Neural Comput & Applic (2013) 23:609–615
DOI 10.1007/s00521-013-1408-9

A survey of artificial neural network training tools

Darío Baptista · Fernando Morgado-Dias

Received: 28 February 2013 / Accepted: 5 April 2013 / Published online: 14 June 2013

Abstract Artificial neural networks (ANN) are currently an additional tool which the engineer can use for a variety of purposes. Classification and regression are the most common tasks; however, control, modeling, prediction and forecasting are common tasks as well. For over three decades, the field of ANN has been the center of intense research. As a result, one of the outcomes has been the development of a large set of software tools used to train these kinds of networks, making the selection of an adequate tool difficult for a new user. This paper aims to help the ANN user choose the most appropriate tool for its application by providing a large survey of the solutions available, as well as listing and explaining their characteristics and terms of use. The paper limits itself to focusing on the tools which were developed for ANN and the relevant characteristics of these tools, such as the operating systems, hardware requirements, license types, architectures and algorithms available.

Keywords Artificial neural networks · Training tools · Training algorithms · Software

D. Baptista (✉) · F. Morgado-Dias
Madeira Interactive Technologies Institute and Competence Center of Exact Science and Engineering, University of Madeira, 9000-390 Funchal, Portugal
e-mail: fdariobaptista@gmail.com

F. Morgado-Dias
e-mail: morgado@uma.pt

1 Introduction

Artificial neural networks (ANN) are currently an additional tool which the engineer can use for a variety of purposes. Classification and regression are the most common tasks; however, control, modeling, prediction and forecasting are common tasks as well. For over three decades, the field of ANN has been the center of intense research. As a result, one of the outcomes has been the development of a large set of software tools used to train these kinds of networks, which make the selection of an adequate tool difficult for a new user.

The set of tools currently available, either gratuitously or commercially, is quite large. These tools may vary from one another due to the different tasks for which each was designed, their use of different algorithms, the different requirements each possesses, as well as the implementation of different types of ANNs or any other characteristic of these ANN implementations. These tools also differ with regard to licenses and the possibility of accessing the source code.

Given the set of available tools and their differing characteristics, it is not an easy task for a user to appropriately choose the tool most adequate for a given task and dataset.

Therefore, with the purpose of reporting which tools are available at present and easing the choice of which tool to use, this paper contains the description of the software which has been developed specifically for ANN. It is important to note that some tools were created and intended for academic use, whereas others are commercial and sophisticated tools.

The paper does not cover ANN such as spiking neurons and other models which are closely biologically inspired.

Table 1 Operating system for each tool

Tools	Operating system					
	Window	Mac OS X	Unix	Linux	Sun	Others
AiNet	✔ (3.1 to NT)	✘	✘	✘	✘	✘
Annie	✔ (XP or later)	✘	✘	✔	✘	✘
Aspirin/migraines	✘	✘	✔	✔	✘	✘
Basis-of-AI-NN	✔ (3.1)	✘	✔	✔	✘	✘
EasyNN	✔ (all version)	✘	✘	✘	✘	✘
Encog	✔ (all version)	✘	✘	✘	✘	✘
FANN	✔ (all version)	✔	✘	✔	✘	✘
FuNeGen	✔ (all version)	✘	✘	✘	✘	✘
GENESIS 2.3	✘	✘	✔	✔	✘	✘
JATTON	✔ (all version)	✘	✘	✘	✘	✘
Java NNS	✔ (NT or later)	✔	✘	✔	✔ (OS 7)	✘
Joone	✔ (all version)	✔	✘	✔	✔	✘
Lens	✔ (all version)	✔	✔	✘	✘	✘
LM-MLP	✔ (95 or later)	✘	✘	✘	✘	✘
Multiple backpropagation	✔	✘	✘	✘	✘	✘
NNFit	✘	✘	✔	✔	✔ (OS 4.1.4)	AIX; IRIX
NeuroModeler	✔ (95 to XP)	✘	✔	✔	✔	✘
NeuroIntelligence	✔ (98 or later)	✘	✘	✔	✘	✘
Neuroph	✔	✘	✘	✘	✘	✘
NeuroSolutions	✔ (XP or later)	✘	✘	✘	✘	✘
Nest	✔	✔ (10.3)	✔	✔	✔ (Solaris)	AIX; SGI;
NevProp	✘	✔	✔	✘	✘	✘
Nico toolkit	✘	✘	✔	✘	✘	✘
Nuclass	✔ (NT or later)	✘	✘	✘	✘	✘
Numap	✔ (NT or later)	✘	✘	✘	✘	✘
PDP++ (known as emergent)	✔ (95 or later)	✔	✘	✔	✔	SGI
Pythia	✔ (95 or later)	✘	✘	✘	✘	✘
SNNS	✘	✘	✔	✔	✔ (OS 4;5)	Ultrix; AIX; IRIX
SOM_PAK—LVQ_PAK	✘	✘	✔	✘	✘	✘
Statistica	✔ (NT or later)	✔	✘	✘	✘	✘
Torch	(*)	✔	✔	✔	✘	FreeBSD
Trajan	✔	✘	✘	✘	✘	✘
Uts	✘	✘	✔	✔	✔ (OS 5.3)	IRIX
WEKA	✔	✔ (10.6)	✘	✔	✘	✘
XNBC	✔ (95 to XP)	✘	✔	✔	✔	AIX; Ultrix; Hpux

