# Computational Intelligence and its Applications Assignment 2 Linguistic Semantic Memory


Mohammad Bordbarkhoo
UiT The Arctic University of Norway
Department of Computer Science
Email: mbo116@post.uit.no

Dániel Nánási
UiT The Arctic University of Norway
Department of Computer Science
Email: dna005@post.uit.no


## I. Introduction

Artificial neural networks are computational models based on the structure of biological neural networks. Unexplained behavior of these networks is the most crucial problem of ANNs. So, rule generation from ANNs is critical; and in this regard, fuzzy systems are used. Therefore, the linguistic model of fuzzy systems (IF-THEN) have combined with a complicated mathematical model of neural networks to provide more comprehensive information for human beings. Both neural networks and fuzzy systems are dynamic processing systems that estimate input-output functions without any mathematical model and using training data. The hybridization of neural networks with the fuzzy system takes advantage of the low-level learning ability of neural networks and high-level reasoning ability of the fuzzy systems. These fuzzy-neural networks interpret semantic memory as a set of IF-THEN fuzzy rules. By adopting a learning process of fuzzy rules, the fuzzy-neural models learn sets of evolving (time-varying) semantic knowledge. Two main requirements of neural fuzzy networks are accuracy and interpretability. Some of the recent research has focused on improving these two issues by rule reduction. Other research has focused on the rule generation algorithm since the goal of the fuzzy neural network was rule extraction. In this work, we discuss fuzzy systems in general, neural fuzzy systems, neural fuzzy rule generation, classifications, and some novel fuzzy neural networks.

## II. Fuzzy systems in general

Fuzzy logic is a different way of thinking about features grouping than the classical ways. Fuzzy logic is based on the observation that people make decisions based on non-numerical information. Fuzzy sets are representing a non exact group or sets of information. For example we cannot program the computer to detect cold or warm temperature exact for people. Because it is subjective, but we are able to make measurements of peoples' subjective opinions, for example most of the people would say it is cold if there is 15 °C in a room, but there would be bigger difference in a 20 °C room. So the subjective sets of temperature are overlapping as it shown on figure 1. [1]

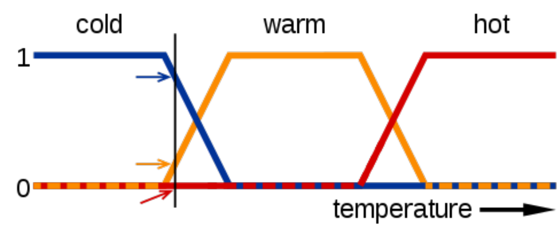


Fig. 1. Fuzzy sets of temperature

This is the essential idea of fuzzy sets and in this section we discuss about how could it be used in linguistic memory.

## III. Neural network implementation of fuzzy logic

In this section we will discuss about the way of creating a neural network with fuzzy logic which could be useful for linguistic memory. Neural network models attempt to achieve good performance via dense interconnection of simple computational elements. In this sense, a neural net structure is based on our current understanding of biological nervous systems. These models can explore many competing hypotheses simultaneously using parallel nets of many computational elements connected

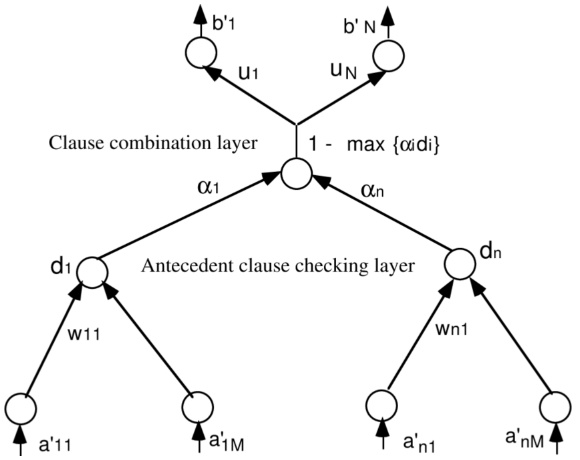


Fig. 2. Neural network configuration for fuzzy logic inference.

by links with variable weights. The hierarchical organizations are intended to interact with objects of the real world in the same way as biological nervous systems do. [2] Each basic network structure implements a single rule in the rule base of the form: if $X_1$ is $A_1$ and $X_2$ is $A_2$ and $X_n$ is $A_n$, then Y is B. The fuzzy sets which characterize the possibility distribution of the facts: $X_1$ is $A_1$ ... $X_n$ is $A_n$ are presented to the input layer of the network. The neural network system combined with fuzzy logic is shown on figure 2. [2] There are two variations of the activities in the antecedent clause checking layer. In both cases, each antecedent clause of the rule determined the weights. In the first variation weights $w_{ij}$ are the fuzzy set complement of the antecedent clause. So

$$w_{ij} = \overline{a_{ij}} = 1 - a_{ij} \tag{1}$$

where the clause $X_i$ is $A_i$ is translated into a possibility distribution

$$\pi_{X_i}(\nu_j) = a_{ij} \tag{2}$$

. The neural network will generate a measure of disagreement between the actual and the antecedent distributions. The next layer will combine the mix of the first layer by the defined activation function. The weights $u_i$ between the last hidden layer and the output carry the information from the consequent of rule. If the proposition Y is B is characterized by the possibility distribution $\pi_Y(x_i) = b_i$ for all $x_i$ in the domain of B, then $u_i = b_i = 1 - b_i$. This network extends classical logic to fuzzy logic as shown by the following theorems. In each theorem we consider a single antecedent clause rule of the form If X is A then Y is B. [2] To sum it up, a neural network structure capable of implementing fuzzy logical inference has two different variations of this network

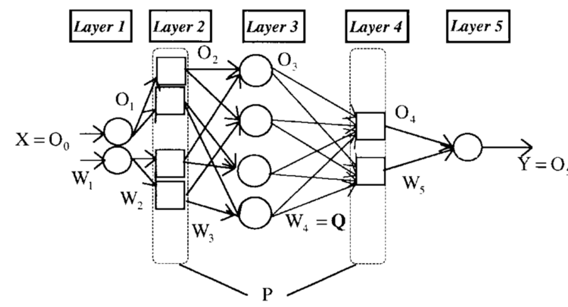


Fig. 3. The FALCON architecture

were developed and the theoretical properties examined. Both versions explicitly encode the knowledge of the rule in the weights of the clause checking and output layers. This is also a theorem which can be useful in linguistic memory by the changing of the weights of the network.

