Searching the Sky for Neural Networks

Erich Schikuta[1], Abdelkader Magdy[1], Irfan Ul Haq[2], A. Baith Mohamed[1], Benedikt Pittl[1], and Werner Mach[1]

[1] University of Vienna, Vienna, Austria,
erich.schikuta@univie.ac.at, shaabana52@univie.ac.at, abdel.baes.mohamed@univie.ac.at, beneditk.pittl@univie.ac.at, werner.mach@univie.ac.at

[2] Department of CIS, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan, irfanulhaq@pieas.edu.pk

Abstract. Sky computing is a new computing paradigm leveraging resources of multiple Cloud providers to create a large-scale distributed infrastructure. N2Sky is a research initiative promising a framework for the utilization of Neural Networks as services across many Clouds integrating into a Sky. This involves a number of challenges ranging from the provision, discovery and utilization of N2Sky services to the management, monitoring, metering and accounting of the N2Sky infrastructure. This paper focuses on the semantic discovery of N2Sky services through a human-centered querying mechanism termed as N2Query. N2Query allows N2Sky users to specify their problem statement as natural language queries. In response to the natural language queries, it delivers a list of ranked neural network services to the user as a solution to their stated problem. The search algorithm of N2Query is based on the semantic mapping of ontologies referring to problem and solution domains.

Keywords: Neural Network as a Service, Virtual Organization, Semantic Description, Cloud Computing

1 Introduction

Sky providers aggregate the services scattered across various Cloud-based infrastructures to provide the concept of sky computing. The sky computing in this way copes with the problem of vendor lock-in and extends the flexibility, transparency and elasticity of the integrated infrastructure as compared to that of a single Cloud. Sky computing has taken another step forward towards the realization of virtual collaborations, where solutions are virtual and resources are logical. The exchanging data among researchers is the main stimulus point for the development. This is just as valid for the neural information processing community as for any other research community [13]. As described by the UK e-Science initiative [16] many goals can be achieved by using new stimulation techniques, such as enabling more effective and seamless collaboration for scientific and commercial communities.

In the context of our work, we designed N2Sky a virtual neural network (NN) simulator for the computational intelligence community [14]. It provides access to neural network resources and enables infrastructures fostering multi Cloud resources.On the one hand, neural network resources can be generic neural networks trained by a specific learning paradigm and training data for given problems whereas on the other hand they can represent already trained networks, which can be used for given application problems. The vision of N2Sky is the provisioning of neural networks where any member of the community can access or contribute neural networks all over the Internet.

The number of neural networks is expected to be very large and continuously growing. These neural network objects are distributed on a worldwide scale on the Internet administratively under the umbrella of the N2Sky virtual organisation on participating resource nodes. Searching for specific neural network resources providing solutions to given problems can be a time consuming and difficult task. We developed an ontology-based approach for searching semantically the resource pool of N2Sky, where the generic idea was presented in [12]. In this paper, we present N2Query, an implementation of our generic semantic query approach as a component of the N2Sky infrastructure. It allows N2Sky users to search for neural networks by using natural language. The N2Query component is depicted in figure 1. The N2Query architecture consists of:

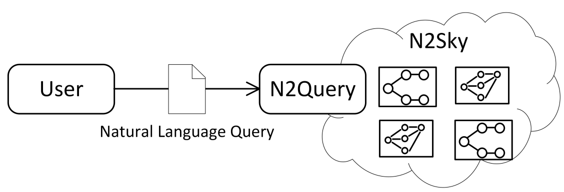

Fig. 1. N2Query in context with N2Sky

– A semantic querying interface that allows the user to specify his/her problem description in natural language form (query).
– An ontology mapping mechanism that allows N2Query to recognise the semantic of the natural language query in form of an ontology (called problem ontology). Then a mapping algorithm is applied to match N2Query's problem ontology against an already constructed solution ontology resulting in a list of adequate neural networks.
– A ranking mechanism to deliver a list of links to neural network resources of the N2Sky virtual organisation for solving the problem.
– An XML based Neural Network resource representation language to maintain and search the problem and solution ontologies.

