Neuro-fuzzy Systems: A Short Historical Review

Detlef D. Nauck and Andreas Nürnberger

Abstract. When the popularity of fuzzy systems in the guise of fuzzy controllers began to rise in the beginning of the 1990s researchers became interested in supporting the development process by an automatic learning process. Just a few years earlier the backpropagation learning rule for multi-layer neural networks had been rediscovered and triggered a massive new interest in neural networks. The approach of combining fuzzy systems with neural networks into neuro-fuzzy systems therefore was an obvious choice for making fuzzy systems learn. In this chapter we briefly recall some milestones on the evolution of neuro-fuzzy systems.[1]

1 Introduction

The term *neuro-fuzzy systems* (also neuro-fuzzy methods or models) refers to combinations of techniques from neural networks and fuzzy systems [26, 50]. This typically does not mean that a neural network and a fuzzy system are used in some kind of combination, but that a fuzzy system is created from data by some kind of (heuristic) learning method that is motivated by learning procedures used in neural networks.

Neuro-fuzzy methods are usually applied, if a fuzzy system is required to solve a function approximation problem — or a special case of it like classification or

Detlef D. Nauck
BT, Research and Venturing, Ipswich, IP5 3RE, UK
e-mail: detlef.nauck@bt.com

Andreas Nürnberger
Faculty of Computer Science, Otto-von-Guericke University of Magdeburg,
39106 Magdeburg, Germany
e-mail: andreas.nuernberger@ovgu.de

[1] This paper is an updated and extended version of a conference paper of the authors that appeared at NAFIPS 2005 [49].

C. Moewes et al. (Eds.): Computational Intelligence in Intelligent Data Analysis, SCI 445, pp. 91–109.
springerlink.com

control [58] — and the otherwise manual design process should be supported or replaced by an automatic learning process. The (manual) design of a fuzzy system requires specification of fuzzy partitions (parameters) for each variable and a set of fuzzy rules (structure). If the fuzzy system does not perform well, structure or parameters or both must be modified accordingly. This can be a very lengthy and error-prone process that is effectively based on trial and error. In order to support this design process learning techniques based on sample data became a popular research topic at the beginning of the 1990s when fuzzy systems in the guise of fuzzy controllers first became successful and widely known.

The history of neuro-fuzzy systems can be roughly structured into first feed-forward systems for control and function approximation and later — mainly due to the success of fuzzy systems in control — approaches for classification and clustering problems, where interpretable solutions and the introduction of prior knowledge into the learning process is also quite often very beneficial. More recently, several researchers studied the usability of hierarchical and recurrent architectures. In the following, we discuss the major approaches that have been proposed for these fields, if possible, in chronological order.

2 Feed-Forward Architectures

One of the first works that proposed a combination of neural network learning methods with the concepts of fuzzy systems was proposed in 1985 by Keller and Hunt [29]. In this paper, the authors proposed an approach to stabilize the perceptron learning algorithm for classification problems using fuzzy techniques. They introduced a fuzzy membership of data items to the searched classes in order to improve the convergence of the learning algorithm. Motivated by this early work, several other approaches had been proposed that deal with the combination of neural networks and fuzzy systems and that have driven this field of research. In the area of approximate reasoning, for example, several approaches have been proposed in 1991 and 1992 [14, 28, 31, 30, 32, 57, 56]. These models are parts of fuzzy expert systems, or support fuzzy decision making with the help of neural networks. Since these methods do not integrate the neural network and fuzzy system structure in a homogenous architecture, but one adapts the parameters of the other in a cooperative way, we do not consider them as (hybrid) neuro-fuzzy system and thus do not cover them as part of this contribution. The same holds for approaches suggested by Miyoshi et al. and Yager and Filev in 1993 and 1992, respectively. In these approaches the fuzzy sets are not modified, but parametrized t-norms and t-conorms are used. Miyoshi et al. [39] proposed an approach to adapt parameters of these operators by backpropagation, while Yager and Filev suggest adaptive defuzzification strategies [77, 78]. Yager and Filev used a parametrized defuzzification operation and define a supervised learning algorithm to determine the parameters. However, even though we do not discuss co-operative approaches in detail, we will refer to them, if other (hybrid) approaches made use of the proposed more general techniques and ideas. The same holds for problems of learnability and interpretability of (neuro) fuzzy systems, where some discussions can be found in [22, 64, 47, 11, 42]. For an overview that is focussed on neuro-fuzzy methods for rule generation see [38].

2.1 Control

Neuro-fuzzy controllers were together with neuro-fuzzy systems for function approximation the first neuro-fuzzy approaches. The principles and architectures are similar and the main difference is basically the learning mechanism. While function approximation models can use supervised learning based on a training set, controllers need to discover a model in a setting where target outputs are not known. Neuro-fuzzy controllers therefore use reinforcement learning and require either a model of or direct feedback from the process they are supposed to control.

2.1.1 ARIC and GARIC

On of the first neuro-fuzzy controllers was suggested by Berenji in 1992. The ARIC model (Approximate Reasoning-based Intelligent Control) implements a fuzzy controller by using several specialized feed-forward neural networks. The architecture of ARIC is similar to an adaptive critic, a special neural controller learning by reinforcement [75], and it generalizes the neural model of Barto et al. [3] to the domain of fuzzy control. ARIC consists of two neural modules, the ASN (Action Selection Network) and the AEN (Action state Evaluation Network). The AEN is an adaptive critic that is trained by backpropagation and that evaluates the actions of the ASN.

The ASN itself consists of two feed-forward three-layer neural networks. One network calculates a confidence value that is used to change the output of the second network which is a direct representation of a fuzzy controller. The input layer represents state variables of a process and the hidden units represent fuzzy rules. Their inputs are the antecedents, and their outputs the consequents of the rules. ARIC assumes that the rule base is known in advance. The output of the control network represents the defuzzified control value of the fuzzy controller. The learning algorithm modifies connections weights in the ASN and so indirectly the represented fuzzy sets.

Implementing learning by modifying connection weights was a popular approach in early neuro-fuzzy systems, but it was later shown that this leads to problems in interpreting the learning outcome [40].

The ARIC model was later extended to GARIC (Generalised ARIC, Fig. 1) [4, 5, 6]. Like ARIC it consists of an evaluation network (AEN) and an action network (ASN). The ASN does not use any weighted connections, but the learning process modifies parameters stored within the units of the network. The other network of the ASN which produces a confidence measure no longer exists.