## IV. Neuro fuzzy systems

Fusion of Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS) are both types of adaptive intelligent systems to solve the real world problems. ANN learns from scratch by adjusting the interconnections between layers, while FIS is a computing framework based on the concept of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. The biggest advantage of FIS is the learning capability and the formation of linguistic rule base is the advantage of ANN. There is a cooperative model as a preprocessor where ANN learning mechanism determines the FIS membership functions or fuzzy rules from the training data. In this model, ANN assists the FIS continuously to determine the required parameters especially if the input variables of the controller cannot be measured directly. In a neuro fuzzy architectures, ANN learning algorithms are used to determine the parameters of FIS. [3] There is a popular architecture for linguistic computations called Falcon, which has a five-layered architecture and two linguistic nodes for each output variable. One is for training data (desired output) and the other is for the actual output. The architecture is shown on figure 3. [4] The first hidden layer does the fuzzification of each input variable, which means it generates the data into fuzzy sets. The second defines the preconditions of the rule followed by rule consequents in the third hidden layer. There is a different method like Falcon, which is also used in linguistic, which is called Generalized Approximate Reasoning based Intelligent Control (GARIC). What implements a neuro-fuzzy controller by using two neural network modules where the connections are not weighted. GARIC has 3 hidden layers like Falcon and the fuzzification happens in the last hidden layer. The first hidden layer is for variables and the second represent the fuzzy rules. These two method Falcon and GARIC are two ways to compute linguistic things with fuzzy neuro networks.

## V. Neuro–Fuzzy Rule Generation: Survey in Soft Computing Framework

In this section, we will discuss a survey of neuro-fuzzy rule generation algorithms. We will introduce the artificial neuron network (ANN), an overview of neuro-fuzzy hybridization, the rule generation algorithm in neuro-fuzzy networks, and knowledge-based networks.

The artificial neural network is a computational model based on the structure and functions of biological neural networks. It attempts to equip machines with some of the cognitive abilities that biological organisms possess. Unexplained behavior of the network is the most crucial problem of ANN. When ANN produces a probing solution, it does not give a clue as to why and how. Rule generation from artificial neural networks used to extract information from the network. Fuzzy sets are an aid in providing this information in a more human-comprehensible. Both fuzzy systems and ANN's are soft computing approaches to modeling expert behavior by using reinforcement learning or supervised learning.

Both neural networks and fuzzy systems are dynamic, parallel processing systems that estimate input-output functions. Hayashi and Buckley [5] proved that 1) any rule-based fuzzy system may be approximated by a neural net, and 2) any neural net (feedforward, multilayered) may be approximated by a rule-based fuzzy system. This kind of equivalence between fuzzy rule-based systems and neural networks is also studied in [6]–[7]-[8].

Fuzzy systems can be categorized into two categories, IF-THEN type, and Sugeno-type systems. The first category includes linguistic models based, which provides a solution with more readability, and the second is more suitable for a more precise solution.

The relation between neural networks and linguistic knowledge is bidirectional. Therefore, a neural network-based classification system can be trained by numerical data and linguistic knowledge, and a fuzzy rule-based classification system can be designed by linguistic. The neural systems are treated in a numeric, quantitative manner, whereas fuzzy systems are treated in a symbolic qualitative manner. Therefore, the integration of neural and fuzzy systems leads to a symbiotic relationship in which fuzzy systems provide a powerful framework for expert knowledge representation, while neural networks provide learning capabilities and exceptional suitability for computationally efficient hardware implementations.

Neuro-fuzzy hybridization is done in two ways: a neural network equipped with the capability of handling fuzzy information (FNN) and a fuzzy system augmented by neural networks to enhance some of its characteristics like flexibility, speed, and adaptability (NFS). In general, these methodologies can be broadly categorized as follows Note that categories 1 and 3–5 relate to FNN's, while category 2 refers to NFS [9].

1) Incorporating fuzziness into the neural net framework: fuzzifying the input data, assigning fuzzy labels to the training samples, possibly fuzzifying the learning procedure, and obtaining neural network outputs in terms of fuzzy sets
2) Designing neural networks guided by fuzzy logic formalism: designing neural networks to implement fuzzy logic and fuzzy decision-making, and to realize membership functions representing fuzzy sets.
3) Changing the basic characteristics of the neurons: neurons are designed to perform various operations used in fuzzy set theory (like fuzzy union, intersection, aggregation) instead of the standard multiplication and addition operations.
4) Using measures of fuzziness as the error or instability of a network: the fuzziness or uncertainty measures of a fuzzy set are used to model the error or instability or energy function of the neural network-based system.
5) Making the individual neurons fuzzy: the input and output of the neurons are fuzzy sets, and the activity of the networks involving the fuzzy neurons is also a fuzzy process.

A fuzzy model, containing a large number of IF-THEN rules, is liable to encounter the risk of overfitting and, hence, poor generalization. Genetic Algorithms have been utilized in fuzzy–genetic hybridization to circumvent this problem.

One of the significant problems in neuro-fuzzy design is the choice of the optimal network structure. One of the significant problems in neuro-fuzzy design is the choice of the optimal network structure. The models are generally very data-dependent, and the appropriate network size also depends on the available training data. Some different knowledge-based networks for rule generation are as follows.

1) Connectionist Models
2) Incorporating Fuzzy Sets
3) With Recurrent Networks
4) Incorporating Genetic Algorithms
5) Incorporating Rough Sets

## VI. NEFCLASS- A Neuro-Fuzzy Approach for the Classification of Data

In this section, we will discuss a neuro-fuzzy system for the classification of data (NEFCLASS) [10]. This approach has been based on the generic model of a fuzzy perceptron, which can be used to derive fuzzy neural networks or neural fuzzy systems for specific domains. It can be used as a common base for different neuro-fuzzy architectures. The goal of this system is to derive fuzzy rules from a set of data that can be separated in different classes. The task of the NEFCLASS model is to discover fuzzy rule (IF-THEN) and to learn the shape of the membership functions.

This model has the usual multilayer architecture, but the weights, the net inputs, and the activation of the output units are modeled as a fuzzy set. This fuzzy perception model is a 3-layer feedforward neural network. This model can be interpreted in a linguistic rule form, and it is the advantage of this model. This model has been compared with (neuro-fuzzy controller) NEFCON model, in which both neuro-fuzzy approaches have the same underlying model.

The presented model derives fuzzy rules from data to classify patterns into a number of (crisp) classes. NEFCLASS uses a supervised learning algorithm based on fuzzy error backpropagation that is used in other derivations of the fuzzy perceptron.

NEFCLASS means Neuro-Fuzzy Classification and is used to determine the correct class or category of a given input pattern. The pattern values are represented by fuzzy sets, and classification is described by a set of linguistic rules. For each input feature, there are fuzzy sets, rule base contains, and fuzzy rules. The fuzzy sets and the linguistic rules have been obtained from a set of examples by learning. A NEFCLASS system can be built from a combination of knowledge and learning, or it can be created from scratch by learning. The user has to define a number of initial fuzzy sets of the input features. Each fuzzy set is labeled with a linguistic term.

