

# A neural network simulator for educational purposes

Ugur Kesen<sup>1</sup>, Sezgin Alsan<sup>2</sup>

Mechatronics Engineering Department, Faculty of Technology, Marmara University, Istanbul, Turkey

<sup>1</sup>Corresponding author

E-mail: <sup>1</sup>ukesen@marmara.edu.tr, <sup>2</sup>salsan@iticu.edu.tr

Received 13 March 2024; accepted 13 May 2024; published online 6 June 2024

DOI <https://doi.org/10.21595/jmai.2024.24085>



Copyright © 2024 Ugur Kesen, et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract.** Artificial neural networks are inspired by biologic processes. Artificial neural networks are important because they can be used to deduct a function from observations, in other words artificial neural networks can learn from experience. Artificial neural network simulator to fulfill a need into the growing interest of neural network education is introduced in this study. NeuroQuick Laboratory simulator is implemented using object-oriented programming by Delphi programming and these classes can be used to create a standalone application with artificial neural networks. The NeuroQuick Laboratory Simulator is designed for a broad range of users, including beginning graduate/advanced undergraduate students, engineers, and scientists. It is particularly well-suited for use in individual student projects or as a simulation tool in one- or two-semester neural network-related courses at universities.

**Keywords:** artificial neural networks, engineering education, simulation design.

## 1. Introduction

An artificial neural network (ANN) is a computational model that emulates biological information processing. Its fundamental ability lies in learning patterns from provided examples [1-2], as it was developed with inspiration from the systematic way the human brain [3]. This learning process involves adjusting the weights of interconnected nodes, typically done through a learning algorithm. The appeal of artificial neural networks extends to a diverse range of individuals, including computer scientists, engineers, cognitive scientists, neurophysiologists, physicists, biologists, philosophers, and others, each with their unique reasons for interest [4-6]. This broad appeal contributes to the popularity of neural network courses in universities.

Before an artificial neural network (ANN) can be effectively used for any practical application, it must undergo a training process using a dataset known as the training set [7-18]. This training set should ideally contain a diverse range of training cases that fully represent the problem being addressed [19]. The training set comprises two parts: the input data and the corresponding target output. Once the training process is completed, another dataset known as the validation set is used to evaluate the performance of the trained neural network. The validation set helps assess how well the neural network will generalize and perform with unseen data.

There are several dedicated simulation software programs available for artificial neural networks. The simulator discussed here is primarily designed for academic use but is also accessible to individuals with diverse backgrounds. It is relatively simpler compared to other available options. Specifically tailored for beginning graduate/advanced undergraduate students requiring software for individual projects, this simulator offers an easier selection of the appropriate neural network model for a given application, eliminating the need to become familiar with multiple independent simulators.

## 2. NeuroQuick laboratory simulator design

The NeuroQuick Laboratory Simulator is tailored for PC users, leveraging the Delphi program to enhance user-friendliness through its visual interface. This design choice enables easy installation and operation on any PC, making it accessible to individuals interested in neural

networks. This accessibility is particularly beneficial for students embarking on neural network projects.

The main menu structure of the NeuroQuick Laboratory Simulator includes the following items: File, Edit, Network Model, and Help in Fig. 1. Under the File menu, users can find options such as Open, New, Save, Print, and Exit. The Edit menu allows users to search within output data and terminate the program based on certain output values. It also includes a Clear All option. The simulator offers several neural network models, including perceptron, multilayer perceptron, Adaline, Hopfield, and Radial Basis Function networks. Each network model has its own menu for configuring the network and selecting adjustable parameters.

