Asian Journal of Computer and Information Systems (ISSN: 2321 – 5658)
Volume 02 – Issue 02, April 2014

Types of Expert System: Comparative Study

Viral Nagori, Bhushan Trivedi

GLS Institute of Computer Technology (MCA), India
Email: viral011 {at} yahoo.com

ABSTRACT--- ***The paper describes the different classifications of expert systems. There are mainly five types of expert systems. They are rule based expert system, frame based expert system, fuzzy expert system, neural expert system and neuro-fuzzy expert system. We discussed the expert systems based on their knowledge representation, inference engine, working of the system and user interface. At the end, we provided comparative study of above five types of expert systems based on different parameters. We can say that choice of an expert system depends on the domain requirements.***

Keywords: rule based expert system, fuzzy expert system, frame based expert system, neural expert system, neuro-fuzzy expert system

---

1. INTRODUCTION

Expert system is a very special branch of Artificial intelligence that makes extensive use of specialised knowledge to solve problem at the level of human expert. There are different types of expert systems. They are rule based expert system, fuzzy expert system, frame based expert system, and hybrid expert systems. Hybrid expert system is the combination of two or more types of intelligent systems. Prominently, there are two types in hybrid expert systems. The first one is neural expert systems and the second one is neuro-fuzzy systems. Neural expert system combines the features of rule based expert system along with neural network features. While neuro-fuzzy expert system combines the features of fuzzy logic along with the features of neural network. Here we are going to study and provide analysis among different types of expert systems.

2. RULE BASED EXPERT SYSTEM

As the name suggests, rule based expert system consists of set of rules. Rule is an expressive, straight forward and flexible way of expressing knowledge. In a rule based expert system, knowledge is represented as a set of rules. Knowledge is a theoretical or practical understanding of a subject or a domain. (Negnevitsky, Artificial Intelligence: A guide to Intelligent systms, 2008) Expert possesses deep knowledge and practical experience over the years which results into expertise. Expert has an ability to code the knowledge in form of rules. Any rule consists of two parts: The IF part, called and antecedent (premise or condition) and THEN part, called the consequent (conclusion or action). The basic syntax or rule base is:

IF {antecedent}
    THEN {Consequent}

A rule can have multiple antecedents joined by the keywords AND, OR, or combination of both. The antecedent of a rule consists of two parts. They are object and its value. Object and value are linked by an operator. The operator can be mathematical or may be logical.

2.1 Structure of rule based expert system

A rule based expert system has five components: The knowledge base, the database, the inference engine, the explanation facility and the user interface. The knowledge base contains the knowledge about the domain. The database has set of facts, which is used to match the against the IF- THEN rules. The inference engine provides reasoning, so that expert system can reach a solution. The explanation facility provides the answer to user about why the particular solution is reached. The user interface enables user to interact with the other components of the expert systems.

The other additional components include the external interface, the developer interface, text editor, book keeping facilities, debugging aids, and run time knowledge acquisition. The external interface allows an expert system to interact with other database and programs. The developer interface allows developer to edit with knowledge base, rules and facts. Text editor provides notepad kind facility to enter inputs. Book keeping facility is provided to monitor the changes made by the knowledge engineer in knowledge base or inference engine. Debugging aids provides tracing of all rules fired during the program execution. Run time knowledge acquisition facility enables to add knowledge or facts, which are not available in knowledge base or database.

The structure of a rule based expert system is given in figure 1. (Negnevitsky, 2008)