The structure of the paper, is as follows: The state of the art of neural network simulators and the baseline research are given in section 2. Section 3

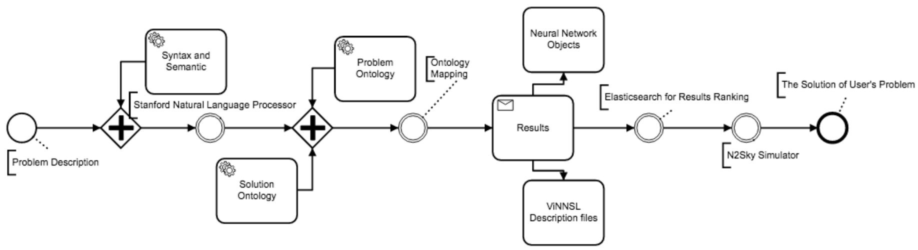

Fig. 2. N2Query Components Control Flow

highlights the activities of the overall N2Query process. Section 3 discusses the architecture of all services and components of the N2Query engine. A use case and the execution process of the N2Query are presented in section 4. Finally, the paper concludes our findings and presents our plans for future work.

2 Related Work

Over the last few years, a lot of simulation environments have been developed to mimic the behaviour of artificial and biological neural networks [9]. IQR [1] is a neurone simulator which allows neuronal models to control the behaviour of real devices in real-time. NeuroSpace [3] aims to integrate neural networks into relational database. NEUVISION [8] is a simulation environment used to simulate large-scale neuronal networks. NeuroWeb [11] lets users exchange information (neural network objects, neural network paradigms) and exploits available computing resources for neural network specific tasks (specifically training of neural networks).

Actually, there is no simulator or environment cover all simulation approaches, being able to solve a different kind of problems in the Cloud. In the course of our research, we designed and developed N2Sky [14], a virtual organization (VO) for the community of computational intelligence (CI), providing access to neural networks and enabling infrastructures to foster federated Cloud resources [12].

N2Sky supports qualified users to easily run their simulations by accessing data related neural network resources that has been published by the N2Sky service manager and the N2Sky data service [5]. Moreover, N2Sky 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 smartphone, 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 [14].

N2Sky aroused strong interest even beyond the CI community[3]. This endeavour of providing an environment for the access to practically unlimited resources faces one specific challenge. We propose a centralised registry approach

[3] http://cacm.acm.org/news/171642-neural-nets-now-available-in-the-Cloud/

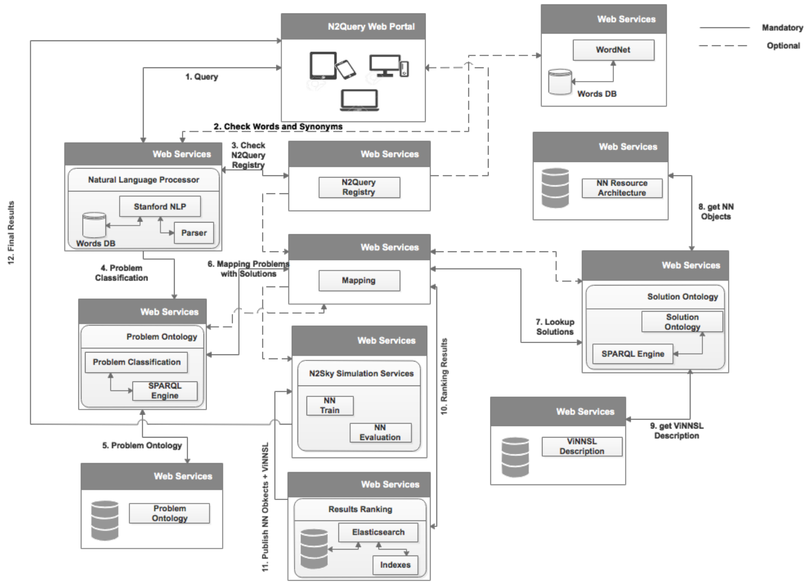

Fig. 3. N2Query Architecture and Components

collecting all semantic knowledge of neural network objects by semantic web technologies [9].