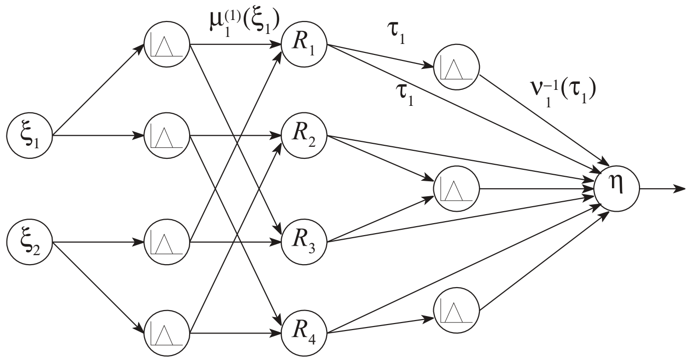

Fig. 1 GARIC represents a fuzzy system as a feed-forward network [5]

2.1.2 NEFCON

NEFCON is a model for neural fuzzy controllers proposed by Nauck in 1994 [44]. It is based on the architecture of the generic fuzzy perceptron [50] and implements a Mamdani-type fuzzy system. NEFCON is probably the first neuro-fuzzy system that tries to introduce the notion of interpretability by preventing identical linguistic terms being represented by more than one membership function, even though the general idea of a fuzzy perceptron has been proposed already earlier by Keller and Tahani as well as Pal and Mitra in 1992 [31, 54] and similar concepts have been discussed by Gupta and Rao in 1994 [18].

Like ARIC and GARIC, the learning algorithm for NEFCON is based on the idea of reinforcement learning but instead of an adaptive critic network it uses a fuzzy rule base to describe a fuzzy error.

Fig. 2 shows a NEFCON system with two input variables, one output variable and five rules. The unit R_3 for instance represents the rule

$$R_3\text{: } \textbf{If } \xi_1 \textbf{ is } \mu_2^{(1)} \textbf{ and } \xi_2 \textbf{ is } \mu_2^{(2)} \textbf{ then } \eta \textbf{ is } \nu_2.$$

The connections in NEFCON are weighted with fuzzy sets instead of real numbers, and some connections always have the same weight (illustrated by ellipses around connections) in order to ensure the integrity of the rule base.

Several learning methods have been proposed for this model. In [53] an overview is given. One major problem of all methods is, that they require at least some prior knowledge of the system to be controlled in order to define the (fuzzy) error signal that is used for learning.

Recently, models have been proposed that try to solve these fundamental problems using hierarchical models and Q-learning, see, e.g., [13, 15]. Q-learning has been already successfully used in combination with neural networks in order to control more complex systems, see e.g. [60].

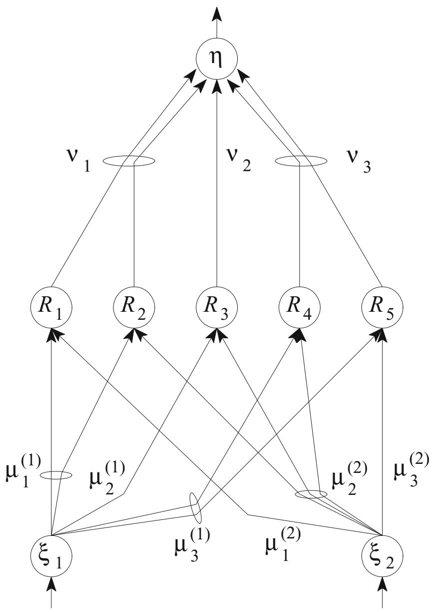

Fig. 2 A NEFCON system with two input variables and five rules

2.2 *Approximation*

Many neuro-fuzzy systems for function approximation are based on Takagi-Sugeno fuzzy systems, because they allow the application of gradient descent learning, if differentiable membership function (e.g. Gaussians) are used.

2.2.1 ANFIS

One of the first and still one of the popular neuro-fuzzy systems is Jang's ANFIS model proposed in 1991 [26, 23, 24, 25]. ANFIS (adaptive network-based fuzzy inference system) is a neuro-fuzzy method to determine the parameters of a Sugeno-type fuzzy model which is represented as a special feed-forward network (see Fig. 3). It encodes fuzzy rules of the form

$$R_r\text{: } \textbf{If } x_1 \textbf{ is } \mu^{(1)}_{j_1} \wedge \ldots \wedge x_n \textbf{ is } \mu^{(n)}_{j_n} \textbf{ then } y = \alpha^{(r)}_0 + \alpha^{(r)}_1 x_1 + \ldots + \alpha^{(r)}_n x_n.$$

Each node of the first layer is connected to exactly one of the n input variables and stores the three parameters of a membership function. The k nodes in the second layer represent the antecedents of the fuzzy rules. They compute the degree of fulfillment by multiplying the degrees of membership. The k nodes of the third layer compute the relative degree of fulfillment for each rule. The output values of the rules are computed by the k nodes of layer 4. They store the consequent parameters. Each node in this layer is connected (not drawn in Fig. 3) to one node of layer 3 and to all input variables. The output node in layer 5 computes the overall output value. If the model must compute $m > 1$ output values, then there are m output nodes and mk nodes in layer 4.

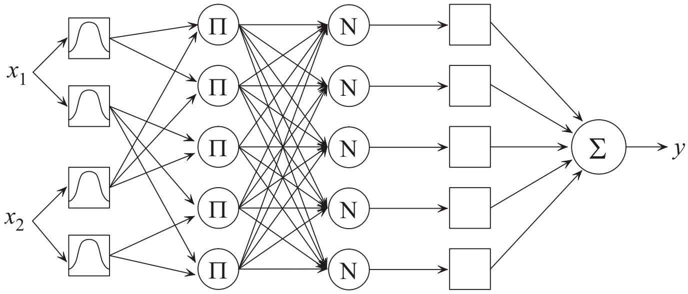

Fig. 3 ANFIS encodes a Sugeno-type fuzzy model in a feed-forward network structure [25]

ANFIS uses only differentiable functions, and therefore it is easy to apply standard gradient descent learning procedures from neural network theory. For ANFIS a mixture of backpropagation (BP) and least mean square estimation (LSE) is suggested by Jang [25]. BP is used to compute the updates of the antecedent parameters, i.e. the parameters of the fuzzy sets, and LSE is used to determine the updates for the consequent parameters, i.e. the coefficients of the linear combinations in the consequents.

