KSÜ Mühendislik Bilimleri Dergisi 21(3):246-257, 2018
Araştırma Makalesi

KSU J Eng Sci, 21(3):246-257, 2018
Research Article

Kahramanmaras Sutcu Imam University
Journal of Engineering Sciences

Geliş Tarihi : 04/08/2017
Kabul Tarihi : 26/09/2018

Received Date : 04/08/2017
Accepted Date : 26/09/2018

Application of Artificial Neural Network for Wheat Type Classification: A Novel Artificial Intelligence Training Software

Buğday Tipi Sınıflandırma için Yapay Sinir Ağı Uygulaması: Yeni Bir Yapay Zeka Eğitimi Yazılımı

Muslum OZTURK[1*], *Turan PAKSOY*[2]

[1]University of Kilis 7 December, Department of Computer Technologies, Kilis, Turkey
[2]Konya Technical University, Department of Industrial Engineering, Konya, Turkey

*Sorumlu Yazar / Corresponding Author: Müslüm ÖZTÜRK, mozturk@kilis.edu.tr

ABSTRACT

In this study, a training software with a visual interface was developed by using C# programming language in .NET platform for the purpose of using it in artificial neural networks' training. The created artificial neural network classification software was applied on wheat type classification case study and successful results were achieved. Additionally, the importance of artificial neural networks was mentioned in the study. Backpropagation algorithm was utilized in wheat type classification case and in the developed software in order to eliminate the difficulty and complexity of the use of current software in an artificial neural network training process. Additionally, the developed software was designed in flexible and convenient structure as to be used in the applications which could be solved with artificial neural networks and other all kinds of studies.

Keywords: Artificial Neural Networks, Artificial Intelligence, Classification, Artificial Intelligence Training Application, Interface

ÖZET

Bu çalışmada, yapay sinir ağlarının eğitiminde kullanılmak üzere .NET ortamında C# programlama dili kullanılarak görsel arayüze sahip bir eğitim yazılımı geliştirilmiştir. Gerçekleştirilen yapay sinir ağı sınıflandırma yazılımı buğday türü sınıflandırılması örneğine uygulanmış ve başarılı sonuçlar alınmıştır. Ayrıca çalışmada, yapay sinir ağları konusunun önemine değinilmiştir. Yapay sinir ağlarının eğitilmesi aşamasında kullanılan mevcut yazılımların kullanım karmaşıklığının ve zorluğunun önüne geçmek için geliştirilen yazılımda buğday türlerinin sınıflandırılmasında geriye yayılım (backprogration) algoritmasından yararlanılmıştır. Ayrıca geliştirilen bu yazılım yapay sinir ağları ile çözülebilecek uygulamalarda ve diğer bütün çalışmalarda kullanılabilecek esneklik ve yapıda tasarlanmıştır.

Anahtar Kelimeler: Yapay Sinir Ağları, Yapay Zeka, Sınıflandırma, Yapay Zeka Eğitim Uygulaması, Arayüz.

1. INTRODUCTION

The human visual system efficiently recognizes and localizes objects within cluttered scenes. For artificial systems, however, this is still difficult, due to viewpoint-dependent object variability, and the high in-class variability of many object types (CireşAn et al., 2012). But; Latest developments in machine training provided solutions for many problems which could not be normally solved in the past. In addition to new approaches, which are based on artificial neural networks (ANN), pattern recognition is very effective on dealing with perception of visual objects, speech recognition, signal processing and estimation (LeCun et al., 2015).

ANN are physical cellular systems that receive, store and use the experimental information (Sağıroğlu et al., 2003). Fuzzy logic is defined as a mathematical set-up established for the study of ambiguous situations and uncertain situations (Gani et al., 2015). Solving the problems with artificial intelligence methods in computers, which are very difficult for people to solve and cannot be depicted with a formula in a mathematical manner, makes indispensable in terms of applications. Artificial intelligence applications are also called as smart systems. The most basic feature of smart systems is that they can make decisions depending on information while working on or providing solutions for incidents and problems (Elmas, 2007). Mathematical models are generally established while working on various systems in classical approaches. On the other hand, ANN are used for those problems which cannot be defined or are very difficult to define in mathematical models. Due to this feature, ANN becomes more popular for modelling of non-linear systems in business, finance and industrial areas (Elmas, 2003).

Artificial neural network, or shortly ANN, is emerged as a result of likening of human brain's operating system in artificial manners. In its most general mean, ANN could be regarded as a complex system which is created with combinations of many neurons (nerve cells) in human brain or artificially simple operators with each other in different influence levels. The studies, which were started with mathematical modelling efforts for neurons in human brain in the basic medical sciences, have been disciplined in the recent ten years. Today, ANN is a research subject in many science areas including physic, mathematics, electric and computer engineering. Generally, practice area of ANN is more about rapidly defining and recognition of information data, which might be found in multiple structure or forms (URL-1). Additionally, according to (Fauset, 1994); learning process, storing the learnt information and making assumptions depending on these information can be achieved on a neural network in which neurons are interconnected as in human brain. Today, ANNs are preferred in control areas due to the fact that they can be applied on multiple input and output systems with the nature of their structure, ability of simulation, generalization learning in real time. Additionally, according to (Tekin ve Gökbulut, 2008), ANNs' ability to decrease error rate and ability to proceed information in a fast manner with parallel to the nature of their structure are important features.

The most important reason of using ANNs in engineering applications in a wide scope is to create an effective alternative for such problems whose solutions with classic technics are difficult. Because, computers yet could not be sufficiently successful in human brain's functions such as learning and recognition although they are hundred times successful in mathematical and algorithmic calculations, which are human's brain ability is the weakest on, e. g. multiplication and division, in terms of speed and accuracy. Comparison of working system structure between computer and human brain is presented in Table 1.

Table 1. Comparison of Working System Structure between Computer and Human Brain

Computer	**Human Brain**
Analytical	Analog
Series	Parallel
Instruction Set	Adapting the Information
The Effects in Association with Wrong Calculations	The Effects of Units on the Main Processes are Less
The Effects in Association with Errors in Input Data	Not Always Sensitive to the Errors in Input Data

A training software, which learning and generalization features of ANNs could easily be seen, was developed in this study. A user interface was designed using C# programming language in software .NET platform (See Figure 1). The training was performed with backpropagation algorithm of ANN by using the data input and output values which were recorded in Excel format with the developed training software. Moreover, the related error rates and weight changes could be observed in the software interface. Additionally, the software was designed with C# .NET technology in such a level and sufficient flexibility that the artificial neural networks' learning and decision making abilities could be applied to all kinds of classification problems.

