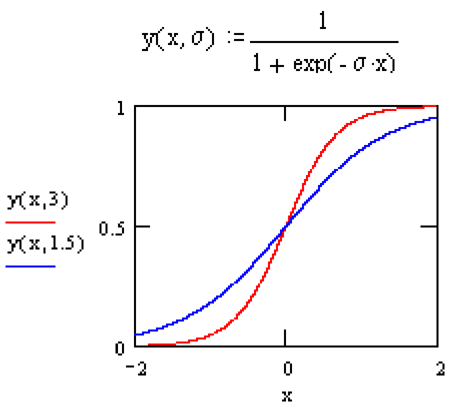


Figure 2: Sigmoid function graphs for σ=3 i σ=1.5

 ISSN: 2349- 5197

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

Za σ=1 sigmoidna funkcija se može zapisati kao $f(x) = \frac{1}{1+e^{-x}}$.[1]

It is assumed that there are only two inputs i.e. n=2 to present capacity of simple neural network. It is also assumed that activation function ***f(.)*** is bipolar function**.**

Neuron input at output level is $s_{in} = x_0 b_0 + x_1 b_1 + x_2 b_2$

If $x_0 = 1$, then $s_{in} = b_0 + x_1 b_1 + x_2 b_2$

Since activation function is bipolar function and if $s_{in} = 0$ then $b_0 + x_1 b_1 + x_2 b_2 = 0$. Variable x_2 value is as follows:
$x_2 = \quad x_1 \frac{b_1}{b_2} - \frac{b_0}{b_2}$

The following figure presents a link between x_1 and x_2.

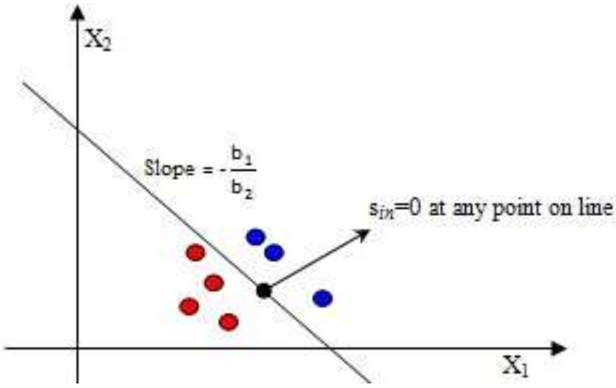


Figure 3: Functional link between x1 and x2 variables

It can be concluded that weighting factors change both incline and line position. For input pair (x_1, x_2), for example inputs determined by blue point in Figure 3, output form neuron is y=1. Output is one (+1) because for $s_{in}>0$, bipolar activation function gives the following expression +1 i.e. $f(s_{in})=1$ for $s_{in}>0$.

For input pair (x_1, x_2), for example, inputs determined by red point in Figure 3, neuron output is y=0. Output is 0 since bipolar activation function for $s_{in} \leq 0$, gives output 0.

Learning algorithm for simple neural network

An aim of simple neural network is to obtain previously set defaulted value at output neuron. That value is ***t***. If output from neural network differs from target value ***t*** then weighting factors have to be changed and adjusted. Each weighting factor must have some value at the beginning of learning. Weighting factors must be initialized. The commonest initial values for perceptron are $x_0=1$, $b_0=b_1=b_2=0$. Learning rate parameter α is a number between 0 and 1 i.e. $0<\alpha\leq 1$, output neural network target value is t, actual neural network output at output neuron is y, while input vector is $x=\{x_1,x_2,x_3,...,x_n\}$

Pseudo-code algorithm for perceptron learning can be written using the following steps:

```
WHILE (weighting factors change)
{
        FOR (every input vector x )
            {
              s_in = Σ_{i=0}^{n} x_i b_i
              y = f(sin_j)
                    IF(y ≠t)
                       {
```

[1] Sigmoid function for σ=1 can be also written as follows $f(x) = \frac{1}{1+\exp(-x)}$.

