Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 46(2), 2019, Pages 433–444
ISSN: 1223-6934

New classification technique using neural networks and genetic algorithms

Charaf Eddine Bailoul and Nour Eddine Alaa

Abstract. Classification is a data mining technique used to predict group membership for data instances, and its one of the most active research and application areas of neural networks. There are several classification techniques that can be used for classification purpose. In this paper, we develop a new classification technique using neural networks and genetic algorithms. A comparison with the classical backpropagation algorithm is also given. Furthermore, various numerical experiments are presented to confirm the accuracy and efficiency of our proposed technique.

2010 Mathematics Subject Classification. Primary 60J05; Secondary 60J20.
Key words and phrases. Neural networks, Classification, Error function, Backpropagation, Genetic algorithms, Confusion matrix.

1. Introduction

Classification is one of the most frequently encountered decision making tasks of human activity. A classification problem occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object. Many problems in business, science, industry, and medicine can be treated as classification problems. Examples include bankruptcy prediction, credit scoring, medical diagnosis, quality control, handwritten character recognition, and speech recognition.

Since the classes are determined before applying the real data, this method is known as a supervised learning algorithm. The majority of practical machine learning uses supervised learning. In supervised learning external knowledge or information is provided, here each example is a pair consisting of an input object and a desired output value. A supervised algorithm analyses the training data and produces an inferred function which can be used for mapping other examples, the data (observations, measurements, etc.) are labeled with pre-defined classes. Like human learning from past experiences, a computer does not have experiences. A computer system learns from data, which represent some past experiences of an application domain. Our focus is to learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk.

So, the supervised learning process in two steps:

- Learning (training): learn a model using the training data.
- Testing: test the model using unseen test data to access the model accuracy.

Received November 28, 2018. Accepted November 15, 2019.
This paper has been presented at the Conference MOCASIM, Marrakesh, 26-27 November 2018.

The unsupervised learning unlike the supervised learning, entirely without reference to external information. It can be achieved through clustering. A way of grouping together data samples that is similar in some way according to some criteria we just pick out. So, it's a method of data exploration, a way of looking for patterns or structure in the data that are of interest. It involves the use of descriptors and descriptors extraction, descriptors are set of words that describe the contents within the cluster. It is considered to be a centralized process. It requires no predefined classes or category.

Traditional statistical classification procedures such as discriminant analysis are built on the Bayesian decision theory. In these procedures, an underlying probability model must be assumed in order to calculate the posterior probability upon which the classification decision is made. One major limitation of the statistical models is that they work well only when the underlying assumptions are satisfied. The effectiveness of these methods depends to a large extentonthe various assumptions or conditions under which the models are developed. Users must have a good knowledge of both data properties and model capabilities before the models can be successfully applied.

Neural networks have emerged as an important tool for classification. The recent vast research activities in neural classification have established that neural networks are a promising alternative to various conventional classification methods. The advantage of neural networks lies in the following theoretical aspects. First, neural networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Second, they are universal functional approximators in that neural networks can approximate any function with arbitrary accuracy. Since any classification procedure seeks a functional relationship between the group membership and the attributes of the object, accurate identification of this underlying function is doubtlessly important. Third, neural networks are nonlinear models, which makes them flexible in modeling real world complex relationships. Finally, neural networks are able to estimate the posterior probabilities, which provides the basis for establishing classification rule and performing statistical analysis.

On the other hand, the effectiveness of neural network classification has been tested empirically. Neural networks have been successfully applied to a variety of real world classification tasks in industry, business and science. A number of performance comparisons between neural and conventional classifiers have been made by many studies. In addition, several computer experimental evaluations of neural networks for classification problems have been conducted under a variety of conditions.

The original idea of this paper is to minimize the error function of the neural network using the genetic algorithms instead of the classical backpropagation algorithm. The overall organization of the paper is as follows. After the introduction, we define neural networks and we present the classical backpropagation algorithm in the second section. In the third section, we present the principle of genetic algorithm and also the algorithm of our new technique. Finally, section four is dedicated to the analysis of our new algorithm. In fact, numerical experiments are carried out to demonstrate the accuracy and efficiency of the proposed algorithm, in comparing with the classical backpropagation algorithm.

