Using Genetic Algorithms in Secured Business Intelligence Mobile Applications

Silvia TRIF
Academy of Economic Studies, Bucharest, Romania
silviatrif@gmail.com

The paper aims to assess the use of genetic algorithms for training neural networks used in secured Business Intelligence Mobile Applications. A comparison is made between classic back-propagation method and a genetic algorithm based training. The design of these algorithms is presented. A comparative study is realized for determining the better way of training neural networks, from the point of view of time and memory usage. The results show that genetic algorithms based training offer better performance and memory usage than back-propagation and they are fit to be implemented on mobile devices.
Keywords: *Genetic Algorithm, Mobile Applications, Back-propagation, Business Intelligence, Security*

1 Introduction

Mobile applications security for Business Intelligence is an area located at the intersection of the following domains: mobile applications, business intelligence and information security.

Nowadays, mobile applications are in a continuous developing process. As the PC applications use neural networks, these are used also in mobile applications development, for predictions, figure recognitions, pattern recognitions and not only.

Artificial neural networks are computational networks which simulate the networks of nerve cell (neurons), simulating the properties of the neurons and the interaction between them. These networks help in solving the problems, general problems, not specific ones; the neural network using supervised learning, input datasets, learn and then it think by itself, as a real biological brain. It uses simple operations, as additions, subtraction, multiplicity and fundamental logic elements to perform complex problems.

Neural networks have an important role in the applications developing process, having the following advantages [1], [2]:

- mapping inputs signals to desired response - supervised learning; it involves the modification of the synaptic weights of a neural network by applying a set of training examples; each training example consist of a unique input signal and a corresponding target response; the difference between the actual response and the desired response should be minimized;
- adaptively - adapt weights to environment and retrained easily; the neural network can be retrained and used in different cases, not only in a specific case;
- nonlinearly - inherently nonlinear signals; the neurons of a network can be linear or nonlinear; the network formed by the interconnection between the nonlinear neurons is a nonlinear one;
- evidential response - confidence level improves classification
- is not need to know the relation between the dependent and independent variables.

Artificial neural network are composed of interconnecting artificial neurons (which simulates the biologically neurons).

Artificial neural networks are like nonlinear models to estimate levels of a dependent variable depending of a number of independent variables [3]. The results obtained using neural networks are often better than those obtained by applying traditional methods. This ability is manifest especially in data sets containing complex and difficult to identify links between independent and dependent variables.

From the structural point of view, a neural network is composed of layers of interconnected neurons. The number of neurons in input layer corresponds to the number of independent variables that give inputs. The output layer consists of neurons that provide network output. In the following chapters are presented the backpropagation neural network and the genetic algorithm.

2 Business Intelligence mobile applications security

Mobile applications are software products developed to run on portable devices, equipped with telephone processing capabilities.

In [4] a mobile application is defined as a computational entity that uses more mobile applications services. A mobile application service is defined as a computational entity that uses at least one model functional component over specified interfaces [4]. There are two types of mobile applications [4]:

– the one developed by the user for a specific purpose;
– generic user agents, the one which are preinstalled on the mobile device.

Depending on the application architecture type, there are various problems to deal with. On the network architecture applications, there should be developed and used a secure protocol and standard for sending and receiving encrypted messages. For standalone applications, the security aspects refer only at user privacy and database integrity. These aspects are treated in [5].

In the process of developing mobile applications, there are limits given by the device for which the product is made for, as: memory, processing power, display, low to moderate complexity, small to medium applications, low to medium assignment speed.

The quality of applications developed is computed depending on the following characteristics: reliability, complexity, usability, portability and maintainability, which are detailed in [6].

A complex approach of mobile applications security is realized in [7]. In this publication are presented the main aspects for mobile applications security platforms, regarding: the device physical security, removable media security, multifactor authentication, secure support for multiple users, the operating system security, viruses, and attacks specific for web mobile applications.

In [8] are presented cryptographic elements and algorithms, dispersion functions, public key, digital signatures, digital certificates. Security standards and protocols are presented.

