



Development of Web Based Courseware for Artificial Neural Networks

Mehmet BILEN^{1,*} , Ali Hakan ISIK² , Tuncay YIGIT³ 

¹ Mehmet Akif Ersoy University, Cavdir Vocational High School, 15000 Burdur, Turkey

² Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, 15000, Burdur, Turkey

³ Suleyman Demirel University, Faculty of Engineering, 32000, Isparta, Turkey

Highlights

- This paper focuses on developing a web based educational courseware.
- Artificial neural network procedures can be performed and results can be traced in the courseware.
- Courseware has the ability to change initial parameters and observe outputs in real time.
- Students can understand better the working mechanism of ANN with the help of the courseware.

Article Info

Received: 22/10/2018
Accepted: 05/06/2019

Keywords

Artificial Neural
Networks
E-Learning
Courseware

Abstract

Artificial Neural Networks (ANN) are important data processing algorithms which are used for solving nonlinear problems. Through classical approaches, mathematical infrastructure and complex equations in ANN are difficult to understand. Interactive and multimedia-based courseware has the potential to overcome these difficulties. In this study, a web based educational courseware for ANN was developed to provide an effective and efficient learning environment so that the difficulties can be overcome. This interactive courseware was also enriched with animations and text-based course contents. In addition to this, the effects of ANN parameters' changes were observed directly through graphical results. In this way, users can easily understand the fundamentals and working mechanism of ANN. Without using any commercial libraries, the courseware was developed with ASP.NET, an object-oriented programming language. The courseware supports file formats such as XML, TXT, and CSV so that it can co-operate with other software. "Balance and Scale" data set was used to evaluate the performance of the courseware. 0.9918 accuracy, 1 specificity and 1 sensitivity values were achieved. When this study is compared to previous studies, improvements in terms of visuality, understandability and interactivity can clearly be identified.

1. INTRODUCTION

ANN are iterative algorithm that import, store and process experimental information [1]. An ANN is a highly parallel processor, consisting of simple process units with a natural capability for storing and delivering experiential information. [2]. On the other hand, ANN are computer programs which mimic working mechanism and problem solving skills of biological neural networks. Thanks to these features, ANN can easily and effectively solve classification, prediction, optimization, associative memory, pattern recognition, and control problems [3-5].

Therefore, ANN have increasingly been included in the curriculum of many engineering courses at graduate and undergraduate level. Due to complex and mathematical nature, it is not easy to understand for students who study engineering. Moreover, ANN evolve during their design and training phases thanks to its dynamic structure [6]. Conventional teaching approaches such as classroom based teaching, books, lecture notes do not help students enough to learn fundamentals and working mechanisms of ANN. However, computer based teaching approaches like simulation software provide students a user-friendly and understandable environment to visualize the working mechanism, design, and training phases of ANN.

This study is organized in four sections. In the first section, the learning difficulties for fundamentals and working mechanism of ANN within the scope of classical methods were presented. In the second section, the literature regarding educational artificial neural network software was evaluated. In the third section under the name

material and method, information about developed web based courseware for ANN was given. In the fourth section, the results and a discussion about the system was presented. Finally, the conclusion and recommendations for future works related to the development of web based courseware using artificial neural networks were given.

2. RELATED WORKS

Numerous commercial and educational software developed for designing and developing artificial neural networks are available in the market.

2.1. Commercial Software

MATLAB is a high-level technical computing language and environment that includes many tools for developing algorithms, analysing data, calculating complex mathematical formulas [7]. Beside these tools, MATLAB has also an ANN tool under the name of “Neural Network Toolbox” that provides adequate environment for ANN projects. It supports the most popular ANN models and related learning algorithms.

Statistica is a kind of statistics and analysis software which also has many different integrated tools [8]. Similarly, Statistica has a tool for ANN, like MATLAB. This tool provides an easy-to-use environment for many widely-used ANN models. Besides this, it helps users to design and develop suitable ANN models for solving their own problems.