ANFIS does not learn the structure of the fuzzy system, but it simply creates rules from all possible combinations of input fuzzy sets. Initial fuzzy partitions have to be specified. The consequent parameters are initialised by small random numbers.

2.2.2 Radial Basis Function Networks

Radial basis function networks (RBFN) are often connected to fuzzy systems, because the activation functions $h(||x-c||^2) = exp(-\frac{||x-c||^2}{2\sigma^2})$ of their hidden units can be interpreted as multidimensional membership functions. If this interpretation is assumed, then fuzzy rules can be extracted from an RBFN. To do this the RBF functions of the hidden units must be projected onto the individual dimensions. This way fuzzy sets are obtained that must be labelled with suitable linguistic terms. In general, this kind of rule generation suffers from the problem that the antecedent of the resulting fuzzy rule is not necessarily equivalent to the original corresponding RBF function. We only have equivalence if the area described by it is an axis-paralell hyperellipsoid and the product is used to compute the degree of fulfilment of the extracted fuzzy rule. If min is used as a t-norm the support of a rule is equivalent to the smallest hyperbox which contains the hyperellipsoid described by $h(||x-c||^2) > \varepsilon$. If the RBFN uses a generalised radial basis function with an inverse co-variance matrix that is not diagonal it is no longer equivalent to a fuzzy system.

In 1993 Jang and Sun showed when a radial basis function network (RBFN) is equivalent to a TSK fuzzy model [27]. They found that the following conditions have to hold:

- The number of hidden units in the RBFN (receptive field units) is equal to the number of fuzzy if-then rules.
- The output of each fuzzy if-then rule is just a constant, i.e. the fuzzy system is a simplified special case of a TSK system which would normally have a linear combination as rule output.
- All membership functions in the fuzzy system are Gaussian functions with the same variance.
- The t-norm operator used to compute each rule's degree of fulfilment is the product.
- Both the RBFN and the fuzzy inference system use the same operation to compute the overall output, i.e.either weighted average or weighted sum.

Note that the third condition can be relaxed such that only the variances for each corresponding dimension (input variable) have to be identical. That means the RBFN can use basis function with an inverse covariance matrix that is a diagonal matrix and the elements on the diagonal need not be identical.

RBF networks have frequently be used to derive neuro-fuzzy approaches. For example, Fuzzy RuleNet [69] as discussed in Sect. 2.3.4 is an extension of the RuleNet model which is a special RBFN. The activation functions of the hidden units use the ∞-vector norm instead of the usual Euclidean vector norm. Thus the activation functions are defined over hyperboxes instead of hyperellipsoids [12, 70].

Fuzzy RuleNet is a typical approach that can be seen as being inspired by RBFN but having outgrown the limitation of functional equivalence to RBFN by using hyperboxes as the support of antecedents and max-min interference instead of product and weighted average. Similar hyperbox-oriented approaches have been presented by Berthold and Huber [10, 8, 9].

2.2.3 NEFPROX

NEFPROX [48] is like NEFCON based on a generic fuzzy perceptron and implements a Mamdani-type fuzzy system. Mamdani-type system are rarely used for function approximation purposes, because it is easier to train a Takagi-Sugeno-type system. NEFPROX has learning algorithms for structure learning and parameter learning, but it is typically less accurate than ANFIS.

2.3 Classification

Neuro-fuzzy classification systems became more popular in the second half of the 1990s. They are a special case of function approximators and their output is typically a fuzzy classification of a pattern, i.e. a vector of membership degrees that indicates membership to different classes. With the rising interest in data mining, fuzzy classifiers became more and more important in the fuzzy system community.

2.3.1 The NNDFR Model

A very interesting and atypical neuro-fuzzy model from 1991 is the NNDFR model (Neural Network Driven Fuzzy Reasoning) by Takagi and Hayashi [66] which was developed around the same time as ANFIS. NNDFR is based on common neural networks that are structured by fuzzy system techniques. Its main purpose is classification.

An NNDFR system is based on n input variables $x_1,\ldots,x_n$, an output variable y and k fuzzy rules $R_1,\ldots,R_k$. It consists of $k+1$ multi-layer feed-forward neural networks trained by backpropagation and representing the fuzzy rules (Fig. 4). The system cannot be used to extract the parameters of a fuzzy system from it. Strictly speaking, we would not consider it as a neuro-fuzzy system. However, the structure of the partial networks and the interpretation of the outputs are motivated by fuzzy system techniques.

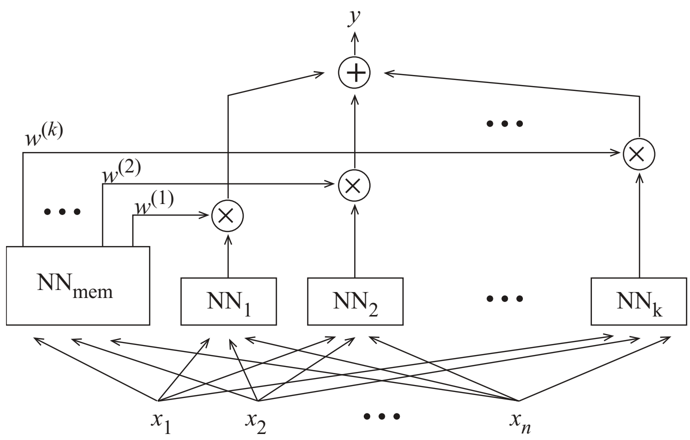

Fig. 4 Structure of an NNDFR system [66]

The linguistic rules used by the NNDFR model are of the form R_r: **If** $(x_1,\ldots,x_n)$ **is** A_r **then** $y = u_r(x_1,\ldots,x_n)$. This is not the usual form of linguistic rules used in fuzzy systems and is caused by the purely neural architecture. A_r is an n-dimensional membership function. There is no combination of single membership values. In an NNDFR system the neural network NN_{mem} provides for each rule R_r a value w_r that is interpreted as its degree of fulfillment.

The functions $u_1, \ldots, u_k$ are implemented by k neural networks $NN_1, \ldots, NN_k$ that determine the output values of the rules $R_1, \ldots, R_k$. The overall system output is given by $y = \sum_{r=1}^{k} w_r \, u_r(x_1, \ldots, x_n) / \sum_{r=1}^{k} w_r$.

