

Mamdani Fuzzy Networks with Feedforward Rule Bases for Complex Systems Modelling

Alexander Gegov*, David Sanders^, Boriana Vatchova+

*University of Portsmouth, School of Computing

Buckingham Building, Portsmouth PO1 3HE, United Kingdom

^University of Portsmouth, School of Engineering

Anglesea Building, Portsmouth PO1 3DJ, United Kingdom

+Bulgarian Academy of Sciences, Institute of Information and Communication Technologies

Block 2, 1113 Sofia, Bulgaria

Abstract: This paper proposes a novel approach for modelling complex interconnected systems by means of Mamdani fuzzy networks with feedforward rule bases. The nodes in these networks are rule bases connected in a feedforward manner whereby outputs from some rule bases are fed as inputs to subsequent rule bases. The approach allows any fuzzy network of this type to be presented as an equivalent Mamdani fuzzy system by linguistic composition of its nodes. The composition process makes use of formal models for fuzzy networks, basic operations in such networks, their properties and advanced operations. These models, operations and properties are used for defining several types of networks with single or multiple horizontal levels and vertical layers. The proposed approach facilitates the understanding of complex interconnected systems by improving the transparency of their models.

Keywords: fuzzy modelling, decision support systems, finance, linguistic modelling, feedforward connections, complex systems.

1. Introduction

Complexity is a versatile feature of existing systems that cannot be described by a single definition. In this context, complexity is usually associated with a number of attributes such

as uncertainty, dimensionality and structure, which make the modelling of systems with these attributes more difficult. Therefore, the complexity of a given system can be accounted for by identifying the complexity related attributes that are to be found in this system.

Fuzzy logic has proved itself as a powerful tool for dealing with uncertainty as an attribute of systemic complexity. In this context, fuzziness is quite suitable for reflecting non-probabilistic uncertainty such as imprecision, incompleteness and ambiguity [1-3].

More recently, fuzzy logic has also been made more effective in dealing with dimensionality as a systemic complexity attribute by means of rule base reduction and compression. Dimensionality in rule base reduction is associated with the number of rules, which is an exponential function of the number of system inputs and the number of linguistic terms per input [4-7]. In rule base compression, dimensionality is associated with the amount of on-line operations required during fuzzification, inference and defuzzification [8].

However, as far as structure is concerned, fuzzy logic is still unable to reflect adequately any interacting modules within a modelled process. This is due to the black-box nature of fuzzy models that cannot take into account explicitly any interactions among sub-processes [9-12]. In this respect, the following paragraphs discuss some of the main approaches in fuzzy modelling and their ability to deal with structure as a systemic complexity attribute.

The most common type of fuzzy system is with a single rule base [13-15]. This type of system is usually referred to as Standard Fuzzy System (SFS). The latter is characterised by a black-box nature whereby the inputs are mapped directly to the outputs without the consideration of any internal connections. The operation of SFS is based on a single Fuzzification-Inference-Defuzzification (FID) sequence and it is usually quite accurate for output modelling as it reflects the simultaneous influence of all inputs on the output. However, the efficiency and transparency of SFS deteriorate with the increase of the number

of rules. Therefore, as the number of rules increases, it not only takes longer to simulate the model output but it is also less clear how this output is affected by the model inputs.

Another type of fuzzy system is with multiple rule bases [16-19]. This type of system is often described by cascaded rule bases and it is referred to as Chained Fuzzy System (CFS) [20-22] or Hierarchical Fuzzy System (HFS) [23-27]. Both CFS and HFS are characterised by a white-box nature whereby the inputs are mapped to the outputs by means of some internal variables in the form of connections. The operation of CFS and HFS is based on multiple FID sequences whereby each connection links the FID sequences for two adjacent rule bases.

CFS is usually used as a detailed presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Also, efficiency is improved because of the smaller number of inputs to the individual rule bases. However, accuracy may be lost due to the accumulation of errors as a result of the multiple FID sequences.

HFS is often used as an alternative presentation of SFS for the purpose of improving transparency by explicitly taking into account all subsystems and the interactions among them. Efficiency is also improved by the reduction of the overall number of rules, which is a linear function of the number of inputs to the subsystems and the number of linguistic terms per input. However, these improvements are at the expense of accuracy due to the accumulation of errors as a result of the multiple FID sequences.

A third type of fuzzy system is with networked rule bases. This type of system is referred to as Networked Fuzzy System (NFS) and a basic theoretical framework for it has been introduced recently in [28]. NFS is characterised by a white-box nature whereby the inputs are mapped to the outputs by means of connections. Subsystems in NFS are represented by nodes and the interactions among subsystems are the connections among these nodes. NFS is

a hybrid between SFS and CFS/HFS. On one hand, the structure of NFS is similar to the structure of CFS/HFS due to the explicit presentation of subsystems and the interactions among them. On the other hand, the operation of NFS resembles the operation of SFS as the multiple rule bases are simplified to a linguistically equivalent single rule base. This simplification is based on the linguistic composition approach that is the main focus of this work. As a hybrid concept, NFS has the potential of combining the advantages of SFS and CFS/HFS.

Properties of fuzzy systems such as accuracy, efficiency and transparency are directly related to attributes of systemic complexity such as uncertainty, dimensionality and structure. In this respect, uncertainty is an obstacle to accuracy as it is harder to build an accurate model from uncertain data [29-32]. Furthermore, dimensionality represents an obstacle to efficiency because it is more difficult to reduce the amount of computations in a FID sequence for a large number of rules [33-36]. Finally, structure is an obstacle to transparency as it is harder to understand the behaviour of a black-box model that doesn't reflect the interactions among subsystems [37-40].

This paper introduces an advanced theoretical framework for NFS as an extension of the existing basic one and validates NFS as a modelling tool for complex systems with respect to SFS and CFS/HFS. It builds on some recent methods for evaluating transparency of interaction between rules in fuzzy models [41-42]. It also builds on some other methods for using algebraic structures to evaluate properties of conventional graphs and networks [43-46].

For clarity and simplicity, NFS is referred to as Fuzzy Network (FN). The paper addresses mainly structure as a systemic complexity attribute and the associated property of transparency. The main reason for this choice is that transparency has always been given less attention in complex systems modelling as opposed to accuracy and efficiency. Besides this,

transparency has recently turned out to be not less important for complex systems modelling than accuracy and efficiency.

The remaining part of this paper is structured as follows. Section 2 introduces formal models for fuzzy networks. Sections 3-5 present basic operations in fuzzy networks, their properties and advanced operations. Section 6 discusses several types of feedforward fuzzy networks. Section 7 illustrates the proposed approach for a decision support system case study and evaluates it in a comparative context. Section 8 summarises the main advantages of the approach and highlights future research directions.

2. Formal Models for Fuzzy Networks

A fuzzy system with r rules, m inputs $x_1 \dots x_m$ taking linguistic terms from the input sets $\{A_{11}, \dots, A_{1r}\}, \dots, \{A_{m1}, \dots, A_{mr}\}$ and n outputs $y_1 \dots y_n$ taking linguistic terms from the output sets $\{B_{11}, \dots, B_{1r}\}, \dots, \{B_{n1}, \dots, B_{nr}\}$ can be described by the following if-then rules

$$\text{Rule 1: If } x_1 \text{ is } A_{11} \text{ and } \dots \text{ and } x_m \text{ is } A_{m1}, \text{ then } y_1 \text{ is } B_{11} \text{ and } \dots \text{ and } y_n \text{ is } B_{n1} \quad (1)$$

.....

$$\text{Rule } r: \text{ If } x_1 \text{ is } A_{1r} \text{ and } \dots \text{ and } x_m \text{ is } A_{mr}, \text{ then } y_1 \text{ is } B_{1r} \text{ and } \dots \text{ and } y_n \text{ is } B_{nr}$$

As a fuzzy network represents an extension of a fuzzy system, i.e. it can be viewed as a system of fuzzy systems or a network whose nodes are fuzzy systems, some of the general formal models for fuzzy systems can be used also for fuzzy networks. However, other formal models that are specific to fuzzy networks are required for the simplification of a fuzzy network to a linguistically equivalent fuzzy system. Most of these formal models contain compressed information about nodes in fuzzy networks and they are discussed further below.

