Exploiting Safety Constraints in Fuzzy Self-organising Maps for Safety Critical Applications

Zeshan Kurd[1], Tim P. Kelly[1], and Jim Austin[2]

[1] High Integrity Systems Engineering Group
[2] Advanced Computer Architectures Group
Department of Computer Science
University of York, York, YO10 5DD, UK
{zeshan.kurd,tim.kelly,austin}@cs.york.ac.uk

Abstract. This paper defines a constrained Artificial Neural Network (ANN) that can be employed for highly-dependable roles in safety critical applications. The derived model is based upon the Fuzzy Self-Organising Map (FSOM) and enables behaviour to be described qualitatively and quantitatively. By harnessing these desirable features, behaviour is bounded through incorporation of safety constraints – derived from safety requirements and hazard analysis. The constrained FSOM has been termed a 'Safety Critical Artificial Neural Network' (SCANN) and preserves valuable performance characteristics for non-linear function approximation problems. The SCANN enables construction of compelling (product-based) safety arguments for mitigation and control of identified failure modes. Illustrations of potential benefits for real-world applications are also presented.

1 Introduction

Artificial Neural Networks (ANNs) are employed in a wide range of applications such as defence, medical and industrial process control domains [1]. There are a plethora of appealing features associated with ANNs such as adapting to a changing environment and generalisation given novel data. They are efficient tools for finding swift solutions using little input from designers. However, there exist several problems associated with ANNs that commonly restrict operation to advisory roles in safety related applications. Recent work [2] has examined verification and validation of ANNs for critical systems. This work aims to provide guaranteed output (within bounds) for ANN models whose behaviour is represented neither in structured nor organised forms. As a result, the approach treats the ANN as a black-box and uses pedagogical approaches to analyse and control behaviour (using error bounds [2]). This analytical approach is common to the main thrust of existing work for developing ANNs for safety critical contexts as reviewed in [3]. Limitations experienced from black-box analysis clearly highlight the need for improved neural models to allow compelling safety and performance arguments required for certification.

Within the scope of this paper, section 2 defines a potentially suitable ANN model with learning algorithms. Section 3 presents an overview of how key failure modes are tackled by means of safety constraints. Section 4 describes the benefits of the approach using an abstract control system example.

Z.R. Yang et al. (Eds.): IDEAL 2004, LNCS 3177, pp. 266–271, 2004.

2 Fuzzy Self-organising Maps

Our previous work [4] has identified 'hybrid' neural networks as potential models for allowing white-box style (decompositional) analysis. 'Hybrid' ANNs facilitate potential arguments about specific functional properties. This section describes an existing ANN model known as the Fuzzy Self-Organising Map (FSOM) [5]. The FSOM is a 'neuro-fuzzy' system and is based upon Kohonen's Self-Organising Map. It is endowed with the ability to describe its behaviour using Takagi-Sugeno (TS) fuzzy rules. TS fuzzy rules encapsulate both qualitative and quantitative descriptions of the functional behaviour where rule outputs are linear functions. The FSOM has been used for pattern recognition problems [5] and non-linear function approximation [6]) with fruitful results. The FSOM architecture consists of six stages and is illustrated in figure 1.

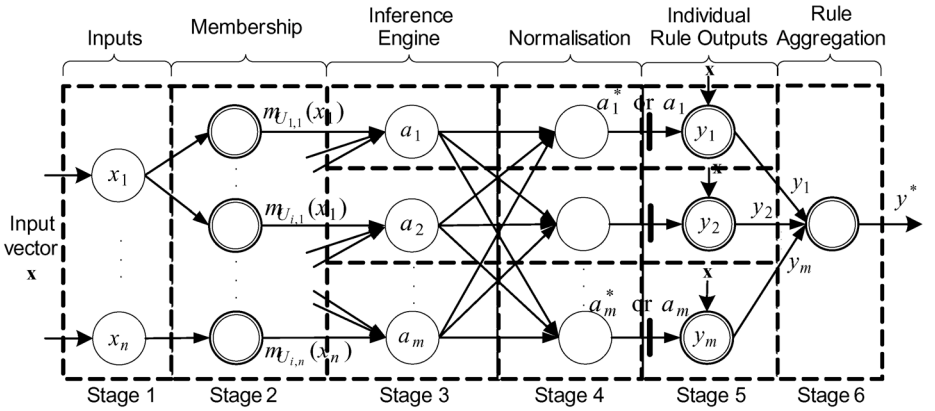

Fig. 1. Fuzzy Self Organising Map with constrained neurons depicted by double circles.