NeuroSolutions is a graphical neural network development tool that allows users to create an ANN model for specific user data [9]. NeuroSolutions has a lot of advanced input and attribute selection methods such as Back-Elimination, Greedy Search, Genetic Optimization and Levenberg-Marquart algorithm.

Mathematica is another software that is used by students and professionals to design and develop their own ANN models [10]. This software also supports the most common ANN models such as MLP, SOM and Hopfield Network.

JOONE is open source development software which was developed in Java programming language [11]. It consists of a core engine, a GUI editor and a distributed training environment. Programmers can extend the features of the software by adding new modules, ANN models, and new learning algorithms.

2.2. Educational Software

NeuroFuzz is an internet-based ANN simulation tool that users can access anywhere in the world with a web browser [12]. The main aims of this tool are to teach the working mechanism and fundamentals of ANN by giving tasks to users and to allow them to solve problems with the support of graphical features. The tool tries to teach ANN by using classical methods without multimedia based approaches such as animations.

In [13], the author presents a program that was developed for teaching the basics of ANN. The effects of learning rate, momentum terms, epoch count, and number of hidden neurons can be observed by this software during training process, nevertheless the software does not provide students a suitable environment to gain knowledge and experience by themselves. It doesn't support different interfaces that are required to solve different problems. Users can not access detailed results after training and test phrases. It makes hard to understand the fundamentals and working mechanism of ANN properly. An effective and useful interface for ANN functions was developed and designed by authors in [14]. Iteration, momentum, learning rate, and activation function can be changed and results are observed through this interface. The effects of changing parameters can be observed through the graphics the program has. Although it was developed for educational purpose, an initial training is required before using the interface. Because of the fact that the interface was designed in a non-modular interface, its usage is very complicated.

EasyLearnNN is an Internet based simulation software that was developed for ANN education and it supports the most popular ANN algorithms like MLP, SOM, LVQ [15]. The working environment in the software used to design and develop ANN represents all parts of the ANN which have a graphical interface to allow students to

use the software in an intuitive and enjoyable way, but the software does not include text-based lecture notes for the ANN.

Neuro-Lab is a software-based environment used to simplify the teaching of ANN. She supports exploratory education in artificial intelligence classes [6]. It has an environment where all the features of a network can be changed by graphical interface and changes can be observed. Preliminary information is needed for students before using this environment. There is not text based course content and animations in the environment.

Yapay Sinir Aglari Egitim Materyali is an educational tool for ANN which was developed in object oriented approach [16]. The tool has modules to add layers and neurons as well as modules to select activation function and related learning algorithms. It allows users to trace every step of training and test phases. Users can learn and gain experience of ANN by this software with effective and exploratory learning techniques. Since the tool is not web based, it does not support remote access.

In the literature, the aim is to provide a better understanding of biological neural networks, but also in studies that allow the production and use of artificial neural networks [17,18].

Although some of these are professional programs, most of commercial programs are not suitable for educational purpose due to their complex design, distractive interface, and a lot of irrelevant knowledge features. As the aim of the study was focused on education, educational materials were used differently from commercial programs. Furthermore, unlike the commercial programs that host many algorithms, only the ANN algorithm is focused. Thus, it is aimed to prepare a more useful and uncomplicated training environment for users to learn of ANN. On the other hand, educational tools are effective to learn working mechanism and fundamentals of ANN. Nevertheless, as mentioned above, each of these tools has different deficiencies such as inadequate in the variety of educational materials, inadequate support by visual elements, real-time tracking and remote access. It is aimed to design and develop a program that combines the advantages of different related programs by eliminating these shortcomings and helping users easily learn the ANN.

3. MATERIALS AND METHODS

3.1. Design

During the design stage of the courseware, the ASP.NET platform was used by taking advantage of Visual Studio 2015, an application development environment. Therefore, the C# language was preferred thanks to its strong and effective object-oriented programming features. In addition to this, current and popular programming languages and approaches such as Ajax with JavaScript were included the project. The architecture of the developed system is shown in Figure 1.