If-then rules as the ones from Equation (1) are established formal models for fuzzy systems that can represent nodes in a FN without the connections. They are used here as a bridge between fuzzy systems and FNs. For example, a FN with four nodes $N_{11}, N_{12}, N_{21}, N_{22}$ can be described by the if-then rules given in Equations (2)-(13).

Rule 1 for N_{11} : If x_{11} is small, then y_{11} is low (2)

Rule 2 for N_{11} : If x_{11} is medium, then y_{11} is high (3)

Rule 3 for N_{11} : If x_{11} is big, then y_{11} is average (4)

Rule 1 for N_{12} : If x_{12} is low, then y_{12} is moderate (5)

Rule 2 for N_{12} : If x_{12} is average, then y_{12} is heavy (6)

Rule 3 for N_{12} : If x_{12} is high, then y_{12} is light (7)

Rule 1 for N_{21} : If x_{21} is small, then y_{21} is average (8)

Rule 2 for N_{21} : If x_{21} is medium, then y_{21} is low (9)

Rule 3 for N_{21} : If x_{21} is big, then y_{21} is high (10)

Rule 1 for N_{22} : If x_{22} is low, then y_{22} is heavy (11)

Rule 2 for N_{22} : If x_{22} is average, then y_{22} is light (12)

Rule 3 for N_{22} : If x_{22} is high, then y_{22} is moderate (13)

For compactness, the linguistic terms of the inputs and the outputs for the four nodes above can also be represented by positive integers. In this case, ‘small’, ‘low’ and ‘light’ are represented by ‘1’, the linguistic terms ‘medium’, ‘average’ and ‘moderate’ are represented by ‘2’ whereas the linguistic terms ‘big’, ‘high’ and ‘heavy’ are represented by ‘3’.

If-then rules as the ones presented above are very suitable for formal modelling of fuzzy systems with a single rule base such as SFSs. However, they are not quite suitable for formal modelling of fuzzy systems with multiple or networked rule bases. This is due to the fact that if-then rules can not take into account any connections among nodes in networked rule bases. Also, if-then rules do not lend themselves easily to manipulation for the purpose of simplifying networked rule bases to a linguistically equivalent single rule base using the linguistic composition approach.

Boolean matrices are novel formal models for fuzzy systems that can represent nodes in a FN. Similarly to if-then rules, these models can represent nodes without the connections.

A Boolean matrix compresses the information from a rule base that is represented by a node. In this case, the row and column labels of the Boolean matrix are all possible permutations of the positive integers representing the linguistic terms of the inputs and the outputs for this rule base. The elements of the Boolean matrix are either zeros or ones whereby each one reflects a rule from the rule base.

The if-then rules for the fuzzy network nodes N_{11} , N_{12} , N_{21} , N_{22} from Equations (2)-(13) can be described by the Boolean matrices in Equations (14)-(17).

$$\begin{array}{r}
 N_{11}: \quad y_{11} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{11} \\
 \quad \quad 1 \quad \quad 1 \quad 0 \quad 0 \\
 \quad \quad 2 \quad \quad 0 \quad 0 \quad 1 \\
 \quad \quad 3 \quad \quad 0 \quad 1 \quad 0
 \end{array} \tag{14}$$

$$\begin{array}{r}
 N_{12}: \quad y_{12} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{12} \\
 \quad \quad 1 \quad \quad 0 \quad 1 \quad 0 \\
 \quad \quad 2 \quad \quad 0 \quad 0 \quad 1 \\
 \quad \quad 3 \quad \quad 1 \quad 0 \quad 0
 \end{array} \tag{15}$$

$$\begin{array}{r}
 N_{21}: \quad y_{21} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{21} \\
 \quad \quad 1 \quad \quad 0 \quad 1 \quad 0 \\
 \quad \quad 2 \quad \quad 1 \quad 0 \quad 0 \\
 \quad \quad 3 \quad \quad 0 \quad 0 \quad 1
 \end{array} \tag{16}$$

$$\begin{array}{r}
 N_{22}: \quad y_{22} \quad 1 \quad 2 \quad 3 \\
 \quad \quad x_{22} \\
 \quad \quad 1 \quad \quad 0 \quad 0 \quad 1 \\
 \quad \quad 2 \quad \quad 1 \quad 0 \quad 0 \\
 \quad \quad 3 \quad \quad 0 \quad 1 \quad 0
 \end{array} \tag{17}$$

Boolean matrices as the ones presented above are very suitable for formal modelling of fuzzy systems with multiple or networked rule bases. In particular, they are well suited for

formal modelling of FNs at a lower level of abstraction whereby detailed input-output mappings are specified for isolated individual nodes. Besides this, Boolean matrices work well with other formal models which can take into account connections among nodes in FNs.

Location and connection structures are other novel formal models that are like compressed images of a FN. These models describe the location of nodes and the connections among them, respectively. For example, the four nodes N_{11} , N_{12} , N_{21} , N_{22} from Equations (2)-(13) can be described by the location structure with two levels and two layers in Equation (18).

$$\begin{array}{rcc}
 & \textit{Layer 1} & \textit{Layer 2} & \\
 \textit{Level 1} & N_{11}(x_{11}, y_{11}) & N_{12}(x_{12}, y_{12}) & \\
 \textit{Level 2} & N_{21}(x_{21}, y_{21}) & N_{22}(x_{22}, y_{22}) &
 \end{array} \tag{18}$$

The location structure above is a formal model for a FN with a node set $\{N_{11}, N_{12}, N_{21}, N_{22}\}$, an input set $\{x_{11}, x_{12}, x_{21}, x_{22}\}$ and an output set $\{y_{11}, y_{12}, y_{21}, y_{22}\}$. This structure specifies the location of nodes as well as their inputs and outputs.

If the nodes N_{11} , N_{12} , N_{21} , N_{22} from Equations (2)-(13) are connected, their connections can be described by the connection set $\{z_{11,12}, z_{21,22}\}$. In this case, the first connection is identical with the output from N_{11} and the input to N_{12} whereas the second connection is identical with the output from N_{21} and the input to N_{22} . These connections can be described by the connection structure with two levels and one layer in Equation (19).

$$\begin{array}{rcc}
 & \textit{Layer 1} & \\
 \textit{Level 1} & z_{11,12}=y_{11}=x_{12} & \\
 \textit{Level 2} & z_{21,22}=y_{21}=x_{22} &
 \end{array} \tag{19}$$

Location and connection structures as the ones presented above are also quite suitable for formal modelling of fuzzy systems with multiple or networked rule bases. In particular, these

structures are well suited for formal modelling of FNs at a higher level of abstraction whereby only locations, inputs, outputs and connections for individual nodes are specified.

Location and connection structures describe FNs at overall network level. They work well with Boolean matrices which describe FNs at individual node level. However, these structures do not lend themselves easily to manipulation for the purpose of simplifying networked rule bases to a linguistically equivalent single rule base using the linguistic composition approach.

Block schemes and topological expressions are also novel formal models that are like compressed images of a FN. Similarly to location and connection structures, these models describe the location of nodes and the connections among them. In this case, the subscripts of each node specify its location in the network whereby the first subscript gives the level number and the second subscript gives the layer number. Besides this, block schemes and topological expressions specify all inputs, outputs and connections with respect to the nodes. For example, the four-node FN from Equations (2)-(13) and Equations (18)-(19) can be described by the block scheme in Figure 1.

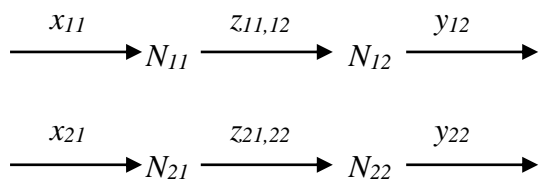


Figure 1: Block scheme for a four-node fuzzy network

The arrows in the block scheme above designate the input set $\{x_{11}, x_{21}\}$ for the nodes in the first layer and the output set $\{y_{12}, y_{22}\}$ for the nodes in the second layer. Also, the arrows designate the connection set $\{z_{11,12}, z_{21,22}\}$ for connected pairs of nodes whereby for each pair of nodes the first node is in the first layer and the second node is in the second layer.

