International Journal of Computer and Communication Technology

Volume 1 | Issue 2 Article 8

April 2010

Computational Intelligence with Automata Theory

Neelam Kumari
Lingaya's Inst. Of Mgmt. & Tech, Haryana, Neelam.k18@gmail.com

Computational Intelligence with Automata Theory

Neelam Kumari

Student, Lingaya's Inst. Of Mgmt. & Tech, Haryana

Neelam.k18@gmail.com

***Abstract*—In this paper I have discussed about a various techniques i.e. Neural Network, Fuzzy Logic, and Genetic algorithm and a few concepts of automata theory. Every application that deals with fuzzy logic also deals with Neural Networks and Genetic algorithm. But how the application will work depends totally on Computational Intelligence. So here I discussed Computational Intelligence with Automata Theory concepts for designing any application.**

I. INTRODUCTION

Before studying an application we must have to know about the term "Computational Intelligence (CI)". It is an advanced concept of Artificial Intelligence (AI). Ai is an area of computer science concerned with designing intelligent computer system. A system that exhibits the characteristics we associate with intelligence in human behavior. So n next, we will see CI with fuzzy.

It's a smart work.CI uses three components for resolving an application. Following are three components:

1. Neural Network
2. Fuzzy Logic
3. Genetic algorithm

No technique is complete in itself. So we use the one which has the strong features of all the mentioned other techniques.

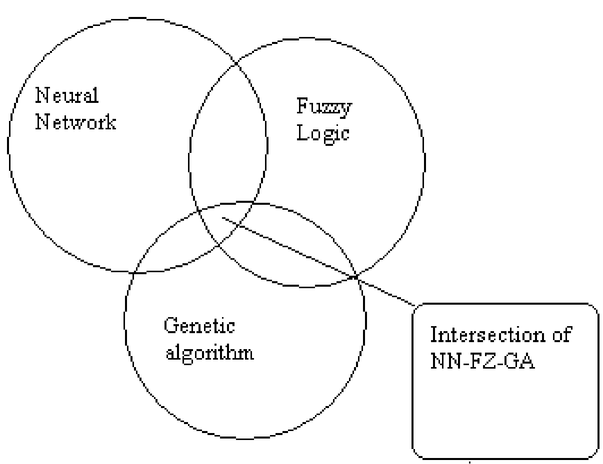


Fig.1 Integrated System

Existing applications based on these techniques are:

- Expert System
- Robotics
- Production System
- Medical Diagnostics System

All three terms are known as soft computing collectively. So we need to integrate these components as shown in fig.1. Such integrated system is known as Hybrid System. These are classified as:

1. Sequential Hybrid
2. Auxiliary Hybrid
3. Embedded Hybrid

Sequential Hybrid : It is used in a pipeline fashion. So, one technology's output becomes another's input and so on/ This is weakest form of hybridization, because an integrated combination of the technology is not present. Example : Genetic algorithm obtains the optical parameter for different instances of a problem and hands over he preprocessed data set to NN for further processing.

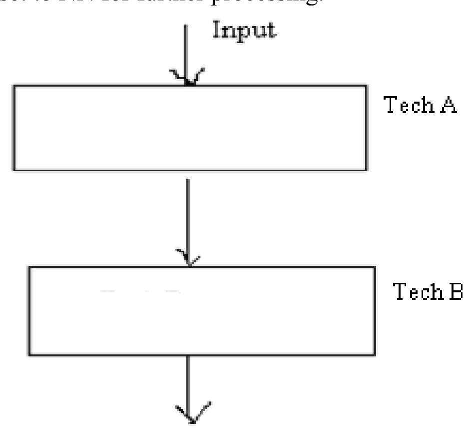


Fig. 2 Sequential Hybrid

Auxiliary Hybrid : Here one technology calls other technology as subroutines. Example : Neuro-genetic system in which a NN employees a GA to optimise its structural parameters.

Embedded Hybrid : NN – FL makes an integrated system. These systems have a NN which receives fuzzy inputs, process it and extracts fuzzy outputs.

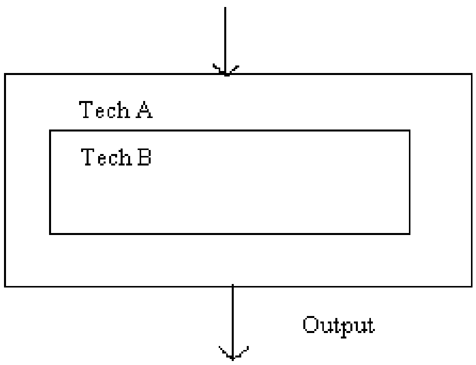


Fig. 3 Auxiliary Hybrid

II. APPLICATION

A Fuzzy Car Using NN concept

It includes rule base, fuzzification, de-fuzzification, inference and feedback which are the realized by recurrent five layers of Neural Network.
Statement: By using this concept the fuzzy car would be able to move along a track that has rectangular turns.