2. Neural networks

Artificial neural networks is that technology which initially grew from the full understanding of some ideas and aspects about how biological systems work, especially the human brain. A multi-layered network consists of numerous neurons, which are arranged into levels. Each level is interconnected with the one above and below it. The first layer receives external inputs and is aptly named the input layer. The last layer provides the classification solution, and is called the output layer. Sandwiched between the input and output layers are any number of hidden layers. It is believed that a three-layered network can accurately classify any non-linear function.

The beginning of Neurocomputing is often taken to be the research article of McCulloch and Pitts [22], which showed that even simple types of neural networks could, in principle, compute any arithmetic or logical function, was widely read and had great influence. Hebb wrote a book entitled The Organization of Behaviour [2], which pursued the idea that classical psychological conditioning is ubiquitous in animals because it is a property of individual neurons. And the first successful neuro-computer (the Mark I perceptron) was developed during 1957 and 1958 by Frank Rosenblatt, Charles Wightman, and others [17]. As we know it today, Rosenblatt as the founder of Neurocomputing. His primary interest was pattern recognition.

In the years 1983 − 1986 John Hopfield, an established physicist of worldwide reputation, had become interested in neural networks a few years earlier. Hopfield wrote two highly readable papers [11], [12] on neural networks and these, together with his many lectures all over the world, persuaded hundreds of highly qualified scientists, mathematicians, and technologists to join the emerging field of neural networks.

Neural networks found their usage in many areas such as: Character recognition; Image compression; Stock market prediction; Medicine, electronic nose and security; Traveling saleman's problem and miscellaneous applications. We will see one of these applications in our numerical simulations, this part will be presented in section 5.

2.1. How neural networks operates: Neural systems are typically organized in layers. Layers are made up of a number of artificial neurons, which contain an activation function. Patterns are presented to the network via the input layer, which communicates to one, or more hidden layers where the actual processing is done through a system of fully or partially weighted connections. The hidden layers are then linked to an output layer where the answer is output. Standard neural networks contain learning rules that modify the weights of the connections according to the input patterns. Generally, the standard neural networks have the capacity to learn, memorize and create relationships amongst data.

The neurons process the input data in two ways: first by forming linear combinations of the input data, then by squashing these linear combinations through the activation function (logsigmoid function).

The logsigmoid function is defined by the following formula:

$$S(t) = \frac{1}{1 + e^{-t}}. \tag{1}$$

An alternative activation function for the neurons in a neural network is the hyperbolic tangent function. It squashes the linear combinations of the inputs within the interval $[-1, 1]$, rather than $[0, 1]$ in the logsigmoid function.

Most artificial neural networks contain some form of learning rule which modifies the weights of the connections according to the input patterns that it is presented with. In a sense, artificial neural networks learn by example as do their biological counterparts; A child learns to recognize dogs from examples of dogs. Although there are many different kinds of learning rules used by neural networks. The delta rule is often utilized by the most common class of artificial neural networks called backpropagational neural networks. Backpropagation is an abbreviation for the backwards propagation of error. Multilayer perceptron (MLP) is the most widely used neural network with the backpropagation algorithm networks. This type of neural networks is excellent at both prediction and classification. Using this neural network algorithm, both input and corresponding desired output data are given to the calibrating phase. As the network starts training, the input layer receives the input signals, and then it processes the data through the hidden layers until the output layer yielding the resulted outputs. These outputs are then compared with the desired outputs computing the error, which is backpropagated through the system causing it to adjust the weights, which control the network.

Once a neural network is trained to a satisfactory level, it may then be used as an analytical tool on other data. To do this, the user no longer specifies any training runs and instead allows the network to work in forward propagation mode only. Inputs are presented to the network through input layer and processed by the middle layers as the training was done. The resulting output is retained and no backpropagation occurs.
The error function is given bye the following formula:

$$\Omega^* = arg\min_{\Omega}(\Psi(\Omega)). \tag{2}$$

Where:
$\Psi(\Omega) = \sum_{t=1}^{T}(y_t - \widehat{y_t})^2$.
$\Omega = \{\omega_{i,j}^{(k)}\}$ is the set of weights in a network.
y is the actual observed output.
$\widehat{y}$ is the output predicted by the network.
T is the number of observations of the output vector y.