2.3.2 FuNe-I

The neuro-fuzzy model FuNe-I [20, 21], proposed in 1992, is based on the architecture of a feed-forward neural network (Fig. 5). The network has five layers. The first layer contains a unit for each input variable and propagates the input values unchanged via weighted links to the second layer. This layer consists of units with sigmoid activation functions that are used to create membership functions. The third layer contains specialized units that are only used to represent fuzzy sets that do not touch the domain boundaries (see below). The units of the second and third layer propagate their activations via unweighted links to the fourth layer. Units from the second layer that have connections to the third layer are not connected to the fourth layer.

The fourth layer consists of units that represent fuzzy rules. Compared to other neuro-fuzzy approaches, the FuNe-I model is special because it uses three kinds of rules: the antecedents can be conjunctions or disjunctions, and there are rules with only one variable as antecedent (simple rules). A unit computes its activation — depending on the kind of rule it represents — by either a differentiable soft minimum, a differentiable soft maximum, or the identity function.

The fifth layer contains the output units that compute their input by a weighted sum and their activation by a sigmoid function. The FuNe-I model provides algorithms for structure and parameter learning and is one of the first neuro-fuzzy approaches that also considers rule learning.

FuNe-I was extended in 1994 to FuNe-II which can be used for fuzzy control problems. In a FuNe-II network a new output layer is created that is connected to the previous output layer. On the connections discrete samples of fuzzy sets are stored to represent control values. The activations of the new output units represent support points of a fuzzy set that must be defuzzified to obtain the final control value [19, 21].

2.3.3 NEFCLASS

NEFCLASS [45, 46, 41], proposed in 1995, is probably the first neuro-fuzzy approach that was able to handle missing values, both numeric and symbolic data in the same data set and to determine a rule-base fully automatically. NEFCLASS is also based on the idea of a generic fuzzy perceptron and focuses on creating small interpretable fuzzy rule bases.

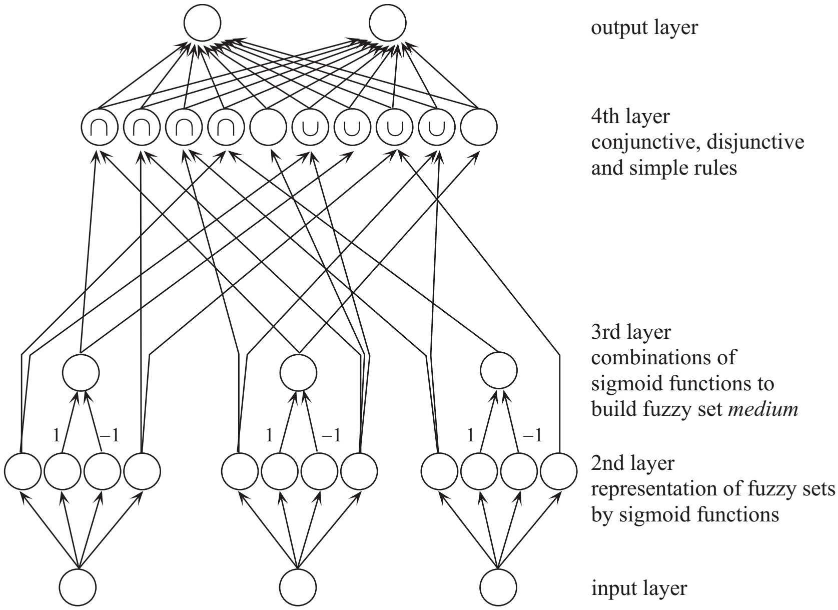

Fig. 5 The architecture of a FuNe-I system

The learning algorithm of NEFCLASS has two stages: structure learning and parameter learning. Rule (structure) learning is done by a variation of the approach by Wang and Mendel [73] which was extended to cover also symbolic patterns [43] and to use a rule performance measure for rule selection. In parameter learning the fuzzy sets are tuned by a simple backpropagation-like procedure that is based on a simple heuristics instead of a gradient descent approach. After learning NEFCLASS uses pruning strategies to reduce the number of rules as much as possible.

2.3.4 Fuzzy RuleNet

Fuzzy RuleNet [69], which was also proposed in 1995, is a neuro-fuzzy approach that is based on the structure of an radial basis function (RBF) network (see also Sect. 2.2.2). It is an extension of the RuleNet model, a special neural network, that can be seen as a variant of an RBF network [12, 70]. Instead of the usual radial basis functions — which represent hyperellipsoids — RuleNet uses hyperboxes for classification.

Fuzzy RuleNet allows hyperboxes to overlap. Each hyperbox represents a multidimensional fuzzy set given by a membership function in form of a hyperpyramid. By projecting the multidimensional fuzzy sets onto the individual dimensions we obtain triangular or trapezoidal fuzzy sets that describe the pattern features. The fuzzy classification rules obtained this way are equivalent to the multidimensional fuzzy sets, i.e. there is no loss of information as it would be in the case of hyperellipsoids used in fuzzy cluster analysis. The hyperboxes are created in a single cycle through the data. The learning algorithm adjusts the sizes of hyperboxes by extending them to cover new data or shrinking them in case of conflicts. This way a rule base and the parameters (fuzzy sets) are created in a single loop.

If a Fuzzy RuleNet is used for classification it computes its output by a winner-takes-all procedure to find the class of a given input pattern. It is also possible to adjust the definition such that the outputs are computed by a weighted sum. This way Fuzzy RuleNet can be used for function approximation.

Similar approaches to Fuzzy RuleNet are sometimes called hyperbox-oriented rule learners, and were known as early as 1992 [62, 63]. Newer variations are also called fuzzy graphs [9, 7]. The idea is always to cover a set of data with hyperboxes and connect each hyperbox with an output value. Hyperbox-oriented fuzzy rule learning can create solutions for benchmark problems in pattern recognition or function approximation very fast. If there are no contradictions in the training patterns and if there is only one output variable, then hyperbox-oriented learning algorithms can create solutions with no errors on the training data. In the worst case this leads to a situation, where each training pattern is covered by its individual hyperbox.