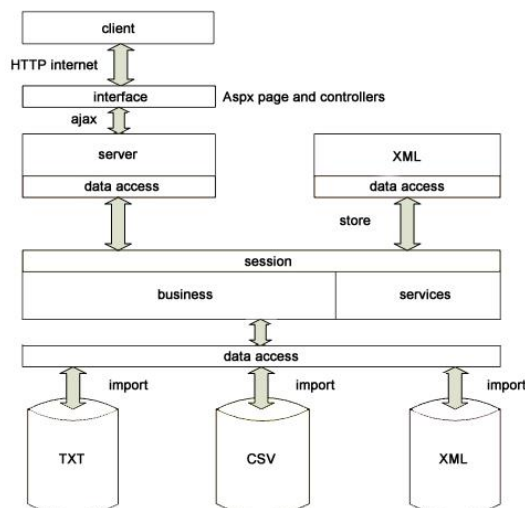


Figure 1. The basic model of the developed system

There is an ASPX based master page that provides placeholders to keep ASP.NET and HTML controls. This page includes contents and required controls for simulation. Processes and feedbacks take place at the backend, which is a part of ASPX page. In traditional approaches, if users interact with the page, the whole page is sent to the server. After server completed process, the whole page is sent to client again. This costs a huge data transfer traffic. Data transfer traffic is of great importance since the courseware works on the internet. Refreshing the page for interaction may also cause loss of settings or model information. In addition, it may cause losing users' attention with continuous updating of the page. Therefore, Ajax objects were utilized with JavaScript programming language. In this way, data exchange between the client and server can be done without refreshing whole page. Ajax solves the problem without having to load the page as a whole by pulling the data of the controls from the server where it is necessary to make the change. Web Access controls and contents without refreshing page increases visibility, usability, interactivity and also reduces data transfer traffic.

The business layer in Figure 1 is one of the most important parts of the developed system. The working mechanism and learning algorithms of ANN and a lot of other necessary features are stored in this layer with classes. It provides a suitable ANN training and simulation environment and it is shown in Figure 2. These classes that make up the core of the system have different and separate tasks from simulation to calculation. Thus, changes in any classes do not effect to any other class.

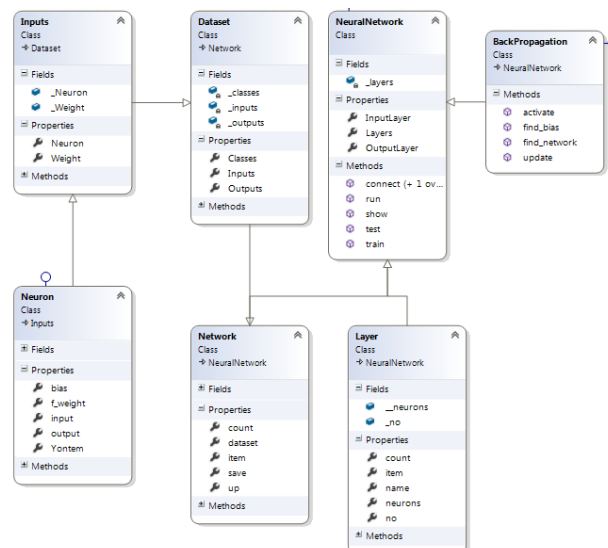


Figure 2. The class diagram of the system

Since the neurons of ANN models are stored in this class, the class named “Neuron” is one of the most basic classes. Instances of *Neuron* class are created for each neuron in the model. All the information about the neurons are kept as variables within these classes. Training and test data sets are kept in “*Network*” class through “*Inputs*” and “*Dataset*” classes. These classes perform different tasks such as import, export, and process data and they work on Data Access Layer. These also support the use of CSV, TXT, and XML file formats as well as the conversion from one file format to another. “*NeuralNetwork*” is a class which is used for training and test phase simulation by utilizing calculations of “*BackPropagation*” class. Besides these classes, there is also a class named “*Services*” which is used for different operations such as show animations or course contents.