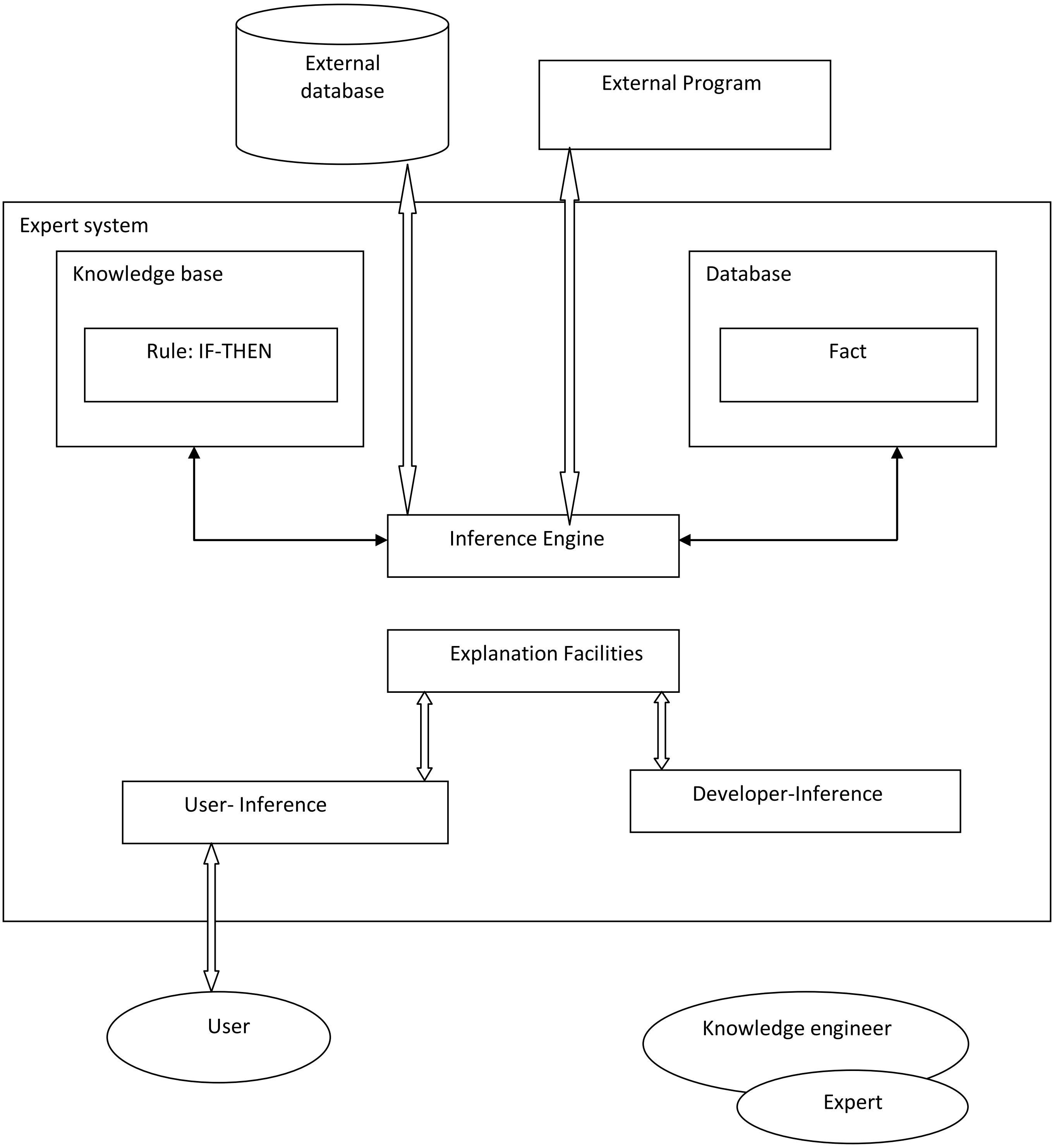


Figure 1: Structure of rule based expert system

2.2 *Inference Techniques*

Rule based expert system has set of IF-THEN rules to represent the domain knowledge and set of facts which will represent current situation. Inference engine compares each rule stored in knowledge base with the facts in the database. When the IF rule is match with a fact, the rule is fired and its action part stated in THEN is executed. The fired rule may change the set of facts by adding new facts in database. The entire above process produces inference chains. The process is shown in the figure 2.(Negnevitsky, 2008)

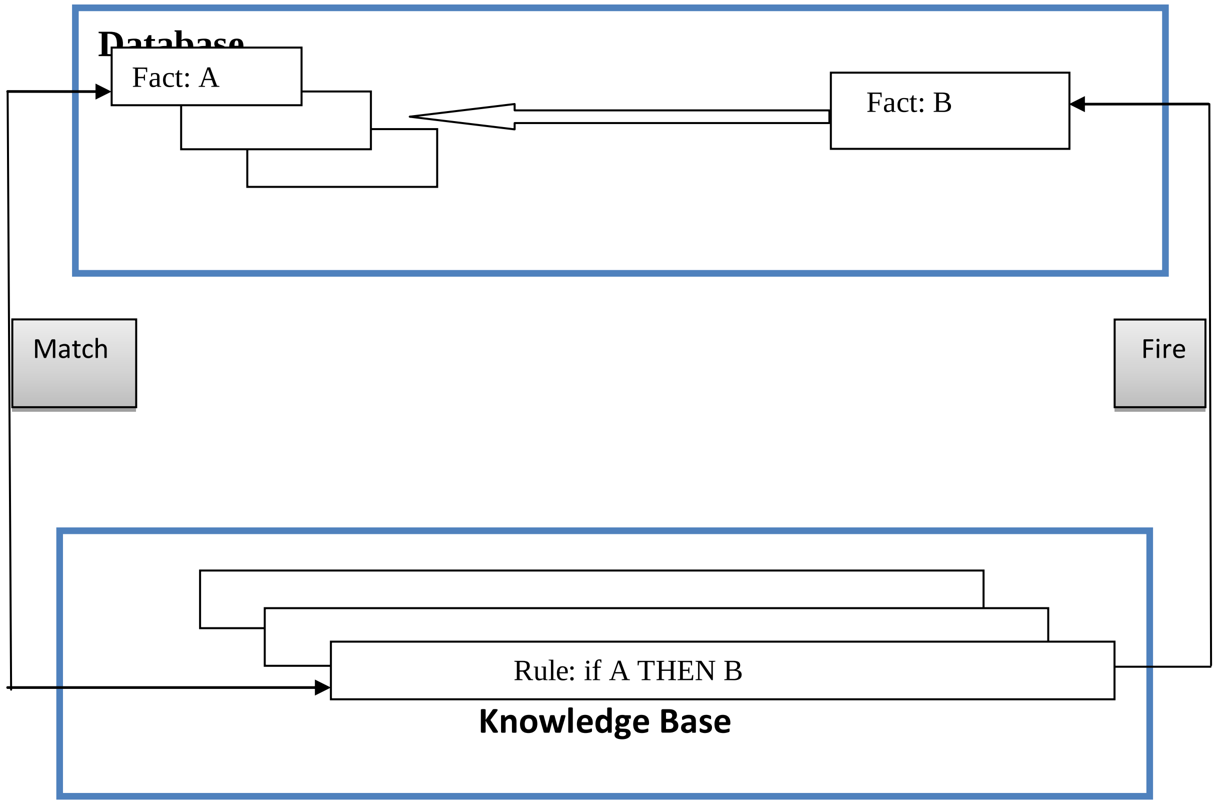


Figure 2: Inference process for rule based expert system

There are two types of inference techniques. Forward chaining and backward chaining.