(*) is not yet windows compatible, coming soon; ✔ there is; ✘ there is not

2 Artificial neural network tools

For most of the ANN's tools, it is a relatively easy task to find appropriate information, since they are documented through scientific papers [1–11] or Web sites [12–41] with sufficient information, allowing the user to make an informed decision of which tools to test and use or in which cases the user needs to search for another tool which best fits their needs.

2.1 System requirements

ANN tools have requirements at the operating system level and minimum requirements in terms of hardware. The first set of requirements usually results from the period in which the tool was developed, whereas the second set results from the processing capacity necessary for the tool and its size.

The most common set of requirements for any tool is the operating system. In order to function correctly, each tool

Table 2 Additional software and hardware requirements

Tools	RAM	HD free space	Video		Processor	JRE
			Resolution	Color		
AiNet	4 MB	5 MB	✘	✘	✘	✘
Encog	✘	✘	✘	✘	✘	1.5 or later
GENESIS 2.3	✘	✘	✘	✘	Intel and AMD 64 bit	✘
JATTON	✘	✘	✘	✘	✘	1.6 or later
Java NNS	✘	✘	✘	✘	✘	1.3 or later
Joone	256 MB	✘	✘	✘	✘	✘
LM-MLP	32 MB	✘	✘	✘	Pentium I or II	✘
Nenet	✘	✘	✘	16 bits	✘	✘
NetLab	✘	456 KB	✘	✘	✘	✘
NeuroModeler	128 MB	✘	✘	✘	Processor speed: 1 GHz	✘
NeuroIntelligence	128 MB	15 MB	800 × 600	8 bits	Pentium II	✘
Neuroph	✘	✘	✘	✘	✘	1.6 or later
NeuroSolutions	✘	100 MB	1,024 × 768	✘	Pentium IV	✘
Nico toolkit	✘	✘	✘	✘	✘	1.4 or later
Nuclass	32 MB	25 MB	✘	✘	✘	✘
Numap	32 MB	25 MB	✘	✘	✘	✘
Pythia	32 MB	1.39 MB	✘	✘	Pentium	✘
SOM	64 MB	1 MB	✘	✘	✘	✘
Statistica	1 GB	172.84 MB	✘	✘	Processor speed: 2 GHz	✘
Weka	✘	✘	✘	✘	✘	1.5 or later

✘ There is not any specification

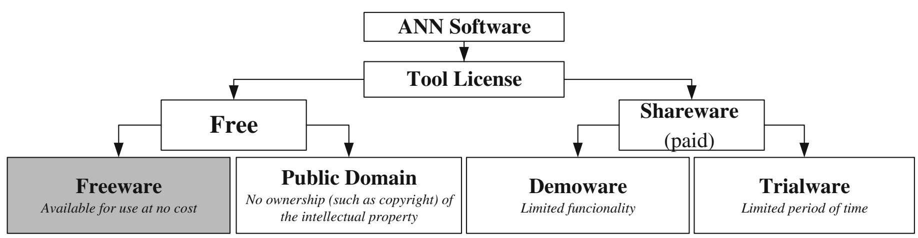

Fig. 1 Tools license classification

needs an appropriate operating system. Table 1 shows the recommended operating system for this proper functioning. It also displays that for the majority of the tools used, Windows acts as the operating system. Outside the Microsoft universe, the operating system most often used for the realization of an ANN's tool is the Mac OS.

The tools FastICA, Fuzzy ART/Fuzzy ARTMAP, NNSYSID, NNT, Netlab, neural network toolbox and SOM act independently of the operating system, and, as a result, they are not presented in Table 1. These tools require only the MATLAB platform because they are implemented by .m files.