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

$b_i(new) = b_i(old) + \alpha t x_1$
}
}
}

Adjustment formula (to change) weighting factors is $b_i(new) = b_i(old) + \alpha t x_1$ ako izlazna vrijednost ***y*** nije jednaka ciljnoj vrijednosti ***t*** tj. ako if output value ***y*** is not equal to target value ***t*** i.e. if $(y \neq t)$. Loop WHILE will be derived (commands will be repeated in loop's body) until weighting factors are changing. Weighting factors' change is stopped if every neural network output (output neuron) is equal to target value (logic structure of selection in pseudo-code algorithm) for every input vector x (loop logic structure FOR in pseudo-code algorithm). Learning rate speed (α) controls changes of weighting factors speed. For example if α=0.001, then changes of weighting factors will be thousand times slower (very small (slow)) in relation to parameter α=1 value.

Find weighting factors to implement AND functions. Use bipolar step function; target values are +1 or -1. Bipolar function is the following:

$$y = f(z_{in}) = \{ \begin{matrix} 1 & \text{if } s_{in} > 0 \\ 0 & \text{if} s_{in} \leq 0 \end{matrix}$$

X_1	X_2	t
1	1	1
1	0	0
0	1	0
0	0	0

Table 1: Logic function AND

Perceptron training in the R language

Appropriate functions and packages that enable application of simple but also of multilayer neural networks are prepared in the R language. The first step in perceptron development for Table 1 data is association of values from columns of Table 1 to vectors x_1, x_2 namely:

>x_1=c(1,1,0,0)
>x_2=c(1,0,1,0)
>t=c(1,0,0,0)
>data=cbind(x_1,x_2,t)

After that package neuralnet() is downloaded to train neural network i.e. to determine weighting factors (weights) between input and output layer of perceptron :

>library(neuralnet)
>nn=neuralnet(t~x_1+x_2,data=data,hidden=0,learningrate=0.001,algorithm="backprop",err.fct="sse", linear.output=TRUE)
>plot(nn)

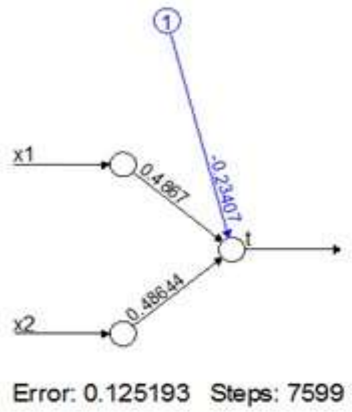


Figure 4: Perceptron neural network

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

R language package *neuralnet()* generated weighting factors in 7599 steps: w_0=-0.23407, w_1=0.4867, w_2=0.48644, and biased value is x_0=1. Value of α=0.001 (learning rate is 0.001 in neuralnet()function) is associated with learning rate parameter. Weighting factor change algorithm is called back propagation (algorithm="backprop"), and neural network error is a sum of deviation square of neural network outputs from target value (err.fct="sse").

Višeslojne neuronske mreže (multilayer perceptron)

Multilayer neural networks have one input layer, one output layer and more hidden layers of neuron. Multilayer neural networks can have different type of application in economics. They can be used in business intelligence system to estimate prices of shares, bonds, interest rates, financial derivatives, etc. scope of multilayer neural networks application is also a classification in marketing, management, and finance (e.g. segmentation of customers, suppliers, credit users, etc.)

The following diagram presents multilayer neural network (multilayer perceptron) with one input layer, one hidden, and one output layer. Input layer has n neurons, there are p neurons on hidden layer, and m neurons on output layer. Weighting factors between input and output layer are marked as a_{ij}, and weighting factors between hidden and input layer are marked as b_{jk}, where $0\leq i \leq n$, $0\leq j \leq p$, and $0\leq k \leq m$[2]

