A Cloud-Based Neural Network Simulation Environment

Erich Schikuta and Erwin Mann

University of Vienna, Faculty of Computer Science,
A-1090 Währingerstr. 29, Vienna, Austria,
erich.schikuta@univie.ac.at

Abstract. We present N2Sky, a novel Cloud-based neural network simulation environment. The system implements a transparent environment aiming to enable arbitrary and experienced users to do neural network simulations easily and comfortably. The necessary resources, as CPU-cycles, storage space, etc., are provided by using Cloud infrastructure. N2Sky also fosters the exchange of neural network specific knowledge, as neural network paradigms and objects, between users following a virtual organization design blue-print. N2Sky is built using the RAVO reference architecture which allows itself naturally integrating into the Cloud service stack (SaaS, PaaS, and IaaS) of service oriented architectures.

Keywords: Artificial Neural Network Simulation, Cloud computing, SOA/SOI, Virtual Organization

1 Introduction

We are living in the era of virtual collaborations, where resources are logical and solutions are virtual. Advancements on conceptual and technological level have enhanced the way people communicate. The exchange of information and resources between researchers is one driving stimulus for development. This is just as valid for the neural information processing community as for any other research community. As described by the UK e-Science initiative [1] several goals can be reached by the usage of new stimulating techniques, such as enabling more effective and seamless collaboration of dispersed communities, both scientific and commercial, enabling large-scale applications and transparent access to "high-end“ resources from the desktop, providing a uniform "look & feel“ to a wide range of resources and location independence of computational resources as well as data.

A Virtual Organisation is a logical orchestration of globally dispersed resources to achieve common goals. It couples a wide variety of geographically distributed computational resources (such as PCs, workstations and supercomputers), storage systems, databases, libraries and special purpose scientific instruments to present them as a unified integrated resource that can be shared transparently by communities.

In the Computational Intelligence community these current developments are not used to the maximum possible extent until now. As an illustration for this we highlight the large number of neural network simulators that have been developed, as for instance SOM-PAK [2] and SNNS [3] to name only a few. Many scientists, scared of existing programs failing to provide an easy-to-use, comprehensive interface, develop systems for their specific neural network applications. This is also because most of these systems lack a generalized framework for handling data sets and neural networks homogeneously. This is why we believe that there is a need for a neural network simulation system that can be accessed from everywhere.

We see a solution to this problem in the N2Sky system. N2Sky is an artificial neural network simulation environment providing basic functions like creating, training and evaluating neural networks. The system is Cloud based in order to allow for a growing virtual user community. The simulator interacts with Cloud data resources (i.e. databases) to store and retrieve all relevant data about the static and dynamic components of neural network objects and with Cloud computing resources to harness free processing cycles for the "power-hungry“ neural network simulations. Furthermore, the system allows to be extended by additional neural network paradigms provided by arbitrary users.

The layout of the paper is as follows: In the following section we give the motivation behind the work done. In section 3 we present the design principles behind the N2Sky development. The system deployment within a Cloud environment is described in section 4. The interface of N2Sky is laid out in section 5. The paper closes with a look at future developments and research directions in Section 5.

2 Towards a Cloud-Based ANN Simulator

In the last years the authors developed several neural network simulation systems fostering up-to-date computer science paradigms then.

NeuroWeb [4] is a simulator for neural networks which exploits Internet-based networks as a transparent layer to exchange information (neural network objects, neural network paradigms). NeuroAccess [5] and NeuroOracle [6] identify neuronal network elements as complex objects in the sense of database technology and integrate them conceptually and physically into the object-relational model. This approach supports an object-oriented point of view which enables a natural mapping of neural network objects and their methods to the service-oriented landscape. The N2Cloud system [7] is based on a service oriented architecture (SOA) and is a further evolution step of the N2Grid systems [8]. The original idea behind the N2Grid system was to consider all components of an artificial neural network as data objects that can be serialized and stored at some data site in the Grid, whereas N2Cloud will use the storage services provided by the Cloud environment.