N2Sky offers neural network resources as a service which dynamically uses the available computing environment to reduce the execution time. Summing up, the N2Sky environment provides [14]:

- Sharing of neural network paradigms, objects and related information between the researchers and end user world wide.
- Reduction of training time of neural network by automatically selecting appropriate parallel implementations of the neural network services exploiting suitable Cloud resources.
- Transparent access to High-end neural network resources stored in Cloud environment.
- Uniform Look and feel for location independence of computational, storage and Network resources.

N2Sky uses ViNNSL description language for describing neural networks paradigm to allow for easier sharing of resources between the paradigm provider and the customers [6].

3 N2Query Architecture

Using the N2Query users can submit his/her query formulated in natural language through the N2Query interface, which further interacts with the N2Sky infrastructure to look for the plausible solutions for the user and responds through the same interface. Figure 2 shows the high level process of N2Query. The detail of every stage through the process will be covered in the subsequent sections however a rather general description is aimed here to gradualy build the understanding of the reader. After the user submits his/her query in the natural language form, the Stanford Language Processor is applied for the syntax and semantic analysis of the query. The processed problem statement is then sought out through the already built problem ontology. The nodes of the problem ontology that are marked relevant with the user problem are then linked with the relevant nodes of already available solution ontology producing the ontology mapping for the specific problem. The ontology mapping results into the identification of the neural networks required to solve the submitted problem. The retrieved neural networks are afterwards ranked by applying ElasticSearch and are conveyed to the user as the solution of his problem.

The architecture of N2Query and the system services are depicted in figure 3.

The N2Query (Mobile) Web Portal It provides the access point to the N2Query system by a web browser interface which can be used on PCs, tablets or even a smartphones. N2Querry provides two different interfaces to the user, a free text and a directory search. Figure 5 shows, the N2Query free text interface. The directory interface depicted in figure 7 allows a category search and offers a brief description of the mechanism of N2Query interface by providing some examples to show how the user can interact with N2Query interface.

Natural Language Processor Service This service analyzes the semantic definition behind user query. The NLP service applies five steps (Lexical Analysis, Syntactic Analysis (Parsing),Semantic Analysis, Discourse Integration, Pragmatic Analysis) to understand the meaning of statements.

Problem Ontology service: This service is based on a hierarchy of known neural network problems. This service aims to classify the user's problem under one or more categories of typical neural network problems, as approximation, optimisation, searching etc.

Solution Ontology service: This service stores all solutions provided by managed neural network resources in the N2Sky virtual organisation in a hierarchical form. The solution ontology can specify one or more solutions for a specific problem.

NN Resource Architecture web service: This service administrates N2Sky neural network objects which are considered as solution(s) for a specific problem and can be used under the umbrella of the N2Sky simulator.

ViNNSL Description web service: This service is responsible for attributing N2Sky with ViNNSL description files. These files describe the structure of managed neural network objects, which are used by N2Sky for creating and training neural networks.

WordNet web service: WordNet is a big lexical database of English language. Nouns, verbs, adverbs and adjectives are grouped into sets of synonyms (synsets), each expressing a distinct concept [10]. This service delivers a list of words and synonyms to the NLP service to create lists of all synonyms of the user problem.

N2Query Registry: This service contains all well answered queries. N2Query checks first if the user question is already asked before. If so, this service delivers the stored solution (NNs + ViNNSL) to N2Sky to retrain the network and get solution(s).

Elasticsearch service: This service is responsible for ranking solutions and publishes a list of filtered solutions.

Mapping service: This service provides the mapping technique to match a specific problem to possible solutions using SPARQL query language and then deliver list of solutions of that problem.

Three ontology combination paradigms can be distinguished, ontology linking, ontology mapping, and ontology importing [4]. For our problem, we apply ontology linking, where individuals from distinct ontologies are connected with links.