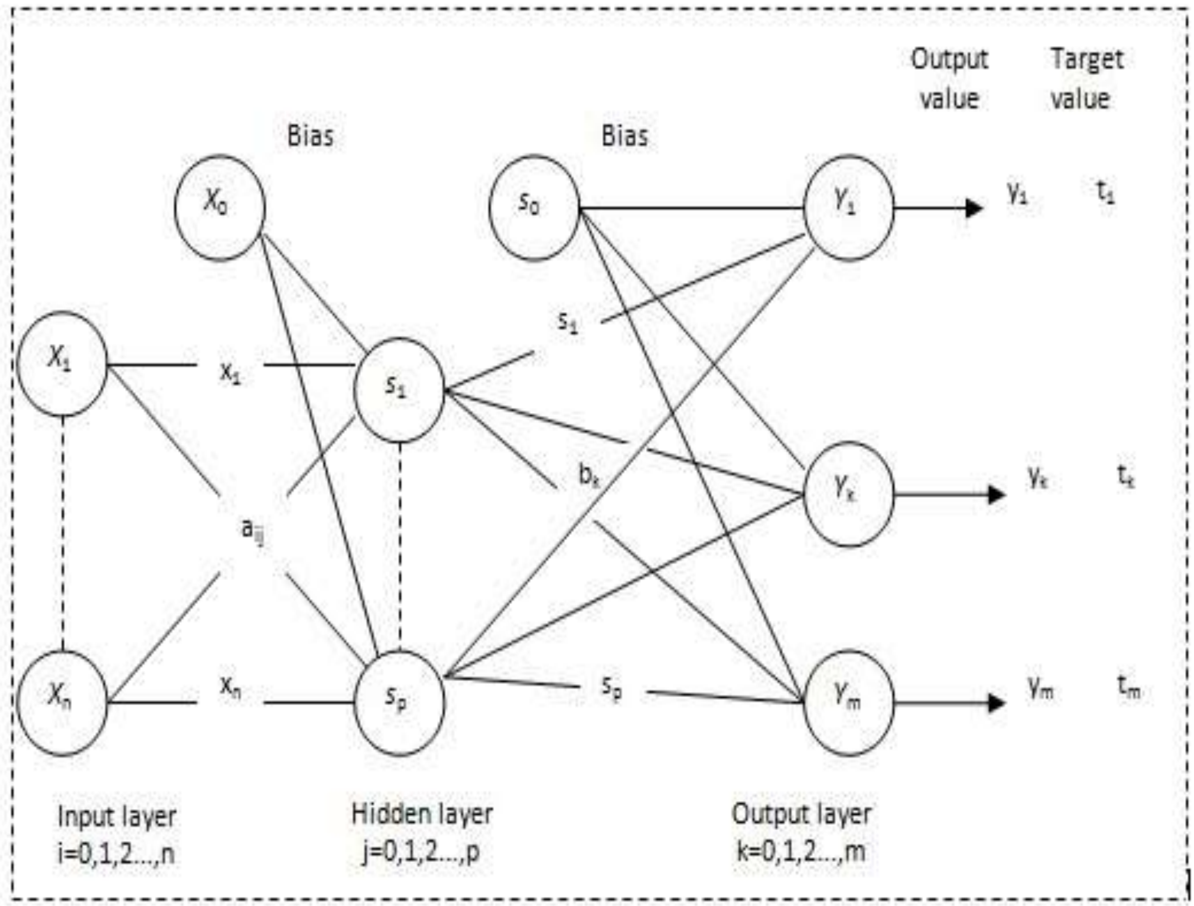


Figure 5: Multilayer neural network

Cybenko and Hornik (1989) have shown that neural network with one hidden layer and with sigmoid activation function can approximate every function with desirable accuracy.

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

More complex neural networks can have more hidden layers. The first layer (input) is the only that receives data (signals) from environment and transmits them into the following layer (hidden layer).
Input layer: every neuron gets only one input directly outside neural network.
Hidden layer: links input and output layer. Hidden layer processes data and transmits them to output layer of a network.
Output layer: output from every neuron is directed into environment. Final results of processing are at the output layer.
Architecture of neural networks with one input, one hidden, and one output layer is the commonest. Every neuron of input layer is linked to every neuron of hidden layer, and every neuron of hidden layer is linked to every neuron of output layer.
According to information direction (signal) spreading over the network, neural networks are divided into feed-forward and feedback ones.
Higher layer does not return data (signal) to lower layer in feed-forward neural network. In feedback neural network higher layer returns data (signal) to lower layer.
Feedback neural network operational manner, learning manner, and data and signal spreading manners will be presented further in this paper. The following markers will be used to explain its operation manner:

- α - learning rate,
- x - training vector $x=(x_1,x_2,x_3,...,x_n)$,
- t - target vector $t=(t_1,t_2,t_3,...,t_m)$,
- x_i - output from i-input neuron,
- z_j - j-hidden neuron output,
- y_k - k-output neuron output,
- t_k - k-input neuron target value.

