Choosing Appropriate Neuro–Fuzzy Models

Detlef Nauck and Rudolf Kruse
Department of Computer Science
Technical University of Braunschweig
Bueltenweg 74 – 75, D–38106 Braunschweig, Germany
Tel. +49.531.391–3155, Fax: +49.531.391–5936, email: nauck@ibr.cs.tu-bs.de

Abstract

To use fuzzy controllers for automization tasks appropriate fuzzy sets and fuzzy rules have to be defined. This can be difficult in some domains, and the resulting controller has to be tuned. Neuro–fuzzy models can help in this tuning process by adapting fuzzy sets and creating fuzzy rules. Combinations of neural networks and fuzzy controllers are suitable if there is only partial knowledge in the form of fuzzy sets and fuzzy rules, but training data is available. To be be able to choose an appropriate model one has to know the different approaches to neural fuzzy control. In this paper we present a classification of generic neural fuzzy controllers and give some hints when to choose a certain type of model.

1 Introduction

Classical control theory is based on mathematical models that describe the behavior of a plant under consideration. The main idea of fuzzy control on the other hand is to build a model of a human control expert who is capable of controlling the plant without thinking in a mathematical model. The control expert specifies his control actions in the form of linguistic rules. These control rules are translated into the framework of fuzzy set theory providing a calculus which can simulate the behaviour of the control expert.

The specification of good linguistic rules depends on the knowledge of the control expert, the translation into fuzzy set theory is not formalized, and arbitrary choices concerning for example the shape of membership functions have to be made. These uncertainties in the process of building a fuzzy controller mostly results in a heuristic tuning process to overcome the initial design errors.

The combination of neural networks and fuzzy controllers to so called *neuro–fuzzy models* can help to enhance the performance of the controller by using special learning algorithms. There are several approaches to neuro–fuzzy systems from which a designer can choose. But to make an appropriate choice an overview and a classification of existing neuro–fuzzy models is needed.

The choice of a special neuro–fuzzy architecture depends on the available data describing the control task. It is not always possible to provide a *fixed learning task,* i.e. samples of input/output data, so that gradient descent methods can be used for learning. Sometimes there is only a *free learning task,* i.e. just samples of input data, and there is no knowledge about the correct output value (control action). In this case reinforcement learning or clustering methods have to be used. Also the availability of special hardware (neural or fuzzy) has an influence on the choice of a neuro–fuzzy control architecture.

In this paper we first discuss two generic approaches to neuro–fuzzy models and give then some guidelines for choosing appropriate neural fuzzy controllers.

2 Cooperative Neuro–Fuzzy Models

One possible kind of combination between neural networks and fuzzy controllers uses both models independently. The combination is determined by the fact that a neural net is learning or optimizing several parameters of a fuzzy controller before the controller is used (offline), or while it operates (online). This kind of combination is called *cooperative neuro–fuzzy controllers.*