3 Recurrent Systems

In contrast to pure feed-forward architectures that have a static input-output behavior, recurrent models are able to store information of the past, e.g. prior system states, and can be thus more appropriate for the analysis of dynamic systems (see, for example, discussions concerning the approximation and emulation capabilities of recurrent neural networks [59, 37, 74]). If pure feed-forward architectures are applied to these types of problems, e.g. prediction of time series data or physical systems, the obtained system data usually has to be preprocessed or restructured to map the dynamic information appropriately, e.g. by using a vector of prior system states as additional input. If we apply a fuzzy system, this may lead to an exponential increase of the parameters — if we want to cover the whole system state space — that soon becomes intractable.

Recurrent neuro-fuzzy systems (RNFSs) can be constructed in the same way as discussed above for feed-forward neuro-fuzzy systems. So, they are based either on a recurrent fuzzy system or a recurrent neural network structure. However, the design and the optimization of (hierarchical) recurrent systems is, due to the dynamics introduced by the feed back connections, more difficult than that of feed forward systems. In Fig. 6 an example of a hierarchical RNFS is shown.

Probably the first recurrent fuzzy system that was combined with a (neural network motivated) learning method was proposed by Gorrini and Bersini in 1994 [17]. The proposed system is a Sugeno-Takagi-like fuzzy system and uses fuzzy rules with a constant consequent. The internal variables of this system may be defined manually, if the designer has sufficient knowledge of the system that should be modeled. No learning method for the rule base itself was proposed except to initialize the rule base randomly. However, the authors propose a learning approach

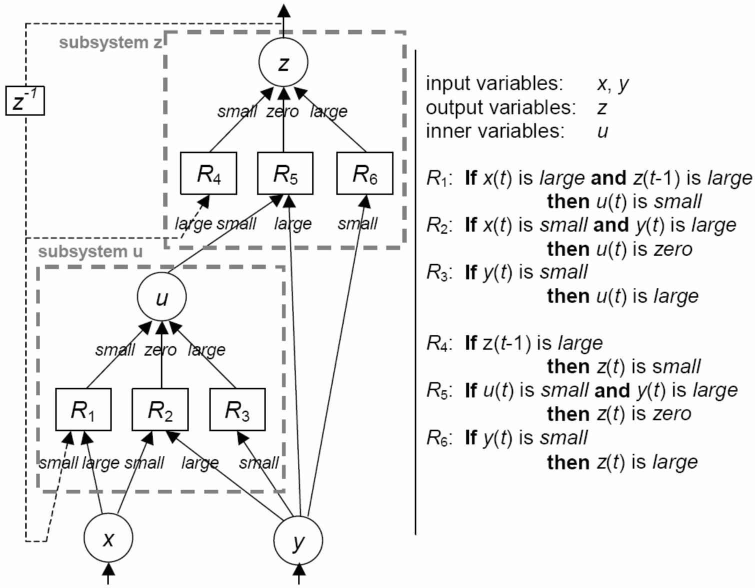

Fig. 6 Example of a simple hierarchical recurrent rule base consisting of two subsystems. The output of the system is reused by each subsystem as time-delayed input.

to optimize the parameters of a recurrent rule base, which was motivated by the real time recurrent learning algorithm [76]. According to Gorrini and Bersini the results for the approximation of a third order non-linear system for a given rule base was comparable to the approximation by a recurrent neural network. Unfortunately, a detailed discussion of the results was not given. Furthermore, the model had some insufficiencies. First of all, the structure has to be defined manually, since no learning methods for the rule base have been proposed. Furthermore, the learning is restricted to symmetric triangular fuzzy sets and the interpretability is not ensured, since the fuzzy sets are modified independently during learning. However, an extension to arbitrary (differentiable) fuzzy sets is easily possible.

Surprisingly, after this first model, for some time not much work had been published on recurrent systems that are also able to learn the rule base itself. Most likely the first models that were successfully applied to control — which, however, do not implement generic hierarchical recurrent models as described above — were proposed by Theocharis and Vachtsevanos in 1996 [68], Zhang and Morris in 1999 [80] and Lee and Teng in 2000 [35]. For example, Lee and Teng proposed a fuzzy neural network, which implements a modified Sugeno-Takagi-like fuzzy system with Gaussian-like membership functions in the antecedents and constant consequents. However, this model did not implement a fully recurrent system as shown in Fig. 6, but they restricted themselves to integrate feed back connections in the membership layer as depicted in Fig. 7.

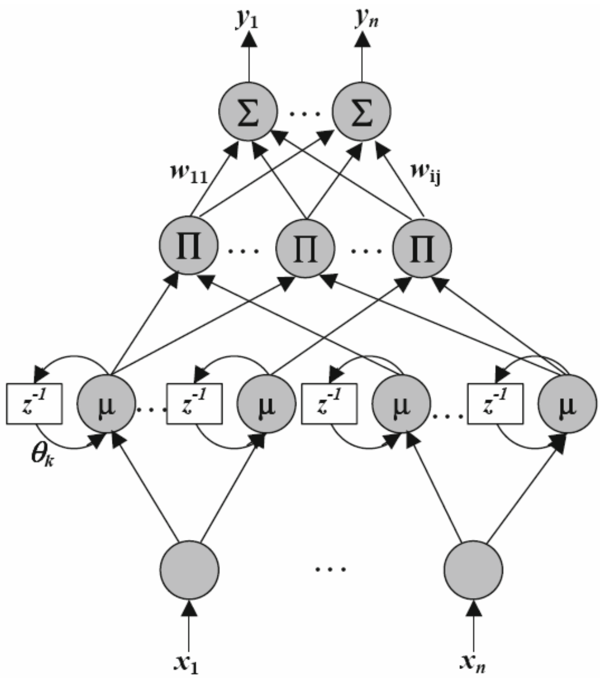

Fig. 7 Fuzzy neural network with simple feedback units as proposed by Lee and Teng

Approaches to learn hierarchical recurrent fuzzy system were presented in 2001 by Surmann and Maniadakis [65], who used a genetic algorithm, and Nürnberger [51], who proposed a template based approach to learn a structured rule base and a gradient descent based method motivated by the real time recurrent learning algorithm [76] to optimize the parameters of the learned rule base. The interpretability of the fuzzy sets of this model is ensured by the use of coupled weights in the consequents (fuzzy sets, which are assigned to the same linguistic terms share their parameters) and in the antecedents. Furthermore, constraints can be defined, which have to be observed by the learning method, e.g. that the fuzzy sets have to cover the considered state space. An example of the network structure is given in Fig. 8. However, the template based learning approach still had insufficiencies due to the use of a heuristic that created inner fuzzy sets. Therefore, in [52] a slightly modified approach was proposed, that learned the rule base using a genetic algorithm.