The fuzzy sets of those connections leading to the same rule unit R are called the antecedents of R. if rule created for a given input pattern of fuzzy sets is not identical to the antecedents of an already existing rule, and if the permitted number of rule units is not yet reached, a new rule node is created. After rule generation, the learning algorithm will adapt the triangular membership functions of the antecedents.

NEFCLASS was tested on the IRIS data set, and the performance was satisfactory compared to usual neural networks.

## VII. Neuro-fuzzy method to learn fuzzy classification

Classification is a tool in data science which can be used with any data related to linguistic, for example spoken audio data, hand-written picture data or even digital written data. We have already mentioned about neuro fuzzy systems in section IX. In this section we will discuss about NEFCLASS (neuro-fuzzy classification) model, where the fuzzy rules describing the data are of the form:
if $x_1$ is $\mu_1$ and $x_2$ is $\mu_2$ and ... and $x_n$ is $\mu_n$ then pattern $(x_1, x_2, ...x_n)$ belongs to class C, where $\mu_i$ are fuzzy sets. And the task of the model is to discover these rules by learning. The pattern feature values are represented by fuzzy sets, and the classification is described by a set of linguistic rules. [11]
The architecture of the NEFCLASS model is like NEF-PROX, mentioned in section IX. The architecture is shown on figure 4. [11] A NEFCLASS system can be build from partial knowledge about the patterns, and can be then refined by learning, or it can be created from scratch by learning. The learning algorithm of the membership functions uses an error measure that tells, whether the degree of fulfillment of a rule has to be higher or lower. This information is used to change the input fuzzy sets. The model's learning process is shown on figure 5. [11] So in every iteration the space is closing for each classes in the dataset, where the grid is created by overlapping fuzzy sets. During the learning process every box is selected if there is a pattern has membership with $R_i$ rule. Of course there could be a misclassification and error. The error is propagated from the output units towards the input units. Duo to the learning algorithm

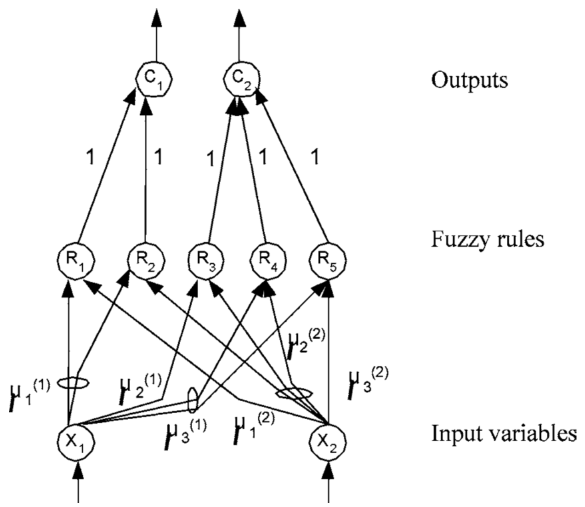


Fig. 4. NEFCLASS architecture

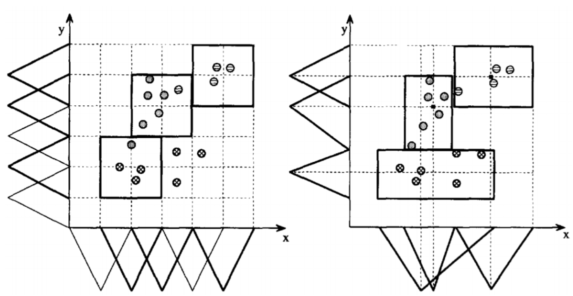


Fig. 5. The NEFCLASS model's learning process

modifies the fuzzy sets directly, the fuzzy sets can be more easily labeled with linguistic expressions.

To sum it up, NEFCLASS is a fuzzy and neuro based model which is likes the model mentioned in the section IX, but the purpose is the classification of a dataset. In the network the connection weights determine the classes. As every neuro-fuzzy system this could be used in cognitive memory, like in linguistic. The usage of this could be for example making an app which categorizes spoken language and finds out something from data, like mood or diseases.

## VIII. Fuzzy logic controllers

In the III we discussed about how the neural networks can works with fuzzy logic and with fuzzy sets data. In this section we will talk about what is a controller, what is a fuzzy logic controller and how can it be used in linguistic memory. By using fuzzy networks rather than just general neural structures, it reduces the amount of learning that is necessary. And the modularisation of the fuzzy neural network system allows for simpler learning processes as well as supporting off line learning

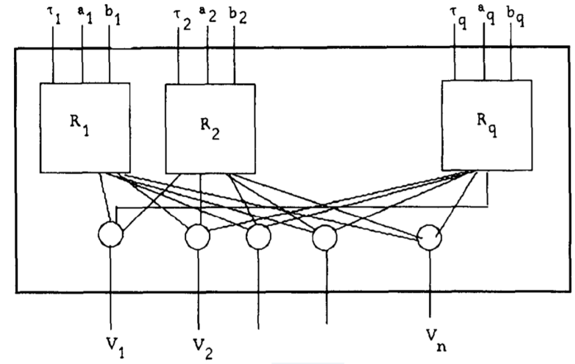


Fig. 6. Knowledge base of the fuzzy controler

of membership grades. [12] The basic model used in fuzzy controllers has a plant and a fuzzy logic controller, where the output is the feedback branch of the controller and it intervenes to the input. The fuzzy logic controller can be seen to be made up of two components, the *Control rules and inference mechanism* and *Defuzzier*, which does the inverse of the fuzzification. If the clause of $V_1$ is $A_1$ and $V_2$ is $B_1$, then U is D, then $V_1$ and $V_2$ are variables corresponding to inputs to the control unit (outputs of the plant) and U is a variable corresponding to an output variable of the controller (an input to the plant). $A_1$, $B_1$, and $D_1$ are linguistic values for the corresponding variable which are represented as fuzzy subsets over the appropriate spaces. [12] We can say that the linguistic vales used in fuzzy controllers. These three values are fuzzy versions of the real numbers. The value of D is the input of the defuzzifier unit of the fuzzy controller, and the output of it is the actual output value. So if the controllers inputs are

$$V_1 = x_0, V_2 = y_0$$

then the inference is for every $R_i$ we calculate the degree to which the rule is satisfied, so

$$\tau_j = min(A_j(x_0), B_j(y_0))$$

Calculate the linguistic value of the control variable U as the fuzzy subset D, where for each Z in the base set Z of U we have

$$D(z) = max_j[min(\tau_j, D_j(z))]$$

And the D value become the input of the defuzzifier unit. This is the main idea of a fuzzy controller and the reason is we are discussing it to learn how useful it in linguistic computations. Each linguistic value will be represented by a separate neural module. The actual knowledge base of the fuzzy controller is made up of the collection of

rules. Figure 6 shows the construction of the complete knowledge base. [12] The output is the knowleddge base are $\tau_j, a_j, b_j$, provides, for its associated rule, the degree to which it fired $\tau_j$ and a range $[a_j, b_j]$ of possible values for the output U. This output gois into the defuzzifier unit and it converts it into real output.
So in conclusion if we would like to implement a fuzzy controller, we must implement a neural network for the knowledge base of the controller. The other part of the controller is the defuzzifier which makes excact output what we can use as a feedback signal for the plant. This is the fuzzy controller circle, which can be used for cognitive knowledge, as the fuzzy systems was invented for.