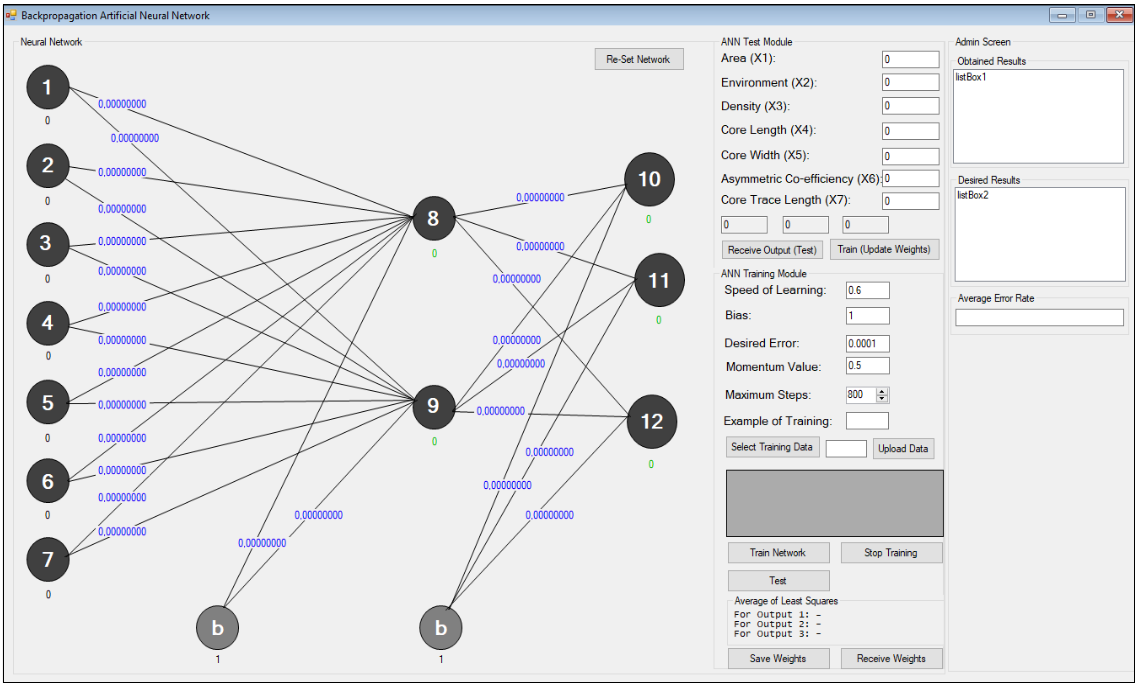

Figure 1. The Developed Artificial Neural Network Visual Interface

The main purpose of preparing the related ANN software was to ensure that the structure, components, training and testing processes of ANN were learnt effectively and users could make classification of three wheat types with a visual interface. The number of learning, the rate of learning and the coefficient of momentum, which are working parameters of ANN, might be varied depending on users. Training is continued until the number of maximum number of learning (the number of iteration). After the completion of training, weight value of ANN structure is stored in an Access database to use in testing processes.

2. FIELD RESEARCH

2.1. Artificial Neural Networks (ANN)

ANN is a parallel spread information processing system. In other words, there is an information processing function, which is composed of intelligence required processes, in the foundation of ANN. The system is composed of process elements, which are inter combined to each other with unidirectional signal channels (connections). The number of output signal is only one as also be increased if needed. There is a similarity between the main thought of ANN approach and human brain's functions. Therefore, ANN system can be called as a model of human brain. ANN is able to adapt its own behaviors depending on environmental conditions. It can reconfigure itself to give different answers as inputs and the desired outputs are entered into the system. However, its internal structure is very complex. Therefore, ANNs, which have been developed up to date, are composite elements that function as taking example of biological functions' main neurons.

2.2. Artificial Intelligence (AI)

Artificial intelligence (AI) is defined as a computer or computer controlled machine's ability to perform the duties, which are related to high intelligence required processes, for example thinking, reasoning, generalization and learning from past experiences that are considered as general characteristics of humans (Nabiyev, 2005).

Moreover, ANN is used in many applications for their superior features for example, their learning ability, easily adaptability to other problems, generalization ability, fast working ability with the help of their parallel structure and users do not have to define the relationship between input and output (Sağıroğlu ve ark., 2003) and (Haykin and Lippmann, 1994).

ANN, from AI technics, is generally used in classification problems. ANN is composed of artificial neurons, which are combined to each other in a hierarchical order, as presented in Figure 2.

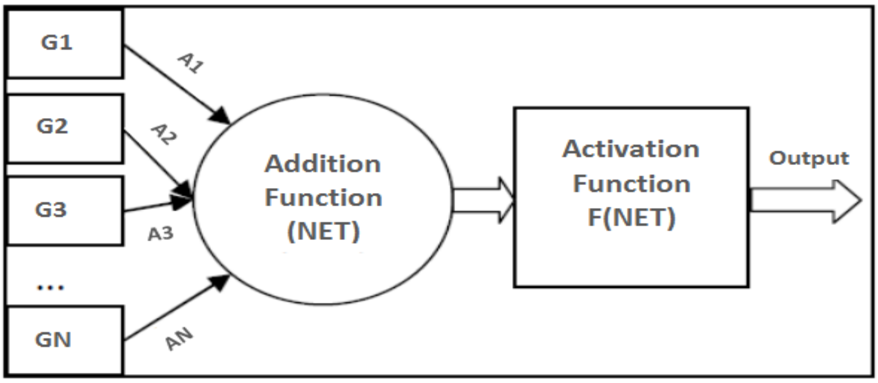

Figure 2. Example of an Artificial Neuron Cell

In Figure 2, inputs which are depicted as G1, G2, G3, ..., GN are known as ANN inputs. A1, A2, A3,....AN are defined as weights and show the effect of the information coming to an artificial neuron cell. Addition function (NET) calculates the net information coming to a neuron cell. Various functions are utilized in order to obtain the net value; however the most frequently used function is the equation which is showed in Equation (1). Here Gi i., Ai and NET represents input value, weight of this input value and total value of the function, representatively.

$$\boldsymbol{NET = \sum_i^n G_i A_i} \qquad \text{Equation (1)}$$

Sigmoid Function is generally preferred for this kind of studies because majority of the application is designed in Multiple Layer Perception format. This function is formulized as shown in Equation (2). In this equation, F(NET) represents the activation function. The software which was developed under the scope of this study, was designed as utilizing Sigmoid function for the solution of the problems.

$$\boldsymbol{F(NET) = \frac{1}{1+e^{-NET}}} \qquad \text{Equation (2)}$$

Activation function F(NET) in an artificial neuron cell determines output value to be produced as calculating net inputs coming to the cell. Sigmoid activation function, which is the most frequently used activation function although there are many activation function for artificial neural networks, was preferred in this software.

There are many ANN structures in literature. Multiple Layer Perception (MLP) model was preferred in this classification study since learning algorithm was usable for training the network, very easy to use and one of the frequently used models. Additionally, MLP is an ANN structure which has been applied to many areas. An MLP structure is presented in Figure 3.