The presented N2Sky environment takes up the technology of N2Cloud to a new dimension using the virtual organisation paradigm. Hereby the RAVO reference architecture is used to allow the easy integration of N2Sky into the Cloud service stack using SaaS, PaaS, and IaaS. Cloud computing is a large scale distributed computing paradigm for utility computing based on virtualized, dynamically scalable pool of resources and services that can be delivered on-demand over the Internet. In the scientific community it is sometimes stated as the neural evolution of Grid computing, which lacks on usability and accountability. Cloud computing therefore became a buzz word after IBM and Google collaborated in this field followed by IBM's "Blue Cloud" [9] launch. Three categories can be identified in the field of Cloud computing:

- **Software as a Service (SaaS):** This type of Cloud delivers configurable software applications offered by third party providers on an on-demand base and made available to geographically distributed users via the Internet. Examples are Salesforce.com, CRM, Google Docs, and so on.
- **Platform as a Service (PaaS):** Acts as a runtime-system and application framework that presents itself as an execution environment and computing platform. It is accessible over the Internet with the sole purpose of acting as a host for application software. This paradigm offers customers to develop new applications by using the available development tools and API's. Examples are Google's App engine and Microsoft's Azure, and so on.
- **Infrastructure as a Service (IaaS):** Traditional computing resources such as servers, storage, and other forms of low level network and physical hardware resources are hereby offered in a virtual, on-demand fashion over the Internet. It provides the ability to provide on-demand resources in specific configurations. Examples include Amazon's EC2 and S3, and so on.

The motivation behind the development of N2Sky is to

- share neural net paradigms, neural net objects and other data and information between researchers, developers and end users worldwide. Provide for an efficient and standardized solution to neural network problems,
- allow for transparent access to "high-end"neural resources stored within the Cloud from desktop or smart phone,
- provide a uniform "look and feel" to neural network resources, and
- foster location independence of computational, storage and network resources.

3 N2Sky Design

Information Technology (IT) has become an essential part of our daily life. Utilization of electronic platforms to solve logical and physical problems is extensive. Grid computing is often related with Virtual Organisations (VOs) when it comes to creation of an E-collaboration. The layered architecture for grid computing has remained ideal for VOs.

However, grid computing paradigm has some limitations. Existing grid environments are categorized as data grid or computational grid. Today, problems being solved using VOs require both data and storage resources simultaneously.

Scalability and dynamic nature of the problem solving environment is another serious concern. Grid computing environments are not very flexible to allow the participant entities enter and leave the trust. Cloud computing seems to be a promising solution to these issues. Only, demand driven, scalable and dynamic problem solving environments are target of this newborn approach. Cloud computing is not a deviation concept from the existing technological paradigms, rather it is an upgradation. Cloud computing centers around the concept of XaaS, ranging from hardware/software, infrastructure, platform, applications and even humans are configured as a service. Most popular service types are IaaS, PaaS and SaaS.

Existing paradigms and technology is used to form VOs, but lack of standards remained a critical issue for the last two decades. Our research endeavor focused on developing a Reference Architecture for Virtual Organizations (RAVO) [10]. It is intended as a standard for building Virtual Organizations (VO). It gives a starting point for the developers, organizations and individuals to collaborate electronically for achieving common goals in one or more domains. RAVO consists of two parts,

1. The requirement analysis phase, where boundaries of the VO are defined and components are identified. A gap analysis is also performed in case of evolution (up-gradation) of an existing system to a VO.
2. The blueprint for a layered architecture, which defines mandatory and optional components of the VO.

This approach allows to foster new technologies (specifically the SOA/SOI paradigm realized by Clouds) and the extensibility and changeability of the VO to be developed.

The basic categorization of the N2Sky design depends on the three layers of the Cloud service stack as they are: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Figure 1 depicts the components of the N2Sky framework, where white components are mandatory, and the other components are optional.

Infrastructure as a Service (IaaS) basically provides enhanced virtualisation capabilities. Accordingly, different resources may be provided via a service interface. In N2Sky the IaaS layer consists of two sub-layers: a Factory layer and an Infrastructure Enabler Layer. Users need administrative rights for accessing the resources in layer 0 over the resource management services in layer 1.