Furthermore, recurrent models that tackle specific problems of the learning process, properties of recurrent fuzzy systems or specific applications have been proposed in, e.g., [36, 72, 33].

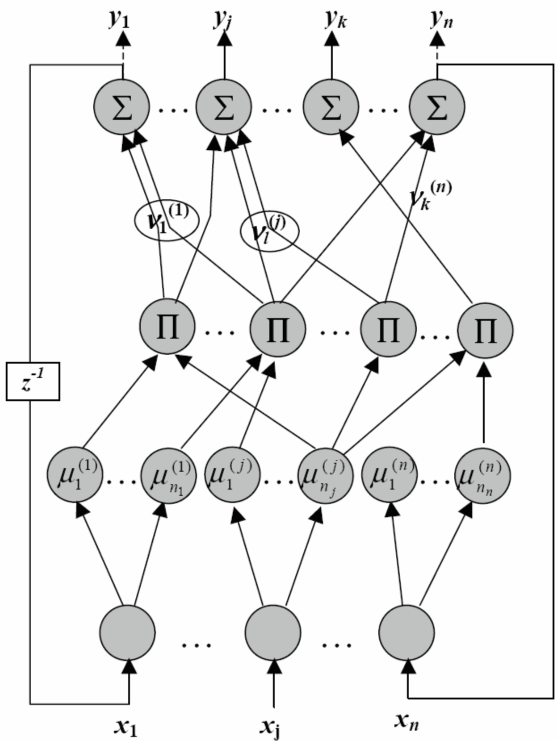

Fig. 8 Possible structure of the recurrent neuro-fuzzy system proposed by Nürnberger in 2001 [51] (using one time-delayed and one hierarchical feed-back). The first row of neurons defines the input variables, the second row the membership functions of the antecedents, the third row the fuzzy rules, and the fourth row the output variables. The membership functions of the consequents that are shared by rules are represented by coupled links from the rule to the output layer.

4 Outlook

Starting with neural network oriented architectures like ARIC and NNDFR neuro-fuzzy system quickly developed into network representations of fuzzy systems like we can see in ANFIS and NEFCON. In the second half of the 1990s we saw a lot of specific architectures for approximation, classification and control where neuro-fuzzy systems have covered a broad area of problems. Meanwhile, they found their way in quite diverse application areas where they are currently regularly applied, see e.g. recent works in geochemistry [81], geology [55], manufacturing [67], time series analysis [2] and signal processing [61].

However, there are still a lot of open research research questions in the area of adaptive control, where the combination of reinforcement learning methods with neuro-fuzzy architectures has made a lot of progress more recently (see, e.g., [16, 34, 71]). The same holds for applications in classification that became more and more important with the growing interest in data mining (see, e.g., [1, 79]). In this area questions of how to completely automate the learning process and how to guarantee a certain level of interpretability remain to be important issues.

5 Remarks

We like to apologize to all researchers we did not mention in this — for this broad topic — short article. This would have been impossible. We tried only to mark major developments in this area and may have missed some that would be considered by others as major contributions.

References

[1] Asaithambi, M., Manoharan, S.C., Subramanian, S.: Classification of Respiratory Abnormalities Using Adaptive Neuro Fuzzy Inference System. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012, Part III. LNCS, vol. 7198, pp. 65–73. Springer, Heidelberg (2012)

[2] Atsalakis, G., Valavanis, K.: Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert Systems with Applications 36(7), 10,696–10,707 (2009)

[3] Barto, A.G., Sutton, R.S., Anderson, C.W.: Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man, and Cybernetics 13, 834–846 (1983)

[4] Berenji, H.R., Khedkar, P.: Fuzzy rules for guiding reinforcement learning. In: Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 1992), Mallorca, pp. 511–514 (1992a)

[5] Berenji, H.R., Khedkar, P.: Learning and tuning fuzzy logic controllers through reinforcements. IEEE Transactions on Neural Networks 3, 724–740 (1992)

[6] Berenji, H.R., Lea, R.N., Jani, Y., Khedkar, P., Malkani, A., Hoblit, J.: Space shuttle attitude control by reinforcement learning and fuzzy logic. In: Proc. IEEE Int. Conf. on Neural Networks 1993, San Francisco, pp. 1396–1401 (1993)

[7] Berthold, M.: Mixed fuzzy rule formation. Int. J. Approximate Reasoning 32, 67–84 (2003)

[8] Berthold, M., Huber, K.P.: Tolerating missing values in a fuzzy environment. In: Mares, M., Mesiar, R., Novak, V., Ramik, J., Stupnanova, A. (eds.) Proc. of 7th Intl. Fuzzy Systems Association World Congress (IFSA 1997), vol. I, pp. 359–362. Academia, Prague (1997)

[9] Berthold, M., Huber, K.P.: Constructing fuzzy graphs from examples. Int J Intelligent Data Analysis 3(1), 37–57 (1999)

[10] Berthold, M.R., Huber, K.P.: Neural network based construction of fuzzy graphs. In: Proceedings of the Fourth Annual Conference on Fuzzy Theory and Technology, North Carolina, pp. 170–173 (1995) (invited paper)

[11] Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling. STUDFUZZ. Physica-Verlag, Heidelberg (2002)