1st layer : The first layer of the network is composed of linguistic nodes. Each node represents a variable. Here we use here variables x_0 , x_1 and x_2. The variables x_0 and x_1 represent distance of centre of car from horizontal and vertical boundaries at the corner. x_2 represent steering angle. One more variable y will be taken as next steering angle.

2nd layer : The second layer of network determine input membership function for the input variables. So it is known as the justification process.

3rd layer : It is composed of rule codes. Each node represents one fuzzy rule.

4th layer : It is composed of tern nodes. They act as membership function for the output, with the links from the 3rd layer to perform as decision making logic.

5th layer : It is composed of two set of linguistics nodes. One set is for training data i.e. desired output. The output is fed back to the fourth layer nodes. The second set of nodes gives the actual control.

At the first layer "Self Organization Clustering" concept of Neural Network is used to locate the membership functions by finding the centres and widths.

"Supervised back propagation" concept of Neural Network is used to adjust the width and centre of the membership optimally. In such type of application, we use database of AI.

,

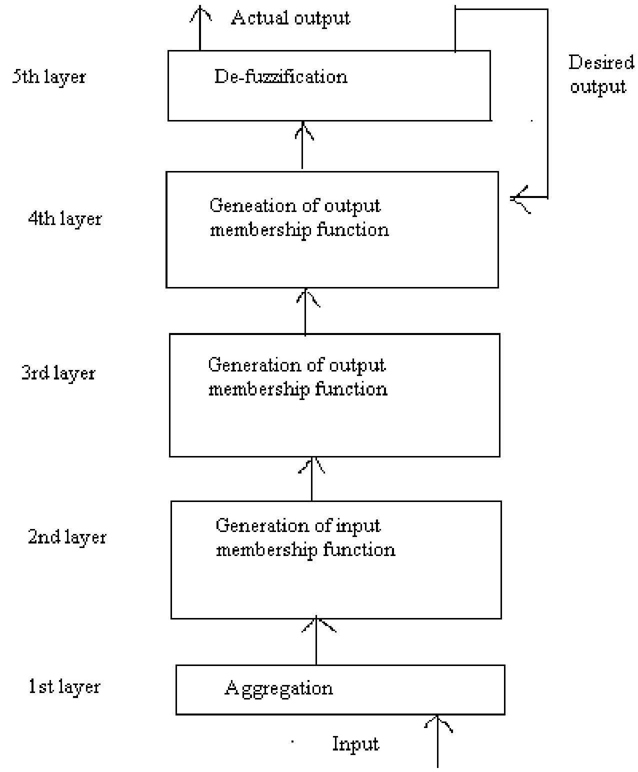


Fig. 4 Layers

III. AUTOMATA THEORY

An automata is generally called a machine. It does all the work automatically. Like automatic packing machines etc. We just need to give the input, we get the output.
Example : AND gate

Input	Output
(0, 0)	0
(0, 1)	0
(1, 0)	0
(1, 1)	1

Such types of basic machines do not deal with states. Computer is also a machine to which we give the inputs, get the desired outputs. Here w have to deal with the states as well.

State : The configuration of machine at particular instant of time is called the state.
There can be many possible states like holding state, wait state, processing state, etc. Here the record of the state is kept. So we always have to us storage elements i.e. auxiliary memory. Such type of automation has the following characteristics :

- Input
- Output
- States
- State Relation
- Output Relation

Input : At each discrete instant of time t_1, t_2,... input values I_1, I_2,.... each of which can take a finite number of fixed values from the input alphabet which are applied to the output side of mode.

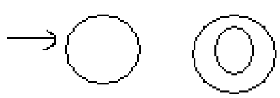
Output : O_1, O_2,... which can take finite numbers of fixed values from an output alphabet.

States: At any instant of time the can be in one of the states Q_1, Q_2,.... Qn.

State Relation : The next state of an automation at any instant of time is determined by the present state and the present input.

Output Relation : Output is related to either state only or to both the input and the state.

An automata is represented by 5 tuple (Q, ∑, δ , q_0, F) where
Q is a finite non-empty set of states
∑ is a finite non-empty set of states called input alphabet. ∑ is transitive function
δ : (Q × ∑) → Q
F is final state, where F ЄQ is the set of final states.



Initial state is represented by a circle with arrow head. Final state is represented by two concentric circles.