Forward Chaining:
Forward chaining is the data driven reasoning. The reasoning starts from the known data and proceeds. It will look for the rules which will move the current state of problem solution closer to final solution. When rule is fired, the new facts are added to the database. (Chakraborty, 2010)

Let us consider the following example.
A rule base consisting of the following rule set.
Rule 1: If A and C then F
Rule 2: If A and E then G
Rule 3: If B then E
Rule 4: If G then D.

Problem: Prove that If A and B is true, then D is true.

Solution: Here we will start with rule 1 and proceed towards the next rule. First rule 3 will be fired and then rule 2 will be fired and then at the end rule 4 will be fired. Hence we can reach to the desired goal to prove that if A and B then D is true. This is how inference engine will process inference chain in forward manner.

The only problem with the forward chaining is that if the rule base has thousands rule, then many a times, unnecessary rules are also fired, which will increase execution time and we will generate many facts which are unrelated to goal. To overcome this problem, backward chaining is used.

Backward Chaining:
Backward chaining is the goal driven reasoning. In backward chaining, an expert system has the goal and the inference engine attempts to find the evidence to prove it. First, inference engine will search knowledge base to find rules that has required solution and such rule will have goal in their action (THEN) parts. If such a rule is found and if its condition (IF) part matches the data in the database, then the rule is fired and then goal is proven.

Let us see how backward chaining works from the above example.
A rule base consisting of the following rule set.
Rule 1: If A and C then F
Rule 2: If A and E then G
Rule 3: If B then E
Rule 4: If G then D.

Problem: Prove that If A and B is true, then D is true.

Solution: Here we will start with the goal D is true and move backwards till a rule is fired. So rule 4 will be fired and we have new goal to prove that G is true. When G is true, it means both A and E is true as per rule 2. While we are given the fact that A is true, hence we have got new goal which states that E is true. If E is true, moving backwards, we ascertain that B is true using rule 3. Hence now we ascertain both A and B is true. Hence the goal is proved.

Table 1: Difference between forward chaining and backward chaining:

Forward chaining	Backward chaining
The data is known at the beginning of the inference process, and the user is not required to input additional facts.	The goal is set up and the only needed data from the database is used for reasoning. User is only required to input facts which is not in the database.
Developers should choose the forward chaining when they need to gather some information first and then want to infer something from that.	Developers should choose backward chaining when they begin with hypothetical solution and then search for facts to prove it.
Dendral, an expert system for determining molecular structure of unknown soil uses forward chaining.	MYCIN, an expert system for diagnosis infectious blood disease uses backward chaining

In conclusion, we can say that a user can use a combination of forward and backward chaining both depends on the type of domain, its requirement and the shell used for development of an expert system.

2.3 *Advantages of rule based expert systems*

- It provides representation of knowledge in natural way of if- then rules.
- It has uniform structure of rule base. Each rule is an independent piece of knowledge.
- It separates the knowledge base from inference engine, i.e., knowledge base can be updates without intervening of processing.
- The rule base expert system can perform even with incomplete and uncertain knowledge by associating certainty factors with it.
- It is easy to understand and rules can be self-documented without the help of explicit translation.
- The rule base system can be developed in phase by manner. A quick prototype can be developed with few rules and if it achieve desired result then new rules can be added to improve the performance and efficiency.

2.4 *Disadvantages of rule base expert systems*

- It lacks hierarchical structure of knowledge representation; hence we cannot understand the logical interdependence of rules.
- It goes through exhaustive search, hence it makes system very slow and cannot be used for real time applications.
- It cannot learn from past experience and cannot break the rules in case if exception occurs.
- It is built on experts past experience, guts and intuitions and trial and error approach.

2.5 *Domain areas where rule based expert system are useful*

- When knowledge is diffuse, i.e. there are large numbers of facts, which are more or less independent of each other. For example, we cannot use rule base in mathematics domain where there are set of inter related principles. But rule base can be applied to clinical trial domain, where comparatively there is large number of interdependencies among rules.
- Where knowledge can be easily separated from its use, i.e. when there is no dependency on how to use knowledge. For example, we can use rule base expert system were we need to decide the ingredients of an item, but we cannot use rule base expert system where we also need to decide how much of each ingredients we need to mix up for a tasteful recipe.

3. FUZZY EXPERT SYSTEM

When we want to express expert knowledge that uses vague and unclear words like ‘slightly overburden’, ‘heavily reduced’, ‘moderately difficult’, ‘not so old’, ‘very tall’, we can use fuzzy set theory. Fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. (Negnevitsky, 2008) Fuzzy logic describes vagueness. But then problem comes, how will you differentiate between a members of a class from non-members. Fuzzy logic is based on the idea that all things are described on a sliding scale, which helps us to differentiate between members of the class from non-members.

Fuzzy logic is also known as multi valued logic. The classical logic operates on two truth values. They are True (1) and False (0). In fuzzy logic, all the truth values are expressed in real numbers from the interval between 0 to 1. A number in the interval is used to represent the possibility that a given statement is true or false. The logic is explained in figure 3.