Stage 1 involves no pre-processing and simply propagates inputs x_j where $j = 1,2,...,n,$ (n is the total number of input variables) to stage 2. The number of neurons equals the number of input variables (or sensors).

Stage 2 performs the set membership function and contains a neuron for every fuzzy set. The membership is defined by the triangular function [6]. Each fuzzy set is defined as $U_{i,j}$ for the i^{th} rule where $i = 1,2,...,m,$ (m is the total number of rules). The adaptable parameters for each fuzzy set are centre, left and right edges of the spread (or support) as defined by (1).

$$\{c_{i,j}, sl_{i,j}, sr_{i,j}\}. \tag{1}$$

Stage 3 performs fuzzy inference, where the firing strength α_i for each rule is determined by the minimum operator. The firing strength is greater than zero only if all rule antecedents are true. **Stage 4** normalises firing strengths and **Stage 5** determines crisp rule outputs for the i^{th} rule using (2).

$$y_i = f_i(x_1,...,x_n) = a_{i,0} + a_{i,1}x_1 + a_{i,2}x_2 +,...,+a_{i,n}x_n. \tag{2}$$

There are $n + 1$ adaptable output parameters per rule defined by (3).

$$\{ a_{i,0}, a_{i,1}, \ldots, a_{i,n} \}. \tag{3}$$

Stage 6 consists of a single neuron which performs weighted averaging (using rule firing strengths α_i). This determines the final output from multiple firing rules.

The FSOM employs a static learning algorithm for tuning parameters (1) and (3) using training samples or input data. It is performed in two stages as defined in [6]:

- **Phase 1:** Antecedent parameters are frozen and the consequent parameters (3) of the rules are adapted (supervised) using the Least Mean Square algorithm.
- **Phase 2:** Consequent parameters are frozen and the antecedent parameters (1) are tuned using the modified LVQ algorithm [6] (supervised or unsupervised mode).

The FSOM can also self-generate (dynamic learning) as described by Vuorimaa [5]. This automatically acquires novel features described by training data whilst adapting the neural architecture without user intervention. Dynamic learning is an integral feature of the safety lifecycle [4] and further details can be found in [6].

3 Safety Arguments

In previous work [7], the safety criteria were established as a number of high-level goals with a safety argument expressed in GSN (Goal Structuring Notation). These goals were based upon encapsulating different failure modes associated with the generalisation and learning abilities of ANNs. A set of failure modes have been identified for the FSOM that have been derived from HAZOP (Hazard Operability Study) [8] guide words (which originates from the process-control theory domain). The principal failure modes are summarised below:

1. Output is too high or too low for the given input vector.
2. Missing output given valid inputs (output omission).
3. Output increases when it should decrease (and vice versa).
4. Output rate of change is too high or too low (derivatives).

Enabling the FSOM to be used in safety critical systems requires integrating mechanisms to prevent systematic faults from being incorporated during learning. This is achieved through inflexible bounds for flexibility (in behaviour) using constraints on both the generalisation and learning processes. As a result, the modified (or constrained) FSOM is called the SCANN and is used for non-linear function approximation [6]. For ease of explanation, several major arguments will be outlined for SISO (Single-Input-Single-Output) fuzzy rules.

3.1 Safety Argument for Function Mappings

There are several interpretations of Fuzzy Logic Systems (FLS) that do not lend well to critical domains. These include ‘likelihood’ and ‘random’ views [9] of set membership which involve probabilistic reasoning. For safety critical domains, the satisfaction of fuzzy rule pre-conditions can lead to safety concerns. For rule firing there must be certainty in set membership for an input (if the rule post-condition is to be considered ‘safe’). Our interest is in interpretations such as ‘measurement’ and ‘similarity’

views [9]. These interpret degree of membership as relative to other members within the set. The first failure mode to consider is if the output is too high or too low for the current input. Existing approaches neglect the internal behaviour by simply using output error bounds [2]. This is extremely limiting as these 'monitor' technologies result in few or no arguments about the implicit underlying functional properties. During learning, behaviour may digress from the desired function using flawed training samples. Remedial actions for this failure mode include incorporating bounds for each fuzzy rule antecedent and consequent.