In [9] is presented an architecture for secured mobile applications, architecture based on Bluethooth communication. Data confidentiality, integrity and authenticity are analyzed and is proposed an authentication solution using cryptographic techniques.

Business Intelligence concept gives a management of business operations, following finished and in process operations and predictions for the future operations. In [10] are treated the following aspects:

– the importance of Business Intelligence in a world based on information
– phases are proposed in the Bussiness process
– strategies for selecting business intelligence.

The business intelligence can be optimized by improving data mining process and the process of taking decisions.

Neural networks have an important role in the business intelligence mobile application developing process; using these networks, the decisions are taken easier and at a higher level of confidence.

3 Genetic algorithm for neuronal networks training

Using artificial intelligence in mobile applications security components is assured an augmented independency, the errors given by the interaction device-user being minimized. Using neural networks with mobile applications have limits determined by the device characteristics: memory used, processing power and the processing speed.

There are presented two different types of learning, using neural networks: backpropagation and genetic algorithm. Using these two, in the same conditions, is aimed to be identified the best method, from the points of view of time and memory usage.

Backpropagation is a supervised learning method, based on given inputs and outputs for learning.

A Backpropagation neural network is a multilevel feed forward network, having an input layer, an output layer and one or more intermediate layers, formed of hidden neurons. The layers are interconnected. The input layer's neurons are interconnected, having random intensities, between 0 and 1. The next layer's nodes receive the input from the previous layer and compute their own outputs; these outputs represent inputs data for the next layer. This is the reason why these neural networks are called feed-forward networks; the values are propagated forward until the final layer is touched. The final output is computed, it is compared with the desired output, computing the error for every node of the output layer. The error is back-propagated and a new initial value is determined for the input layer's neurons intensity. The learning process is computed when the total error is less than a given tolerance level.

A neural network consists in the following types of layer:

– first layer is the input layer- the initial input,
– hidden layers: the ones which propagate the previous layer's outputs to the next layer and back propagate the following layer's error to the previous layer. These are a bridge between the input and the output layer;
– output layer- the final layer.

In Figure 1 is represented the back-propagation neural network.

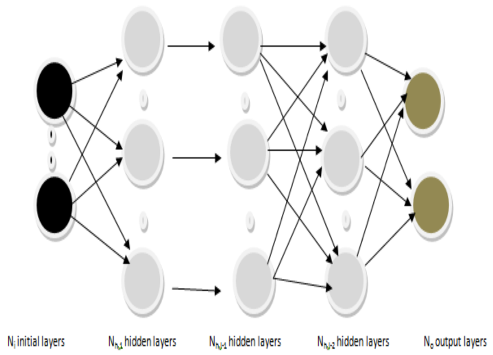

Fig. 1. Backpropagation representation [11]

The training process involves two phases, which are continued until the square of output errors reaches an acceptable value:

– the inputs are propagated forward to compute the outputs for each output node; each outputs are subtracted from its desired output, causing an error for every output node
– each output errors is passed backward and the weights are fixed .

The source code is written using Visual Studio 2010 Express for Windows Phone, using C# language.

The following source code presents a method of determining the time needed to learn and the memory usage, when back-propagation is used. The system timer is read before the call to the learning method and immediately after the learning stopped. Also, the memory usage is recorded.

```
long x= DateTime.Now.Ticks;
vectm[0] = (long)
DeviceExtendedProperties.GetValue
("ApplicationCurrentMemoryUsage");
vectmpeak[0] = (long)
DeviceExtendedProperties.GetValue("A
pplicationPeakMemoryUsage");
                    …

nn.LearningAlg.Learn(inputs,
outputs);
 vectm[3] = (long)
DeviceExtendedProperties.GetValue("A
pplicationCurrentMemoryUsage");
 vectmpeak[3] = (long)
DeviceExtendedProperties.GetValue("A
pplicationPeakMemoryUsage");
…
 long xf = DateTime.Now.Ticks;
long tickf =( xf – x);
```