## IX. Neuro-fuzzy systems for function approximation

In this section we discuss about a neuro-fuzzy architecture for function approximation based on supervised learning, which is presented in [13].

### A. About the neuro fuzzy systems

Neuro-fuzzy is described by the following five points:

1) Trained by a learning algorithm, the learning process is not knowledge based, but data driven.
2) A neuro-fuzzy system can be viewed as a special 3-layer feedforward neural network. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables.
3) It is possible to create the system from scratch, and it is also possible to initialize it by prior knowledge in form of fuzzy rules.
4) The learning procedure of a neuro-fuzzy system takes the semantical properties of the underlying fuzzy system into account.
5) A neuro-fuzzy system approximates an n-dimensional function that is partially given by the training data. [13]

### B. Architecture

In [13] paper discussing a general approach to function approximation by a neuro-fuzzy model based on plain supervised learning. A NEFPROX system is a special 3-layer fuzzy perceptron, shown on figure 7. The input variables are $x_1, x_2, ...x_n$, the units of the hidden layer $R_1, R_2, ...R_n$ represents the fuzzy rules and the output variables are $y_1, y_2, ...y_n$. Each connection between units is labeled linguistic term $A_{k_r}^{(i)}$ between x and R, $B_{k_r}^{(i)}$ between R and y. The connections which coming from

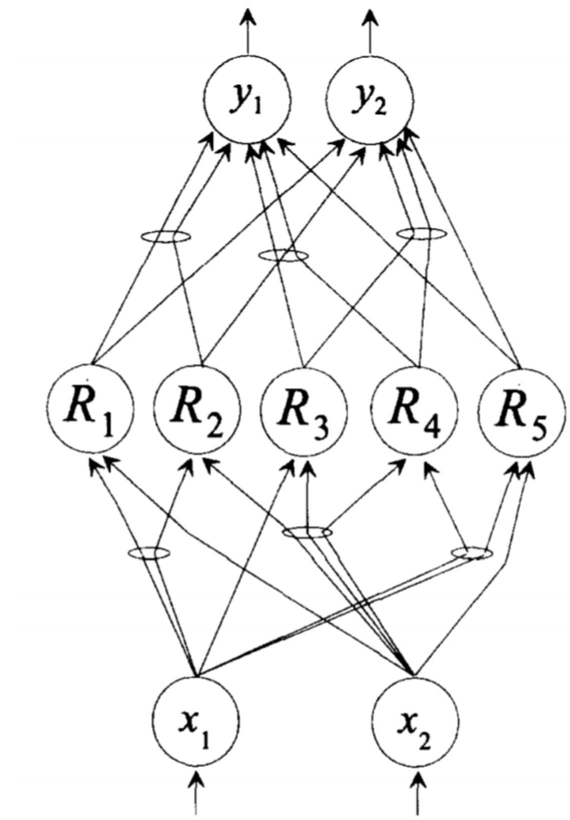


Fig. 7. NEPROX 3-layer architecture

the same input have the same fuzzy weight either. $L_{x,R}$ is the label of the connection between the input and the rule. Each hidden unit represents a fuzzy if-then rule.

### C. Learning

A neuro-fuzzy method to be a tool for creating fuzzy systems from data. The learning result should also be interpreted, and the insights gained by this should be used to restart the learning procedure to obtain better results if necessary. The learning algorithms should be constrained such that adjacent membership functions do not exchange positions, do not move from positive to negative parts of the domains or vice versa, have a certain degree of overlapping. [13]
In the beginning of the learning process, we must specify initial fuzzy partitions for each input variable, but not necessary for the output variables. The learning algorithm selects patterns from the input and the target data and create new rules for the connections.

### D. Usage in linguistic

The NEFPROX model is a fuzzy and neural network based learning system, offering a method for structure learning. Fuzzy and neural network systems are similar to cognitive scheme as people think and do all day things. This system is optimized for approximation calculations which is also can be used in linguistic. For example if we would like implement a system to find out keywords from spoken language, this model could be a good start.

## X. POPFNN: A Pseudo Outer-Product Based Fuzzy Neural Network

In this section, we will discuss a novel fuzzy neural network, called the pseudo-outer-product-based fuzzy neural network (POPFNN) [14]. The integration of fuzzy systems and neural networks combines the human inference style and natural language description of fuzzy systems with the learning and parallel processing of neural networks. Fuzzy neural networks are hybrid systems that possess the advantages of both neural networks and fuzzy systems. In addition, hybrid systems alleviate the shortcomings of the respective techniques. The problems such as the determination of the membership functions and the operation of fuzzy inferences in fuzzy systems can be resolved in these hybrid systems. On the other hand, the integration of fuzzy concepts in fuzzy neural systems greatly improves the transparency of neural networks. There are numerous approaches to integrate fuzzy systems and neural networks.

In general, there are two primary information sources for the construction of fuzzy neural systems, human beings for providing linguistic descriptions and sensors for providing the numerical measurement of variables. The fuzzy neural networks can be divided into two classes according to the information sources aspect. The first class is fuzzy neural networks based on linguistic information, in which neural network techniques are used for the network adjustment. This class has some drawbacks, which are described in [14]. The other class is fuzzy neural networks based on numerical information, in which neural network techniques are used for the network adjustment. This class of fuzzy neural networks is similar to the previous one class, but the initial value of parameters and structures are not derived from linguistic information. Instead, they are achieved by an unsupervised learning algorithm from a set of training data. This class is suitable in applications where there are direct observations from the system, but there are not expert opinions.

We can divide rule-identification algorithms into two categories. The first group is based on knowledge elicitation, and fuzzy IF-THEN rules are obtained from human experts. This approach is used in the first class of fuzzy neural networks. The other is based on neural network techniques, and fuzzy rules are achieved from unsupervised learning algorithms such as self-organizing and competitive learning algorithm. This approach is used in the second class of fuzzy neural networks.

