Application of Neural Networks in High Assurance Systems: A Survey

Johann Schumann, Pramod Gupta, and Yan Liu

Abstract. Artificial Neural Networks (ANNs) are employed in many areas of industry such as pattern recognition, robotics, controls, medicine, and defence. Their learning and generalization capabilities make them highly desirable solutions for complex problems. However, they are commonly perceived as black boxes since their behavior is typically scattered around its elements with little meaning to an observer. The primary concern in safety critical systems development and assurance is the identification and management of hazards. The application of neural networks in systems where their failure can result in loss of life or property must be backed up with techniques to minimize these undesirable effects. Furthermore, to meet the requirements of many statutory bodies such as FAA, such a system must be certified. There is a growing concern in validation of such learning paradigms as continual changes induce uncertainty that limits the applicability of conventional validation techniques to assure a reliable system performance. In this paper, we survey the application of neural networks in high assurance systems that have emerged in various fields, which include flight control, chemical engineering, power plants, automotive control, medical systems, and other systems that require autonomy. More importantly, we provide an overview of assurance issues and challenges with the neural network model based control scheme. Methods and approaches that have been proposed to validate the performance of the neural networks are outlined and discussed after a comparative examination.

1 Introduction

Since the 1980s, artificial neural networks have evolved from biologically inspired layered networks connected by neurons into various categories of computational models with different algorithms and a large variety of architectural designs. They have been widely adopted in many applications as

J. Schumann, Y. Liu (Eds.): Appl. of Neural Networks in High Assur. Sys., SCI 268, pp. 1–19. springerlink.com © Springer-Verlag Berlin Heidelberg 2010

function approximators and classifiers for over 30 years. Typical applications include function approximation, pattern recognition, image recognition, AI applications, and control. Over the past decades, applications of neural networks have emerged in many industrial fields, performing challenging tasks in medical experiments, flight control, automotive industry, robotics, power plants, etc. In many of these applications, neural networks have outperformed traditional computational approaches, with their compelling adaptive capabilities in learning and reacting to changing environments to accommodate novel situations, noise, and even failures. For systems that often operate in an evolving environment and thus requires a high level of self-adaptation, the employment of neural networks for online adaptation to accommodate system faults and recuperate against environmental changes has revolutionized the operation of real-time automation and control applications. However, the majority of neural networks are commonly treated as black boxes and their performance heavily relies on empirical validation. Since many of these industrial applications are in high assurance areas, the lack of in-depth information about the neural network learning coupled with its inherent nonlinearity has posed a great challenge for high assurance applications.

High assurance systems are those that require a warranted high level of robustness in system performance and a guaranteed set of critical properties including reliability, availability, safety, security, as well as other important qualitative attributes. Convincing evidence of adequately meeting the high assurance requirements must be provided when these systems are used in environments where failure can cause substantial financial loss or even loss of life. Hence, without a doubt, high assurance systems are of extreme importance in high profile missions, safety critical operations, and military applications. Examples include avionics, space exploration missions, power plant control, weapon control, life support in medical applications, etc.

Research communities, avionics in particular, have addressed the need for high assurance systems as well as the challenging issues they entail in terms of system verification, validation, and certification. The United States Federal Aviation Administration (FAA) has issued a series of certification processes for quality assurance of software used in aircraft (most notably the DO-178B standard (8)), which heavily rely on software verification and validation methods and theories. Verification and validation (V&V) is considered a crucial path towards certification in order to obtain proven reliability, availability, and safety for high assurance systems. For traditional software systems that are designed, developed, and deployed according to the common software lifecycle and well certified development models, empirical testing is considered an imperative and quite effective means to verify and validate a software. These tests take various forms such as coverage testing, requirements testing, regression testing, stress testing, etc. However, testing of safety-related software has been found to be a main driver for software costs (e.g., (10)).

The adoption of neural networks in an increasing number of high assurance systems has raised growing concern regarding their performance assurance. Most traditional V&V methods are rendered inapplicable and/or ineffective for neural networks. This is simply because, unlike any traditional software, the structure of these novel computational models can evolve over time; and as a consequence, their performance, even validated, would remain subject to unforeseen changes, potentially causing instability and uncertainty in system operations. One of the most critical application areas for neural networks is control. The wide employment of neural networks in control started in 1990's with the advances of adaptive control theory (e.g., (3; 50)) and the rise of dynamic learning models. Many of these control applications are high assurance systems as they are designed/required to tolerate system failures and respond to these failures promptly in order to continue operation and preserve system stability and integrity. For example, as one of the most promising real-time automation and control applications, an adaptive flight control system must be designed to achieve adaptability through judicious online learning, aiding the adaptive controller to recover from operational damage (sensor/actuator failure, changed aircraft dynamics due to broken aileron or stabilator, etc.). Some of these conditions are severe enough to be considered "failure" mode conditions that significantly affect system performance. The National Aeronautics and Space Administration (NASA) conducted a series of experiments evaluating adaptive computational paradigms (neural networks, AI planners) for providing fault tolerance capabilities in flight control systems following sensor and/or actuator failures¹. Experimental success suggests significant potential for further development and deployment of adaptive controllers (24; 5).