Genetic algorithm represents a heuristic that mimics the process of natural evolution. Genetic algorithms generate solutions to optimization problems using techniques inspired by natural evolution: inheritance, mutation, selection and crossover [12].
This type of algorithm has numerous applications in economics: the symbolic regression that has evolved with genetic programming [3]; starting with a dataset where are known the dependent and independent variables, an initial chromosomes population is generated, which is than replicated using the genetic operators. Other applications are related to any optimization problems where the solution may be encoded as a linear structure and a set of rules are applied in order to evaluate that solution. This linear structure represents a chromosome and it is an abstraction of a concrete element from the real world. The principles of genetic evolution aim to improve the solution in an iterative process where an initial random population of chromosomes is subject to various genetic operators. These operators are:

– evaluation; each chromosome is evaluated in terms of how good the solution it encodes is; the population is sorted according to the performance of each individual;
– selection; a proportion of the initial population is selected to move to the next generation; better individuals have better chances to be selected;
– crossover; randomly chosen chromosomes exchange genetic information giving birth to new chromosomes;
– mutation; randomly chosen chromosomes suffer random changes in their contents; mutation is an operation that occurs with low probability as it usually destroys the individual but sometimes mutation produces a better individual otherwise impossible to evolve from existing population.

A **chromosome** is a fixed length linear structure, formed by the genes and symbols. The chromosome role is to encode the characteristics of an individual.
The gene role is to encode a part of the characteristics of an individual. The symbols encode the basic elements that define the individual.
In the case of neural networks, the chromosome must encode the network structure. The network is made up of layers. Layers contain neurons. Neurons between consecutive layers are connected and the power of the connection is given by a parameter called weight. Neurons, internally, have thresholds. In dynamics, the network propagates the outputs of the neurons. The static structure is defined by the weights and the thresholds of the neurons and the order of the connections.
In this case, it is possible to encode the network as a linear structure, where each position is mapped to the corresponding weight or threshold in the network.
Supposing the next neural network, having one input, 2 hidden layers and one output the visual representation is given in Figure 2. There are highlighted the neurons, the weights and the thresholds.

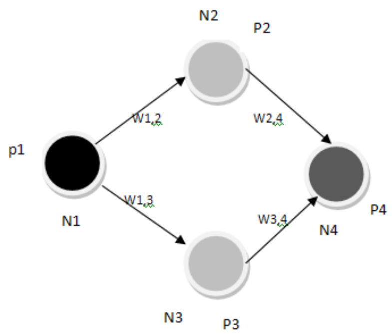

Fig. 2 Neural Network with 1 input, 2 hidden layers and 1 output

In Figure 2, the following notations are made:

– N1 (input), N2, N3 (hidden), N4 (output) neurons;
– p_1, p_2, p_3 neural network thresholds
– $w_{i,j}$, i,j=1,3 the network weights.

The relationship between the neural network and the genetic algorithm is given by the fact that genetic chromosome is an array formed by the network thresholds and weights. A chromosome that encodes the network from figure 2 is:

$CR=p_1,p_2,p_3,p_4,w_{1,2},w_{1,3.},w_{2,4}, w_{3,4}$

Considering this chromosome structure, genetic operations are possible.
Given the chromosomes CR1 and CR2, described as:
CR1

0	1	2	3	4	5	6	7
0.3	0.7	0.6	0.2	0.1	0.9	0.9	0.4

CR2

0	1	2	3	4	5	6	7
0.5	0.3	0.8	0.3	0.1	0.7	0.1	0.8

the crossover operation in position 3 leads to obtaining the new chromosome CR1' and CR2', formed by the combination between the chromosome CR1 and CR2:
CR1'

0	1	2	3	4	5	6	7
0.3	0.7	0.6	0.3	0.1	0.7	0.1	0.8

CR2'

0	1	2	3	4	5	6	7
0.5	0.3	0.8	0.2	0.1	0.9	0.9	0.4

which are two new chromosomes.
In Figure 3 is presented the neural network corresponding to chromosome CR2' with highlight on the values of the weights and thresholds.