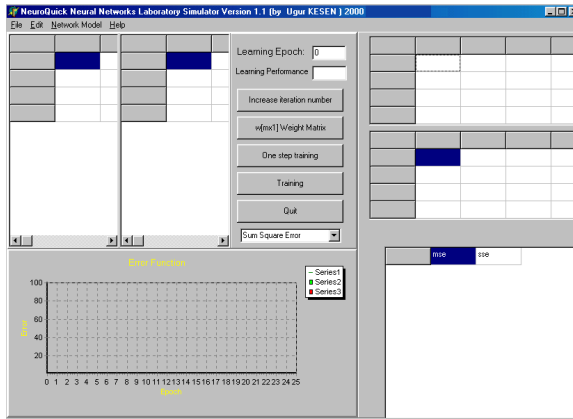


Fig. 1. The main menu structure

The Network Model menu item in the NeuroQuick Laboratory Simulator, as depicted in Fig. 2, allows users to access various neural network models. The Help menu, located at the end of the main menu window, provides users with assistance on using the simulator. Additionally, it offers general information about artificial neural networks to enhance the user's understanding.

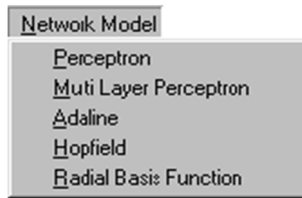


Fig. 2. Network model menu item

### 3. Configuration of network models

In the NeuroQuick Laboratory Simulator, users can select a neural network model for any given application from the Network Model menu item, as shown in Fig. 2 in the main menu. Upon selecting a model, a new menu is opened, which includes options such as Learning Data and Transfer Function. Fig. 3 depicts the Learning Data menu item for the Perceptron, Adaline, and Hopfield network models, while Fig. 4 illustrates the Learning Data menu item for the Multilayer Perceptron network model.

In the NeuroQuick Laboratory Simulator, users can determine the network configuration, including the size of the network and the weights of connections, by selecting the Input Vector and Weight options under the Learning Data menu item, as depicted in Fig. 5 and Fig. 6.

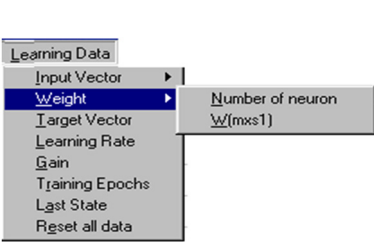


Fig. 3. The Learning Data menu item

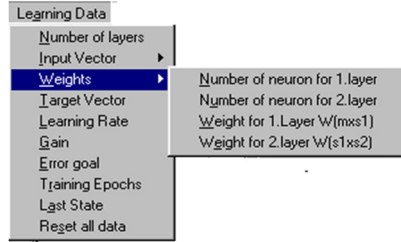


Fig. 4. The Learning Data menu item for multilayer perceptron

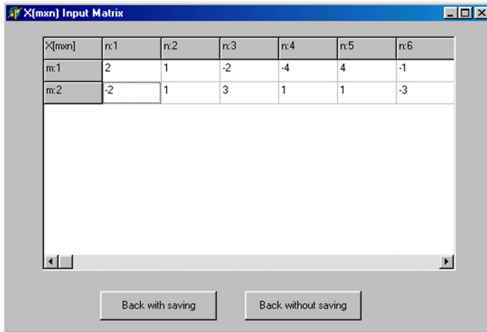


Fig. 5. Input vector selection box

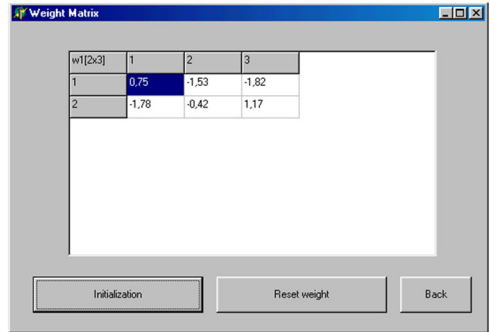


Fig. 6. Weight matrix selection box

In the NeuroQuick Laboratory Simulator, users can assign random values to the weight matrix by clicking on the “Initialization” button. Additionally, they can determine free parameters using the “Learning Rate” and “Gain” options. The desired outputs for the network can be specified using the “Target Vector” option, as shown in Fig. 7.