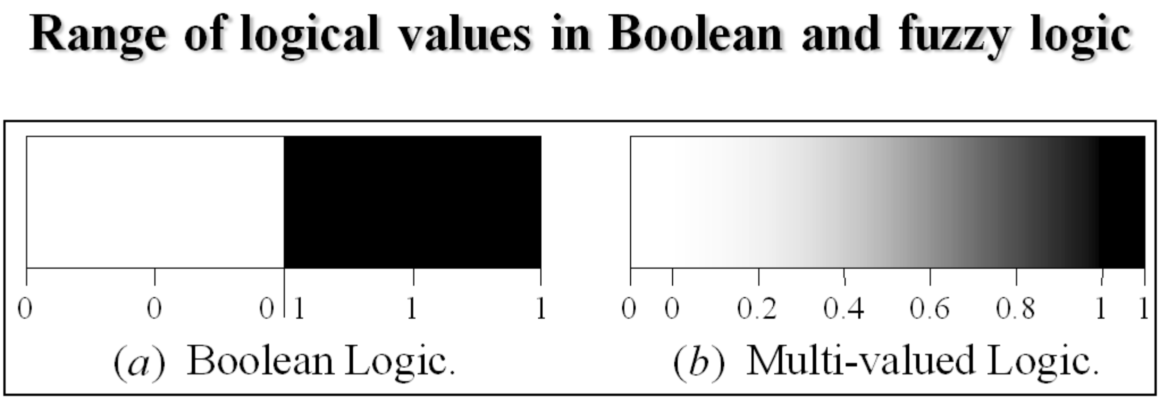


Figure 3: Fuzzy logic

Fuzzy logic is determined for knowledge representation based on degree of membership rather than on crisp membership of classical binary logic. (Zadeh, 1965)

3.1 *Fuzzy sets*

Fuzzy sets can be simply defined as a set with fuzzy boundaries. A fuzzy set is capable of providing a graceful transition across a boundary.
Let x can be universe of discourse and its elements be denoted as x. in classical set theory, crisp set A of X defined as function fA(x)called the characteristic function of A.

fA(x):X--> 0,1,

Where

$$fA(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

The set map inverse X is asset of two elements. For any element x of universe of X, characteristic function fA(x) is equal to 1, if x is an element of set A and is equal to 0 if x is not an element of A.

In fuzzy set A of universe X is defined by function µA(x) called the membership function of set a.

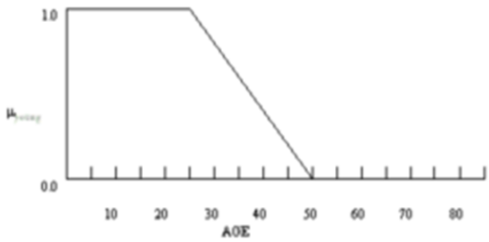
$\mu A(x): \quad X \longrightarrow [0,1]$,
Where

µA(x) = 1 if x is totally in A;
µA(x) = 0 if x is not in A;
0 < µA(x) < 1 if x is partly in A.

This allows a continuum of possible choices. For any element x of universe X, membership function µA(x) equals to the degree to which x is an element of set A. This degree, a value between 0 and 1 represents the degree of membership, also called membership value, of element x in set A. (Negnevitsky, 2008) For example, the fuzzy set young can be represented in the following figure 4.

Fuzzy set 'young'

Fuzzy Term *young*	
Age	Grade of Membership
25	1.0
30	0.8
35	0.6
40	0.4
45	0.2
50	0.0



Possibility distribution of *young*

$$\mu young(25) = 1,\ \mu young(30) = 0.8,\ \ldots,\ \mu young(50) = 0$$

Figure 4: Probability distribution of fuzzy set young

3.2 *Linguistic variables and hedges*

A linguistic variable is a fuzzy variable. When we say, X is tall, so here X takes linguistic value tall.
For example,
IF house is old
THEN resale value is low.

IF speed is fast
THEN time to travel is less.

The range of possible values of a linguistic variable represents the universe of discourse of that variable. For variable speed, the values can be very slow, slow, medium, fast and very fast. A linguistic variable carries with it the concept of fuzzy set qualifiers, called hedges. Hedges are terms that modify the shape of fuzzy sets. They include adverbs like very, somewhat, quite, more or less and slightly.

Hedge can modify verbs, adjective, adverbs or even whole sentences.They are used as

- All-purpose modifiers, such as very, quite or extremely.
- Truth-values, such as quite true or mostly false.
- Probabilities, such as likely or not very likely.

- Quantifiers, such as most, several or few.
- Possibilities such as almost impossible or quite possible.

3.3 *Fuzzy rules*

A fuzzy rule can be defined as a conditional statement in the form :
IF x is A
THEN I is B
Where x and y are linguistic variable and A and b are Linguistic Values determined by fuzzy sets on the universe of discourse X and Y, respectively. (Negnevitsky, 2008)

Difference between classical and fuzzy rules

A classical IF_THEN rule uses binary logic, for example,
Rule:1
IF speed is >100
THEN stopping distance is long
Rule:2
IF speed is < 40
THEN stopping distance is short

A IF-THEN rule using fuzzy logic,
Rule1:
IF speed is fast
THEN stopping distance is long
Rule 2:
IF speed is slow
THEN stopping _distance is short

The most important advantage with fuzzy expert system is that it merges the rules and consequently cut the number of rules significantly.

Fuzzy set includes two distinct parts: evaluating the rule antecedent (the IF part of the rule) and implication or applying the result to the consequent (Then part of the rule). In fuzzy systems, where the antecedent is a fuzzy statement, all rules fire to some extent, or in other words they fire partially. If the antecedent is true to some degree of membership, then the consequent is also true to that same degree.