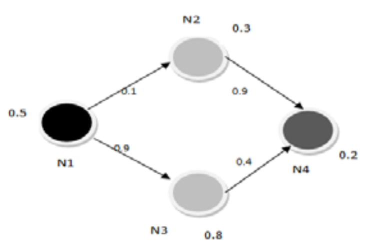

Fig. 3. Neural network corresponding to chromosome CR2'

The mutation operation performed against CR1 chromosome, in position 2 changes the weight of the threshold encoded by that position to a random value, obtaining CR1''.
CR1''

0	1	2	3	4	5	6	7
0.3	0.7	0.4	0.3	0.1	0.7	0.1	0.8

In the design phase of the algorithm, is made a transformation from neural network to a chromosome and at run time phase a transformation is made from the chromosome to the network, which is the encoded individual. It is needed only one network when the evaluation of the chromosome occurs.
The source code used in the genetic algorithm learning for computing the time and memory used, is the same as the code used for back-propagation, only the learning method is different.

```
…
nng.LearningAlg = new
WPNeuralNetwork.
GeneticLearningAlgorithm (nng);
        …
nng.LearningAlg.Learn(inputs,
outputs);
…
```

Using this representation the algorithm is implemented in source code and tested to verify the hypothesis that genetic algorithm based training is appropriated to be used on mobile devices.

4 Experimental results

A common application of business intelligence is the classification of the elements from a set. A set of elements having 4 binary characteristics and 2 classes are considered. It is desired that a neural network learns the training samples and later to classify correctly given inputs.
The inputs training sets and the desired outputs are presented in Table 1.

Table 1. The training set and desired outputs

Inputs				Outputs
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0

1	0	0	1	1
1	0	1	0	0
0	1	1	1	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

Next, both back-propagation and genetic algorithm based learning are used to learn the training set.
The neural network used has 4 inputs, 7 hidden neurons and 1 output.

```
public WPNeuralNetwork
.NeuralNetwork nn=new
WPNeuralNetwork .NeuralNetwork(4,new
int[]{7,1});
public WPNeuralNetwork.NeuralNetwork
 nng = new
WPNeuralNetwork.NeuralNetwork(4, new
int[] {7, 1 });
```

Backpropagation algorithm steps are [13]:

- initialize the weights in the network (often randomly)
- for each example ex_train in the training set DO-UNTIL all examples classified correctly or stopping criterion satisfied
  - the forward pass: compute O(network, ex_train) - neural network output
  - the desired output is given- T-teacher output for ex_train
  - calculate the error at the output unit (T-O)
  - compute delta_wh for all weights from hidden layer to output layer; backword pass
  - compute delta_wi for weights from input layer to hidden layer; backward pass continued
  - update the weights in the network
- return the network

The steps followed in the genetic algorithm are:

- an initial population of random chromosomes is created; the chromosomes encode the network score;
- the scores of every chromosome are transferred into a network and tested
- the chromosomes are ordered according to their performance
- the best chromosomes are chosen
- these chromosomes are passed in the next generation
- genetic operators are applied

These steps repeat until a solution is found or until the maximum numbers of period is passed.
In [2] are presented the following steps:

- at the beginning a large population of random chromosomes is created; each one, when decoded will represent a different solution to the current problem;
- the following steps are repeated until a solution is found
  - each chromosome is tested for seeing how good it is in solving the problem and a fitness score is assigned according to the chromosome performance in solving the problem;
  - two members from the current population are selected; The chance of being selected is proportional to the chromosomes fitness;
  - depending on the crossover rate crossover the bits from each chosen chromosome at a randomly chosen point
  - step through the chosen chromosomes bits and flip dependent on the mutation rate;
  - repeat step 2, 3, 4 until a new population of initial number of chromosomes members has been created.

In Figures 4 and 5 are presented the outputs of the learning process, at two periods of time, using backpropagation and genetic algorithm.