The other common requirement is related to the minimal hardware and software characteristics. These requirements correspond to the elements which are necessary to be installed on the computer in order to provide an optimal functioning of the tool. It is important to remember that these software pre-requisites are not included in the tool installation package and thus need to be installed separately before installing the required tool.

Table 2 displays the various aspects of software and hardware requirements necessary for each tool. It is important to remind the user that greatly exceeding these requirements does not guarantee that it will run absolutely smoothly or function at its best.

2.2 License characteristics

In this subsection, the licenses of the different tools are analyzed. The classification of the licenses is described through the diagram presented in Fig. 1.

The diagram shows the two kinds of licenses available for any software: paid tools and free tools. It should be noted, however, that within the paid tools, special licenses

Table 3 General product characteristics

Tools	License	Acad. Lic.
AiNet	F	–
Annie	F	–
Aspirin/migraines	F	–
Basis-of-AI-NN	F	–
EasyNN	T	Y
Encog	F	–
FANN	F	–
FastICA	F	–
FuNeGen	F	–
Fuzzy ART and fuzzy ARTMAP	F	–
GENESIS 2.3	F	–
JATTON	F	–
Java NNS	F	–
Joone	F	–
Lens	F	–
LM-MLP	F	–
Multiple backpropagation	F	–
Multilayer perceptron	F	–
MUME	F	–
Nenet	D	N
NNFit	PD	–
NNSYSID	F	–
NNT—neural network trainer	F	–
NeoC explorer	F	–
NetLab	F	–
NeuroModeler	F	–
Neural networks at your fingertips	F	–
Neural network toolbox	T	Y
NeuroIntelligence	T	N
Neuroph	F	–
NeuroSolutions	T	N
Nest	F	–
NevProp—Nevada backpropagation	F	–
Nico toolkit	F	–
Nuclass	D	N
Numap	D	N
PDP++ (also known as emergent)	F	–
Pythia	D	N
Rochester connectionist simulator	F	–
SNNS—stuttgart neural network simulator	F	–
SOM	F	–
SOM_PAK—LVQ_PAK	PD	–
Statistica	T	Y
Torch	F	–
Trajan	D	Y
Uts	F	–
WEKA	F	–
XNBC	F	–

Legal status: *PD* public domain, *F* freeware, *T* trialware, *D* demoware

Acad. Lic.: *Y* yes, *N* no

Table 4 Network architecture for each tool

Tools	Architecture
AiNet	N38
Annie	N22; N27; N32; N38; N47
Aspirin/migraines	N13; N43
Basis-of-AI-NN	N4; N6; N9; N20; N22; N27; N29
EasyNN	N38
Encog	N2; N4; N6; N9; N16; N19; N27; N31; N32; N38; N42; N47; N50
FANN	N38
FastICA	N30
FuNeGen	N12
Fuzzy ART/fuzzy ARTMAP	N4; N5
GENESIS 2.3	N20; N26
JATTON	N11; N16; N32
Java NNS	N20
Joone	N20; N32; N39; N43; N52
Lens	N9; N20; N22; N29; N32
LM-MLP	N38
Multiple backpropagation	N20; N38
NNFit	N38
NNSYSID	N20; N38
NNT	N1; N8; N10; N38
NetLab	N20; N24; N35; N38; N47; N50
NeuroModeler	N2; N4; N6; N9; N16; N38; N42
Neural network toolbox	N2; N18; N19; N22; N27; N33; N38; N47; N50; N52
NeuroIntelligence	N11; N38
Neuroph	N2; N6; N15; N24; N27; N32; N34; N38; N47
NeuroSolutions	N12; N19; N20; N22; N23; N37; N31; N38; N39; N43; N46; N47; N50; N51; N52; N53; N54; N55
Nest	N4; N28
NevProp	N20; N38
Nico toolkit	N18; N38; N47
Nuclass	N38; N45; N50
Numap	N38; N45; N50
PDP++ (also known as emergent)	N9; N17; N20; N24; N27; N27; N36; N40; N43; N49; N50
Pythia	N11; N38
SNNS	N3; N4; N6; N16; N19; N27; N33; N44; N50; N22; N52
SOM	N43; N50
SOM_PAK—LVQ_PAK	N33; N50
Statistica	N23; N38; N41; N46; N47; N50
Torch	N14; N38; N40; N47; N52
Trajan	N23; N38; N46; N47; N50
Uts	N37; N50
WEKA	N7; N38; N47