In: Proc. EUFIT'94, Aachen 1994, pp. 552–557

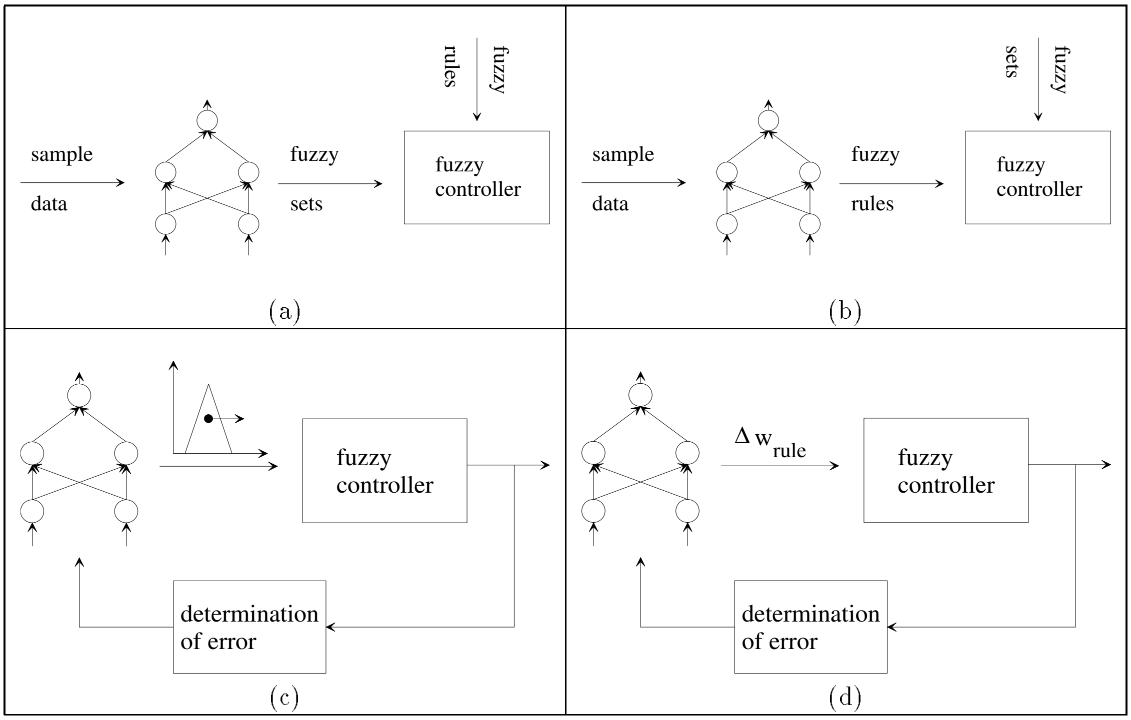

Figure 1: Different approaches to cooperative neuro–fuzzy systems

Fig. 1 shows four different kinds of possible neuro–fuzzy models:

(a) The neural net derives the membership functions from training data. This can be done by learning special parameters or by using several neural nets do approximate the functions directly. The fuzzy sets are learned offline, and are used with predefined fuzzy rules to implement a fuzzy controller. An approach to learn parameters of membership functions is described e.g. in [Nomura et al., 1992]. Enhanced versions of this learning algorithm can be found in [Bersini et al., 1993; Nauck, 1994b; Nauck et al., 1994].

(b) The neural net derives linguistic control rules from training data. This is mostly done by using clustering algorithms, like Kohonen's feature maps or similar neural architectures. The learning is done offline, and the fuzzy sets have to be defined in another way. An approach to learn fuzzy rules by linguistic interpretation of a self–organizing map is described in [Pedrycz and Card, 1992]. A less sophisticated approach using adaptive vector quantization can be found in [Kosko, 1992].

(c) The neural net adapts parameters of the fuzzy sets online, i.e. while the fuzzy controller operates. It is necessary to know the fuzzy rules, and initial fuzzy sets. In addition an error measure has to be defined that guides the learning process. If a fixed learning task is available this model can also learn offline. An approach that uses fuzzy associative memories (FAM) and learns online is described in [Yamaguchi et al., 1992].

(d) The neural net learns weight factors applied to the fuzzy rules in an online or offline mode. These kind of weights are usually interpreted as the "importance" of a rule [von Altrock et al., 1992; Kosko, 1992], and it modifies the output of a rule. Fuzzy rules and fuzzy sets have to be known in advance. The semantics of this kind of weights is not clear, though. Weighting of a rule is equivalent to change the fuzzy sets in the conclusions. Usually this leads to non–normal fuzzy sets, and identical linguistic values are represented in different ways in different rules.

In addition to these kind of cooperative models one can think of approaches where neural nets are used to preprocess or postprocess the input or output of a fuzzy controller. These approaches do not care about an optimization of the controller itself, but are concerned about optimizing the control behavior of the whole system. Learning takes place only in the neural network part, the fuzzy controller remains unchanged.

For this reason we do not call these approaches neuro–fuzzy models. They may be called neural pre– or postprocessing approaches, and they are used in some Japanese consumer products that constantly adapt to the customers' desires [Takagi, 1994].

3 Hybrid Neuro–Fuzzy Models

The other kind of combination between neural networks and fuzzy controllers are called *hybrid neuro–fuzzy models*. These approaches create an homogenous architecture that is usually neural network oriented. This can be done by interpreting a fuzzy controller as a special neural net or by implementing a controller using neural networks.