2.2. Classical backpropagation algorithm: Backpropagation is an algorithm for supervised learning of artificial neural networks using gradient descent which adjusts the weights to move down the steepest slope of the error surface. Finding the local minimum is not guaranteed since the error surface can include many local minima in which the algorithm can become stuck.

Backpropagation was invented in the 1970s as a general optimization method for performing automatic differentiation of complex nested functions. However, it wasn't until 1986, with the publishing of a paper by Rumelhart and al. [3], that the importance of the algorithm was appreciated by the machine learning community at large. Backpropagation was one of the first methods able to demonstrate that artificial neural networks could learn good internal representations, i.e. their hidden layers learned nontrivial features.

The objective of training is to find the set of weights between the neurons that determine the global minimum of error function. Here we discuss when to stop training a neural network and the selection of learning rate and corresponding momentum

values. There are two schools of thoughts regarding the number of iteration for training before stopping. The first school of thought emphasized that the researcher can only stop training when there is no improvement in the error function based on the reasonable number of randomly selected starting weights. The point at which the network error does not improve is called point of convergence. The objective of convergence training is to attain a global minimum. This school of thought emphasized the need/danger of not getting trapped in a local minimum. The other school of thought advocates a series of train-test interruptions; here the training is stopped after a pre-determined number of iterations and the network ability to generalize on the testing set. Generalization refers to the idea that a model based on sample of the data is suitable for forecasting the entire population. It is the aim of the study to use the first school of thought as pre-determined the number of iteration ahead may lead to the bias of either over fitting or under fitting, instead the network is allowed to determine the number of iteration that give optimal result.

The learning algorithm that controls backpropagation follows a number of steps:

- Initialization: initialize connection weights and neurons to small random values.
- Data introduction: introduce the continuous data set of inputs and actual outputs to the backpropagation network.
- Calculations of outputs: calculate the outputs, and adjust the connection weights several times applying the current network error.
- Adjustments: adjust the activation thresholds and weights of the neurons in the hidden layers according to Delta rule.
- Repeat steps (2) to (4) until desired network accuracy (error) is reached.

Delta rule recursive algorithm re-adjusts connection weights starting at the output nodes and working back to the first hidden layer. Connection weights are adjusted by:

$$W_{i,j}(t+1) = W_{i,j}(t) + \alpha\delta_j x_i. \tag{3}$$

Where:
$W_{i,j}$: the weight from hidden node i or from an input to node j at time t.
x_i: either the output of node i or an input.
α: the learning rate, the positive constant that measures the speed of the convergence of the weight vector.
δ_j: an error term for j (k goes over all nodes in the layers before node j):

$$\delta_j = \begin{cases} (d_j - y_j)y_j(1-y_j) \text{ if } j \text{ is an output node,} \\ (\sum\limits_k \delta_k W_{jk})x_j(1-x_j) \text{ if } j \text{ is a hidden node.} \end{cases} \tag{4}$$

Where:
d_j: the desired output of node j.
y_j: the actual output of the node j.

3. Genetic algorithm

Genetic Algorithm (GA) is a searching technique in computer science and artificial intelligence to find the optimal solution for the searching issues. GA is one of the Evolutionary Algorithms which is inspired by science of biology such as inheritance, mutation, sudden selection, natural selection and composition. It is briefly stated that GA is a programming technique which uses genetic evolution as a model of problem

solving. Most of the time, when the term of struggle for survival is used, its negative value comes to our mind. However, to rest assured, you can think that strongest was not always the winner. Despite their massive size and power, dinosaurs have ceded the survival game and having generation during a quite natural process, while much weaker creatures have continued their lives. It seems that the nature chooses the bests not only based on the size. In fact, it is more correct to say the nature chooses the fittest not the best. In the law of natural selection, having descendant is limited to some population's species which have the best features. Those who do not possess these characteristics will gradually disappear over time. For example, suppose a certain kind of people which are much more intelligent than the rest of the colony. In normal conditions, these people will improve more and have a relatively higher welfare. This welfare will lead to longer life and better reproduction. If intelligence is inheritable, the number of intelligent offspring will be more in the next generation of that community. If this trend continues, you will see that our sample population becomes smarter over the generations. In this way, a natural simple mechanism has managed to remove low-intelligent people of community. In addition, the average intelligence of community is constantly increasing.