3.2. Development

In the developed system, a multilayer ANN model is designed using back propagation algorithm, a popular learning algorithm. Back propagation algorithm is used for calculating the weights in a multilayer ANN. It is effective for handling complex networks which have lots of neurons and weights. Thanks to development of this algorithm, multilayer ANN becomes one of the most commonly used algorithm for YSA models [19]. Back propagation algorithm was implemented by making use of following equations.

Net values of each neuron are obtained by the following Equation (1)

$$Net_j^H = \sum_{k=1}^n W_{kj}^{IH} \cdot O_k^I, \quad (1)$$

where j means the neuron number in the hidden layer, k is the output numbers, W_{kj}^{IH} represents weights from every k outputs to the j neuron in the hidden layer. O_k^I is the value of k output.

Outputs of each neuron in the hidden layer can be defined by Equation (2):

$$O_j^H = \frac{1}{1 + e^{-(Net_j^H + \beta_j^H)}}. \quad (2)$$

Here, β_j^H means the bias value for j neuron in the hidden layer. Changes in quantities of weights are obtained by the following Equation (3)

$$\Delta W_{jm}^{HO}(t) = \gamma \delta_m O_j^H + \alpha \Delta W_{jm}^{HO}(t-1), \quad (3)$$

where γ means the learning rate, α is the momentum, j is the neuron number in the hidden layer, m is the neuron number in the output layer, and t is the time. In addition, δ_m is calculated by Equation (4):

$$\delta_m = O_m^O(1 - O_m^O)(E_m - O_m^O), \quad (4)$$

O_m^O means the output of each m neuron in the output layer. E_m is the expected output value of each m neuron in the output layer.

3.3. Main Features of The Developed Courseware

The software was designed and developed as a web page in modules to provide simple and easy to use interface that does not disturb users. As can be seen in Figure 3, these modules are data set process, simulation, documentation, and animation modules.

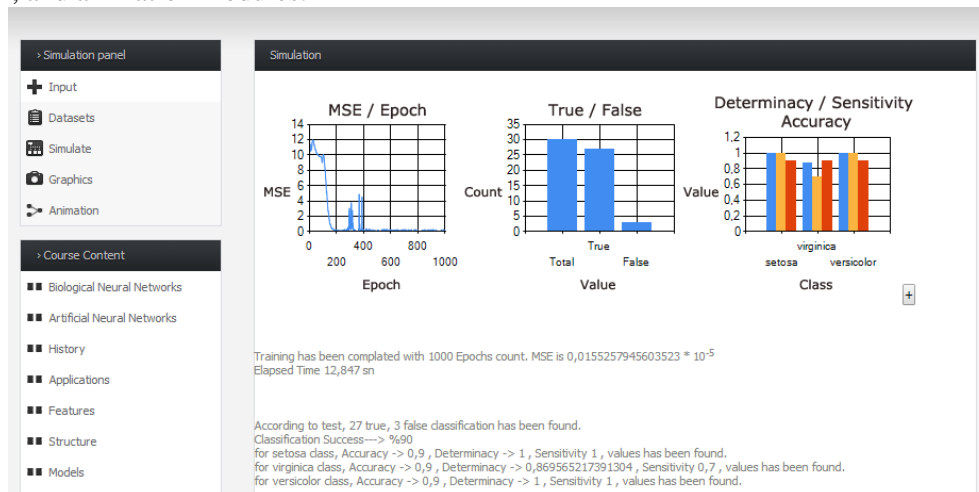


Figure 3. General appearance

The courseware also has controls to import and manage data sets in the Data Set Process module. Data sets can be imported for training and test phases through these units. The developed software supports the most popular file formats such as .CSV, .TXT and .XML to increase usability and interoperability of the courseware.

