Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 603 – 608

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Design of a Prototype Neural Network for Smart Homes and Energy Efficiency

Tobias Teich, Falko Roessler, Daniel Kretz, Susan Franke*

[a] *University of Applied Sciences Zwickau, Dr.-Friedrichs-Ring 2A, 08056 Zwickau, Germany*

Abstract

As a part of smart homes, a subsystem consisting of three components including a neural network is designed to provide personalized services. Unique factor combinations of building specifics, user profiles and external influences lead to the necessity of self-adaptive systems for personal comfort. The system supports room temperature control in order to heat rooms energy-efficiently at a set time. Smart home systems require a software architecture that allows services to be deployed on virtual and hardware devices.The design of automated processes is the first step of later programming and implementation into smart home systems that will automatically supervise and re-train its components and will also allow live feedback.

Keywords: Neural networks; learning systems; energy efficiency; smart homes

1. Introduction

This work contributes to a collaborative research project at the UAS Zwickau. Newly refurbished tenements are provided with KNX bus bars that collect and distribute information as well as various sensors that capture data of electrical energy consumption, air quality, temperature and weather conditions [1]. In addition to thermal insulation, one aspect of the project to further reduce thermal energy consumption is to use the smart home system to support the heating process. In order to support reducing energy consumption, two different modes were implemented. The 'presence' mode provides all standard and individually programmed services. The 'absence' or 'sleeping' mode

* Corresponding author. Tel.: +49 375 536 3250; fax: +49 375 536 3104.
E-mail address: Susan.Franke@fh-zwickau.de

doi:10.1016/j.proeng.2014.03.032

ensures that all unnecessary electrical power sockets and services are turned off to prevent energy consumption of standby-devices. Temperature control can be programmed for each room separately on demand or by set timetables. While in absence mode, the system will keep the rooms at a stable, lower temperature to prevent energy loss through cool down of the walls. The heating process starts either by switching to presence mode or according to timetable. Both choices lead to one problem: The heating process will start too late and the tenant has to wait for his preferred temperature or the temperature is reached too early and energy loss results from maintaining a high temperature level for a longer time than necessary.

The objective of the current research and the author's work is to provide additional energy saving potentials by using means of intelligent algorithms such as neural networks. Data from sensors can be used to train neural networks that assist users in creating room based heating profiles. These profiles represent the individual preferences of a tenant and help to automatically adjust the starting time for heating. As a part of smart home systems, neural networks enable the design of highly personalized services on a basis of accessible data for profiling and context sensitive actions.

1.1. Limitations

Particular problems that this work is dealing with are the design of automated procedures that create, train and evaluate neural networks as well as providing interfaces between the control system and the operating system in which the trained neural network is embedded. The major tasks after the design and conceptual work on the automated processes include the programming and implementation process. As some milestones have already been achieved, the next steps are extending the working early prototypes to support automated learning and evaluation as well as preparing software architecture interfaces to start the first simulation under real conditions. The specific scenario for this work is limited to room temperature control. Future perspectives may see more small and easy-to-use applications within smart home projects.

2. Learning systems

Biologically inspired procedures such as genetic algorithms, ant colony optimization and neural networks apply to areas where traditional mathematical methods cannot be realized due to the fact that there are not enough resources to find an exact solution [2]. These procedures mimic the way of cooperation and information processing that can be found in nature. Artificial neural networks represent an abstract method comparable to the processing capabilities of the human brain. In modern practice they are used as parts of subsystems in algorithmic frameworks because they are not suitable to provide an overall system [3].

Neural networks are used to apply case-based reasoning for smart home applications such as eldercare, health-care and emergency solutions [4]. There are also approaches to energy-efficiency in wireless sensor networks, yet it is limited to energy consumption of a sensor nodes network itself [5]. Other works address forecasting models and decision support systems for more specific applications, e.g. electrical load forecasting or energy-efficiency in manufacturing. Most of the research concerning energy-efficiency and neural networks is conductedu for industrial needs. There are very few applications for the private living sector and no known projects that use neural networks for highly personalized tasks such as heating in Germany [6].

2.1. Conceptual and data requirements

Configuration of room-specific desired temperature providing individual comfort, depending on parameters such as weather conditions, seasonal influence and user profile needs to be automated, self-adaptive and possible without direct user interaction. Additional system requirements imply a low maintenance level, allowing live feedback from the user as well as self-evaluation and automatic re-training.

For the model of starting time estimation for the heating process, it is necessary to pre-process and adjust the input data, so they can be used with a neural network. The output that the neural network is designed to reproduce

has to be defined similarly. For designing the model, a set of core data consisting of the variables as in Table 1 is used. Additional dimensions are optional, e.g. direct sunlight duration. Depending on the individual tenement's specifics, i.e. location and cardinal direction bias, the influence of direct sunlight can be significant. Therefore it is necessary to identify possible factors in a tenement that could otherwise affect the learning performance under altering circumstances.

