Fuzzy neurules: Edging over Neurules

IOANNIS HATZILYGEROUDIS, CONSTANTINOS KOUTSOJANNIS
Dept of Computer Engineering & Informatics
School of Engineering
University of Patras
265 00 Patras,
HELLAS (GREECE)

Abstract: - The combination of techniques from expert systems and neural networks has the potential of producing more powerful systems. Neuro-fuzzy systems combine the positive attributes of both these approaches producing fuzzy systems with the ability to learn and adapt to real world observations. In this paper, we present a method for incorporating fuzzy inputs to neurules, that is a type of hybrid rules integrating symbolic (production rules) with connectionist (adaline unit) representation, presenting satisfactory behavior, by a uniform and tight integration of the two components. The final model based on fuzzy neurons is more global and efficient, since it requires the least possible training effort, the number of the produced units is kept as small as possible, and it works in an environment of fuzzy as well as crisp data. The system is ideal for applications such as: online real-time dynamic control; classification; and sensor condition monitoring.

Keywords: - hybrid rules, neurofuzzy modeling, fuzzy hybrid architectures, fuzzy hybrid inference

1 Introduction

Hybrid architectures for intelligent systems are used to exploit the strengths of its constituent systems (e.g. expert systems and neural networks) and expand the application domain beyond the set of applications to which either system could be applied individually [1, 2, 19]. Hybrid systems offer means to overcome some of the major drawbacks of conventional expert systems: 1) their total reliance on consultation with human experts for knowledge acquisition (bottleneck), 2) their inability to synthesize new knowledge and 3) their inability to allow for dynamic environments by miffing knowledge whenever it becomes necessary [3, 9]. On the other hand, in building a hybrid system, we cannot compromise certain basic characteristics of expert systems, such as 1) the ability to receive and represent knowledge; 2) the capacity to deal with data absence and to query additional data; 3) the intelligibility of knowledge base and 4) the ability to explain and convince the user that the system’s reasoning is correct [4, 6].

Approximate reasoning, based on fuzzy logic, has been successfully employed in reasoning of traditional expert systems [3, 21]. Both neural networks and fuzzy logic are structurally different, but they share a complementary nature as far as strengths and weaknesses are concerned. Neurofuzzy modeling refers to the combination of fuzzy set theory and neural networks in a way that exploits the advantages of both, such as managing imprecise, partial, vague or imperfect information; self-learning, self-organizing and self-tuning capabilities; no need of prior knowledge of relationships of data; mimic human decision making process and fast computation using fuzzy number operations[3, 5].

Neural networks can be modified in order to incorporate fuzzy techniques. Thus, neural networks with improved output are created. These systems give priority in the connectionist representation. An approach of fuzzy neural networks allows the system to receive and to process fuzzy entries. Other approaches concern the addition of levels before and/or afterwards neural network for the fuzzyfication of regular data of an entry or expense. With the combination of vague rules and neural networks fuzzy connectionist expert systems[15] are created. These systems continue having the disadvantages of connectionist expert systems with regard to the knowledge base that concerns the difficulty of comprehension from the user and the difficulty of progressive growth.

The main focus of fuzzy neural nets is on fuzzyfication of inputs and aggregation operations of conventional neurons. This has resulting in a variety of fuzzy neurons with different properties. The objective of introducing fuzzy operations within neural networks is to improve the expressiveness and flexibility of the neural networks. The objective of bringing neuronal learning capabilities into fuzzy systems is to make this fuzzy system capable to on-line adaptations
These systems are ideal for applications such as: online real-time dynamic process modelling and control; classification; and sensor condition monitoring [11, 12].

Neurules are a kind of hybrid rules that integrate neurocomputing and production rules. This is achieved incorporating neurocomputing within the symbolic framework of production rules. Neurules are produced by converting existing symbolic rules. It is achieved by a uniform and tight integration of a symbolic component and a connectionist one [7,8]. Each neurule is represented as an adaline unit. Thus, the corresponding neurule base consists of a number of autonomous adaline units called neurules.

In this paper, we introduce a method for improving the performance of neurules to handle with fuzzy inputs (fuzzy neurules). A fuzzy neurule has the same structure as the basic neurule model except that some or all of its parameters are described through mathematics of fuzzy logic. For example the aggregation operators used in fuzzy neurules are mostly min and max types.

With the incorporation of fuzzy logic in neurules three individual representations (production rules, fuzzy logic, neural networks) are combined closely, by a uniform and tight integration. Moreover, they seem to potentially effectively face the disadvantages that the fuzzy-rule-based expert systems have.

The structure of this paper is as follows. Section 2 presents the hybrid formalism and corresponding system architecture. Section 3, deals with the inference mechanism and finally section 4 concludes.