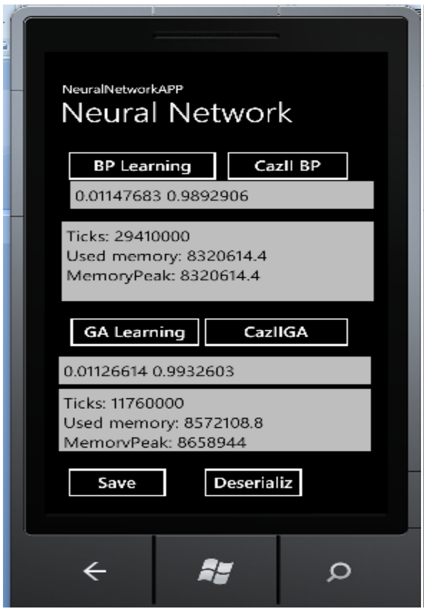

Fig. 4. Results after first run

Fig. 5. Results after the second run

In table 2 are presented the characteristics followed: time and memory used by the neural network to learn. The neural network has 4 inputs, 7 hidden neurons, 1 output and 16 testing sets.

Table 2. Back-propagation and Genetic algorithm comparative results

Method	**BackPropagation**	**Genetic Algorithm**	**Time moment**
Time (ticks)	13770000	20540000	1
Memory usage(bytes)	8556544	8345190.4	
MemoryPeak(bytes)	8556544	8430387.2	
Time (ticks)	16280000	1100000	2
Memory usage(bytes)	8781824	8626176	
MemoryPeak(bytes)	8781824	8757248	
Time (ticks)	16820000	80000	3
Memory usage(bytes)	8847360	8634368	
MemoryPeak(bytes)	8847360	8757248	
Time (ticks)	15200000	1220000	4
Memory usage(bytes)	8855552	8638464	
MemoryPeak(bytes)	8855552	8757248	
Time (ticks)	14360000	990000	5
Memory usage(bytes)	8881766.4	8638464	
MemoryPeak(bytes)	8881766.4	8757248	
Medium time used	15.286.000,00	4.566.000,00	
Medium memory usage	8.784.609,28	6.851.297,28	

After 5 run times, the medium values for time consumption and memory usage are computed. From table 1, from the time and memory usage points of view, results the fact that the genetic algorithm is better than back-propagation method.

In Table 3 are presented the characteristics followed: time and memory used by the neural network to learn. The neural network has 10 inputs, 20 hidden neurons, 1 output and 200 testing sets.

Table 3. Back-propagation and genetic algorithm comparative results

Method	BackPropagation	Genetic Algorithm	Time moment
Time (ticks)	802010000	3040140000	1
Memory usage(bytes)	8319795.2	8602419.2	
MemoryPeak(bytes)	8319795.2	8602419.2	
Time (ticks)	562310000	3807650000	2
Memory usage(bytes)	8577024	9109504	
MemoryPeak(bytes)	8577024	9109504	
Time (ticks)	530160000	2978220000	3
Memory usage(bytes)	8650752	9121792	
MemoryPeak(bytes)	8650752	9121792	
Time (ticks)	651110000	3211020000	4
Memory usage(bytes)	8676966.4	9121792	
MemoryPeak(bytes)	8676966.4	9121792	
Time (ticks)	836300000	2780740000	5
Memory usage(bytes)	8742502.4	9121792	
MemoryPeak(bytes)	8742502.4	9121792	
Medium time used	676.378.000,00	3.127.554.000,00	
Medium memory usage	8.593.408,00	9.015.459,84	

After 5 run times, the medium values for time consumption and memory usage are computed. From Table 3, from the time and memory usage points of view, results the fact that the back-propagation method is better than genetic algorithm. From the cases treated, result the fact that choosing an algorithm to learn depends on the number of inputs, hidden neurons and outputs; if the numbers of inputs, hidden neurons and training sets are bigger, the back-propagation is better to be used.

5 Genetic algorithms security

Business intelligence applications are subject to security issues. It is desired to protect sensitive information from unauthorized access. In the considered application sensitive information includes:

- the data sets used in the application
- the structure of the neural network: number of inputs, number of outputs, number of hidden neurons, activation functions, weights, thresholds; these are sensitive because there is a great effort to find optimum values for these parameters and a potential intruder may be interested on this actual intelligence;
- the set up for the genetic algorithm: population size, probabilities of applying genetic operations, the maximum number of generations and other parameters involved in the genetic algorithm.