In this paper, we survey the application of neural networks in high assurance systems that have emerged in various fields, with a strong focus on control applications which include flight control, chemical engineering, power plants, automobile control, medical systems, and other systems that require autonomy. Section 2 provides a comprehensive list of neural network applications in these domains. Section 3 presents an overview of the assurance issues and challenges with the neural network model based control scheme. Methods and approaches that have been proposed to validate the performance of the neural networks are also outlined and discussed after a comparative examination. Section 4 concludes the paper with a summary and some observations of how future research could improve upon the existing body of applications and studies of neural networks in high assurance systems.

2 Application Domains

In the following, we will have a closer look at several important industrial areas, where neural networks have been applied. This selection is certainly

¹ For the NASA IFCS (Intelligent Flight Control System) project see http://www.nasa.gov/centers/dryden/research/IFCS/index.html and several chapters in this book.

not complete (for other areas, e.g., business see (29)) and it puts some focus on the application of neural networks for control. Other literature surveys include (28; 30), or (9).

2.1 Aircraft Control

One of the most prominent applications of neural networks is its use in damage-adaptive aircraft control. NASA's IFCS project (as discussed earlier) has been able to demonstrate that a manned aircraft can be successfully controlled using a neural network in the presence of (simulated) damage to the aircraft that change its aerodynamic behavior. Details will be presented in several chapters of this book. There are many different approaches to network-based adaptive control for aircraft, e.g., (39) for autonomous helicopters or (7). Different control architectures as well as different kinds of neural networks (e.g., Multilayer Perceptrons (MLP), Sigma Pi, Adaline networks, Radial Basis Functions (RBF), or Dynamic Cell Structures (DCS)) have been used for this task. In some approaches, the neural network will actually try to learn the dynamics model of the damaged aircraft; other approaches rely on control-augmentation, i.e., the neural network is trained to produce an additional control signal to counteract the effects of the damage. Very tight timing requirements set this application of neural networks aside from other applications: the neural network training algorithm must be capable to adjust within a few seconds. Otherwise, the damaged aircraft can get into an unrecoverable unstable state and will crash.

In this book, individual chapters will discuss important aspects of these kind of applications ranging from a more controls theoretic perspective to results and lessons learnt of the practical project with flight tests. Because of the high safety-criticality of aircraft control systems (in particular for manned aircraft), V&V is of particular importance and will be discussed in this book.

2.2 Automotive

With the advent of digital engine and drive-train control, neural networks have been used in this area, for example, to adapt toward different fuel quality or different driving styles. (20) describes an early drive-train control, which uses a neural network. More recently, advanced technology is being used to reduce fuel consumption, e.g., with a neural network based controller for the Toyota Prius (43), or to improve environmental impact by, e.g., networkbased recognition of misfiring in diesel engines (12). Most applications in this area (for an overview see, e.g., (17)) are somewhat safety-critical and are supposed to work reliably in a wide range of situations. Factory recalls, due to software problems can be extremely costly, so a substantial effort is made toward verification and validation. In our book, we have two chapters devoted to neural networks in automotive applications. Other applications are concerned with the effective production of cars, e.g., by using neural networks to optimize the welding process (37).

2.3 Power Systems

The electric power industry is central for each country, as it has to reliably provide electric power to the customers, facing vastly increasing demands, aging infrastructure, and unforeseen natural events (e.g., lightning strikes, down power lines due to icing). Neural networks have been applied to various problems in design and development of power systems ever since the 1990's. Existing bibliographical studies of neural network applications to power systems outline five main application areas for neural networks in system operation and control: load forecasting, fault diagnosis, economic dispatch, security assessment, and transient stability. As a result of this literature review, a classification of publications on neural network applications to power systems between 1992 and 2004 is presented by Bansal (4). This article further discusses trends of adoption of this technology. Another classification of publications can be found in (18) for neural network applications to power systems (covering 2000 to 2005). Both analyses clearly show the growing interest as well as the success in applying neural networks to solve various problems in this domain. It is also noted that although the adoption has been popular, the variety of used neural network architecture remains limited: multi-layer perceptrons, Hopfield networks, and Kohonen neural networks are the three major models that have been employed in power systems.

Neural network applications in power systems have been well studied with a strong focus to improve its prediction accuracy. The economic value of employing neural networks for prediction, load forecasting, and economic dispatching in particular, is reinforced by their successful applications in areas like short-term electric demand forecasting or combustion optimization with reduced NOx emissions. Two major conferences were held in late 90's, the 1998 American Power Conference (1) and the 1999 International Business Forecasting Conference, featuring the findings and practices of neural network applications in these areas. An exemplary application of neural networks to nuclear power systems with an assessment of economic benefits is given by Lisboa (31). Neural networks are applied together with an expert system in a staged approach to retrieve useful information from a gigantic amount of data generated from inspections of the reactor core control assemblies (RCCAs). Approximately 800MBytes of data is produced per core inspection. By extracting out the 5% of data that contains the meaningful information using a neural network based approach, the inspection time is greatly reduced and as a result, the Duke Power Company could save a substantial cost estimated at \$28,000 per inspection and projected to save "\$361k in the next 5 years"².