The proposed POPFNN belongs to the second class. It is based on the truth value restriction method, and a novel pseudo-outer-product (POP) learning algorithm for rule identification is used, which is falls under the second approach. The POP learning algorithm has many advantages in performance since it is a single-pass learning algorithm. A detailed description of the POP learning algorithm has been presented in [14]. The learning process consists of three phases.

- The membership function initialization has been done in the first phase, self-organizing algorithm.
- The fuzzy rule identification has been performed in the second phase, the POP learning algorithm.
- Fine-tuning has been done in the last phase, the backpropagation learning algorithm.

The proposed POPFNN architecture is a five-layer neural network. Each layer in POPFNN performs a specific fuzzy operation. Fuzzification of the input data and defuzzification of the output data are respectively performed by the input and output linguistic layers, while the fuzzy inference is collectively performed by the rule-base and the consequence layers.

## XI. POPFNN-AARS(S): A Pseudo Outer-Product Based Fuzzy Neural Network

In this section, we discuss a novel fuzzy neural network, the pseudo-outer-product-based fuzzy neural network using the singleton fuzzifier together with the Approximate Analogical Reasoning Schema (singleton fuzzifier POPFNN–AARS) [15].

The singleton fuzzifier POPFNN-AARS employs the AARS instead of the commonly used Truth Value Restriction (TVR) method. In the previous proposed POPFNN-TVR method, implication rules are used to derive the truth values of the consequences from the truth values of the conditions. This process is essentially based on the matrix operation, which has caused the inference process of POPFNN-TVR is conceptually unclear. The singleton fuzzifier POPFNN–AARS is proposed to solve this issue. This avoids the computational complexity of the TVR method.

The whole fuzzy inference process of using AARS is based on some modification methods. Given an observed fact A0 and a simple fuzzy rule "if A then B;" the basic idea of AARS is to modify the consequence B of the fuzzy rule according to the closeness of the observed fact A0 to the antecedent A. Once the similarity between A and A0 exceeds the value of the threshold; the fuzzy rule is fired. An MF is subsequently constructed and is used to modify the consequence B of the fuzzy rule to deduce a conclusion B0.

The SM in fuzzy sets is based on distant metric and obtained from a distance measure (DM). Then, a correlation analysis is used to assess the quality of distance measures. For distinguishing between degrees of similarity or dissimilarity, the five measures including Disconsistency Measure, Hausdorff Measure 1, Hausdorff Measure 2, Kaufman, and Gupta Measure, Kaufman and Gupta Measure are found to have relatively satisfactory performances.

The MF is dependent on SM, and its construction is subjective. That is, the form of MF based on the expert's experience or historical data can be adjusted. In the main article [15], two forms of MF have been introduced, Expansion form (More or Less form) and Reduction form. It should be noted that if the results are unsatisfactory, another SM, and/or threshold value adjustment, and/or another MF can be chosen.

Both AARS and the TVR method use truth degree to deduce the conclusion, but they have different ways of representing truth degrees. In AARS, truth degree is represented by a real value—that is the similarity measure. While in the TVR method, truth degree is represented by a fuzzy set—that is the truth value. This difference makes the inference process of AARS simpler but decreases the accuracy of the system.

The structure of the singleton fuzzifier POPFNN-AARS comprises five layers, including the input layer, condition layer, rule-based layer, consequence layer, and output layer. The similarity measures between the fuzzified inputs and the input labels are derived at the second layer, and the MF are performed in the third layer. Between the five similarity measures, only Disconsistency Measure, Hausdorff Measure 1, and Hausdorff Measure 2 are suitable for this structure.

Derivation of the Singleton Fuzzifier POPFNN-AARS Model from a Singleton Fuzzifier AARS Fuzzy Rule-Based System is done in 5 steps. The first step is Fuzzification, in which each input element is fuzzified by using a singleton fuzzifier. In step two, the SM's are derived by using different DM's. The overall similarities for the rules are determined in step 3, and a threshold is applied to each fuzzy rule. In step four, the consequences of firing the fuzzy rules are derived using the MF. In the last step, the final consequences derived at the output labels are defuzzified with a modified center average defuzzifier.

The learning algorithms consists of three stages.

1) the unsupervised learning, the process for initialization
2) The pseudo-outer-product (POP) learning for rule identification
3) The backpropagation learning for fine-tuning.

The results of performance evaluation have been indicated that the singleton fuzzifier POPFNN-AARS using the reduction form of the MF generally have better performances than the expansion form of the MF.

## XII. POPFNN-CRI(S): Pseudo Outer Product Based Fuzzy Neural Network Using the Compositional Rule of Inference and Singleton Fuzzifier

In this section, we will discuss a pseudo-outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier [POPFNN-CRI(S)] [16]. Fuzzy neural networks are hybrid systems that possess the advantages of both neural networks and fuzzy systems. The integration of fuzzy systems and neural networks combines the human inference style and natural language description of fuzzy systems with the learning and parallel processing of neural networks. There are numerous approaches to integrate fuzzy systems and neural networks. The proposed POPFNN-CRI(S) is developed on the basis of possibility theory and the fuzzy compositional rule of inference. The singleton POPFNN-CIR(S) employs the Compositional Rule of Inference (CRI) instead of the commonly used Truth Value Restriction (TVR) method.

The information to be stored in a database is not always precise and certain. Information about attribute values maybe even missing for objects of interest. Fuzzy set and possibility theory offers a unified framework which enables us to provide a solution both to the handling of flexible queries and to the management of imprecise and uncertain information.

Membership functions were introduced by Zadeh, who suggested the use of membership functions in the theory of fuzzy sets. A fuzzy set A is an extension of the idea of a regular set viewed in terms of the membership function. It describes a subpart of a universe X whose boundaries are not strictly defined, and a grade of membership (valued in [0,1]) is attached to each element of X. So, there is a gradual transition between full membership and exclusion, and it is then possible to have a better representation of gradual properties, vague classes, and approximate descriptions. As a consequence, if we consider a precise value x in X, it is possible to estimate the extent to which x is compatible with the concept represented by fuzzy set A. The membership function characterizes a possibility distribution of simple propositions. The possibility distribution of composed propositions is derived from the possibility distributions of component propositions by operations using a triangular norm, a triangular co-norm, and the negation connectives of fuzzy logic, which the corresponding notations are provided.

The proposed POPFNN-CRI(S) architecture for a multi-input multi-output (MIMO) system is a five-layer neural network, which is shown in [16]. Each layer in POPFNN-CRI(S) performs a specific fuzzy operation. Fuzzification of the input data and defuzzification of the output data are respectively performed by the input and output linguistic layers, while the fuzzy inference is collectively performed by the rule-base and the consequence layers.