– Factory Layer (Layer 0): contains physical and logical resources for the N2Sky. Physical resources comprise of hardware devices for storage, computation cycles and network traffic in a distributed manner. Logical resources contain experts knowledge helping solving special problems like the Paradigm Matching.
– Infrastructure Enabler Layer (Layer 1): allows access to the resources provided by the Factory layer. It consists of protocols, procedures and methods to manage the desired resources.

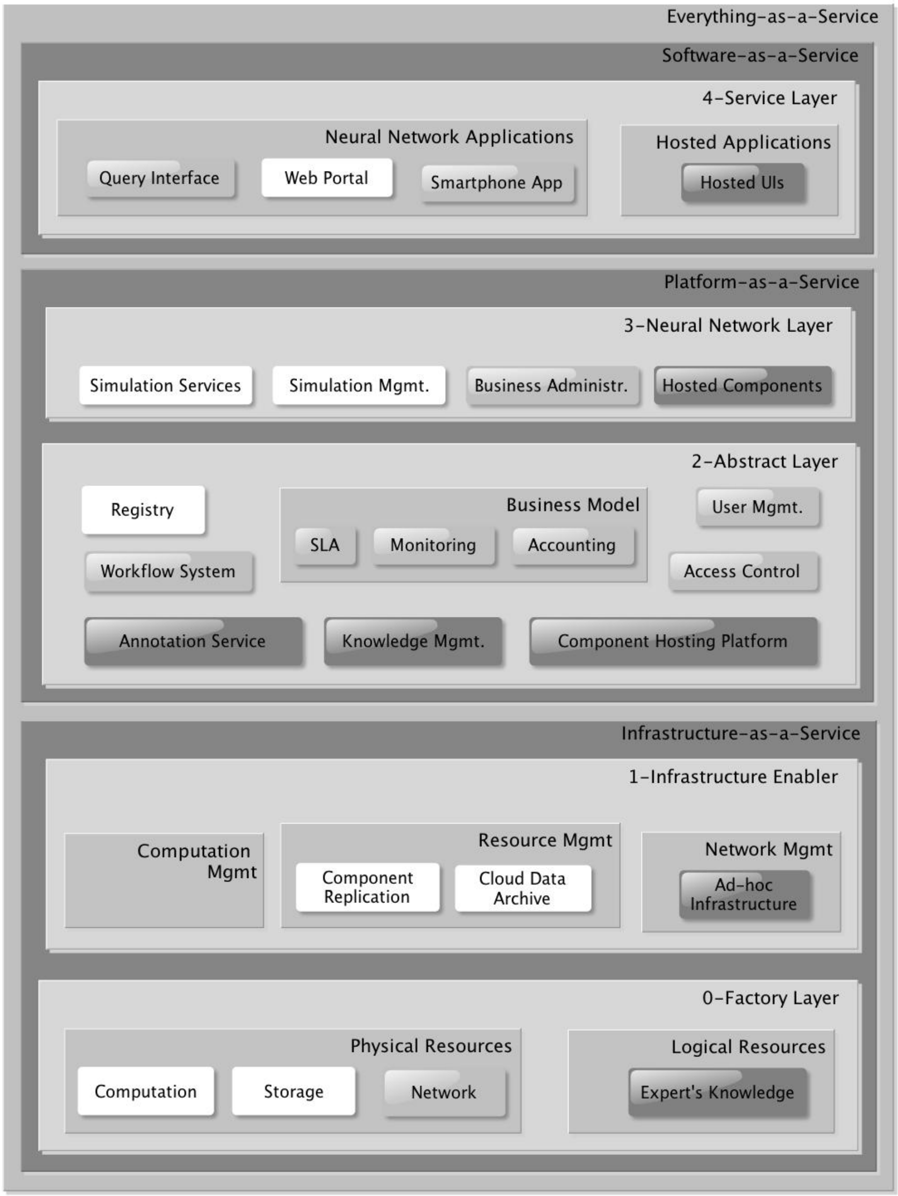


Fig. 1. N2Sky design based on RAVO

Platform as a Service (PaaS) provides computational resources via a platform upon which applications and services can be developed and hosted. PaaS typically makes use of dedicated APIs to control the behaviour of a server hosting engine which executes and replicates the execution according to user requests. It provides transparent access to the resources offered by the IaaS layer and applications offered by the SaaS layer. In N2Sky it is divided into two sublayers:

- Abstract Layer (Layer 2): This layer contains domain-independent tools that are designed not only for use in connection with neural networks.
- Neural Network Layer (Layer 3): This layer is composed of domain-specific (i.e. neural network) applications.