The concept is as follows: We administer basically two ontologies, a problem ontology and a solution ontology [6]:

- The *problem ontology* consists of a hierarchical organisation of typical neural network application problems, as classification, optimization, approximation, storage, pattern restoration, cluster analysis, feature extraction etc. In the ontology hierarchy these main domains are finer distinguished till the single problem specifications show up in the leave nodes.
- The *solution ontology* stores all known N2Sky neural networks organized according to their paradigm, as perceptron, multi-layer backpropagation, self-organizing maps (Kohonen cards), recurrent networks (Elman, Jordan, etc.), cellular neural networks, etc. Here the ontology delivers a fine grained structure finally giving the neural network objects (trained neural networks for a specific problem) as leaves.

Figure 4 depicts the two ontologies for solving the problem of "Face Recognition" (see case study section). Our N2Query component has already connected them (the nodes are connected with links). The links between problem-solution ontology simulate the mapping process between the defined problem with solution(s). We generate a mapping of problem ontology nodes, describing a specific problem, to solution ontology nodes, denoting network objects, which deliver a solution for this problem. Links can be defined not only between leaves of the hierarchies but also between internal nodes.

Based on the ViNNSL semantic information of an N2Sky resource the mapping between problems to solutions can be done by N2Sky administrators manually, and by an automatic mapping during insertion of the new network objects.

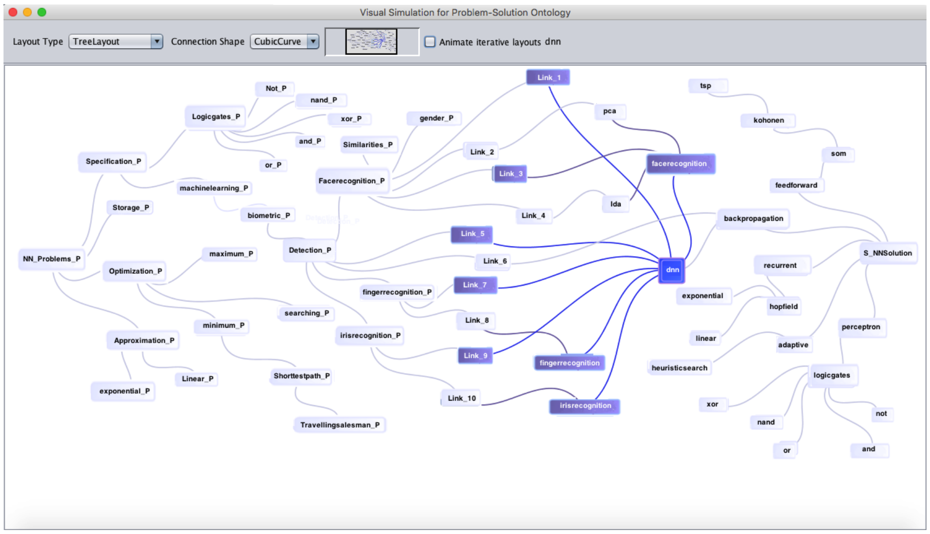

Fig. 4. Mapping between solution ontology (right) and the problem ontology (left) for the "Face Recognition" Problem

Integrating with the Ontologies This workflow for integrating new neural network resources into the knowledge repository of N2Sky can be described by the following algorithm A1 [12].

Algorithm A1:

1. N2Sky resource provider (*RP*) attributes its Neural Network Resource (*NNR*) with a ViNNSL Description (*VD*) specifying structural and semantic information.
2. *RP* sends *VD* together with *NNR* or URI of *NNR* to N2SKy knowledge repository.
3. *VD* is integrated into Problem Ontology (*PO*) according to problem domain.
4. *NNR* or its URI is integrated into Solution Ontology (*SO*) according to network paradigm.
5. Link between *VD* insertion node in *PO* and *NNR* insertion node in *SO* is created.