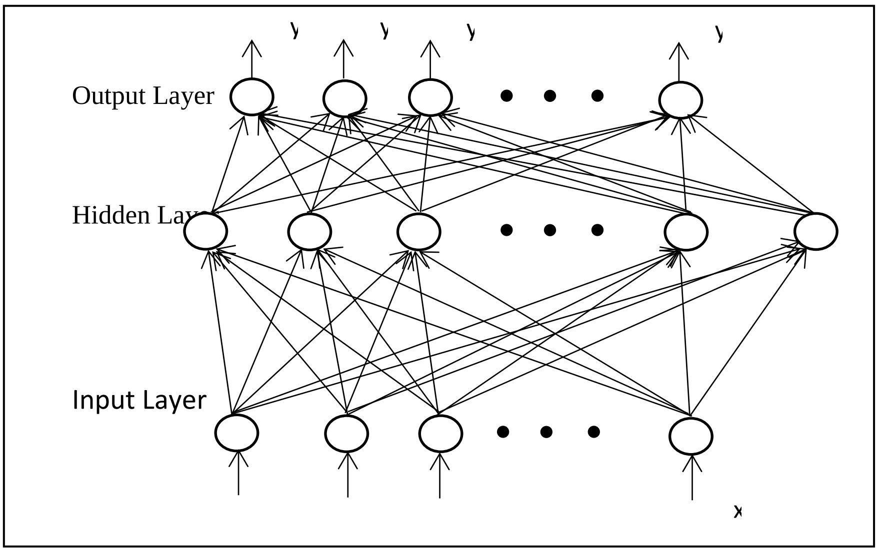

Figure 3. MLP-ANN Model

Basically the duty of an ANN is to determine an output set corresponding to an input set which is introduced to it. In order to perform this duty, network reaches the ability to solve functions about the related problem as it is trained with the problem's examples (learning) (Kaastra and Boyd, 1995).

ANN is composed as a combination of neuron cells. This process is not conducted randomly. Cells are generally in a 3 layer format and each layer constitutes the network in parallel formats. These layers are as follows;

- **Input Layer:** Cells in this layer are responsible of sending input information to hidden layer.
- **Hidden Layer:** The information coming from input layer is sent to output layer as being processed. There might be more than one hidden layer in a network depending on the problem conditions.
- **Output Layer:** Cells in this layer are responsible of sending the information coming from hidden layer to output layer. Outputs that are produced in this layer provide for the problem solution (Uğur ve Kınacı, 2006).

A general network structure, which is created by input layer, hidden layer and output layer, is presented in Figure 4.

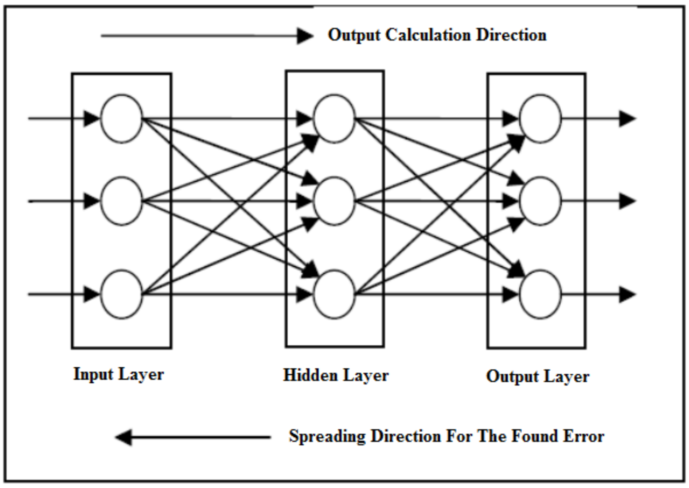

Figure 4. General Network Structure of ANN

There are many learning algorithms which are used in artificial neural networks. Backpropagation algorithm was used in this study. Backpropagation algorithm is the most general algorithm which is used in instructively learning. It is simple and has a good learning capacity enabling it to be applied to many areas. Backpropagation algorithm is an ANN model which is trained instructively, feedforward and a multiple layer format in which there is a complete connection between the layers that perform functions in two phases as Backpropagation and Adapt. Finding error signals between inputs and outputs, weights are updated according to error signals. In an other saying; the back propagation calculate the gradient from the loss augmented inference layer to convolutional layer (Witoonchart and Chongstitvatana, 2017).

The calculated error signals are transferred to nodes in interlayer corresponding to each output node. Therefore, each of interlayer nodes includes only the calculated part of total error. This process is repeated until input layer as nodes in every layer includes a certain part of total error. Taking basis of the obtained error signals, connection weights are updated in all nodes. This update enables the convergence of the network to a condition in which all data can be coded.

In feedforward phase, neurons in input layer directly transfer data values to hidden layer. Each neuron in hidden layer calculates the total value as weighting its own input value and transfers the results to next layer or directly to output layer as processing the results in an activation function. Weights between layers are randomly selected from small numbers in the beginning.

After calculating each neuron’s weighted value in output layer, error is minimized by making a re-comparison of the values with the set activation function. Iteration process is repeated until error value is reduced to a certain degree and therefore, training process of the network is completed. Training of weight values in the connection between layers are obtained from the network and stored to be used in testing phase (Hagan et al., 1996).

Learning rate is constant which is used in changes of ANN weights. They have to be less than one and positive values. In calculation, positive values enable better convergence of the calculated values to the desired values and values that are less than one enables for system to be consistent. Learning rate has an important effect on network performance. Training duration takes a long time for small learning rate values however training process can be completed shorter amount of time while increasing the values. In case of increasing the learning rate, a decrease in necessary steps in learning process (number of learning - iteration) occurs. Increasing the learning rate creates an improvement in network’s total error. However, giving too large values to learning rate does not create a good convergence (Elmas, 2003).

Learning rate and momentum rate are basic two variables affecting the speed of training process (Haykin, 2009). The most accurate value of the momentum parameter can be found through experiments. By way of the momentum parameter, the smallest steps are slightly increased in magnitude. It is more appropriate to reduce the learning coefficient so that this increase amount is more compatible. If a large learning coefficient (close to 1) and a large momentum coefficient (close to 1) are used, the steps will be very large. Perhaps it will not be possible to find the absolute smallest value on this tab.

There are two methods to determine determining learning rate and momentum rate (Kandil, 1993). These methods;

- **Trial-and-Error Method:** Since the trial and error method requires human interaction, artificial neural networks take a long time to learn and require a lot of effort.
- **Second-Order Method:** This method can be used in feedback artificial neural networks; is a method that adjusts the learning rate by using the information obtained in the education process. This method is rarely used because it requires large computational resources.

Sigmoid Function from activation functions and Error Backpropagation Model from learning models were preferred in three type wheat classification application which was developed for validation of the software that was developed in the scope of this study.

3. ANALYSIS OF THE DEVELOPED ANN SOFTWARE

ANN is one of the most frequently preferred methods for artificial intelligence applications. ANN is preferred to analyze or control the workings and effects of a system. ANN program in this study was designed as to have a visual interface for the purpose of classifying three wheat types. Additionally, the developed ANN software has a flexibility and visual interface which can be an example for other studies. In addition, we can list the advantages of the prepared ANN software according to the WEKA software used in the classification field as follows:

- The use of both Turkish and English interfaces,
- Using the commonly used excel file format instead of the .arff file extension used by Weka and not known by many users,
- All operations can be performed through a single module,
- It is easy to use for someone with a basic level of knowledge about ANN.

The developed ANN software can easily conduct an ANN study as preparing a basic information level and proper data about ANN. Buttons must be passive-active depending on order of the processes to instruct users in the developed ANN program. The differences, which can be created as updating the parameters, can easily be followed. A full screenshot of ANN training panel is presented in Figure 5.