Total input value of j-neuron at hidden layer is as follows: $sin_j = \sum_{i=0}^{n} x_i a_{ij}$.

Output from j-neuron with hidden layer is $s_j = f(sin_j)$.

Total input of k-neuron at output layer is $yin_k = \sum_{j=0}^{p} s_j b_{jk}$.

Output from k-neuron at output layer is $y_k = f(yin_k)$.

It is important to perceive the following rules:

1. There are not link between neurons that belong to the same level (layer),
2. Neuron at one layer is linked **only** to neurons at the following layer neuron (feed-forward),
3. Layers' skipping is not allowed.

Error function of multilayer neural network

Error function of multilayer neural network is sum of deviation squares of all outputs in relation to output target values. Neural networks error is equal to $E = \frac{1}{2}\sum_k (t_k - y_k)^2$. Neural network error can be changed applying weighting factors between input and hidden layers (v_{ij}), and applying weighting factors between hidden and output layers (w_{jk}).

Alternation of error function (E) alternating weighting factors between input and hidden layer

Error function alternation (E) is determined by derivation that depends of weighting factors a_{ij} changes:

$$\frac{\partial E}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}}\{\frac{1}{2}\sum_k (t_k - y_k)^2\}$$

If an error from one k-neuron is marked as u_k:
It is easy to perceive that expression in parenthesis does not contain indexes i and j. Therefore it can be written:

$$\frac{\partial E}{\partial a_{ij}} = \sum_k u_k \frac{\partial u_k}{\partial a_{ij}} = \sum_k (t_k - y_k)\frac{\partial (t_k - y_k)}{\partial a_{ij}} = \sum_k (t_k - y_k)\frac{\partial y_k}{\partial a_{ij}}$$

Since $y_k = f(yin_k)$ the following can be written:

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

$$\frac{\partial E}{\partial a_{ij}} = \sum_k (t_k - y_k)\frac{\partial y_k}{\partial a_{ij}} = \sum_k (t_k - y_k)\frac{\partial y_k}{\partial yin_k}\frac{\partial yin_k}{\partial a_{ij}} = \sum_k (t_k - y_k) f(yin_k)\frac{\partial yin_k}{\partial a_{ij}}$$

If $\delta_k = \sum_k (t_k - y_k) f'(yin_k)$ then the previous expression can be written in shorter form as follows: $\frac{\partial E}{\partial a_{ij}} = \sum_k \delta_k \frac{\partial yin_k}{\partial a_{ij}}$. Since: $yin_k = \sum_j s_j b_{jk}$ can be written as follows: $\frac{\partial yin_k}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}}(s_0 w_{0k} + s_1 w_{1k} + s_2 w_{2k} + \ldots)$

can be written as follows a_{ij} do not affect (they are not mutually dependent) weighting factors b_{jk}, then the following formula can be applied: $\frac{\partial yin_k}{\partial a_{ij}} = b_{jk}\frac{\partial s_j}{\partial a_{ij}}$. Therefore: $\frac{\partial E}{\partial a_{ij}} = \sum_k \delta_k b_{jk}\frac{\partial s_j}{\partial a_{ij}}$.

Since: $z_j = f(sin_j)$, then the following formula can be applied:

$$\frac{\partial s_j}{\partial a_{ij}} = \frac{\partial f(sin_j)}{\partial a_{ij}} = \frac{\partial f(sin_j)}{\partial (sin_j)}\frac{\partial (sin_j)}{\partial a_{ij}} = f'(sn_j)\frac{\partial (sin_j)}{\partial a_{ij}}$$

$$sin_j = \sum_i x_i a_{ij} = x_0 a_{0j} + x_1 a_{1j} + x_2 a_{2j} + \ldots$$

therefore $\frac{\partial sin_j}{\partial a_{ij}} = x_i$. $\frac{\partial E}{\partial a_{ij}} = \sum_k \delta_k b_{jk} f'(sin_j) x_i$. If we introduce exchange: $\delta_j = \sum_k \delta_k b_{jk} f'(sin_j) = f'(sin_j)\sum_k \delta_k b_{jk}$

then: $\frac{\partial E}{\partial a_{ij}} = -\delta_j x_i$

Finally weighting factors changes between input and output layers of neural network can be expressed as follows:

$$\Delta a_{ij} = -\alpha\frac{\partial E}{\partial a_{ij}} = \alpha\delta_j x_i$$