The FN from the four-node FN from Equations (2)-(13) and Equations (18)-(19) can also be described by the topological expression in Equation (20).

$$\{[N_{11}](x_{11} / z_{11,12}) H [N_{12}](z_{11,12} / y_{12})\} V \{[N_{21}](x_{21} / z_{21,22}) H [N_{22}](z_{21,22} / y_{22})\} \quad (20)$$

Each node in the topological expression above is placed within a pair of square brackets ‘[]’. The inputs and the outputs for each node are placed within a pair of simple brackets ‘()’ right after the node. In this case, the inputs are separated from the outputs by a vertical slash ‘/’. Nodes in sequence are designated by the symbol ‘H’ for horizontal relative location whereas nodes in parallel are designated by the symbol ‘V’ for vertical relative location. In this case, the higher priority of horizontal relative location with respect to vertical relative location in Equation (20) is specified by pairs of curly brackets ‘{ }’.

Block schemes and topological expressions as the ones presented above are very suitable for formal modelling of fuzzy systems with multiple or networked rule bases. In particular, they are well suited for formal modelling of FNs at a higher level of abstraction whereby only inputs, outputs and connections for individual nodes are specified.

Like location and connection structures, block schemes and topological expressions describe FNs at overall network level. They work well with Boolean matrices which describe FNs at individual node level. Besides this, block schemes and topological expressions lend themselves easily to manipulation for the purpose of simplifying networked rule bases to a linguistically equivalent single rule base using the linguistic composition approach.

This work focuses on Boolean matrices for formal modelling of nodes. As far as formal modelling of connections is concerned, the focus is on block schemes and topological expressions. The choice of these formal models is justified by their better suitability for the use of the linguistic composition approach in comparison to if-then rules, location and connection structures.

3. Basic Operations in Fuzzy Networks

The process of simplifying networked rule bases to a linguistically equivalent single rule base is central to the linguistic composition approach used in this work. This approach is based on three basic operations on nodes – horizontal, vertical and output merging. These operations are binary in that they can be applied to a pair of nodes, i.e. they are like elementary building blocks in the process of simplifying a FN to a fuzzy system. For simplicity, all basic operations are illustrated with examples of nodes with scalar inputs, outputs and connections but their extension to the vector case is straightforward.

Horizontal merging is a binary operation that can be applied to a pair of sequential nodes, i.e. nodes located in the same level of a FN. This operation merges the operand nodes from the pair into a single product node. The operation can be applied when the output from the first node is fed forward as an input to the second node in the form of an intermediate variable. In this case, the product node has the same input as the input to the first operand node and the same output as the output from the second operand node whereas the connection does not appear in the product node.

When Boolean matrices are used as formal models for the operand nodes, the horizontal merging operation is identical with Boolean matrix multiplication. The latter is similar to conventional matrix multiplication whereby each arithmetic multiplication is replaced by a ‘minimum’ operation and each arithmetic addition is replaced by a ‘maximum’ operation. In this case, the row labels of the product matrix are the same as the row labels of the first operand matrix whereas the column labels of the product matrix are the same as the column labels of the second operand matrix.

Example 1:

This example considers the sequential operand nodes N_{11} and N_{12} located in the first level of the four-node FN from Figure 1. These nodes are described there by the Boolean matrices

in Equations (14)-(15). The connections among these nodes are given by the connection structure in Equation (19). In this context, nodes N_{11} and N_{12} represent a two-node FN that is a subnetwork of the four-node FN. This two-node FN can be described by the block-scheme in Figure 2 and the topological expression in Equation (21).

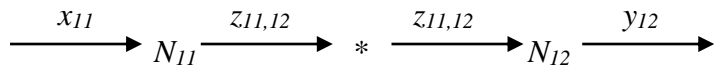


Figure 2: Two-node fuzzy network with operand nodes N_{11} and N_{12}

$$[N_{11}] (x_{11} / z_{11,12}) * [N_{12}] (z_{11,12} / y_{12}) \quad (21)$$

The use of the symbol ‘*’ in Figure 2 and Equation (21) implies that the horizontal merging operation can be applied to the operand nodes N_{11} and N_{12} . In this context, the use of the symbol ‘*’ makes valid the precondition for horizontal merging of nodes N_{11} and N_{12} .

The horizontal merging of the operand nodes N_{11} and N_{12} results into a single product node N_{11*12} which represents a simplified image of the two-node FN in the form of a one-node FN. The latter can be described by the block scheme in Figure 3 and the topological expression in Equation (22).

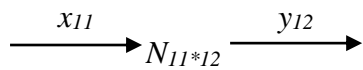


Figure 3: One-node fuzzy network with product node N_{11*12}

$$[N_{11*12}] (x_{11} / y_{12}) \quad (22)$$

The use of the symbol ‘*’ in Figure 3 and Equation (22) implies that the application of the horizontal merging operation has resulted in the product node N_{11*12} . This is justifiable due to the disappearance of the connection $z_{11,12}$ as well as to the fact that the input x_{11} to the product node is the same the input to the first operand node and the output y_{12} from the product node is the same as the output from the second operand node. In this context, the use of the symbol

‘*’ makes valid the postcondition for the formation of the product node N_{11*12} as a result of horizontal merging. This node can be described by the Boolean matrix in Equation (23).

$$\begin{array}{rcccl}
 N_{11*12}: & y_{12} & 1 & 2 & 3 & \\
 & x_{11} & & & & \\
 & 1 & 0 & 1 & 0 & \\
 & 2 & 1 & 0 & 0 & \\
 & 3 & 0 & 0 & 1 &
 \end{array} \tag{23}$$

Vertical merging is a binary operation that can be applied to a pair of parallel nodes, i.e. nodes located in the same layer of a FN. This operation merges the operand nodes from the pair into a single product node. In this case, the inputs to the product node represent the union of the inputs to the operand nodes whereas the outputs from the product node represent the union of the outputs from the operand nodes. The operation of vertical merging can always be applied due to the ability to concatenate the inputs and the outputs of any two parallel nodes.

When Boolean matrices are used as formal models for the operand nodes, the vertical merging operation is like an expansion of the first operand matrix along its rows and columns. In particular, the product matrix is obtained by expanding each non-zero element from the first operand matrix to a block that is the same as the second operand matrix and by expanding each zero element from the first operand matrix to a zero block of the same dimension as the second operand matrix. In this case, the row labels of the product matrix are all possible permutations of row labels of the operand matrices whereas the column labels of the product matrix are all permutations of column labels of the operand matrices.

Example 2:

This example considers the parallel operand nodes N_{11} and N_{21} located in the first layer of the four-node FN from Figure 1. These nodes are described there by the Boolean matrices in Equations (14) and (16). The connections of these nodes with the nodes in the second layer of this FN are given by the connection structure in Equation (19). In this context, nodes N_{11} and

N_{21} represent a two-node subnetwork of this FN. This two-node FN can be described by the block-scheme in Figure 4 and the topological expression in Equation (24).

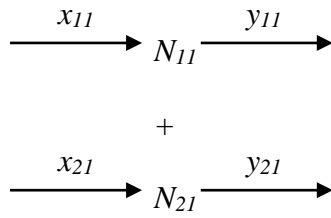


Figure 4: Two-node fuzzy network with operand nodes N_{11} and N_{21}

$$[N_{11}] (x_{11} / y_{11}) + [N_{21}] (x_{21} / y_{21}) \quad (24)$$

The use of the symbol ‘+’ in Figure 4 and Equation (24) implies that the vertical merging operation can be applied to the operand nodes N_{11} and N_{21} . In this context, the use of the symbol ‘+’ confirms the validity of the precondition for vertical merging of nodes N_{11} and N_{21} .