² http://www.nuc.berkeley.edu/thyd/ne161/rtse/dukerods.html

In addition to forecasting, system control and operations also benefit from neural network applications. In power generation, neural networks can be used to estimate certain process variables to model a dynamic and often nonlinear process (e.g., pulverized fuel flow to the boiler) that otherwise cannot be measured or computed directly. Unlike linear modeling techniques (e.g., Kalman Filters), neural networks can approximate linear functions as well as any nonlinear functions and provide an accuracy at any required level with high cost efficiency. For this reason, more and more neural networks are adopted by power stations in dynamic environments for detecting and diagnosing faults and gaining transient stability. Neural networks are also used in the control and monitoring of steam turbines, as their proper operation strongly influences the overall power plant efficiency. Siemens, for example, is using neural networks for the estimation of the blade temperature (49; 36).

2.4 Medical Systems

It is obvious that many medical software applications are highly safetycritical; failures in the software can cost human lives. A prominent example is the Therac-25, a radiation therapy device, where faulty software caused several (fatal) accidents (38). The noisy and statistical nature of most medical data and measurements seem to be ideally suited for the application of neural networks. The earliest and most widely used neural network based system in health care is Papnet, which has been developed by Neuromedical Systems, Inc. in the 90's. There is a number of studies that this software improves detection rates for cervical cancer from Papnicolau stained smear slides. However, the cost-effectiveness of this application was never satisfactory (31).

An example of a hybrid decision support system in health care is GLADYS (GLAsgow system for the diagnosis of DYSpepsia)³, developed by the Glasgow Southern General Hospital with support from the University of Glasgow's Department of Public Health. GLADYS uses a Bayesian model for the diagnosis of several conditions related to dyspepsia. It uses statistical representations to encode knowledge of clinical staff in a structural form that can be updated numerically and is used to process uncertain knowledge in a consistent manner.

Questar⁴ (54) is a sleep analysis package, developed initially by the Engineering Department at Oxford University and marketed by Oxford Instruments. It was awarded a British Computer Society medal in 1996 and gained FDA approval in 1997. The purpose of this software is to automate sleep staging into awake, rapid eye movement (REM) or light sleep, and deep sleep as accurately as an expert user, but on a continuous scale and with a much faster sampling rate of 1Hz. It does this by combining three electrical

³ http://students/dcs.gla.ac.uk/students/lamkc/CPI.html

⁴ http://www.eng.ox.ac.uk/World/Research/Summary/B-Neural.html

measurements, electro-encephalogram (EEG), electro-oculogram (EOG) and electro-myogram (EMG), which measure mental activity, eye movement, and muscular activity, respectively.

Another health care project developed at Oxford University is a software monitor for intensive care patients. are very high. As a result, a The demand for intensive care beds is very variable and costs for intensive care are extremely high. Thus, the decision to whether or not to admit a critical patient into intensive care can have substantial impact. Such a decision can be supported with the aid of a statistical advisory system, e.g., the commonly used Apache II (Acute Physiology and Chronic Health Evaluation II, Glasgow University) (25),

In the same domain of management and intensive care for critically ill patients, several software packages using rigorous statistical methods and neural networks for knowledge discovery are used in European hospitals. A Bayesian model of clinical data has been used to test the hypothesis that Cerebral Partial Pressure does indicate the presence of sub-clinical damage by trending during the first 24 hours following admission (31). This indicates that careful monitoring of this highly invasive measurement can improve the management of patients, who "talk and die" (35).

The original Apache II monitor processes five standard physiological measurements (EEG, systolic blood pressure and oxygen saturation, breathing rate, and temperature) and produces alarms based on novelty of the data. Thus, it is an example of a data based system where the available signals define a nominal state, which is not of interest. The challenge is to accurately and robustly determine deviations in this multivariate data stream. The problem becomes harder due to the low density in the input space and the necessity to accommodate different states of the patient during recovery and robustness against artifacts (e.g., sensor displacement).

A web-based advisory system using neural networks has been developed for the automated interpretation of myocardial perfusion images⁵. Another system, also developed at Lund University, is used for acute myocardial infarction (AMI) detection. It was tested on a data base of approximately 1,000 electro cardiograms (ECGs) from patients with AMI and approximately 10,000 control ECGs. In this application, the neural network system was found to be more sensitive and has a higher discrimination accuracy than benchmark ECG software, or expert cardiologists.

2.5 Other Applications

The Sharp LogiCook (54) was the first microwave oven that used neural network technology. It was originally developed at Oxford University⁶. Based

⁵ http://www.weaidu.com/software/index.html

⁶ http://www.eng.ox.ac.uk/World/Research/Summary/B-Neural.html or http://www.scit.wlv.ac.uk/~cm1822/acn17.htm

upon user input (food or liquids), the optimum cooking time was obtained from an analysis of the proportional, integral and derivative humidity profiles using a neural network. The software is also capable of dealing with differentsized portions and can detect dangerous conditions.

An industrial area where neural network control has been successfully applied for a long time is a steel rolling mill. Here, accurate control of temperature of the strip and the rolling force are critical for the quality of the product. Based upon a prototype, developed for Hoesch (Dortmund, Germany), Siemens has deployed this technology world wide since then. (45) claims efficiency gains of 30% due to better accuracy in rolling force modeling with prediction improvements leading to savings of \$200K in material costs annually. In this application, the neural network's capability to handle non-linear data has been beneficial.