Similarly error function change (E) can be shown applying weighting factors between hidden and output layers.
Finally weighting factors changes between input and hidden layers and weighting factors between hidden and output layers can be written as follows:

$$a_{ij}(new) = a_{ij}(old) + \Delta a_{ij}$$

$$b_{jk}(new) = b_{jk}(old) + \Delta w_{jk}$$

Algorithm backpropagation
Algorithm backpropagation starts from error function and adapts weighting factors between hidden and output layer, and then between input and hidden layer. Algorithm backpropagation steps are shown in the following table:

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

Step	Action
0:	Initialisation of weighting factors (random generating - weighting factors between -0.5 and 0.5)
1:	**WHILE** (condition $E > \varepsilon$)
	{
2:	**FOR** (Every input pattern)
	{
3:	enter x_i for $i = 1,2,3\ldots,n$
4:	$ain_j = \sum_{i=0}^{n} x_i a_{ij}$ for j=1,2,3,...,p. $s_j = f(sin_j)$
5:	$yin_k = \sum_{j=0}^{p} s_j b_{jk}$ $y_k = f(yin_k)$ for $k = 1,2\ldots,m$
6:	$\delta_k = \sum_k (t_k - y_k) f'(yin_k)$ $\Delta b_{jk} = \alpha \delta_k s_j$
7:	$\delta_j = \sum_k \delta_k b_{jk} f'(sin_j) = f'(sin_j) \sum_{k=1}^{m} \delta_k w_{jk}$ $\Delta a_{ij} = \alpha \frac{\partial E}{\partial a_{ij}} = \alpha \delta_j x_i$
8:	$b_{jk}(new) = b_{jk}(old) + \Delta b_{jk}$ $a_{ij}(novi) = a_{ij}(stari) + \Delta u_{ij}$
	} /* end of FOR loop * /
	} / * end of WHILE loop * /

Nakon što se ažuriraju težinski faktori potrebno je testirati izlazne neuronske mreže s postavljenim ciljevima. U sljedećoj tablici su prikazani koraci izračunavanja izlaza iz neuronske mreže:

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

Step	Action
0:	Initialize weighting factors form training algorithm
1:	**FOR** (for every input vector)
	{
2:	enter x_i for $i = 1,2,3...,n$
3:	$sn_j = \sum_{i=0}^{n} x_i v_{ij}$ for j=1,2,3,...,p $s_j = f(sin_j)$
4:	$yin_k = \sum_{j=0}^{p} s_j b_{jk}$ $y_k = f(yin_k)$ for k=1,2,3,...,m
	}

Backpropagation algorithm modifications - momentum

Weighting factors can be changed applying momentum. It will be marked with μ as it follows:

$$\Delta b_{jk}(new) = \alpha\delta_k s_j + \mu\Delta w_{jk}(old)$$

$$\Delta a_{ij}(new) = \alpha\delta_j x_i + \mu\Delta u_{ij}(old)$$

If $\mu = 0$ then $\Delta b_{jk}(new) = \alpha\delta_k s_j$ and $\Delta a_{ij}(new) = \alpha\delta_j x_i$ as in previous examples. Momentum value is between 0 and 1 i.e. $0 \leq \mu \leq 1$

If weighting factors changes are positive then momentum accelerates those changes more. Momentum use decreases probability of neural network input into local minimum (extreme) during network training. Momentum produces changes increasing error so that neural network continues trainings (learning). Typical values for $\alpha < 0.1$, and for momentum $0.5 < \mu < 0.99$.

Research methodology

It is possible to identify key factors by careful creation of a questionnaire, which contains important questions related to purchase decision and socio-demographic characteristics of a customer. It is of a great importance to ensure sample representativeness. Stratified random sample has been chosen. Stratification is based on characteristics of gender, age, education, average monthly income, and geographic disposition of youth in the Federation of Bosnia and Herzegovina, so the proportions of chosen characteristics in sample "respond" to proportions of those characteristics in population (in basic set). There is an array of closed type questions (with pre-offered answers) and it is consisted of four parts as an instrument to conduct a questionnaire:

1. introduction,
2. questions about socio-demographic characteristics of respondents (taking care not to ask for information that endanger respondents' anonymity),
3. questions about certain brand purchase (bottled drinking water),

The second step is selection of respondents from basic set by random selection method respecting defined stratum. 500 questionnaires were sent and respondents filled in 150 of them.