Software as a Service (SaaS) offers implementations of specific business functions and business processes that are provided with specific Cloud capabilities, i.e. they provide applications / services using a Cloud infrastructure or platform, rather than providing Cloud features themselves. In context of N2Sky, SaaS is composed of one layer, namely the Service Layer.

- Service Layer (Layer 4): This layer contains the user interfaces of applications provided in Layer 3 and is an entry point for both end users and contributors. Components are hosted in the Cloud or can be downloaded to local workstations or mobile devices.

Each of the five layers provide its functionality in a pure service-oriented manner so we can say that N2Sky realizes the Everything-as-a-Service paradigm.

4 N2Sky Cloud Deployment

N2Sky facilitates Eucalyptus [11], which is an open source software application that implements a Cloud infrastructure (similar to Amazon's Elastic Compute Cloud) used within a data center. Eucalyptus provides a highly robust and scalable Infrastructure as a Service (IaaS) solution for Service Providers and Enterprises. A Eucalyptus Cloud setup consists of three components the Cloud controller (CLC), the cluster controller(s) (CC) and node controller(s) (NC). The Cloud controller is a Java program that, in addition to high-level resource scheduling and system accounting, offers a Web services interface and a Web interface to the outside world. Cluster controller and node controller are written in the programming language C and deployed as Web services inside an Apache environment.

Communication among these three types of components is accomplished via SOAP with WS-Security. The N2Sky System itself is a Java-based environment for the simulation and evaluation of neural networks in a distributed environment. The Apache Axis library and an Apache Tomcat Web container are used as a hosting environment for the Web Services. To access these services Java Servlets/JSPs have been employed as the Web frontend.

N2Sky system can be deployed on various configurations of the underlying infrastructure. It is even possible to use a federated Cloud approach, by fostering the specific capabilities (affinities) of different Cloud providers (e.g. data Clouds, compute Clouds, etc.). A possible specific deployment is show in Figure 2.

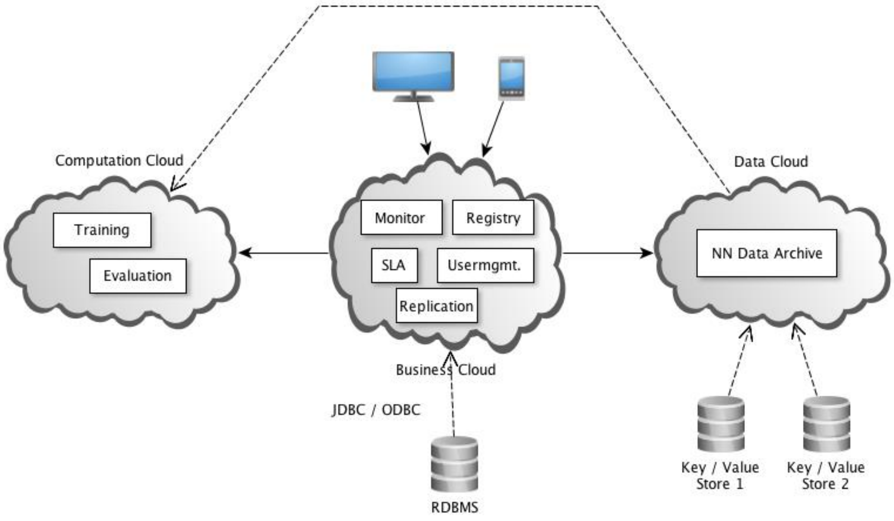


Fig. 2. N2Sky Cloud Deployment

5 N2Sky Interface

The whole system architecture and its components are depicted in Figure 3.