Once a good solution is obtained, the business intelligence mobile application stores the resulted network or corresponding chromosome for later use. Also datasets are saved or result data is saved. An attacker may try to decode application files to extract sensitive information. All the outputs from the application have to be secured via encryption.

In Visual Studio 2010 Express for Windows Phone, is available the namespace **System.Security.Cryptography** for assuring the mobile application security issues. The Cryptography namespace provides cryptographic services, including secure encoding and decoding of data, hashing operations, random number generations, message authentication [14].

This namespace has the following classes [14]:

- AES - represents the abstract base class from which all implementations of the Advanced Encryption Standard (AES) must inherit;

- AesManaged - provides a managed implementation of the Advanced Encryption Standard (AES) symmetric algorithm;
- CryptographicException - exception that is thrown when an error occurs during a cryptographic operation;
- CryptographicUnexpectedOperationException - exception that is thrown when an unexpected operation occurs during a cryptographic operation;
- CryptoStream - defines a stream that links data streams to cryptographic transformations;
- DeriveBytes - represents the abstract base class from which all classes that derive byte sequences of a specified length inherit;
- HashAlgorithm - represents the base class from which all implementations of cryptographic hash algorithms must derive;
- HMAC - represents the abstract class from which all implementations of Hash-based Message Authentication Code (HMAC) must derive;
- HMACSHA1 - computes a Hash-based Message Authentication Code (HMAC) using the SHA1 hash function;
- HMACSHA256 - computes a Hash-based Message Authentication Code (HMAC) using the SHA256 hash function;
- Rfc2898DeriveBytes - implements password-based key derivation functionality, PBKDF2, by using a pseudo-random number generator based on HMACSHA1;
- SHA1 - computes the SHA1 hash for the input data;
- SHA1Managed - computes the SHA1 hash for the input data using the managed library;
- SHA256 - computes the SHA256 hash for the input data;
- SHA256Managed - computes the SHA256 hash for the input data using the managed library;
- SymmetricAlgorithm - represents the abstract base class from which all implementations of symmetric algorithms must inherit.

The inputs and outputs can be encrypted and decrypted using one of the algorithms presented (SHA, HASH, and symmetric algorithm).
The elements used by the application are received and saved using Xml format. For the security component found inside the application this type of data is handled via the simple type of string. There are needed methods for encrypting and decrypting a string and for saving and loading a string to a file and from a file.
The method used to encrypt a string using a symmetric algorithm is:

```
public static string
EncryptString(string InputText, string
Password)
{

Aes RijndaelCipher=new AesManaged ();
byte[] PlainText =
System.Text.Encoding.Unicode.GetBytes(In
putText);
byte[] Salt = new byte[]{ 1, 2,
3,4,5,6,7,8 };
Rfc2898DeriveBytes  SecretKey = new
Rfc2898DeriveBytes(Password, Salt);
ICryptoTransform Encryptor =
RijndaelCipher.CreateEncryptor(SecretKey
.GetBytes(32), SecretKey.GetBytes(16));
MemoryStream memoryStream = new
MemoryStream();
CryptoStream cryptoStream = new
CryptoStream(memoryStream, Encryptor,
CryptoStreamMode.Write);
cryptoStream.Write(PlainText, 0,
PlainText.Length);
cryptoStream.FlushFinalBlock();
byte[] CipherBytes = memoryStream.
ToArray();
memoryStream.Close();
cryptoStream.Close();
string EncryptedData =
Convert.ToBase64String(CipherBytes);
return EncryptedData;
}
```

The method used to decrypt a string using a symmetric algorithm is:

```
public static string
DecryptString(string InputText, string
Password)
{
Aes  RijndaelCipher = new AesManaged ();
byte[] EncryptedData =
Convert.FromBase64String(InputText);
```

```
byte[] Salt = new byte[] { 1, 2, 3, 4,
5, 6, 7, 8 };
Rfc2898DeriveBytes SecretKey = new
Rfc2898DeriveBytes(Password, Salt);
ICryptoTransform Decryptor =
RijndaelCipher.CreateDecryptor(SecretKey
.GetBytes(32), SecretKey.GetBytes(16));
MemoryStream memoryStream = new
MemoryStream(EncryptedData);
CryptoStream cryptoStream = new
CryptoStream(memoryStream, Decryptor,
CryptoStreamMode.Read);
byte[] PlainText = new
byte[EncryptedData.Length];
int DecryptedCount =
cryptoStream.Read(PlainText, 0,
PlainText.Length);
memoryStream.Close();
cryptoStream.Close();
string DecryptedData =
Encoding.Unicode.GetString(PlainText, 0,
DecryptedCount);
return DecryptedData;
}
```