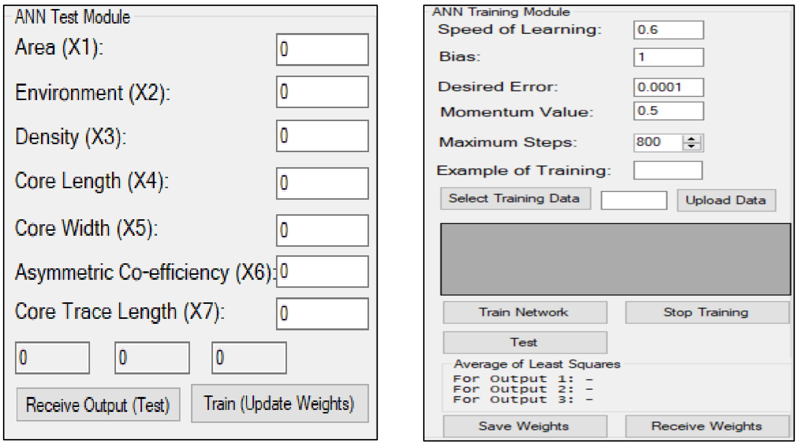

Figure 5. ANN Training Panel

Duties of menus in ANN Training Panel screenshot can be summarized as follows:

- **Maximum Step:** The highest value which the number of learning can reach in ANN training process. It is also known as the number of iteration. Training is stopped when this value is reached.
- **Speed of Learning:** The learning constant, which is used while updating ANN weight values. A positive and greater than one value must be used. Learning rate in the software is used in calculations as being defined in percentages.
- **Momentum:** It is a number between 0 and 1 to decrease error and create a recovery in weight updates. This value in the software is used in calculations as taking percentage converting it to between 0 and 100.
- **Button of Select Data to be used in Training (Select Training Data Button):** It is used to select the training data which is stored in a computer.
- **Button of Upload Data to be used in Training (Select Training Data Button):** It is used to upload the file of training data set.
- **Desired Error:** It determines the training stop criterion when desired error value is reached.
- **Button of Launch the Training (Train the Network Button):** It is the button, which becomes active to start the training after all uploading is completed.
- **Stop Training Button:** It stops the study in any moment of the training process.
- **Save Weights Button:** It is used to save the calculated weights values in an Access database after completing the training.
- **Receive Weights Button:** It is used to transfer weight values, which was previously saved to an Access database after the training, to the related weight variables.
- **Maximum Number of Learning:** It shows the number of learning which is processed during the training.

According to (Kılıç, 1998), a feedback network structure is generally preferred for training different networks as grouping them in studies. Because the situation that affects input would like to be transferred to output by establishing a connection between input and output. Learning becomes more successful since this situation affects the new inputs. In this study, a wheat classification case study was conducted to show the success of this software.

ANN software, which was developed for wheat type classification in this study, had single hidden layer, feedforward and error backpropagation model and also was designed in the format of multiple layer perception as to use Sigmoid activation function. User interface of the software, which was created two modules as training and testing phases, is presented in Figure 6.

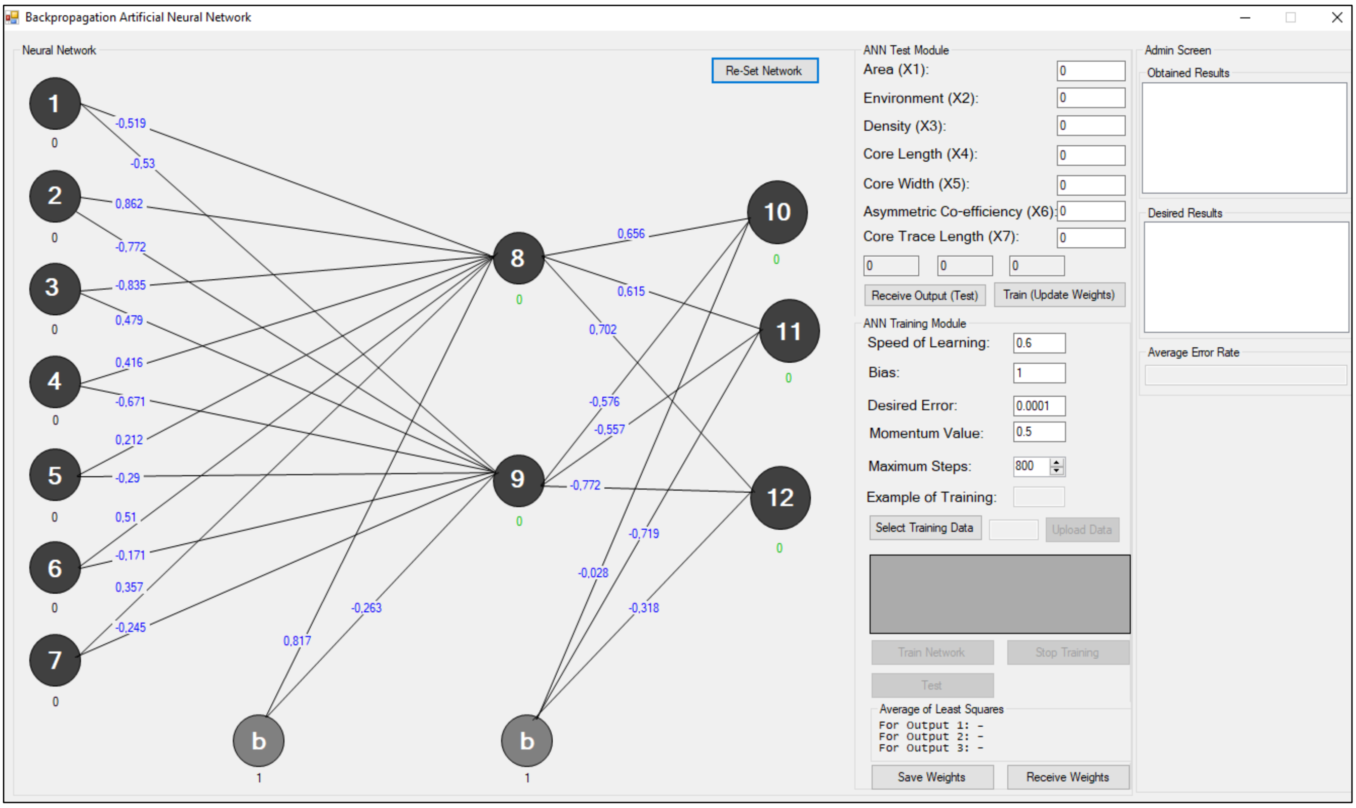

Figure 6. The Developed ANN Training Software Interface

In the developed ANN software, basic functions in training and testing processes of ANN are managed through the interface. These features are the basic features which might be required in an ANN training software.

