Internet based Neural Network Online Simulation Tool

Milos Manic
Department of Computer Science
University of Idaho
800 Park, Blvd.
Boise, ID 83712, USA
misko@ieee.org

Bogdan Wilamowski
College of Engineering
University of Idaho
800 Park, Blvd.
Boise, ID 83712, USA
wilam@ieee.org

Aleksander Malinowski
ECE Department
Bradley Univ
1501 West Bradley avenue
Peoria, Illinois 61625
olekmali@ieee.org

Abstract – Since the legendary work of McCulloch and Pitts from early 1940-ies and introduction of concept of artificial neuron, numerous attempts aiming to automate the process of training neural networks have been made. Neural networks, even though successfully applied in many different areas still bear significant problems with respect to adequate choice of network parameters, architecture, etc. As the concept of neural networks evolved through more than six decades, the technology also underwent through tremendous changes. Educational tools aiming to help everyday users to learn, gain and apply knowledge in neural network training have evolved accordingly. Unlike many commercial or freeware tools available on market today, authors have decided to go with the implementation that would provide two significant advantages to an existing tools. In this paper authors are proposing an educational tool that is characterized by both transparent accessibility with respect to hardware and software platform on one, and ease of use on the other hand. The tool is web based, therefore user is relieved of installation and setup hurdles. Besides, by employing a remote server that hosts the application, user saves local resources for other jobs.

I. INTRODUCTION

Ever since McCulloch and Pitts have introduced the concept of artificial neuron [1] numerous attempts toward the tools that would automate the process of learning have been performed. Neural networks, even though proven as successful in various areas still cope with tedious problems of adequate choice of network parameters, architecture, etc.

As the concept of neural networks evolved through more than six decades, the technology also underwent through tremendous changes. Educational tools aiming to help everyday users to learn, gain and apply knowledge in neural network training have evolved accordingly. Unlike many commercial or freeware tools available on market today, authors have decided to go with the implementation that would provide two significant advantages to an existing tools.

In this paper authors are proposing an educational tool that is characterized by both transparent accessibility with respect to hardware and software platform on one, and ease of use on the other hand.

This work is supported by the NSF-Idaho EPSCoR Program and by the National Science Foundation under award number EPS-0132626.

The tool is web based, therefore user is free of problems related to download, installation, setup, even running an application on local computer. This is because web application is hosted on remote computer that does all the work.

As a consequence, user does not employ local machine, and therefore enjoys huge savings in local computational time and resources.

Tools available on market follow different ideas. One of the freeware available tools, "Stuttgart Neural Network Simulator" is based on widely available C platform and is distributed in both executable and source code version [2] However, just an installation of such tool requires certain knowledge with respect to compiling and setting up the application. Also, it is based on XGUI that is not freeware. Finally, it is limited to though powerful still single type architecture – Unix architecture.

Commercial neural network software called "Neurodesigner" is product of CyberSoft [3] Even though aiming to attract users by Java based windows interface, this tool lacks some fundamental algorithms, like Levenberg-Marquardt algorithm. With a commercial price it also places a hurdle on educational use of such a tool.

There is a variety of tools available online, however, with no intention for educational use. The reason is these tools have only a few learning algorithms implemented, or are targeted to solve a single problem. Most of those online tools were one time projects done for specific purpose, and have not been updated for years.

Tools for developing another systems based on fuzzy logic and neural networks technology [4], Artificial Neural Network Lab on the Web [5] demonstrated one method (Bayesian Self-Organizing Maps), Artificial Neural Network Lab on the Web (last updated in 1997) [6] solves XOR and 3D Kohonen problems, Ball Balancing Problem [7], Neural Network FN Predictions for Stainless Steel Welds [8], Nenet tool targeting the problem of Self-Organizing Map (SOM) [9].

List of these and some others are given by Pacific Northwest National Laboratory [10], Network Cybernetics Corporation [11].

The rest of the paper is organized as follows. Second section discusses accessing the application. The third section discusses running neural network tool (input, output parameters). The fourth section discusses help feature. The fifth section concludes this paper with directives for future work. Last, sixth section give references used in this paper.

0-7803-7474-6/02/$17.00 ©2002 IEEE

II. INSTALL, SETUP, EXECUTE – NOT IN THIS CASE!

Unlike others available education tools, once the user connects to a url of the described application by his/her popular internet browser, the process of installation, setup or any other related operation is over. This is due to the web based implementation of this educational tool (Fig. 1), that enables transparent, universal and cheap access. The URL of an application is: http://husky.engboi.uidaho.edu/nn/.