Table 1. Dimensions for the neural network.

ID	Variable	Source
I1	Outside temperature	Weather station building
I2	Wind (m/s)	Weather station building
I3	Room temperature	Sensors (live) + internal database
I4	Target temperature	Room control panel + database
I5	Direct sunlight	(optional)
I6	Presence information	Presence / absence mode activation (optional)
O1	Required time for heating	Target vector = output

The data have to be pre-processed, because activity of neurons can only have values of between 0 and 1 or -1 and 1 respectively. Data of temperature have both negative and positive values, so it was a logical decision to scale them within the [-1,1] interval.

3. Course of action

The first approach to building a stable neural network was done manually, though the objective of this work is the automation of most of the included tasks in order to create solutions for individual requirements on a larger scale. After the initial steps of creating the scenario as well as collecting and pre-processing necessary data, the neural network architecture was built. The early experiments for this research were made with a freeware tool that offers a graphical neural networks editor and simulator, which allows designing simple networks [7]. The results of the experiments are taken as reference for project-specific programming.

Creating a set of training data as well as one or multiple sets of test data for evaluating the quality of the network represents the next step. It is imperative to ensure that the training data consist of a representative variety of realistic data. Without that, the neural network is not able to create abstraction levels and to reproduce correct patterns under real conditions. Examining the mapping by representing the test data set(s) to the trained network is also necessary to evaluate the quality of the network. Although neural networks possess the feature that they are able to generate reasonable output with deviant or missing data and thus, show some sort of stability, in case of inability to learn correctly, it may be necessary to re-evaluate the scenario or alter data (e.g. in scale) or input dimensions.

3.1. Results of the manual test phase

The first models used a multidimensional input vector containing data of temperatures, time and weather conditions of a time frame of two months within the heating period. A supervised feed forward neural network with multiple layers of hidden units showed good results after a short training period. Individual tests were iterated with different setups of network architecture, i.e. hidden layer and unit count as well as the number of input dimensions. For network models that were similar in results, the model with the leaner architecture was preferred, to enhance performance and simplify later implementation. In order to harmonize the transition between seasons, it may also be necessary to add new (artificial) parameters that help prediction under the influence of seasonal specifics. For the scenario, one set of training data as well as multiple data sets for testing the neural network were created. The

creation of a training data set requires a representative variety of data. Without that, neural networksare not able to recreate its abstraction levels and to reproduce correct patterns under real conditions.

For this kind of scenario, several possible solutions were found, including different types of simple recurrent networks (SRN), though the networks which showed the best results were types of feed forward neural networks. Figure 1 shows the training result of the best suited networks. The thicker line represents the target output, while the thinner line shows the behavior (actual output) of the neural network.