The vertical merging of the operand nodes N_{11} and N_{21} results into a single product node N_{11+21} which represents a simplified image of the two-node FN in the form of a one-node FN. The latter can be described by the block scheme in Figure 5 and the topological expression in Equation (25).

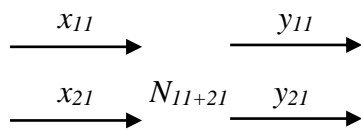


Figure 5: One-node fuzzy network with product node N_{11+21}

$$[N_{11+12}] (x_{11}, x_{21} / y_{11}, y_{21}) \quad (25)$$

The use of the symbol ‘+’ in Figure 5 and Equation (25) implies that the application of the vertical merging operation has resulted in the product node N_{11+12} . This is justifiable due to the concatenation of the inputs to the operand nodes as inputs x_{11}, x_{21} to the product node and the concatenation of the outputs from the operand nodes as outputs y_{11}, y_{21} from the product

node. In this context, the use of the symbol ‘+’ makes valid the postcondition for the formation of the product node N_{11+21} as a result of vertical merging. This node can be described by the Boolean matrix in Equation (26).

$$\begin{array}{r}
 N_{11+21}: \\
 \begin{array}{c}
 y_{11}, y_{21} \\
 x_{11}, x_{21}
 \end{array}
 \begin{array}{cccccccccc}
 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\
 11 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 12 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 13 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 21 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 22 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 31 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 32 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 33 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
 \end{array}
 \end{array} \tag{26}$$

Output merging is a binary operation that can be applied to a pair of parallel nodes with common inputs. This operation merges the operand nodes from the pair into a single product node. In this case, the inputs to the product node are the same as the common inputs to the operand nodes whereas the outputs from the product node represent the union of the outputs from the operand nodes. The operation of output merging can always be applied due to the ability to concatenate the outputs of any two parallel nodes with common inputs.

When Boolean matrices are used as formal models for the operand nodes, the output merging operation is like an expansion of the first operand matrix along its columns. In particular, the product matrix is obtained by expanding each non-zero element from the first operand matrix to a row-block that is the same as the corresponding row of the second operand matrix and by expanding each zero element from the first operand matrix to a zero row-block of the same dimension as the rows of the second product matrix. In this case, the row labels of the product matrix are the same as the identical row labels of the operand matrices whereas the column labels of the product matrix are all possible permutations of column labels of the operand matrices.

Example 3:

This example considers the parallel operand nodes N_{11} and N_{21} located in the first layer of the four-node FN from Figure 1 in a modified context. In particular, the two independent inputs x_{11} and x_{21} to these nodes are replaced by a common input $x_{11,21}$. The nodes are described by the Boolean matrices in Equations (14) and (16). The connections of these nodes with the nodes in the second layer of this FN are given by the connection structure in Equation (19). In this context, the nodes N_{11} and N_{21} represent a modified two-node subnetwork of this FN. This two-node FN can be described by the block-scheme in Figure 6 and the topological expression in Equation (27).

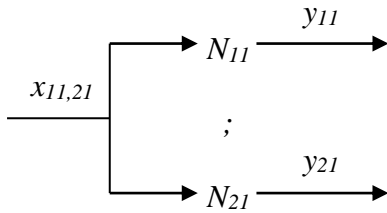


Figure 6: Two-node fuzzy network with operand nodes N_{11} , N_{21} and common input

$$[N_{11}] (x_{11,21} / y_{11}) ; [N_{21}] (x_{11,21} / y_{21}) \quad (27)$$

The use of the symbol ‘;’ in Figure 6 and Equation (27) implies that the output merging operation can be applied to the operand nodes N_{11} and N_{21} . In this context, the use of the symbol ‘;’ confirms the validity of the precondition for output merging of nodes N_{11} and N_{21} .

The output merging of the operand nodes N_{11} and N_{21} results into a single product node $N_{11;21}$ which represents a simplified image of the two-node FN in the form of a one-node FN. The latter can be described by the block scheme in Figure 7 and the topological expression in Equation (28).

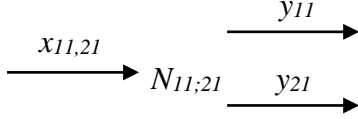


Figure 7: One-node fuzzy network with product node $N_{11;21}$

$$[N_{11;12}] (x_{11,21} / y_{11}, y_{21}) \quad (28)$$

The use of the symbol ‘;’ in Figure 7 and Equation (28) implies that the application of the output merging operation has resulted in the product node $N_{11;12}$. This is justifiable due to the concatenation of the outputs from the operand nodes as outputs y_{11}, y_{21} from the product node while preserving the common input to the operand nodes as an input $x_{11,21}$ to the product node. In this context, the use of the symbol ‘;’ makes valid the postcondition for the formation of the product node $N_{11;21}$ as a result of output merging. This node can be described by the Boolean matrix in Equation (29).

$$\begin{array}{r}
 N_{11;21} : \\
 \quad \quad \quad y_{11}, y_{21} \\
 \quad \quad \quad x_{11,21}
 \end{array}
 \begin{array}{ccccccccccc}
 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\
 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
 \end{array}
 \quad (29)$$

4. Properties of Basic Operations

The basic operations can be applied to fairly simple FNs with only a pair of nodes or a single node. However, an arbitrarily complex FN may have a large number of nodes whereby all of them have to be manipulated for the purpose of using the linguistic composition approach. Therefore, it is important to know how the basic operations can be applied in this more realistic context.

A key to the solution of the above problem are the associativity properties of basic operations. These properties facilitate the manipulation of nodes within an arbitrarily complex FN. This is illustrated briefly on FNs with three nodes but the extension to FNs with

an arbitrary number of nodes is straightforward. Therefore, the properties of basic operations in FNs are like the glue that makes these operations stick together.

The associativity property of horizontal merging allows three operand nodes A , B and C to be merged horizontally into a product node $A*B*C$ by means of a sequence of two binary merging operations that can be applied either from left to right or from right to left. The property can be applied when the output from the first node A is fed forward as an input to the second node B in the form of a connection and the output from the second node B is fed forward as an input to the third node C as another connection. In this case, the product node $A*B*C$ has the same input as the input to the first operand node A and the same output as the output from the third operand node C whereas the two connections do not appear in the product node.

Therefore, horizontal merging is associative in accordance with Equation (30). In this case, the horizontal merging of any three operand nodes A , B and C from left to right is equivalent to their horizontal merging from right to left.

$$(A*B)*C = A*(B*C) = A*B*C \quad (30)$$

The associativity property of vertical merging allows three operand nodes A , B and C to be merged vertically into a product node $A+B+C$ by means of a sequence of two binary merging operations that can be applied either from top to bottom or from bottom to top. The property can be applied when the inputs to and the outputs from each of the three nodes A , B and C are not connected with each other in any way. In this case, the input set to the product node $A+B+C$ is the union of the inputs to the operand nodes A , B and C whereas the output set from the product node is the union of the outputs from the operand nodes.

Therefore, horizontal merging is associative in accordance with Equation (31). In this case, the vertical merging of any three operand nodes A , B and C from top to bottom is equivalent to their vertical merging from bottom to top.

$$(A+B)+C = A+(B+C) = A+B+C \quad (31)$$

The associativity property of output merging allows three operand nodes A , B and C to be output merged into a product node $A;B;C$ by means of a sequence of two binary merging operations that can be applied either from top to bottom or from bottom to top. The property can be applied when the nodes A , B and C have common inputs and their outputs are not connected in any way. In this case, the input to the product node $A;B;C$ is the same as the input to each of the operand nodes A , B and C whereas the output set from the product node is the union of the outputs from the operand nodes.

$$(A;B);C = A;(B;C) = A;B;C \quad (32)$$

5. Advanced Operations in Fuzzy Networks

The properties of basic operations facilitate the application of these operations to a wide range of FNs with nodes that may be sequential, parallel or with common inputs. However, some FNs may include more complex connections among the nodes which would require preliminary manipulation before the basic operations can be applied. For this purpose, it is necessary to define some advanced operations such as input augmentation and output permutation.