Querying the Ontologies The search algorithm is as follows: Based on the natural language keywords of the user query a scan over the problem ontology is performed. Hits, patterns matching the scan, resembled by nodes in the hierarchy, are collected and the links to the solution ontology are followed. There, a scan of the network objects, representing solutions to the problems, is done and fitting results are reported to the user. The sequence of the results can be guided by a fitting rank of problem to solution matches. By this approach, the effort for delivering matching neural network resources is centralized in the management service. By this approach, the number of network resources to be checked is pruned dramatically by only checking (solution) resources which are (obviously) targeting the problem domain. The generic workflow for executing a query on the knowledge repository can be described by the following algorithm A2 [12].

Algorithm A2:

1. User describes in natural language his/her Problem Description PD using N2Query interface.
2. Cognitive representation of the problem description delivering a synset SS (set of cognitive synonyms).
3. Problem description is classified according to SS in PO delivering set of PO nodes.
4. Links from PO nodes to respective SO nodes are followed.
5. SO nodes are starting points of tree search delivering URIs of possible solution candidates.
6. VD of solution candidates are analyzed and ranked according to match with PD.
7. Ranked list is reported to user.

4 N2Query Case Study

In this work, we choose the problem of Face-Recognition as a case study example [7].

In our use case, the user submits a query "How to solve face recognition problem" as shown in figure 5. The query is then processed and the results are displayed in the same interface.

Our approach to solve this problem consists basically of two phase, the ontology integration phase and the ontology query phase.

4.1 Ontology Integration Phase

The prerequisite for the user query is that the required neural network object must be present in the ontology architecture as well as the required ontology mapping should already be available within the system. The search process is performed on this ontology architecture. In the integration phase a neural network resource, e.g. a trained neural network object, which is provided by a member of the N2Sky virtual organization, is entered into the solution ontology. We use as running example the face recognition problem [7]. It is assumed the respective Backpropagation network was realized and contributed to N2Sky. Hereby the algorithm A1, presented in section 3, has to be executed:

The provider of the NN resource, the Backpropagation network, uses the ViNNSL language for the description of the problem and paradigm domain in step A1/1. Hereby, the `paradigm` and `problem domain` tags of ViNNSL are used.

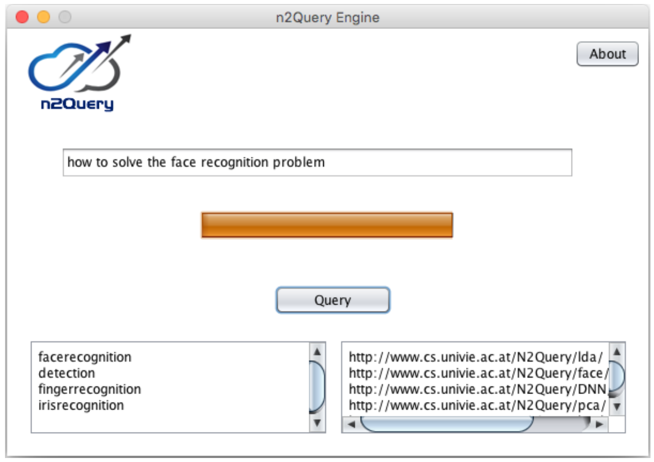

Fig. 5. N2Query Free Text User Interface

In step A1/3 the description of the NN resource, Classification → MachineLearning → Biometric → FaceDetection → Gender, is integrated into the problem Ontology.

The NN resource is integrated into the Solution Ontology accordingly to its paradigm family in step A1/4, Backpropagation → DNN → FaceRecognition → PCA.

In step A1/5 an appropriate link from problem to Solution Ontology is created pointing from problem description to the respective physical NN resource, see figure 4.

4.2 Ontology Query Phase

In the following we show how a natural language free text user query is analyzed by the N2Query system.

The following query analysis steps refer to the workflow from figure 3, marked as ordered numbers integrating various components of the N2Query architecture.