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

The research results

The methodology in this paper is based on induction rules and knowledge base built by using the R programming language. At the same time R denotes three things: data analysis software, calculator and programming language.[3]

Data set

Data set is the result of pre-processing activity followed by receiving completed surveys from respondents. The questionnaire is consisted of the questions about the customers' socio-demographic characteristics in the market of drinking water: gender, age, education, and household revenues

Respondent	Gender	Age	Education	Revenues	Brand	Purchase
1	male	31	bachelor	2500	-	does not buy
2	female	27	master	950	Jana	buys
...	.....	...	...	...	....	...
95	female	26	bachelor	1450	Olimpija	buys
96	male	41	high school	1250	Biser	buys
...	...	...	...	...	...	...
150	female	19	high school	950	...	does not buy

Table 2: Dataset with customers' socio-demographic characteristics

[3] R costs nothing and is completely free. To install R on your computer visit the site http://cran.r-project.org/mirrors.html and choose the nearest mirror.

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

Data analysis in the R language is relatively simple and easy. So to display data in the form of data cube is sufficient following sequence of statements stored in script file named AgeEdRe.R :

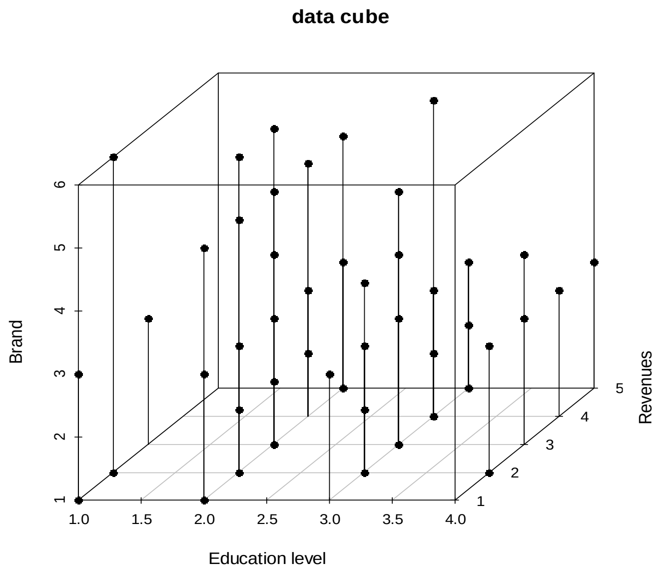
```
library(RODBC)
library(sqldf)
library(tcltk)
w<-odbcConnectAccess("C:\\Radovi2014\\DAAAM\\wD")
qrNN=sqlQuery(w, "SELECT * FROM wNN")
join_prvi= "SELECT Gender, Age, Education, Revenues, Buy FROM qrNN"
nnData<-sqldf(join_prvi)
library(scatterplot3d)
X<- nnData [,3]
Y<- nnData [,4]
Z= nnData [,5]
```

scatterplot3d(X,Y,Z,pch=16,type="h",main="data cube",xlab="Education level", ylab="Revenues",

The script is executed using the following statement:

>source(„C:/ AgeEdRe.R “)

The result is data cube where X-axis represents education level[4], Y-axis represents level of household revenues,



and Z-axis represents brand

Figure 8. Data cube

Therefore dataset is divided into two parts: learning part (70% of total number of dataset patterns) and testing part (rest 30% patterns). The first step in neural network development in the R language is to prepare data. The simplest structure in the R language is vector with numerical values. It is data structure that contains array of numbers (numerical values). Data in nnData dataset are scaled and therefore they do not have to be normalised.

[4] All attributes (gender, age, education level, household revenues, brand) are represented numerically: gender: 1-male, 2-female; age: 1- less then 25 years, 2 - between 25 and 35 years, 3 - between 35 and 45 years, 4–more then 45 years; education level: 1-elementary school, 2-high school, 3-bachelor, 4-master or PhD degree; level of household revenues: 1- revenues up to 500 BAM, 2- from 501 BAM to 1000 BAM, 3 - 1001 BAM to 1500 BAM, 4- 1501 BAM to 2000 BAM and 5 – more than 2001 BAM; brand: 1- Jana, 2- Studena, 3 – Olimpija, 4- Biser, 5- some other.