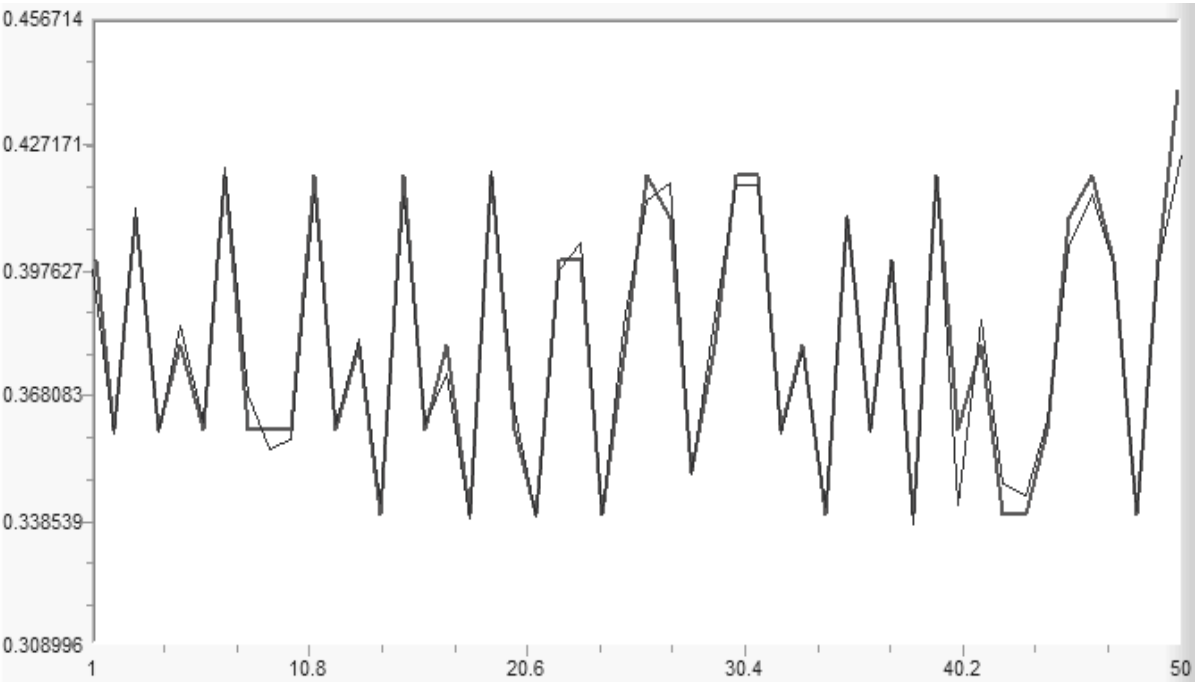

Fig.1. target and actual output of the neural network (x-axis = pattern number; y-axis = Output).

3.2. Designing the automated processes

The main task in order to enable automated processes for easy-to-use applications and customization is to create a three-step service design. First, an overriding program (i) is needed to execute all necessary functions, including sending commands to the neural network component core (ii), which in turn incorporates methods of creating individual neural networks. The third component is the active element (iii) within a running service in the smart home system and uses the network from step two as a basis.

The overriding program is responsible for preparing the data needed for the learning process. This includes the execution of queries for both internal and external data sources via configured interfaces. After consolidation and completion, the data are used to create training and test data sets. It must be secured, that the selected data are relevant, hence the queries need to be very precise but at the same time flexible enough for additional requirements. The overriding program has to select the right algorithms depending on the tenement's respective characteristics. For that reason, it is necessary to create categories of different types of units.Before the training process can start, a neural network has to be created by the neural network component. Definition of required neuron types and numbers, activation functions and connections to other units is made by that component and will for the most part be pre-programmed and defined by manual model results. It is also necessary to choose an appropriate teaching algorithm as well as to configure the condition, when training will stop to save the current network. The progress of

teaching is evaluated by the 'net error' and reference results with one or multiple sets of test data. The overriding program supervises the output tables generated by the neural network. With the help of interfaces connecting the neural network component and the overriding program, output tables can be created to evaluate the training. If actual output data comply with the defined criteria for 'success', the current neural network setting is saved and exported to a subprogram, which contains the active element and integrated into the room control functions.

Once a neural network becomes part of the subsystem for room control and transfers to the operating mode, the active element retrieves timetables for the individual temperature preferences. At that time, the neural network has learned to start the heating process at the right time in order to reach the target temperature at the desired time and to minimize energy loss implied by a too long lasting heating process. The active element also allows live feedback and supervises the 'live success'. If required, it initializes new design or training processes by the overriding program.

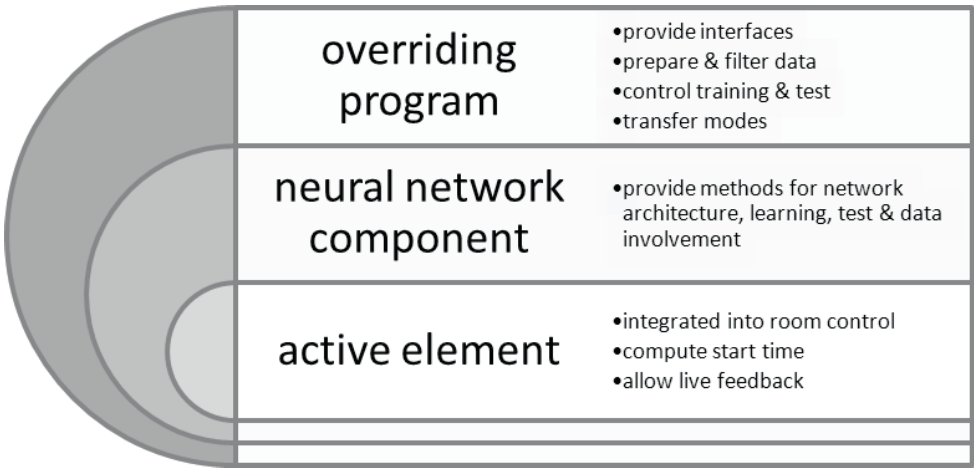

Fig.2. structure of the automated process design.

3.3. Embedded services for smart homes

Hardware devices that have been installed in the refurbished buildings and tenements provide the basis for realizing energy savings and building automation solutions. By taking entire building complexes into consideration, even bigger energy saving potentials can be utilized. The information exchange among devices and systems that is required in order to reach that goal can only be achieved, if all devices and systems use the same language. If that is not the case, cost intensive measures like additional converters and configuration efforts are necessary to connect hardware devices from different manufacturers with the system. Although using manufacturer-specific properties may lighten the workload at the beginning, it is considered very risky, because it causes non-reusable software components in consequence. Eventually, the installation of new devices from different vendors with similar functionality or integrating new bus systems results in an unpredictable re-factoring effort of existing software. In order to react to these kinds of changes, it is imperative to design software services and program modules as flexible as possible, to be able to adapt them as needed.