A neural network has to be configured or trained (supervised or unsupervised) so that it may be able to adjust its weights in such a way that the application of a set of inputs produces the desired set of outputs. By using a particular paradigm selected by the user the *N2Sky Simulation Service* allows basically three tasks: **train** (the training of an untrained neural network), **retrain** (training of a previously trained network again in order to increase the training accuracy), **evaluate** (evaluating an already trained network). The *N2Sky Data Archive* is responsible to provide access to data of different objects (respectively paradigms) of neural networks by archiving or retrieving them from a database storage service. It can also publish evaluation data. It provides the method *put* (inserts data into a data source) and *get* (retrieves data from a data source) The main objective of the *N2Sky Database Service* is to facilitate users to benefit from already trained neural networks to solve their problems. So this service archives all the available

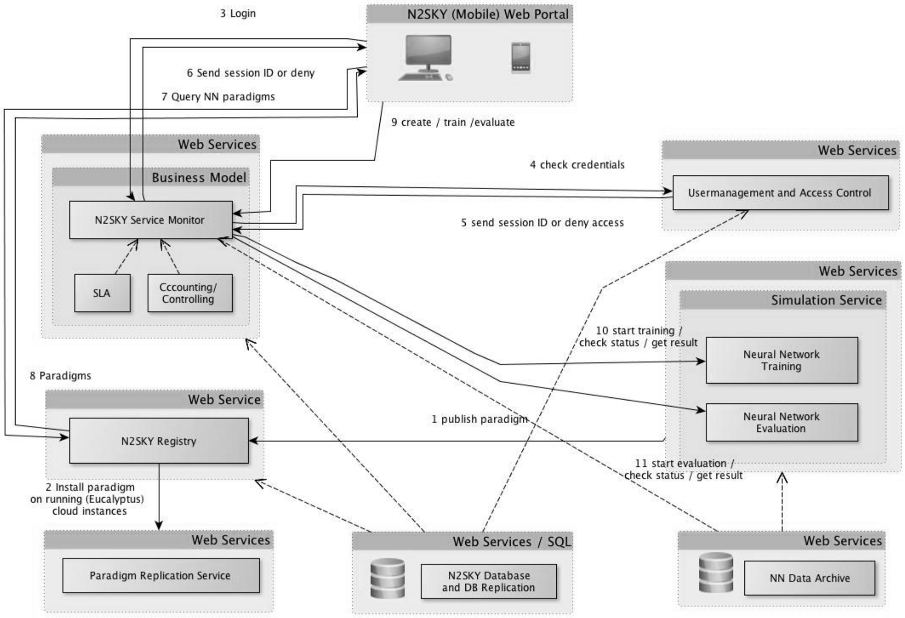


Fig. 3. N2Sky Architecture and Components

neural network objects, their instances, or input/output data related to a particular neural network paradigm. This service dynamically updates the database as the user gives new input/output patterns, defines a new paradigm or evaluates the neural network. The *N2Sky Service Monitor* keeps tracks of the available services, publishes these services to the whole system. Initially user interact with it by selecting already published paradigms like Back Propagation, Quick Propagation, Jordan etc. or submit jobs by defining own parameters. This module takes advantage of virtualization and provides a transparent way for the user to interact with the simulation services of the system. It also allows to implement business models by an accounting functionality and restricting access to specific paradigms. The *N2Sky Paradigm/Replication Service* contains the paradigm implementation that can be seen as the business logic of a neural network service implementation. The *N2Sky Registry* administrates the stored neural network paradigms. The main purpose of N2Sky system is to provide neural network data and objects to users. Thus the *N2Sky Java Application/Applet* provides a graphical user interface (GUI) to the user. It especially supports experienced users to easily run their simulations by accessing data related neural network objects that has been published by the N2Sky service manager and the N2Sky data service. Moreover the applet provides a facility to end-users to solve their problems by using predefined objects and paradigms. For the purpose of thin clients a simple Web browser, which can execute on a PC or a smart phone, can be used to access the front-end, the *N2Sky (Mobile) Web Portal*. It is relying on the *N2Sky User management Service* which grants access to the system.

The user can choose to work with the N2Sky system via a PC or a smart phone (e.g. an iPhone).

The N2Sky interface provides screen for the classical neural network tasks:

- **Subscription:** Choosing published existing neural network paradigms and instantiating new neural networks based on this paradigm.
- **Training:** Specifying training parameters, starting training and monitoring the training process.
- **Evaluation:** Using trained neural networks for problem solution.

