SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

Proceedings of the 10th Annual Sessions

29th November 2013

Sri Lanka Association for Artificial Intelligence – 2013

Proceedings of the 10th Annual Sessions
29th December 2013
Nawala
Sri Lanka

The materials in this publication have been supplied by the respective authors, and the views expressed remain the responsibility of the named authors. The statements and opinions stated in these publications do not necessarily represent the views of the Sri Lanka Association for Artificial Intelligence.

Editors:

Dr. (Ms) DDM Ranasinghe
Dr. (Ms) HUW Ratnayake

Sri Lanka Association for Artificial Intelligence, Department of Electrical & Computer Engineering, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka

CONTENTS

A Fuzzy-Mathematical Model to Recover Motion with Monocular Vision

S. Pathirana[1], J. G Samarawickrama[2], A. S. Karunananda[3]
[1]Department of Computational Mathematics, University of Moratuwa, Sri Lanka
[2]Department of Electronic & Telecommunication Engineering, University of Moratuwa, Sri Lanka
[3]Department of Computational Mathematics, University of Moratuwa, Sri Lanka

Abstract– **This paper describes a fuzzy based mathematical model to recover the motion path of an obstacle observed via a single camera. The strategy is analysing a sequence of images captured in regular time intervals, more specifically, studying the variation of the apparent size of an obstacle and the relative position change on reference frame. Those two measures are the only inputs to the fuzzy-mathematical model, the major emphasis of the research, which recovers the motion path as an equation. Necessary feature extraction is being achieved via a real time image processing module which relies on an optical flow technique as the key technique to recognize dynamic objects. It was reported a 91.98% average accuracy from the fuzzy-mathematical model in simulation environments, where the inputs were generated by a simulator program in order to study the precision of the fuzzy-mathematical model as a standalone application. An average accuracy of 59.8% was experienced at the real time application, an artefact to test the postulated concept in real time dynamic environment, which comprised of three major modules: a real-time image processing module, the fuzzy mathematical model and a mobile robot.**

1. INTRODUCTION

Obstacle avoiding is fundamental to the domain of mobile robots. But, coping with dynamic obstacles is absolutely a challenging task since it is extremely difficult to predict the motion behaviour of dynamic objects. This paper describes an attempt to accept the challenge with a fuzzy-mathematical model.

We postulate that a single camera, known as monocular vision in computer vision, is adequate to obtain the perception from the world rather than the widely used technique binocular or stereo-vision. Although stereo vision is appropriate to have the depth information of objects, it is identified as an expensive technique in terms of processing. Nevertheless, it was realized that even with monocular vision, the objective of obtaining the estimated motion path of objects can be achieved, while avoiding the excess burden of aligning and calibrating the two cameras, if stereo vision preferred instead. This advantage was gained due to the innovative concept of using a fuzzy-mathematical model to simulate the motion behaviour, since it requires the only inputs - the apparent size variation of the obstacle in interest and the relative position change of it on the reference frame, both can be measured with only one visual streaming channel.

The fuzzy-mathematical model postulated in this paper produces the equation of the motion path of an object which moves on a straight line, in form of a equation $y = mx + c$, according to a pre-defined 2-D coordinate system. Hence, the primary objective of the research was deriving the relationships among the key parameters of the motion path equation gradient (m), intersection (c) and the variation of apparent size of the obstacle. The milestone was successfully completed when the relationships between those parameters were found as fuzzy membership functions. Then the model designed in postulation stage was refined and fine-tuned with real time experimental data to infer the required outputs: gradient-m and intersection-c of the predicted motion path, once the data set of two variations mentioned above are provided as inputs.

The idea of the FMM (Fuzzy Mathematical Model) was inspired by nature, more specifically, how human judge hazardous dynamic objects seen and react in order to avoid them.

2. RECENT TRENDS IN OBSTACLE AVOIDANCE WITH MONOCULAR VISION

Ashutosh Saxena from Stanford University is a well-known researcher in this domain. A study done by himself and his team [1] was to estimate depth from a single monocular (still) image. They have named it as ' Recovering 3-D depth'. In their research, they also have studied that how human estimate depth from a monocular image. They have initiated with collecting a training set of monocular images (of unstructured outdoor environments which include forests, trees, buildings, etc.). The researchers have used a 3-D distance scanner to collect training data, which comprised a large set of images and their corresponding ground-truth depth-maps. Their model has used a discriminatively-trained Markov Random Field (MRF). Using this training set already collected, they have trained MRF discriminatively to predict depth.

According to them, their algorithm was frequently able to recover fairly accurate depth-maps.

Although their algorithm appeared to predict the relative depths of objects quite well (i.e. their relative distances to the camera), it has made more errors in absolute depths. They mention that their algorithm appeared to incur the largest errors on images which contain very irregular shapes like trees. Anyway, some of the errors were attributed to errors or limitations of the training set by the researchers.

Our reviews regarding their research are as follows: They have followed a supervised learning approach in order to deal with this problem. The key technique they have used to extract depth information of the image, Markov Random Field (MRF), is a probabilistic distribution analysis approach, more specifically, an undirected graphical model having a set of random variables having a 'Markov property' described by an undirected graph. Since the graph is undirected and subjected to be cyclic, visiting nodes of the graph might be costly due to higher order of connectivity combinations and dependencies. Obviously the model is mathematically complicated and leads to a computationally expensive implementation.

The other fact worth mentioning is that their model has focused on detecting static objects rather than considering dynamic objects. In addition to that, it need to be trained before use, therefore cannot expect functioning properly in previously unseen environments.

Francisco Bonin-Font, Alberto Ortiz and Gabriel Oliver, University of the Balearic Islands, Spain [3] who have done a comprehensive survey on visual navigation for mobile robots point out that the vision-based navigation solutions have mostly been devised for Autonomous Ground Vehicles (AGV), but recently, visual navigation is gaining more and more popularity among researchers developing *Unmanned Aerial Vehicles* (UAV). They also highlight the fact, even a number of solutions for *Autonomous Underwater Vehicles* (AUV) can already be found for many undersea critical applications rely on vision.

They have thoroughly investigated about 40 different research with good-quality results; the interesting observation is 29 out of 40 with satisfactory achievements have been based on single camera, means monocular vision. (That is approximately 72.5%). They have represented the whole three (3) above mentioned domains of applications, ground, UAV and AUV.

They classify the main techniques of vision based object detection as optical-flow-and appearance-based, where Optical-flow-based solutions estimate the motion of objects or features within a sequence of images. They mention that the researchers compute optical flow mostly using (or improving) pioneering techniques from Horn and Lucas and Kanade. In their report, (page 14), they clearly state that vision based systems on optical flow have proved to be especially useful for Unmanned Aerial Vehicles (UAV) because optical flow gives the scene qualitative characteristics that cannot be extracted in detail even from single low quality images.

Yoko Watanabe, states in his Ph.D. thesis [4] that monocular vision based systems have been operating in even more complicated scenarios in UAVs (Unmanned Aviation Vehicles).

He has experimented with UAVs to detect and avoid obstacles while executing a given mission such as preplanned path following or way point tracking and reported satisfactory success in both simulation environments and real-world applications. They have attempted to cope with stochastic behavior of object motion with predictions (page 59). Drawback of the research is the approach of both image processing with Extended Kalman filter (EKF) and estimations based on higher order derivatives in mathematical equations were consuming many computational resources.

The group of researchers lead by Animesh Garg Institute of Technology, India [5] have also done a study on how monocular vision based systems behave in a real world scenario such as vision based obstacle detection and mapping techniques for identifying objects in urban environments for Autonomous ground vehicle navigation. According to them, the applicability of monocular vision has been successfully tested on a test vehicle under variable outdoor lighting conditions and stable results have been obtained. Anyway, their approach was entirely image processing based inference, where recognition of distinct objects done using a primitive method, RGB color separation. But, using ultrasonic sensors for emergency stops in case of proximity to other objects conveys the message that they are not confident of their depth estimation. We interpret it as not a weak point of monocular vision, but inadequate effort of depth estimation and inference. It seems they have experimented this in controlled environments, rather than busy roads.

Study done by Tobias Low from University of Southern Queensland, Australia and Antoine Manzanera from Ecole Nationale Sup´erieure de Techniques Avanc´ees, Paris, France [6] also implies the fact, ability of monocular vision for object recognition. Although they focus on classification of static objects, an optical flow technique has been employed.

Nicolau Leal Werneck and Anna Helena Reali Costa from Universidade de São Paulo, Brazil [7], Chau Nguyen Viet and Ian Marshall representing University of Kent, Canterbury, United Kingdom [8], from the same research domain, computer vision have also studied use of monocular vision for object recognition, but none of them were interested in coping with moving obstacles rather than classifying and avoiding static objects.

The researchers in a group lead by Jan Hoffmann from Institut f¨ur Informatik, Berlin, Germany have presented a working model [10][11] of use of monocular vision in the RoboCup 2003 obstacle avoidance challenge in the Sony Four Legged League. Their system had enabled the robot to detect even unknown obstacles and reliably avoid them while advancing toward a target. It has used monocular vision data with a limited field of view. Obstacles were detected on a level surface of known color(s). Although system has proved highly successful by winning the obstacle avoidance challenge and was also used in the RoboCup championship games, we cannot accept it as a pure vision based perception due to the reason of using additional sensory devices for 'odometry'. (Odometry is the use of data from moving sensors to estimate change in position over time. Odometry is used by some robots, whether they be legged or wheeled, to estimate (not determine) their position relative to a starting location.)

Above studies generalize the conclusion that, obstacle avoidance task can be achieved via monocular vision, more specifically with image sequence from a video stream. The predictability of motion path of an obstacle with such technique is also confirmed. In addition to that, they promote the technique optical flow to recognize dynamic objects. But none of them have preferred an artificial intelligence approach, hence affected from the drawback, suffering from high computational overhead.

We postulate a fuzzy based modeling is appropriate to model rather than algorithmic or mathematical where the domain complexity and increased computational overhead is unavoidable. The idea is inspired by nature, how human and animals safeguard themselves from moving objects.

Since human are not capable of performing fast mathematical equation solving (i.e. find factors) or even finding the square root of a simple whole number, obviously they do nothing with complicated mathematics such as Markov Random Fields. They never use any odometry sensors, simply use vision as the primary sensory device to perceive the world and do simple inference with less effort.

Attempts for decision making based on analysis of a live video stream with fuzzy logic are rare, but not absent at all.

One of the best example is a study done under the leadership of Rafaelmun˜ Oz-Salinas at Department of Computer Science and Artificial Intelligence, University of Granada. Spain [12] was to recognize doors, distinguish from misguiding objects by shape such as almirahs or large windows for a mobile robot which rely on vision. They mention that the fuzzy logic approach lead to achieve their objectives as expected.

3. SYSTEM OUTLINE

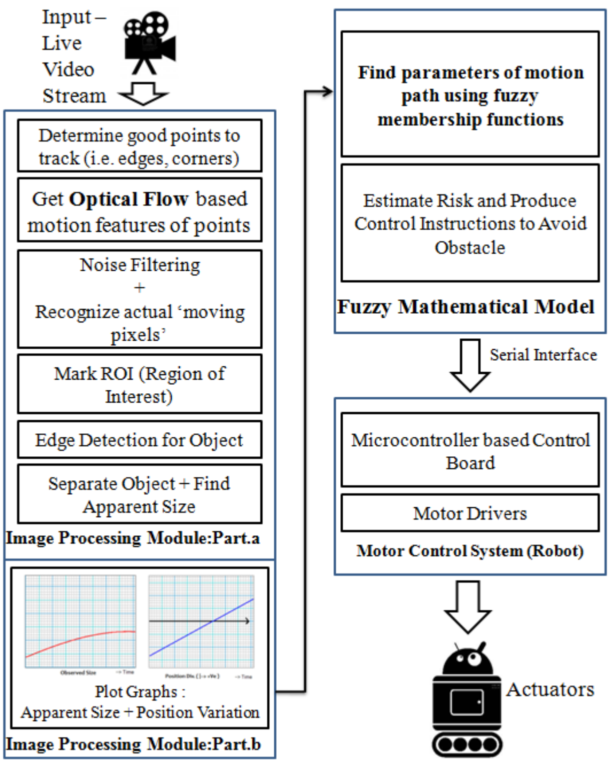

Input – Live Video Stream
Determine good points to track (i.e. edges, corners)
Get Optical Flow based motion features of points
Noise Filtering + Recognize actual 'moving pixels'
Mark ROI (Region of Interest)
Edge Detection for Object
Separate Object + Find Apparent Size
Image Processing Module:Part.a
Observed Size
→ Time
→ Time
Plot Graphs : Apparent Size + Position Variation
Image Processing Module:Part.b
Find parameters of motion path using fuzzy membership functions
Estimate Risk and Produce Control Instructions to Avoid Obstacle
Fuzzy Mathematical Model
Serial Interface
Microcontroller based Control Board
Motor Drivers
Motor Control System (Robot)
Actuators

Figure 1: System design overview

The entire artefact is comprised of 3 major modules, namely Image Processing Module, Fuzzy Mathematical Model and The Robot. Task assigned to Image Processing Module is producing the data sets, the apparent size and position variation, those are the inputs for the Fuzzy Mathematical Model. It again consists of many sub-components of modules, basically for recognizing the target object, after noise filtering. We have improved the existing noise filtering algorithms with density based clustering, which is a statistical approach. Anyway, the key algorithm of this module is an Optical Flow, more specifically Lucas-Kanade (LK) Optical Flow, a Sparse Optical Flow algorithm. The particular optical flow algorithm was selected among many Sparse and Dense Optical Flow algorithms, purely on experimental basis. Dense Optical Flow considers almost all the points to predict the optical flow, which is precise, but was very slow in real time processing. Horn-Schnuck and Gunner Farneback's Optical Flow algorithms are well known. For example, the better one, Gunner Farneback's Optical Flow could process at a rate of 0.25 fps, only 1 frame once in 4 seconds. Therefore, the module was designed with Sparse Optical Flow, which concerns only interesting points of the image, i.e. edges and corners of objects for optical flow predictions. Lucas-Kanade algorithm was selected due to the fast processing capability, which is a must, although it took some time to initialize (track interesting points). We conclude that LK is the most appropriate for real-time processing among existing optical flow algorithms.)

The Fuzzy Mathematical Model, the major emphasis of the research was innovated with the following interesting findings.

One of the most important discoveries was finding the relationship between the gradient (m) of the object's motion path and the apparent size variation. It was observed that the gradient value has some relationship with the skewness (equation (1)) of the curve of the apparent size variation graph, and most probably independent from other factors. This was clearly noticed at fixed points in Table 01.

$$Skewness = \frac{\sum_{i=1}^{N}(x_i-\bar{x})^3}{(N-1)s^3} \quad \text{Where } s=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2} \qquad (1)$$

It was noticed that, the skewness value and gradient values are having a relationship in fuzzy nature. But the challenge was to find the actual relationship and below is the strategy followed to determine that.

We constructed a hypothetical fuzzy membership function as Figure.2 (The skewness value was divided in to several ranges and for skewness value between the range *l1* and *l2*) and derived the relationship (3) between the *y* value and some fixed values of gradient *m* as in Table 02.

Table 1: Relationship between the gradient and skewness: Comprehensive

Gradient (m)	Skewness Approx.
-0.1	1.00
-0.2	0.83
-0.3	0.55
-0.5	0.15
-0.6	0
-0.7	-0.14
-0.8	-0.26
-0.9	-0.40
-1.0	-0.52
-1.5	-1.30
-2.0	-3.00

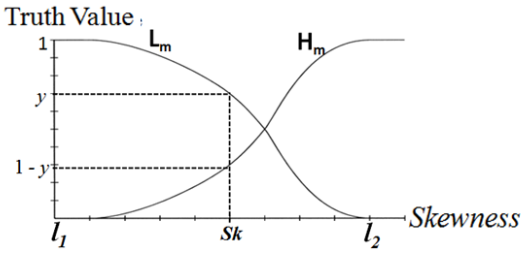


Figure 2: Hypothetical fuzzy membership function for m and skewness

$$m = [y*L_m] + [(1-y)*H_m].(2)$$
$$=>y = (m - H_m) / (L_m - H_m).(3)$$

Table 2: Relationship between the gradient and skewness: Comprehensive

Gradient (*m*)		Skewness (*Sk*)	
H_m(High)	L_m(Low)	*l*1 (lower bound)	*l*2(upper bound)
-0.1	-0.6	1.00	0
-0.6	-1.0	0	-0.52
-1.0	-2.0	-0.52	-3.00

Then, *y* values (*y* axis) (for curve L_m) were found by substituting different *m* values to equation *(3)*, while the corresponding skewness values - *x* axis were experimentally found. A large amount of *x, y* pairs were collected to have a smooth, precise curve as figure 3 therefore the actual fuzzy relationship(s) could be formulated for all skewness value ranges in Table 2 enclosed within *l*1(lower bound) and *l*2 (upper bound). Curve *(1-y)* (for H_m) could be also plotted accordingly (Since y values known). Such an actual fuzzy membership function (When H_m=-1.0, L_m=-2.0, *l*1=-0.52, *l*2=-3.00, refers to last row of Table 2.) is as figure 3.