3.4 Fuzzy inference

Fuzzy inference can be defined as a process of mapping from a given input to an output, using the theory of fuzzy sets. (Negnevitsky, 2008) The fuzzy inference process is performed in four steps. They are fuzzification of the input variable, rule evaluation, aggregation of the rule output and finally defuzzification. (Assilian, 1975)

Step 1: Fuzzification
In this step, we take the crisp inputs, x1 and y1 and determine the degree to which these inputs belong to each f the appropriate fuzzy sets.
Step 2: Rule evaluation
In this step, we take fuzzified inputs and apply them to the antecedents.
Step 3: Aggregation of the rule outputs
In this step, we take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set. Aggregation is the process of unification of the outputs of all rules . Thus the input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.
Step 4: Defuzification
Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The above process of converting output to crisp number is called defuzzification. The input for the defuzzification process Is the aggregate output fuzzy set and it is a single number.

3.5 Fuzzy expert system development process

The process has following steps.
1) Specify the problem and define linguistic variables.

2) Determine fuzzy sets.
3) Elicit and construct fuzzy rules.
4) Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.
5) Evaluate and tune the system.

4. FRAME BASED EXPERT SYSTEMS:

In rule base expert system and fuzzy expert system, IF- THEN rules are used to represent knowledge. In frame based expert system, frames are used to represent knowledge. Now we will try to explore the concept of frames.

4.1 Frame

A frame is a data structure with typical knowledge about a particular object or concept. Frames are used to capture and represent knowledge in a frame based expert system. (Minsky, 1975) Each frame has its own name and set of attributes or slots associated with it.

The frame provides a natural way for the structured and concise representation of knowledge. We can combine all necessary knowledge about a particular object or concept in a single entity. In general frames are an application of object-oriented programming for expert systems. The advantage what frames offer over the rule base is that we just need to search through frames only to execute rules unlike of rule base expert system, where it goes through systematic search through all rules for execution.

4.2 Frame as a knowledge representation technique

The concept of a frame is defined by a collection of slots. Each slot describes a particular attribute or operation of the frame.Slots are used to store values. A slot may contain a default value or a pointer to another frame, a set of rules or procedure by which the slot value is obtained. (Negnevitsky, 2008)
In general, slot may include the following information:

1) Frame name
2) Relationship of the frame to the other frames.
3) Slot value
4) Default slot value
5) Range of slot value
6) Procedural information: A procedure is executed if the slot value is changed or needed. There are two types of procedures attached to slots.
    a. When changed procedure is executed when new information is placed in the slot.
    b. When needed procedure is executed when information is needed for the problem solving but the slot value is unspecified. Such procedural attachments are called demons.

Frame based expert systems also provide an extension to the slot value structure through the application of facets.

4.3 Facet

A facet is a means of providing extended knowledge about an attribute of a frame. Facets are used to establish the attribute value, end-user queries, and tell the inference engine how to process the attribute.

There are three kinds of facets which can be attached with frame based expert system. They are value facets, prompt facets and inference facets. Value facet specifies default and initial values of an attribute. Prompt facets enable the end-user to enter the attribute value on-line during a session with the expert system. And finally inference facet allows us to stop the inference process when the values of a specified attribute changes.

4.4 Methods and demons

Frames provide us structured way of representing knowledge. But what should we do when we want to validate and manipulate the knowledge. The answer to that lies in methods and demons.

Method is a procedure associated with a frame attribute that is executed whenever requested. (Durkin, 1994) Method is represented by a series of commands similar to a macro in Microsoft excel.

In general, demons have an IF-THEN structure. It is executed whenever an attribute in the demon's IF statement changes its value.

4.5 Inference engine

In a frame based expert system, the inference engine searches for the goal or a specific attribute. In a frame based expert system, rule plays a secondary role. Knowledge is stored in frames and both methods and demons are used to add action to the frames. (Negnevitsky, 2008) The inference engine find those rules whose consequents contain the goal of interest and examine them one by one in order of rule base. If all the rules are valid, then inference engine will conclude that goal is reached else if any of the antecedents are invalid, then it is concluded that goal is not reached.

4.6 Frame based expert system development process

The steps in the frame based expert system development process are as follows: (Negnevitsky, 2008)

1) Specify the problem and define the scope of the system.
2) Determine the classes and their attributes.
3) Define instances.
4) Design displays
5) Define when changed and when needed methods, and demons.
6) Define rules.
7) Evaluate and expand the system.

5. HYBRID EXPERT SYSTEM

The above discuss technologies, i.e. rule based expert system, fuzzy expert system, and frame based expert system have positive and negative points. Let say, in rule based expert system, the time to search and execute rule is longer, while Fuzzy logic allows you deal with imprecise knowledge. Frame based expert system allows us to represent knowledge in hierarchical way, while in rule based expert system is most easy to construct because of it simple structure. When we want to combine he advantages of two components, we can create hybrid technology. But the hybrid technology can be good or bad depending on which two technologies are combined. Here we are exploring two hybrid expert systems.

i. Neural expert systems
ii. Neuro-fuzzzy systems

It is important to note that, in both of hybrid system; neural network is one of the important components.