ANN Simulation Module is an important module for users to understand fundamentals and working mechanism of ANN. By using this module, users can change the parameters of ANN. In this way, users can understand the importance of each parameter through observing the output. A screenshot of this interface is shown in Figure 4.

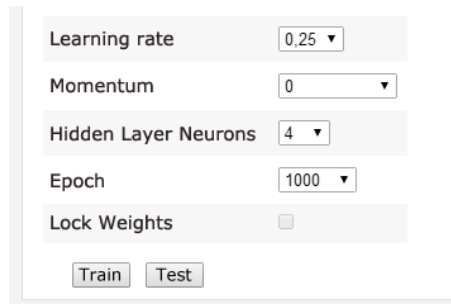


Figure 4. Interface for selection of parameters

The ANN training process is carried out through update of weights among neurons. However, it is very important to select the initial values of the parameters during the training phase because the effect of the parameters associated with the output cannot be predicted. Due to the dynamic nature of ANN, the effect of the parameters associated with the output parameters may vary. Therefore, it is not easy to teach the ANN with traditional techniques. The courseware allows users to change parameters such as earning rate, momentum, neurons count, and epoch count through the controls in the simulation module. Thanks to the graphics module, the response of network under different learning rate and momentum values can easily be seen by users. The difference in the training graph of the network with the increase of the learning rate is shown in Figure 5. The change in the training graph of the network because of the increase in the momentum is shown in Figure 6.

Users can access text-based content that include the fundamentals of ANN through the Documentation Module. The courseware also contains multimedia material. For instance, an animation of a simple ANN can be seen in the Animation Module. Users can track the phases of an ANN both in back propagation and forward feed. In addition, they can understand how changes in the weights and in the bias values affect the training of the ANN through the animations.

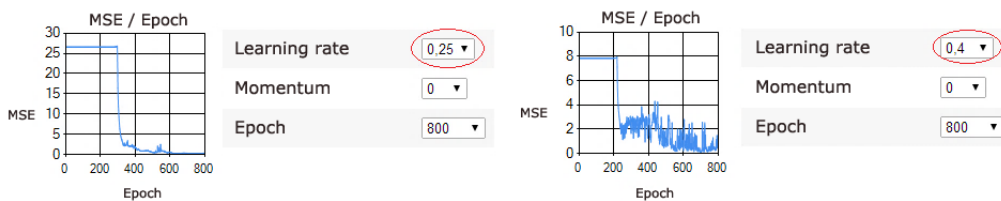


Figure 5. Results according to different learning rate values

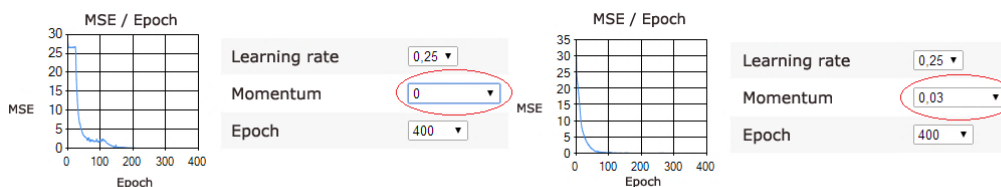


Figure 6. Results according to different momentum values

4. RESULTS

4.1 Performance Analysis

“BalanceAndScale” data set which was defined by psychological experience model results was chosen to evaluate the performance of the network [20]. There are 614 samples in this dataset. Each sample in the dataset belongs to one of these classes: Balanced (B), Right (R), or Left (L). Each sample has 5 properties. These properties are left weight, left distance, right weight, right distance, and a class property. 80 percent of the samples in the data set are reserved for training phase and the others are reserved for test phase. Almost equal samples are taken from each class. The distribution of samples on training and test data sets are shown in Table 1.

Table 1. The Distribution of Samples

Class	Data Set Samples	Learning Data Set	Test Data Set
B	47	37	10
L	284	228	56
R	283	227	56

The ANN algorithm was run with different initial values to obtain the highest performance values. The results of training and test phases are shown Table 2 and Table 3. MSE / Epoch graphics which were created by the results are shown in Figure 7.