In the Airline business operation area, BehavHeuristics, Inc. (started in 1986 and later part of Airline Automation Inc.) uses reinforcement learning to predict no-shows in air-flights, thus maximizing the passenger load through controlled overbooking. Their Airline Marketing Tactician (AMT) (22) was an early success for neural networks.

The Boeing Company's NIRS (Neural Information Retrieval System) (48; 23), is probably still the largest scale manufacturing application of neural networks. It uses a binary Adaptive Resonance Theory network (ART1), to cluster binary templates of aircraft parts. The systems arranges them in a complex hierarchical network covering over 100,000 items. These are then grouped into thousands of self-organized clusters. Substantial savings in manufacturing costs (several \$M per year) have been reported.

3 Toward V&V of NNs in High Assurance Systems

3.1 V&V of Software Systems

Any subsystem and component of a high assurance application must undergo a rigorous process in order to make sure that all requirements regarding safety, performance, and reliability are met. This refers to any hardware component as well as to software. Since most neural network based applications are ultimately implemented as a software program, we will focus on software components only.

In any software development lifecycle, there are, in addition to activities for the design and implementation of the software, activities to ensure that the final software is working as expected. Traditionally, we distinguish between verification and validation: Verification is often informally described as "Are you building the right thing?", whereas validation can be paraphrased as "Are you building the thing right?". It is also obvious that different V&V activities are performed at different stages of the software lifecycle. Figure 1 shows a simplified version of the software development stages and related



Fig. 1 Verification and validation activities during software development ("V-shape"). Verification activities are marked by dashed lines, validation by dotted arrows

V&V activities (see e.g., (42)). Several observations can be made, which will be helpful to tackle V&V issues for network based systems:

- Verification tasks are performed on the left side of the "V" and thus are mainly performed during earlier stages of the development process, whereas validation tasks (mainly testing) relate the finished products (on the right side of the "V") with the corresponding artifact on the left side. Ultimately, in the system qualification (or acceptance) testing, the entire system is to be tested against the requirement specifications.
- It is well known that the removal of faults can be orders of magnitude more expensive in later stages than in an early development phase (42). In particular for safety-critical applications, costs for V&V are the main cost drivers for software (e.g., (10)).
- Verification activities can be loosely grouped into design-time V&V and code V&V. In particular, when complex algorithms like neural network learning algorithms or multivariate optimization algorithms are used, this distinction is important and we will discuss it in detail below.
- Traditional V&V ends when the software is deployed. However, if the software is to work in unknown or changing environments, or has to react toward unforeseen events, additional activities are necessary in order to ensure that the software is working correctly. Such techniques range from simple exception handling and dynamic performance monitoring to runtime verification and certification (e.g., (46). Techniques for recovery from failures include reconfiguration or code repair (e.g., via self healing code (13)).

Virtually all software for high assurance applications is developed according to a specific software process. Usually such processes are highly standardized (e.g., according to IEEE or ANSI). (2) gives an overview of several traditional ANSI standards. Usually, a Software V&V process is an integral component of a software development process. A V&V process describes which tests are to be carried out, which activities for verification are to be performed, and how the tests and their results are documented.

In many safety-critical application areas, all systems (and also the software) have to go through a certification process. This often highly standardized process has the goal to demonstrate to a certification board that all required steps have been carried out and that due diligence has been applied to make sure that the system under consideration adheres to all safety and performance requirements.

Probably the best-known certification standard is DO-178B (8), which is the standard prescribed by the FAA for all safety-critical software to be used in civil transport aircraft in the US. It is a very detail-oriented and resourceconsuming process, so certification is a major cost driver for safety-critical software.

As discussed earlier, existing standards, however, cannot be used as is for the V&V of neural networks. In the following, we will discuss a number of V&V issues, which prevent the use of current certification standards for neural network applications.

3.2 V&V Issues and Gaps for NN-Based Applications

Different scientific and engineering communities use different notations and nomenclature. This can lead to substantial misunderstandings like the following examples:

"non-deterministic". In computer science (CS), the notion of non-deterministic piece of code is always attached to a program "execution with one or more choice points where multiple different continuations are possible without any specification of which one will be taken"⁷. Practical implementations of non-determinism thus usually use random number generators. In general, a specific state (or computation sequence) cannot be reproduced, making testing of such software extremely difficult.

In engineering disciplines, a system is usually coined non-deterministic if it is non-Markovian, i.e., that the system state \mathbf{x}_t cannot be totally described by \mathbf{x}_{t-1} . Rather, the entire history (e.g., the entire flight since take-off) needs to be taken into account, i.e., \mathbf{x}_t can only be calculated given $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$. With all the history present, the state \mathbf{x}_t can be exactly reproduced. Thus, this notion is not based upon any random number generators.

Virtually all adaptive control systems are non-Markovian, but deterministic (in the CS sense). Since some forms of neural network algorithms start with randomly initialized weights (e.g., standard multi-layer

⁷ http://en.wikipedia.org/wiki/Non-deterministic_algorithm

perceptrons), the opinion that neural networks (and thus neural-networks based adaptive controllers) are nondeterministic (in the CS sense) persisted.

"Neural Network". Many architectures for adaptive control systems have been developed using neural networks (e.g., (40; 47; 7)). The research area of neural networks, in general, has traditionally been put into the vicinity of artificial intelligence. Hence, the notion of a neural network is often attached to terms like "AI", "brain-like", "bio-inspired" possibly leading to confusion and low confidence, when considered within a safety-critical environment.