3.1. Data Uploading to ANN Software

ANN produces new output data from different input data by establishing a relation between input and output data in the system. The process of establishing the relation between input and output is called as ANN Training. Therefore, the data set that will be used for the training purposes should be prepared as to exemplify the system working. Data in the ANN software of this study was used and processed in training and testing processes in the same ways. The data to be used was prepared in MS Excel file (*.xls). The related excel file is selected with select training button and the related classification data is transferred to dataGridView object that is located in the software with upload data button so that the related input and output data becomes ready for ANN training. Examples to be included to training button represents the number of input values to be used in the training in numerical version. An example of data set uploading condition is presented in Figure 7 and ten example data of input and output data in excel format is presented in Figure 8.

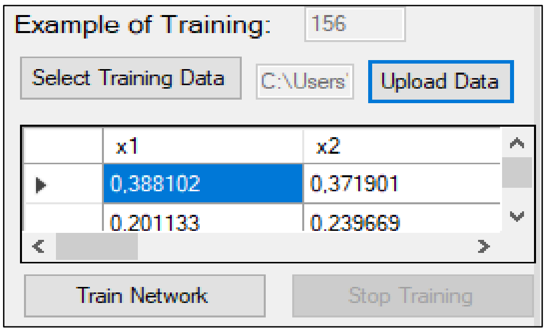

Figure 7. A Data Set Example, Which is Uploaded to the System

	A	B	C	D	E	F	G	H	I	J
1	x1	x2	x3	x4	x5	x6	x7	y1	y2	y3
2	0,388102	0,371901	0,972777	0,172297	0,595866	0,130271	0,064008	1	0	0
3	0,201133	0,239669	0,549002	0,184122	0,298646	0,433876	0,194485	1	0	0
4	0,33711	0,411157	0,456443	0,427365	0,355666	0,299952	0,323486	1	0	0
5	0,332389	0,382231	0,58167	0,349662	0,383464	0,250023	0,344658	1	0	0
6	0,499528	0,514463	0,823049	0,404842	0,625089	0	0,281635	1	0	0
7	0,140699	0,169421	0,529038	0,112613	0,218104	0,084502	0,217627	1	0	0
8	0,417375	0,485537	0,522686	0,501126	0,438346	0,133391	0,237322	1	0	0
9	0,528801	0,568182	0,696915	0,525901	0,563792	0,01793	0,387986	1	0	0
10	0,229462	0,278926	0,508167	0,279279	0,282252	0,339089	0,150665	1	0	0

Figure 8. A Data Set Example in Excel Format which is Prepared for ANN Training

3.2. Training the Prepared ANN Software

According to the information that was provided in Figure 7, ANN software will perform the training of ANN until the determined error rate by updating weight values using error backpropagation algorithm as much as the determined number of iteration after clicking on train network button following the upload of wheat classification data set to the software. On the other hand, stop training button stops the training of ANN in a certain t time and saves the weight changes in t time to database. The training of ANN and weight changes in a certain t time are presented in Figure 9 and stopping of the training of ANN in a certain t time is presented in Figure 10.