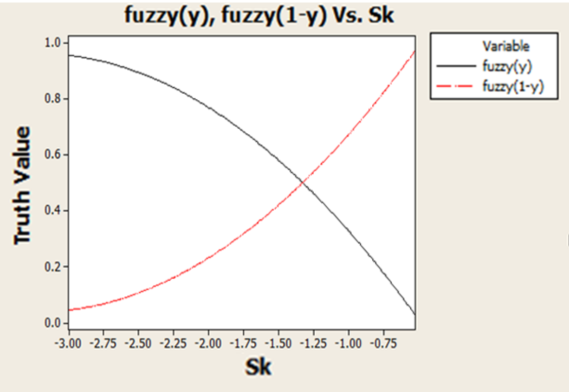


Figure 3: Actual fuzzy membership function of m and skewness

Finding the relationship with the Intersect (*c*) of the motion path equation and the graph features was another key milestone. This was also successfully achieved with many experiments. Another interesting finding, the relationship between the intersect(*c*), gradient *(m)* and the initial apparent size(s) was derived. The experimental findings are graphically illustrated in Figure 4.

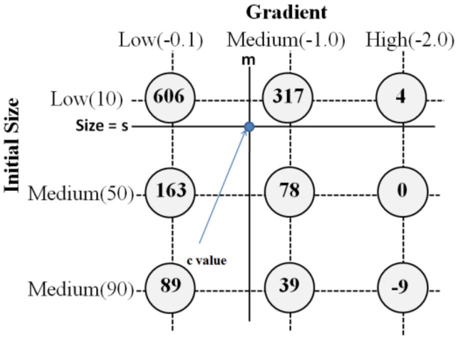


Figure 4: Fuzzy relationship between *c*, *m* and initial apparent size(s)

The numerical values within circles are the actual *c* values at a particular fixed gradient and initial size. For example, the top left most value, 606 is the *c* value when *m*=-0.1 and initial apparent size is 10. It was noticed that, the *c* is varying with *m* following a linear manner approximately, but *c*'s variation with initial apparent size is non-linear, but maintains a fuzzy relationship.

The variation of *c* with respect to change of initial apparent size, while keeping *m* constant, was more scientifically studied. The variation of *c*, with respect to initial size change, at constant gradients *m*=-0.1 and *m*=-1.0 respectively was modeled by Minitab 16 statistical analyzer and was as below:

case 1 (m=-0.1) :

$$\mathrm{c} = 849.6 - 32.46s + 0.5116s^2 - 0.002789s^3. \quad (4)$$

case 2 (m=-1.0) :

$$c = 441.0 - 16.65s + 0.2507s^2 - 0.001294s^3. \quad (5)$$

Approximately;

$$c(5) = 2.c(4). \qquad (6)$$

This implies that, *c* value is given by a common function is multiplied with a constant *k*, i.e. in above case, *k* = 2.0. This can be mathematically represented as:

$$c = k \,.\, f(s). \; (7)$$

where *k* is a constant at when gradient m is maintained uniformly, otherwise *k* becomes a variable, that is determined by the gradient *m*. Therefore, in order to find *c*, it was obvious to find the general function of *f*(*s*), then the correlation of *k* with gradient *m*. Below is the procedure followed to find the *f*(s).

Again, hypothetical fuzzy functions were built as Figure 5 and the actual fuzzy relationships were also found on experimental basis in Figure 6.

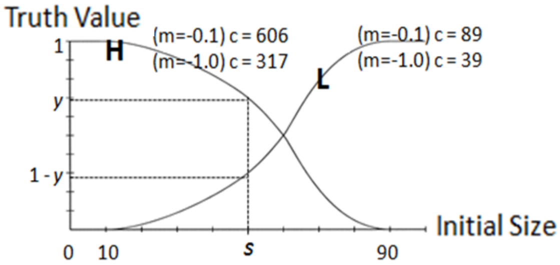


Figure 5: Hypothetical fuzzy membership function of c vs. initial apparent size

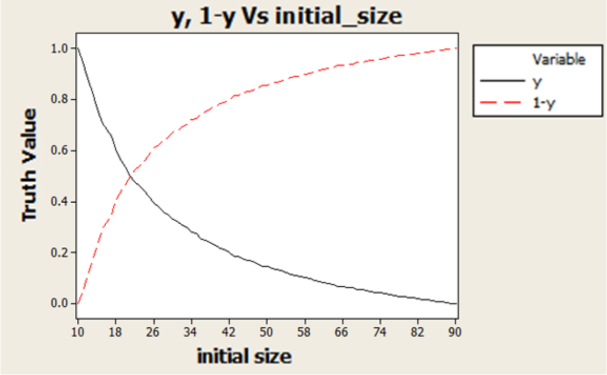


Figure 6: Fuzzy membership function of c with initial size

Then the c value is given by:

$$c = (\, y * H \,)+[\, (1 - y) * L \,]. \qquad (8)$$

Since collected real world data of initial size and *c* value pairs are available, a value matrix of *y* for corresponding initial sizes was found on experimental data, hence the function *y* could be plotted.

As *y* is known from the above fuzzy membership function of *c* with initial size, the two candidate *c* values (for two different fixed gradients) named as c_1 and c_2 is found.

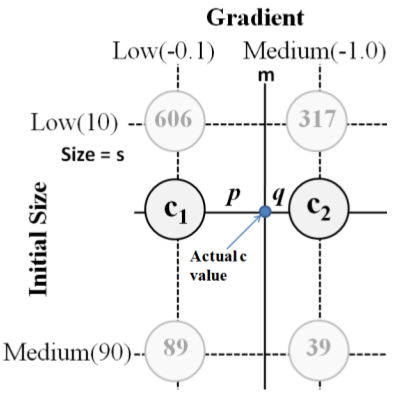


Figure 7: Finding actual c, in between c1 and c2

When the ratio of c_1, c_2 connector line divided by the point "Actual *c* value", (*p:q*) is known (assuming relationship between *c* and *m* is linear), final *c* value can be estimated. The final value for intersection *c* is given by the fuzzy membership function specified in Figure 8 as its *y* value.

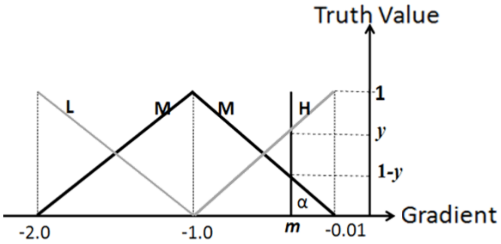


Figure 8: Fuzzy membership function of gradient and c

Once both gradient-m and intersection-c is found using the fuzzy mathematical model, the predicted equation of the motion path of the obstacle is finalized.

In order to test the theory in real world applications, a mobile robot in Figure 9 was especially prototyped. The relationships between the desired angle to be turned, encoder (Figure 10) count, the radius of robot / drive wheels was mathematically derived. And rules i.e. distance to be moved away depending upon an estimated risk factor were compiled based on experimental observations. Instructions to be executed in order to avoid obstacles one the predicted motion path is given, were stored on a Single-Chip microcontroller based Arduino Mega 2560 powered embedded system that handles the task, controlling the actuators of the model robot.

4. EVALUATION

For a sample size of 60 test cases, the implemented Fuzzy Mathematical Model possessed a maximum error for estimating the gradient of the object's motion path m as 0.089 while the maximum m error reported was 13.33 % as a percentage. The average m error reported was 0.021, meanwhile keeping an average m error as lower as 2.78 %. The result can be further summarized with an average m accuracy: 95.55 %.

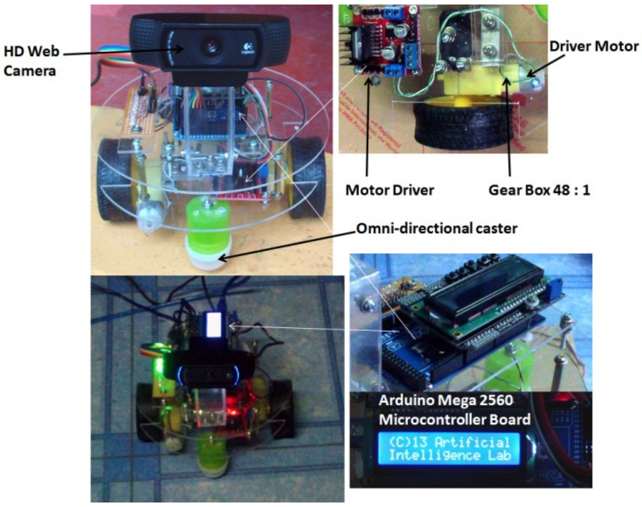


Figure 9: The mobile robot

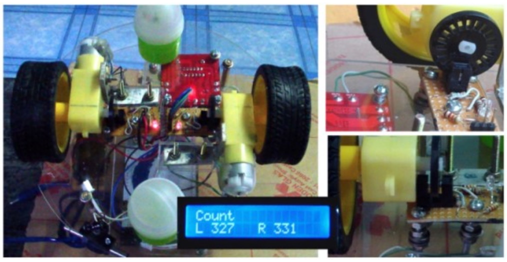


Figure 10: Velocity encoders

Prediction of intersection c by the Fuzzy-Mathematical Model is as follows:

The maximum *c* error reported was 30 (only one instance, in most cases it was below 15), while it was a maximum of 48 % as a percentage. The average *c* error reported was low as 5.12, which is 6.34 % as a percentage. (Figure 11 denotes the statistical analysis results). However it could maintain an average c accuracy 91.99% (approximately 92%) for the whole cases.

In general, if the minimum of m accuracy and c accuracy concerned, the model is precise as to produce predictions with an average accuracy of 92%.

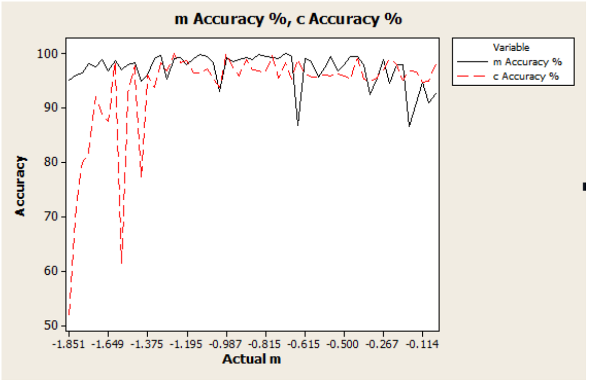


Figure 11 : Accuracy of estimated m and c, as a percentage

5. DISCUSSION AND FUTURE WORK

Experimental results and statistically analyzed evaluation reports produced with proper analytical techniques convey that the Fuzzy-Mathematical approach is appropriate to model the motion behaviour of an observed object and recover its motion path using the apparent size variation graph as the primary resource. The concept is appropriate, applicable and adaptable with real time applications. In addition, it provides benefits such as consuming a minimal computational resource, since it is an extremely nature-inspired optimized technique. Other than the less computational overhead, monocular vision is adequate instead stereo-optics where complicated mathematics and mechanisms required for calibration, align and depth estimation.

The theory postulated here can be further extended to model motion behaviours of objects in 3-D coordinate systems and having complicated and variable motion paths rather than simple linear motion.

A blind guidance system when voice instructions integrated, or a directly controlled electrically powered wheel-chair that safeguards the disable user from risky objects i.e. incoming vehicles, would be possible real world applications of the concept innovated.

REFERENCES

[1] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng, *Learning Depth from Single Monocular Images* [Online], Available: http://ai.stanford.edu/~asaxena/learningdepth/NIPS_LearningDepth.pdf

[2] Ashutosh Saxena, Min Sun and Andrew Y. Ng, *Make3D: Depth Perception from a Single Still Image* [Online], Available: http://robotics.stanford.edu/~ang/papers/aaai08-Make3dDepthPerceptionSingleImage.pdf

[3] Francisco Bonin-Font, Alberto Ortiz, Gabriel Oliver, *Visual Navigation for Mobile Robots: a Survey,* [Online], Available: http://dmi.uib.es/~fbonin/survey.pdf

[4] Yoko Watanabe, *Stochastically Optimized Monocular Vision-Based Navigation And Guidance* [Online], Available: http://smartech.gatech.edu/jspui/bitstream/1853/22545/1/watanabe_yoko_200804_phd.pdf

[5] Animesh Garg, Anju Toor, Sahil Thakkar, Shiwangi Goel, Sachin Maheshwari, Satish Chand, *Object Identification and Mapping using Monocular Vision in an Autonomous Urban Driving System* [Online], Available: http://www.ijcte.org/icmv/ICMV2010/136-ICMV2010-W12016.pdf

[6] Tobias Low, Antoine Manzanera, *Ground-Plane Classification for Robot Navigation* [Online], Available:http://www.ensta-paristech.fr/~manzaner/Publis/icarcv10.pdf

[7] Nicolau Leal Werneck, Anna Helena Reali Costa, *Mapping with Monocular Vision in Two Dimensions* [Online],Available: http://nwerneck.sdf.org/almoxarifado/WerneckCosta10.pdf

[8] Chau Nguyen Viet, Ian Marshall, *VISION-BASED OBSTACLE AVOIDANCE FOR A SMALL, LOW-COST ROBOT* [Online], Available: http://eprints.lancs.ac.uk/27256/1/27256.pdf

[9] Jiandong Tian, Yandong Tang, *Learning and Vision-Based Obstacle Avoidance and Navigation* [Online], Available: http://ir.sia.ac.cn/bitstream/173321/9008/2/ZZZJ000014.pdf

[10] Jan Hoffmann, Matthias J¨ungel, Martin L¨otzsch,*A Vision Based System for Goal-Directed Obstacle Avoidance* [Online], Available: http://www2.informatik.hu-berlin.de/~juengel/papers/hoffmann-juengel-loetzsch-rc04.pdf

[11] Jan Hoffmann, Matthias J¨ungel, Martin L¨otzsch, *A Vision Based System for Goal-Directed Obstacle Avoidance used in the RC'03 Obstacle Avoidance Challenge* [Online], Available: http://martin-loetzsch.de/publications/rc04-obstacle.pdf

[12] Rafaelmun˜ Oz-Salinas, Eugenio Aguirre, Miguel Garci´A-Silvente, Antonio Gonza´ Lez, *Door-detection using computer vision and fuzzy logic* [Online], Available: http://www.wseas.us/e-library/conferences/athens2004-b/papers/474-224.pdf

MaSIO – a Solution to Information Overflow in Agriculture

B. Hettige[1], A. S. Karunananda[2], G. Rzevski[3]
[1]Department of Statistics and Computer Science, University of Sri Jayawardenapura, Sri Lanka
[2,3]Faculty of Information Technology, University of Moratuwa, Sri Lanka
buddditha@yahoo.com[1], asoka@itfac.mrt.ac.lk[2], rzevski@gmail.com[3]

Abstract— **Information overflow has been a hindrance to search for information even by computing experts and subject matter experts. The situation is worst with the persons who have limited computing knowledge and unfamiliarity in the relevant subject. We argue that technological advancements in computing could provide solutions for the said issue. Our research has been inspired by the power of Multi Agent Systems (MAS) technology to model complex systems encompassing large number of distributed and interconnected entities that might change over the time. We have developed a Multi Agent System, MaSIO, to address the above issue. MaSIO mimics a scenario where an IT expert communicates with reliable subject matter experts and find information. In MaSIO, the roles of an IT expert and the subject matter experts have been implemented as Agents. MaSIO has been implemented as a Java-based generic solution. This paper presents customization of MaSIO for the domain of Agriculture. The paper also demonstrates how MaSIO extract reliable information regardless of the level of computing literacy of user.**

1. INTRODUCTION

Information is the most valuable resource for the development of mankind. Multifaceted developments in computing technology have contributed to storing and processing of information beyond the expectations. In the modern world people are geographically distributed, yet they are connected as if everybody lives in a single village. Unlike those days, one does not necessarily goes to the source of information to access and the use of information, yet it can be done through a notebook computer or a mobile phone. Many information are also accessible free of charge. There is no central authority to control the dissemination of information, and this leads to an issue of reliability of information published by some people. More seriously, there is an information overflow in huge knowledge sources such as the Internet. As such, for example, a person with little or no knowledge in a certain subject area might be misled when searching for information on the Internet. Of course, technology itself made the world so resourceful, yet unfortunately everyone cannot be benefited. The issue of information overflow is analogous to the fact that development of complex road networks does not guarantee to improve transport facilities for everyone. Such road networks will be useful only for the people who know how it can be used.

Undoubtedly, information overflow [1] is an issue arisen out of advancements of computing technologies. Therefore, we argue that solution to this issue should also come from the computing technology. However, solutions such as Search engines (e.g. Google [2], Yahoo [3] and Bing [4]) cannot amicably address the issue. Those solutions expect a person to have a certain level of IT skills and knowledge about the area, which is interested to the person. Therefore, such solutions work well only for the persons who are experts in both computing and the subject matter. It is evident that there are many subject matter experts who are not necessarily experts in computing. This needs not to be challenged.

Our research has revealed that the Multi Agent Systems technology [5] has been able to model the interconnected distributed systems involving many entities that are in fact dynamically changing over the time. As such, we have postulated to exploit the power of Multi Agent Technology to enable easy access to the huge reservoir of knowledge on the Internet by a person who is neither a subject matter expert nor a computing expert. Our Multi Agent System – MaSIO, an acronym for Multi Agent Solution for Information Overflow, for addressing the issue of information overflow can be customized for any subject area and persons with little computing skills. MaSIO also make ease of information searching by computing experts and the subject matter experts. This paper demonstrates how MaSIO has been customized to use for the Agriculture domain.

MaSIO has been design through our own agent development framework, MaSMT [6], which is specially design for the development of Machine

Translation systems. MaSMT is a java based multi agent system development framework that provides two types of agents as ordinary agents and dispatcher agents. A dispatcher agent consists with number of ordinary agents within its control. The framework primarily implements object-object communication, XML-based data passing and MySQL database connectivity for manipulation and use of the domain ontology.