Technically speaking, the neural networks in the adaptive controllers have been purely used as multivariate non-linear function approximators; the "learning" is (in most cases) a recursive least-squares optimization algorithm. Described in these terms, a lot of "hype" about potential and "dangers" of neural networks can be avoided.

An adaptive control system or other NN based software in a high assurance application is handled like any other highly safety-critical piece of software: it must undergo rigorous V&V and the software must be certified. However, most traditional techniques for V&V as prescribed in these standards cannot be used on an online neuro-adaptive system because this system

- has to deal with a dynamically changing, unknown, non-linear plant model. Typically, damages to an aircraft (e.g., a stuck rudder) introduces biases, non-linearities and unknown interactions (e.g., correlations between the different aircraft axes). Moreover, most aerospace analysis techniques are restricted to the linear case.
- is a system, which contains non-linear functions and approximators. Except for the most primitive kinds of neural networks, neural networks use nonlinear activations functions and can, in principle, approximate any smooth function.
- is adapted using a complicated algorithm. In most cases, the neural network is being trained using some kind of machine learning algorithm. Such algorithms usually are variants of a recursive multivariate (quadratic) numerical optimization routine.

3.3 V&V Approaches for Neural Networks

In the following, we will discuss V&V approaches for neural networks and systems, containing neural networks, in particular neuro-adaptive controllers. This area can be roughly subdivided into the following categories, concerning techniques that

• specifically subdivide V&V activities into algorithm V&V and code V&V. In particular, theoretical results, obtained during algorithm V&V (e.g., Lyapunov stability proofs) must be used to guide and augment code V&V.

- focus on the analysis of the neural network architectural design (e.g., number of hidden layers, number of hidden nodes),
- consider neural networks as function approximators or data classification tools,
- help the human reader to understand the inner workings of the neural network (e.g., by rule-extraction or representation as a Fuzzy System),
- focus on the specifics and characteristics of the learning (training) algorithm,
- analyze the selection and quality of the data used for adaptation, and
- provide means for the dynamic (i.e., during operation) monitoring of the performance of the adaptive component.

Obviously, the techniques and approaches in each of these categories heavily overlap and have synergistic effects. Moreover, the various techniques range from mathematical theorem and proof (e.g., universal function approximation of a MLP), statistical methods, methods from design of experiments, testing, simulation, and dynamic analysis and monitoring of the behavior and performance of the neural network. The term "dynamic" here indicates that the monitoring occurs during the actual operation of the neural network based system after deployment.

In many cases, certain performance and safety aspects of the neural network are necessary in order to analyze the larger system. A typical example is a neural-networks based adaptive controller. In order to show (eventual) stability of the controller using Lyapunov stability theory, assumptions about the neural network (e.g., on bounds of the error) are required. Such proofs can be pretty involved. In several chapters of this book such stability proofs are discussed. Other examples can be found, for example, in (47).

3.3.1 NN as Function Approximator

Traditional literature describes adaptive computational paradigms, neural networks in particular, with respect to their use, as function approximators or data classification tools. Validation on these systems is usually based on a train-test-re-train empirical procedure. Some bibliographic references also propose methods as part of the training algorithm of neural networks for validation (55; 6). The ability of interpolating and/or extrapolating between known function values is measured by certain parameters through testing. This evaluation paradigm can be reasonably effective only for pre-trained adaptive systems, which does not require online learning and adaptation and remain unchanged in use.

3.3.2 V&V for NN Design

In (11), Fu interprets the verification of a neural network to refer to its correctness and interprets the validation to refer to its accuracy and efficiency. He establishes correctness by analyzing the process of designing the neural network, rather than the functional properties of the final product. Peterson presents another similar approach in (41) by discussing the software development process of a neural network. He describes the opportunities for verification and validation of neural networks in terms of the activities in their development life cycle, as shown in Figure 2.



Fig. 2 The development cycle of a neural network

As we can see from Figure 2, there is a focus on V&V of adaptive systems based on the training data. Verification of the training data includes the analysis of appropriateness and comprehensiveness. However, in online learning mode, this technique may not be appropriate due to its real-time training aspects. Data are collected in such a way that the training is completed under intensive computational requirements. An applicable approach for verifying the data is novelty detection.

3.3.3 V&V for NN Training

Verification of the training process typically examines the convergence properties of the learning algorithm, which is usually pre-defined by some criteria of error measure. In (21), K.J. Hunt et. al. investigate all different methods for error estimation techniques and make detailed comparison among them. Nonetheless, effective evaluation methods of interpolation and extrapolation capabilities of the network and domain specific verification activities are still based on empirical testing (26). Literature addressing the problem analytically is very scarce. In the field of function approximation theory, MLP networks have been proven to be universal approximators for being able to achieve any given accuracy provided a sufficient number of hidden neurons (19). The mathematical analysis and proof can be seen as another effort for validating the learning process as it can provide theoretical proof for the capabilities of function approximation. The weakness of such analytical proof is that it remains impractical for online adaptive learning systems as the system function evolves.