6. NEURAL EXPERT SYSTEMS

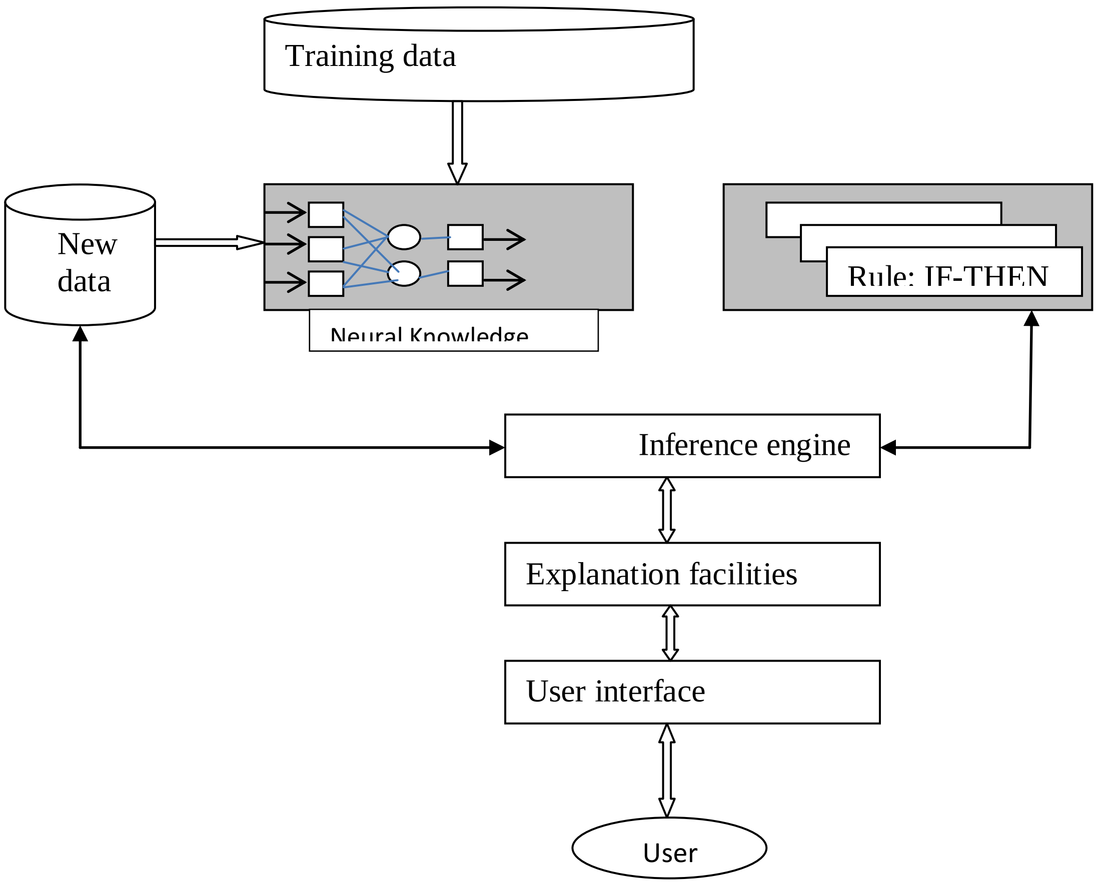


Figure5: Basic structure of neural expert system

Neural network and expert systems both have a common goal of imitating the human intelligence. A hybrid expert system, which combines a neural network and a rule based expert system, is called a neural expert system (Connectionist expert system). Here with the obvious advantages of rule based expert system, we also combine the advantages of neural network like learning, generalisation, robustness and parallel information processing.

6.1 Structure of neural expert system

The basic structure of neural expert system is given in figure 5. (Negnevitsky, 2008)

A Neural expert system has neural knowledge base in place of traditional knowledge base, where knowledge is stored as weights in neurons. The combination of rule extraction component along with neural network enables neural expert system to justify and provide explanation facility for its conclusion.

Neural network also allows dealing with noisy and incomplete data because of its capability of generalisation. Hence it allows approximate reasoning. The rule extraction unit examines the neural knowledge base and produces the rules implicitly buried in the trained neural network. The user interface provides the interaction between user and the neural expert system.

6.2 Rule extraction in neural expert system

Neurons in the network are connected by links, each of which has a numerical weight attached to it. The weights in a trained neural network determine the strength or importance of the associated neuron inputs. There are three layers in neural network. The first layer is input layer. Neurons in the input layer transmit external signals to the next layer. The middle layer is the conjunction layer. It is also known as hidden layer. Here activation function is used to calculate weights for neurons. There are different types of activation functions. The number of hidden layers depends on the types of input, type of domain and type of problem to be solved. The last layer is the output layer. Each output neuron receives an input from a single conjunction neuron. The weights between the second and the third layers are set to unity.

IF-THEN rules are mapped into multi-layer neural network where the last layer represents the action parts of the rules. We can train neural network according to a given set of training data using back propagation training algorithm. Once the initial training phase is completed, we can examine the neural network knowledge base, extract and refine the set of initial IF-THEN rules. Hence neural expert system provides a bi-directional link between neural networks and rule based systems.

The only problem with neural expert system is that it cannot represent continuous input variable as it may lead to creation and execution of infinite rules. The problem can be overcome by Neuro-fuzzy systems.

7. NEURO-FUZZY SYSTEMS

Fuzzy logic and neural networks are natural complementary tools in building intelligent systems. While neural networks are low-level computational structure that performs well when dealing with raw data, fuzzy logic deals with reasoning on higher level, using linguistic information acquired from domain experts. (Negnevitsky, 2008)

However, fuzzy systems lack the ability to learn and cannot adjust themselves to a new environment. On the other hand, although neural networks can learn, they are opaque to the user, i.e. the process is black box for the end users.