2 The hybrid formalism

2.1 Syntax and semantics of neurules

Neurules are a kind of hybrid rules. Neurules are produced by converting existing symbolic rules. It is achieved by a uniform and tight integration of a symbolic component (production rules) and a connectionist one (an adaline unit). In a neurule structure rather than a loose one. The form of a neurule is depicted in Fig.1a. Each condition Ci is assigned a number sfᵢ, called its significance factor. Moreover, each rule itself is assigned a number sf₀, called its bias factor. Internally, each neurule is considered as an adaline unit (Fig.1b). The inputs Ci (i=1,...,n) of the unit are the conditions of the rule. The weights of the unit are the significance factors of the neurule and its bias is the bias factor of the neurule. Each input takes a value from the following set of discrete values: [1 (true), -1 (false), 0 (unknown)]. The output D, which represents the conclusion (decision) of the rule, is calculated via the formulas:

\[ D = f(\mathbf{a}) = \sum_{i=1}^{n} \mathbf{a}_i \cdot C_i \]  

where \( \mathbf{a} \) is the activation value and \( f(x) \) the activation function, a threshold function:

\[ f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases} \]

Hence, the output can take one of two values, ‘-1’ and ‘1’, representing failure and success of the rule respectively.

The general syntax of a condition Ci and the conclusion D:

\[ \langle \text{condition} \rangle ::= \langle \text{variable} \rangle \ \langle \text{l-predicate} \rangle \ \langle \text{value} \rangle \]

\[ \langle \text{conclusion} \rangle ::= \langle \text{variable} \rangle \ \langle \text{r-predicate} \rangle \ \langle \text{value} \rangle \]

where \( \langle \text{variable} \rangle \) denotes a variable, that is a symbol representing a concept in the domain, e.g. ‘sex’, ‘pain’ etc, in a medical domain. A variable in a condition can be either an input variable or an intermediate variable, whereas a variable in a conclusion can be either an intermediate or an output variable or both. An input variable takes values from the user (input data), whereas intermediate and output variables take values through inference, since they represent intermediate and final conclusions respectively. \( \langle \text{l-predicate} \rangle \) and \( \langle \text{r-predicate} \rangle \) are one of \{is, isnot\}. \( \langle \text{value} \rangle \) denotes a value. It can be a symbol or a number. Neurules are distinguished in intermediate and output rules, depending on whether their conclusions contain intermediate or output variables respectively.
In this way, various benefits of symbolic rule-based representation, such as naturalness and modularity, are retained. So, neurules are understandable, since significance factors represent the contribution of corresponding conditions in drawing the conclusion. On the other hand, inferences are made more efficiently for two reasons. First, the number of participating rules has been drastically reduced. Second, the number of the evaluated conditions has been also reduced, due to embedded heuristics [8].

2.2 The Fuzzy Neurons

A hybrid neural net, like the one consisted of neurules; is a neural net with crisp signals and weights and crisp transfer function [17, 10]. However, we can: 1) combine inputs and weights using a t-norm or some other continuous operation, 2) aggregate the results using a t-norm or some other continuous function and transfer function can be a continuous function from input to output. We emphasize that all inputs, outputs and the weights of a hybrid neural net are real numbers taken from interval [0,1]. This modification leads to a structure of fuzzy neuron [12], based on fuzzy operators. Using fuzzy logical neurons [11], the output is more or less influenced by the values of inputs. This influence depends on both the weights and the operation of fusion: 1) for a neuron of type AND, the influence of its inputs having a weak weight is most important (min) 2) for a neuron of type OR the inputs whose weight is significant are rather taken into account (max) This defined the interval of possible values for the output (Fig. 2).

The expression of the output \( y \) of a neuron with \( N \) inputs of type AND is given by:

\[
y = \min(\max(w_1, x_1), \max(w_2, x_2), \ldots , \max(w_n, x_n))
\]

with \( y \in [\min(w_1, w_2, \ldots , w_n), 1] \)

The expression of the output \( y \) of a neuron with \( N \) inputs of type OR is given by:

\[
y = \max(\min(w_1, x_1), \min(w_2, x_2), \ldots , \min(w_n, x_n))
\]

with \( y \in [\max(w_1, w_2, \ldots , w_n), 1] \)

Hybrid neurons with traditional arithmetic functions and triangular norms and co-norms have been used to implement fuzzy neurons. The output may be carried out using a function of activation of sigmoid type for example. The max fuzzy neuron is a neuron that uses an aggregation function that selects the maximum of the inputs. It is an implementation of logical OR (Fig. 2).

![Fig.2 AND and OR fuzzy neurons](image)

The min fuzzy neuron is a neuron that uses an aggregation function that selects the minimum of the inputs. It is an implementation of logical AND. In neuronal fuzzyfication each fuzzy neuron may be thought of as a presentation of a linguistic value such as LOW, MEDIUM and so on. The output of the neuron could be associated with membership function to some linguistic value and expresses the degree to which the inputs belong to a given linguistic category. The output is a real value in the interval [0,1]. The character of the fuzzy neuron is determined from the aggregation operator and the activation function used. This gives rise to many types of fuzzy neurons (AND, OR, NOR, etc).