MaSIO consists of two dispatcher agents (GUI agent and context manager agent) and number of client agents (ordinary agents) which are available in the MaSMT framework. The GUI agent provides graphical user interface of the system. The context manager agent and its client agents are used to read the context from the internet. MaSIO system maintains user profile. Therefore user needs to login and complete the profile before access the searching tools. After user login to the system, MaSIO manager download and activate relevant user profile from its ontology (database). According to the user profile context manager agent creates its clients to access the internet at the initial stage.

After client agents are being created context manager agent wait for input of key word(s) from GUI agent. If key word(s) are received from GUI agent, the context manager send the information to its clients. Clint agents use this key word(s) and find the relevant information from internet. At this point client agents can see what are the information another agent has been collecting. If same information is collected by another agents then, the agent tries to find some other information. In addition to the above client agents also recorded the successful visit to the page (if page consist of the result for a particular key word, system count as a successful visit). Hit ratio is calculated using the successful visit. If the particular site has more than 70% hit ratio then site is automatically added to the user profile. If hit ratio is less than 40% than system removes the particular web site from the user's profile.

The rest of the paper is organized as follows. Section 2 gives brief description about existing information systems, then section 3 gives multi-agent approach to solve issues of the information overflow. Section 4 gives design of the MaSIO with brief description of each module. Then section 5 reports how systems work for the agriculture domain and finally section 6 gives conclusion and further work of the project.

2. EXISTING SYSTEMS

At present billons of websites available on the internet and these sites consist of huge amount of data. According to lager number of data availability, it is difficult to identify which data is more relevant? In addition to the above, many of these existing search engines provides only keyword searching. Therefore, it is difficult to make sure that certain keyword is related to the real meaning of the text, how to remove garbage due to extensive usage of keywords and how to understand the context of the request etc. [7]. However, existing search engines (Google, yahoo) use different mechanism to search information from the web.

Google[2] is the most popular search engine in the world that hunts text from publicly accessible documents offered by web servers [1]. In addition to searching the web, Google identify specific information which are available offline or on specialized sites [8]. To obtain search results Google uses search-results page which is based on a priority rank called a "PageRank". PageRank is a link analysis algorithm used by the Google Internet search engine that assigns a numerical weighting to each element of a hyperlinked set of documents [9].

Yahoo [3] is another popular search engine which was the 2nd largest search directory on the web. Sine 2009 Yahoo Search would be powered by Bing. Yahoo! Search ranks results according to their relevance to a particular query by analyzing the web page text, title and description accuracy as well as its source. Yahoo spider indexes the full text of web pages, up to a 500,000 characters.

Bing [4] is another search engine from Microsoft. Bing uses back-end search infrastructure, with the goal of delivering faster and slightly more relevant search results for users [1].

In addition to the above, many other research have been done for the information searching on the web. Kanteev and others have developed Multi-agent based meta-search engine based on domain ontology [7][10]. This system search HTML pages via Internet, which is based on the semantic understanding of pages content by means of multi-agent technology. They have applied this development to the analysis of web-pages related to car industry.

Huang and zhongzhi had developed Multi-agent based web search system using the heterogeneous semantics as a solution for the issues on keyword-based searching [11]. This system provides efficiency and reliability of search.

Min-Huang and others [12] have developed a search engine named octopus, which was based on Internet Search Service (ISS). This octopus system provides personalized search facilities for the users. In addition to the normal services, personal information-filtering agent is used to achieve the functionality of personalized search in a search engine.

Enembreck and Barthes have developed Multi-agent based tool for internet search [13]. This open multi-agent system has been implemented with personal assistants, library agents, filter agents, and search agents.

However, these existing solutions do not meet the user requirement when persons who have limited computing knowledge and unfamiliarity in the relevant subject.

3. MULTI-AGENT SYSTEM TECHNOLOGY

Multi Agent Systems technology has emerged as a new software paradigm, which exploits the power of message passing as the key strategy for problem solving. Communication, negotiation and coordination among agents produce high quality solutions that cannot be generated by a single agent in its individual capacity.

Multi-agent systems are used to handle complex knowledge. In general any multi Agent system contains four key components namely Multi-agent engine, virtual world, ontology and interfaces [5]. The Multi-agent engine provides a run time support for agents. The engine starts as the first step of the system. Virtual world is the environment of the Multi agent system. Using this virtual world agents cooperate and compete with each other as they construct and modify the current scene. The Ontology contains conceptual problem domain knowledge of each agent.

Nowadays hundreds of well-established general purpose toolkits and frameworks are available for the development of Multi Agent Systems. Among others, JADE [14], Jason [15], SeSAm [16] and AgentBuilder [17] are the stranded Multi Agent System development frameworks. JADE (Java Agent DEvelopment Framework) is a software framework fully implemented in Java language. This framework provides supporting GUI tools for debugging and deployment phases in the multi agent development. Jason is an interpreter for an extended version of AgentSpeak [18]. AgentBuilder is an integrated software development tool that allows software developers to build agents quickly and easily without sound knowledge of multi agent technology. SeSAm (Shell for Simulated Agent Systems) is another framework that provides a generic environment for modeling and experimenting with agent-based simulation. These frameworks are especially designed to develop general purpose applications, machine learning and simulations of complex systems. To develop Natural language processing applications (especially Machine Translation Systems) we have developed our own Multi-agent System development framework, MaSMT.

MaSMT is a java based multi agent system development framework, especially designed for development of Machine Translation systems. MaSMT provides two types of agents as ordinary agents and dispatcher agents. A dispatcher agent consists of number of ordinary agents within its control. Further, dispatcher agents can directly communicate with other dispatcher agents and the ordinary agents in the swarm that are assigned to a particular dispatcher agent. Agents in a swarm can directly communicate only with the agents in the own swarm and the relevant dispatcher agent. The framework primarily implements object-object communication, XML-based data passing and MySQL database connectivity for manipulation and use of the domain ontology. Agent communication in the framework has been implemented to comply with FIFA-ACL specification [19].

4. DESIGN: MASIO

MaSIO system has been designed through the MaSMT framework. MaSIO systems run in different locations and use single MySQL database as the ontology. Figure 1 shows the top level architecture of the MaSIO systems.

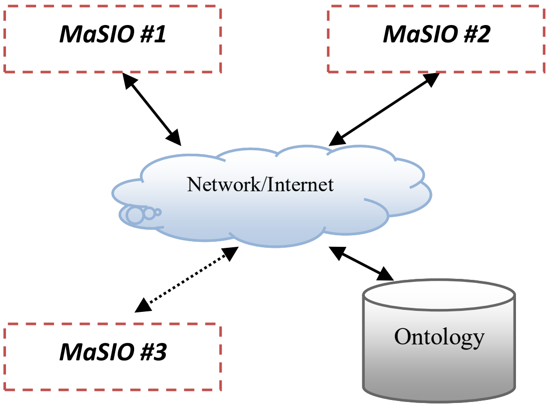


Figure1: Design architecture of the MaSIO

MaSIO system consists of two dispatcher agents (GUI agent and context manager agent) and number of client agents which are available in the MaSMT framework. The GUI agent provides graphical user interface of the system. This interface gives users to login, edit his/her user profile and change some system setting to internet access. Context manager agent is also a dispatcher agent that handles all the context reading process. Figure 2 shows the design of the context manager agent.

Context manager agent (java thread) manages its client agents. According to the MaSMT architecture, each manager can fully control its client agents. Therefore, dispatcher can create, remove or control its client agent(s). Dispatcher agent creates all its clients automatically at the initialization stage. This agent accesses the rule-base in the database (used to create client agents) and assigns each rule for a client agent. It means that the dispatcher creates an agent for a rule in the rule base. In addition to the above, dispatcher can directly access each agent and send messages directly for its clients. The dispatcher agent reads input massages from the global message queue and provides relevant tasks for the client agents. Also dispatcher can control the priority of the client agent and the stage of the clients. This facility removes the unnecessary work load form its client agents.

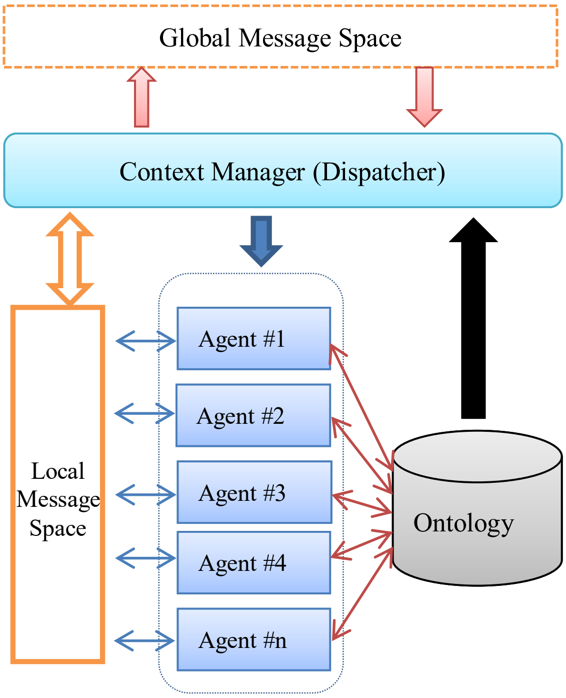


Figure 2: Top level Design of the Context Manager

The Ordinary/client agents work under the control of the dispatcher agent and each ordinary agent must have a dispatcher agent. Ordinary agent is a simple java program (Thread) which support limited task(s). These agents can communicate with each other through the messages space, through the use of peer-peer or object-object communication methods. This ordinary agent consists with local message queue, access rule, communication module and a ontology. Figure 3 shows the design of the ordinary agent. These agents are responded for the messages which are available in its local message queue. Each agent has been assigned for only the simple task and it respond only for the assign task (task is available as an access rule).

For instance, context agents respond only for the two messages which contain “get context” and “about you”. When the agent receive message “about you” from the message queue, it provides information about itself to the message sender by using message space. After receiving the message “get context”, it tries to do the context analysis with the support of the rule(s), site(s) and its ontology.

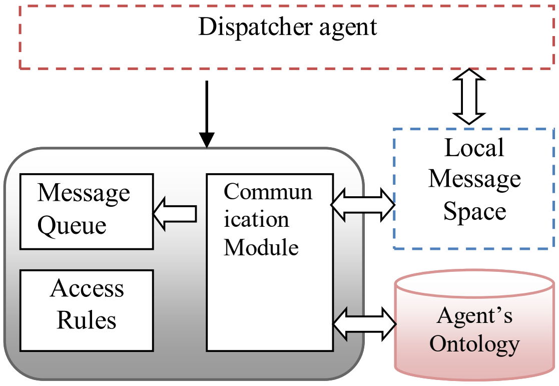


Figure 3: Design of the context agent

The communication module provides access to the ontology and the message spaces through the given media such as MySQL database Access, Objet-Object access, XML database access or network peer-to-peer access or client server access.

Local message space is the visible area of each agent in the local agent group (Swam). Each agent can directly communicate others through the local message space. This allows an object-object connection. Context agent can send or receive messages for this local message space. Each agent consists with a local message queue with public access. Dispatcher agent and other local clients can directly access this message queue to add messages.

Global message space is the message space which can be used to communicate through the system. By sending messages to the global message space other agents (other system) can communicate through the dispatcher agents. This message space is visible only for the dispatchers who are in the MaSMT framework. Dispatchers send messages to other dispatchers through the Global message space.

Ontology is the knowledge of the each agent (ordinary and dispatcher agents). Agent uses ontology to create the action. For instance, Context agent in the MaSIO system uses search domain, URL, and request parameter (GET method) as its ontology. Context agent points of view and each rule is also stored in its ontology.

Messages used to communicate with each other have been developed by using FIPA ACL message stranded [19]. ACL Message consists with Participant in communication: sender, receiver, reply-to, Content of message: content, Description of Content: language, encoding, ontology. Control of conversation: protocol, conversation-id, reply-with, in-reply-to, reply-by etc. MaSMT messages are compatible with the FIFA ACL messages.

Moreover, context agent also stores the number of visits to the site and number of successful visits. Hit ratio is calculated through number of successful visits.

$$Hit\ Ratio = \frac{Sucessful\ visit}{Totl\ number\ of\ visit}\ X\ 100$$

If hit ratio is more than 70% and the total number of visits is more than 25, then the particular site is added to the particular domain (to public access) or else stored in the user specific domain. If hit ratio is less than 40% and total number of visits are more than 25, then the particular site is removed from public access domain and or particular user specific domain.

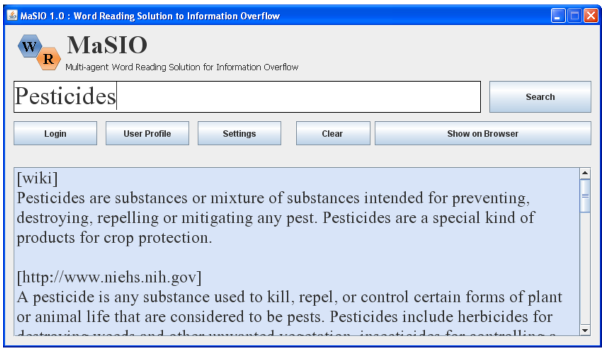
After collecting available information on the web source, context agent communicates with other agents to remove duplicate contexts. If duplicates do not exit then the agent send the context to the GUI agent through the context manager agent. The GUI agent which received this information through the global message space and creates the output web page to show the information. Finally GUI agent displays the above created web page through the web browser. To communicate with a browser GUI agent uses its web server engine.

5. HOW SYSTEM WORKS

MaSIO system has been implemented using Java application and can run on PCs. System requires internet access to search information and communicate with its ontology. A new user can be created through the system and the system uses available information to create its initial ontology. Figure 3 shows the user interface of the MaSIO.

After user successfully login to the system, GUI agent download and activate relevant user profile from its ontology (database). According to the user profile, context manager agent (dispatcher agent) creates its clients to access the internet (By default, system consist of 3 agents to search information: namely Google agent, Yahoo agent and Wikipedia agent). Context manager agent creates all its clients automatically at the initialization stage.

After the client agents creation, Context manager agent wait for a keyword(s) input from GUI agent. The GUI agent send user entered keyword(s) to the context manager through the global message space. If user gives keyword(s) to search, Context manager agent gets these key word(s) from the GUI agent. After receiving keyword(s), context manager agent send these keywords to the context agents. In the initial stage Google agent, Yahoo agent and Wikipedia agent get the keywords and try to search the context. Google and yahoo agent get reference from its search pages and search the original context what actually its available (this process is same as a person reading the Google search page using actual reference).


MaSIO 1.0 : Word Reading Solution to Information Overflow
W
R
MaSIO
Multi-agent Word Reading Solution for Information Overflow
Pesticides
Search
Login
User Profile
Settings
Clear
Show on Browser

[wiki]
Pesticides are substances or mixture of substances intended for preventing, destroying, repelling or mitigating any pest. Pesticides are a special kind of products for crop protection.

[http://www.niehs.nih.gov]
A pesticide is any substance used to kill, repel, or control certain forms of plant or animal life that are considered to be pests. Pesticides include herbicides for

Figure 4: GUI of the MaSIO

If information is available then agent marks the URL and also update the hit ratio. This update process is used to update profile automatically. In addition to the above, MaSIO also support adding new web resources into the its user profile manually. As an example, a user can add some online resources according to its domain. This is the way how MaSIO system can be customized for any selected domain. In the agriculture domain it has been done by adding new agricultural information sites for the user profile such as Wiki Goviya [20], web site of the department of agriculture, Sri Lanka [21] and Goviya [22].

After collecting the information, MaSIO provides output result as a web page therefore user can easily read the given web context as well as search more information from the web through the given reference links.

6. Conclusion and Further Works

This paper has reported, a multi-agent solution, MaSIO for addressing the issue of information overflow. The MaSIO system can be customized for any subject area and for people with little computing skills. MaSIO also makes information searching easy by using computing experts and the subject matter experts. MaSIO system has been designed using MasMT framework. Through the MaSIO system we have successfully tested the customization ability of the domain of agriculture. As a conclusion MaSIO is capable to use as a tool for the information searching which gives a reasonable solution for the problem of Information overflow. Use of the MaSIO system as semantic handling tools in the English to Sinhala Machine Translation has been considered as further work of this project.

REFERENCES

[1] Wikipedia,[online], Available: http://en.wikipedia.org

[2] Google [online], Available: http://www.google.com

[3] Yahoo [online], Available: http://www.yahoo.com

[4] Bing [online], Available: http://www.Bing.com

[5] G. Rzevski, 'A new direction of research into Artificial Intelligence', Sri Lanka Association for Artificial Intelligence 5th Annual Sessions. – 2008

.

[6] B. Hettige, A. S. Karunananda, "Multi-agent System Technology for Morphological Analysis", Accepts to present, 9th Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI), 2011

[7] Argo – Intelligent multi-agentMeta-Search Engine,based on domain ontology, [online], Available: http://www.magenta-technology.com

[8] Google Guide, Online URL :http://www.googleguide.com/

[9] B. Sergey, P. Lawrence,The Anatomy of a Large-Scale HypertextualWeb Search Engine [online], Available: http://ilpubs.stanford.edu:8090/361/1/1998-8.pdf

[10] M. Kanteev, I. Minakov, G. Rzevski, P. Skobelev, S. Volman, Multi-agent meta-search engine based on domain ontology, Proceeding of the 2nd international conference on Autonomous intelligent systems: agents and data mining, AIS-ADM'07, 2007.