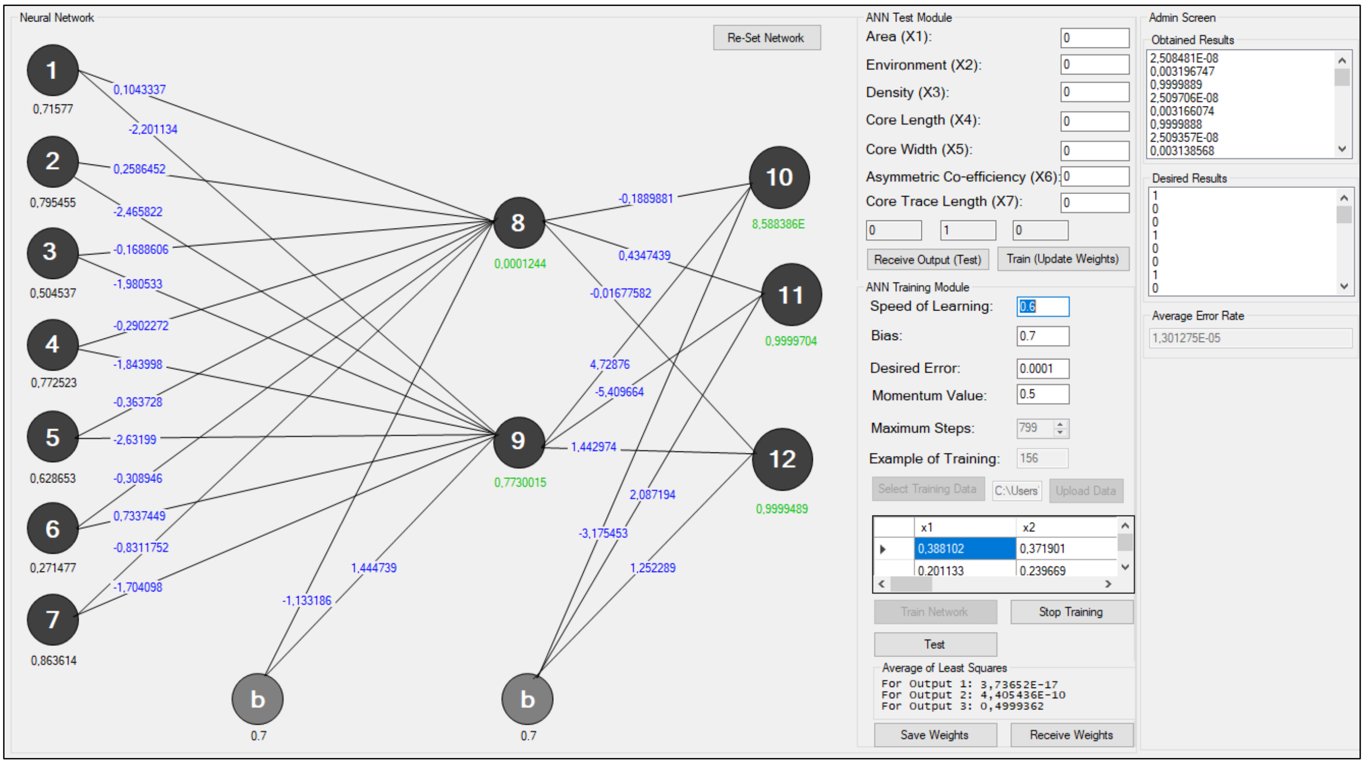

Figure 9. The Training of ANN in a Certain t Time

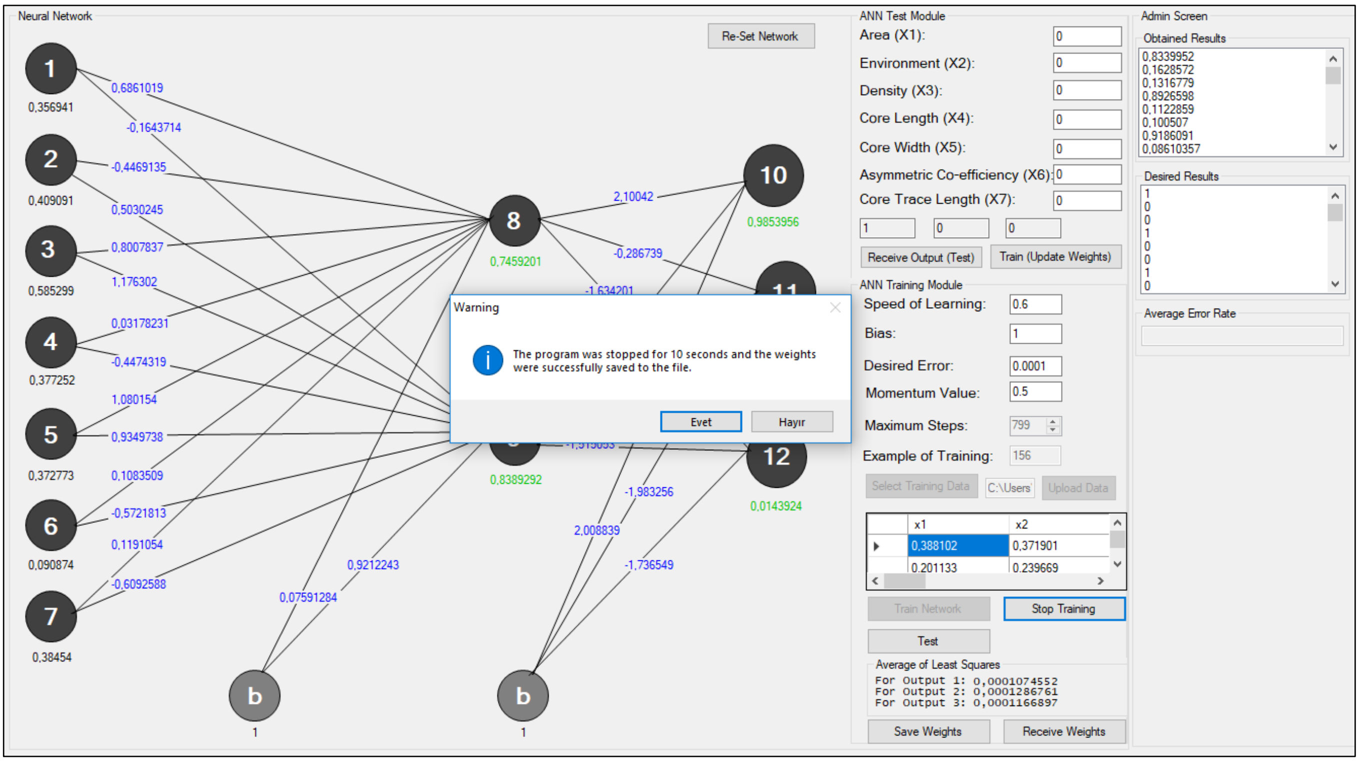

Figure 10. Stopping the Training of ANN in a Certain t Time and Saving Weight Values

If "Train Network" button is clicked on the software interface after determining parameters of network structure and input/output values in the program, the system continues training until the desired number of steps hence determines the most convenient network parameters for problem solution as updating weight values, which were previously determined in a random manner. Calculation process stops if the desired number of error is achieved during this process. In our case study, the developed software trained itself in 285 steps according to the introduced input values for three types wheat classification and completed the training in 40 minutes with the error rate of 5,678281E-05. As a result of the training, the obtained weight values were presented in Figure 11.

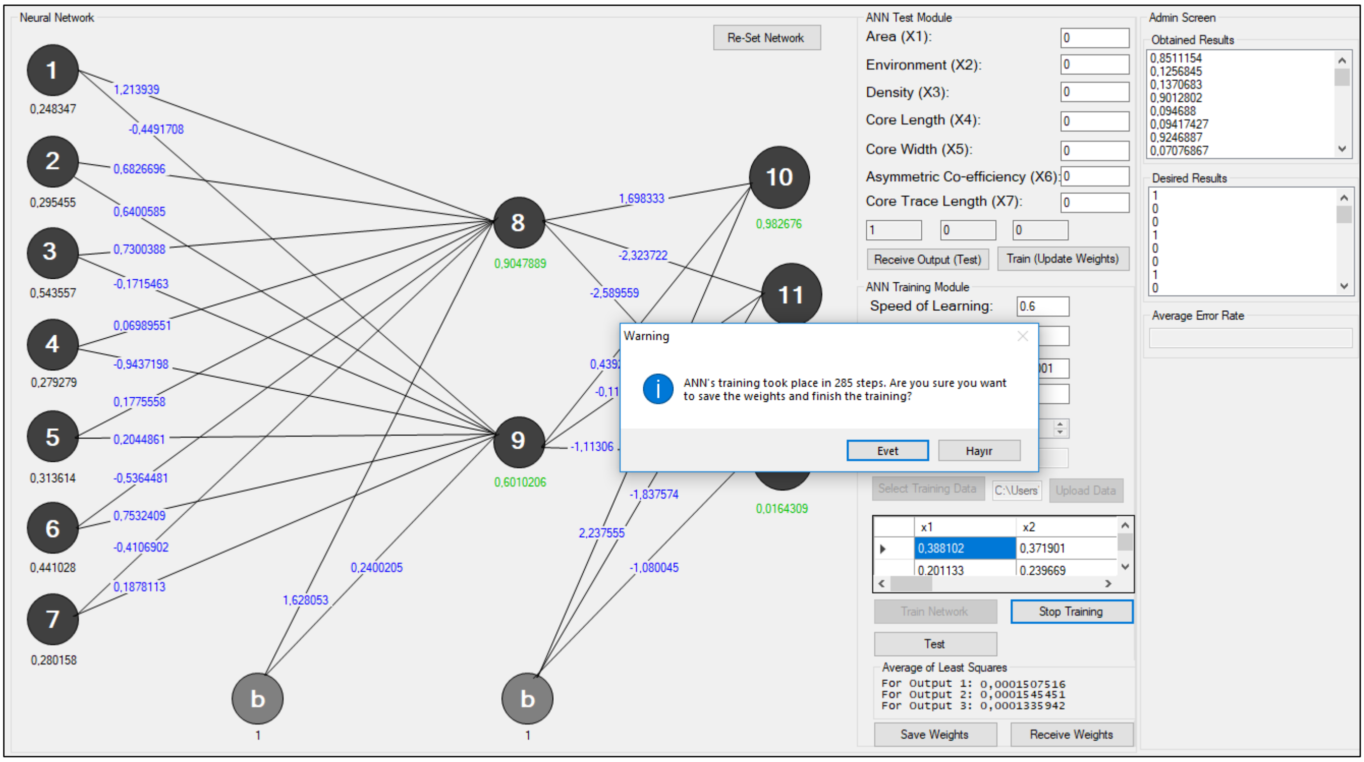

Figure 11. General Figure of the Program After 285 Steps (Iteration)

3.3. Saving Process of the Trained ANN

As seen in Figure 12 and Figure 13, the weight values, which were found as a result of completing the ANN software training, were saved in an access database as clicking on save weights button. On the other hand, receive weights button uploads the weight values, which were saved after completing the training of ANN, to the software so that testing process can be launched from weight values when needed.