Table 4 continued

Tools	Architecture
XNBC	N26; N28

N1—Arbitrarily connected neuron; N2—adaline linear neuron; N3—autoassociative memory; N4—adaptive resonance theory; N5—adaptive resonance theory mapping; N6—bidirectional associative memory; N7—BayesNet/Bayesian network; N8—bipolar network; N9—Boltzmann machine N10—bridged multiplayer perceptron; N11—backpropagation network; N12—coactive network-fuzzy interference system; N13—canonical discriminants analysis N14—convolutional network; N15—competitive neural network; N16—counter-propagation neural network; N17—deep Boltzmann machines/mean-field; N18—dynamic network; N19—Elman Network; N20—feed-forward neural network; N21—functional link network; N22—recurrent network; N23—generalized regression neural network; N24—Hebbian Network; N25—hybrid models; N26—Hodgkin & Huxley neural network; N27—hopfield network; N28—integrate and fire; N29—interactive activation network; N30—independent component analysis; N31—Jordan recurrent; N32—Kohonen networks; N33—learning vector quantization nets; N34—maxnet; N35—mixture density network; N36—multidimensional scaling; N37—mixture of Gaussians networks; N38—multiplayer perceptron; N39—modular neural network; N40—mixture of experts; N41—neocognitron neural network; N42—neuroevolution of augmenting topologies; N43—principal component analysis; N44—pruned cascade-correlation; N45—piecewise linear network; N46—probabilistic neural network; N47—radial basis function; N48—space mapping; N49—stochastic neural network; N50—self-organizing map; N51—support vector machine; N52—time-delay neural network

for academic institutions can be found. Table 3 shows the tools' name accompanied by their license.

2.3 Network architectures

The architecture of an ANN, that is, the structure and type of network, is one of the most important choices concerning the training software. Some networks are more appropriate for some tasks than others, and thus each software tool in turn only covers a subset of all the architectures available. Table 4 shows the networks architectures available for each tool. It can be seen that the types of networks which are more usual are radial basis function, multiplayer perceptron and self-organizing map.

2.4 Training algorithms

The training algorithms available and the quality of their implementation is another important aspect to consider when choosing the training tool. Table 5 summarizes the algorithms available for each tool. As one would expect, due to its popularity, the training algorithm which is most usual is the Backpropagation (steepest descent).

Table 5 Training algorithm for each tool

Tools	Architecture
AiNet	–
Annie	A1; A8
Aspirin/migraines	A1; A7; A44; A49
Basis-of-AI-NN	A1; A10; A25; A27; A37
EasyNN	–
Encog	A1; A6; A8; A16; A29
FANN	A1; A2; A22; A37; A39
FastICA	A13
FuNeGen	A1; A7
Fuzzy ART/fuzzy ARTMAP	A1
GENESIS 2.3	A1
JATTON	A1; A8
Java NNS	A1
Joone	A1; A2; A7; A10; A39
Lens	A1; A8; A10; A31
LM-MLP	A29
Multiple backpropagation	A1
NNFit	A35
NNSYSID	A1; A3; A21; A23; A29; A30; A38
NNT	A1; A11; A12; A32; A40
NetLab	A1; A6; A29
NeuroModeler	A1; A6; A16; A35; A46
Neural network toolbox	A1; A2; A3; A4; A5; A9; A15; A17; A18; A19; A20; A22; A29; A33; A34; A35; A41
NeuroIntelligence	A3; A6; A29; A35; A36; A37; A50; A51; A52
Neuroph	A1; A8
NeuroSolutions	A1; A6; A10; A29; A31; A37; A43
Nest	–
NevProp	A14; A37
Nico toolkit	–
Nuclass	A24; A25
Numap	A24; A25
PDP++ (also known as emergent)	A1; A8; A28; A42
Pythia	–
SNNS	A1; A37
SOM	A24; A25; A27
SOM_PAK—LVQ_PAK	A27
Statistica	A1; A6; A10; A29; A35; A37
Torch	A7
Trajan	A1; A6; A10; A29; A35; A37
Uts	A6
WEKA	A1; A26