[11] R. Huang, S. Zhongzhi, Multi-Agent Based Web Search with Heterogeneous Semantics, Agent Computing and Multi-Agent Systems, Pages 158 - 170, 2009.

[12] H. Min-Huang, C. Yue-Shan, Y. Shyan-Ming, L. Winston, An Agent-Based Personalized Search on a Multi-search Engine Based on Internet Search Service, Lecture Notes in Computer Science Volume 1983, 2000, pp 404-409.

[13] F.Enembreck, J. Barthes, Multi-agent based internet search, International Jurnal of Product Lifecycle Management, Vol.2, No.2, pp.135 - 156, 2007.

[14] JADE, [online], Available: http://jade.tilab.com/

[15] Jason[online], Available: http://jason.sourceforge.net/wp/

[16] SeSAm, [Online], Available: http://www.simsesam.de/

[17] AgentBuilder,[online], Available: URL: http://www.agentbuilder.com/Documentation/Lite/

[18] M. Dinverno, M. Luck, Engineering AgentSpeak(L): A Formal Computational Model, Logic Computat., Vol. 8 No. 3, pp. 1-27, 1998.

[19] FIPA:Agent Communication specifications, [online], Available: http://www.fipa.org

[20] Wiki Goviya, [online], Available: http://www.goviya.lk

[21] Department of Agriculture, [online], Available: http://www.agridept.gov.lk/

[22]Goviya [online], Available: http://goviya.com

Modeling of Hidden Layer Architecture in Feed-forward Artificial Neural Networks

N. M. Wagarachchi[1]**, A. S. Karunananda**[2]
[1,2]Faculty of Information Technology, University of Moratuwa.
[1]mihirini@is.ruh.ac.lk, [2]asoka@itfac.mrt.ac.lk

Abstract— **Determining the appropriate architecture of a neural network is one of the main unsolved problems in artificial neural networks. The architecture has a great impact on its generalization power. More precisely, by changing the number of layers and neurons in each hidden layer, generalization ability can be significantly changed. Therefore, the architecture is crucial in artificial neural network and determining the hidden layer architecture has become a research challenge. In this paper, a pruning technique is presented to obtain an appropriate architecture by using the delta values of hidden layers. Pruning is done by using the delta values of hidden layers. The proposed method has been tested with three benchmark problem datasets in artificial neural networks and machine learning namely, breast cancer, Iris and car evaluation. The experimental results show that the modified architecture with lesser number of neurons performs better in generalization than that of the back-propagation algorithm.**

1. INTRODUCTION

Artificial neural networks (ANN) have been applied in many real world problems such as signal processing, pattern recognition, and medical diagnosis problems [1], [2], [3], [4]. Although they provide highly generalized solutions, we find several unanswered problems when using ANNs. Determining the most appropriate architecture of artificial neural network is identified as one of those major problems because, the performance of a neural network strongly depends on the size of the network. By increasing the number of layers generalization ability can be improved. But this solution may not be computationally optimized. On the other hand, too many hidden neurons may over-train the data which may cause poor generalization. Also, too few neurons under-fit the data and hence, network may not train the data properly. Thus, too many or too few neurons results in bad generalization. Therefore, determining the most suitable architecture is very important in artificial neural networks. As such, a large number of research have been carried out to model the hidden layer architecture by using various techniques. These techniques can be categorized as pruning methods, constructive methods and evolutionary methods. Pruning algorithms start with an oversized network and remove nodes until the optimal architecture occurs [5],[6],[7],[8],[9]. Constructive algorithms [10],[11],[12],[13] work the other way round. They build the appropriate neural network during the training process by adding hidden layers, nodes and connection weights to a minimal architecture. However, most of these methods are confined to networks with small number of neurons or a single hidden layer neurons and hence, they have not addressed the existing problem of hidden layer architecture properly.

In this paper, we have proposed a novel pruning algorithm to model hidden layer architecture based on back-propagation training. The algorithm reaches to the optimal architecture in two phases. Initially, it decides the number of hidden layers required for the best generalization. Usually, generalization power can be improved by increasing the number of hidden layers. However, to train a network with a large number of hidden layers, it needs more effort and more time. Therefore, it is essential to decide the number of hidden layers for the most effective neural network. To decide this number we use the 'Accuracy Ratio' (κ). The number of hidden layers corresponds to the greatest κ is to be considered as the number of hidden layers in the most effective network. In the second phase, algorithm prunes the network by removing unimportant nodes from the hidden layers. Accordingly, we selected to delta values of hidden layers to identify the unimportant nodes. This is based on the fact that the delta values of the hidden layers are used to compute the error term of the next training cycle. Hence, delta value is a significant factor in the error term. Thus, delta values are used to identify the less saliency neurons and remove them from hidden neurons, so that error term comes to desired limit faster than the backpropagation training. Moreover, delta depends on the several other parameters such as connection weights learning rate and the activation function. Hence, it automatically considers these parameters when removing the nodes from hidden layers. In [14] authors have discussed how to use delta values in optimization of hidden layer architecture. This paper presents a modified algorithm based on a pruning technique in designing artificial neural networks. This algorithm determines the optimal solution by removing unimportant nodes from feed-forward multilayered artificial neural networks.

The rest of the paper is organized as follows. The related works done by other researchers are discussed in section II. Section III describes the new algorithm and how to use delta values in optimization of hidden layer architecture. The experimental method and results are discussed in section IV. Finally, section V presents the conclusions.

2. RELATED WORKS

As the architecture of artificial neural network is very important there are many attempts in modeling hidden layer architecture. These approaches are based on pruning techniques, constructive techniques or both pruning-constructive techniques and some other evolutionary methods.

2.1 Pruning Algorithms

The objective of a pruning algorithm is to obtain the optimum structure of network by incrementally removing low-saliency neurons from the architecture. Optimal Brain Damage (OBD), presented by Le Cun and Jhon Denker [5] is known as the most well-known and commonly used pruning algorithm, where parameters with low-saliency are detected based on the second derivative of the objective function and removed from the network. The intention of OBD is that, it is possible to obtain a network which perform in same manner (or better), by eliminating about half of its weights. However, a major issue arises with the enormous size of the Hessian matrix. This causes computational barriers and also takes considerable time to train. Hence, it assumes the Hessian matrix is diagonal.

To avoid the limitations of OBD, authors in [6] introduced Optimal Brain Surgeon (OBS) and claimed that Hessian matrix is non-diagonal for every problem and hence, weights elimination of OBD may not be accurate. Even though, it argues that OBS is better than OBD and yields better generalization, much computational limitations arises especially working with the inverse of the Hessian matrix.

Giovanna et al. [7] proposed a conjugate gradient algorithm in least-square sense to prune neurons after training the network. It reaches the optimal size by removing neurons successfully from a large trained network and then adjusting the remaining weights to maintain original input-output behavior. They claim that this algorithm yields better results relative to other existing methods. However, in this research it does not consider a multi-layered architecture and hence, it deviates from our aim. On the other hand, our goal is to identify and prune the neurons before training the network. In addition, we concentrate on removing a cluster of neurons rather than removing one neuron at a time.

Authors in [8] have used hybrid sensitive analysis and adaptive particle swarm optimization (SA-APSO) algorithm to determine the minimal architecture of artificial neural networks. Firstly, prune neurons with less saliency using the impact factor defined by,

$$\boldsymbol{ImF_i = \sum w_{ji}^2 \sigma_i^2}$$

The neurons with *ImF* less than a certain threshold value $\boldsymbol{\sigma_{ir}}$ have been identified as removable neurons with less saliency. Each removal is tried to replace with some other suitable neuron which has the similar effect. The similarity is defined by using the correlation coefficient. If the two neurons are highly correlated they can be merged. Anyhow, this research is restricted only for two layered networks. But there is no such rule that two layer network can solve almost all the problems. Also there are some problems which require three layer networks [15].

Reference [16] claims that for a particular network configuration, there is a continuous set of weights and biases that have the same value of the cost function

$$\sum_{n,p} \left(d_{pn} - o_{pn}\right)^2.$$

Where the sum is taken over *n* outputs and *p* training patterns while *d* and *o* are desired and observed outputs respectively. This set of weights defines a contour of the cost function in the space of all possible weights. In order to determine the optimum architecture, a network trained by Back propagation algorithm can be obtained by eliminating weights which are close to zero.

2.2 Constructive Algorithms

In constructive neural networks the network structure is built during the training process by adding hidden layers, nodes and connections to a minimal neural network architecture. However, determining whether a weight should be added and whether it should be added to the existing hidden layer of a new layer is not straightforward. Therefore in most algorithms, pre-defined and fixed number of nodes are added in the first hidden layer and the same number of nodes are added to second layer and so on [12]. This number is crucial for the better performance of the network and it reduces the network size as much as possible to avoid the complex computations during the training. The cascade correlation algorithm (CCA), proposed by S. E. Falhman [10] is a well known and mostly used constructive algorithm. This algorithm is proposed as a solution to problems such as local minima problem. The dynamic node creation (DNS) [11] algorithm is supposed to be the first constructive algorithm for designing single layer feed-forward networks dynamically. Authors in [13] improved the adaptive learning algorithm for multi-category tiling constructive neural networks for pattern classification problems. Md. Moniral et al [17] have proposed an adaptive merging and growing algorithm to design artificial neural networks. This algorithm merges and adds hidden neurons repeatedly or alternatively during the training, based on the learning ability or the training progress of the artificial neural network.

2.3 Evolutionary Methods

Other than the pruning and constructive methods some researchers have suggested evolutionary methods to model the hidden layer architecture based on various mathematical concepts.

An alternative method proposed by Stathakis [18] was to obtain a near-optimal solution by searching with a genetic algorithm. This research was carried out to determine the optimum architecture and it claims that obtained network is more efficient in problem classification. Although it gives a very small error in the training cycle, topology of network is comparatively large. Hence, it is difficult to apply especially when there is large number of inputs in the network. In [19] authors have presented an approach selecting the best number of hidden units based on some mathematical evidence. According to their suggestion number of hidden nodes can be given as

$$n = C\left(\frac{N}{dlogN}\right)^{1/2}$$

Where N is the number of training pairs and C is a constant. The approach is to increase C and train the feed-forward network for each n and then determine the optimal value for n which gives the smallest rooted mean square error. Although, they have applied this to medium-sized data set and there is no guarantee of applying the theory for large number of hidden neurons.

Even though methods to optimize the hidden layer architecture have been developed, they have not fully addressed the problem regarding the topology of multilayer artificial neural networks such as determining the number of required layers for better performances and then minimize the error in training.

3. NEW ALGORITHM

In this research we propose a new algorithm based on the Backpropagation training algorithm. Backpropagation algorithm is the most well-known and widely used among the optimization algorithms of artificial neural networks. Our proposed algorithm optimizes the artificial neural networks based on a pruning technique and reaches to the desired solution faster than the Backpropagation training. The main objective of the new algorithm is, while maintaining the same error rate; it decreases the size of the network and the number of training cycles with better performances.

At the first stage, it decides the number of hidden layers which gives the most efficient network. Then removes unimportant nodes from each layer to reach the optimal solution. Here we consider a network with n inputs and m outputs. The number of input/output training cycles is N. Let each network consists of M total number of hidden neurons. Then, train the different architectures of network by backpropagation algorithm. The number of hidden layers of network is h, where h = 1,2,3,... . The number of hidden neurons in each layer is $n_{H_i}, i = 1, 2, \dots, h$ where

$$\sum_i n_{H_i} = M$$

Define Accuracy ration (κ)

Where,

$$\kappa = \frac{Accuracy\ Rate}{Number\ of\ training\ cycles}$$

Then higher values of κ indicates that the network is more efficient. Hence, the number of hidden layers that provides the highest κ is to be considered as the most appropriate network.

After deciding the exact number of hidden layers (h), trim the network for better performance. Start with fully connected network with h hidden layers and M number of total hidden neurons. Initially, it identifies the less saliency neurons and removes from hidden layers during the training. After removing all the unimportant nodes network trains by using backpropagation algorithm until error $E(n)$ defines in (1) tends to zero.

Consider a fully connected network with n inputs, m outputs and h hidden layers. Hidden layers are denoted by $H_1, H_2, \dots H_h$ and the layer H_j contains n_{H_j} number of neurons.

The error of the n^{th} training cycle $E(n)$ can be written as

$$E(n) = \frac{1}{2}\sum_{k=1}^{m}(t_k(n) - o_k(n))^2 \qquad (1)$$

Where, t_k and o_k. denote the expected output and actual output on the neuron k respectively.

Let the error $e_k(n)$ be the difference between the desired and actual outputs of the k^{th} neuron at the n^{th} training cycle. That is

$$e_{\,k}(n) = t_k(n) - o_k(n) \qquad (2)$$

Then

$$E(n) = \frac{1}{2}\sum_{k=1}^{m}\left(e_k(n)\right)^2 \qquad (3)$$

When there are N input/output training sets total error of the network becomes

$$E_N = \frac{1}{2N}\sum_{p=1}^{N}\sum_{k=1}^{m}\left(t_{pk} - o_{pk}\right)^2 \quad (4)$$

The delta value of the output layer is defined as

$$\delta_k = f_o'(net_k)(t_k \; - o_k)$$

$$= \qquad f_o'(net_k)e_k(n) \qquad (5)$$

Where f_o is the activation function of the output layer. The delta value of the i^{th} neuron of the j^{th} hidden layer can be written as

$$\delta_i^{H_j} = f_{H_j}'(net_i)\sum_{k=1}^{n_{H_{j+1}}} w_{ki}\,\delta_k \quad (6\,)$$

Where f_{H_j} is the activation function defined for the hidden layerH_j. w_{ki} is the connection weight from i^{th} neuron of the j^{th} layer to k^{th} neuron of $(j+1)^{th}$ layer. δ_k is the delta value of k^{th} neuron in the hidden layer H_{j+1}. So that, the delta value of each neuron is computed by using the delta values of next adjacent hidden layer. After computing all delta values weights are updated according to

$$w_{kj}^{H_j}(n+1) = w_{kj}^{H_j}(n) + \eta\delta_k^{H_j}(n)f_{H_j}(n) \;\; (7)$$

Error $E(n+1)$ will be calculated by using the updated weights. Hence, $E(n+1)$ has been computed by using $\delta(n)$. Therefore, it is clear that there is a relation between $\delta(n)$ and $E(n+1)$.

The procedure begins with a fully connected multilayered artificial neural network. This approach is not restricted to only one hidden layer. Therefore, we consider a network with h number of hidden layers. Each layer contains a large number of hidden neurons. There are various upper bounds for the number of hidden neurons in ANNs [13],[15]. So that we have assigned a large number ($>{}^{N}\!/_{h}$) of neurons for each hidden layer. This number is much more than the maximum of the above upper bounds.

The delta values of hidden layers of the n^{th} training cycles $\left(\delta_i^{H_j}(n)\right)$ are used to compute the error of the $(n+1)^{th}$ training cycle $E(n+1)$. Thus, the correlation coefficient between the summations of delta values of hidden layer H_j at the n^{th} training cycle $\sum\delta_i^{H_j}(n)$ and $E(n+1)$ has been used to identify the less salient neurons. For our convenience, this correlation is denoted by r_{E,H_j}. i.e.o

$$r_{E,H_j} = corr\left(\sum_{i=1}^{n_{H_j}}\delta_i^{H_j}(n), E(n+1)\right) \quad (8)$$

A positive r_{E,H_j} implies that eliminating positive nodes from the hidden layerH_j, summation of delta values decreases and hence, $E(n+1)$ can be reduced. If r_{E,H_j} is negative, by removing neurons with negative delta values summation of delta values increases. Therefore error can be reduced. Thus eliminating nodes with positive or negative delta values according their correlation error $E(n+1)$ becomes smaller than that of earlier. Therefore, it converges faster than the backpropagation. On other words number of the training cycle requires to reach the solution becomes lesser relative the backpropagation training.

Equation (7) implies that when delta is zero

$$w_{ki}^{H_j}(n+1) = w_{ki}^{H_j}(n)$$

i.e. there is no update of weight $w_{ki}^{H_j}$. Therefore, neurons with zero delta values do not have significant effect in minimizing the error. So that by removing nodes with zero delta values, size of the network can be reduced without degrading the performance of the network. Therefore, by removing neurons with,

Small positive values when $r_{E,H_j} > 0$

and

Large negative values when $r_{E,H_j} < 0$

The size of the network can make smaller while reaching the desired solution faster than the backpropagation training. The method of removing nodes is described in the following algorithm.

3.1 The New Algorithm

Step 1: Design fully connected networks for hidden layers , $2, 3, \ldots$. Each contains *n* input and *m* output vectors respectively. Initiate the connection weights which are randomly chosen and normalized. Train the network using backpropagation algorithm for *N* input/output datasets and computeκ. *h* is the number of hidden layer of the network, which provides them maximum κ.

Step 2: Compute r_{E,H_j} $j = 1,2 \ldots h$

Step 3: Apply the first input and by using the same set of connection weights compute error,oo

$$E = \frac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2$$

Step4: Find the delta values of the last hidden layer H_h . If the correlation r_{E,H_h} is positive remove neurons with positive delta values those less than certain threshold value τ.i.e remove j^{th} neuron from H_h if ,

$$0 < \delta_j^{H_h} < \tau.$$

If r_{E,H_h} is negative, remove neurons with negative delta values if

$$\tau < \delta_j^{H_h} < 0$$

Let the remaining number of hidden olayers in H_h. be n_h,

Step 5: Use these delta to calculate delta of the neurons of $(h\text{-}1)^{th}$ layer. Remove all possible neurons as in step 4.