1. The user query is sent to the Natural Language Processing (Stanford Parser) web service to analyze the semantic of that query.
2. The NLP web service connects with WordNet web service which delivers a list of words and synonyms of the user problem.
3. If the query has been asked before, the N2Query Registry service sends all details to the Mapping service. This service is responsible for gathering neural network paradigms as solutions of that problem. Afterwards, the Mapping web service publishes solution(s) to N2Sky for retraining networks.
4. The Problem Ontology service receives the recognised statement of the user query in form of tokens. That service classifies the user problem under the hierarchal structure of the most known neural network problems.

5. The SPARQL query algorithm is applied on the problem ontology to match the user problem with stored problems. Conclusively the SPARQL query engine sends all matched classifications to the Mapping service.

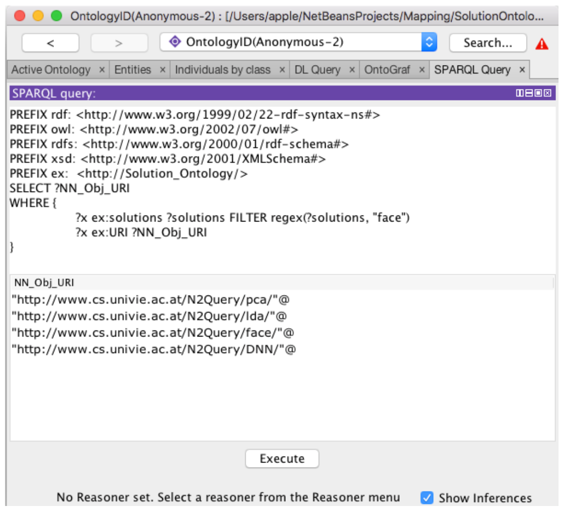

Fig. 6. SPARQL Query and List of URI's of NN Objects as Solutions Using Protégé

6. The Mapping service matches the problem(s) to the solution ontology. Figure 4 shows the mapping between solution ontology and the problem ontology for the face recognition problem.
7. Solution Ontology service looks up the solutions by a respective SPARQL query.
8. The Solution Ontology service gets the respective neural network objects from the NN Resources Architecture service. Figure 6 shows the SPARQL query and the list of URIs of neural network objects as solutions using Protégé ontology tool.
9. The solution Ontology service receives also the respective ViNNSL description file(s) which describe the received neural network objects.
10. The Elasticsearch service is applied on the received solution(s) for ranking and filtering results and publishes a list of solutions of the user problem to the N2Sky simulator.
11. N2Sky receives the published solution(s) (NN objects and ViNNSL) and starts creating, training, retraining and evaluating neural networks for the user problem.
12. N2Sky send the final result(s) to the user as shown in the bottom right pane in figure 5. Figure 7 represents another possibility to present results in a structured directory interface.

In this process, we use the Stanford parser for Natural Language Processing. The problem-solution ontologies are implemented by RDF and processed by the

OWL semantic web languages [15]. Huge storage repositories for RDF data have been developed, which store the RDF triples in a relational database (RDB) [2]. So, we use the Eclipse RDF4J Framework for ontology storage. According to the steps in the last section N2Query starts to map from the solution to the problem ontology to retrieve the existing solutions to that problem. In this process we use SPARQL language for the matching process. Thus, specifically the algorithm A2, see section 3, is executed by applying sematic web techniques:

In step A2/1, querying the N2Query tool by a natural language phrase like, "How to Solve the Face Recognition Problem", the N2Query system tries to recognise the semantic representation of this problem using Stanford natural language processor (step A2/2), and receives in the synnet set a phrase like "Face Recognition". In step A2/3 the SPARQL Query algorithm classifies the user query into the problem domain, and, following the link to the Solution Ontology (step A2/4), delivers by a subtree traversal (A2/5) the possible URIs of the existing NN objects. Based on the ViNNSL descriptions in step A2/6 the system produces and reports a ranked list of qualified solution URIs to the user's problem (A2/7).