Thus, we can see that the nature is able to improve each generation in terms of different features, using a simple mechanism; gradual elimination of undesirable species and reproduction of optimal species. The solutions are typically represented as binary 0 and 1, but there are other ways to display it. Evolution is started from a random set of entities and repeated in subsequent generations. In each generation, the fittest is selected not the best. One solution to this problem is shown by a list of parameters called chromosome or genome. Generally, chromosomes are displayed as a simple string of data. Other types of data structures can be used. At first, several features are produced randomly to create the first generation. During each generation, each feature is evaluated and the fitness value is calculated by fitness function. Typically the genetic algorithms have a number of possible connections which is between 0.6 and 1. It indicates the possibility of children creation. Organisms are combined together with this possibility. Two chromosomes connection creates the child which is added to the next generation. These are done until the appropriate candidates for response in the generation are emerged. The next step is to change the new children. Genetic algorithms have a fixed and small transition probability which is about 0.2 or less. Based on this possibility, child chromosomes are randomly changed or mutated.

The next step is to create a second generation of community. It is conducted by genetic operators (Chromosomes connected to each other and changed) based on the selection processes and production of selected features. For each individual, a pair of parents is selected. The selections are in such a way that the best elements are selected so that the weakest elements have the chance to be selected to prevent achieving the local response. This process creates a new generation of chromosomes which is different from previous generation. The whole process is repeated for the next generation. Pairs are selected for combination. The third generation population comes to existence and so on. This process is repeated until we reach the last stage.

Terminating conditions of genetic algorithms include:

- To reach a fixed number of generations.
- Allocated funds to be finished (computation time / money).

- One person (the produced child) to be found who meet minimum (lowest) criteria.
- To reach the highest degree of children process and not to yield better results.
- Manual inspection.
- High combinations.

GA were first proposed by Holland [10] as a means to find good solutions to problems that were otherwise computationally intractable. And we can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear.

Artificial neural networks consider classification as one of the most dynamic research and application areas. The neural network was trained by backpropagation algorithm, but in this paper we will use a new technique using genetic algorithms in order to avoid the calculation of the derivative of the squared error function with respect to the weights of the network.

4. Proposed algorithm

Our proposed algorithm is the following:

- Initialization: create an initial population. This population is usually randomly generated and can be any desired size, from only a few individuals to thousands. If the population is too small, the genetic algorithm is likely to converge to the local optimal solution, not to converge to the optimal solution. The main reason is that population size is too small, will lead to a decrease in the diversity of the population, leading to meaningful searches and the most advantage is lost. On the contrary, if the size of the population is set too large, the calculation needed in each iteration process will become very large, result in the slow, which will affect the practical application value of the algorithm.
- Each member of the population is then evaluated and we calculate a fitness for that individual. The fitness value is calculated by how well it fits with our desired requirements. The fitness function is used to evaluate the pros and cons of chromosomes in the population, the fitness function and all selection, crossover operation in genetic algorithms have a close relationship and driving force of the GA process. GA carries on the comparison to the fitness of each chromosome in the population, and to determine selection probability based on the ranking, so the fitness function must be positive.
- Selection:
  In a population, chromosome according to certain choice probability directly into the next iteration populations. The fitness value of the chromosome is the higher; the probability of being selected is greater. Poor fitness value of chromosome might not have the opportunity to be selected.
  Here, we want to be constantly improving our populations overall fitness. Selection helps us to do this by discarding the bad designs and only keeping the best individuals in the population. There are a few different selection methods but the basic idea is the same, make it more likely that fitter individuals will be selected for our next generation.