The application itself is accessible from any platform, regardless of hardware and operating systems architecture. The only required items are any kind of popular internet browser and the internet connection. Cookies should be enabled for fool utilization of this application.

The application interface is made of html, Javascript and Perl code, and computationally intensive tasks are for the sake of speed implemented in C language. The application is hosted on Apache server in Windows workstation environment. However, due to careful coding and inherent portability of both http server and interface software, application is easily portable to any Unix flavor platform.

A. User Levels in Neural Network Tool

Software differentiates among registered users. The purpose is not to restrict the usage and access to a tool, rather to optimize its usage.

There are 4 levels of registered users: novice, regular, student, and research type of user. There is also an option for unregistered users (guests) that are able to verify existing features in demo mode. Demo mode has all features visible, but none of them user is able to change (hardcoded).

The higher the user level, the more options are available. These options are larger possible number of iterations, smaller required error, larger interval of possible values for gain and learning constant, etc.

For example, novice users in neural network field will be able to successfully train simpler architectures with restricted set of parameters, and therefore easier gain valuable experience. On the other hand, higher level users will be able to experiment more with less "safe" parameters and go beyond known frontiers.

Options that are not restricted in any way are printing and plotting options, since these are always helpful and descriptive in learning process. However, for the sake of shorter calculation time, these options can be selectively checked.

III. USING NEURAL NETWORK TOOL

The educational tool for Neural Network training, *NeuroFuzz 6.22* is organized through 4 different options for

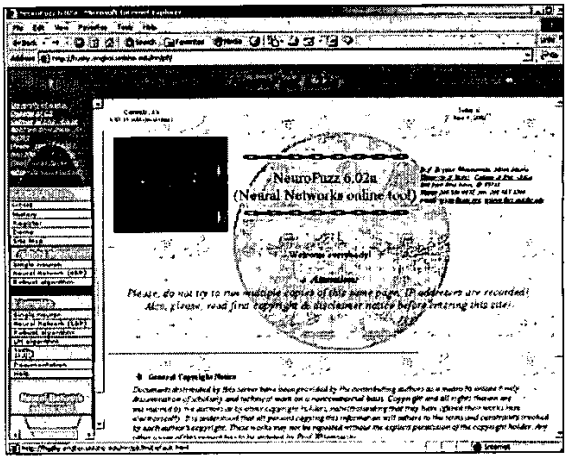


Fig. 1. Neural Network online tool – first screen (screen has been whitened to emphasize central portion of the screen as well as menu part).

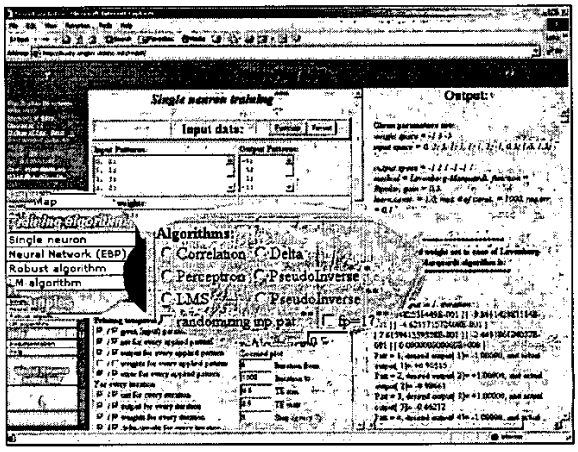


Fig. 2. Neural Network online tool - HTML help (the rest of the screen has been whitened to emphasize html help feature of the application).

training. Generally, user is given 2 options: single neuron and multiple layer network training.

For the single neuron training, following algorithms are available (Fig. 2) [13-14]:

- Correlation rule,
- Delta rule,
- Perceptron rule,
- LMS (Widrow-Hoff) rule,
- PseudoInverse rule: with linear activation function and based on Andersen-Wilamowski rule.

For the multiple layer network training, following training algorithms are available:

- Error Back Propagation (EBP),
- Levenberg-Marquardt, and

- Robust algorithm, a novel algorithm proposed and implemented by the authors of this paper [15].

All of these have the options of unipolar and bipolar activation functions.

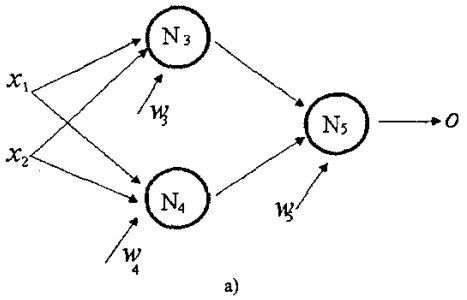
A. Input parameters

Even though an application is equipped with various validity checks with respect to input parameters, certain syntax has to be followed. For input/output patterns, initial set of weights and topology of the network itself, syntax of input parameters is as following.