Advanced operations make possible the manipulation of nodes in a FN with a more complex structure. These operations transform the nodes in the FN such that basic operations can be applied, i.e. they are like sophisticated building blocks in the process of simplifying a FN to a fuzzy system. For simplicity, the advanced operations are illustrated with examples of nodes with a small number of inputs and outputs but their extension to higher dimensional cases is straightforward.

Input augmentation can be applied when two or more nodes in a particular layer of a FN have some common inputs but also other inputs that are not common to all these nodes. In this case, it is necessary to augment the nodes with the missing common inputs such that all nodes have only common inputs. The purpose of this virtual augmentation is to allow the output merging operation to be applied to all nodes in this layer of the FN. As a result, the nodes with the augmented inputs have to be transformed appropriately to reflect the presence of these inputs.

When a Boolean matrix is used as a formal model for a node during input augmentation, the transformation of this node represents an expansion of this matrix along its rows. In particular, the product matrix is obtained by replicating each row from the operand matrix as many times as the number of permutations of linguistic terms for the augmented inputs minus one. The location of the replicated rows in the product matrix depends on the place of the augmented inputs in the extended set of inputs.

Example 4:

This example considers an operand node N with output y and input x that is augmented with an input x^{AI} . This node can be described by the Boolean matrix in Equation (33). In this context, node N represents a one-node FN that can be described by the block-scheme in Figure 8 and the topological expression in Equation (34).

$$\begin{array}{rcccl}
 N: & & y & 1 & 2 & 3 & & (33) \\
 & & x & & & & & \\
 & & 1 & 0 & 1 & 0 & & \\
 & & 2 & 1 & 0 & 0 & & \\
 & & 3 & 0 & 0 & 1 & &
 \end{array}$$

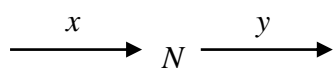


Figure 8: One-node fuzzy network before input augmentation

$$[N] (x / y) \tag{34}$$

As a result of this input augmentation, the operand node N is transformed into a product node N^{AI} with extended input set $\{x, x^{AI}\}$ and output y . This node can be described by the Boolean matrix in Equation (35). In this context, node N^{AI} represents a one-node FN that can be described by the block-scheme in Figure 9 and the topological expression in Equation (36).

$$\begin{array}{r}
 N^{AI}: \\
 \begin{array}{c}
 y \quad 1 \quad 2 \quad 3 \\
 x, x^{AI} \\
 11 \quad 0 \quad 1 \quad 0 \\
 12 \quad 0 \quad 1 \quad 0 \\
 13 \quad 0 \quad 1 \quad 0 \\
 21 \quad 1 \quad 0 \quad 0 \\
 22 \quad 1 \quad 0 \quad 0 \\
 23 \quad 1 \quad 0 \quad 0 \\
 31 \quad 0 \quad 0 \quad 1 \\
 32 \quad 0 \quad 0 \quad 1 \\
 33 \quad 0 \quad 0 \quad 1
 \end{array}
 \end{array} \tag{35}$$

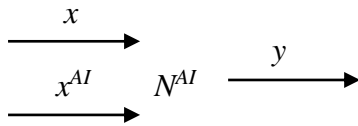


Figure 9: One-node fuzzy network after input augmentation

$$[N^{AI}] (x, x^{AI} / y) \tag{36}$$

Output permutation can be applied when two or more adjacent nodes in the same level of a FN have some connections with crossing paths. In this case, it is necessary to permute the output points of these connections such that the corresponding paths become parallel. The purpose of this permutation is to allow the horizontal merging operation to be applied to all nodes in this level of the FN. As a result, the nodes with the permuted outputs have to be transformed appropriately to reflect the changed ordering of these outputs.

When a Boolean matrix is used as a formal model for a node during output permutation, the transformation of this node is based on relocation of the non-zero columns of this matrix. In particular, the product matrix is obtained by moving each non-zero column from the

operand matrix under a column label with linguistic terms permuted in accordance with the associated permuted outputs. The space vacated by a relocated non-zero column in the product matrix is filled with a zero column unless another non-zero column is moved there as part of the overall node transformation process.

Example 5:

This example considers an operand node N with input x and output set $\{y_1, y_2\}$ whose outputs are permuted, i.e. y_2 comes first and y_1 comes second in the reordered set of outputs. Before the permutation, this node can be described by the Boolean matrix in Equation (37). In this context, node N represents a one-node FN that can be described by the block-scheme in Figure 10 and the topological expression in Equation (38).

$$\begin{array}{rcccccccccc}
 N: & & y_1, y_2 & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 & (37) \\
 & x & & & & & & & & & & & \\
 & 1 & & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\
 & 2 & & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\
 & 3 & & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &
 \end{array}$$

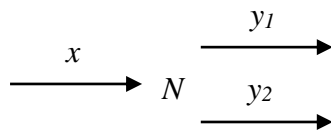


Figure 10: One-node fuzzy network before output permutation

$$[N] (x / y_1, y_2) \tag{38}$$

As a result of this output permutation, the operand node N is transformed into a product node N^{PO} with input x and reordered output set $\{y_2, y_1\}$. This node can be described by the Boolean matrix in Equation (39). In this context, node N^{PO} represents a one-node FN that can be described by the block-scheme in Figure 11 and the topological expression in Equation (40).

$$\begin{array}{rcccccccccc}
N^{PO}: & y_2, y_1 & 11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 \\
& x & & & & & & & & & \\
& 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
& 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
& 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array} \tag{39}$$

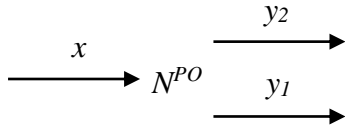


Figure 11: One-node fuzzy network after output permutation

$$[N^{PO}] (x / y_2, y_1) \tag{40}$$

6. Feedforward Fuzzy Networks

The basic operations, their properties and the advanced operations introduced are illustrated mainly on fairly simple FNs so far. Although these networks are assumed to be part of the structure of more complex FNs, the latter are taken into account only implicitly in the considerations. Therefore, it is also necessary to consider the application of the above operations and their properties to the overall structure of more complex FNs.

The current section discusses briefly the application of basic operations, their properties and advanced operations in feedforward FNs. The latter are FNs all of whose connections are only in a forward direction, i.e. from nodes residing in specific layers to nodes residing in subsequent layers. In particular, four types of feedforward FNs are considered here depending on the number of levels and layers.

The simplest type of FN is the one with single level and single layer. This network has only one node residing in the single level and the single layer of the associated location and connection structures. Due to the absence of other nodes, there are not any feedforward connections from and to this node. Therefore, a FN with single level and single layer is a single node network that is identical to a fuzzy system with a single rule base. This implies that a fuzzy system is a simple FN, i.e. a special case of a FN with single node. Similarly, a

FN can be viewed as a complex fuzzy system, i.e. a general case of a fuzzy system with networked rule bases.

A more complex type of FN is the one with single level and multiple layers. This network has at least two nodes residing in the single level and the multiple layers of the associated location and connection structures, i.e. it is identical to a queue of fuzzy systems with single rule bases. In this case, there may be connections between outputs from nodes and inputs to other nodes in subsequent layers. However, there shouldn't be any connections between outputs from nodes and inputs to same nodes or other nodes in preceding layers as such connections would be of feedback type.

Another more complex type of FN is the one with multiple levels and single layer. This network has at least two nodes residing in the multiple levels and the single layer of the associated location and connection structures, i.e. it is identical to a stack of fuzzy systems with single rule bases. In this case, there may be common inputs to nodes. However, there shouldn't be any connections between outputs from nodes and inputs to same nodes or other nodes in the same layer as such connections would be of feedback type.

The most complex type of FN is the one with multiple levels and multiple layers. This network has at least two nodes residing in the multiple levels and layers of the associated location and connection structures, i.e. it is identical to a grid of fuzzy systems with single rule bases. In this case, there may be feedforward connections between or common inputs to nodes but there shouldn't be any feedback connections between nodes in the context of the previous two types of FNs.

