

Chapter 5

Evolving Neuro-Fuzzy Inference Systems

Prof. Nik Kasabov
nkasabov@aut.ac.nz
<http://www.kedri.info>

Overview

- Different types of rules in knowledge based neural networks (KBNN), e.g. ANFIS
- Hybrid fuzzy inference systems (HyFIS)
- Dynamic evolving neuro-fuzzy inference systems (DENFIS)
- Different Types of Fuzzy Rules in ECOS
- Type-2 Evolving Connectionist Systems
- Interval-based Evolving Connectionist Systems and other Ways of Defining Receptive Fields.

Introduction

- Fuzzy neural networks are connectionist models that are trained as neural networks, but their structure can be interpreted as a set of fuzzy rules.
- Neuro-fuzzy inference systems consist of a set of rules and an inference method that are embodied or combined with a connectionist structure for a better adaptation.
- Evolving neuro-fuzzy inference systems are such systems, where both the knowledge and the inference mechanism evolve, change in time, with more examples presented to the system.
- In these models, knowledge is represented as both fuzzy *rules* and statistical features that are learned in an on-line, life long learning mode.

Rules and Inference in KBNNs

- Knowledge (e.g. rules) is the essence of what a knowledge-based neural network (KBNN) has learned during its operation.
- Manipulating rules in a KBNN can pursue the following objectives:
 - » Knowledge discovery, i.e. understanding and explanation of the data used to train the KBNN. The extracted rules can be analysed either by an expert, or by the system itself.
 - » Improvement of the KBNN system, e.g. maintaining an optimal size of the KBNN that is adequate to the expected accuracy of the system.

Types of Rules used in KBNN

- (1) Simple propositional rules (e.g., IF x_1 is A AND/OR x_2 is B THEN y is C, where A, B and C are constants, variables, or symbols of true/false type) (see for example, Feigenbaum, 1989; Gallant, 1993; Hendler and Dickens, 1991). As a partial case, interval rules can be used, for example:
IF x_1 is in the interval $[x_{1min}, x_{1max}]$ AND x_2 is in the interval $[x_{2min}, x_{2max}]$ THEN y is in the interval $[ymin, ymax]$, with $Nr1$ examples associated with this rule.
- (2) Propositional rules with certainty factors (e.g., IF x_1 is A (CF1) AND x_2 is B (CF2) THEN y is C (CFc)), (see for example Fu, 1989).
- (3) Zadeh-Mamdani fuzzy rules (e.g., IF x_1 is A AND x_2 is B THEN y is C, where A, B and C are fuzzy values represented by their membership functions) (see for example Zadeh, 1965; Mamdani, 1977).
- (4) Takagi-Sugeno fuzzy rules (for example, the following rule is a first order rule: IF x_1 is A AND x_2 is B THEN y is $a.x_1 + b.x_2 + c$, where A and B are fuzzy values and a, b and c are constants) (Takagi and Sugeno, 1985; Jang, 1993). More complex functions are possible to use in higher-order rules.

Types of Rules used in KBNN

- (5) Fuzzy rules of type (3) with degrees of importance and certainty degrees (e.g.; IF x_1 is A (DI_1) AND x_2 is B (DI_2) THEN y is C (CF_c), where DI_1 and DI_2 represent the importance of each of the condition elements for the rule output, and the CF_c represents the strength of this rule) (see Kasabov, 1996).
- (6) Fuzzy rules that represent associations of clusters of data from the problem space (e.g., Rule j : IF [an input vector x is in the input cluster defined by its centre (x_1 is A_j , to a membership degree of MD_{1j} , AND x_2 is B_j , to a membership degree of MD_{2j}) and by its radius R_{j-in}] THEN [y is in the output cluster defined by its centre (y is C, to a membership degree of MD_c) and by its radius R_{j-out} , with $N_{ex}(j)$ examples represented by this rule]. These are the EFuNN rules discussed in Chapter 3.
- (7) Temporal rules (e.g., IF x_1 is present at a time moment t_1 (with a certainty degree and/or importance factor of DI_1) AND x_2 is present at a time moment t_2 (with a certainty degree/importance factor DI_2) THEN y is C (CF_c)).