Step 6: update weights

$$w_{ki}^{H_j} = w_{ki}^{H_j} + \eta\delta_i^{H_j} f_{H_j}(net)$$

Step 7: Apply the next input and continue steps 3 - 6.

Step 8: Compute

$$E_N = \frac{1}{2N}\sum_{p=1}^{N}\sum_{k=1}^{m}(t_{pk} - o_{pk})^2$$

Step 9: Repeat the procedure until the optimum solution occur.

Step 10: After obtaining the optimum solutiono train the optimal architecture by using the back propagation algorithm.

4. EXPERIMENTS AND RESULTS

Experiments on deciding appropriate number of hidden layers and obtaining the optimal solution are discussed in this section. For these experiments, we used datasets chosen from three well-known benchmarking problems namely; car evaluation, breast cancer and Iris. More details on these data sets are available on [21] and [22]. These data sets have been used to explain many studies in artificial neural networks and machine learning. All the sets were divided in to two classes for training and testing purposes. Different architectures of the above problems were considered. The log sigmoid and linear functions were chosen as activation functions for hidden layers and output layer respectively. The learning rate in each case was 0.1. Network trained for *N* input/output training sets until

$$E_N = \sum_{p=1}^{N}\sum_{k=1}^{m}(t_{pk} - o_{pk})^2 < 10^{-4}$$

In order to determine the suitable number of hidden layers, training sets of size *N* were chosen from each problem and designed fully connected networks with layers 1, 2, 3 4 and 5. Each network was trained by Backpropagation algorithm. The accuracy of the results was tested by using the testing set and the accuracy ratio(κ) was computed.

Table1 shows the accuracy rate in each problem. *h* and *M* refer to the number of hidden layers and total number of hidden neurons respectively. TC denotes the number of training cycles. It is clear that generalization ability is improving when the number of layers is increasing. For example, in car evaluation problem, the accuracy rate of the output was 72.34% when there were two layers.

Table 1: Accuracy Ratio κ for Different Benchmarking Problems

Problem	N	h	M	TC	Accuracy Rate	Accuracy Ratio (κ)
Car Evaluation	100	2	120	16	72.34	4.52
		3	120	23	74.28	3.23
		4	120	521	78.53	0.15
		5	120	312	77.97	0.25
Breast Cancer	120	2	160	7	94.28	13.07
		3	160	7	96.89	13.84
		4	160	17	97.24	5.72
		5	160	55	97.24	1.77
Iris	50	2	60	7	74.46	10.64
		3	60	19	78.70	4.14
		4	60	306	79.33	0.26
		5	60	387	81.93	0.21

However, same data set agrees with 78.53%, for 4 hidden layers. A similar difference can be observed in breast cancer and Iris problems.

However, to train higher number of hidden layers it requires more time. When there are only 2 and 3 hidden layers in the car problem, network can be trained only by 16 and 23 training cycles respectively. But the numbers of training cycles require to train network with 4 or 5 hidden layer are much greater than that. Hence in some cases, the improvement of performance is negligible compared to the effort and time we put to train the networks. It can be described by the accuracy ratio. In this problem, when there are 2 or 3 hidden layers κ is much higher than that for 4 or 5 hidden layers. Hence, we can consider network with two hidden layers with the highest value of κ is the most appropriate network for this problem. Similarly, in cancer and Iris problems, networks with 3 and 2 hidden layers respectively provide the highest values for κ. The decision for hidden layers in each problem is shown in Table 2.

To test the modified algorithm consider the networks with hidden layers given in Table 2. The total number of hidden neurons (*M*) appearing in Table 1 is divided among the hidden layers as shown in Table 4. While training the data with Backpropagation algorithm, correlation coefficient was observed. Table 3 shows the correlation r_{E,H_j} for each layer. Then the network

Table 2. Number of hidden Layers in the most appropriate network

Problem	N	h
Car Evaluation	100	2
Breast Cancer	120	3
Iris	50	2

Table3. Correlation Coefficients

Problem	N	h	r_{E,H_1}	r_{E,H_2}	$r_{E,,H_3}$
Car Evaluati	100	2	0.85	−0.23	
Breast Cance	120	3	−0.29	0.52	0.79
Iris	50	2	0.07	0.42	

	N	Backpropagation Algorithm			Modified Algorithm		
		Initial Configuration	TC	Accuracy Rate	Modified Architecture	TC	Accuracy Rate
Car Evaluation	100	70 - 60	16	72.34	66 - 56	13	72.51
Breast Cancer	120	60 - 50 - 50	7	96.89	42 - 34 - 38	5	97.06
Iris	50	30 - 30	11	78.56	28 - 24	9	81.20

Table 4. Comparison between Backpropagation and New Algorithms

was trained by using the new algorithm. A benchmark comparison was done with Backpropagation algorithm. The required numbers of training cycles for Backpropation and the modified algorithm respectively, are shown in Table 4. Also it shows the number of hidden neurons in initial configuration and the modified architecture. Accuracy rate of each case was calculated. It shows that for every case, it was able to find an architecture which has lesser number of neurons and shows better performance. First example in car evaluation problem shows positive correlation for the 1st layer and negative correlation for the 2nd layer. Hence, by removing neurons with positive small delta values from 1st layer and negative large delta values from the 2nd layer, modified architecture was obtained. It needs 16 training cycles to reach error $E = 10^{-3}$ by backpropagation algorithm. But if we train it by using new algorithm within 13 training cycles, network tends to the desired limit and new model decreases number of hidden neurons by about 9%. Moreover, it has upgraded the accuracy rate from 72.34% to 72.51%. In the car problem it was able to remove about 20% of hidden neurons without degrading the performance.

Also in cancer problem with 3 layers of hidden neurons were reduced by 28% while improving the generalization ability. Fig. 1 shows how the error changes in two training processes. It is clear new algorithm decreases the error rapidly and reaches to the desired error faster than the Backpropagation training.

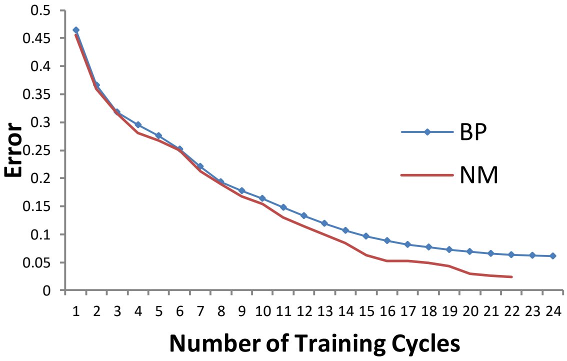


Figure1: The number of training cycles required in each training backpropagation and new algorithm.

5. CONCLUSION

In this paper, a new algorithm for multilayer hidden architecture was proposed. The algorithm is based on a pruning technique. First it decides the number of hidden neurons by using accuracy ratio. Then hidden neurons were pruned by using the delta values of hidden layers. The correlation between the summation of delta values of the n^{th} training cycle and error of the $(n+1)^{st}$ training cycle was considered to identify the less salient neurons. Moreover, neurons with zero delta values were recognized as unimportant neurons as they do not have much effect on updating the weights. Therefore, hidden neurons with small positive or large negative values (depending on the correlation) can be successfully used to reduce the size of the multilayer artificial network. In this paper a benchmark comparison was done with the backpropagation and it demonstrates that a new approach can be used to minimize the network by maintaining the same error rate as back propagation training with lesser number of training cycles. Further, the modified architecture can be obtained with very limited computations. Generally, 5% - 30% of neurons can be removed from hidden layers without degrading the performance of the output.

However, the performance of the network depends on several parameters such as weights of the connections, learning rate η, number of hidden neurons etc. Hence, some moderate changes will be done to these parameters as future improvements.

REFERENCES

[1] R. Setiono, W. K. Leow and J. M. Zurada, "Extraction of Rules from Artificial Neural Networks for Nonlinear Regression", IEEE Transaction of Neural Network, vol.13 May 2002, pp.564-577.

[2] S. M. Kamruzzaman, A. R. Hasan, A. B. Diddiquee and Md. Ehsanul, "Medical Diagnosis Using Neural Networks" Proceedings of the International Conference on Electrical and Computer Engineering (ICECE-2004) BUET, Dhaka , 2004, pp. 185-204.

[3] S. M. Karuzzaman and Md. M. Islam, "An Algorithm to Extract Rules from Artificial Neural Networks for Medical Diagnosis Problems", International Journal of Information Technology, vol. 12 No. 8,2006, pp. 41-59.

[4] F. Amato,A. Lopez,E. M. Pena-Mendez, P. Vanhara, A. Hample, J Havol, "Artificial Neural Networks in Medical Diagnosis", Journal of Applied Bio-Medicine, pp. 47-58.

[5] Le Cunn Y.,Denkar, Solla S. A. "Optimal Brain Damage," *Advances in Neural Information Processing Systems* D.S. Touaretzky, Ed, San Mateo CA:Vol 2 pp 598-605, 1990.

[6] Hassabi . B., Stork. D. G., "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon," *Neural Information Processing Systems*-vol 5,1993.

[7] Giovanna C., Anna M. F.,Marcello P., "An Iterative Pruning Algorithm for Feedforward Neural Networks," *IEEE Transactions on Neural Networks,* vol. 8, pp. 519-531, May 1997.

[8] Faisal Muhammad Shah, Md. Khairul Hasan, Mohammad Moinul Hoque, Suman Ahmmed, "Architecture and Weight Optimization of ANN Using Sensitive Analysis and Adaptive Paticle Swam Optimization," *IJCSNS International Journal of Computer Science and Network Security,* vol. 10, no. 8, August 2010.

[9] D. Sabo, X. H. Yu, "A New Pruning Algorithm for Neural Network Dimension Analysis", *IEEE International Joint Conference on Neural Networks*, IJCANN 2008, pp.3313 – 3318, Jun. 2008

[10] Fahlman S. E, "The Cascade-Correlation Architecture," May 1991.

[11] M. R. A. S, "Recursive Dynamic Node Creation in Multi Layer Neural Network," *IEEE Transactions on Neural Networks*, vol. 4, no. 2, 1993.

[12] P. C. Sudir Kumar Sharma, "Constructive Neural Networks: A Review," *International Journal of Engineering Science and Technology,* vol. 2, no. 12, pp. 7847-7855, 2010.

[13] Sridhar. S.S, "Improved Adaptive Learning Algorithm for Constructive Neural Networks," *International Journal of Computer and Electrical Engineering,* vol. 3, no. 1, 2011.

[14] N. M. Wagarachchi, A. S. Karunananda, “Optimization of Multi-layer Artificial Neural Networks Using Delta Values of Hidden Layers,” *IEEE Symposium on Computation Intelligence, Cognitive Mind and Brain* , pp. 80-86, April, 2013

[15] B. G. H. Don R. Hush, "Progress in Supervised neural networks," *IEEE Signal Processing,* vol. 10, pp. 8-39, 1993.

[16] A. N. Burkitt, "Optimization of the Architecture of Feed-forward Neural networks with hidden layers by Unit Elimination," *Complex Systems 5,* pp. 371-380, 1991J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[17] Md. Monirul Islam, Md. Abdus S, Md. Faujul A, Xin Y., "A New Adaptive Merging and Growing Algorithm for Designing Neural Networksf," *IEEE Transactions on Systems Man and cybernetics,* vol. June 2009.

[18] Stathakis D., "How many Hidden Layers and Nodes," *International Journal of Remote Sensing,* vol. 30, pp. 2133-2147, April, 2009

[19] Sxuxinang Xu, Ling Chen, " Novel Approach for Determining the Optimal Number of Hidden LayerNeurons for FNN's and its Application in Data Mining," *5th International Conference on Information Technology and Applications,* pp. 683 - 686, ICITA 2008.

[20] Baum E. B, Haussler D., "What Size Network Gives Valid Generalization" *Neural Computations*-January,1989.

[21] AFrank, " UCI Machine Learning Repository [.," Irvine, CA: University of California, School of Information and Computer Science., 2010. [Online]. Available: http://archive.ics.uci.edu

[22] L. Prechelt, “PROBEN1—“A set of neural network benchmark problems and benchmarking rules,” Faculty Informatics, Univ. Karlsruhe, Germany, Tech. Rep. 21/94,1994

WMAC: Web-Based Multi-Agent Solution for Agriculture Community

H. M. H. R. Jayarathna[1] and B. Hettige[2]
[1]Faculty of Information Technology, University of Moratuwa, Sri Lanka
[2]Department of Statistics and Computer Science, University of Sri Jayawardenapura, Sri Lanka
harshanajayarathna@gmail.com[1], buddhitha@dscs.sjp.ac.lk[2]

Abstract - **Multi-agent System Technology (MAS) is one of the powerful technologies which is used to solve real world problems. Compared with existing Multi-agent applications, web based solutions are more useful than the standalone applications. This paper presents web based multi agent system named WMAC which can be used to communicate with the people engaged in Agriculture industry. Web based multi-agent system uses a common MYSQL database as the ontology of each agent. Agents work as web clients and design through the PHP and AJAX technologies. System provides four types of agents as farmer, buyer, seller and instructor; which represent farmers, buyers, sellers and technical instructors in the agricultural community and which makes the communication among persons in the agricultural industry as required. The WMAC is a web based development of existing Java based standalone multi-agent system named AgriCom. The WMAC system has been successfully tested with the practical environment and successful results were achieved.**

1. INTRODUCTION

In traditional artificial intelligent systems such as Expert Systems, Genetic Algorithms, Artificial Neural Networks, Fuzzy Logic and Decision Trees it is considered that intelligence can be achieved via stored knowledge. Multi-Agent System (MAS) technology is the novel concept in the artificial intelligent field and this system does not contain pre stored knowledgebase as intelligence. Though MAS technology is considered as a sub area of Artificial intelligence (AI), the MAS technology is beyond the field of AI.

MAS software are having some advantages compared to conventional software such as being Distributed, Knowledge guided, messaging capability, emergent behavior, adaptive, self-organizing and evolving. In multi agent technology a large number of small programs are used which are called agents. Those small programs are run simultaneously and they to communicate among themselves to find suitable answers for problems.

Multi-Agent Engine, Virtual World, Ontology and Interfaces are the common components which are available in multi agent systems [1]. At first the Multi-Agent Engine starts and gives support at the run time of the system. The environment of the Multi agent system is provided by Virtual world. Agents in Virtual world compete and cooperate together. Knowledge of the each agent of conceived problem domain is available in Ontology. The interface provides support to communicate with the user and the system.

At present web technology is a very powerful information distributed media among people around the world, because anyone can access to websites from any place where the computer and internet connection are available. To access websites, the only requirement is web browser and no need for any other software to be installed. But the use of a standalone application is somewhat difficult, because it may need to install some software such as java runtime environment, .Net framework, etc… Therefore development of standalone applications using java, Visual basic or any other languages is reduced to the field of information distribution.

This paper presents a web based multi-agent system named WMAC which can be used to communicate with people in the Agriculture industry. WMAC system provides four types of agents as farmer; buyer, seller and instructor which represent farmers, buyers, sellers and technical instructors in the agricultural community. System provides the facility to communicate among persons in the agricultural industry when required.

The rest of the paper is organized as follows. Section 2 gives a brief introduction about multi-agent systems including existing multi-agent systems. Section 3 presents design of the WMAC system including a brief description of each module. Then section 4 gives how system works for the practical environment. Finally section 5 gives conclusion and further works for the project.

2. MULTI-AGENT TECHNOLOGY

Multi-Agent System Technology is a novel concept in software development field. This technology is very effective than conventional software development due to number of advantages. Multi-agent engine, Virtual world, Ontology and interfaces are the common components which are available in multi agent systems [1]. At first the Multi-agent engine starts and gives the support to run time of system. The environment of the Multi agent system is provided by Virtual world. Agents in Virtual world compete and cooperate together. Knowledge of the each agent of conceived problem domain is available in Ontology. The Interface provides support to communicate with the user and the system.

At present number of toolkits and frameworks are available to develop MAS software. Most of them are built up with Java programming language and few are built up with C/ C++ programming language.

Among them; JADE [10], Jason [11], AgentBuilder [12], ZEUS [13], NetLogo [14], Madkit [15], FIPA-OS [16], AgentSheets [17], Jack [18] and OpenCybele [19] are popular.

JADE (Java Agent DEvelopment Framework) is more than a development framework, which can be considered as a platform. In fact JADE can be distributed over several hosts rather than hosting on a single machine. Agents can also migrate or clone themselves to other hosts of the platform, regardless of the OS. JADE runs on Java virtual machine.

Jason is an Open Source fully java based Multi agent development framework developed through the improved version of agent oriented language AgentSpeak. Through the Jason, developers can easily programmed agent behavior of individual agents.

AgentBuilder is an integrated software development tool to develop general purpose multi-agent systems that allows software developers with no intelligent agent technologies to quickly and easily build intelligent agent-based applications. To develop multi-agent systems, researchers used standard framework or their own agent development frameworks. Further, there are number of agricultural systems which have been designed using Multi-agent System technology.

ZEUS is an Open Source agent development framework, which is also implemented in Java. This framework is also complying with FIPA specifications. ZEUS is famous as an agent development framework, which has the facility to plan and schedule the actions by agents. It has been specifically mentioned that ZEUS is ideal for the development of collaborative multi agent systems.

MaDKit is an open source modular and scalable multi-agent platform written in Java and built upon the AGR (Agent/Group/Role) organizational model. MaDKit agents play roles in groups and thus create artificial societies.