The learning process of POPFNN-CRI(S) consists of two phases. On is the fuzzy membership learning, and the other one is POP learning. Similar to the POPFNN-TVR architecture, a self-organizing type of learning algorithm is employed in the first phase to determine the membership functions of the condition and consequence layers. The only difference between the learning process of CRI and TVR methods is that the former only requires two phases of learning, while the latter requires additional supervised learning to adjust the membership functions. A detailed description of the two-phase learning process, known as pseudo fuzzy Kohonen partition (PFKP) and fuzzy Kohonen partition (FKP) for the CRI method, has been presented in [16].

Experimental results have been provided to test the performance of the proposed POPFNN-CRI(S). The results have been indicated that the proposed POPFNN-CRI(S) has better performances comparing to the other techniques.

## XIII. Structure and Learning Algorithms of a Nonsingleton Input Fuzzy Neural Network Based on the Approximate Analogical Reasoning Schema

In this section, we will discuss the pseudo-outer-product-based fuzzy neural network using the nonsingleton fuzzifier together with the approximate analogical reasoning schema (nonsingleton fuzzifier POPFNN–AARS) [17].

The previous structures and learning algorithms of POPFNN–TVR and the singleton fuzzifier POPFNN–AARS [18], have reduced the learning process from 3-phase to 2-phase learning, which corresponds to the derivation of fuzzy sets and the identification of the fuzzy rules using the POP learning algorithm. Similar to most existing fuzzy neural systems, both POPFNN–TVR and the singleton fuzzifier POPFNN–AARS uses the singleton fuzzifier to fuzzify its crisp inputs into fuzzy sets. However, because of noise and inappropriate extraction methods, the inputs to the system usually are imprecise and cannot satisfy the crisp nature of the singleton. A novel fuzzy neural network called the pseudo-outer-product based fuzzy neural network using the nonsingleton fuzzifier together with the approximate analogical reasoning schema (nonsingleton fuzzifier POPFNN–AARS(NS)) has been proposed to solve this problem.

The AARS is an alternative to CRI and TVR methods. The proposed nonsingleton fuzzifier POPFNN–AARS(NS), has a similar structure and notations as to the singleton fuzzifier POPFNN–AARS(S). It comprises five layers, including the input layer, condition layer, rule-based layer, consequence layer, and output layer. The similarity measures between the fuzzified inputs and the input labels are derived at the second layer. The process of a nonsingleton fuzzifier fuzzy rule-based system based on the AARS fuzzy is consists of five steps.

- The first step is fuzzification, in which each input element is fuzzified by using a nonsingleton fuzzifier.

- In the second step, the fuzzified inputs are used to compare against their corresponding input labels, which, based on the comparison results, the similarity measures are derived by using different distance measures.
- In the third step, the overall similarities for the rules are determined, and a threshold is applied to each fuzzy rule.
- In the fourth step, the consequences of firing the fuzzy rules are derived using the modification function.
- In the fifth step, the final consequences derived at the output labels are defuzzified with a modified center average defuzzifier.

The process of learning algorithms is consists of three stages.

- The first stage is the unsupervised learning process for initialization.
- The second stage is the pseudo-outer-product (POP) learning for rule identification.
- The third stage is supervised learning for fine-tuning.

Three real-life data samples have been provided to test the performance of the nonsingleton fuzzifier POPFNN–AARS [[17] Sec. 6]. From the results can be observed that when some imprecise, noisy data samples are involved, the nonsingleton fuzzifier POPFNN–AARS tends to have a superior performance over the singleton fuzzifier POPFNN–AARS. Also, the results indicate that the modified supervised learning algorithm for finding the best value for the threshold is trapped in a certain local minimum.

## XIV. A BRAIN-INSPIRED PSEUDO-INCREMENTAL ENSEMBLE ROUGH SET PSEUDO-OUTER PRODUCT FUZZY NEURAL NETWORK

In this section we talk about a pseudo-incremental ensemble rough set pseudo-outer product fuzzy neural network (PIE-RSPOP), what is incorporating the theories of learning and memory encoding in human brain. While a normal FNN system needs much time and the complexity could be huge, a PIE-RSPOP model creators try to reduce it with a brain-inspired model. PIE-RSPOP assumes that the active memory is finite, though a practically infinite cache is available for storage. This makes PIE-RSPOP reduce the computational while using more complex networks and calculate with relative old knowledge too. But the number of maximum networks in active memory and stored data are limited. PIE-RSPOP model has an active and a cache memory too. There are rules for forgetting the old data optimized if there is an actual task what is good with an old data, it turns up. Before a new network is created, RS based attribute reduction is performed for the current network using only this data store. [19] In this model, each network represents a concept, which is based on fuzzy rules. This model represents 5 layer

1) The layer 1 is the input layer where $x$ is the input vector,
2) The layer 2 stores the fuzzy linguistic variables or the fuzzy labels $X_{i,j}$,
3) The layer 3 stores fuzzy rules $R_k$,
4) The layer IV is the layer of fuzzy labels $Y_{m,ml}$ of the output attributes, RSPOP and POPFNN are implemented here,
5) Layer 5 is the output layer, the output vector is $y$.

In the POP-FNN models the total number of possible rules is equal to the total number of combinations of the input attributes' membership functions. So this model is designed to incorporate extensively the concepts of learning and memory encoding in the human brain to enhance the accuracy of incremental learning.

## XV. THE POP LEARNING ALGORITHMS: REDUCING WORK IN IDENTIFYING FUZZY RULES

In this section, we will discuss a Pseudo-Outer Product based Fuzzy Neural Network (POPFNN), and its two fuzzy-rule-identification algorithms, the Pseudo-Outer Product (POP) learning and the Lazy Pseudo-Outer Product (LazyPOP) learning algorithms [20]. The integration of fuzzy systems and neural networks combines the human inference style and natural language description of fuzzy systems with the learning and parallel processing of neural networks. Fuzzy neural networks are hybrid systems that possess the advantages of both neural networks and fuzzy systems. Also, hybrid systems alleviate the shortcomings of the respective techniques. The problems such as the determination of the membership functions and the operation of fuzzy inferences in fuzzy systems can be resolved in these hybrid systems. On the other hand, the integration of fuzzy concepts in fuzzy neural systems dramatically improves the transparency of neural networks.

Rule-identification algorithms can be divided into three categories.

1) The first approach is based on linguistic information.

2) The second approach uses unsupervised learning algorithms.
3) The last approach uses a supervised learning algorithm (particularly the back-propagation technique) to identify fuzzy rules in the neural networks.