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

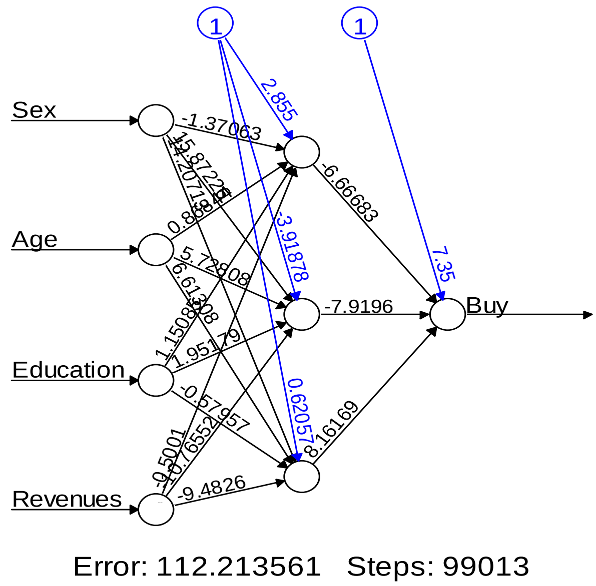
Multilayer neural network architecture for of customers' classification and segmentation is made of three layers. There are four input neurons at the input layer, four at hidden layer (one is hidden neuron and its value is 1), and at output layer there is only one neuron. Naturally, selection of appropriate number of neurons at hidden level is often based on trial and error method. It is suggested to choose total number of neurons at hidden level using $p = \sqrt{n * m}$ expression where ***n*** is number of neurons at input layer, ***m*** is number of neurons at output layer, and ***p*** is number of neurons at hidden layer[5].

It is sufficient to write the following command (before that download neuralnet package:

library(neuralnet)

>nn=neuralnet(Buy~Gender+Age+Education+Revenues, data=nnTraining, hidden=3, learningrate=0.001, algorithm="backprop", err.fct = "sse", linear.output=TRUE)

>plot(nn)



Elements of Buy vector are the values of output neuron, and values of Gender, Age, Education, and Revenues vector are the values at input layer of multilayer neural network. Parameter hidden=3 symbolizes that there are three neurons at hidden layer. Weighting factors between neurons at input and hidden, and hidden and input layers are presented in the following table:

> nn$result

> nn$result

	1
error	112.213560897953

[5] If data are not scaled they have to normalize. Data are normalized i.e. all values in row of nnData table are numerical ones in closed interval [0,1]. In the given example for normalization it would be enough to write the following array of commands in the R language:

The R langugage command that normalizes value of nnData dataset columns is the following:

>data <- **nnData**[, c('Gender', 'Age', 'Education', 'Revenues','Buy')]

>nnData.norm<-cbind((nnData[,1] - min(data[, 'Gender']))/(max(data[, 'Gender'])-min(data[, 'Gender'])),(nnData[,2] - min(data[,'Age']))/(max(data[, 'Age'])-min(data[, 'Age'])),(nnData[,3] - min(data[,'Education']))/(max(data[, 'Education'])-min(data[, 'Education'])),(nnData[,4] - min(data[,'Revenues']))/(max(data[,'Revenues'])-min(data[,'Revenues'])),(nnData[,5] - min(data[,'Buy']))/(max(data[,'Buy'])-min(data[,'Buy'])))

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

```
reached.threshold            0.009616580556
steps                    99013.000000000000
Intercept.to.1layhid1        2.855000544651
Gender.to.1layhid1          -1.370633735509
Age.to.1layhid1              0.868443638831
Education.to.1layhid1        1.150850059795
Revenues.to.1layhid1        -0.500097516878
Intercept.to.1layhid2       -3.918777822898
Gender.to.1layhid2          15.872264317892
Age.to.1layhid2              5.728084870455
Education.to.1layhid2        1.951785277926
Revenues.to.1layhid2       -10.765519836023
Intercept.to.1layhid3        0.620573744502
Gender.to.1layhid3          14.207126052849
Age.to.1layhid3              6.613076001803
Education.to.1layhid3       -0.579574806157
Revenues.to.1layhid3        -9.482604350210
Intercept.to.Buy             7.350002608578
1layhid.1.to.Buy            -6.666825664498
1layhid.2.to.Buy            -7.919603080376
1layhid.3.to.Buy             8.161687012768
```