But when you merge a neural network with fuzzy expert system, then it provides you more powerful technology to design expert system. Integrated neuro-fuzzy systems can combine the parallel computation and learning abilities of neural network with the human-like knowledge representation and explanation abilities of fuzzy systems. As a result, neural networks become more transparent, while fuzzy system become capable of learning.

A Neuro-fuzzy system is, in fact, a neural network that is functionality equivalent to a fuzzy inference model. It can be trained to develop IF-THEN fuzzy rules and determine membership functions for input and output variables of the system.At the same time, the connectionist structure avoids fuzzy inference, which entails a substantial computational burden.

7.1 Structure of neuro-fuzzy expert system

The general structure of neuro-fuzzy expert system is given below in the figure 6. Here it has input and output layers and three hidden layers that represent membership functions and fuzzy rules.

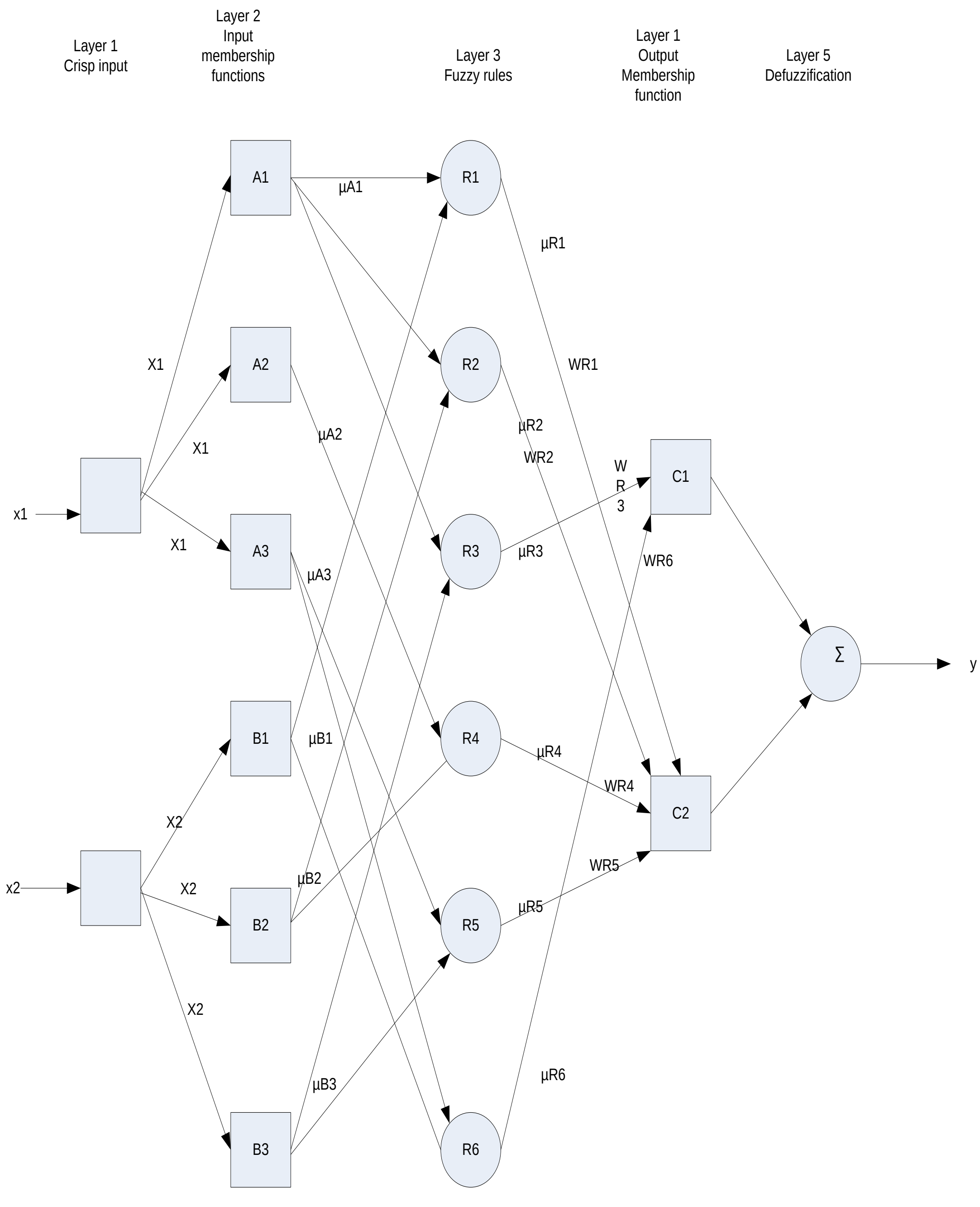


Figure 6: Architecture of Neuro-fuzzy system

7.2 Neuro-Fuzzy Inference process

Each layer in the neuro-fuzzy system is associated with a particular step in the fuzzy inference process. (Negnevitsky, 2008)

Layer 1 is the input layer. Each neuron in this layer transmits external crisp signals directly to the next layer. That is,

$$y_i^{(1)} = x_i^{(1)}$$

Where Xi is input and Yi is the output of input neuron i in Layer 1.

Layer 2 is the fuzzification layer. Neurons in this layer represent fuzzy sets used in the antecedents of fuzzy rules. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron's fuzzy set. The activation function of a membership neuron is set to the function that specifies the neuron's fuzzy set.