Types of Rules used in KBNN

- (8) Temporal, recurrent rules (e.g., IF x_1 is A (DI1) AND x_2 is B (DI2) AND y at the time moment $(t-k)$ is C THEN y at a time moment $(t+n)$ is D (CFc)).
- (9) Type-2 fuzzy rules, that are fuzzy rules of the form of: IF x is A^{\sim} AND y is B^{\sim} THEN z is C^{\sim} , where A^{\sim} , B^{\sim} , and C^{\sim} are type-2 fuzzy membership functions (see the extended glossary, and also the section in this chapter on type-2 ECOS).

Generic Methods for Rule Extraction from KBNN

- Rule extraction through activating a trained KBNN on input data and observing the patterns of activation (the short-term memory). Not practical for on-line, incremental learning as past data may not be available for a consecutive activation of the trained KBNN.
- Rule extraction through analysis of the connections in a trained KBNN (the long-term memory). Allows for extracting knowledge without necessarily activating the connectionist system again on input data. It is appropriate for on-line learning and system improvement.
- Hybrid of the first two methods

Methods for Inference Over Rules from KBNN

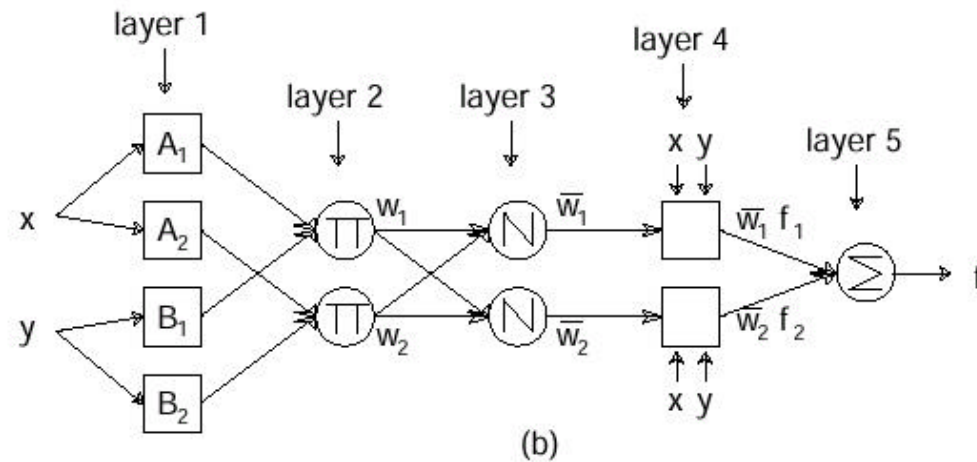
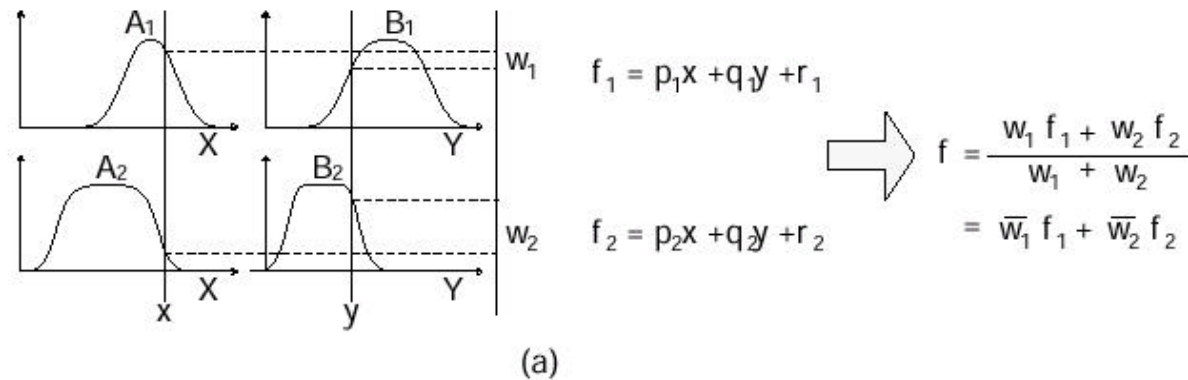
There are three types of methods used in the KBNN to infer new information :

1. The rule learning and rule inference modules constitute an integral structure where reasoning is part of the rule learning, and vice versa (used in most fuzzy neural networks and neuro-fuzzy inference systems).
2. The rules extracted from a KBNN are interpreted in another inference machine. The learning module is separated from the reasoning module (used in many expert systems)
3. The two options from above are possible within one intelligent system.