FIPA-OS (FIPA Open Source) is an open agent platform originating from Nortel Networks. The platform supports communication between multiple agents using an agent communication language which conforms to the FIPA (Foundation for Intelligent Physical Agents) agent standards. A key focus of the platform is that it supports openness. This is naturally supported by the agent paradigm itself and by the design of the platform itself whose parts have loose coupling such that extensions and innovations to support agent communication can occur in several key areas. FIPA-OS is being deployed in several domains including virtual private network provisioning, distributed meeting scheduling and a virtual home environment. It has been demonstrated to interoperate with other heterogeneous FIPA compliant platforms and is in use in numerous institutions around the world.

AgentSheets is an agent-based simulation tool that lets end-users create simulations within an easy to use GUI development environment. This implements the society through a combination of Java authoring tools, spreadsheets and agents. Methods, rules, conditions, and actions are all predefined structures within a GUI palette and they are drag-and dropped into agents by the user. The user is responsible for filling in the corresponding parameters. Tactile programming is the concept which is used by the AgentSheets. In this model agents are visually developed and must be graphically represented within their environment [31]. The emphasis is placed on visual development and the communication of the underlying rules, behaviors, and actions of agents during real-time simulation. The programming becomes tactile because of the heavy emphasis on GUI's for developing and manipulating agents.

Jack is a development environment that is built on top of java and acts as an extension of Java that offers classes for implementing agent behavior.

In OpenCybele, Agents are defined as "a group of event-driven activities that share data, thread, and execution concurrency structure". Here Activities are "active objects" internal to the agent and act on internal data in response to incoming events. The fact that other agents are incapable of manipulating the internal data of another reinforces the notion of agent autonomy.

3. DEVELOPED SYSTEMS VIA MAS TECHNOLOGY

The multi agent intervention to web technology is low compared to others. However there are number of systems but not for the web technology. Those systems have been developed using Java, C, LIPS and etc.,

iJADE Reporter has been developed by Eddie C.L. Chan and Raymond S.T. Lee [2]. iJADE Reporter is an intelligent context-aware news reporting system. In this, according to the multi-agent architecture news content are categorized by information retrieval algorithm. Frequency with Inverse Document Frequency (TFIDF) is used to separate the news content. This system provides simple, fast and efficient query in internet.

Catholijin M. Jonker, Remco A. Lam and Jan Treur proposed a Multi-agent Architecture for an intelligent website in insurance [3]. The agents in this architecture are based on generic broker agent model.

Przemyslaw Kazienko has developed a Multi-agent system for web advertising [4]. The main function of this system is to provide advertisements, which are most suitable for the given anonymous user navigating the web site.

Matsim.org has developed a framework to implement large-scale agent-based transport simulations which is called MATSim (Multi-Agent Transport Simulation) [5]. This is used as stand-alone framework. Demand-modeling, agent-based mobility-simulation (traffic flow simulation), re-planning, a controller to iteratively run simulations as well as methods to analyze the output generated by the modules are the things which offers by this framework.

Agricultural value chain management system (iAgri) has been developed by Kumarasiri and Karunananda to cut down the inefficiency and to control the balance in demand and supply chain of foods [6]. According to the iAgri, there are six stages in agricultural value chain such as Deciding, Seeding, Preparing and Planting, Growing, Harvesting and Selling.

A multi-agent system for agricultural stakeholders has also been developed by Adikari and Karunananda [7]. This system contains five agents such as Message Agent, Interface Agent, User Profile Agent, Crop Management Agent, and Selling Agent. This has been developed using JADE environment and can connect though internet.

Information System for Cultivation has been developed by Tharaka and kulawansa [8]. This was developed based on modern information technology infrastructures such as mobile and web services to disseminate domain knowledge to the farmer to enhance the cultivation process.
Jayarathna and Hettige have developed a java based distributed communication system (AgriCom) as stand-alone for the Agricultural domain [9]. The system contains four agent namely; Farmer agent, Buyer agent, Seller agent and agricultural instructor agent. That system helps to reduce the communication gap among the agricultural community. The WMAC is the improved web based development of the AgriCom.

4. DESIGN

WMAC is a web application which has been developed to reduce the communication gap among the people/stakeholders who are in the agricultural domain. To achieve that objective MAS technology has been used as a concept. The front-end of the system has been developed using HTML and back-end of the system has been developed using PHP and AJAX.

This web application provides four types of users and any person can get registered in any type as he required. The system uses common MYSQL database as ontology. Figure 1 shows the top level design of the web application.

WMAC database is used as ontology of the Multi agent system. This database consists of all the user information and messages. This database also uses as a common message space of the system.

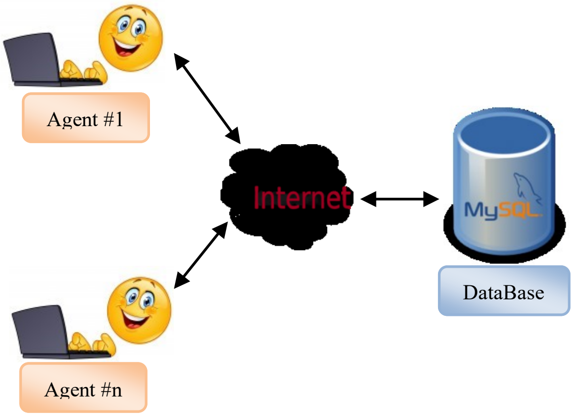


Figure 1: Top level design of the WMAC web application

Figure 2 shows the design architecture of the WMAC agent. WMAC agent consists of three components namely Web Interface, WMAC

Engine and the Web Server. Brief description of each module is given below.

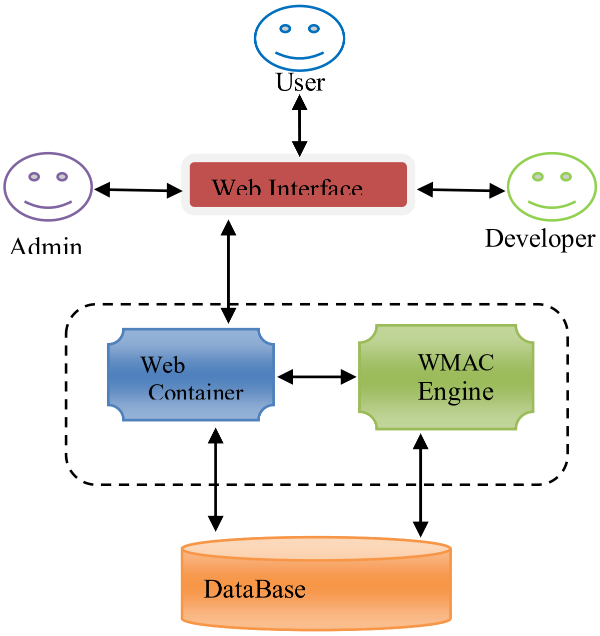
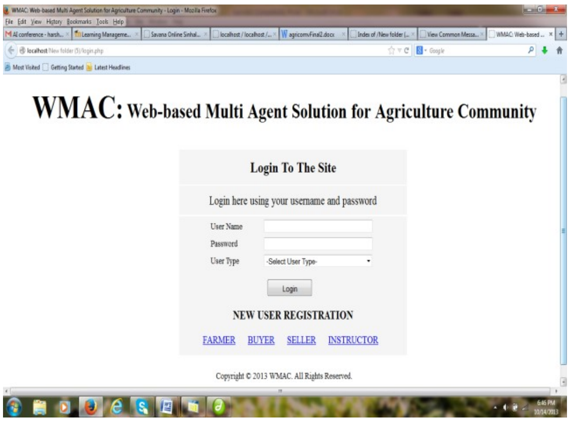


Figure 2: Design of the WMAC

A. *Web Interface*

Agents and the users communicate via web interface. According to the user registered types, features of the web interfaces are different. The figure 3 shows the initial WMAC web interface for all users/ visitors.


WMAC: Web-based Multi Agent Solution for Agriculture Community - Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/New folder (5)/login.php
Most Visited Getting Started Latest Headlines

WMAC: Web-based Multi Agent Solution for Agriculture Community

Login To The Site

Login here using your username and password

User Name
Password
User Type -Select User Type-

Login

NEW USER REGISTRATION

FARMER BUYER SELLER INSTRUCTOR

Copyright © 2013 WMAC. All Rights Reserved.

Figure 3: Initial WMAC Web interface

After creating a new account user can easily login to the system and can get all the facilities to use the system. At the login, the system automatically set the agent type by using ontology of user.

B. *WMAC Engine*

WMAC Engine is the important part of this system. This has been developed using the AJAX and is the controlling module of the WMAC which is capable of controlling of the all the features/ tasks that are in the WMAC. WMAC Engine communicates with the Web container and also Database. With the certain time intervals, the WMAC Engine checks the Database and updates the Web container. A user can give particular tasks to do while he logged in and also he can stop that any time. WMAC Engine is the backbone of the WMAC. This makes a communication between web server and the WMAC database.

C. *Web Container*

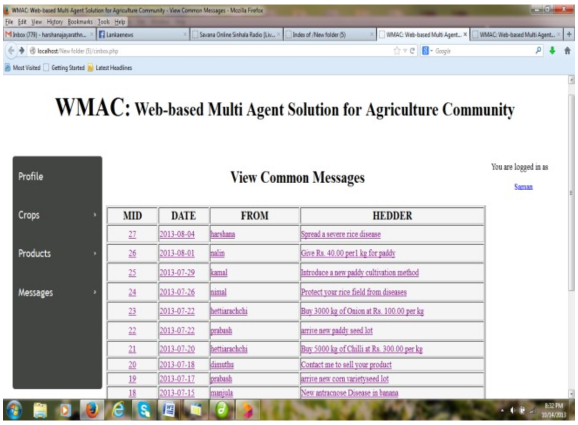
When the user requests a web page, the browser displays the content of the webpage on the web browser. Web container contains all the web pages. This provides the web pages to your browser. For WMAC, the Apache is used as the web server because of the most valuable features of Apache.

5. HOW SYSTEM WORKS

This shows how WMAC works in the practical environment. To use WMAC, users have to run any web browser and go to the WMAC website. WMAC provides all the facilities to members of WMAC. Therefore, at first a new comer has to be a member. For this, a new user has to get registered in the WMAC. Then new user becomes a member as any user type and he can get all the facilities from WMAC according to the system given.

There are two types of working modes WMAC system as manual and automated. If system works in the manual mode then following operation can be done. After registration of new user, he can add new records, send and view new messages and also delete viewed messages. While he is using automated mode, the WMAC system replies to received messages with predefined message structure. Those predefined message structures have been designed at the development stage. In order to provide automated mode, the system time to time check the received messages and read the tags assigned with those messages. Then the system generates new messages and sends them to relevant parties.

In the manual mode each user can set the tasks. After setting the task(s) and change the mode as automate, users can leave from the system. At this point, multi agent system is an active and does the relevant tasks. The figure 4 shows the common message interface for a particular person.


WMAC: Web-based Multi Agent Solution for Agriculture Community - View Common Messages - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/New folder (5)/cinbox.php
Profile
Crops
Products
Messages

WMAC: Web-based Multi Agent Solution for Agriculture Community

View Common Messages

You are logged in as
Saman

MID	DATE	FROM	HEDDER
27	2013-08-04	harshana	Spread a severe rice disease
26	2013-08-01	nalin	Give Rs. 40.00 per1 kg for paddy
25	2013-07-29	kamal	Introduce a new paddy cultivation method
24	2013-07-26	nimal	Protect your rice field from diseases
23	2013-07-22	hettiarachchi	Buy 3000 kg of Onion at Rs. 100.00 per kg
22	2013-07-22	prabash	arrive new paddy seed lot
21	2013-07-20	hettiarachchi	Buy 5000 kg of Chilli at Rs. 300.00 per kg
20	2013-07-18	dimuthu	Contact me to sell your product
19	2013-07-17	prabash	arrive new corn varietyseed lot
18	2013-07-15	manjula	New antracnose Disease in banana

Figure 4: WMAC Common Message Web Interface for Viewers

6. EVALUATION

The WMAC has been tested with number of standard testing method with the development. After development of WMAC, this has been evaluated with 22 persons. For this, those persons were allowed to register as their suitable user type with WMAC and to use WMAC for two month period. After two month period questionnaires were distributed among them and collected the results which were showed in table 1.

TABLE 1: Results obtained from questionnaires

Parameter	Very Efficient	Efficient	Satisfactory	Inefficient
Efficiency	3	12	6	1

Parameter	Very Easy	Easy	Satisfactory	Difficult	Very Difficult
User interface understandability	3	10	8	1	0
usability	3	13	6	0	0
Ability to order products	1	3	2	0	0
Ability to sell products	2	2	1	1	0
Ability to buy products	2	2	2	0	0
Ability to sell items	0	3	2		
Ability to instruct	0	3	1	1	0

Parameter	Excellent	Very Good	Good	Poor	Very Poor
Attractiveness	3	13	5	1	

Parameter	Very Useful	Useful	Satisfactory	Not Useful
Usefulness	4	9	7	2

According to the results, most like to use WMAC because this reduces the communication gap among the people who deal with the agriculture.

7. CONCLUSION AND FURTHER WORKS

This paper has illustrated the design architecture of the web based multi-agent system, WMAC. This is a web based multi agent system which has an ability to develop communication between persons in the agricultural sector as required. WMAC provides four types of agents as farmer, buyer, seller and instructor agent which represent actual farmers, buyers, sellers and technical instructors in the agricultural community.

WMAC agents work in two modes namely automated and manual. If a user stays with a computer, the agent works with the manual mode and provides facilities to users to communicate each other through the web interface. The system has been tested with 22 users and successful results were obtained. As further work of the project, it is very useful to develop WMAC as a mobile version as this web based system is very much useful for the agricultural domain to reduce the communication gap among the people.

REFERENCES

[1] George R., A new direction of research into Artificial Intelligence", Sri Lanka Association for Artificial Intelligence 5th Annual Sessions, 2008

[2] Eddile C.L Chan and Raymond S.T. Lee (2005), iJADE Reporter- An Intelligent Multi-agent Based Context-Aware News Reporting System, Springer-Verlag Berlin Heidelberg. [Online], Available: http://link.springer.com/chapter/10.1007%2F11552413_99#page-2

[3] Catholojin M. Jonker, Remco A. Lam and Jan Treur (1999). A Multi-agent Architecture for an Intelligent Website in Insurance. Springer-Verlag Berlin Heidelberg, [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-48414-0_5#page-2

[4] Przemyslaw Kazienko, Multi-agent System for Web Advertising, Wroclaw university of technology, Institute of applied informatics, Wybrzeze S. Wyspianskiego 27, 50-370 Wroclaw, Poland

[5] MATSim (2013, June 20), [Online]. Available: http://www.matsim.org/

[6] W.C.M. Kumarasiri and A.S. Karunananda, iAgri – An Agricultural Value Chain Management System, Proceedings of the ITRU Research Symposium, Moratuwa, 2010.

[7] A.A.P.S. Adikari and A.S. Karunananda, Multi Agent Systems for Agricultural stakeholders, Proceedings of the ITRU Research Symposium, Moratuwa, 2009.

[8] K.J. Tharaka, K.A.D.T. Kulawansa, Information System for Cultivation, Joint National Conference on Information Technology in Agriculture, 2011

[9] H.M.H.R.Jayarathna (2013), Online Agricultural Communication System (Dissertation submitted to the

Faculty of Information Technology, University of Moratuwa, Sri Lanka

[10] JADE. (2012, June 4). [Online]. Available: http://jade.tilab.com/

[11] Jason. (2012, May 27). [Online]. Available: http://jason.sourceforge.net/wp/

[12] AgentBuilder. (2012, July 18). [Online]. Available: URL: http://www.agentbuilder.com/Documentation/Lite/

[13] Zeus Agent System, LOWA STATE UNIVERSITY (2013, July 27). [Online]. Available: http://www.cs.iastate.edu/~baojie/acad/current/zeus/zeus.htm

[14] NetLogo (2013, July 23). [Online]. Available: http://ccl.northwestern.edu/netlogo/

[15] MADKIT (2013, July 29). [Online]. Available: http://www.madkit.org/

[16] School of Electronic Engineering and Computer Science, Queen Mary University of London (2013, July 23). [Online]. Available: http://www.eecs.qmul.ac.uk

[17] AgentSheets (2013, May 23). [Online]. Available: http://www.agentsheets.com/

[18] Agent Link (2013, August 2). [Online]. Available: http://eprints.agentlink.org/4671/

[19] OpenCybele (2013, August 6). [Online]. Available: http://eprints.agentlink.org/5422/

Multi Agent System for Artificial Neural Network Training

S. M. Dharmakeerthi[1], A. S. Karunananda[2,]

[1,2] Department of Computational Mathematics,University of Moratuwa, Sri Lanka.