The principle we have used is to associate to each individual $pop(i)(1 \leq i \leq N)$ a probability of reproduction proportional to his fitness:

$$prb(i) = \frac{f(pop(i))}{sum(f(pop(i)))}. \tag{5}$$

To each individual $pop(i)$ we assign a segment $L(i)$ that has for size $prb(i)$. Therefor, individuals are identified by a particular segment of length $prb(i)$. Then, We draw out u uniformly in $[0, 1]$, $u = rand(1, 1)$, and test if u belongs to $L(i)$, then, $pop(i)$ is selected, otherwise $pop(i)$ is eliminated.
We reproduce N times this draft, therefore, we determine N new elements thanks to this draft.

- Crossover:
Selecting operation does not produce new chromosomes, just selecting excellent individuals from the parent generation chromosomes. In order to achieve optimum operation, chromosomal crossover operation must be carried out, only in this way can produce new excellent individual to implement genetic algorithm optimization process. Through the selection and crossover operator, GA can result in higher average fitness of children group, which is in the optimal direction to evolve. Crossover is the main means of genetic algorithm for optimizing search, by crossover operation to produce the new individual, it greatly accelerate the speed of the search in the process of group evolution.
Select a certain number of chromosomes from the population according to a certain probability, and random group. After the completion of the group, exchange a certain genes of the chromosome from the selected location, the selected location can be one also can be more than one. The probability called the crossover probability, it gives the expected number of participating in the exchange of chromosome.
During crossover we create new individuals by combining aspects of our selected individuals. We can think of this as mimicking how sex works in nature. The hope is that by combining certain traits from two or more individuals we will create an even fitter offspring which will inherit the best traits from each of it's parents.
The crossover operator is applied to couples of parents P_1 and P_2 with a probability P_c and generates couples of children C_1 and C_2 by linear combination:

$$\begin{cases} C_1(i) = aP_1(i) + (1-a)P_2(i) \\ C_2(i) = (1-a)P_1(i) + aP_2(i). \end{cases} \tag{6}$$

Where a is a random weighting coefficient adapted to the domain of extension of the genes.
In practice we fix a probability P_c. We draw out a pair $[P_1, P_2]$ and a random probability $u = rand(1, 1)$. We test if $u < P_c$, then, we apply the crossover and the new couple $[C_1, C_2]$ replaces the former one, otherwise, $[P_1, P_2]$ remains in the new population.

- Mutation:
The mutation operation of genetic algorithm is replacing some genes in the chromosome with the other allele of the gene to form a new individual. The important role of the mutation operator is to ensure the diversity of population. And it can

make the genetic algorithm find the optimal solution near the current solution. An important parameter of mutation operation is the mutation probability.
Here, we need to add a little bit randomness into our new populations genetics, otherwise every combination of solutions we can create would be in our initial population.
The mutation operator is applied to other elements P with the probability P_m and generates mutated individuals P'. The mutation operator usually consists of randomly drawing out a gene in the chromosome, to which is added a generally Gaussian noise.
In practice we fix a probability P_m and a standard deviation σ. We draw out an individual P and a random probability $u = rand(1, 1)$. We test if $u < P_m$, then, we apply the mutation and the new individual P' replaces the former one.

$$P' = P + \sigma u. \tag{7}$$

The adaptation level of children (C_1, C_2) and mutated individuals P' are then evaluated before insertion into the new population.

- Different criteria for stopping the algorithm can be chosen:
  - The number of generations.
  - The algorithm can be stopped when the population does not evolve anymore.

5. Application: crab gender classification

Here we attempt to build a classifier that can identify the sex of a crab from its physical measurements. Six physical characteristics of a crab are considered:

- Species (we used two species, 0 and 1)
- Frontallip
- Rearwidth
- Length
- Width
- Depth.

The problem on hand is to identify the sex of a crab given the observed values for each of these 6 physical characteristics.

Our data is organized into two matrices, the input matrix X of size $(6, 200)$ and the target matrix T of size $(2, 200)$. Each column of the input matrix will have six elements representing a crabs species, frontallip, rearwidth, length, width and depth. Each corresponding column of the target matrix will have two elements. Female crabs are represented with a one in the first element, male crabs with a one in the second element. For example, the measures in Table 1 correspond to a female crab. And the ones in Table 2 correspond to a male crab.