Most recently proposed techniques on V&V of neural networks are based on empirical evaluation through simulation and/or experimental testing using statistical methods such as K-fold cross-validation, bootstrapping, repeated random sampling, etc. There also exists some approaches to V&V of dynamic neural networks by modifying the training algorithms. In an attempt to solve the dilemma of plasticity and stability for neural networks, S. Grossberg (14; 15) derives a new paradigm, referred to as the Adaptive Resonance Theory (ART-1/2/3). Within such a network, there are two components charging seen and unseen data respectively. The Validity Index network presented by Leonard et. al. in (27) is an example of modification to the network training algorithm for V&V of the neural networks. When tested, the validity index in a Radial Basis Function neural network provides a confidence interval associated with each network prediction for a given input.

3.3.4 Dynamic Monitoring

For online neural networks that are adopted in adaptive control applications, static V&V methods for neural network design and training fall short to warrant online performance assurance due to the dynamic nature of the network. Because not all conceivable situations can be validated upfront, it is almost impossible to guarantee the assurance of reliable performance and safety. In order to validate the online adaptation performance, dynamic monitoring tools can be used that work during the actual execution of the software. Such tools that can dynamically monitor the quality of the neural network and its internal parameters have been proposed, mainly focusing on the learning performance and prediction performance. A few major approaches are listed below.

• Online Learning Performance Analysis.

Lyapunov stability theory based monitors are proposed for the Dynamic Cell Structure (DCS) Networks in Yerramalla et. al. (56; 57) and Chapter 6 of this book. The proposed online monitoring system is composed of several dynamic stability monitors. Each monitor is essentially a Lyapunov-like function that is designed to analyze and capture unstable behavior from a particular aspect of online learning. These monitors provide an observation of how well a set of associated neural centers of the online neural network are being overlaid over corresponding relative elements of the presented training data set.

Schumann and Liu (52; 53) propose another estimate for online learning performance by calculating the *parameter sensitivity* in the context of flight controllers. In an online adaptive system, the internal control parameters are changing while the system is in operation. The sensitivity of a parameter with respect to changes is computed as the probability $p(\mathbf{o}|\mathcal{P}, \mathbf{x})$ for network output \mathbf{o} , given parameters \mathcal{P} , and inputs \mathbf{x} . Assuming a Gaussian probability distribution, the parameter sensitivity can be obtained as the variance $\sigma_{\mathcal{P}}^2$. Such measures can provide useful information to improve neural network design and learning paradigms.

• Network Prediction Confidence Estimation.

The Validity Index tool proposed by Liu et. al. (32; 34) calculates reliability measures for the output of a DCS network, which has been used for the Gen-I IFCS controller. Following the definition of Validity Index (VI) in RBF networks by Leonard et. al.(27), the validity index in DCS networks is defined as an estimated confidence measure of a DCS output, given the current input. The VI can be used to measure the accuracy of the DCS network fitting and thus provides information for future validation activities. By examining the statistical properties of the best matching neuron and its neighbors that are activated during learning and prediction, the validity index tool takes into account the topology-based learning structures and produces a quality metric for the output. Details can be found in (33).

Schumann et. al. (51) developed the Confidence Tool using a Bayesian statistical approach to estimate the quality of learning and the accuracy of estimation. Considering all inputs and outputs of the neural network as statistical variables with a given probability density function (e.g., Gaussians), the algorithm determines the variance σ^2 as a quality metric on the output. This tool has been developed for the IFCS adaptive flight controller and has been test-flown on a manned F-15 NASA aircraft.

4 Conclusions

This chapter serves as our attempt to provide an overview of applications of neural networks, where failure is not an option. Such high-assurance applications can be found in many domains, most prominently in aerospace, automotive industry, medical applications, and power industry. Here, the use of neural networks provides substantial benefits with respect to performance, accuracy, and/or handling of unforeseen situations. However, the algorithms implementing neural networks can be very complex, in particular the training algorithms, which adjust the neural network's parameters based upon given data. These training algorithms, most often variants of non-linear multivariate quadratic optimization algorithms, are at the core of many neural network based applications. Because these applications, which have been discussed above, are safety-critical i.e., failures or erroneous behavior can ultimately claim human lives, the software that implements the algorithmic learning of neural networks has to undergo rigorous verification and validation (V&V) before deployment. In this chapter we discussed that traditional V&V for safety-critical code is not sufficient for neural-network based applications, in particular for those applications where the neural network is trained in an online fashion during operation.

The presented overview also serves the purpose of structuring the field, as well as illustrating the widespread application potential of neural networks in safety-critical applications and the issues in terms of their V&V that had to be addressed by the research community.

The remaining chapters of this book are ordered in a similar way. Chapters 2 and 3 discuss theoretical and design-time analysis on semi-global boundedness and margins of adaptive control and assessment of network complexity.

The next several chapters are devoted to applications of neural networks for adaptive aircraft control. One chapter focuses on the development of a damage-adaptive flight controller all the way through manned flight tests. The other one presents approaches for stability and convergence of adaptive flight control, a central challenge for V&V of such controllers. Some neural network architectures change their architecture and size, while they are being trained. Using such networks in an adaptive controller poses substantial challenges for V&V. This chapter deals with dynamic allocation in such neural network architectures.

The next chapter is centered around the automobile: it describes the use of an immune-system approach to help with the localization of faults in automotive engines. Moving yet to another element, the subsequent chapter discusses the design of a neuro-adaptive controller for a submarine.