Neural network can be used to classify customers and to estimate product purchase that has market value for that customer. Namely, it is enough to enter scaled values for gender, age, education, and revenues for a certain customer (his/her socio-demographic characteristics), and neural network will forecast numerical value of a product code at output neuron. The function compute() enables that in the R language. If data on socio-demographic characteristics of the customers are given in the following table:

Gender	Age	Education	Revenues	Buy
2	3	3	5	-
2	3	2	4	-

```
> n.izlaz = compute (nn, covariate = matrix (c (2, 3, 3, 5, 2, 3, 2, 4  ), byrow = TRUE, ncol = 4))
> n.izlaz
$neurons
$neurons[[1]]
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    3    5
[2,]    1    2    3    2    4

$neurons[[2]]
     [,1]         [,2]          [,3]         [,4]
[1,]    1 0.9751837343 0.04916129002 0.4310196266
[2,]    1 0.9534861172 0.99713265310 0.9999436846

$net.result
            [,1]
[1,] 3.977132046
[2,] 1.257609445
```

Neural network result can be presented in the following table:

INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

Gender	Age	Education	Revenues	Buy
2	3	3	5	3.977132046
2	3	2	4	1.257609445

Neural network forecasts that a customer whose scaled values of socio-demographic characteristics are in matrix row (2, 3, 3, 5) will most probably purchase product no. 4 (i.e. drinking water *Biser*), while customer (2, 3, 2, 4) will most probably purchase product no.2 (i.e. drinking water *Jana*).

Neural network to classify purchase forecast for certain brand based on customers' socio-demographic characteristic behaves dynamically since every customer, depending on his/her socio-demographic characteristics forecasts product selection that has the highest market value for him/her.

Conclusion

This paper presents complexity of customer management and their behaviour forecast in the market. Customer behaviour is conditioned by great number of factors among which the most important are socio-demographic ones. They determine product market value for a certain customer. Namely, a customer verifies product market value at the moment of its purchase. At that moment customer is ready to exchange money for product usage value that satisfies his/her need. The authors of this paper assume that consumers follow certain behaviour pattern depending on their socio-demographic characteristics. Besides, the authors assume that it is possible to discover and foresee customer behaviour applying method that transforms data into business intelligence. Multilayer neural networks have a special place among those methods. Multilayer neural network has proved strong implementation power and satisfactory accuracy in forecasting product selection based o socio-demographic characteristics of customers. In methodological sense this paper presents all steps of knowledge extraction of customer behaviour (in marketing), which is based on data collected using questionnaire. The R language has proved satisfactory development and application power in developing neural network that detects and forecasts product market value for a customer with certain socio-demographic characteristics.

References

1. Abbasi, Z., (2007), A review of models of knowledge management implementation in Organizations, Knowledge Management Conference.
2. Aaker A., David, (1991), Managing brand equity, The Free press, New York.
3. Bijakšić Martinović, S., Bevanda A., Markić, B., (2014), Marketing i metrika, marketinški splet, podaci, mjerila, Napredak Glavna podružnica Mostar, Mostar.
4. Kantaradžić, M., (2003) Data mining, Concepts, Models, Methods, and Algorithms, Wiley-Interscience.
5. Kotler P., Wong V., Saunders J. & Armstrong G., (2006),Osnove marketinga, Zagreb.
6. Markic, B. (2010), Marketing Intelligent Systems Using Soft Computing Studies in Fuzziness and Soft Computing, Volume 258/2010, Springer-Verlag Berlin Heidelberg 2010., pp 79-111.
7. Markić, B., (2014), Sustavi potpore odlučivanju i poslovna inteligencije, podaci, modeli i algoritmi, Napredak Glavna podružnica Mostar, Mostar.
8. Markic, B. (2011), Customer segmentation by integrating unsupervised and supervised learning, Proceeding from international conference Economic Theory and Practice: Meeting the New Challenges, Faculty of Economics University of Mostar, october 2011, Mostar.
9. Vranešević, T., (2007), Upravljanje markama, Accent, Zagreb.