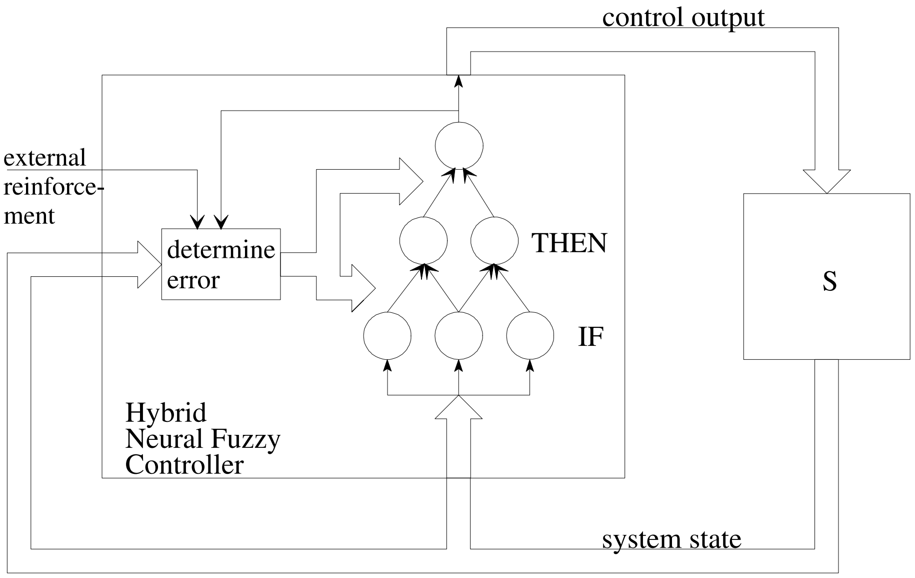

Figure 2: A generic model of a hybrid neuro–fuzzy controller

In fig. 2 a generic hybrid neural fuzzy controller is shown. The advantage of this model is the consistent architecture that makes the communication between to different models unnecessary. The idea of an hybrid model is the interpretation of the fuzzy rulebase in terms of a neural network. This way the fuzzy sets can be interpreted as weights, and the rules, input variables, and output variables can be represented as neurons.

The learning algorithm results, like in neural networks, in a change of the architecture, i.e. in an adaption of the weights, and/or in creating or deleting connections. These changes can be interpreted both in terms of a neural net and in terms of a fuzzy controller. This last aspect is very important, as the black box behavior of neural nets is avoided this way. This means a successful learning procedure results in an explicit increase of knowledge, that can be represented in form of a fuzzy controller's rule base.

Hybrid neural fuzzy controllers learn in a supervised mode. For this either a fixed learning task or a free learning task together with an external reinforcement signal is needed. A fixed learning task can be created by observing someone who is capable to control the plant under consideration. This data can be used for offline learning. Under certain circumstances a fixed learning task can be created online. For this a meta controller is needed that acts as a teacher for the neural fuzzy controller. This approach makes sense, if the meta controller is a neural controller. By using it to train a neural fuzzy controller one obtains a system that can be interpreted in terms of a fuzzy rule base. This way the knowledge stored implictely in the neural controller is made accessible.

Reinforcement learning based on a free learning task can only be done online at the system to be controlled or a simulation of it. The system states create the free learning task, and an error or reinforcement signal that depends on the system state following a control action drives the learning procedure. This signal can be provided externally by a trainer, or it can be derived by the training algorithm itself. For this it is necessary to know the goal of the control process and to be able to judge system states as "good" or "bad", respectively. Most of the time this is more easily done than to provide a fixed learning task.

Of course, reinforcement learning is only possible, if operating the plant during the learning process cannot result in any harmful situations, or if the plant can be simulated. A simulation can be e.g. obtained by training a neural network to approximate the input/output behavior of the plant.

Hybrid neuro–fuzzy controllers are realized by approaches like ARIC [Berenji, 1992], GARIC [Berenji and Khedkar, 1992; Berenji et al., 1993], ANFIS [Jang, 1992], or the NNDFR model [Takagi and Hayashi, 1991]. These approaches consist all more or less of special neural networks, and they are capable to learn fuzzy sets. Some of them have severe semantical problems when it comes to interpret the learning results (for an overview see [Nauck et al., 1993]).

An approach that is capable of learning fuzzy sets and fuzzy rules is presented by the NEFCON model [Nauck and Kruse, 1993; Nauck and Kruse, 1994; Nauck, 1994a]. NEFCON learns by reinforcement using a rule based fuzzy error measure. The learning results are always interpretable in terms of a fuzzy rule base.

4 Guidelines for Selecting Neuro–Fuzzy Models

Before a neuro–fuzzy controller can be applied either a set of learning data has to be provided, or a function to evaluate the system states must be known in order to create a learning task. If this basic requirement is met, it has to be decided, whether an exisiting fuzzy controller is to be optimized, or the necessary fuzzy sets and/or fuzzy rules are to be created by the desired neuro–fuzzy controller.

If an existing controller has to be optimized by using a neural network, there is only the possibility to adapt the fuzzy sets. Combinations that are able to optimize a predefined rule base are not known. One possible solution would be to transform the controller into the hybrid FUN model [Sulzberger et al., 1993] that is able to change a rule base by a stochastic learning procedure, but without changing the number of rules. After the learning process the model could be retransformed, but there may be undesired semantical problems in the interpretation of the rule base [Nauck et al., 1994].