Layer 3 is the fuzzy rule layer. Each neuron in this layer corresponds to a single fuzzy rule. A fuzzy rule neuron receives inputs from the fuzzification neurons that represent Fuzzy sets in the rule antecedents. For instance, neuron R1, which corresponds to Rule 1, receives inputs from neurons A1 and B1.

The weights between layer 3 and layer 4 represents he normalised degrees of confidence of the corresponding fuzzy rules. These weights are adjusted during training of a neuro-fuzzy system.

Layer 4 is the output membership layer. Neurons in this layer represent fuzzy sets used in the consequent of fuzzy rules. An output membership neuron combines all its inputs by using the fuzzy operation union.

Layer 5 is the defuzzification layer. Each neuron in this layer represents a single output of the neuro-fuzzy system. It takes the output fuzzy sets clipped by the respective integrated firing strength and combines them into a single fuzzy. The output of the neuro-fuzzy system is crisp and combined output fuzzy set must be defuzzified. For that standard defuzzification methods can be applied.

7.3 Learning of Neuro-fuzzy system

Neuro-fuzzy system can learn using standard learning algorithms developed for neural networks for example, back propagation.(Altock, 1997)

When a training input-output of example is presented to the system, the back –propagation algorithm computes the system output and compares it with the desired output and compares it with the desired output of the training example. The difference is propagated backwards through the network from the output layer to the input layer.
The neuron activation functions are modified as the error is propagated. To determine the necessary modifications, the back-propagation algorithm differentiates the activation function of the neurons.
This is how neuro-fuzzy system learns.

8. COMPARISON OF DIFFERENT TYPES OF EXPERT SYSTEMS

Here we have discussed in total five expert systems. They are rule based expert system, fuzzy expert system, frame based expert system, neural expert system, and neuro-fuzzy expert system. The parameters which we are using for comparison are: Knowledge representation, learning ability, uncertainty tolerance, imprecision tolerance, explanation facility, inference engine, knowledge update facility, maintenance, adaptability, knowledge structure and processing time of the system. The comparison is given in the table 2

9. CONCLUSION

In conclusion, we can say that each type of expert system has its own merits, demerits and requirement to use based on domain analysis.

Table 2: Comparison of different types of expert system.

Parameters	**Rule based expert system**	**Fuzzy expert system**	**Frame based expert system**	**Neural expert system**	**Neuro-fuzzy expert system**
Knowledge Representation	IF-THEN rules in the knowledge base	Based on degree of membership using fuzzy logic	Knowledge in frames using hierarchical structure	IF-THEN rules in the neural knowledge base	In linguistic variable and using IF-THEN rules in fuzzy structure
Learning ability	Cannot learn on its own, and update existing knowledge base	Lacks ability to learn from the experience	Cannot learn and adjust to the new environment	Neural network can learn but the learning is black box process for the user	Has learning ability because of neural network as one of the component
Uncertainty tolerance	Difficult to measure uncertainty	Probabilistic reasoning can deal with uncertainty	Not possible because of knowledge structure	Approximate reasoning	Using script value and probabilistic reasoning
Imprecision tolerance	Very low, required precise information	High, as fuzzy logic can deal with imprecision	Very low, imprecise data can lead to faulty output	Neural network component can deal with imprecise data	Very high because of combination of neural network and fuzzy logic
Explanation facility	Yes	Yes, using linguistic variable	Yes, good explanation for the output	Yes, because of rule based component	Yes, very effective
Inference Engine	Process rules and derive conclusion	Process rules using fuzzification and defuzzification	Search for goals using methods and demons	Three layer neural network combined with rule extraction	Fuzzy inference process using fuzzification and defuzzification
Knowledge update	Difficult to add new rules	Difficult to introduce new linguistic variables in existing structure	Not possible	Yes, With new knowledge, neural network component can learn	Yes, new linguistic variables can be added into existing knowledge structure
Maintainability	Moderately difficult	Very difficult	Easy	Easy	Difficult because of fuzzy component
Adaptability	No	No	No	Yes	Yes
Processing time	Very high due to each rule is processed	Processing time is reduced compare to rule base expert system	The knowledge stored in the frames can be processed very rapidly.	Learning for neural network takes time, but once knowledge is stored in neurons, then processing time for rule is fast.	Learning for neural network takes time, but then processing time is significantly reduced.
Knowledge Structure	Adhoc, cannot understand logical dependence of rules	Quite unstructured	Highly structured	Structured, but stored in neurons	Moderately structured

10. REFERENCES

[1] Altock, V. (1997). *Fuzzy logic and neurofuzzy applications in business and finance.* New Jursey: Prentice hall.
[2] Assilian, E. M. (1975). An experiment in linguistic synthesis with a fuzzy logic controlle. *International journal of man-machine studies*, 1-13.
[3] Chakraborty, R. C. (2010, jun 1). *artificial_intelligence.html*. Retrieved 1 17, 2014, from myreaders.info: myreaders.info/artificial_intelligence.html
[4] Durkin, J. (1994). *Expert system desing and development.* New jersy: Prentice Hall, Englewood Cliffs.
[5] Minsky, M. (1975). A framework for representing knowledge. *The psycology of computer vision*, 211-277.
[6] Negnevitsky, M. (2008). *Artifical Intelligence.* Pearson Education.
[7] Zadeh, L. (1965). Fuzzy sets, information and control. *Information science*, 338-353.