Bounds are placed upon (1) to provide assurance that the input fuzzy set always lies within the interval $[\min sl_{i,j}, c_{i,j}]$ for $sl_{i,j}$ and $[c_{i,j}, \max sr_{i,j}]$ for $sr_{i,j}$. The constants $[\min sl_{i,j}, \max sr_{i,j}]$ are the extremes the fuzzy set support can expand to. Moreover, the centre of the input set is constrained to lie within the input set as defined by the interval $[sl_{i,j}, sr_{i,j}]$. This prevents the centre going beyond the spread edges leading to false satisfaction of rules pre-conditions. The rule post-condition (consequent) is also bounded to $[\min y_i, \max y_i]$ although there is no adaptation of the output set (illustrated in figure 2(a)). Attempts to violate input bounds are rejected or used again when the learning rate is smaller. One potential fault is that the rule may output a value that is beyond the output bounds. To avoid over-constraining learning, this problem can be solved by bounding the rule output as described by (4):

$$y_i = \begin{cases} \min y_i, & \text{if } f_i(x_1,\ldots,x_n) < \min y_i, \\ \max y_i, & \text{if } f_i(x_1,\ldots,x_n) > \max y_i. \end{cases} \tag{4}$$

All bounds (constraints) placed upon the semantic interpretations of fuzzy sets are determined from safety analysis [4]. This contributes to providing safe post-conditions (output) for all pre-conditions (inputs) during generalisation and learning.

3.2 Safety Argument for Input Space Coverage

Failure mode 2 is related to faults associated with an incomplete knowledge base or faulty input set tuning. The input space that must be covered (at all times) is defined prior to certification. This is provided through analytical processes during hazard analysis [4]. Once the required input space is defined, the safety argument can be described as forming two main branches. The strategy is to first argue that the rule base completely covers the defined input space during generalisation. Assurance for coverage is provided through Preliminary System Safety Assessment (PSSA) [4]. This evaluates the input space coverage by examining rule input sets to identify "holes". Even if the input space is covered, there may still be omission of output, since the output function may partially cover the input set. The solution to this problem is provided by the rule output bounds defined by (4).

The second branch of the safety argument is concerned with input space coverage during static learning. This argument relies upon the property that no "hole" should be created between input sets of overlapping rules. The solution is to prevent spread updating which may result in exposure of the input space (which can occur in phase 2 of the static learning algorithm). This argument contributes to providing assurance about the functional input-output mappings during generalisation and learning phases.

3.3 Safety Argument for Function Derivatives and Discontinuities

Further safety requirements may be expressed using fuzzy rules of the form (5):

$$\hat{A}i : \textbf{IF } (x \text{ is } INCREASING) \textbf{ THEN } (y_i \text{ is } DECREASING) \quad (5)$$

The purpose of this rule is to qualitatively express a constraint on the input-output relationship (related to failure mode 3) by constraining the sign of $a_{i,1}$ in (3). Optionally, another constraint can be expressed by quantifying the rule (5). This quantification simply prescribes limits on output derivatives. The maximum output derivative for $a_{i,1}$ is $\pm \max a_{i,1}$ and the minimum is $\pm \min a_{i,1}$ (for SISO rules). These constraints remove potentially hazardous output fluctuations for failure mode 4. The static learning algorithm can adhere to these constraints through enforcement during optimisation of (3). This is achieved by analysing overlapping rule outputs at each input set edge and ensuring the difference is within prescribed limits. This type of product-based argument prevents failure modes that may result in the output changing rapidly, too slowly or in the wrong direction. Incorporating such constraints highlight the potential to control various functional properties according to safety requirements.