ANFIS (Adaptive Neuro-Fuzzy Inference Systems)

- ANFIS (Jang,1993) implements Takagi-Sugeno fuzzy rules in a five layer MLP network.
 - » The first layer represents fuzzy membership functions.
 - » The second and the third layers contains nodes that form the antecedent parts in each rule.
 - » The fourth layer calculates the first-order Takagi-Sugeno rules for each fuzzy rule.
 - » The fifth layer – the output layer, calculates the weighted global output of the system.
- Backpropagation is used to modify the initially chosen membership functions and the least mean square algorithm determines the coefficients of the linear output functions.

ANFIS (Adaptive Neuro-Fuzzy Inference Systems)



a) An exemplar set of two fuzzy rules and the inference over them performed in an ANFIS structure

b) The exemplar ANFIS structure for these two rules (Fig 5.2)

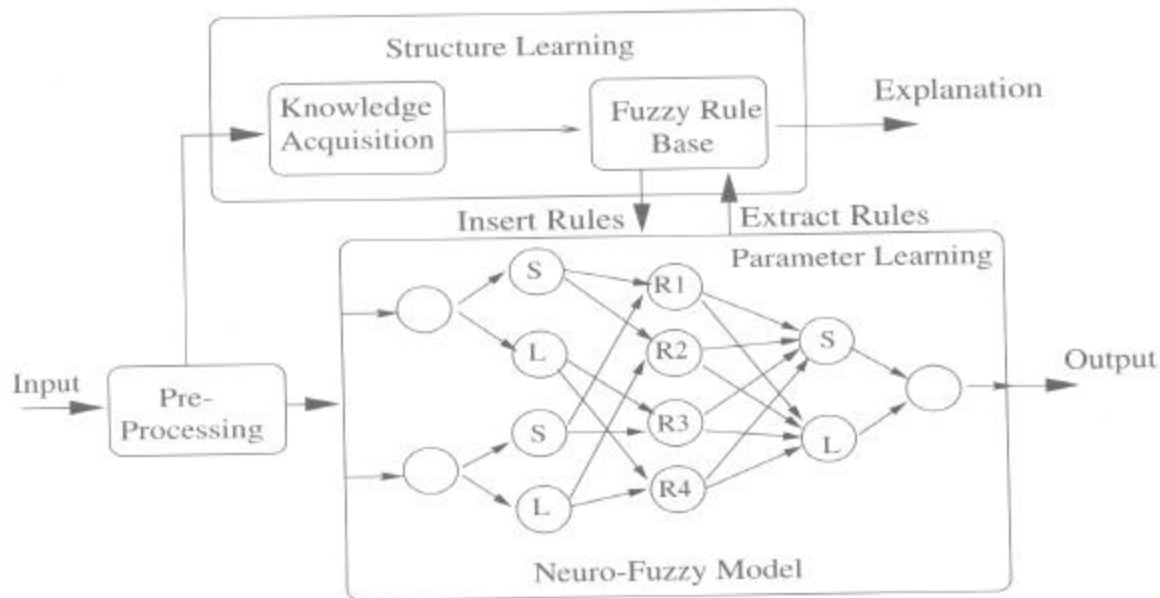
ANFIS (Adaptive Neuro-Fuzzy Inference Systems)

- Uses a multiple iteration learning procedure and a hybrid learning algorithm => fast convergence
- Cannot handle high dimensionality cases, eg more than 10 variables
- Fixed structure, cannot adapt to incoming data => limited abilities for incremental on-line learning.
- Only one output => suitable for prediction or non-linear function approximation tasks
- In contrast, on-line learning and local optimisation in a fuzzy-neural inference system allow to trace the process of knowledge emergence, for analysing how rules change over time.