7. Application to Decision Support Systems

The proposed approach for complex systems modelling is applied to a case study from the bank industry. This case study is about a decision support system for assessing mortgage applications whereby the assessment is based on separate evaluations of the applicant and the

property. The input factors taken into account for the evaluation of the applicant are their asset and the income. For the evaluation of the property, the input factors taken into account are its price and location. The outputs from these two evaluation stages are the applicant and the property status. These outputs, together with the interest on the mortgage and the income of the applicant, are fed as input factors for the evaluation of the amount of credit that can be given to the applicant. The output from this third evaluation stage is the credit status.

The decision support system above can be represented by an initial FN. The latter can be described by the block-scheme in Figure 12 and the topological expression in Equation (41). The notations used in the figure and the equation are as follows: N_{11} is the rule base for the applicant evaluation, N_{21} is the rule base for the property evaluation, N_{12} is the rule base for the credit evaluation, $x_{11,12}^{1,3}$ is the applicant income, x_{11}^2 is the applicant asset, x_{12}^2 is the mortgage interest, x_{21}^1 is the property location, x_{21}^2 is the property price, $z_{11,12}^{1,1}$ is the applicant status, $z_{21,12}^{1,4}$ is the property status and y_{12} is the credit status. In these notations, the subscripts designate the location of the associated rule bases in terms of level and layer numbers whereas the superscripts refer to the position of the associated scalar variables in the case of multiple inputs, outputs or connections.

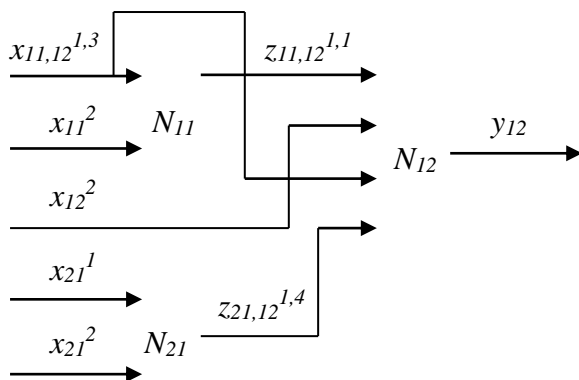


Figure 12: Initial fuzzy network for case study

$$\{[N_{11}] (x_{11,12}^{1,3}, x_{11}^2 | z_{11,12}^{1,1}) + [N_{21}] (x_{21}^1, x_{21}^2 | z_{21,12}^{1,4})\} * \quad (41)$$

$$[N_{12}] (z_{11,12}^{1,1}, x_{12}^2, x_{11,12}^{1,3}, z_{21,12}^{1,4} | y_{12})$$

There are two identity mappings propagating through the first layer of the underlying location structure of the initial FN - $x_{11,12}^{1,3}$ and x_{12}^2 . These mappings can be presented by the identity nodes I_{01} and $I_{1.5,1}$, respectively. As a result of this presentation, the initial FN can be transformed into a first interim FN. The latter can be described by the block-scheme in Figure 13 and the topological expression in Equation (42).

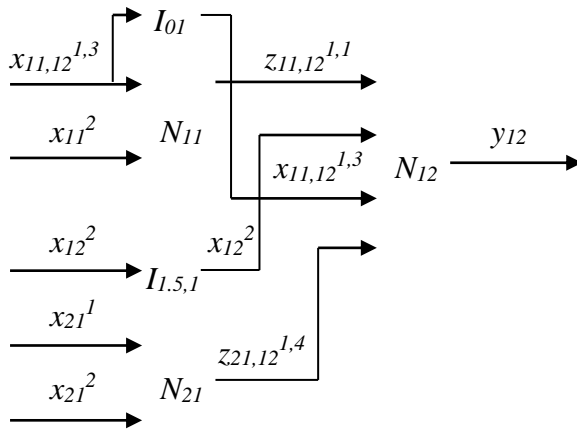


Figure 13: First interim fuzzy network for case study

$$\{[I_{01}] (x_{11,12}^{1,3} | x_{11,12}^{1,3}) ; [N_{11}] (x_{11,12}^{1,3}, x_{11}^2 | z_{11,12}^{1,1})\} + \quad (42)$$

$$[I_{1.5,1}] (x_{12}^{12} | x_{12}^{12}) + [N_{21}] (x_{21}^1, x_{21}^2 | z_{21,12}^{1,4})\} *$$

$$[N_{12}] (z_{11,12}^{1,1}, x_{12}^2, x_{11,12}^{1,3}, z_{21,12}^{1,4} | y_{12})$$

The outputs $x_{11,12}^{1,3}$ and $z_{11,12}^{1,1}$ from nodes I_{01} and N_{11} in the first interim FN could be merged if I_{01} is first augmented as I_{01}^{AI} with the input x_{11}^2 . This augmentation operation transforms the first interim FN into a second interim FN. The latter can be described by the block-scheme in Figure 14 and the topological expression in Equation (43).

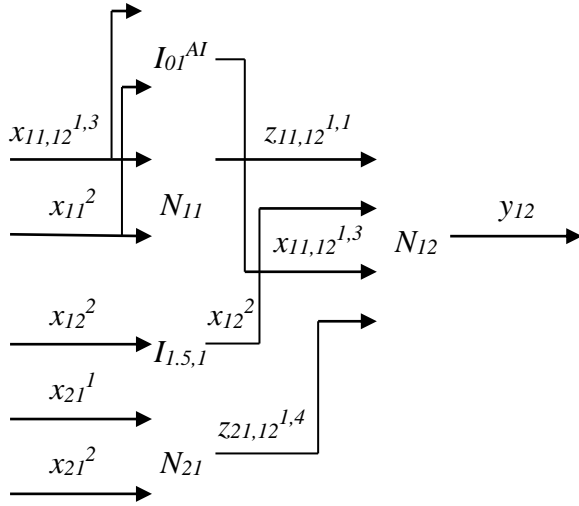


Figure 14: Second interim fuzzy network for case study

$$\begin{aligned}
 & \{ [I_{0I}^{AI}] (x_{11,12}^{1,3}, x_{11}^2 \mid x_{11,12}^{1,3}) ; [N_{11}] (x_{11,12}^{1,3}, x_{11}^2 \mid z_{11,12}^{1,1}) \} + \quad (43) \\
 & [I_{1.5,1}] (x_{12}^{12} \mid x_{12}^{12}) + [N_{21}] (x_{21}^1, x_{21}^2 \mid z_{21,12}^{1,4}) * \\
 & [N_{12}] (z_{11,12}^{1,1}, x_{12}^2, x_{11,12}^{1,3}, z_{21,12}^{1,4} \mid y_{12})
 \end{aligned}$$

The outputs $x_{11,12}^{1,3}$ and $z_{11,12}^{1,1}$ from nodes I_{0I}^{AI} and N_{11} in the second interim FN can already be merged as both nodes have the same common inputs $x_{11,12}^{1,3}$ and x_{11}^2 . This merging operation transforms the second interim FN into a third interim FN. The latter can be described by the block-scheme in Figure 15 and the topological expression in Equation (44).

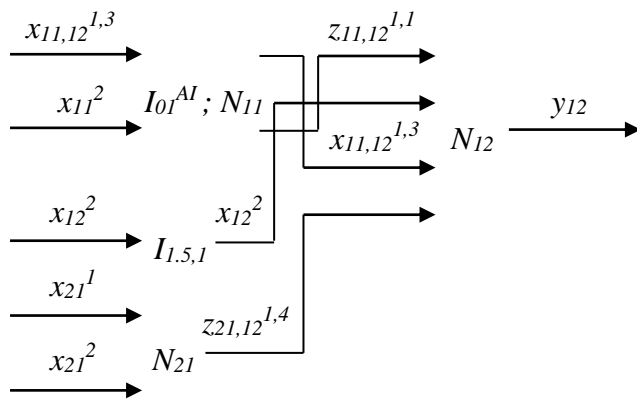


Figure 15: Third interim fuzzy network for case study

$$\{[I_{01}^{AI}; N_{11}] (x_{11,12}^{1,3}, x_{11}^2 / x_{11,12}^{1,3}, z_{11,12}^{1,1}) + [I_{1.5,1}] (x_{12}^{12} / x_{12}^{12}) + [N_{21}] (x_{21}^1, x_{21}^2 / z_{21,12}^{1,4})\} * [N_{12}] (z_{11,12}^{1,1}, x_{12}^2, x_{11,12}^{1,3}, z_{21,12}^{1,4} / y_{12}) \quad (44)$$

The nodes $I_{01}^{AI}; N_{11}, I_{1.5,1}$ and N_{21} in the third interim FN can be merged vertically. This merging operation transforms the third interim FN into a fourth interim FN. The latter can be described by the block-scheme in Figure 16 and the topological expression in Equation (45).