[12] Dabija, V., Tschichold Gürman, N.: A framework for combining symbolic and connectionist learning with equivalent concept descriptions. In: Proc. 1993 Int. Joint Conf. on Neural Networks (IJCNN 1993), Nagoya (1993)
[13] Dias, P., Figueiredo, K., Vellasco, M., Pacheco, M.A., Barbosa, C.H.: Reinforcement learning hierarchical neuro-fuzzy model for autonomous robots. In: Proc. of 2nd Int. Conf. on Autonomous Robots and Agents (ICARA 2004), Palmerston, New Zealand, pp. 107–112 (2004)
[14] Eklund, P., Klawonn, F.: Neural fuzzy logic programming. IEEE Trans. Neural Networks 3, 815–818 (1992)
[15] Figueiredo, K., Vellasco, M., Pacheco, M., Souza, F.: Reinforcement learning hierarchical neuro-fuzzy politree model for control of autonomous agents. In: Proc. of 4th Int. Conf. on Hybrid Intelligent Systems (HIS 2004), Kitakyushu, Japan, pp. 107–112 (2004)
[16] Figueiredo, K., Campos, L.C.D., Vellasco, M.B.R., Pacheco, M.A.C.: Reinforcement learning-hierarchical neuro-fuzzy politree model for autonomous agents–evaluation in a multi-obstacle environment. In: Proc. of 5th Intl. Conf. on Hybrid Intelligent Systems (HIS 2005), pp. 551–554 (2005)
[17] Gorrini, V., Bersini, H.: Recurrent fuzzy systems. In: Proc. of the 3rd Conference on Fuzzy Systems (FUZZ-IEEE 1994). IEEE, Orlando (1994)
[18] Gupta, M., Rao, D.: On the principles of fuzzy neural networks. Fuzzy Sets and Systems 61, 1–18 (1994)
[19] Halgamuge, S.K.: Advanced methods for fusion of fuzzy systems and neural networks in intelligent data processing, PhD thesis, Technische Hochschule Darmstadt (1995)
[20] Halgamuge, S.K., Glesner, M.: A fuzzy-neural approach for pattern classification with the generation of rules based on supervised learning. In: Proc. Neuro-Nimes 1992, Nanterre, pp. 167–173 (1992)
[21] Halgamuge, S.K., Glesner, M.: Neural networks in designing fuzzy systems for real world applications. Fuzzy Sets and Systems 65, 1–12 (1994)
[22] Hammell II, R.J., Sudkamp, T.: A two level architecture for fuzzy learning. Journal of Intelligent and Fuzzy Systems 3(4), 273–286 (1995)
[23] Jang, J.S.R.: Fuzzy modeling using generalized neural networks and kalman filter algorithm. In: Proc. Ninth National Conf. on Artificial Intelligence (AAAI 1991), pp. 762–767 (1991)
[24] Jang, J.S.R.: Self-learning fuzzy controller based on temporal back-propagation. IEEE Transactions on Neural Networks 3, 714–723 (1992)
[25] Jang, J.S.R.: Anfis: Adaptive-network-based fuzzy inference systems. IEEE Transactions on Systems, Man, and Cybernetics 23, 665–685 (1993)
[26] Jang, J.S.R., Mizutani, E.: Levenberg-marquardt method for anfis learning. In: Proc. Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS 1996, pp. 87–91. IEEE, Berkeley (1996)
[27] Jang, J.S.R., Sun, C.T.: Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Trans. Neural Networks 4, 156–163 (1993)
[28] Keller, J.M.: Experiments on neural network architectures for fuzzy logic. In: Lea, R.N., Villareal, J. (eds.) Proc. Second Joint Technology Workshop on Neural Networks and Fuzzy Logic, NASA, Lyndon B, pp. 201–216. Johnson Space Center, Houston (1991)
[29] Keller, J.M., Hunt, D.J.: Incorporating fuzzy membership function into the perceptron algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 7, 693–699 (1985)

[30] Keller, J.M., Tahani, H.: Backpropagation neural networks for fuzzy logic. Information Sciences 62, 205–221 (1992)
[31] Keller, J.M., Tahani, H.: Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks. Int. J. Approximate Reasoning 6, 221–240 (1992)
[32] Keller, J.M., Yager, R.R., Tahani, H.: Neural network implementation of fuzzy logic. Fuzzy Sets and Systems 45, 1–12 (1992)
[33] Kempf, R., Adamy, J.: Equilibria of recurrent fuzzy systems. Fuzzy Sets and Systems 140(2), 231–257 (2003)
[34] Kuremoto, T., Yamano, Y., Feng, L.-B., Kobayashi, K., Obayashi, M.: A Neuro-fuzzy Network with Reinforcement Learning Algorithms for Swarm Learning. In: Zhang, Y. (ed.) Future Wireless Networks and Information Systems. LNEE, vol. 144, pp. 101–108. Springer, Heidelberg (2012)
[35] Lee, C.H., Teng, C.C.: Indentification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Transactions on Fuzzy Systems 8(4), 349–366 (2000)
[36] Lin, C.M., Hsu, C.F.: Identification of dynamic systems using recurrent fuzzy neural network. In: Proc. of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, pp. 2671–2675 (2001)
[37] Medsker, L.R., Jain, L.C.: Recurrent Neural Networks: Design and Applications. CRC Press (1999)
[38] Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Trans. Neural Netw. Learning Syst. 11(3), 748–768 (2000)
[39] Miyoshi, T., Tano, S., Kato, Y., Arnould, T.: Operator tuning in fuzzy production rules using neural networks. In: Proc. IEEE Int. Conf. on Fuzzy Systems 1993, San Francisco, pp. 641–646 (1993)
[40] Nauck, D.: Adaptive rule weights in neuro-fuzzy systems. Neural Computing & Applications 9(1), 60–70 (2000)
[41] Nauck, D.: Fuzzy data analysis with NEFCLASS. Int. J. Approximate Reasoning 32, 103–130 (2003)
[42] Nauck, D.: Measuring interpretability in rule-based classification systems. In: Proc. IEEE Int. Conf. on Fuzzy Systems 2003, pp. 196–201. IEEE, St. Louis (2003)
[43] Nauck, D.: Neuro-fuzzy learning with symbolic and numeric data. Soft Computing 8(6), 383–396 (2004)
[44] Nauck, D., Kruse, R.: Nefcon-i: An x-window based simulator for neural fuzzy controllers. In: Proc. IEEE Int. Conf. Neural Networks 1994 at IEEE WCCI 1994, Orlando, FL, pp. 1638–1643 (1994)
[45] Nauck, D., Kruse, R.: Nefclass–a neuro-fuzzy approach for the classification of data. In: George, K.M., Carrol, J.H., Deaton, E., Oppenheim, D., Hightower, J. (eds.) Proc. 1995 ACM Symposium on Applied Computing, Applied Computing 1995, Nashville, February 26-28, pp. 461–465. ACM Press, New York (1995)
[46] Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classification rules from data. Fuzzy Sets and Systems 89, 277–288 (1997)
[47] Nauck, D., Kruse, R.: How the learning of rule weights affects the interpretability of fuzzy systems. In: Proc. 7th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 1998), Anchorage, pp. 1235–1240 (1998)
[48] Nauck, D., Kruse, R.: Neuro-fuzzy systems for function approximation. Fuzzy Sets and Systems 101, 261–271 (1999)
[49] Nauck, D., Nürnberger, A.: The evolution of neuro-fuzzy systems. In: Proc. 24th International Conf. of the North American Fuzzy Information Processing Society (NAFIPS 2005), pp. 98–103. IEEE (2005)