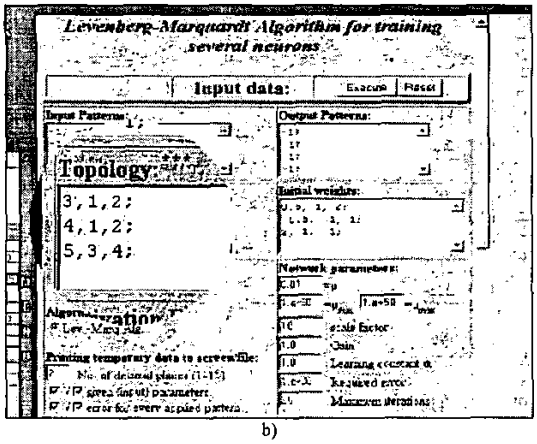
For the patterns, each line represents one pattern (applies for both input and output). the end of line is denoted by semi column (;). For topology, each line represents one neuron. Neurons have to be numbered in ascending other. For example, following line:

3,1,2;

represents neuron 3 with input connections from neurons (in this case inputs) 1 and 2. Then, the topology from Fig. 3 b) corresponds to a hierarchy from the Fig. 3 a).



a)



b)

Fig. 3. Neural Network online tool – Topology:
a) graphical form, b) syntax (screen has been whitened to emphasize relevant portion of the picture)

Parameters regarding algorithms available are implemented in form of radio buttons (only one selection possible). Same applies for activation function.

Parameters specific for algorithm in question will not be discussed here, since they are part of methodology itself. These parameters are however discussed in help feature available in proposed tool. All parameters can be input in any possible numerical format (integer, fixed point and exponential format). All input parameters are for the convenience of the user printed again at the beginning of the temporary and permanent output stream of results. This means that at the beginning of either of these two possible output streams, user has all input data including the name of the training algorithms selected.

B. Output values – temporary and/or final

In proposed online neural network tool, user is given a variety of options for monitoring how the algorithm progresses with the given task. Data can be printed to either screen or file or both. For user, this means he or she can start the application and then get back to it later and download the txt file with all wanted temporary and final data.

C. Output graphs and plots.

The proposed online training tool is equipped with two possible plotting options. Total error through iterations is given and pattern separation line movement (2-dimensional cases only).

Total error can be printed through all iteration steps, as well as the zoomed total error. The latter is zoomed in terms of both interval of iterations, and interval of total error, whichever comes first. This enables easy emphasizing of certain points in training process. Figures 4 a) and b) represent total error trend for all iterations and zoomed total error.

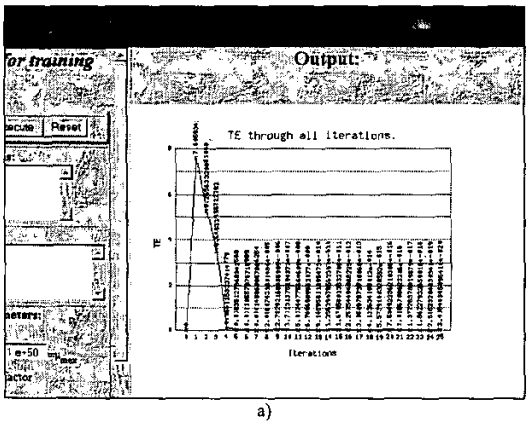
Separation line plotting is especially valuable in teaching process for 2-dimensional problems that are easier for humans to visualize. User also has two options here (Fig. 5). First plot gives the separation lines through all iterations and final position of separation line in different color. Patterns are differently colored and shown in different shape.

All four plots are provided in either jpeg or png format, widely recognized by most of popular web browsers. All plots are dynamically generated. The scale of these plots is also dynamically determined so the (0,0) point of Cartesian system is always in center of graph in final looks.