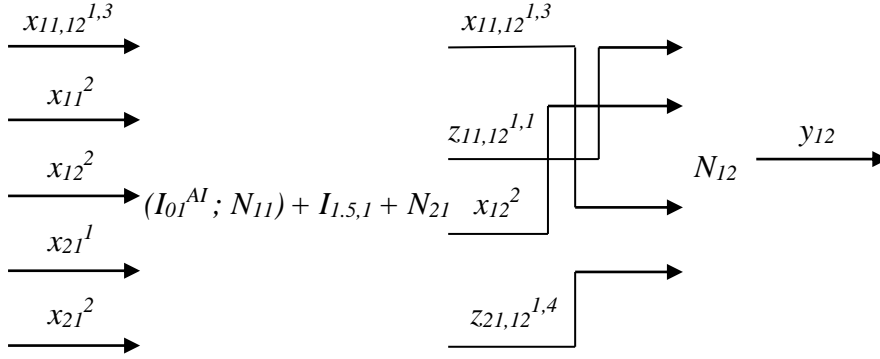


Figure 16: Fourth interim fuzzy network for case study

$$[(I_{01}^{AI}; N_{11}) + I_{1.5,1} + N_{21}] \quad (45)$$

$$(x_{11,12}^{1,3}, x_{11}^2, x_{12}^{12}, x_{21}^1, x_{21}^2 / x_{11,12}^{1,3}, z_{11,12}^{1,1}, x_{12}^{12}, z_{21,12}^{1,4}) *$$

$$[N_{12}] (z_{11,12}^{1,1}, x_{12}^2, x_{11,12}^{1,3}, z_{21,12}^{1,4} / y_{12})$$

The first three outputs from node $(I_{01}^{AI}; N_{11}) + I_{1.5,1} + N_{21}$ in the fourth interim FN can be permuted such that the first output becomes third, the second output becomes first and the third output becomes second. This permutation operation transforms the fourth interim FN into a fifth interim FN. The latter can be described by the block-scheme in Figure 17 and the topological expression in Equation (46).

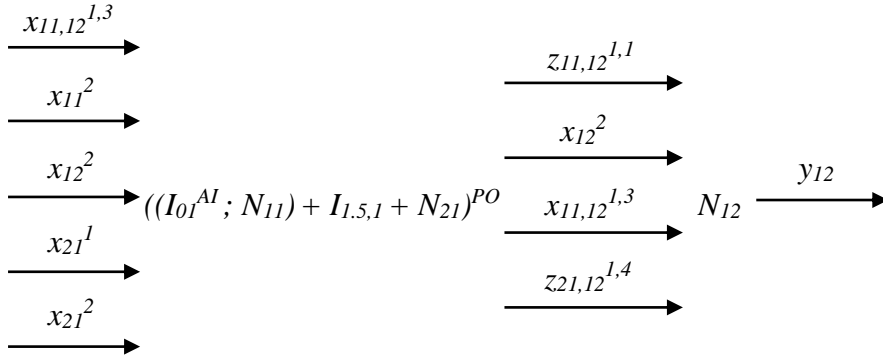


Figure 17: Fifth interim fuzzy network for case study

$$\begin{aligned}
 & [((I_{0I}^{AI}; N_{11}) + I_{1.5,1} + N_{21})^{PO}] \tag{46} \\
 & (x_{11,12}^{1,3}, x_{11}^2, x_{12}^{12}, x_{21}^1, x_{21}^2 \mid z_{11,12}^{1,1}, x_{12}^{12}, x_{11,12}^{1,3}, z_{21,12}^{1,4}) * \\
 & [N_{12}] (z_{11,12}^{1,1}, x_{12}^2, x_{11,12}^{1,3}, z_{21,12}^{1,4} \mid y_{12})
 \end{aligned}$$

The nodes $((I_{0I}^{AI}; N_{11}) + I_{1.5,1} + N_{21})^{PO}$ and N_{21} in the fifth interim FN can be merged horizontally. This merging operation transforms the fifth interim FN into a final FN. The latter can be described by the block-scheme in Figure 18 and the topological expression in Equation (47).

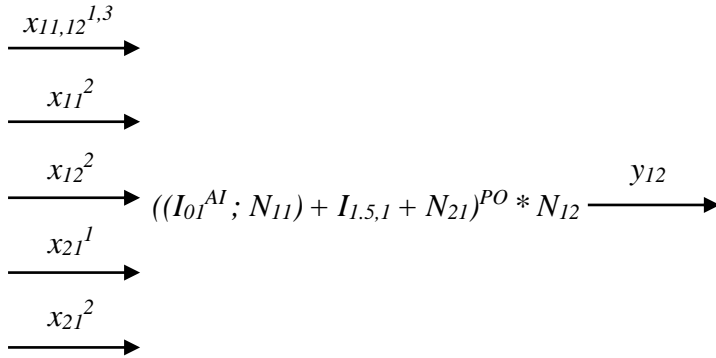


Figure 18: Final fuzzy network for case study

$$[((I_{0I}^{AI}; N_{11}) + I_{1.5,1} + N_{21})^{PO} * N_{12}] (x_{11,12}^{1,3}, x_{11}^2, x_{12}^{12}, x_{21}^1, x_{21}^2 \mid y_{12}) \tag{47}$$

The proposed approach for complex systems modelling is evaluated comparatively in terms of model transparency for the fuzzy network and an associated fuzzy system. In this

case, the fuzzy network model is based on the initial fuzzy network whereas the fuzzy system model is similar to the final fuzzy network.

The model transparency index used is given by the formula in Equation (48)

$$(s+z)/(m+n) \tag{48}$$

where s is the number of subsystems, z is the number of connections, m is the number of inputs and n is the number of outputs. The formula implies that the model transparency increases with the increase in the number of subsystems and connections or with the decrease in the number of inputs and outputs.

The fuzzy system and fuzzy network models are evaluated comparatively in terms of their transparency Table 1. The transparency index figure obtained for these two models is, 0.16 and 0.83, respectively. This shows that the fuzzy network is about five times superior to the fuzzy system in terms of modelling transparency and ability to reflect qualitative complexity.

Table 1: Transparency evaluation for fuzzy system and fuzzy network

Fuzzy model / Parameters	Number of subsystems	Number of connections	Number of inputs	Number of outputs	Transparency index
Fuzzy system	1	0	5	1	0.16
Fuzzy network	3	2	5	1	0.83

8. Conclusion

The proposed approach for complex systems modelling by fuzzy networks with feedforward rule bases improves the transparency of the models used. This allows the structure of a fairly complex interconnected process to be reflected explicitly in the model. As a result, any complex process can be modelled by a fuzzy network in a more transparent way than by a fuzzy system due to the better visibility inside the process. This also leads to better understanding of the modelled process.

The proposed approach is based on formal models for fuzzy networks, basic operations in such networks, their properties and advanced operations. The formal models used are Boolean matrices, block-schemes and topological expressions. The basic operations are binary and they include horizontal, vertical and output merging of rule bases. The basic operations are also associative which facilitates the merging of an arbitrary number of rule bases. The advanced operations are used for more complex fuzzy networks and they include input augmentation and output permutation.

The proposed approach is illustrated for feedforward fuzzy networks with a fairly small number of inputs, outputs and connections. However, it can be easily extended to feedback fuzzy networks with an arbitrarily large number of inputs, outputs and connections. In this case, all binary merging operations can be applied repetitively in a flexible way by means of the associativity property. This would lead only to a linear increase of the associated quantitative complexity.