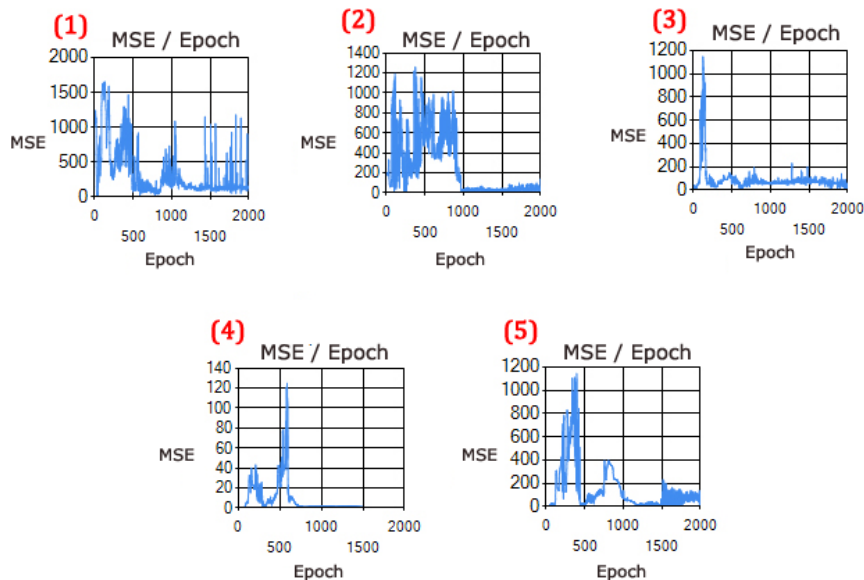


Figure 7. Graphical results of learning phrases

Thanks to MSE / Epoch graphics, the effects of changes in parameters of the outputs can be observed. It was realized that processing capacity increases up to a point with the increasing number of neurons in the hidden layer. If the numbers of neurons in the hidden layer increase, the processing time increases significantly. It was also noticed that huge learning rates lead to fast or slow training in some circumstances. However, there is no direct relationship between the learning rate and the processing time.

It was observed that the results of the performance tests of B class are worse than other classes because of the insufficient amount of dataset samples. Therefore, some of the examples belonging to the B class aren't identified successfully.

According to the results of the performance tests, high rate of classification and high-performance values are observed with only one failure in the fourth network. Classification results of test samples that were used in the training process prove that this network is trained.

Table 2. Performance results obtained from trained networks

Network number	Accuracy for all classes	Determinacy			Sensitivity		
		B	L	R	B	L	R
1	0,9262	0,9724	0,9672	0,9384	0,7	0,9642	0,9285
2	0,9590	0,9908	0,9841	0,9552	0,9000	0,9821	0,9464
3	0,9836	0,9910	1	0,9848	0,9000	1	0,9821
4	0,9918	1	0,9850	1	1	0,9821	1
5	0,9508	0,9649	0,9687	1	0,6	0,9642	1

Table 3. Success rate and time consumption of networks

Network Number	Epoch	Learning Rate	Neurons count in the hidden layer	True	False	Success	MSE (*10 ⁻⁵)	Total Elapsed Time (sec)
1	2000	0,30	3	113	9	%92,62	1,1783114	28,61
2	2000	0,30	5	117	5	%95,90	0,3276797	72,08
3	2000	0,30	10	120	2	%98,36	0,0285055	173,85
4	2000	0,30	20	121	1	%99,18	0,0203843	459,67
5	2000	0,20	20	116	6	%95,08	0,5657506	487,08

In the experiment, it was observed that the optimum values for learning rate, momentum, and neuron counts of the hidden layer for each dataset can vary. In order to choose the most appropriate values, trial-error methods or intuitive methods must be preferred instead of linear ones. Thanks to the graphical interface, users can find the optimum ANN parameters without changing source codes.