Hybrid Neuro-Fuzzy Inference Systems (HyFIS)

- HyFIS (Kim and Kasabov, 1999, “Neural Networks”) consists of two main parts:
 1. A fuzzy analysis module for fuzzy rule extraction from incoming data with the use of Wang’s method (1994)
 2. A connectionist module that implements and tunes the fuzzy rules through applying the backpropagation algorithm.

Hybrid Neuro-Fuzzy Inference Systems (HyFIS)



A schematic block diagram of HyFIS (Fig. 5.3)

Hybrid Neuro-Fuzzy Inference Systems (HyFIS)

The system operates in the following mode:

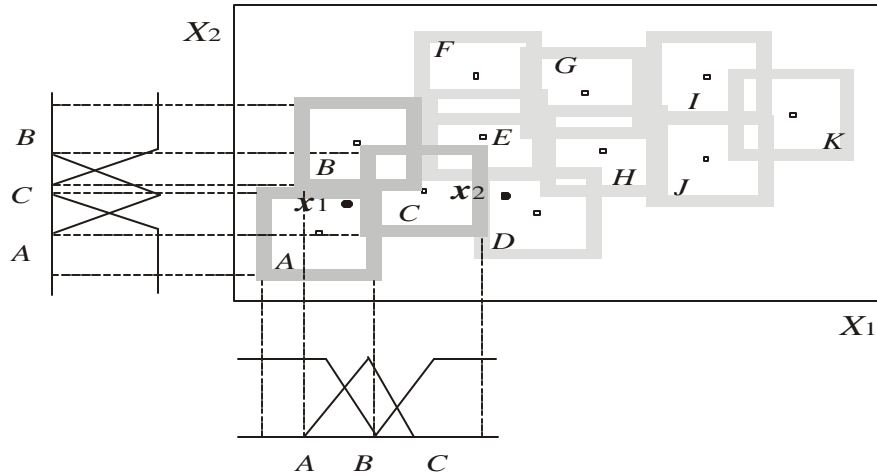
- Data examples (\mathbf{x} , \mathbf{y}) are assumed to arrive in chunks of m (as a partial case, $m=1$).
 - For the current chunk K_i , consisting of m_i examples, n_i fuzzy rules are extracted as described below. They have a form illustrated with the following example:
IF x_1 is Small AND x_2 is Large THEN y is Medium (certainty 0.7)
3. The n_i fuzzy rules are inserted in the neuro-fuzzy module, thus updating the current structure of this module
 4. The updated neuro-fuzzy structure is trained with the backpropagation algorithm on the chunk of data K_i , or on a larger data set if such is available.
 5. New data \mathbf{x}' that do not have known output vectors, are propagated through the neuro-fuzzy module for recall.

Dynamic Evolving Neuro-Fuzzy Inference Systems (DENFIS)

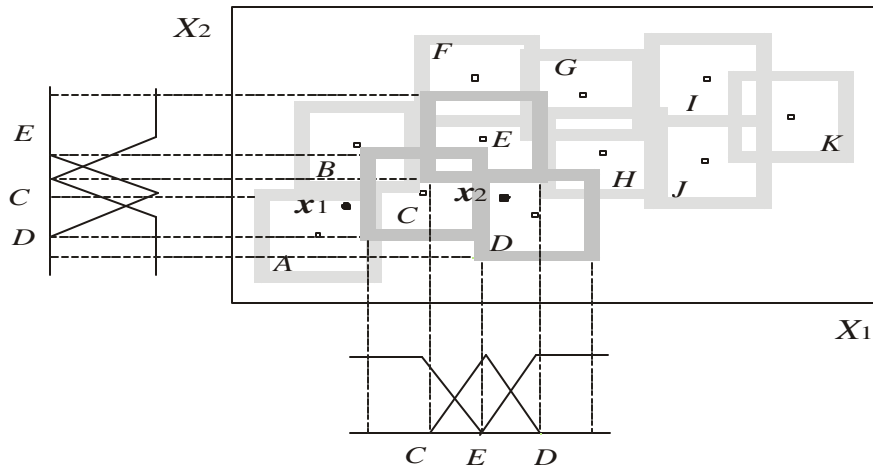
- DENFIS, in its two modifications - for on-line, and for off-line learning, use Takagi-Sugeno type of fuzzy inference method (Kasabov and Song, IEEE Tr.Fuzzy Systems, April, 2002)
- In the on-line model of DENFIS, the linear functions in the consequence parts are created and updated through learning from data by using the linear least-square estimator.
- The Takagi-Sugeno fuzzy inference system is a dynamic inference system. Advantages over other inference systems:
 - » For each input vector, the DENFIS model chooses m fuzzy rules from the whole fuzzy rule set for forming a current inference system
 - » Second, depending on the position of the current input vector in the input space, the antecedents of the fuzzy rules chosen to form an inference system for this input vector may vary