The work of another project within the research group at the UAS Zwickau deals with that very problem [8]. A dynamic Java based OSGi platform is used to create a flexible architecture. The framework supports software modularization, allows collecting services and service implementations into bundles as well as configuration in a modular way [9]. Application logic can be developed in different system environments with distinct automation technology, but it is necessary to decouple service implementation and the underlying hardware devices and bus systems. As a result, hardware devices become abstract units which are mapped into a uniform virtual

representation. These abstract units are not restricted to any fixed functionality but behave like data structures. Decisions or interaction logic is implemented solely by services and business processes. Device specific attributes and parameters are mapped into generalized data structures of virtual proxy objects. Time consuming hardware programming, which is redundant for similar devices, is avoided by extracting the device specific application logic into the service layer. This method can be used to install (sub-) programs that use a neural network as a core component e.g. for regulating room temperature. The service will be available to tenants by demand, which allows them to automate their preferred comfort room-based temperature profiles.

By generalizing hardware components within the technological infrastructure into standardized virtual objects, it is possible to develop completely reusable services and business processes. Instead of traditional redundantly programmed hardware devices, services can now provide the same functionality for arbitrary devices from an identical category. Data-rich environments are well-suited for analytical applications and decision support systems (DSS). There are many perspectives for such systems, including smart homes [10]. The solid OSGi platform makes decision support systems and intelligent algorithms realistic opportunities for smart homes, e.g. in the form of medical applications or elder-care solutions.

4. Conclusion

This work shows the stability of a neural network based solution for building personalized smart home systems. The scenario is the first one that is going to be implemented for the upcoming heating period in a number of test tenements in order to see the live system running. Future work includes further programming and the implementation process according to the automation design. Well-defined interfaces according to the software service architecture will enable more applications. One of the next prospective practical expansion is coupling the tenement-based neural network system with the across building heating system that will dynamically adjust flow temperature and pump power with the help of a specifically designed neural network.The integration process into the service architecture will represent a major milestone to develop new smart home subsystems. The ability to provide personalized functions simply by learning from user specific preferences and behavior is a huge factor that simplifies both the daily life of tenants or homeowners and the work implied by providing that kind of service.The first implementation of the automated service will show the qualities of the solution as well as further required work on the project. It may be necessary to iterate some of the design steps and to improve the algorithms in order to apply the service on a larger scale.

References

[1] Teich, T., Zimmermann, M. and Other (2010). Intelligent Building Automation.International Conference on Automation, Robotics and Control Systems, Karras, D.A., MoustafaK.A.F., Tang, D. (Eds.), ISRST. Orlando, Florida, pp. 53–57.

[2] Kramer, O., Computational Intelligence, Springer-Verlag, ISBN 978-3-540-79738-8, BerlinHeidelberg, Germany, 2009.

[3] Deco, G. & Schuermann, B., Information Dynamics: Foundations and Applications, Springer,ISBN 0–387–95047–8, Berlin Heidelberg, Germany, 2001.

[4] Leake, D., Maguitman, A. & Reichherzer, T., Cases, Context, and Comfort: Opportunities forCase-Based Reasoning in Smart Homes. Lecture Notes in Computer Science, Vol. 4008, ISSN 0302-9743, July 2006, pp.109-131.

[5] Enami, N., Moghadam, R. A., Dadashtabar, K. & Hoseini, M., Neural Network Based EnergyEfficiency in Wireless Sensor Networks: A Servey. Internationa Journal of Computer Science &Engineering Survey, Vol. 1, No. 1, DOI:10.5121/ijcses.2010.1104, August 2010, pp. 39-55.

[6] Botthof, A.; Domroese, W. & Gross, W. (2011). Technologische und wirtschaftlichePerspektiven Deutschlands durch die Konvergenz der elektronischen Medien, technical report.VDI/VDE Innovation + Technik GmbH, Berlin, Germany.

[7] Jetter, T., MemBrain, NN-Editor and Simulator, URL http://www.membrain-nn.de , 2013.

[8] Kretz, D., Teich, T., Kretzschmar, M., Neumann, T., Development of a general virtualization approach for energy efficient building automation, International Journal of Energy Engineering, Vol. 3, No. 1, ISSN:2225-6563, February 2013, pp. 1-6.

[9] McAffer, J., Vanderlei, P. & Archer, S. (2010). OSGi and Equinox: Creating Highly Modular Java Systems, Addison-Wesley, ISBN: 978-0321585714.

[10] Marschollek, M., Decision support at home (DS@HOME) – system architectures and requirements, BMC Medical Informatics and Decision Making, 12:43, DOI:10.1186/1472-6947-12-43, 2012.