4.2. The Questionnaire

The developed courseware was used in Mehmet Akif Ersoy University Çavdır Vocational School/ Turkey for two semesters. After each semester, the students were asked to complete the questionnaire, which can be seen in Table 4. The options of the questionnaire were in a scale from 1 to 5. These answers are Strongly Disagree, Disagree, Neutral (neither agree nor disagree), Agree and Strongly Agree. The questionnaire was answered by 65 students. According to the statistical analysis of the questionnaire results, the mean value obtained was 3,76. This result indicates that the courseware helped students understand the working mechanism of ANN.

Table 4. Questionnaire applied to students after using the developed software

Questions	Average	Standard deviation
I can access desired information through ANN educational courseware whenever i want.	3,968750	0,951459
ANN courseware saves my time to access learning resources.	3,812500	0,826797
I use ANN educational courseware to pass examination	3,718750	0,819084
I clearly understood ANN subject with the help of the ANN educational courseware.	3,765625	0,914163
ANN courseware increased my motivation for the lesson.	3,687500	0,899218
I use ANN educational courseware to repeat the ANN subjects.	3,765625	0,896908
ANN courseware is easy to use and user friendly.	3,843750	0,887742

As seen in Table 5, an Independent sample t-test was performed so as to find whether or not there is a significant statistical differentiation among the questionnaire answers in terms of gender. In our study, significance level was accepted as 0.05. It was found that all significance values (p) are higher than 0.05. Therefore, it was determined that there is not a meaningful differentiation between males and females. It was found that females and males are equally satisfied with the courseware.

Table 5. T test results according to the questionnaire answers in terms of gender

Questions	Gender	Number	Std. Error Mean	t	p(Sig.)
I can access desired information through ANN educational courseware whenever I want.	Male	33	.17058	-.510	.532
	Female	31	.17029		
ANN courseware saves my time to access learning resources.	Male	33	.15764	.354	.115
	Female	31	.13686		
I use ANN educational courseware to pass examination	Male	33	.14071	.994	.599
	Female	31	.15153		
I clearly understood ANN subject with the help of the ANN educational courseware.	Male	33	.16494	1.292	.938
	Female	31	.15847		
ANN courseware increased my motivation for the lesson.	Male	33	.16872	.635	.210
	Female	31	.15153		
I use ANN educational courseware to repeat the ANN subjects.	Male	33	.15377	-.624	.800
	Female	31	.16782		
ANN courseware is easy to use and user friendly.	Male	33	.15302	.600	.959
	Female	31	.16533		

5. CONCLUSION

In this study, a web based courseware which is supported with animations and contents was designed and developed for artificial neural networks. Artificial neural network procedures such as training, test, and simulation can be performed in this courseware and results can be traced through graphics. The courseware includes easy to use interface which can be accessed all over the world through the internet. It provides an environment which can be used by anyone who has interest in artificial neural networks without having fundamental foreknowledge. Thus, exploratory learning is supported. The study combines the advantages of previous software. If this study is compared to the previous studies, it became clear that there are improvements in the fields of visuality, understandability and interactivity. Unlike commercial software, the study focused on the ANN education. The deficiencies in the educational software were eliminated and contribution was made to the educational activities. The developed software is an alternative to and supporter of similar software.

Real Time changing of ANN parameters and observing the effects on the output through the web page allows obtaining high performance results. The ability to change parameters which are used in ANN training and the ability to observe different outputs in real time make it easy to choose optimum parameters. As a result, test results were for accuracy, determinacy, and sensitivity are 0.998, 1, and 1, respectively. It was also concluded that the ability to observe changes in real-time helped students understand the working mechanism and fundamentals of ANN.

Innovative technologies such as Web 2.0, Ajax controls, and XML are used in the courseware. The courseware was based on an object oriented programming language, C#. These features make the courseware fast, powerful, flexible, interoperable with other popular software for future works. The courseware was also designed in a way that new algorithms can be attached and also updates can be patched easily. In addition, design independent coding makes the software fast to transfer among different platforms such as mobile platforms and desktop platforms.