In most mechanical systems, friction between moving components is a major issue. Our chapter describes how neural networks can be used to provide accurate friction control. Due to the transition between slipping and sticking this problem is highly nonlinear. The final two chapters discuss how neural networks can improve the efficiency of processes (blending of crude oil) and fuel cells.

We hope that the wide range of applications and methods described in the book illustrate the potential of neural networks in safety-critical and highassurance applications and help the reader to be more aware of issues and approaches and to drive the advances of V&V of such systems to ultimately make them safe and reliable.

References

- 1. American Power Conference. Proceedings of the American Power Conference 1998, vol. 60, 2 vol. set (1998)
- 2. American National Standards. Software Engineering Standards. IEEE, Wiley and Sons (1984)
- Astrom, K.J., Wittenmark, B.: Adaptive Control. Addison-Wesley, Reading (1994)

- Bansal, R.C.: Overview and literature survey of artificial neural networks applications to power systems (1992–2004). Journal - Institution of Engineers India. Electrical Engineering Division (86), 282–296 (2006)
- 5. The Boeing Company. Intelligent Flight Control: Advanced Concept Program. Technical Report (1999)
- Boyd, M.A., Schumann, J., Brat, G., Giannakopoulou, D., Cukic, B., Mili, A.: Validation and verification process guide for software and neural nets. Technical report, NASA Ames Research Center (2001)
- Calise, A., Rysdyk, R.: Nonlinear adaptive flight control using neural networks. IEEE Control Systems Magazine 21(6), 14–26 (1998)
- 8. DO-178B: Software Considerations in Airborne Systems and Equipment Certification (1992), http://www.rtca.org
- 9. DTI. DTI final report: Evaluation of parallel processing and neural computing application programmes. Assessment Paper 38 URN 99/922 (1999)
- Ebert, C., Parro, C.H., Suttels, R., Kolarczyk, H.: Improving Validation Activities in a Global Software Development. In: Proc. ICSE 2001, p. 545. IEEE, Los Alamitos (2001)
- Fu, L.: Neural Networks in Computer Intelligence. McGraw Hill, New York (1994)
- Getman, A., Zhou, H., Jammalamadaka, S.: An automated network for detecting diesel engine misfire. In: Proc. IJCNN 2007, pp. 3017–3021. IEEE, Los Alamitos (2007)
- Ghosh, D., Sharman, R., Rao, R.H., Upadhyaya, S.: Self-healing systems survey and synthesis. Decision Support Systems 42(4), 2164–2185 (2007)
- Grossberg, S.: Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. Biological Cybernetics 23, 121–134 (1976); Reprinted in Anderson and Rosenfeld (1988)
- 15. Grossberg, S.: Competitive learning: From interactive activation to adaptive resonance. Cognitive Science 11(1), 23–63 (1987)
- Gupta, P., Guenther, K., Hodgkinson, J., Jacklin, S., Richard, M., Schumann, J., Soares, F.: Performance Monitoring and Assessment of Neuro-Adaptive Controllers for Aerospace Applications Using a Bayesian Approach. In: Guidance, Navigation and Control (GNC) Conference, AIAA 2005-6451 (2005)
- Gusikhin, O., Rychtyckyj, N., Filev, D.: Intelligent systems in the automotive industry: Applications and trends. Knowl. Inf. Syst. 12(2), 147–168 (2007)
- Tarafdar Haque, M., Kashtiban, A.M.: Application of neural networks in power systems: A Review. Transaction on Engineering, Computing and Technology (6), 53–56 (2005)
- Hornik, K.M., Stinchcombe, M., White, H.: Multilayer Feedforward Networks are Universal Approximators. Neural Networks 2, 359–366 (1989)
- Hrycej, T.: Neural-network-based car drive train control. In: Proc. 42nd IEEE Vehicular Technology Conference, vol. 2, pp. 1042–1045 (1992)
- Hunt, K.J., Sbabaro, D., Zbikowski, R., Gawthrop, P.J.: Neural Networks for Control Systems—A Survey. Automatica 28(6), 1707–1712 (1996)
- Hutchison, W.R., Stephens, K.R.: The airline marketing tactician (AMT): A commercial application of adaptive networking. In: Proc. First IEEE International Conference on Neural Networks, vol. 2, pp. 753–756. IEEE, Los Alamitos (1987)