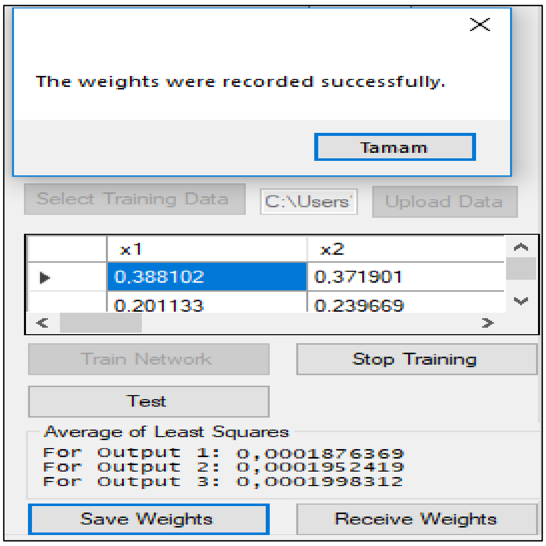

Figure 12. Saving Weight Values to ANN

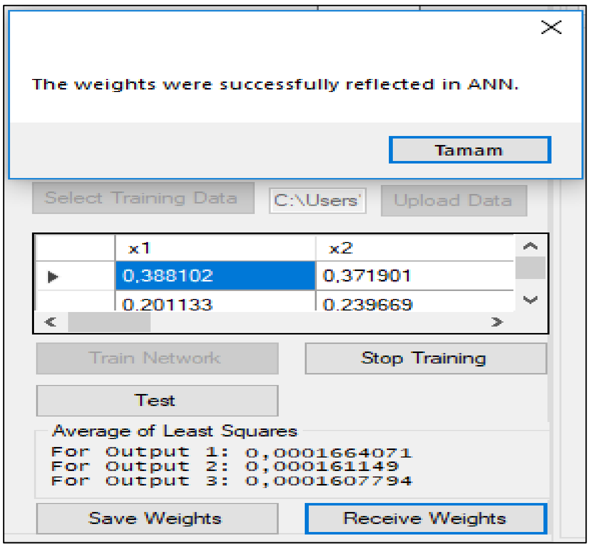

Figure 13. Re-Uploading Weights to ANN

3.4. Testing Process for the Prepared ANN Software

The form division, as could be seen in Figure 14, can be used to test the software after completing the training of the prepared ANN software.

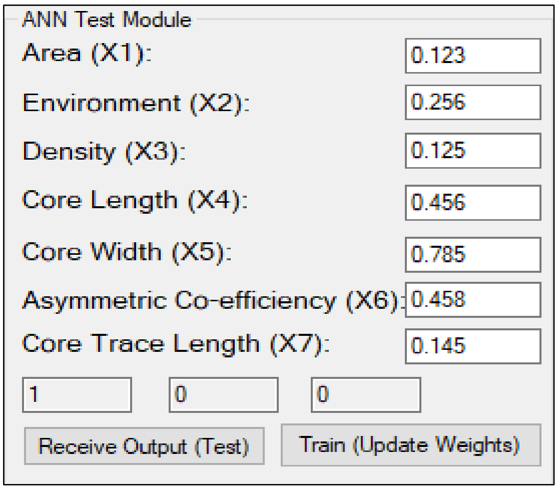

Figure 14. ANN Test Module

In this module, there was 7 data input for three types of wheat classification. When receive output button was clicked after introducing the related input values, three-output data output set in types of (1, 0, 0), (0, 1, 0) and (0, 0, 1) was obtained, as could be seen in Figure 14. Through this module, ANN was tested by making data input after completing the training of the prepared ANN. The obtained values, as a result of separate training and testing process for each of three wheat types, are presented in Table 2.

Table 2. ANN Test Results

Input Values							Expected Output Values			Output Values which were Obtained in Testing Process			Success (%) Classification
Input1 (x1)	**Input2 (x2)**	**Input3 (x3)**	**Input4 (x4)**	**Input5 (x5)**	**Input6 (x6)**	**Input7 (x7)**	**Output1 (y1)**	**Output2 (y2)**	**Output3 (y3)**	**Output1 (y1)**	**Output2 (y2)**	**Output3 (y3)**	
0.440982	0.502066	0.57078	0.486486	0.486101	0.189302	0.34515	1	0	0	1	0	0	
0.405099	0.446281	0.662432	0.368806	0.501069	0.032883	0.215165	1	0	0	1	0	0	
0.349386	0.347107	0.87931	0.220721	0.50392	0.251453	0.150665	1	0	0	1	0	0	
0.306893	0.316116	0.793103	0.239302	0.533856	0.194243	0.140817	1	0	0	1	0	0	
0.524079	0.533058	0.864791	0.427365	0.664291	0.076701	0.322994	1	0	0	1	0	0	
0.357885	0.371901	0.789474	0.274212	0.486101	0.220637	0.215165	1	0	0	1	0	0	
0.387158	0.429752	0.651543	0.373874	0.448325	0.366784	0.344658	1	0	0	1	0	0	
0.332389	0.349174	0.753176	0.293356	0.478974	0.251583	0.236829	1	0	0	1	0	0	
0.570349	0.630165	0.604356	0.649775	0.595153	0.165767	0.668636	1	0	0	1	0	0	
0.552408	0.586777	0.725045	0.554617	0.623664	0.156536	0.499261	1	0	0	1	0	0	**100%**
0.440982	0.504132	0.558076	0.458896	0.436208	0.491217	0.391433	1	0	0	1	0	0	
0.324835	0.36157	0.64882	0.303491	0.406985	0.12377	0.237322	1	0	0	1	0	0	
0.311615	0.332645	0.725045	0.304054	0.40556	0.418794	0.107829	1	0	0	1	0	0	
0.301228	0.340909	0.615245	0.326577	0.374911	0.308273	0.173806	1	0	0	1	0	0	
0.29745	0.338843	0.601633	0.328266	0.344975	0.281749	0.150665	1	0	0	1	0	0	
0.377715	0.386364	0.827586	0.254505	0.501069	0.444668	0.129	1	0	0	1	0	0	
0.321058	0.293388	1	0.123874	0.536707	0.581063	0.129	1	0	0	1	0	0	
0.481586	0.483471	0.88657	0.353604	0.630078	0.108427	0.259478	1	0	0	1	0	0	
0.664778	0.737603	0.537205	0.727477	0.663578	0.430496	0.75874	0	1	0	0	1	0	
0.590179	0.673554	0.491833	0.618806	0.608696	0.50838	0.668636	0	1	0	0	1	0	
0.629839	0.68595	0.618875	0.607545	0.687099	0.490697	0.626292	0	1	0	0	1	0	
0.804533	0.795455	0.907441	0.706644	0.926586	0.282269	0.768095	0	1	0	0	1	0	
0.588291	0.640496	0.639746	0.629505	0.610121	0.421134	0.650911	0	1	0	0	1	0	
0.583569	0.663223	0.505445	0.578829	0.575909	0.540236	0.628262	0	1	0	0	1	0	
0.635505	0.72314	0.470054	0.655968	0.550962	0.39773	0.690793	0	1	0	1	0	0	
0.955619	0.995868	0.618875	0.945946	0.843906	0.479255	0.951256	0	1	0	0	1	0	
0.78848	0.842975	0.607078	0.870495	0.719173	0.558959	0.907435	0	1	0	0	1	0	
0.616619	0.64876	0.735935	0.535473	0.667142	0.272127	0.604136	0	1	0	0	1	0	**94%**
0.560907	0.605372	0.673321	0.54955	0.596579	0.61981	0.670113	0	1	0	0	1	0	
0.767705	0.780992	0.813067	0.623311	0.874555	0.592765	0.669621	0	1	0	0	1	0	
0.90746	0.92562	0.73775	0.780405	0.879544	0.573132	0.82127	0	1	0	0	1	0	
0.84797	0.894628	0.633394	0.836149	0.81397	0.091914	0.863614	0	1	0	0	1	0	
0.842304	0.88843	0.634301	0.826014	0.83464	0.285649	0.820286	0	1	0	0	1	0	
0.725212	0.760331	0.715971	0.717342	0.727726	0.218167	0.826194	0	1	0	0	1	0	
0.782814	0.795455	0.805808	0.66723	0.808268	0.114928	0.782866	0	1	0	0	1	0	
0.792257	0.878099	0.461887	0.929054	0.741269	0.380437	0.974397	0	1	0	0	1	0	
0.234183	0.311983	0.362069	0.322635	0.259444	0.590165	0.431315	0	0	1	0	0	1	
0.25779	0.316116	0.482759	0.361486	0.315752	0.815236	0.453471	0	0	1	0	0	1	
0.259679	0.318182	0.489111	0.275901	0.316465	0.680011	0.387986	0	0	1	0	0	1	
0.153919	0.188017	0.518149	0.182995	0.2402	0.611619	0.345643	0	0	1	0	0	1	
0.116147	0.204545	0.175136	0.233671	0.104775	0.481855	0.324471	0	0	1	0	0	1	
0.058546	0.14876	0.07804	0.213964	0.040627	0.702636	0.37223	0	0	1	0	0	1	
0.07932	0.14876	0.23049	0.155968	0.063435	0.189302	0.301822	0	0	1	0	0	1	
0.179415	0.216942	0.523593	0.207207	0.2402	0.475354	0.237814	0	0	1	0	0	1	
0.199245	0.268595	0.372051	0.274212	0.200285	0.324396	0.392418	0	0	1	0	0	1	
0.018886	0.107438	0.023593	0.23536	0.01283	0.610709	0.332349	0	0	1	0	0	1	**100%**
0.117092	0.169421	0.376588	0.204955	0.149679	0.575992	0.387986	0	0	1	0	0	1	
0.134089	0.229339	0.15245	0.28491	0.104063	0.809645	0.369769	0	0	1	0	0	1	
0.157696	0.245868	0.228675	0.286599	0.14469	0.518912	0.414082	0	0	1	0	0	1	
0.055713	0.130165	0.167877	0.180743	0.044904	0.333758	0.237322	0	0	1	0	0	1	
0.07271	0.132231	0.27314	0.155405	0.089095	0.426855	0.366322	0	0	1	0	0	1	
0.056657	0.132231	0.15608	0.197635	0.032074	0.656347	0.344658	0	0	1	0	0	1	
0.070822	0.095041	0.467332	0.086712	0.156094	0.335708	0.238306	0	0	1	0	0	1	
0.14542	0.272727	0	0.278716	0.081967	0.527884	0.34515	0	0	1	0	0	1	