2.3 Fuzzy neural networks

A network of fuzzy neurons is different from a traditional network because of the function of each neuron and the semantics that are used for its identification. The function of such networks is the modeling of inference rules [21,22]. The outputs give a measurement of the realization of a rule, i.e. the membership of an expected value. If the value is described by only one rule, the knowledge can be represented as follows:

**IF** \([x_1 \text{ is } A_1]*[x_2 \text{ is } A_2]* \ldots *[x_i \text{ is } A_i]*[x_n \text{ is } A_n] \)**

**THEN** \((y \text{ is } B)\)

The operators *i are operators of conjunction or disjunction used to combine the premise. The structure is generally split into functional layers: 1) the layer of input weights the values of input according to their importance, 2) the hidden layers combine the premises and 3) he layer of output measures the degree of adequacy of the premises for the value considered. In such networks, the
learning cannot be carried out by the traditional algorithm of backpropagation. The backpropagation algorithm must adapt the computation of the errors to the structure of each neuron.

2.4 Types of Fuzzy inference systems

Most fuzzy inference systems proposed in literature can be classified in three types:

Type I: the overall output is the weighted average of each rule’s crisp output induced by the rules firing strength (the product or minimum of the degrees of match with the premise part)

Type II: the overall fuzzy output is derived by applying “max” operation to the qualified fuzzy outputs (each of which is equal to the minimum of firing strength and the membership function of each rule)

Type III: Takagi and Sugeno’s fuzzy if-then rules were the output of each rule is a linear combination of input variables plus a constant term [10, 11],

\[
\text{IF} \ (x \ \text{is} \ A_1) \ \text{AND} \ (y \ \text{is} \ B_1) \\
\text{THEN} \quad z = p_1 \cdot x + q_1 \cdot y + r_1,
\]

\[
\text{IF} \ (x \ \text{is} \ A_2) \ \text{AND} \ (y \ \text{is} \ B_2) \\
\text{THEN} \quad z = p_2 \cdot x + q_2 \cdot y + r_2
\]

and finally the overall output is the weighted average of each rule’s output [17, 22]:

\[
z = \frac{z_1 \cdot w_1 + z_2 \cdot w_2}{w_1 + w_2}
\]

where:

\[z_1 = p_1 \cdot x + q_1 \cdot y + r_1 \quad \text{and} \quad z_2 = p_2 \cdot x + q_2 \cdot y + r_2\]

3 Fuzzy neurules: a hybrid knowledge representation scheme

3.1 Fuzzy inference

Our fuzzy inference system is composed of five functional blocks (Fig.3).

- A **rule base** containing a number of fuzzy if-then rules
- A **database** which defines the membership functions of the fuzzy sets used in the fuzzy rules
- A **decision-making** unit which performs the inference operations on the rules

- A **fuzzification** block which transforms crisp inputs into degrees of match with linguistic values
- A **defuzzification** block which transforms the fuzzy results of the inference into a crisp output [13, 15].

![Fig.3 Fuzzy inference system](image)

The steps of fuzzy reasoning are:

1) compare the input variables with the membership functions on the premise part to obtain the membership values (fuzzification)

2) combine (through a specific t-norm operator, usually multiplication or min), the membership values on the premise part to get firing strength (weight) of each rule

3) generate the qualified consequent (fuzzy or crisp) of each rule depending on the firing strength

4) aggregate the qualified consequences to produce a crisp output (defuzzification).

Our system follows Type III inference system.

3.2 The Fuzzy hybrid architecture

The term ‘fuzzy neural networks’ denotes neural networks that are fuzzy and deal with fuzzy signals and fuzzy weights. By employing a hybrid neural learning procedure, this architecture refines fuzzy if-then rules obtained from human experts, to describe the input-output behavior of a complex system. Even if human expert is not available, it can still set up intuitively reasonable initial membership functions and start the learning process to generate a set of if-then fuzzy rules to
approximate a desired data set. In Fig. 4 the functional architecture of our hybrid system (except the fuzzification and defuzzification modules) to incorporate the integrated formalism is presented. The run-time system consists of three modules, functionally similar to those of a conventional fuzzy rule-based system: hybrid rule base (HRB), hybrid inference mechanism (HIM) and working memory (WM).

Fig. 4 The fuzzy neurule-based architecture

HRB (corresponding to the Knowledge Base in Fig. 3) contains fuzzy neurules produced by conversion from the fuzzy rule base (FRB) via the conversion and training mechanism (CTM). FRB contains only fuzzy rules. So, production of the fuzzy neurules takes place before run-time.

HIM (corresponding to the Inference Engine in Fig. 3) is responsible for making inferences by taking into account the initial conditions in the WM and the fuzzy neurules in the HRB. WM contains facts. A fact has the same format as a condition/conclusion of a rule.

4 Conclusions

In this paper, we present a way of incorporating fuzzy logic to neurules, which are a type of hybrid rules integrating symbolic rules and neurocomputing, thus producing fuzzy neurules. Fuzzy neurules can handle imprecise data and uncertainty as well as numerical data. The final model based on fuzzy neurons is more global and efficient, since it requires the least possible training effort, the number of the produced units is kept as small as possible, and it works in an environment of either crisp or fuzzy data. Our system is ideal for applications such as: online real-time dynamic control; classification; and sensor condition monitoring.

References:


**Acknowledgements**

This work was supported by the Research Committee of the University of Patras, Greece, Program Caratheodoris-2001, project No 2788.