DENFIS

(a) Fuzzy rule group 1 for a DENFIS



(b) Fuzzy rule group 2 for a DENFIS



Two fuzzy rule groups are formed by DENFIS to perform inference for an input vector (Fig 5.7)

DENFIS Off-Line learning model

- On-line model can also be used for off-line batch mode training, but not efficient when used on relatively small data sets.
- The DENFIS off-line learning process is implemented in the following way:
 - » cluster (partition) the input space to find n cluster centres by using off-line evolving clustering method with constrained optimisation (ECMc)
 - » create the antecedent part for each fuzzy rule and also the current position of the cluster centre (rule node)
 - » find n data sets, each of them including one cluster centre and p learning data pairs that are closest to the centre in the input space.

Depending on the DENFIS off-line learning model

- » Estimate the functions f to create the consequent part for each fuzzy rule with n data sets;

OR

- » Each consequent function f of a fuzzy rule is learned by a corresponding MLP network after training it on the corresponding data set with the use of the back propagation algorithm.

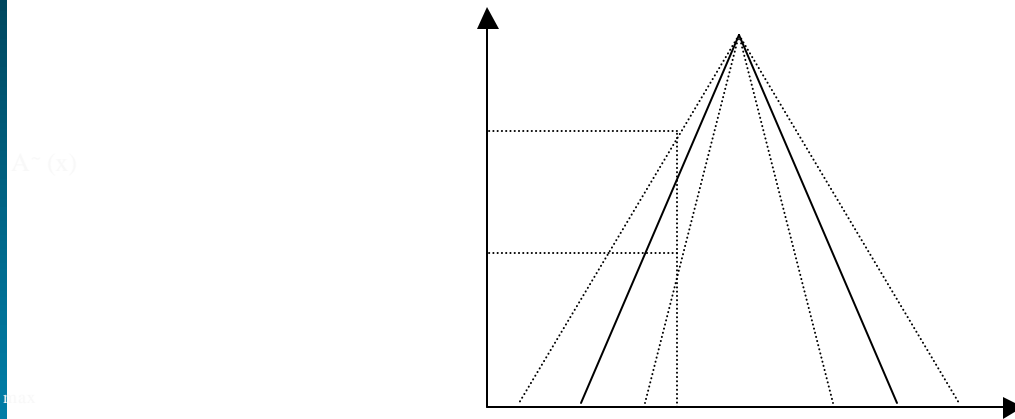
Types of Fuzzy Rules in ECOS

- Using different type of rules in an ECOS architecture, may lead to different results depending on the task in hand.
- ECOS allow for both fuzzy and propositional (e.g. interval) rules to be used depending on if there is a fuzzy membership layer or not.
- Some Types:
 - » Takagi-Sugeno – better for function approximation and time series prediction (e.g. DENFIS)
 - » Zadeh-Mamdani – better for classification tasks (e.g. EFuNN)

Type-2 ECOS

- Extension to the ECOS paradigm to use a higher order representation, suitable when:
 - » Data is time varying (e.g. changing dynamics of a chaotic process)
 - » Noise is non-stationary
 - » Features are non-stationary (they change over time)
 - » Dealing with inexact human knowledge that change over time and varies across humans
- Some principles from the theory of type-2 fuzzy systems can be used
- Type-2 fuzzy sets are sets to which elements belong with a membership degree that is represented not by a single number but by an interval of min-max membership degrees.