Depending on the type of learning task, one of the approaches in Fig. 1.a or 1.c can be chosen to optimize fuzzy sets. Approach 1.a can be used if a fixed learning task is available that defines membership degrees, or if the membership functions are implemented by neural nets. Learning has to be done offline, and has to be resumed, if the controller fails the subsequent tests. Approach 1.c can be used, if online reinforcement learning at the physical system or its simulation is possible. This seems to be the most reasonable way to refine a controller that is already able to perform to some extent.

One can also think of using approach 1.d to optimize an existing controller, because scaling the rule outputs is equivalent to change the membership functions of the conclusions. However there may be the aforementiond semantical problems. It is also possible to use neural postprocessing approaches.

The model 1.b can only be used to create fuzzy rules if the fuzzy sets are known. After the rule base has been build based on training data the resulting controller is not adaptive anymore. If the controller fails, a completely new rule base has to be created. A solution to this problem is to use this approach together with 1.a or 1.c to adapt the predefined fuzzy sets so they fit to the rule base learned by the model.

If a new controller has to be constructed, and the fuzzy sets and/or fuzzy rules are not known or are only known to some extent, a hybrid model is more suitable. But even if only the fuzzy sets of an existing controller have to be adapted the use of hybrid models make sense. The controller can be transformed into the hybrid model, and be retransformed after learning.

To construct a fuzzy controller based on cooperative neuro–fuzzy approaches it may be necessary to successively use several models to learn fuzzy sets and fuzzy rules. A suitable hybrid model on the other hand, has the advantage of an homogeneous architecture, and may have the availability to do both. Learning of rules can also be sped up by using prior knowledge to define a partial rule base. This is of no use for approaches like 1.b, because they create the rules completely out of training data.

In Table 1 the NEFCON model is compared to hybrid approaches that are only able to learn membership functions. The NEFCON architecture can be interpreted as a neural network and as a fuzzy controller. It can make use of prior knowledge in the form of fuzzy sets and fuzzy rules, and the learning process does not afflict any semantical problems. For this reason it is suited for all kinds situations, where any kind of neuro–fuzzy model is applicable.

Table 1 displays under which circumstances the discussed generic models can be applied. If the decision for a certain type of model has been made, a specific model of this type has to be chosen. We cannot compare specific approaches here, this will be done in a longer comprehensive version of this paper. We have discussed several approaches in [Nauck et al., 1994].

	Cooperative Models				Hybrid Models	
Criteria	(a)	(b)	(c)	(d)	FS only	NEFCON
Fuzzy controller exists, adapt FS	$\circ^{(1)}$		$\bullet$	$\circ^{(2)}$	$\bullet$	$\bullet$
FR known, FS unknown	$\circ^{(1)}$		$\bullet^{(3)}$		$\bullet^{(3)}$	$\bullet^{(3)}$
FR unknown, FS known		$\bullet$				$\bullet$
FR unknown, FS unknown		$\circ^{(4)}$				$\bullet^{(3)}$
FR partly known						$\bullet$

Legend: FS: fuzzy set , FR: fuzzy rule

$\bullet$: usable, $\circ$: restrictedly usable

$^{(1)}$ only with fixed learning task that defines membership degrees

$^{(2)}$ only if the semantics of the controller is not of interest

$^{(3)}$ parameters of membership functions have to be initialized

$^{(4)}$ only together with (c), or (a) and (c), respectively

Table 1: Criteria for choosing neuro–fuzzy combinations

5 Conclusions

In this paper we have discussed several generic approaches to neuro–fuzzy systems. When a fuzzy controller has to be built a combination with neural networks can avoid problems in the tuning process, and in some case a neural fuzzy controller may be the only solution if only partial knowledge in the form of fuzzy rules is available but there are is a lot of training data. If a neuro–fuzzy approach shall be used several points have to checked:

- Are state/action examples available, to create a fixed learning task, and to make offline learning possible?
- Is online reinforcement learning possible, i.e. is there an performance evaluation function and can learning be done during plant operation, or is there a simulated plant?
- Is a semantical interpretation of the learning result desirable?
- Is an existing controller to be optimized, or a new one to be created?
- Is there prior knowledge (fuzzy rules / fuzzy sets) or just training data?

Cooperative approaches can be suitable to optimize the fuzzy sets of existing fuzzy controllers (1.c), but they are less suitable to learn fuzzy rules (1.b), for what fuzzy clustering methods [Bezdek and Pal, 1992] can be used alternatively, and there seldom exists training data to learn fuzzy sets from scratch (1.a). Learning weights for fuzzy rules (1.d) causes semantical problems.