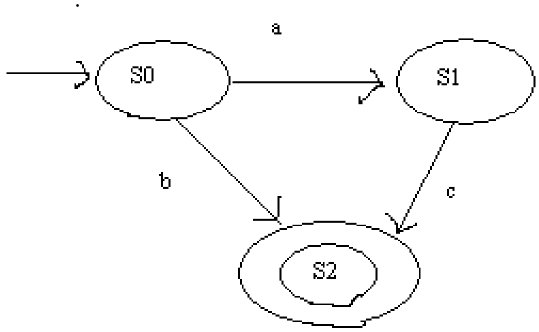


Fig. 5 Transition Diagram

Q = { s_0 , s_1 , s_3}
∑ = { a, b, c}
δ (s_0, a) = s_1
δ (s_0, b) = s_2
δ (s_0, c) = s_2
F = {s_2} i.e. final state
Q_0 = {s_0}

IV. NON DETERMINISTIC FINITE AUTOMATA

Non deterministic means a choice of moves for automation rather than a unique move. In each situation we allow a set of possible moves. In the NFA there can be a transition to more than one state on the same input symbol.

Transition function for figure 6 are as :

δ (q_0, a) → q_1, q_3
δ (q_1, a) → q_1
δ (q_3, a) → q_f
δ (q_1, b) → q_1, q_3
δ (q_0, a) → not reachable to final state
δ (q_f, b) → q_f reachable to final state

Example : Suppose input string is : aab

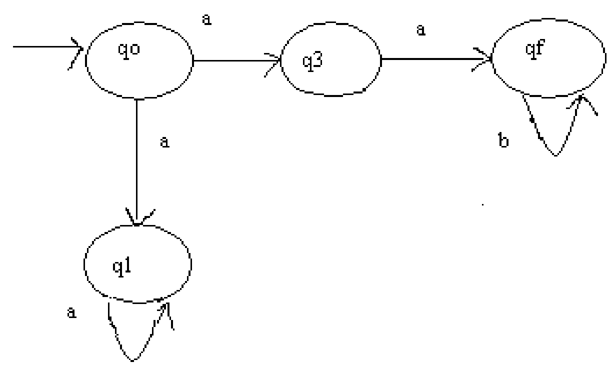


Fig. 6 Transition Graph for NDFA

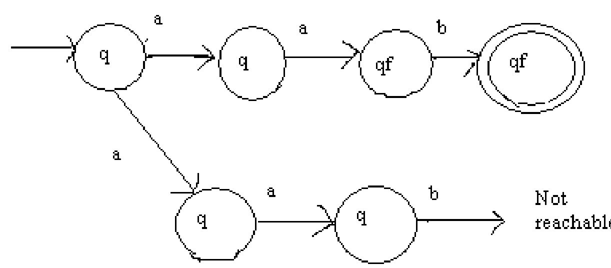


Fig. 7 Another alternative graph for the same NDFA

From figure 7, it can be seen that we can reach on final state q_r by the following path only, still there exist some more alternatives.

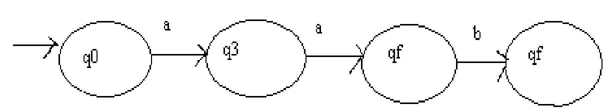


Fig 8 Final path for accepting "aab"

Relation between NDFA and fuzzy inputs

In NDFA, we can reach to the final state by verifying all the inputs in polynomial time. But it is not sure that we will reach to the final state. Similarly in fuzzy, all the inputs we collect from the past history of data. We try to fuzzify all the inputs by membership grade. We can have different diagrams for all the possible inputs in fuzzy systems, the diagram which will be very near to our expected output, we select that one. So, we can define the NDFA diagram for all he fuzzy inputs to get the membership grade of all inputs applied.

V. CONCLUSIONS AND PROPOSED SOLUTIONS

In this paper I have defined NDFA as a design technique. We can easily design NDFA transition where input is uncertain. In the above fuzzy car example we can show that what should be the next inputs combination with transition diagrams.

So we can say that that NDFA representation of fuzzy input with uncertainty and can be applied to all the present and future applications based on fuzzy.

VI. REFERENCES

[1] Japanese Technology Evaluation Center, Fuzzy Logic Research and Life, 1995 http://itri.loyola.edu/kb/c5_s4.htm

[2] Enrique Ruspini, Two Responses to Elkan's Paper, 1993 http://www-cgi.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/doc/elkan/response.txt

[3] Togai InfraLogic Inc., Fuzzy logic Primer – A Brief Introduction to Fuzzy Logic,1992 http://www-cgi.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/fuzzy/doc /intro/primer.tgz

[4] William Siler, Building Fuzzy Expert System http://members.aol.com/wsiler/index.html

[5] Earl Cox, The Seven Noble Truths of Fuzzy Logic http://freeware.aus.sps.mot.com/freeweb/pub/fuzzy/news5 .txt

[6] Earl Cox, Logic of the Future http://darwin1.ucsd.edu:8000/connect/articles/success/logic.html

[7] Perles, M. Robin, M. O. Sharmir, "The Theory of Definite Automata"

[8] Hubert Comon, Max, Dauchet, Remi Glleren, "Tree Automata Technques and Applications"

[9] Joost Engelfriet and Hendrik Jan Hoogeboom, "Tree Walking Pebble Automata"

[10] John Von Neumann, "The General and Logical Theory of Automata"