Type-2 ECOS



An exemplar of a type-2 triangular fuzzy membership function (MF) of a variable x (Fig 5.10)

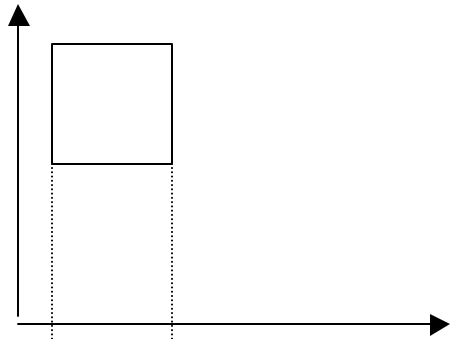
Interval-based ECOS

- Most of the ECOS methods in this part of the book used hyper-spheres to define a receptive field of a rule node r
- Another method is to use a hyper-rectangular receptive field with a minimum and a maximum interval values for each rule node and for each variable derived through the evolving process
- Using intervals and hyper-rectangular receptive fields allows for a better partitioning of the problem space and in many cases leads to better classification and prediction results.

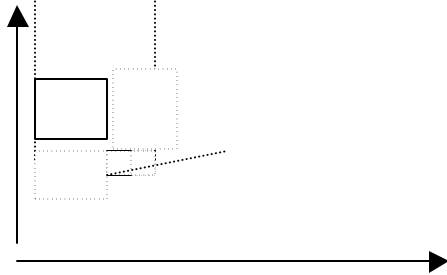
Interval-based Receptive Fields

- Divide and Split receptive Fields
 - » On-line modification of the receptive fields is crucial for a successful on-line learning.
 - » Sometimes a receptive field is created for a rule node, but within this receptive field there is a new example that belongs to a different class.
 - » In this case the new example will be assigned a new node that divides the previous receptive field into several parts.
 - » Each of these parts will be assigned new nodes that will have the same class label as the “mother node”.
 - » This approach is very efficient when applied for on-line classification in complex multi-class distribution spaces.

Interval-based Receptive Fields



a) An existing receptive field of a rule node in a 2D input space; b) a new class example divides the receptive field of the existing node (Fig 5.12)



Summary

- Evolving neuro-fuzzy inference systems combine both on-line learning from data, rule insertion, rule extraction, and inference over these rules.
- ANFIS is not flexible in terms of changing the number of membership functions and rules over time
- HyFIS, DENFIS and EFuNN are truly on-line knowledge-based learning systems.
- HyFIS and EFuNN use Zadeh-Mamdani simple fuzzy rules, DENFIS uses Takagi-Sugeno first order fuzzy rules.
- Each of the above systems has its strengths and weaknesses while applied on different tasks.

Further Readings

- Fuzzy sets and systems (Zadeh, 1965; Terano et al, 1992; Dubois and Prade, 1980, 1988; Kerr, 1991; Bezdek, 1987; Sugeno, 1985).
- Zadeh-Mamdani fuzzy rules (Zadeh, 1965; Mamdani, 1977).
- Takagi-Sugeno fuzzy rules (Takagi and Sugeno, 1985; Jang, 1995).
- Type-2 fuzzy rules and fuzzy inference systems (Mendel,2001).
- Neuro-fuzzy inference systems and fuzzy neural networks (Yamakawa et al, 1992; Furuhashi, 1993; Hauptman and Heesche, 1995; Lin and Lee, 1996; Kasabov, 1996; Feldcamp, 1992; Gupta, 1992; Gupta and Rao, 1994).
- ANFIS (Jang, 1993).
- HyFIS (Kim and Kasabov, 1999).
- DENFIS (Kasabov and Qun, 2001).
- Rule extraction from neuro-fuzzy systems (Hayashi, 1991; Mitra and Hayashi, 2001; Duch et al, 1997).
- -Neuro-fuzzy systems as universal approximators (Hayashi, 1991).