Table 5 continued

Tools	Architecture
XNBC	–

A1—Back propagation; A2—batch training; A3—batch version of the backpropagation; A4—batch training with weight and bias learning rules; A5—Bayesian regularization; A6—conjugate gradient; A7—conjugate gradient descent; A8—competitive learning; A9—cyclical order incremental update; A10—delta-bar-delta; A11—enhanced self-aware algorithm; A12—evolutionary gradient; A13—fast fixed-point algorithm; A14—fixed global adaptive; A15—Fletcher–powell conjugate gradient back propagation; A16—genetic algorithm; A17—gradient descent back propagation; A18—gradient descent with adaptive learning rule back propagation; A19—gradient descent with momentum and adaptive learning rule backpropagation; A20—gradient descent with momentum back propagation; A21—recursive prediction error (Gauss–Newton); A22—incremental training; A23—iterated generalized least squares; A24—k-means; A25—k-nearest neighbor; A26—Lazy Bayesian rules; A27—learning vector quantization; A28—Leabra; A29—Levenberg–Marquardt; A30—memory-saving implementation of the LM; A31—momentum; A32—neuron by neuron algorithm; A33—Polak–Ribiére conjugate gradient back propagation; A34—Powell–Beale conjugate gradient back propagation; A35—Quasi-Newton; A36—Quasi-Newton (limited memory); A37—quick propagation; A38—recursive (/incremental) version of backpropagation; A39—resilient propagation (RPROP); A40—self-aware algorithm; A41—scaled conjugate gradient backpropagation; A42—Soft competitive learning; A43—step by step; A44—stability issues; A45—sequential minimal optimization algorithm; A46—simulated annealing; A47—weight perturbation; A48—weight update driven node splitting; A49—weight update issues

3 Conclusions

ANN are widely used throughout many different application areas. An ever increasing number of practitioners use ANN as a tool to solve a variety of problems such as modeling, control and forecasting.

Most of these ANN users do not consider developing their own implementation of a training tool and therefore require a fast, reliable and appropriate tool for their own applications.

Throughout this paper, a simplification of this task has been provided by presenting the tools currently available on the market as well as their characteristics.

The options relating to licenses, additional software requirements, training algorithms and network architectures are likewise featured for an easier selection by the user.

Acknowledgments The authors would like to acknowledge the Portuguese Foundation for Science and Technology for their support in this work through the project PEst-OE/EEI/LA0009/2011.

References

1. Huxhold WL, Henson TF, Bowman JD; IBM Corp., Houston, TX (1992) ANNIE: a simulated neural network for empirical studies and application prototyping. Annie. Simulation symposium. Proceedings of 25th annual USA, pp 2–8
2. Aapo H (1999) Fast and robust fixed-point algorithms for independent component analysis. FastICA. IEEE Trans Neural Netw 10(3):626–663
3. Yu H, Wilamowski BM (2009) Efficient and reliable training of neural networks. NNT—neural network trainer. IEEE human system interaction conference, Italy, May 21–23, pp 109–115
4. Ian TN (2004) NETLAB: algorithms for pattern recognition. Springer, UK
5. Lopes N, Ribeiro B (2009) GPU implementation of the multiple back-propagation algorithms. In: Proceedings of intelligent data engineering and automated learning, Springer, pp 449–456
6. Mark H, Eibe F, Geoffrey H, Bernhard P, Peter R, Ian HW (2009) The WEKA data mining software: an update, vol 11, no 1. SIGKDD Explorations
7. Shankar A (2002) Annie—artificial neural network library. Last Update: 18th Jun 2002. Available at: http://annie.sourceforge.net/
8. Leighton R (2010) Elegant software. Aspirin/migraines. Last accessed: 13rh Out 2010. Available at: http://www.elegant-software.com/software/aspirin/
9. Tveter D (2010) The pattern recognition basis of AI neural networking software. Last accessed: 14th Out 2010. Available at: http://www.dontveter.com/nnsoft/nnsoft.html
10. Neural Planner Software. EasyNN Plus. Last accessed: 12th Out 2010. Available at: http://www.easynn.com/
11. Laboratory of Computer and Information Science. The FastICA package for MATLAB. Last accessed: 12th Out 2010. Available at: http://www.cis.hut.fi/projects/ica/fastica/
12. GenAlgo Team. FuNeGen 1.0. Last accessed: 13th Out 2010. Available at: http://www.genalgo.com/index.php?option=com_content&task=view&id=211&Itemid=31
13. Garrett A. MatLab central: fuzzy ART and fuzzy ARTMAP neural networks. Last accessed: 16th Out 2010. Available at: http://www.mathworks.com/matlabcentral/fileexchange/4306
14. GENESIS. Last accessed: 15th Out 2010. Available at: http://www.genesis-sim.org/GENESIS/
15. Aires de Sousa J. JATOON—Java tools for neural networks. Last accessed: 13rd Out 2010. Available at: http://www.dq.fct.unl.pt/staff/jas/jatoon/
16. Zell A. JavaNNS—Java neural network simulator. Last accessed: 11rd Out 2010. Available at: http://www.ra.cs.unituebingen.de/software/JavaNNS/welcome_e.html
17. Java Tips, Joone. Last accessed: 15th Out 2010. Available at: http://www.java-tips.org/javalibraries/neuralnetworks/joone.html
18. Rohde D, Lens the light, efficient network simulator. Last accessed: 14th Out 2010. Available at: http://tedlab.mit.edu/~dr/Lens/
19. Neural Decision Lab LLC, Software. LM_MLP, Nuclass and Numap. Last accessed: 13rd Out 2010. Available at: http://www.neuraldl.com/Software.php
20. Lopes N. Multiple back-propagation. Last accessed: 16th Out 2010. Available at: http://mbp.sourceforge.net/
21. Patrick Cloutier, Cristian Tibirna, Bernard Grandjean; Jules Thibault. NNFit (neural network fitting). Last accessed: 11rd Out 2010. Available at: http://www.gch.ulaval.ca/nnfit/english/index.html
22. Ravn O, Nørgaard M. The NNSYSID toolbox for use with MatLab. Last accessed: 12nd Out 2010. Available at: http://www.iau.dtu.dk/research/control/nnsysid.html
23. Wilamowski BM. Neural network training software for networks with arbitrarily connected neurons. Last accessed: 15th Out 2010. Available at: http://www.eng.auburn.edu/~wilambm/nnt/
24. MathWorks. Neural network toolbox—design and simulate neural networks. Last accessed: 11th Out 2010. Available at: http://www.mathworks.com/products/neuralnet/