Acknowledgement

The first author would like to thank the Faculty of Technology at the University of Portsmouth for the granted research sabbatical that made possible the writing of this paper.

References

- [1] Fernandez-Caballero A (2009) Contribution of fuzziness and uncertainty to modern artificial intelligence. *Fuzzy Sets and Systems* 160/2:129-129
- [2] Garibaldi J M, Ozen T (2007) Uncertain fuzzy reasoning: a case study in modelling expert decision making. *IEEE Transactions on Fuzzy Systems* 15/1:16-30
- [3] Gil-Aluja (2004) *Fuzzy sets in the management of uncertainty*. Springer, Berlin
- [4] Chen B, Liu X (2005) Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by backstepping and application to chemical processes. *IEEE Transactions on Fuzzy Systems* 13/6:832-847

- [5] Chen B, Tong S, Liu X (2007) Fuzzy approximate disturbance decoupling of MIMO nonlinear systems by back stepping approach. *Fuzzy sets and systems* 158/10:1097-1125
- [6] Wang W J, Lin W W (2005) Decentralized PDC for large-scale fuzzy systems. *IEEE Transactions on Fuzzy Systems* 13/6:779-786
- [7] Wang W J, Luoh L (2004) Stability and stabilization of fuzzy large-scale systems. *IEEE Transactions on Fuzzy Systems* 12/3:309-315
- [8] Gegov A (2007) Complexity management in fuzzy systems: a rule base compression approach. Springer, Berlin
- [9] Kim K J, Cho S B (2007) Personalized mining of web documents link structures and fuzzy concept networks. *Applied Soft Computing* 7/1:398-410
- [10] Xu C, Shin Y C (2007) Interaction analysis of MIMO nonlinear systems based on fuzzy basis function network model. *Fuzzy sets and systems* 158/18:2013-2015
- [11] Zhang H, Dang C, Li C (2009) Decentralized H-infinity filter design for discrete-time interconnected fuzzy systems. *IEEE Transactions on Fuzzy Systems* 17/6:1428-1440
- [12] Zhang T P (2001) Stable adaptive fuzzy sliding mode control of interconnected systems. *Fuzzy Sets and Systems* 122/1:5-19
- [13] Buckley, J J (2005) *Simulating fuzzy systems*. Springer, Berlin
- [14] Piegat A (2001) *Fuzzy modelling and control*. Springer, Berlin
- [15] Ross T (2004) *Fuzzy logic with engineering applications*. Wiley, Chichester
- [16] Ji-Cheng Duan, Fu-Lai Chung (2001) Cascaded fuzzy network model based on syllogistic fuzzy reasoning. *IEEE Transactions on Fuzzy Systems* 9/2:293-306
- [17] Lendek Z, Babuska R, De Schutter B (2009) Stability of cascaded fuzzy systems and observers. *IEEE Transactions on Fuzzy Systems* 17/3:641-653
- [18] Mar J, Lin H T (2005) A car-following collision prevention control device based on the cascaded fuzzy inference system. *Fuzzy Sets and Systems* 150/3:457-473

- [19] Yeh Z M, Li K H (2004) A systematic approach for designing multistage fuzzy control systems. *Fuzzy Sets and Systems* 143/2:251-273
- [20] Bucolo M, Fortuna L, La Rosa M (2004) Complex dynamics through fuzzy chains. *IEEE Transactions on Fuzzy Systems* 12/3:289-295
- [21] Hall L O (2001) Rule chaining in fuzzy expert systems. *IEEE Transactions on Fuzzy Systems* 9/6:822-828
- [22] Igel C, Temme K H (2004) The chaining syllogism in fuzzy logic. *IEEE Transactions on Fuzzy Systems* 12/6:849-853
- [23] Aja-Fernandez S, Alberola-Lopez C (2008) Matrix modelling of hierarchical fuzzy systems. *IEEE Transactions on Fuzzy Systems* 16/3:585-599
- [24] Chen Y, Yang B, Abraham A, Peng L (2007) Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms. *IEEE Transactions on Fuzzy Systems* 15/3:385-397
- [25] Cheong F, Lai R (2007) Designing a hierarchical fuzzy logic controller using the differential evolution approach. *Applied Soft Computing* 7/2:481-491
- [26] Joo M G, Lee J S (2005) A class of hierarchical fuzzy systems with constraints on the fuzzy rules. *IEEE Transactions on Fuzzy Systems* 13/2:194-203
- [27] Lee M L, Chung H Y, Yu F M (2003) Modelling of hierarchical fuzzy systems. *Fuzzy Sets and Systems* 135/2:343-361
- [28] Gegov A (2010) *Fuzzy networks for complex systems: a modular rule base approach*. Springer, Berlin
- [29] Landago M, Rio M J, Perez A (2001) A note on smooth approximation capabilities of fuzzy systems. *IEEE Transactions on Fuzzy Systems* 9/2:229-237
- [30] Li Y M, Shi Z K, Li Z H (2002) Approximation theory of fuzzy systems based upon genuine many-valued implications –MIMO cases. *Fuzzy Sets and Systems* 130/2:159-174

[31] Moreno-Velo F J, Baturone I, Barriga A, Sanchez-Solano S (2007) Automatic tuning of complex fuzzy systems with X-fuzzy. *Fuzzy sets and systems* 158/18:2026-2038

[32] Wan F, Shang H, Wang L X, Sun Y X (2005) How to determine the minimum number of fuzzy rules to achieve given accuracy: a computational geometric approach to SISO case. *Fuzzy Sets and Systems* 150/2:199-209

[33] Gil-Aluja (2004) *Fuzzy sets in the management of uncertainty*. Springer, Berlin

[34] Joo M G, Sudkamp T (2009) A method of converting a fuzzy system to a two-layered hierarchical fuzzy system and its run-time efficiency. *IEEE Transactions on Fuzzy Systems* 17/1:93-103

[35] Pal N R, Eluri V K, Mandal G K (2002) Fuzzy logic approaches to structure preserving dimensionality reduction. *IEEE Transactions on Fuzzy Systems* 10/3:277-286

[36] Xiong N, Litz L (2002) Reduction of fuzzy control rules by means of premise learning – method and case study. *Fuzzy Sets and Systems* 132/2:217-231

[37] Fiordaliso A (2001) A constrained Takagi-Sugeno fuzzy system that allows for better interpretation and analysis. *Fuzzy Sets and Systems* 118/2:307-318

[38] Guillaume S (2001) Designing fuzzy inference systems from data: an interpretability-oriented review. *IEEE Transactions on Fuzzy Systems* 9/3:426-443

[39] Kumar M, Stoll R, Stoll N (2006) A robust design criterion for interpretable fuzzy models with uncertain data. *IEEE Transactions on Fuzzy Systems* 14/2:314-328

[40] Zhou S M, Gan J Q (2008) Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets and Systems* 159/23:3091-3131

[41] Pancho D P, Alonso J M, Cordon O, Quirin A, Magdalena L (2013) FINGRAMS: visual representations of fuzzy rule-based inference for expert analysis of comprehensibility. *IEEE Transactions on Fuzzy Systems* 21/6:1133-1149

[42] Zhou S M, Lyons R A, Bodger O, John A, Brunt H, Jones K, Gravenor M B, Brophy S (2014) Local modelling techniques for assessing micro-level impacts of risk factors in complex data: understanding health and socioeconomic inequalities in childhood educational attainments. PLOS ONE 9/11:e113592

[43] Engelfriet J, Vereijken J J (1997) Context-free grammars and concatenation of graphs. Acta Informatica 34/10:777-803

[44] Bozapalidis S, Kalampakis A (2004) An axiomatization of graphs. Acta Informatica 41/1:19-61

[45] Bozapalidis S, Kalampakis A (2006) Recognizability of graph and pattern languages. Acta Informatica 42/8-9:553-581

[46] Bozapalidis S, Kalampakis A (2008) Graph Automata. Theoretical Computer Science 393:147-165