A specific highlight of the N2Sky system is the use of the standardized and user-friendly SQL language for searching for network paradigms and objects and defining the training and evaluation data set. This unique feature allows for combining globally stored, distributed data within the N2Sky environment easily.

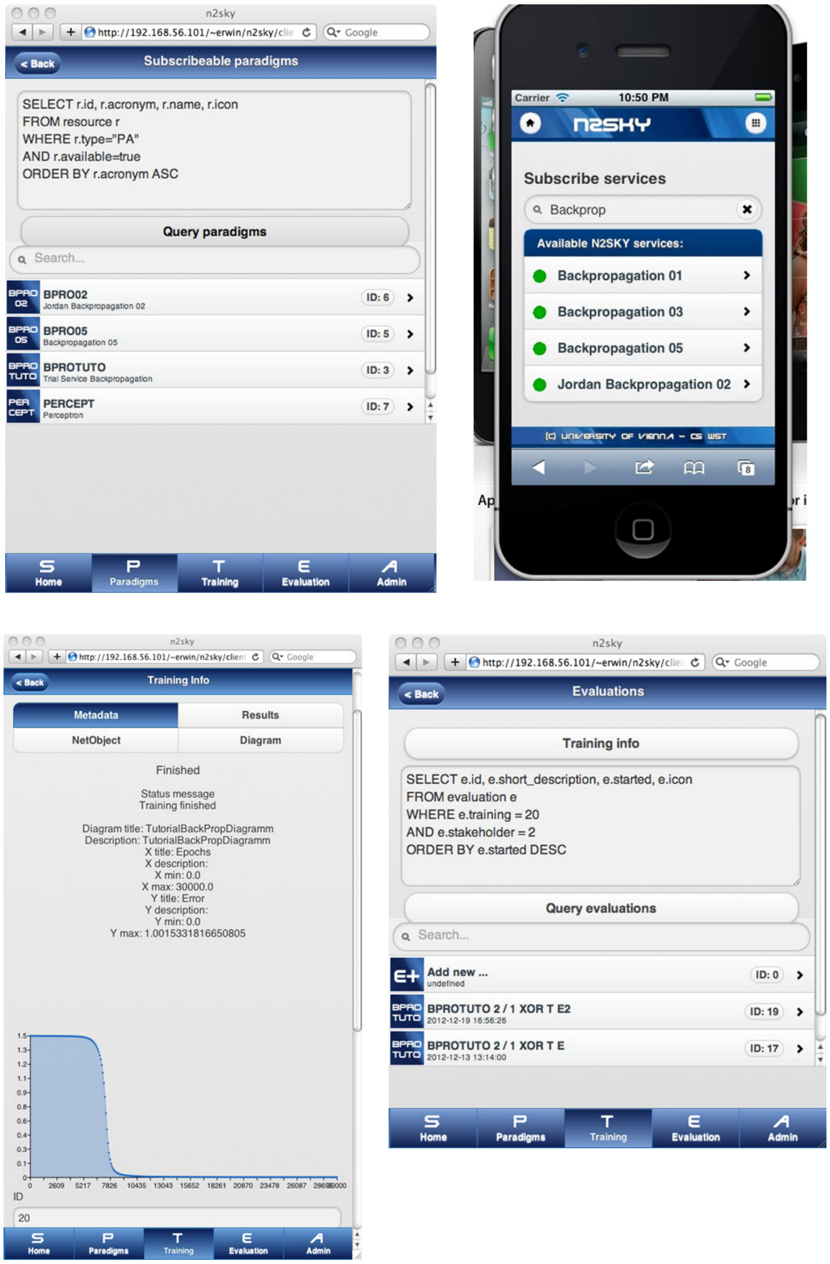
6 Conclusion and Future Work

In this paper we presented N2Sky, a Cloud-based framework enabling the Computational Intelligence community to share and exchange the neural network resources within a Virtual Organisation. N2Sky is a prototype system with quite some room for further enhancement. Ongoing research is done in the following areas:

- We are working on an enhancement of the neural network paradigm description language ViNNSL [12] to allow for easier sharing of resources between the paradigm provider and the customers. We are also aiming to build a generalized semantic description of resources for exchanging data.
- Parallelization of neural network training is a further key for increasing the overall performance. Based on our research on neural network parallelization [13] we envision an automatically definition and usage of parallelization patterns for specific paradigms. Furthermore the automatic selection of capable resources in the Cloud for execution, e.g. multi-core or cluster systems is also a hot topic within this area.
- A key element is to find neural network solvers for given problems, similar to a ”Neural Network Google”. In the course of this research we are using ontology alignment by mapping problem ontology onto solution ontology.