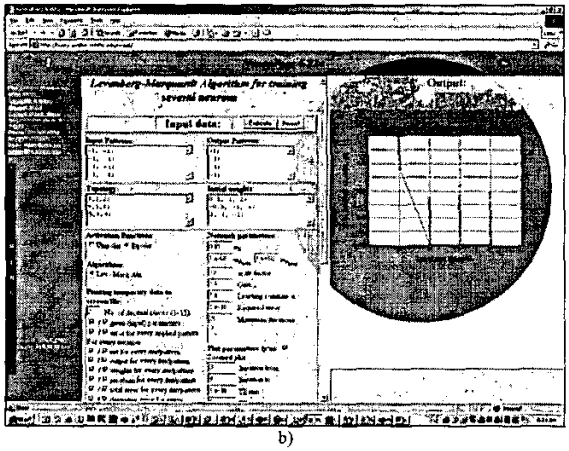
At the very end of temporary data printed to screen, user has at his disposal 4 different plots and two txt files, all downlodable and dynamically generated (Fig. 6). This means user can connect at later time and download to a local machine data that includes given (input) parameters, temporary parameters and final values, results returned by application.

Output stream of results in both temporary and permanent (txt) form ends with the statement of time spent in calculations. This time does not include the time needed to transmit data to local computer from remote machine where the application has been run. This brings us to the issue of speed.

Speed is totally up to a server where the application is hosted. However, program can have a faster response if less amount of temporary data is forwarded to a local computer that has initiated the training.

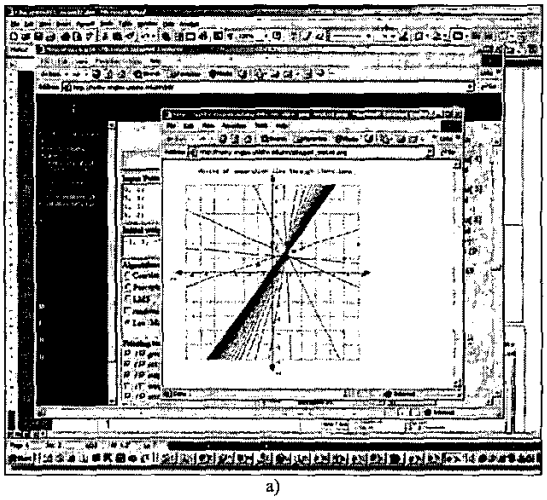


a)

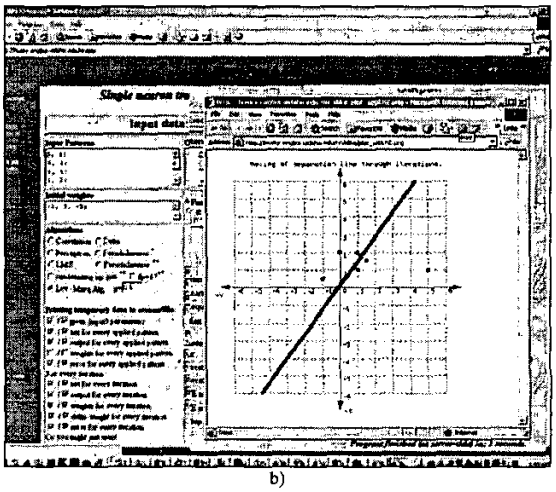


b)

Fig. 4. Neural Network online tool – Total Error
a) through all iterations, - zoomed (screen whitened to emphasize relevant portion of the picture).



a)



b)

Fig. 5. Neural Network online tool – Separation Lines
a) through all iterations, b) final one

IV. NEURAL NETWORK TOOL - HELP FEATURE

The proposed online tool is equipped with the electronic documentation. This help feature is produced by Scribble [16] software package. HTML Help files features professional looks like any other html-help commercial package (Fig. 7).

It has 5 books inside: NN Online Tool, Algorithms (training), Outputs, Inputs, User types and registration, with subsequent help menus. It can be searched through variety of indices and also has search by word feature. For search for specific word or phrases, Windows optimizes a database (list) that contains every word from help files. After the

process is over, while user types the word, he or she is given an option to select matching words that narrow the search, and corresponding topics are displayed. After clicking on desired topic, a window with corresponding html based description is shown.

V. CONCLUSION

Internet based neural network online simulation tool has been described in this paper. Detailed software usage is given. Implemented methods, assignment of input parameters and setting up of output parameters is given. Configuring final dynamically generated plots for total error and pattern separation lines are also discussed.

Online tool is available 24/7 and is accessible without restrictions. All questions and comments are highly appreciated. Authors hope this tool will reach its users, mainly in academic environment.

Future work might encompass implementation on Windows server platform. The idea is that user will be able to start an application and then disconnect, while application still runs remotely (savings in connection time and local resources that are left free for some other jobs). Then the user might only occasionally connect to a server in order to check how algorithm progresses with the given task.