Species	Frontallip	Rearwidth	Length	Width	Depth
0	15.6	14	31.6	35.3	13.8

TABLE 1. Female crab.

In this section we numerically investigate the performance of the proposed algorithm using genetic algorithms in comparison with the classical algorithm. The

Species	Frontallip	Rearwidth	Length	Width	Depth
0	20.6	14.4	42.8	46.5	19.6

TABLE 2. Male crab.

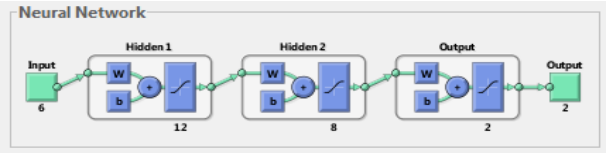

FIGURE 1. Neural network architecture.

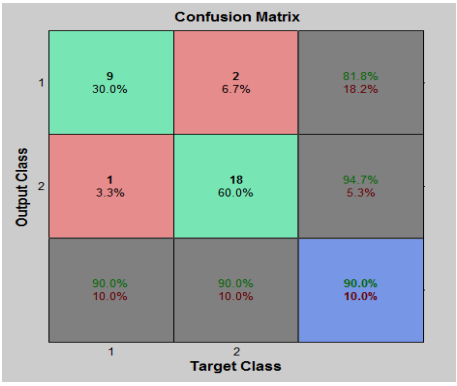

FIGURE 2. The confusion matrix obtained using the Backpropagation algorithm.

architecture of our neural network is given in Figure 1: we used a network with two hidden layers, the first one contains 12 neurons and the second one 8 neurons. The probabilities P_c and P_m are respectively $P_c = 0.7$ and $P_m = 0.05$, the standard deviation is $\sigma = 0.5$. Using a number of generations equal to 20, the numerical results obtained are given in Figure 2 and Figure 3. On the confusion matrix plot, the rows correspond to the predicted class (Output Class) and the columns correspond to the true class (Target Class). The diagonal cells correspond to observations that are correctly classified. The off-diagonal cells correspond to incorrectly classified observations. Both the number of observations and the percentage of the total number of observations are shown in each cell.

The column on the far right of the plot shows the percentages of all the examples predicted to belong to each class that are correctly and incorrectly classified. These metrics are often called the precision (or positive predictive value) and false discovery rate, respectively. The row at the bottom of the plot shows the percentages of all the examples belonging to each class that are correctly and incorrectly classified. And these metrics are often called the recall (or true positive rate) and false negative rate, respectively. The cell in the bottom right of the plot shows the overall accuracy.

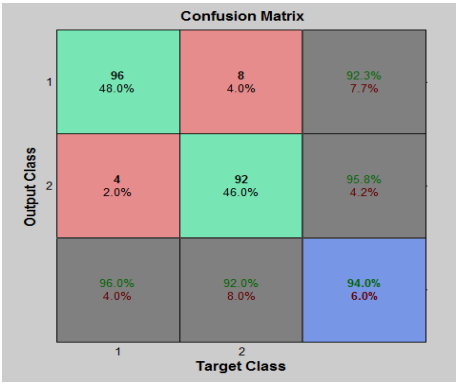

FIGURE 3. The confusion matrix obtained using the proposed algorithm.

By applying the classical backpropagation algorithm to our data, we obtain the confusion matrix presented in Figure 2. As we can see, 90% of the predictions are correct and 10% are wrong.

And by applying our proposed algorithm, we obtain satisfactory results (Figure 3). In this figure, the first two diagonal cells show the number and percentage of correct classifications obtained by the trained network. For example, 96 crabs are correctly classified as male, this corresponds to 48% of all the 200 crabs. Similarly, 92 crabs are correctly classified as female and this corresponds to 46% of our data. 8 male crabs are incorrectly classified as female and this corresponds to 4% of the data. Similarly, 4 female crabs are incorrectly classified as male and this corresponds to 2% of our data.