The proposed POPFNN uses two novel learning algorithms; POP learning and the LazyPOP learning algorithms instead of commonly used self-organizing and competitive learning algorithms. The proposed learning algorithms have many advantages in performance, such as being fast, reliable, efficient, and easy to understand. The POP learning algorithm is a simple one-pass learning process. The second algorithm, the LazyPOP learning algorithm, truly identifies the fuzzy rules that are relevant, and despite the POP learning algorithm, it is not based on a rule-selection method. So, it is possible to adjust the structure of the fuzzy network by deleting invalid feature inputs according to the identified fuzzy rules.

The learning process of POPFNN consists of three phases.

1) The membership function initialization has been done in the first phase, self-organizing algorithm.
2) The fuzzy rule identification has been performed in the second phase, the POP or LazyPOP learning algorithm.
3) Fine-tuning has been done in the last phase, the back-propagation learning algorithm.

The proposed POPFNN architecture is a five-layer neural network, and the layers are as follows.

1) The input layer
2) The condition layer
3) The rule-base layer
4) The consequence layer
5) The output layer.

Each layer in POPFNN performs a specific fuzzy operation. The linguistic nodes (such as width and height) are defined in the first layer. The input-label nodes (such as 'small' and 'large') are represented by using the membership function in the condition layer. The IF-THEN fuzzy rules are defined in the third layer. The output-label nodes (such as 'small' and 'large') are represented by the consequence layer. Defuzzification of the output data is performed in the output layer.

The POP learning algorithm needs the training data to identify all the relevant fuzzy rules correctly. In this method, the weights of the links between the rule-base layer and the output layer are initially set to zero. After the proposed POP learning process, these weigh represent the strengths of the fuzzy rules having the corresponding output label nodes. Then the link with the highest weight is chosen, and the others are deleted. On the other hand, when all the weights are very small or almost equal to each other, all links can be deleted, and this implies that the represented rule is irrelevant. Consideration of all the possible rules is the drawback of this method. To resolve this issue, the LazyPOP learning algorithm is developed. It is the same as the POP learning algorithm, but two criteria have been applied. The input linguistic variables that have little or no relation with the output linguistic variables are deleted in this algorithm. Also, the output linguistic variables that have little or no relation with the output linguistic variables are marked.

## XVI. A Novel Yager-Based Fuzzy Neural Network With the Discrete Clustering Technique

In this section we discuss about discreet classification and novel yager-based fuzzy neural networks. A novel pattern classifying network called the Yager fuzzy neural network. The Yager pattern classifying fuzzy neural network is a novel neurolinguistic approach to pattern recognition using the Yager inference rule and a supervised clustering algorithm. [21] In a Yager neural network the inputs and outputs are fuzzy sets of discrete points pairs and their membership level. With this method it is not necessary to just four parameters to define the fuzzy sets, because there is no need to save memory space.

The discrete clustering technique use a neural network architecture the first layer consisting a single neuron, the middle layer consisting $c$ neurons, where $c$ is the number of the clusters, and the last layer (membership layer) consisting $m$ neurons, where $m$ is the number of evenly spaced discrete points in the sample space. The fuzzy membership functions are stored in the weights between the connections of the middle layer and the input layer. The weights between the middle and the output layer are represents the membership grade in the fuzzy set definition.

The model has been tested with the Iris data, what contains 50 sample for each classes of iris subspecies (setosa, versicolor, virginica). The number of clusters was set to 3.

The algorithm uses the pure clustering behavior of the input data to form the membership functions. As with the supervised learning techniques, to preserve the shape of the Gaussian, trapezoidal, and triangular membership functions, a lot of discrete sample points would have to be used for a Yager neural network. The UDCT algorithm uses just 15 discrete points to generate the membership functions. [21]

## XVII. GenSoFNN: A Generic Self-Organizing Fuzzy Neural Network

In this section, we will discuss a novel neural fuzzy system, the generic self-organizing fuzzy neural network (GenSoFNN), which has a strong noise tolerance capability by employing a new clustering technique known as discrete incremental clustering (DIC) [22].

The main advantage of a neural fuzzy network is its ability to model a problem domain using a linguistic model (a set of IF-THEN fuzzy rules) instead of complex mathematical models. In addition, the black-box nature of the neural network is resolved. Moreover, a neural fuzzy network can self-adjust the parameter of the fuzzy rules using neural-network-based learning algorithms.

In general, neural fuzzy systems can be classified into two groups. The first group is essentially fuzzy systems with self-tuning capabilities and requires an initial rule base to be specified prior to training. The second group of neural fuzzy networks, on the other hand, is able to automatically formulate the fuzzy rules from the numerical training data without needing to initial rule base. Since both groups have some deficiencies, this novel neural fuzzy system has been proposed.

A consistent rule base for the knowledge interpretation and the choice of clustering techniques is important in neural fuzzy networks. Rule extraction will be meaningless and/or obscure if rule base has not been defined properly. Different clustering techniques have been proposed in some references, but they have weak resistance to noisy/spurious training data or require prior knowledge.

The GenSoFNN network automatically formulates the fuzzy rules from the numerical training data and maintains a consistent rule base. The GenSoFNN network employs a new clustering technique known as DIC to enhance its noise tolerance capability. It is noted that clustering techniques may be classified into hierarchical-based and partition-based techniques. The main drawback of hierarchical clustering is that the clustering is static, and the main drawback of partition-based clustering techniques is the requirement of prior knowledge.

DIC [[22] Part. II Section. B] creates separate clusters for noisy/spurious data that have a poor correlation to the genuine or valid data and does not require prior knowledge of the number of clusters present in the training data set. In addition, the proposed GenSoFNN network does not require the predefinition of the number of fuzzy rules, as the rule formulation process is entirely data-driven.

GenSoFNN is suitable for on-line applications as its training cycle takes place in a single pass of the training data. The training cycle of the GenSoFNN network consists of three phases: self-organizing, rule formulation, and parameter learning. DIC methods for the self-organizing phase and backpropagation (BP) learning algorithm for the parameter tuning have been developed into the GenSoFNN network.

GenSoFNN-CRI[1](S) network is developed by mapping the CRI inference scheme onto the GenSoFNN structure. The CRI inference scheme provides a robust fuzzy logic theoretical foundation for the operations of the GenSoFNN-CRI(S) network.

## XVIII. A Novel Generic Hebbian Ordering-Based Fuzzy Rule Base Reduction Approach to Mamdani Neuro-Fuzzy System

The hybridization of neural networks and fuzzy systems take advantage of the low-level learning ability of neural networks and the high-level reasoning ability of the fuzzy systems. Neuro-fuzzy modeling usually comes with two contradictory requirements: interpretability and accuracy. The fuzzy modeling research is divided into two main areas: linguistic fuzzy modeling (LFM) for interpretability issue and precise fuzzy modeling (PFM) for accuracy issues. The interpretability usually can be improved by input variable selection and reduction, fuzzy rule selection and reduction, and use of other rule expressions.