References

1. UK e-Science: UK e-Science programme. http://www.escience-grid.org.uk
2. SOM Programming Team SOM-PAK: The self-organizing map program pakage, user guide. (1992)



```
SELECT r.id, r.acronym, r.name, r.icon
FROM resource r
WHERE r.type="PA"
AND r.available=true
ORDER BY r.acronym ASC
```

BPRO 02	BPRO02 Jordan Backpropagation 02	ID: 6
BPRO 05	BPRO05 Backpropagation 05	ID: 5
BPRO TUTO	BPROTUTO Trial Service Backpropagation	ID: 3
PER CEPT	PERCEPT Perceptron	ID: 7

Finished

Status message
Training finished

Diagram title: TutorialBackPropDiagramm
Description: TutorialBackPropDiagramm
X title: Epochs
X description:
X min: 0.0
X max: 30000.0
Y title: Error
Y description:
Y min: 0.0
Y max: 1.0015331816650805

ID: 20

```
SELECT e.id, e.short_description, e.started, e.icon
FROM evaluation e
WHERE e.training = 20
AND e.stakeholder = 2
ORDER BY e.started DESC
```

E+	Add new ... undefined	ID: 0
BPRO TUTO	BPROTUTO 2 / 1 XOR T E2 2012-12-19 16:56:26	ID: 19
BPRO TUTO	BPROTUTO 2 / 1 XOR T E 2012-12-13 13:14:00	ID: 17

Fig. 4. N2Sky Interface

3. A. Zell, G. Mamier, M. Vogt, N. Mache, R. Hbner, S. Dring, K.-U. Herrmann, T. Soyez, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, D. Posselt, T. Schreiner, B. Kett, G. Clemente, and J. Wieland: SNNS Stuttgart Neural Network Simulator user manual. Technical report, University of Stuttgart. (March 1992)
4. Schikuta, E.: NeuroWeb: an Internet-Based neural network simulator. In: 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'02), Washington, D.C., USA, IEEE Computer Society (November 2002) 407–414
5. Brunner, C., Schulte, C.: NeuroAccess: The Neural Network Data Base System. Master's thasis, University of Vienna, Vienna, Austria (1998)
6. Schikuta, E., Glantschnig, P.: NeuroOracle: Integration of Neural Networks into an Object-Relational Database System. In: ISNN '07: Proceedings of the 4th international symposium on Neural Networks, Berlin, Heidelberg, Springer-Verlag (2007) 1115–1124
7. Huqqani, A.A., Xin, L., Beran, P.P., Schikuta, E.: N2Cloud: Cloud based Neural Network Simulation Application. In: Neural Networks (IJCNN), The 2010 International Joint Conference on. (July 2010) 1–5
8. Schikuta, E., Weishäupl, T.: N2Grid: Neural Networks in the Grid. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. Volume 2. (July 2004) 1409–1414
9. IBM: IBM Blue Cloud. http://www-03.ibm.com/press/us/en/photo/22615.wss (November 2007) last accessed Jan 20, 2010.
10. Khalil, W.: Reference Architecture for Virtual Organization. Ph.d. thesis, Faculty of Computer Science, University of Vienna, Austria (2012)
11. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on. (2009) 124–131
12. Beran, P.P., Vinek, E., Schikuta, E., Weishäupl, T.: ViNNSL - the Vienna Neural Network Specification Language. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008. (2008) 1872–1879
13. Weishäupl, T., Schikuta, E.: Cellular Neural Network Parallelization Rules. In: CNNA '04: Proceedings of the 8th IEEE International Biannual Workshop on Cellular Neural Networks and their Applications, Los Alamitos, CA, USA, IEEE Computer Society (2004)