4 Example of SCANN Operation

The original FSOM has been used for a wide range of non-linear function approximation problems. The ability of the SCANN can be demonstrated by dynamic and static learning algorithms. Due to space constraints, details of full case study cannot be presented here. However, figure 2(a-b) illustrates the ability to adapt given unrepresentative and representative training data. Figure 2(c) illustrates how unsafe behaviour is constrained within bounds hinting that performance is always limited to keep within safe regions.

A real-world example which has used fuzzy control systems is the gas turbine aero-engine [10]. This approach can potentially help reduce cost by optimising the fuel flow under changing conditions (engine wear). All attempted bound violations can be logged and used to indicate the need for engine maintenance. The SCANN approach can provide efficiency in terms of reduced cost though maximising performance without compromising on safety.

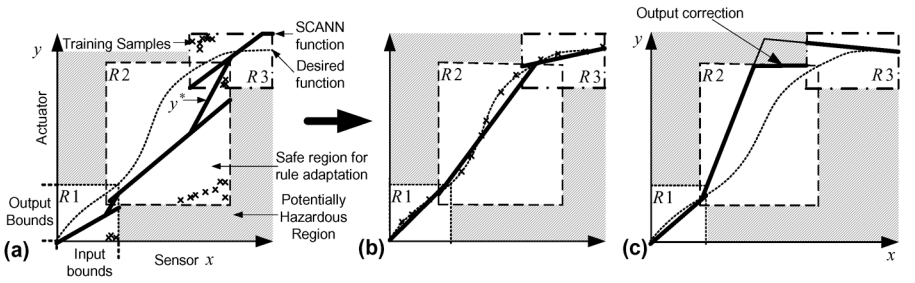

Fig. 2. (a) Bounding boxes used to define extremes for function mappings and converges onto unrepresentative data. (b) Non-linear function using representative data. (c) Constraints force behaviour within safe region (for *R2*) during generalisation.

5 Conclusions

This paper exploits the 'transparency' offered by the FSOM to easily integrate safety constraints over various functional properties. Our approach demonstrates how the behaviour of the SCANN is both predictable and controllable whilst enabling generalisation and learning post-certification. To adhere to safety requirements, constraints are enforced by the SCANN learning algorithms. For ease of explanation, a handful of solutions to safety arguments (extracted from a complete safety case) have been outlined. Compelling analytical certification arguments such as these are required for highly-dependable roles in safety critical systems.

References

1. Lisboa, P., Industrial use of safety-related artificial neural networks. Health & Safety Executive 327, (2001).
2. Hull, J., D. Ward, and R. Zakrzewski, Verification and Validation of Neural Networks for Safety-Critical Applications, Barron Associates, Inc. and Goodrich Aerospace, Fuel and Utility Systems (2002).
3. Kurd, Z., Artificial Neural Networks in Safety-critical Applications, First Year Dissertation, Department of Computer Science, University of York, 2002
4. Kurd, Z. and T.P. Kelly, Safety Lifecycle for Developing Safety-critical Artificial Neural Networks. 22nd International Conference on Computer Safety, Reliability and Security (SAFECOMP'03), 23-26 September, (2003).
5. Vuorimaa, P., Fuzzy self-organising map. Fuzzy Sets and Systems. 66 (1994) 223-231.
6. Ojala, T., Neuro-Fuzzy Systems in Control, Masters Thesis, Department of Electrical Engineering, Tampere University of Technology, Tampere, 1994
7. Kurd, Z. and T.P. Kelly, Establishing Safety Criteria for Artificial Neural Networks. In Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'03), Oxford, UK, (2003).
8. CISHEC, A Guide to Hazard and Operability Studies, The Chemical Industry Safety and Health Council of the Chemical Industries Association Ltd. (1977).
9. Bilgic, T. and I.B. Turksen, *Measurement of membership functions: theoretical and empirical work*, in Handbook of fuzzy sets and systems, In Dubois and Prade (1997).
10. Chipperfield, A.J., B. Bica, and P.J. Fleming, Fuzzy Scheduling Control of a Gas Turbine Aero-Engine: A Multiobjective Approach. IEEE Trans. on Indus. Elec. 49(3) (2002).