An iterative process to reduce the rule base to derive good interpretability while maintaining a high level of modeling accuracy has been proposed [[23] Sec.2]. It is based on the Hebbian ordering of the rules and is called Hebb rule reduction. The Hebbian ordering indicates that rules with higher importance are more likely to be preserved. The least mean square (LMS) algorithm is used to tune the membership functions of the rule base after each time step of the reduction. Interpretability and accuracy are balanced using this iterative process.

The fuzzy neural network has five layers, the input layer, the condition layer, the rule-base layer, the consequence layer, and the output layer. The Gaussian membership function is used in the condition and consequence layers to perform the fuzzification and defuzzification.

The proposed method for creating a balance between interpretability and accuracy consists of three phases:

1) The initial rule generation
2) The iterative rule reduction and refinement
3) The membership function tuning

In the first phase, the fuzzy rules are formulated to cover all the training samples. During the rule initialization phase, whenever a new data sample is presented, if there are no rules in the rule base or if the strength of the rule with the largest firing strength is below a specified threshold, a new rule node will be created. An iterative tuning process consists of rule reduction, and membership function (MF) refinement is employed in the second phase. In the rule reduction, based on Hebbian ordering scheme, MFs are merged according to the degree of their overlap, and redundant and conflictive rules are deleted at the same time. This process has been performed in three steps as follows.

1) Rules ranking
2) MFs merging
3) The redundant and conflictive rules reduction

After that, the LMS algorithm is employed to tune the centers and widths of the membership functions to reduce the modeling error. The third phase is the fine-tuning of the MFs to achieve a high level of accuracy.

The proposed model has achieved superior performance compared to the other models, despite the fact that it uses a more extensive set of attributes. It has been concluded that the proposed Hebb rule reduction achieves better performance than RSPOP on both interpretability and accuracy.

The proposed Hebb rule reduction outperforms the other benchmarked models, which indicates its excellent capability for the classification task. The performance evaluation of the proposed model on the highway traffic flow density data set [[23] Sec 3.3] shows that the proposed method derives on the average only 8.1 fuzzy rules, which is lowest among others established neuro-fuzzy models, and its modeling accuracy is superior.

## References


[1] (Sep. 24, 2019). Fuzzy logic, Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Fuzzy_logic (visited on 09/29/2019).

[2] R. R. Y. James M. Keller and H. Tahani, “Neural network implementation of fuzzy logic,” *Department of Electrical Computer Engineering, University of Missouri-Columbia, Missouri 6521 I, USA*,

[3] A. Abraham, “Neuro fuzzy systems: State-of-the-art modeling techniques,” *School of Computing Information Technology Monash University*, Jun. 12, 2001.

[4] (Feb. 1999). The falcon architecture, ResearchGate, [Online]. Available: https : / / www . researchgate . net / figure / The - FALCON - architecture_fig1_3302577 (visited on 09/29/2019).

[5] Y. Hayashi and J. J. Buckley, “Approximations between fuzzy expert systems and neural networks,” *Int. J. Approx. Reas., vol. 10, pp. 63–73*, 1994.

[6] J. J. Buckley and Y. Hayashi, “Numerical relationship between neural networks, continuous functions and fuzzy systems,” *Fuzzy Sets Syst., vol. 60, no. 1, pp. 1–8*, 1993.

[7] J. J. Buckley, Y. Hayashi, and E. Czogala, “On the equivalence of neuralnets and fuzzy expert systems,” *Fuzzy Sets Syst., vol. 53, no. 2, pp. 129–134*, 1993.

[8] J. M. Benitez, J. L. Castro, and I. Requena, “Are artificial neural networks black boxes?” *IEEE Trans. Neural Networks, vol. 8, pp. 1156–1164*, 1997.

[9] S. Mitra and Y. Hayashi, “Neuro–fuzzy rule generation: Survey in soft computing framework,” *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 11, NO. 3*, 2000.

[10] D. Nauck and R. Kruse, “Nefclass- a neuro-fuzzy approach for the classification of data,” *Technical university of braunschweigh, Dep. of Computer Science*, 1995.

[11] R. K. Detlef Nauck*, “A neuro-fuzzy method to learn fuzzy classification rules from data,” *Faculty of Computer Science, Otto-von-Guericke University of Magdeburg, Universitaetsplatz 2, D-39106 Magdeburg, German*, Dec. 1996.

[12] R. R. Yager, "Implementing fuzzy logic controllers using a neural network framework," *Machine Intelligence Institute, Iona College, New Rochelle, USA*, Jan. 1991.

[13] R. K. Detlef Nauck*, "Neuro-fuzzy systems for function approximation," *Faculty of Computer Science, Otto-von-Guericke University of Magdeburg, Universitaetsplatz 2, D-39106 Magdeburg, German*, Feb. 1997.

[14] C. Quek and R. Zhou, "Popfnn: A pseudo outer-product based fuzzy neural network," *Neural Network, Vol. 9, No. 9, PP 1569-1581*, 1995.

[15] C. Quek and R. W. Zhou, "Popfnn-aars(s): A pseudo outer-product based fuzzy neural network," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 1999.

[16] K. K. Ang and C. Quek, "Popfnn-cri(s): Pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 2003.

[17] C. Quek and R. Zhou, "Structure and learning algorithms of a nonsingleton input fuzzy neural network based on the approximate analogical reasoning schema," *School of Computer Engineering, Nanyang Technological University*, Oct. 2003.

[18] C. Quek and R. Zhou, "Popfnn-aar(s): A pseudo outer-product based fuzzy neural network," *IEEE Trans Syst Man Cybern B Cybern*, Jan. 1999.

[19] C. H. Q. Aparna Ramesh Iyer Dilip K. Prasad, "Pie-rspop: A brain-inspired pseudo-incremental ensemble rough set pseudo-outer product fuzzy neural network," *School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore*, Nov. 22, 2017.

[20] C. Quek and R. Zhou, "The pop learning algorithms: Reducing work in identifying fuzzy rules," *School of Computer Engineering, Nanyang Technological University*, 1997.

[21] C. Q. A. Singh and S.-Y. Cho, "A novel yager-based fuzzy neural network with the discrete clustering technique," *IEEE*, Apr. 4, 2008.

[22] W. L. Tung and C. Quek, "Gensofnn: A generic self-organizing fuzzy," *IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 13, NO. 5*, 2002.

[23] F. Liu, C. Quek, and G. S. Ng, "A novel generic hebbian ordering-based fuzzy rule base reduction approach to mamdani neuro-fuzzy system," *School of Computer Engineering, Nanyang Technological University*, 2007.