[1]sanethdha@gmail.com, [2]asoka@itfac.mrt.ac.lk

Abstract - **Artificial neural networks are heavily used in the areas of pattern recognition, feature extraction, function approximation, scientific classification, control systems, noise reduction and prediction. Feed-forward and back-propagation neural networks are the most commonly used artificial neural networks. Many researchers face difficulties when selecting a proper ANN architecture and training parameters. The manual ANN training process is not the best practical solution because it is a time consuming task. Also most of the people conduct the manual process in an ad-hoc manner without having a proper basis for changing parameters. This research project has developed a multi-agent based approach to automate the optimization of neural network architecture and its training for feed-forward and back-propagation neural network. The ontology of the agent system comprises of commonly used heuristics for training of neural networks. Our experiments show that the more rational results can be obtained from the system with both simple datasets like XOR as well as with real life data sets. We can conclude that the neural network optimization and training tasks can be successfully accomplished by the agent based approach by analysing the results of the evaluation.**

Keywords- *Artificial Neural Network, Neural Network Training, Multi Agent Systems*

1. INTRODUCTION

Artificial neural network (ANN) is a very prominent artificial intelligent (AI) technique in the area of pattern recognition, function approximation, scientific classification and control systems [1]. Feed-forward and back-propagation neural network (FBPNN) is commonly used to perform the above activities [2] [3]. Selecting the appropriate neural network architecture is difficult to most of the people. The key problem is in deciding the number of hidden layers and the number of neurons that each hidden layer should contain. If the appropriate number of layers and the correct number of neurons in the hidden layers are not selected properly, the neural network will not be able to model the problem in a correct way and in turn results in poor output [4]. Since learning cycles on ANN take long time, performing repeated training and evaluating ANN while changing the hidden layer architecture manually would not be the best practical solution.

This project, MASAnnt (Multi Agent System for Artificial Neural Network Training) proposes a multi agent based approach to automate the neural network optimization and training tasks. In this proposed project, a group of expert agents works collaboratively to perform the neural network optimization and training. As an output, it will deliver well trained neural network with most rational neural network architecture without any human interaction. Most of the artificial neural network users are non-AI experts. Even a user who does not have much knowledge of artificial neural network can easily use this toolkit to perform their tasks. Hence those who do research in the areas like biology, zoology, chemistry, geology and medical science can be highly benefited by using this toolkit. Multi agent technology is a popular AI technology. Agents are fascinated by their simple characteristics and most of the time these agents do small jobs. They use only a very few resources and keep themselves idle after finishing the task. These agents can perform their tasks in a parallel and independent manner. The most significant feature of these agents is their ability to solve problems through communication, coordination and negotiation. Multi agent system is highly valuable when needed to model complex systems. So it is an ideal system of a model human brain which has a high density of complexity (high degree of interconnectivity and high degree of complexity) [7] [8].

The authors see the vital need of a system that can provide the most suitable and rational artificial neural network architecture for a given input and output data set or for a given problem.

2. CURRENT MOVEMENTS IN ARTIFICIAL NEURAL NETWORK TRAINING

An artificial neural network is a popular information processing mechanism which is inspired by the biological neural system of our brain which processes the information and makes inference. It works in parallel and can successfully solve nonlinear problems. ANN is made by a collection of highly interconnected procession units working collectively to solve a specific problem. Like a human brain, ANN also learns by example. Similar to the human brain, ANN also makes adjustments to the connection in between neurons [9].

Artificial neural networks are different from conventional programming because conventional programs use algorithmic approach, whereas the ANN uses a non-algorithmic approach. Mainly ANN is used in the areas of classification (pattern recognition, feature extraction), noise reduction (where removing of the noise elements is input data) and prediction (forecasting based on historical data).

The two types of ANN training mechanisms are supervised learning and unsupervised learning. Feed-forward and back-propagation is the most popular algorithm in supervised learning, where errors are propagated backwards through the hidden layers and the weights are adjusted accordingly. This is the most commonly used artificial neural network technique.

2.1 Manual Processes in Neural Network Training

In most of artificial neural network projects the training process is done manually. Trial and error method is used to determine the training parameters where several ANN models are developed by comparing the results with one and another [10]. Previous experiences and the intuition of the person are essential in selecting the correct structure. Therefore trial and error method is not a practical method as the ANN learning phase takes considerable time and needs human interaction at the end of each cycle and may cause many difficulties to the new researchers and to the ordinary people.

If the selection of the amount of hidden neurons are too large, the training time will be very high and may results in the neural network with over fit data [4] [6]. This may end up with local minima. If the hidden neuron amount is less than the required amount it cannot be converted to a good stage [5]. Learning rate of a neural network is to do adjustments to weights. If this is high, it might have quick converge or might diverge from solution. If too small, it will take longer time to converge to a final solution [11]. Momentum is helpful to skip from local minima. Incorrect momentum value some time slows down the ANN training, and in some cases can become unstable [12]. There are several activated functions available (threshold, piecewise linear and sigmoidal) so researchers have to select the appropriate function according to the nature of the data set. If E_{max} or error goal is too high we will get a half trained neural network. If it is too low, the ANN has a risk of never converging into a solution.

Accordingly, If the correct structure and parameters are not selected it will be impossible to get the required results from the artificial neural network and also affect the computational efficiency of the system.

2.2. Approaches to Neural Network Training

Amaresh and his team [13] of Anusandhan University, India has done a research and has proposed an experimental structure optimization technique for ANN using adaptive PSO (Particle Swarm Optimization), GA (Genetic Algorithm) and Performance Analysis Based on Boolean Identities like AND, OR and XOR. This includes the optimizing the architecture and the weights of ANN with minimum training error and with optimal number of hidden layers and nodes. They have run both the PSO and GA optimization until max error 0.001 is reached. In this research, the optimization is limited to the number of hidden layers and their neurons. Other training parameters like learning rate, learning momentum and activation functions were not the main focus of this project.

Miloš and Miroslav [10] have proposed an application of Taguchi Method for the optimization of the ANN model where the training is done by Levenbrg-Marquardt algorithm. In their case study, there were 4 input neurons and one output neuron to get the results. Though they have addressed more training parameters the training was done using a selected predefine parameter set. Always the number of hidden layer count is fixed to 2. So this approach did not provide a generalized solution.

XLMiner™ [14] is a data mining add-in for Excel that provides the data classification service using artificial neural network. In the neural network training it does the optimization task to get the suitable number of hidden layers and the number of neurons should contain in the hidden layer. In this case, not much intelligent work is happening, it keeps adding layers and neurons until it is satisfied with the results. This system forgets about the tuning of other main two artificial neural network parameters; learning rate and momentum.

Nayer [15] and a team from the University of Waterloo present an approach to get the optimal number of hidden nodes in a neural network. According to their research paper they have experimentally provided that best performance of the neural network occurs when the hidden node count is equal to $\log_2$ (T), where T is the number of training samples.

Apart from the above mentioned tools there are lots of artificial neural network tools available. Neuroph Studio, FANN (Fast Artificial Neural Network Library) and Neural Network Toolbox are few examples. The major disadvantage of these tools is that the user should have a better understanding about how the neural network function and need to conduct the tuning process manually.

3. MULTI AGENT BASED APPROACHES TO MODEL COMPLEX SYSTEMS

Multi agent systems have opened a new paradigm in computer world, mostly in the field of artificial intelligence. Agents are simple software programs that act autonomously in distributed environments. It has shown prominent achievements in problem solving in complex systems [18].

3.1 Complex Systems

Conventional software development technologies cannot provide good solutions in the domain of complex systems. Complex systems are equipped with high level of uncertainty, hence the behaviour of a complex system is difficult to predict. High degree of interactions among diverse entities (Agents) results in this uncertainty [19]. These diverse agents have rich interactions among them and are interdependent on each other. These agents are self-controlled based on defined rules and laws. They have their own autonomy and also have self-organization capabilities. Because of the high degree of nonlinear interactions they have come up with emergence solutions. It is not possible for a complex system to return to the previous system. Therefore it is said that complex systems are far from equilibrium (non-equilibrium).

3.2 Attempts to Solve Complex Problems using Multi Agents Technology

Many successful researches have been conducted to give solutions to the problems in complex systems. Multi agent technology has shown remarkably good results in problem solving in complex systems which is discussed below.

A research [20] has successfully provided a multi agent solution to vulnerability analysis of power grid in north China, by replaying existing reductionism mechanism. This power grid is a typical multi-level complex system. SCALA [7] toolkit (Cooperative System of Software Autonomous Agent) has been developed by Irene to design complex systems using multi agent technology which is executed on top of JACK. Another [21] successful multi agent system was developed by University of Michigan-Dearborn for complex vehicle fault diagnostics and health monitoring. Two set of agents works autonomously in this system. Signal agents do the monitoring and fault diagnosing the signals coming from sensors. Special case agents are used for detecting faults on specific components. Research [22] done by Nanjing University of Aeronautics and Astronautics shows that multi agent technology can provide a successful solution to complex mechatronics. This project has combined the agent concept with a physical control system and produce controller agents. By using these controller agents they devolved a multi controller system (multi agent system).

By analyzing research done by George Rzevski from The Open University, UK [19] proves that multi agent systems can model and solve problems in complex environments in many domains. One research provides a dynamic scheduling solution for rent-a-car business domain [23]. This was considered as the first industrial multi-agent system developed for dynamic scheduling. This was successfully implemented in the largest taxi service in London [24].

Like mentioned above they have carried out a lot of research and development to provide solutions to the problems in complex environments like real-time scheduling (taxis, air taxis, car rentals, seagoing tankers, trucks, space craft's), dynamic data mining, dynamic knowledge discovery and semantic search.

By deeply analyzing the research done by people in an around the world to provide solutions to problems in complex systems, it can be clearly stated that multi-agent technology have a solid basis for addressing the issues in complex environments.

4 MULTI AGENT TECHNOLOGY FOR NEURAL NETWORK TRAINING

At a glance anyone might wonder whether a solution can be given to the neural network training problem using an expert system. The simple answer is yes; we can develop a set of rules for all parameters and develop an expert system to perform this task. However, with the same set of rules different experts might execute those rules in a different manner. So in this case it is highly valuable to have a set of experts, to let them communicate and see each other's inferences. Definitely a much better solution can be brought across rather having one expert.

When analyzing closely neural network training also create a complexity. There is a high level of unpredictability of the final result of the neural network training. This uncertainty is the result of the higher interactions among these training parameters. These parameters have their own autonomy. They are governed by their own rules but they have a high level of interdependency. In the middle of the training, the neural network cannot reverse to the previous stage as it is irreversible. The neural network evolves with its environment and these parameters collectively provide an emerging solution in the training cycle. Because of this training an ANN creates a complex system. Hence, the multi agent technology can be considered as the most suitable approach to automate the neural network architecture optimization.

We postulate that multi agent technology can be used to automate the artificial neural network optimization and training task. Interaction among agents enables the emergence of quality training sessions which cannot be archived by an ad-hoc training sessions conducted by humans. It is straight forward to recognize training parameters such as number of hidden layers, number of neurons in each the hidden layer, momentum, learning rate, E_{max} (Error goal) and the activate function of an ANN as a set of agents. Inherent features of agents includes coordination, communication and negotiation and are able to mimic the ANN optimizing and training process by manipulating theses parameters.

5. DESIGN OF MASAnnt

Figure 1 shows a high level design of the developed system. It contains four main modules; System Control Agent, Training Unit, Expert Agent Unit and Ontology. Except the Ontology each module has at least one agent. System control agent work as the main request agent and the agents that are functioning in Expert agent unit are designed as resource agents. Agents inside the Training unit work as task agents. The ontology that is shown in the Figure 1 can be accessible by every agent in the system. All agents are designed to run on top of the JADE agent development framework.

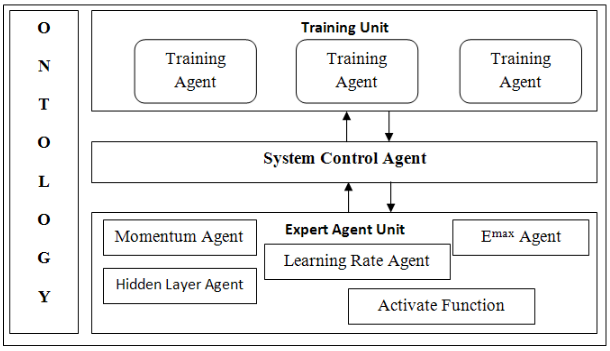


Figure 1: High Level Design of MASAnnt

5.1 System Control Agent

This is the main control unit of the system. Users will directly interact with this unit to do their training tasks. System control agents carry out the start and termination of the training sessions and request the parameter proposals from expert agents. This agent has several behaviours.

1) GUI behaviour

The main task of this behaviour is to enable the visibility of the activities of the system to the end users. The GUI contains necessary components to control the system. There are several components in the GUI to show the agent communication and the message space. Also it shows the latest updates on the status of each and every agent in the system. Separately it shows how the training process is functioning and how the error rate behaves. All past training attempts and their results are also visible in the GUI. All these information will be in single frame so the end user does not need to do any task to see this information. The system control agent will initiate the GUI and then it makes sure that each and every agent can access it. To accomplish that, system control agent will share a reference of the GUI in the common ontology. Because of that other agents can easily access the GUI and do the necessary updates. But key updates of the training process will be done by the system control agent itself.

2) Negotiation behaviour

This is the key behaviour in the system. All the intelligent results are highly depended on the functionality of this behaviour. System control agent conducts and manages the negotiation process. This behaviour includes sending the training data set to expert agents, monitoring the negotiation, controling if needed and sending the final results to the training unit to conduct the training session. If the negotiation goes beyond the predefined threshold time the system control agent takes necessary actions to finish it quickly and start the training task with available data. All the threshold values are stored in the common ontology so users can do the changes without any effect.

3) Message space behaviour

The system control agent will create a common profile and keep it under its control. This will ensure that each and every agent write messages and read them when needed. This is very much useful in agent negotiation and indicated the status of agents to others. So the expert agents can update their proposing values and suggestion in the negotiation phase. Because of this message space, it reduces the communication overhead at the negotiation phase.

Victual Message Space in GUI	
Messages through JADE	Messages through Common Profile

Figure 2: Message Space

As shown in the Figure 2 there will be a virtual message space in the GUI that contains all the messages that are communicated through the common profile as well as directly using JADE. Because at the end, users need not be aware of how the agent communication is conducted. For the users, the most important thing is what the agents are communicating.

5.2 Training Unit

This unit has a set of training agents. These agents will execute training tasks with the training parameter sets provided by the System Control Agent. More than one training agent can be executed in parallel if sufficient hardware resources are available. Currently this is only designed for one training agent. But later this can be designed to work with remote training agents then we can execute more parallel training sessions. Training agent should send training information to the system control agent to display in the GUI.

5.3 Expert Agent Unit

There are set of expert agents in this unit where each is an expert on one training parameter such as one is an expert on hidden layers, one may be with momentum, one for activation function, one for E_{max} and one for the learning rate. With the communication among these agents, they will negotiate about the ANN training parameters that need to be setup.

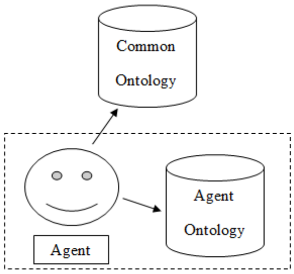


Figure 3: Expert Agent and Ontology

As shown in Figure 3 each expert agent has its own local ontology. Accordingly, agent code only contains how to do the negotiation and the communication. All the agent specific knowledge is located inside the local ontology. So user can add or remove any knowledge without modifying the agent code. This will simplify the agent code. When a new data set is received from system control agent these expert agents quickly move to active mode and suggests the initial parameters which are needed to start the first training session. Depending on the results of the previous training session they again start another negotiation session to decide the next set of parameters. These agents will continue this cyclic process until they get terminated by a system control agent.

5.4 Ontology

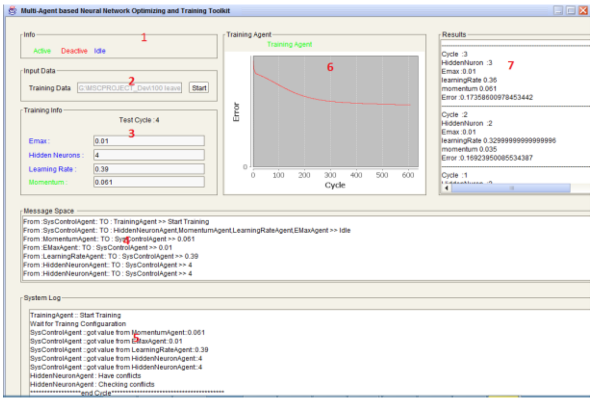
All the common knowledge required for agents to work is stored in here. This contains names of agents, the syntax needed for communication, negotiation rules and threshold values. Information about all previous training sessions are also saved in here. This is accessible for all the agents and this common ontology does not have any contradictory facts with any expert agent local ontology. Also this does not contain any information for deciding the neural network parameters. As mentioned in the above section expert agent specific information will be stored in their own ontologies.

6. IMPLEMENTATION OF MASAnnt

JADE was used as the multi agent framework. JADE was selected because it provides fast and easy framework for agent application development. The Java based development environment was used to implant the prototype on top of JADE. NetBeans was used as the development framework.

6.1 Implementation of User Interface

All the UI components are embedded into one single window as shown in Figure 4, so the users need not navigate through many windows. This enhances the usability of this toolkit. Component 1 shows the colour codes used to show the agent status. Component 2 is the only input field in this application which is used to specify the training data set. In the 3rdcomponent, it shows the current proposed values for the training parameter and also shows the status of each expert agent by using the colour codes explained in component 1. The component 4 and 5 illustrates the message space and the system log respectively. Component 6 gives a real-time visual understanding on how the error is reduced. Component 7 keeps information on previous training sessions.