Hybrid approaches can be used to optimize existing fuzzy controllers, by transforming them into the respective model, or they can be used to create a new controller. Suitable hybrid approaches like NEFCON have an interpretable homogenous architecture and do not have problems with undesired semantical inconsistencies of the learning process. NEFCON can learn fuzzy sets and fuzzy rules, and can use prior knowledge in the form of partial rule bases. It can be used to optimize or to create fuzzy controllers under most circumstances.

References

C. von Altrock, B. Krause and H. Zimmermann (1992). *Advanced Fuzzy Logic Control Technologies in Automotive Applications*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 835–842, San Diego.

H. R. Berenji (1992). *A Reinforcement Learning-Based Architecture for Fuzzy Logic Control*. Int. J. Approximate Reasoning, 6:267 –292.

H. R. Berenji and P. Khedkar (1992). *Learning and Tuning Fuzzy Logic Controllers Through Reinforcements*. IEEE Trans. Neural Networks, 3:724–740.

H. R. Berenji, R. N. Lea, Y. Jani, P. Khedkar, A. Malkani and J. Hoblit (1993). *Space Shuttle Attitude Control by Reinforcement Learning and Fuzzy Logic*. In *Proc. IEEE Int. Conf. on Neural Networks 1993*, pages 1396–1401, San Francisco.

H. Bersini, J.-P. Nordvik and A. Bonarini (1993). *A Simple Direct Adaptive Fuzzy Controller Derived from its Neural Equivalent*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1993*, pages 345–350, San Francisco.

J. C. Bezdek and S. K. Pal, eds. (1992). *Fuzzy Models for Pattern Recognition*. IEEE Press, New York.

J.-S. R. Jang (1992). *Self-learning Fuzzy Controller Based on Temporal Back-Propagation*. IEEE Trans. Neural Networks, 3:714–723.

B. Kosko (1992). *Neural Networks and Fuzzy Systems. A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, Englewood Cliffs.

D. Nauck (1994a). *Building Neural Fuzzy Controllers with NEFCON–I*. In R. Kruse, J. Gebhardt and R. Palm, eds.: *Fuzzy Systems in Computer Science*, Artificial Intelligence. Vieweg, Wiesbaden.

D. Nauck (1994b). *Modellierung Neuronaler Fuzzy–Regler (Modeling Neural Fuzzy Controllers [in German])*. PhD thesis, Technische Universität Braunschweig.

D. Nauck, F. Klawonn and R. Kruse (1993). *Combining Neural Networks and Fuzzy Controllers*. In E. P. Klement and W. Slany, eds.: *Fuzzy Logic in Artificial Intelligence (FLAI93)*, pages 35–46, Berlin. Springer–Verlag.

D. Nauck, F. Klawonn and R. Kruse (1994). *Neuronale Netze und Fuzzy–Systeme*. Vieweg, Wiesbaden.

D. Nauck and R. Kruse (1993). *A Fuzzy Neural Network Learning Fuzzy Control Rules and Membership Functions by Fuzzy Error Backpropagation*. In *Proc. IEEE Int. Conf. on Neural Networks 1993*, pages 1022–1027, San Francisco.

D. Nauck and R. Kruse (1994). *NEFCON–I: An X–Window based Simulator for Neural Fuzzy Controllers*. In *Proc. IEEE Int. Conf. Neural Networks 1994 at IEEE WCCI'94*, Orlando.

H. Nomura, I. Hayashi and N. Wakami (1992). *A Learning Method of Fuzzy Inference Rules by Descent Method*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 203–210, San Diego.

W. Pedrycz and W. C. Card (1992). *Linguistic Interpretation of Self-Organizing Maps*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 371–378, San Diego.

S. M. Sulzberger, N. N. Tschichold-Gürman and S. J. Vestli (1993). *FUN: Optimization of Fuzzy Rule Based Systems using Neural Networks*. In *Proc. IEEE Int. Conf. on Neural Networks 1993*, pages 312–316, San Francisco.

H. Takagi (1994). *Cooperative Systems of Neural Networks and Fuzzy Logic and its Application to Consumer Products*. In J. Yen, R. Langari and L. A. Zadeh, eds.: *Industrial Applications of Fuzzy Control and Intelligent Systems*. IEEE Press. To be published.

H. Takagi and I. Hayashi (1991). *NN–Driven Fuzzy Reasoning*. Int. J. Approximate Reasoning, 5:191–212.

T. Yamaguchi, K. Goto, T. Takagi, K. Doya and T. Mita (1992). *Intelligent Control of a Flying Vehicle using Fuzzy Associative Memory System*. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 1139–1149, San Diego.