25. Alyuda Research. ALYUDA. NeuroSolutions. Last accessed: 12nd Out 2010. Available at: http://www.alyuda.com/neural-networks-software.htm
26. Softpedia. Neuroph is lightweight Java neural network framework. Last accessed: 15th Out 2010. Available at: http://neuroph.sourceforge.net/index.html
27. NeuroDimension. NeuroSolutions–premier neural network development environment. Last accessed: 15th Out 2010. Available at: http://www.neurosolutions.com/
28. Goodman P. Easy to use feed-forward backpropagation program. Nevada backpropagation (NevProp). Last accessed: 10th Out 2010. Available at: http://hpux.connect.org.uk/hppd/hpux/NeuralNets/NevProp-1.6/
29. KTH. The NICO toolkit. Last update. Last accessed: 14th Out 2010. Available at: http://nico.nikkostrom.com/
30. The PDP++ Software Home Page. Last accessed: 10th Out 2010. Available at: http://archive.cnbc.cmu.edu/Resources/PDP%2B%2B/PDP%2B%2B.html
31. Runtime Software. Pythia—The neural network designer. Last accessed: 16th Out 2010. Available at: http://www.vyomlinks.com/download/pythia-the-neural-network-designerv1.00-2523.html
32. Zell A. Stuttgart neural network simulator. Last accessed: 11rd Out 2010. Available at: http://www.nada.kth.se/~orre/snns-manual/
33. Laboratory of Computer and Information Science. SOM Toolbox 2.0. Last accessed: 12th Out 2010. Available at: http://www.cis.hut.fi/projects/somtoolbox/
34. Hynninen J. SOM_PAK and LVQ_PAK. Last accessed: 14th Out 2010. Available at: http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml
35. Statsoft. Statistica. Last accessed: 17th Out 2010. Available at: http://www.statsoft.com/
36. Collobert R. Torch. Last accessed: 14th Out 2010. Available at: http://www.torch.ch/
37. TrajanSoftware. TRAJAN 6.0 PROFESSIONAL. Last accessed: 9th Out 2010. Available at: http://www.trajan-software.demon.co.uk/
38. Zhang QJ, Carleton University. NeuroModeler. Last accessed: 6th Jan 2012. Available at: http://neuroweb.doe.carleton.ca
39. Heaton Research and Encog. Encog Java and DotNet neural network framework. Encog. Last accessed: 6th Jan 2012. Available at: http://www.heatonresearch.com/encog
40. Vibert J-F, Alvarez F. XNBC: a software package to simulate biological neural networks for research and education. XNBC. Last accessed: 17th Out 2010. Available at: http://www.b3e.jussieu.fr/xnbc/
41. Gewaltig M-O. Nest initiative. Nest. Last accessed: 17th Out 2010. Available at: http://www.nest-initiative.org