4. CONCLUSION AND SUGGESTIONS

An artificial neural network classification software was developed in this study using .NET platform to eliminate the difficulty and complexity of use of the current software in artificial neural networks' training process. The developed artificial neural network classification training software was applied to three wheat type classification case study and successful results were obtained. The software can be successfully used in all studies when necessary input and output values are introduced to the system properly. Hence this highlights the important aspect of the developed instructive artificial neural network software.

The training of ANN was performed by using backpropagation algorithm in the developed training software. Data set for three wheat type was used for the training of ANN. Then, the trained ANN software was tested for three wheat type classification.

Finally, the developed software was applied to three types wheat classification and produced a successful classification by completing the training process in nearly 40 minutes with 285 steps at an error rate of 5,678281E-05. The success of the developed software was determined as 100% for the first type of wheat, 94% for the second type of wheat and 100% for the third type of wheat. Average of classification success rate for the developed software was calculated as nearly 98%. Additionally, the software has a sufficient level and flexibility that the ANN software can be used for other studies. The application area of the software can be improved by integrating other learning algorithms into the ANN software prepared for futurework.

5. REFERENCES

CireşAn, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32, 333-338.

Elmas, Ç. (2003). Yapay Sinir Ağları. *Seçkin Yayıncılık*, Ankara, 27-37.

Elmas, Ç. (2007). Yapay zeka uygulamaları. *Seçkin Yayıncılık*, Ankara, 379-401.

Fausett, L., & Fausett, L. (1994). Fundamentals of neural networks: architectures, algorithms, and applications (No. 006.3). *Prentice-Hall*.

Gani, A., Keçecioğlu, Ö. F., Açıkgöz, H., & Şekkeli, M. (2015). Uyarlamalı Bulanık-PI Denetim Esaslı Dinamik Senkron Kompanzatör ile Reaktif Güç Kompanzasyonu Benzetim Çalışması. *Kahramanmaras Sutcu Imam University Journal of Engineering Sciences*, *18*(2), 72-76.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). Neural network design, *PWS Pub. Co*., Boston, 3632.

Haykin, S. (1994). Neural networks: a comprehensive foundation. *Prentice Hall PTR*.

Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S. (2009). Neural networks and learning machines (Vol. 3). Upper Saddle River, NJ, USA: *Pearson*.

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3), 215-236.

Kandil, N., Khorasani, K., Patel, R. V., & Sood, V. K. (1993, September). Optimum learning rate for backpropagation neural networks. In Electrical and Computer Engineering, 1993. Canadian Conference on (pp. 465-468). IEEE.

Kılıç, H. B. (1998). Yapay Sinir Ağları İle Karakter Algılama. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Nabiyev, V. V. (2005). Yapay zeka: problemler-yöntemler-algoritmalar. *Seçkin Yayıncılık*.

Sagiroglu, S., Besdok, E., & Erler, M. (2003). Mühendislikte Yapay Zeka Uygulamaları-I: Yapay Sinir Ağları. *Ufuk Yayıncılık*, Kayseri.

Tekin, A., & Gökbulut, M. (2008). Yapay Sinir Ağları ile Asenkron Motorların Hız Kontrolü için Bir Eğitim Yazılımın Geliştirilmesi. *Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 20(3), 449-453.

Uğur, A., & Kınacı, A. C. (2006). Yapay zeka teknikleri ve yapay sinir ağları kullanılarak web sayfalarının sınıflandırılması. XI. Türkiye'de İnternet Konferansı (inet-tr'06), Ankara, 1-4.

URL-1:Internet: Online.http://www.yildiz.edu.tr/~gulez/3k1n.pdf, (2017).

Witoonchart, P., & Chongstitvatana, P. (2017). Application of structured support vector machine backpropagation to a convolutional neural network for human pose estimation. *Neural Networks*, 92, 39-46.