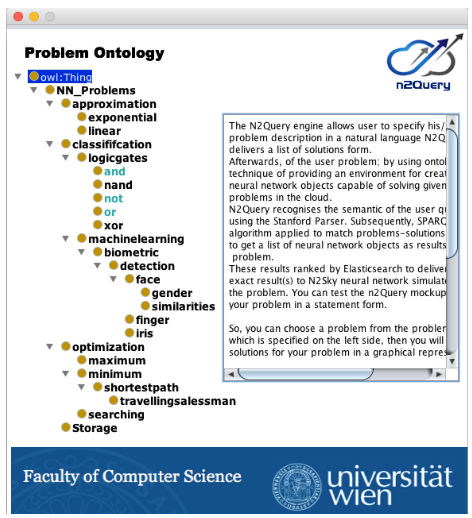

Fig. 7. N2Query Directory and Guide Interface

5 Conclusion and Future Work

The N2Sky project manages neural network resources on a worldwide basis. Provisioning of adequate networks to given problems is a critical success factor of the N2Sky project. Hence we developed the N2Query component which is able to process natural language queries for finding matching networks.

In this paper, we have outlined the technical architecture and implementation of the N2Query system using semantic web tools. Further, we defined the different workflows for the ontology management and query processing.

N2Query is an integral part of the next release of N2Sky which delivers a comprehensive redesign of N2Sky's architecture by micro-services, docker technique and dynamic Cloud deployment.

References

1. Bernardet, U., Blanchard, M., Verschure, P.F.: Iqr: a distributed system for real-time real-world neuronal simulation. Neurocomputing 44, 1043–1048 (2002)
2. Bönström, V., Hinze, A., Schweppe, H.: Storing rdf as a graph. In: Web Congress, 2003. Proceedings. First Latin American. pp. 27–36. IEEE (2003)
3. Cornelis, H., De Schutter, E.: Neurospaces: separating modeling and simulation. Neurocomputing 52, 227–231 (2003)
4. Homola, M., Serafini, L.: Towards formal comparison of ontology linking, mapping and importing. Proc. DL 10, 291–302 (2010)
5. Huqqani, A.A., Li, X., Beran, P.P., Schikuta, E.: N2cloud: Cloud based neural network simulation application. In: Neural Networks (IJCNN), The 2010 International Joint Conference on. pp. 1–5. IEEE (2010)
6. Huqqani, A.A., Mann, E., Schikuta, E.: Novel concepts for realizing neural networks as services in the sky. Procedia Computer Science 29, 2315–2324 (2014)
7. Huqqani, A.A., Schikuta, E., Ye, S., Chen, P.: Multicore and gpu parallelization of neural networks for face recognition. Procedia Computer Science 18, 349–358 (2013)
8. Mulas, M., Massobrio, P.: Neuvision: a novel simulation environment to model spontaneous and stimulus-evoked activity of large-scale neuronal networks. Neurocomputing 122, 441–457 (2013)
9. Prieto, A., Prieto, B., Ortigosa, E.M., Ros, E., Pelayo, F., Ortega, J., Rojas, I.: Neural networks: An overview of early research, current frameworks and new challenges. Neurocomputing (2016)
10. Princeton University: WordNet: A Lexical Database for English. https://wordnet.princeton.edu
11. Schikuta, E.: Neuroweb: An internet-based neural network simulator. In: ICTAI. pp. 407–412 (2002)
12. Schikuta, E., Magdy, A., Mohamed, A.B.: A framework for ontology based management of neural network as a service. In: International Conference on Neural Information Processing. pp. 236–243. Springer (2016)
13. Schikuta, E., Mann, E.: A cloud-based neural network simulation environment. In: International Work-Conference on Artificial Neural Networks. pp. 133–143. Springer (2013)
14. Schikuta, E., Mann, E.: N2sky-neural networks as services in the clouds. In: Neural Networks (IJCNN), The 2013 International Joint Conference on. pp. 1–8. IEEE (2013)
15. Staab, S., Studer, R.: Handbook on ontologies. Springer Science & Business Media (2013)
16. UK e-Science: UK e-Science programme. http://www.escience-grid.org.uk (2016)