- Irwin, D.: The industrial Electronics Handbook. Technology & Engineering (1997)
- 24. Jorgensen, C.: Feedback linearized aircraft control using dynamic cell structures. In: World Automation Congress (ISSCI), Alaska, pp. 050.1-050.6 (1991)
- Knaus, W.A., Draper, E.A., Wagner, D.P., Zimmerman, J.E.: Apache II: a Severity of Disease Classification System. Critical Care Medicine 13, 813–829 (1985)
- Lawrence, S., Tsoi, A.C., Back, A.D.: Function approximation with neural networks and local methods: Bias, variance and smoothness. In: Australian Conference on Neural Networks, pp. 16–21 (1996)
- Leonard, J.A., Kramer, M.A., Ungar, L.H.: Using radial basis functions to approximate a function and its error bounds. IEEE Transactions on Neural Networks 3(4), 624–627 (1992)
- Lisboa, P. (ed.): Current Applications of Neural Networks. Chapman and Hall, Boca Raton (1992)
- Lisboa, P., Vellido, A., Edisbury, B. (eds.): Neural Network Applications in Business. World Scientific, Singapore (2000)
- Lisboa, P., Ifeachfor, E., Szczepaniak, P. (eds.): Artifical Neural Networks in Biomedicine. Springer, Heidelberg (2000)
- Lisboa, P.: Industrial use of safety-related artificial neural networks. Contract Research Report 327/2001, Liverpool John Moores University, Liverpool (2001)
- 32. Liu, Y., Yerramalla, S., Fuller, E., Cukic, B., Gururajan, S.: Adaptive Control Software: Can we guarantee safety? In: Proc. of the 28th International Computer Software and Applications Conference; Workshop on Software Cybernetics (2004)
- Liu, Y., Cukic, B., Jiang, M., Xu, Z.: Predicting with Confidence—An Improved Dynamic Cell Structure. In: Advances in Neural Computation, vol. 1, pp. 750– 759. Springer, Heidelberg (2005)
- Liu, Y.: Validating A Neural Network-based Online Adaptive System. PhD thesis, West Virginia University, Morgantown (2005)
- Marshall, L.F., Tool, B.M., Bowers, S.A.: The National Traumatic Coma Data Bank Part 2: Patients who talk and deteriorate: Implications for treatment. Journal of Neurosurgery 59(2) (1983)
- Mathur, A., et al.: Turbine Back Pressure Identification and Optimization with Learning Neural Networks. Advances in Instrumentation and Control (45): 1, 229–236 (1990)
- Ramírez Mendoza, R.A., Morales-Menéndez, R., Cantú-Ortiz, F.J.: Neural nets modelling for automotive welding process. In: Hamza, M.H. (ed.) Modelling and Simulation. Acta Press (2005)
- 38. Neumann, P.G.: Computer Related Risks. ACM Press, New York (1995)
- Ng, A., Jin Kim, H.: Stable adaptive control with online learning. In: Proceedings of NIPS, vol. 17 (2005)
- Norgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modeling and Control of Dynamic Systems. Springer, Heidelberg (2002)
- 41. Peterson, G.E.: A foundation for neural network verification and validation. In: SPIE Science of Artificial Neural Networks II, vol. 1966, pp. 196–207 (1993)
- Pressman, R.: Software Engineering: A Practitioner's Approach. McGraw-Hill, New York (1999)

- Prokhorov, D.: Toyota Prius HEV Neurocontrol and Diagnostics. Neural Networks 21(2-3), 458–465 (2008)
- Pullum, L., Taylor, B., Darrah, M.: Guidance for the Verification and Validation of Neural Networks. Emerging Technologies. Wiley-IEEE Computer Society Press (2007)
- Ruha, P., Peura, P., Oyj, R., Jansen, M.: Using Neural Networks in Steel Strip Rolling. In: Steels and Materials for Power Plants, pp. 51–56. Wiley, Chichester (2006)
- Rushby, J.: Runtime Certification. In: Leucker, M. (ed.) RV 2008. LNCS, vol. 5289, pp. 21–35. Springer, Heidelberg (2008)
- Rysdyk, R., Calise, A.: Fault tolerant flight control via adaptive neural network augmentation. AIAA American Institute of Aeronautics and Astronautics, AIAA-98-4483, pp. 1722–1728 (1998)
- Smith, S.D.G., Escobedo, R., Anderson, M., Caudell, T.P.: A deployed engineering design retrieval system using neural networks. IEEE Trans. on Neural Networks 8(4), 847–851 (1997)
- Santoso, N., Petsche, T.: A Method for Blade Temperatrue Estimation in a Steam Turbine. European Patent, No EP0944866 (2001)
- Sastry, A., Bodson, M.: Adaptive Control: Stability, Convergence and Robustness. Prentice Hall, Englewood Cliffs (1994), http://www.ece.utah.edu/~bodson/acscr
- Schumann, J., Gupta, P.: Monitoring the performance of a neuro-adaptive controller. In: Proc. of the 24th Intl. Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (2004)
- Schumann, J., Liu, Y.: Performance Estimation of a Neural Network-based Controller. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 981–990. Springer, Heidelberg (2006)
- Schumann, J., Liu, Y.: Tools and Methods for the Verification and Validation of Adaptive Aircraft Control Systems. In: IEEE Aerospace Conference. IEEE Press, Los Alamitos (2007)
- Tarassenko, L.: LogiCook and QUESTAR: two case studies in successful technology transfer. IEE Digest 1997(14) (1997)
- Tibshirani, R.: Bias, Variance and Prediction Error for Classification Rule. Technical Report, Statistics Department, University of Toronto (1996)
- Yerramalla, S., Cukic, B., Fuller, E.: Lyapunov Stability Analysis of Quantization Error for DCS Neural Networks. In: Proc. Intl. Joint Conference on Neural Networks, IJCNN 2003 (2003)
- 57. Yerramalla, S., Fuller, E., Cukic, B.: Lyapunov Analysis of neural network stability in an adaptive flight control system. In: Huang, S.-T., Herman, T. (eds.) SSS 2003. LNCS, vol. 2704, pp. 77–91. Springer, Heidelberg (2003)