Windows Phone security model also includes the concept of isolated storage.
Isolated storage is a space assigned to every application where this can read or write files. Other applications do not have access to another isolated storage than its own. However, encryption is still a solution because an attacker may change the application such way it makes visible sensitive information.

6 Conclusions

Mobile applications' security aspect has an important role in software developing process. Using a secure component the application is more reliable, because there are identified, taken into accounts and treated all the possible errors that can occur. Using the security, mobile applications tend to become more and more reliable.
Neural networks have an important role in applications development. Depending on the neural network type, there are advantages and disadvantages of using each type.
It has been shown that genetic algorithms are a solution that can be used on mobile devices to solve optimization problems like training a neural network. The obtained solutions are good and the resources used to obtain the solution are reasonable compared to classic training methods.
There are various implementations of these methods on the web, but the aspect of neural network aspect is still developing, being used in predictions, design pattern recognitions and not only.

Acknowledgements

This work was cofinaced from the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213„ Ph.D. for a career in interdisciplinary economic research at the European standards" (DOCCENT).

References

[1] What are the major benefits of neural network? [Online] Available: http://wiki.answers.com/Q/What_are_the_major_benefits_of_neural_networks
[2] S. Haykin, *Neural networks and learning machines*, Prentice Hall Publisher, 2009
[3] A. Vișoiu, *Rafinarea Metricilor Software*, PHD Thesis, Academy of Economic Studies, Bucharest, 2009
[4] M. Klemettinen, *Enabling technologies for mobile services: the MobiLife book*, Wiley-Blackwell Publisher, 2007.
[5] A. Visoiu and S.Trif, "Open Source Security Components for Mobile Applications", *Open Source Science Journal*, Vol. 2, No. 2, 2010, pp. 155-166.
[6] S. Trif, "Open Source Security Components Complexity for mobile applications", *Open Source Science Journal*, Vol. 2, No. 4, 2010, pp. 39-51
[7] H.Dwivedi, C. Clark and D. Thiel, *Mobile Application Security*, McGraw Hill Professional Publisher, USA, 2010
[8] I. Ivan and C. Toma, *Informatics security handbook 2th edition*, ASE Publisher, Bucharest, 2009.
[9] C. Boja, et al.," Secure arhitecture for M-learning bluethooth services", *,Informatica Economică*, Vol. 14, No. 3, 2010, pp. 47-59.
[10] B. Ghilic, M.Stoica and M. Mircea, "How to Succeed in Business Intelligence Initiative: A Case Study for

Acquisitions in Romania Public Institutions", *in Proc. WSEAS TRANSACTIONS on BUSINESS and ECONOMICS*, Issue 6, Vol. 5/2008, pp. 298-309

[11] Back-propagation neuraql network [Online] Available at: http://www.codeproject.com/KB/cs/BackPropagationNeuralNet.aspx

[12] Genetic algorithm [Online] Available: http://en.wikipedia.org/wiki/Genetic_algorithm

[13]System.Security.Cryptography namespace [Online] Available at: http://msdn.microsoft.com/en-us/library/system.security.cryptography.aspx

[14] D. Graupe, *Principles of Artificial Neural Networks 2nd Edition*, World Scientific Publisher, 2007.

[15] Back-propagation [Online] Available at: http://en.wikipedia.org/wiki/Backpropagation

Silvia TRIF graduated the Faculty of Cybernetics, Statistics and Economic Informatics. She has a Master's Degree in Project Management. She is a PhD student of the Doctoral School of Bucharest Academy of Economic Studies in the field of Economic Informatics. Her interests are mobile applications, information security, web applications and project management.