Developed courseware was used in Mehmet Akif Ersoy University Cavdir Vocational School / Turkey for two semesters. Then a questionnaire was given to the students. The result of the questionnaire shows that students understand the fundamentals and working mechanism of ANN with the help of the courseware.

Different user levels and student tracking module will be added in the next versions of the courseware to make ANN education more effective and more efficient. Database support will be added to evaluate findings such as students' assignments, learning processes, and results. Therefore, it is planned to obtain a virtual laboratory that uses all advantages of web based systems.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

- [1] Masters, T. Practical neural network recipes in C++, Morgan Kaufmann, (1993).
- [2] Simon, H. Neural networks: a comprehensive foundation. New Delhi, Prentice-Hall of India, (2008).
- [3] Blanton, H. "An introduction to neural networks for technicians, engineers, and other non-PhDs," Proceedings of the 1997 Artificial Neural Networks in Engineering Conference, St.Louis, (1997).
- [4] Fulcher, J. "Experience With Teaching a Graduate Neural Networks Course", Computer Science Education, 3(3):297-314, (1992).
- [5] Fulcher, J. "Laboratory Support for The Teaching of Neural Networks", International Journal of Electrical Engineering Education, 35(1),:29-36, (1998).
- [6] Roselló, E. G., Pérez-Schofield, J. B., Dacosta, J. G., & Pérez-Cota, M. "Neuro-Lab: A highly reusable software-based environment to teach artificial neural networks", Computer Applications in Engineering Education, 11(2):93-102, (2003).
- [7] MATLAB [Computer software]. Natick, MA: MathWorks, (2006).
- [8] Statistica [Computer software]. Tulsa, OK: Statsoft, (1997).
- [9] NeuroSolutions [Computer software]. NeuroDimension, (2015).
- [10] Mathematica [Computer software]. Champaign (IL): Wolfram Research, (1997).
- [11] JOONE [Computer software]. Paolo Marrone and the Joone team, (2015).
- [12] Manic, M., Wilamowski, B., & Malinowski, A. "Internet based neural network online simulation tool", 28th Annual Conference of the Industrial Electronics Society, Sevilla, 4(4):2870-2874, (2002).
- [13] Venayagamoorthy, G. K. "Teaching Neural Network Concepts and Their Learning Techniques", Proceedings of the 2004 American Society for Engineering Education Midwest Section Conference, Pittsburg, (2004).
- [14] Bayindir, R., Sesveren, O. "Design of Visual Interface for ANN Based Systems", Pamukkale University Journal Of Engineering Sciences, 14:101-109, (2008).

- [15] Ugur, A., Kinaci, A. C.” A web-based tool for teaching neural network concepts”, *Computer Applications in Engineering Education*, 18(3):449-457, (2009).
- [16] Deperlioglu, O., Kose, U. “An educational tool for artificial neural networks”, *Computers & Electrical Engineering*, 37(3):392-402, (2011).
- [17] Dragly, S. A., Hobbi Mobarhan, M., Våvang Solbrå, A., Tennøe, S., Hafreager, A., Malthe-Sørenssen, A., Fyhn, M., Hafting, T., Einevoll, G. T. “Neuronify: An Educational Simulator for Neural Circuits”, *eNeuro*, 4(2):17, (2017).
- [18] Cope, A.J., Richmond, P., James, S.S. et al., “SpineCreator: a Graphical User Interface for the Creation of Layered Neural Models”, *Neuroinform*, 15-25, (2017).
- [19] Yegnanarayana, B. *Artificial neural networks*. New Delhi: Prentice-Hall of India, (2006).
- [20] Internet: Bache, K., & Lichman, M. Uci Machine Learning Repository, Balance and Scale Data Set, <http://archive.ics.uci.edu/ml/datasets/balance+scale> , (2017).