Out of 104 male predictions, 92.3% are correct and 7.7% are wrong. Out of 96 female predictions, 95.8% are correct and 4.2% are wrong. Out of 100 male cases, 96% are correctly predicted as male and 4% are predicted as female. And out of 100 female cases, 92% are correctly classified as female and 8% are classified as male.

Overall, 94% of the predictions are correct and 6% are wrong. It is to note that our new technique is more accurate and efficient in comparison with the classical backpropagation.

6. Conclusion and perspective

In this paper, we are interested in the optimization of the error function. The classical method and the most popular one used for the treatment of this type of problems is the backpropagation using the gradient descent method. To avoid the calculation of the derivative of the squared error function with respect to the weights of the network, we have developed a new technique using genetic algorithms. The simulation results are compared to those of the backpropagation algorithm, they confirm the accuracy and efficiency of our proposed algorithm. In the near future we will develop an algorithm to bring a satisfactory solution to different problems in Finance, and we will try to deepen on the subject and especially on the use of deep learning.

References

[1] A.M. Oyewale, Evaluation of artificial neural networks in foreign exchange forecasting, *American Journal of Theoretical and Applied Statistics* **4** (2013), no. 2, 94–101.

[2] D. Hebb, *The organization of behavior*, Wiley, New York, 1949.

[3] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Representations by Back-Propagating Errors, *Nature* **323** (1986).

[4] F. Rosenblatt, *Principles of neurodynamics: Perceptions and the theory of brain mechanism*, Spartan Books, Washington, DC, 1961.

[5] F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review* **65** (1958), no. 6, 386–408.

[6] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Contr. Signals Syst.* **2** (1989), 303–314.

[7] G.P. Zhang, Neural Networks for Classification: A Survey, *IEEE Transactions On Systems, Man, And Cybernetics-part C: Applications And Reviews* **30** (2000), no. 4.

[8] J. Zhu, H. Zhang, L. Zhou, J. Guo, Order Batching Optimization in Dual Zone Type Warehouse Based on Genetic Algorithms, *Science Journal of Business and Management* **3** (2015), no. 3, 77–81.

[9] J. McCall, Genetic algorithms for modelling and optimisation, *Journal of Computational and Applied Mathematics* (2005), 205–222.

[10] J.E. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press Cambridge, MA, 1992.

[11] J.J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proceedings of the National Academy of Sciences of the United States of America* **79** (1982), no. 8, 2554–2558.

[12] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, *Proceedings of the National Academy of Sciences of the United States of America* **81** (1984), 3088–3092.

[13] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* **4** (1991), 251–257.

[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* **2** (1989), 359–366.

[15] K. Deepshikha, Supervised and unsupervised document classification-a survey, *International Journal of Computer Science and Information Technologies* **2** (2015), no. 6, 1971–1974.

[16] M.D. Richard, R. Lippmann, Neural network classifiers estimate Bayesian a posteriori probabilities, *Neural Comput.* **3** (1991), 461–483.

[17] N. Yadav, A. Yadav, M. Kumar, *An Introduction to Neural Network Methods for Differential Equations*, Springer, 2015.

[18] N.K. Dizaji, N. Masoudi, A. Sakhvati, A new method to detect circles in images based on genetic algorithms, *International Journal of Intelligent Information Systems* **6-1** (2014), no. 3, 49–55.

[19] P.D. McNelis, *Neural networks in Finance: gaining predictive edge in the market*, Academic Press, 2005.

[20] P.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[21] S.O. Olatunji, M.S. Al-Ahmadi, M. Elshafei, Y.A. Fallatah, Forecasting the Saudi Arabia stock prices based on artificial neural networks model, *International Journal of Intelligent Information Systems* **5** (2013), no. 2, 77–86.

[22] W.S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* **5** (1943).

(Charaf Eddine Bailoul, Nour Eddine Alaa) Laboratory LAMAI, University Cadi Ayyad, Faculty of Sciences and Technologies, 112 Boulevard Abdelkrim Al Khattabi, Marrakesh, Morocco

E-mail address: bailoul.charaf.eddine@gmail.com, n.alaa@uca.ac.ma