VI. REFERENCES

[1] McCulloch, W.W., and W.H. Pitts, *"A logical calculus of the ideas imminent in nervous activity"*, Bull Math. Biophy, 5: 115-133.

[2] Stuttgart Neural Network Simulator, URL from May 2002: http://www-ra.informatik.uni-tuebingen.de/SNNS/.

[3] Neurodesigner - comprehensive neural network software, URL from May 2002: http://www.neurodesigner.com/

[4] A. Sanz, *"A Unified Tool for Fuzzy/Neural Network Systems"*, IEEE Micro, Vol. 17, No.2, pp. 65-69, 1997,

[5] Artificial Neural Network Lab on the Web, URL from May 2002: http://www.aist.go.jp/NIBH/~b0616/Lab/index-e.html

[6] The HTML Neural Net Consulter, URL from May 2002: http://www.hav.com/nnhtml.htm

[7] Neural Networks for Control: The Ball Balancing Problem, URL from May 2002: http://neuron.eng.wayne.edu/bpBallBalancing/ball5.html

[8] Neural Network FN Predictions for Stainless Steel Welds [h], URL from May 2002: http://engm01.ms.ornl.gov/FNPLotsNeural.HTML

[9] Nenet Neural Networks tool, URL from May 2002: http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html

[10] Pacific Northwest National Laboratory, URL from May 2002: http://www.emsl.pnl.gov:2080/proj/neuron/neural/demos.html

[11] Network Cybernetics Corporation, URL from May 2002: http://www.ncc.com/misc/ai_sites.html

[12] J.M. Zurada, *Artificial Neural Systems*, PWS Publishing Company, St. Paul, MN, 1995.

[13] [Hagan 96] M.T. Hagan, H.B. Demuth, M.Beale, *Neural Nework Design*, PWS Co., Boston MA, 1996.

[14] R.J. Schalkoff, *Artificial Neural Networks*, McGraw-Hill, New York, 1997.

[15] Manic, M. Wilamowski, D., *Robust Algorithm for Neural Network Training*, International Joint INNS-IEEE Conference on Neural Networks, Honolulu, Hawaii, May 12-17, pp. 1528-1533, 2002.

[16] Scribble electronic documentation, URL from May 2002: http://www.jgsoft.com/

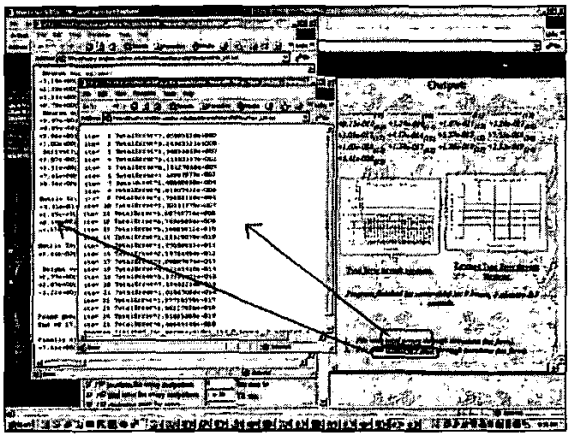


Fig. 6. Neural Network online tool – separation lines for 2-dimensional cases, through all iterations and final position (screen has been whitened to emphasize html help feature of the application).

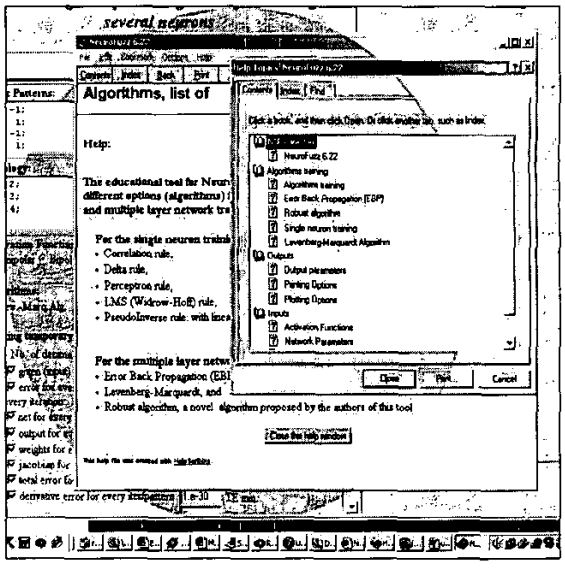


Fig. 7. Neural Network online tool – HTML-Help (Neural Network online tool - HTML help (the rest of the screen has been whitened to emphasize html help feature of the application).