In the NeuroQuick Laboratory Simulator, users can stop training at a requested number of training epochs by specifying a certain number in the “Training Epochs” option. Additionally, all data information entered can be viewed using the “Last State” option, as shown in Fig. 8. Users can also reset all data using the “Reset all data”.

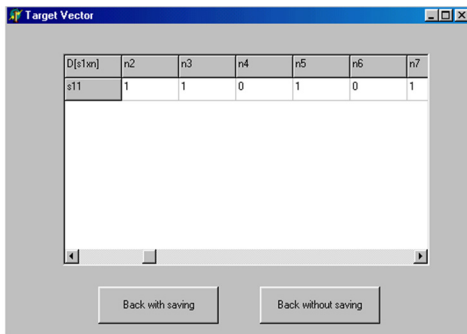


Fig. 7. Target vector selection box

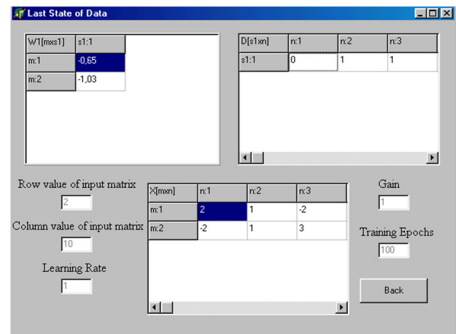


Fig. 8. The last state of data box

The Learning Data menu item for multilayer perceptron has a few more options. These are Number of layers and Error goal. One or two layers can be chosen. Furthermore, the Weights option allows to initialize the weight of the connections between the layers.

Various activation functions can be selected using Transfer Function menu item depending on the given application data. This item is given in Fig. 9.

An application example is given in Fig. 10. In this application, the network model has been chosen as a perceptron with two inputs and 10 patterns. Transfer function was hard limiting function. It was seen that the network converged to a desired output at 11 training epochs.



Fig. 9. Transfer function menu item

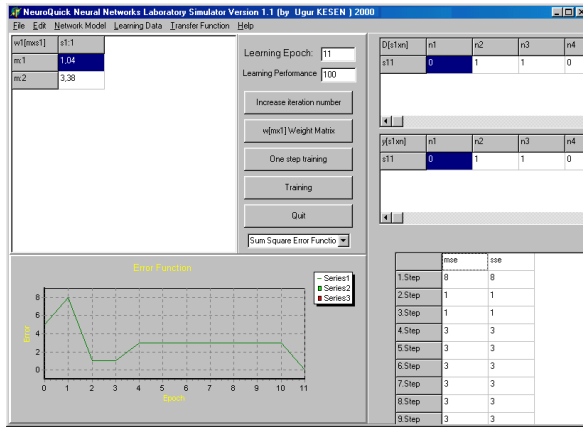


Fig. 10. Simulation example

#### 4. Conclusions

An artificial neural network is a computational technique with versatile applications across various fields. ANNs stand out for their ability to learn from datasets, making them invaluable in scenarios where a model is overly complex or nonexistent. This paper introduces NeuroQuick Lab, a simulator proposed for designing and testing artificial neural networks. The simulator has been testing in Neural Network and Applications graduate course at Yildiz Technical University. Adding certain well-known neural network databases and more network models, such as the models using unsupervised learning, to the simulator is in progress.

Using the activex function of Delphi programming, the NeuroQuick Lab simulator was enabled to run in internet explorer. In this way, this simulator can be used remotely on a web-based basis. Thus, students will be able to use and test ANN in distance education applications to better understand it. This simulator can be developed for different artificial intelligence algorithms in the future.

#### Acknowledgements

The authors have not disclosed any funding.

#### Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Author contributions

Ugur Kesen: formal analysis, investigation, resources, software, validation, visualization, writing-original draft preparation, writing- review and editing. Sezgin Alsan: conceptualization, data curation, methodology, project administration, supervision.

## Conflict of interest

The authors declare that they have no conflict of interest.

## References

- [1] R. J. Schalkoff, *Artificial Neural Networks*. Singapore: McGraw-Hill Inc., 1999.
- [2] J. M. Zurada, *Introduction to Artificial Neural Systems*. Boston, Massachusetts, The USA: PWS Publishing, 1995.
- [3] A. N. Camgöz, Z. Ekşi, M. F. Çelebi, and S. Ersoy, “Analysis of electroencephalography (EEG) signals and adapting to systems theory principles,” *Journal of Complexity in Health Sciences*, Vol. 4, No. 2, pp. 39–44, Dec. 2021, <https://doi.org/10.21595/chs.2021.22038>
- [4] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York, The USA: Macmillian College Publishing, 1994.
- [5] D. R. Hush and B. G. Horne, “Progress in supervised neural networks,” *IEEE Signal Processing Magazine*, Vol. 10, No. 1, pp. 8–39, Jan. 1993, <https://doi.org/10.1109/79.180705>
- [6] B. J. A. Kröse and P. van der Smagt, *An Introduction to Neural Networks*. Amsterdam, The Netherlands: The University of Amsterdam, 1996.
- [7] K. Gurney, *An Introduction to Neural Networks*. London, UK: UCL Press, 1996.
- [8] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. Boston, Massachusetts, The USA: PWS Publishing, 1996.
- [9] N. K. Bose and P. Liang, *Neural Networks Fundamentals with Graphs, Algorithms, and Applications*. Singapore: McGraw-Hill Inc., 1996.
- [10] A. Abraham, “Artificial neural networks,” in *Handbook of Measuring System Design*, London, UK: John Wiley & Sons, 2005.
- [11] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles Techniques and Tools*. Boston, Massachusetts, The USA: Addison-Wesley, 2006.
- [12] M. Karakaya, M. F. Çelebi, A. E. Gök, and S. Ersoy, “Discovery of agricultural diseases by deep learning and object detection,” *Environmental Engineering and Management Journal (EEMJ)*, Vol. 21, No. 1, pp. 163–173, 2022.
- [13] F. Amato, A. López, E. M. Peña-Méndez, P. Vañhara, A. Hampl, and J. Havel, “Artificial neural networks in medical diagnosis,” *Journal of Applied Biomedicine*, Vol. 11, pp. 47–58, 2013.
- [14] H. Cartwright, *Artificial Neural Networks. Methods in Molecular Biology*. New York, the USA: Humana, 2015.
- [15] K. D. Cooper and L. Torczon, *Engineering a Compiler*. Burlington, Massachusetts, The USA: Morgan Kaufmann Publishers, 2012.
- [16] M. W. Craven and J. W. Shavlik, “Learning symbolic rules using artificial neural networks,” in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 73–80, 1993.
- [17] R. S. Govindaraju and A. R. Rao, *Artificial Neural Networks in Hydrology*. Berlin, Germany: Springer Science & Business Media, 2013.
- [18] D. Graupe, *Principles of Artificial Neural Networks*. Singapore: World Scientific, 2013.
- [19] M. Kubat, *Artificial Neural Networks: an Introduction to Machine Learning*. Berlin, Germany: Springer Science & Business Media, 2015.



**Ugur Kesen** received a Ph.D. degree in Electronics and Telecommunications Education in The Institute of Pure and Applied Sciences from Marmara University, Istanbul, Türkiye, in 2001. Now he works at Marmara University. His current research interests include electronics, telecommunications, and computer science.



**Sezgin Alsan** received a Ph.D. degree in Nuclear Instrumentation from Grenoble University, Saint-Martin-d'Hères, France, in 1972. His current research interests include electronics, telecommunications, and electronics education.