[50] Nauck, D., Klawonn, F., Kruse, R.: Foundations of Neuro-Fuzzy Systems. Wiley, Chichester (1997)
[51] Nürnberger, A.: A hierarchical recurrent neuro-fuzzy system. In: Proc. of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, pp. 1407–1412. IEEE (2001)
[52] Nürnberger, A.: Approximation of dynamic systems using recurrent neuro-fuzzy techniques. Soft Computing 8(6), 428–442 (2004)
[53] Nürnberger, A., Nauck, D., Kruse, R.: Neuro-fuzzy control based on the nefcon-model. Soft Computing 2(4), 182–186 (1999)
[54] Pal, S.K., Mitra, S.: Multi-layer perceptron, fuzzy sets and classification. IEEE Trans. Neural Networks 3, 683–697 (1992)
[55] Park, I., Choi, J., Lee, M.J., Lee, S.: Application of an adaptive neuro-fuzzy inference system to ground subsidence hazard mapping. Computers and Geosciences (2012) (online first)
[56] Pedrycz, W.: Neurocomputations in relational systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 289–297 (1991)
[57] Pedrycz, W.: A referential scheme of fuzzy decision making and its neural network structure. IEEE Transactions on Systems, Man, and Cybernetics 21, 1593–1604 (1991)
[58] Pedrycz, W., Kandel, A., Zhang, Y.Q.: Neurofuzzy systems. In: Nguyen, H.T., Sugeno, M. (eds.) Fuzzy Systems Modeling and Control. The Handbooks on Fuzzy Sets ch. 9, pp. 311–380. Kluwer Academic Publishers, Norwell (1998)
[59] Pineda, F.J.: Recurrent backpropagation networks. In: Chauvin, Y., Rumelhart, D.E. (eds.) Backpropagation: Theory, Architectures and Applications, pp. 99–135. Lawrence Erlbaum Associates, Hillsdale (1995)
[60] Riedmiller, M.: Selbständig lernende neuronale Steuerungen. VDI-Verlag GmbH, Düsseldorf (1997)
[61] Sarma, K.K., Mitra, A.: Recurrent fuzzy-neural mimo channel modeling. Journal of Intelligent Systems (2012)
[62] Simpson, P.K.: Fuzzy min-max neural networks–part 1: Classification. IEEE Transactions on Neural Networks 3, 776–786 (1992a)
[63] Simpson, P.K.: Fuzzy min-max neural networks–part 2: Clustering. IEEE Transactions on Fuzzy Systems 1, 32–45 (1992)
[64] Sudkamp, T., Hammell II, R.J.: Scalability in fuzzy rule-based learning. Information Sciences 199, 135–147 (1998)
[65] Surmann, H., Maniadakis, M.: Learning feed-forward and recurrent fuzzy systems: A genetic approach. Journal of Systems Architecture 47(7), 649–662 (2001)
[66] Takagi, H., Hayashi, I.: NN-driven fuzzy reasoning. Int. J. Approximate Reasoning 5, 191–212 (1991)
[67] Taylan, O., Darrab, I.A.: Fuzzy control charts for process quality improvement and product assessment in tip shear carpet industry. Journal of Manufacturing Technology Management 23(3), 402–420 (2012)
[68] Theocharis, J.B., Vachtsevanos, G.: Recursive learning algorithms for training fuzzy recurrent models. International Journal of Intelligence Systems 11(12), 1059–1098 (1996)
[69] Tschichold-Gürman, N.: Generation and improvement of fuzzy classifiers with incremental learning using fuzzy rulenet. In: George, K.M., Carrol, J.H., Deaton, E., Oppenheim, D., Hightower, J. (eds.) Proc. 1995 ACM Symposium on Applied Computing, Applied Computing, Nashville, February 26-28, pp. 466–470. ACM Press, New York (1995)

[70] Tschichold-Gürman, N.: Rulenet–a new knowledge-based artificial neural network model with application examples in robotics. PhD thesis, ETH Zürich, Zürich (1996)
[71] Vatankhah, R., Etemadi, S., Alasty, A., Vossoughi, G.: Adaptive critic-based neuro-fuzzy controller in multi-agents: Distributed behavioral control and path tracking. Neurocomput. 88, 24–35 (2012)
[72] Wang, J.S., Lee, C.S.G.: Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle. IEEE Transactions on Robotics and Automation 19(2), 283–295 (2003)
[73] Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. IEEE Trans Syst., Man, Cybern. 22(6), 1414–1427 (1992)
[74] Wermter, S.: Neural fuzzy preference integration using neural preference moore machines. International Journal of Neural Systems 10(4), 287–310 (2000)
[75] White, D.A., Sofge, D.A. (eds.): Handbook of Intelligent Control. Neural, Fuzzy, and Adaptive Approaches. Van Nostrand Reinhold, New York (1992)
[76] Williams, R.J., Zipser, D.: Experimental analysis of the real time recurrent learning algorithm. Connection Science 1, 87–111 (1989)
[77] Yager, R.R., Filev, D.P.: Adaptive defuzzification for fuzzy logic controllers. BUSEFAL 49, 50–57 (1992)
[78] Yager, R.R., Filev, D.P.: Adaptive defuzzification for fuzzy system modelling. In: Proc. Workshop of the North American Fuzzy Information Processing Society (NAFIPS 1992), pp. 135–142. Puerto Vallarta (1992)
[79] Zhang, C., Qiu, F.: Hyperspectral image classification using an unsupervised neuro-fuzzy system. Journal of Applied Remote Sensing 6(1), 063, 515 (2012)
[80] Zhang, J., Morris, A.J.: Recurrent neuro-fuzzy networks for nonlinear process modelling. IEEE Transactions on Neural Networks 10(2), 313–326 (1999)
[81] Ziaii, M., Ardejani, F.D., Ziaei, M., Soleymani, A.A.: Neuro-fuzzy modeling based genetic algorithms for identification of geochemical anomalies in mining geochemistry. Applied Geochemistry 27(3), 663–676 (2012)