Multi-Agent based Neural Network Optimizing and Training Toolkit

Info: Active Deactive Idle

Input Data: Training Data: C:\MSCPROJECT_Dev\100 leave [Start]

Training Info
Test Cycle :4
Emax : 0.01
Hidden Neurons : 4
Learning Rate : 0.39
Momentum : 0.061

Training Agent: Training Agent

Results
```
Cycle :3
HiddenNuron :3
Emax :0.01
learningRate 0.36
momentum 0.061
Error :0.17358600978453442
Cycle :2
HiddenNuron :2
Emax :0.01
learningRate 0.32999999999999996
momentum 0.035
Error :0.16923950085534387
Cycle :1
```

Message Space
```
From :SysControlAgent:: TO : TrainingAgent >> Start Training
From :SysControlAgent:: TO : HiddenNeuronAgent,MomentumAgent,LearningRateAgent,EMaxAgent >> Idle
From :MomentumAgent:: TO : SysControlAgent >> 0.061
From :EMaxAgent:: TO : SysControlAgent >> 0.01
From :LearningRateAgent:: TO : SysControlAgent >> 0.39
From :HiddenNeuronAgent:: TO : SysControlAgent >> 4
From :HiddenNeuronAgent:: TO : SysControlAgent >> 4
```

System Log
```
TrainingAgent :: Start Training
Wait for Trainng Configuaration
SysControlAgent ::got value from MomentumAgent::0.061
SysControlAgent ::got value from EMaxAgent::0.01
SysControlAgent ::got value from LearningRateAgent::0.39
SysControlAgent ::got value from HiddenNeuronAgent::4
SysControlAgent ::got value from HiddenNeuronAgent::4
HiddenNeuronAgent : Have conflicts
HiddenNeuronAgent : Checking conflicts
******************end Cycle*******************************************
```

Figure 4 User Interface of MASAnnt

Component 1 shows the colour codes used to show the agent status. Component 2 is the only input field in this application which is used to specify the training data set. In the 3rd component, it shows the current proposed values for the training parameter and also shows the status of each expert agent by using the colour codes explained in component 1. The component 4 and 5 illustrates the message space and the system log respectively. Component 6 gives a real-time visual understanding on how the error is reduced. Component 7 keeps information on previous training sessions.

6.2 Implementation of System Control Agent Behaviours

First the control agent will process the data set and make them into a system understandable format. It will assign a unique key to each type of data set. After all data processing is over system control agent will then activate all the expert agents. Then system control agent will send a request message to all expert agents to get their proposals on the training set parameters. Control agent will search in the DFService to receive the information about all expert agents available in the system. Then it adds those expert agents as receivers and sends the message. Then expert agent will be in the loop for a predefine time. Within this time period expert agent can do the negotiation among them and propose the values.

6.3 Training Agent

When the training agent received the training parameter set it will initialize the neural network that is configured to that agent and start the training session. At the end it will provide results to the control agent. By configuring, this agent can add any external neural network framework. Since this implementation separate the parameter deciding and network training. Either of this can be changed any time.

6.4 Expert Agent

Every expert agent has two Java classes. One is for agent functionality and other is for the local ontology.

The source codes of all the agents are identical except for the reference code of their ontologies. Because of that all agents function in the same way. The difference of their decision making is powered by their ontologies. Each agent conducts its local ontology to conduct the negotiation.

6.5 Ontology

Currently the ontology was developed as a Java class and it contains all the syntax needed for the communication to share information among other agent to store the history of the training sessions. Communication ontology contains all needed syntaxes to the communication. Common message will be wrapped in this class to ensure maintainability. Common ontology contains all history of training sessions kept in a result list. Agent can easily find historical data by navigating through the list available in the result list object. Training parameter ontology is used for temporary keeping the values of training parameters. Expert agent updates this ontology to share their proposals in the negotiation stage.

7. EVALUATION OF MASAnnt

One of the major requirements of this prototype was the usability. As mentioned in earlier chapters it can be used by any person who has little knowledge about artificial neural networks. According to the requirements, the system only requires the training data set as the input. User interface only has one input location and the insertion of text file contains the training data set. So the user does not have to select any artificial neural network parameter for the training. Therefore, the input requirement is successfully completed in this prototype.

Adhering to the requirement this prototype displays all information about the training session conducted and also shows the live view of the training session. The output requirements were also successfully completed in this prototype. Later in this paper the results and the behaviour of the developed prototype will be discussed in depth.

7.1 Evaluation with XOR

XOR gate was used to conduct the initial evaluation. Since the neural network training takes longer time for converge, authors have used this XOR data pattern, so the results can be seen in a less time duration.

Table 1: XOR training data

Inputs	0.0,0.0	0.0,1.0	1.0,0.0	1.0,1.0
Outputs	0.0	1.0	1.0	0.0

Table 2: XOR training results

cycle	Hidden Neurons	Emax	Learning Rate	momentum	Error
1	2	0.01	0.3	0.035	0.2573
4	6	0.01	0.3600	0.0402	0.0205
6	6	0.01	0.4000	0.0428	0.0104
8	8	0.01	0.4400	0.0453	0.0099

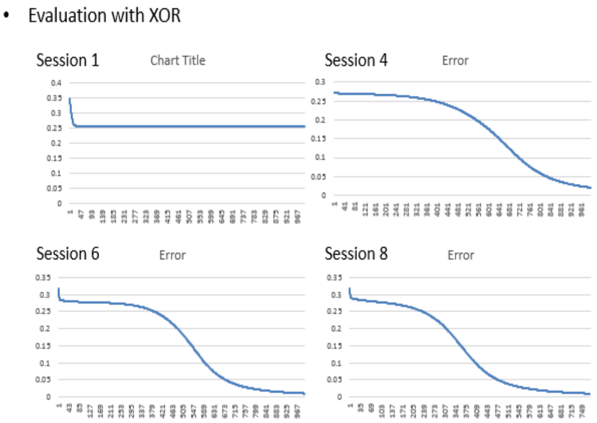


Figure 5: Error reduction in XOR

By analysing the results in Table 2 and Figure 5 it can be easily concluded that the system was able to provide a more rational solution. It is able to reduce the error rate to reach the E_{max} value. At the training session 8 the system was able to converge to the target. This test was a good evidence that the developed prototype is functioning correctly. The test reached the error goal without any problem and since the training data set was small, functionality of it could be easily monitored.

8. EVALUATION WITH IRIS DATA SET

This evaluation was conducted by using the available UC Irvine Machine Learning Repository (The University of California, Irvine). This is a well-known data set in the area of pattern recognition literature.

8.1 50, 000 training cycles

50,000 training cycles were used in each training session, if the targeted E_{max} value could not be reach by the system; it starts a new session with a new set of parameters.

Table 3: XOR training results-50, 000 training cycles

cycle	Hidden Neuron s	Emax	Learnin g Rate	momentum	Error
1	4	0.01	0.3	0.035	0.3020
4	8	0.01	0.39	0.087	0.1979
8	16	0.01	0.630	0.19099	0.1170
16	32	0.113	0.7500	0.2429	0.107

If the system cannot reduce the error rate to the given E_{max} value within a pre-define number of training sessions, system will automatically increase the E_{max}. Since this Iris data set has lot of values it takes longer time to train. So threshold number of session was set to 10 in this evaluation.

8.2 100,000 training cycles

Another training session was conducted using 100,000 cycles in each session.

Table 4: XOR training results-100, 000 training cycles

cycle	Hidden Neurons	Emax	Learning Rate	momentum	Error
1	4	0.01	0.3	0.035	0.3020
4	10	0.01	0.39	0.087	0.1979
12	32	0.0225	0.6300	0.19099	0.1170
16	46	0.1139	0.7500	0.24299	0.1073

It is clear that by increasing the number of training cycles in a session have a positive impact on the results. By analysing the results in Table 3 and Table 4, it can be concluded that the developed prototype is capable of doing the neural network optimization task. Starting with a higher error rate, it is able to reduce the error rate to the targeted value.

9. CONCLUSION

Starting from the initial training parameter values, expert agents changes their parameters to reduce the error rate by negotiating with each other. The evaluation shows that the MASAnnt was able to reduce the error rate to the given target E_{max} value. The toolkit only requires the training of the data set to be performed. So as expected initially, this MASAnnt can be used by any person; the user need not be an artificial neural network expert. Because of that researchers from any domain can use this tool to conduct their research easily.

Defining the expert agent's rules was the major problem faced in this research. It was hard to find the rules that could define the values of the training parameters. After a lot of reading and experiments a simple rule set was implemented in the expert agent ontology. Most of the rules behaved in a heuristic manner to select the training parameters. These rules simulated the training session conducted by an artificial neural network expert.

Since the agents were developed in a manner that the agent knowledge could be easily edited, the agent ontology can be continuously improved by doing more research. All improvements that are adding to the agent ontology are highly influential to the quality of the MASAnnt. Also new expert agents can be introduced to the system to enhance the results. Since the artificial neural network has longer training time, this toolkit should be deployed in a high-end server, web service or in a similar technology for the users to get the service without having to bear any additional computational cost.

REFERENCES

[1] L. N. Long and A. Gupta, "Scalable Massively Parallel Artificial Neural Networks," in *Aerospace Conference- American Institute of Aeronautics and Astronautics*, 2005.

[2] J. Heaton, "A Feedforward Neural Network," Heaton Research, Inc., 2012. [Online]. Available: http://www.heatonresearch.com/node/704. [Accessed 10 September 2012].

[3] P. McCollum, "An Introduction to Back-Propagation Neural Networks," Seattle Robotics Society, [Online]. Available: http://www.seattlerobotics.org/encoder/nov98/neural.html. [Accessed 2 September 2012].

[4] DTREG : Software For Predictive Modeling and Forecasting, "Multilayer Perceptron Neural Networks," DTREG : Software For Predictive Modeling and Forecasting, [Online]. Available: http://www.dtreg.com/mlfn.htm. [Accessed 5 September 2012].

[5] S.-i. Kazuhiro and K. Maizuru, "A Two Phase Method for Determining the Number of Neurons in the Hidden Layer of a 3-Layer Neural Network," in *SICE Annual Conference 2010*, 2010.

[6] A. Kretinnin, Y. Bulygin and S. Valyuhov, "Intelligent Algorithm for Forecasting of Optimum Neurons Quantity in Perceptron with One Hidden Layer," in *International Joint Conference on Neural Networks (IJCNN 2008)*, 2008.

[7] I. Degirmenciyan-Cartault, "A Multi-Agent Approach for Complex System Design," 2002. [Online]. Available: http://ftp.rta.nato.int/public//PubFullText/RTO/EN/RTO-EN-022///EN-022-05.pdf. [Accessed 2012 August 27].

[8] P. Stone, "Multiagent Systems," 24 September 1997. [Online]. Available: http://www.cs.cmu.edu/afs/cs/usr/pstone/public/papers/97MAS-survey/node2.html. [Accessed 26 August 2012].

[9] C. Stergiou and D. Siganos, "NEURAL NETWORKS," Imperial College London, [Online]. Available: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html. [Accessed 7 November 2012].

[10 J. M. Miloš and R. R. Miroslav, "Optimal Selection of ANN Training and Architectural Parameters Using Taguchi Method: A Case Study," *FME Transactions,* vol. 39, no. 2, pp. 79-86, 2011.

[11 R. Berteig, "Basic Concepts for Neural Networks," Cheshire Engineering Corporation, 2003. [Online]. Available: http://www.cheshireeng.com/Neuralyst/nnbg.htm. [Accessed 1 Octomber 2012].

[12 University of Wisconsin, "A Neural Network Approach For Interpolating Species Density Patterns From Remotely Sensed & GIS data: An Example Using The Desert Tortoise," University of Wisconsin, [Online]. Available: http://pages.cs.wisc.edu/~bolo/shipyard/neural/tort.html. [Accessed 1 Octomber 2012].

[13 S. Amaresh, K. P. Sushanta and P. Sabyasachi, "Optimization of ANN Structure Using Adaptive PSO & GA and Performance Analysis Based on Boolean Identities," *International Journal of Computer & communication Technology,* vol. 2, no. VIII, pp. 70-77, 2011.

[14 XLMiner, "XLMiner Data Mining Add-in For Excel," Frontline Systems, Inc, 2012. [Online]. Available: http://www.solver.com/xlminer-data-mining. [Accessed 5 November 2012].

[15 N. Wanas, G. Auda, M. S. Kamal and F. Karray, "ON THE OPTIMAL NUMBER OF HIDDEN NODES IN A NEURAL NETWORK," in *Canadian Conference on Electrical and Computer Engineering*, Waterloo, 1998.

[16 Advanced Agent-Robotics Technology Lab, "Multi-Agent Systems," Carnegie Mellon University, [Online]. Available: http://www.cs.cmu.edu/~softagents/multi.html. [Accessed 5 2 2013].

[17 G. Rzevski, "Modelling Large Complex Systems Using Multi-Agent Technology," in *ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2M12.

[18 X. Dong, G. Xiong, J. Hou, F. Dong and T. R. Nyberg, "Vulnerability analysis of power grid based on multi-agent complex systems," in *IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI)*, 2011.

[19 Y. Lu Murphey and Z. Chen, "A Multi-Agent System for Complex Vehicle Fault Diagnostics and Health Monitoring," in *15th IEEE International Conference on Engineering of Complex Computer Systems*, Oxford, 2010.

[20 X. Wu, P. Lou and D. Tang, "A Multi-agent Controller on Embedded System for Complex Mechatronics," in *Chinese Control and Decision Conference (CCDC 2009)*, Guilin, 2009.

[21 G. Rzevski, S. Andreev, P. Shveykin, P. Skobelev and I. Yankov, "Multi-Agent Scheduler for Rent-a-car companies," in *Forth International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, Linz, 2009.

[22 G. Rzevsk, A. Glaschenko, A. Ivaschenko and P. Skobelev, "Multi-Agent Real-Time Scheduling System for Taxi Companies," in *8th Int. Conf. on Autonomous Agents and Multiagent Systems*, Budapest, 2009.

[23 G. Rzevski, P. Skobelev and V. Andreev, "MagentaToolkit: A Set of Multi-agent Tools for Developing Adaptive Real-Time Application," in *Third International Conference on Industrial Applications of Holonic and Multi-Agent Sys*, Regensburg, 2007.

[24 D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning respresentations by back-propergation errors," *NATURE,* vol. 323, pp. 533-536, 1986.

[25 B. Krose and P. v. d. Smagt, An Introduction to Neural Networks, University of Amsterdam, 1996.

Conflict Reduction Analysis of Bulk Agent Approach in Multi Agent Systems

P. M. Gunathilaka[1], A. S. Karunananda[2]
Department of Computational Mathematics, University of Moratuwa, Moratuwa, Sri Lanka
prageeth@codegen.co.uk[1], asoka@itfac.mrt.ac.lk[2]

Abstract— **Our universe can be considered as the largest multi agent system with no visible conflicts. Particles in different dimensions interacts, based on different gravitational rules, which defines the universal extra dimensions called 'Bulk'. The same concept can be modeled, as the Bulk Agent Approach in multi agent systems to overcome potential conflicts, which also empowers the direction of the emergent success of the overall system. On the other hand, it is a design challenge in multi agent systems, how to avoid unnecessary conflicting chaos, which could consume large computational resources and valuable time. Lack of resources or social knowledge could lead to either resource conflicts or knowledge conflicts. As a solution, Argumentation Based Negotiation (ABN) with the support of conflict evading and re-planning has been presented in the literature as one of the best approach in conflict resolution techniques. However, conflict evading and re-planning would not be useful in an environment where resources are not abundant. Therefore, we present our novel approach as a solution for the burning limitations of conflict evading and re-planning. Philosophical explanations and Brane Cosmology, which explains on how gravity governs on brane particles, based on the concept of universal extra dimensions, are the main inspirations for our research. Any multi agent environment can be considered as a multi-dimensional universe, where the universal norms originate in a higher dimension. These universal norms provide the guidelines for emergent success of the whole system. However, universal norms can change dynamically based on the social and environmental changes in the lower dimensions. Therefore, in our architecture we define higher dimensions by an agent type called Bulk Agents whereas agents in the lower dimensions are called Brane Agents. The Bulk Agent monitors behaviors of the Brane Agents and provides the direction or the guideline for the success of the overall system. These directions were shared among brane agents as Volatile Ontology so that the overall agent society is well capable of avoiding potential conflicts which otherwise would increase the failure rate of the system. Our analysis are experimented based on an application called Multi Agent Marketplace. Our experiments were analyzed based on statistical figures which has shown that the conflicts can be avoided or resolved with minimal computational time and resources by introducing Bulk Agents, which represents the extra dimensions in multi agent systems. This paper presents the results of our analysis on identifying the level of effectiveness of the Bulk Agent approach in conflict resolution in Multi Agent Systems. Keywords-multi agent systems; bulk agent; conflict resolution; brane agent; cosmology**

Keywords-multi agent systems; bulk agent; conflict resolution; brane agent; cosmology

1. INTRODUCTION

Autonomous agents must communicate to overcome hurdles of conflicts on knowledge and resources in achieving goals. Conflicts should be avoided whenever possible, so that the limited computational resources can be better utilize for the intended role of the agent, which converges the whole society to the emergent success as early as possible. Our novel and alternative approach has been conceptualized based on the inspiration on cosmological and philosophical studies on how natural systems manifest their existence. Our hypothesis is that conflicts in MAS can be resolved with minimal amount of time and effort, by an upper level agent called bulk agent which is supported by the knowledge and control of the dimensions higher than other operating agents, called brane agents.

Bulk Agent approach is a model with a novel multi agent environment structure, which improves the potential of conflict evading, and it also supports agents to resolve conflicts, consuming minimal amount of time and resources.

There are eight sections in this paper including the introduction. Second section explains current trends and practices such as ABN approach in conflict resolution of Multi Agent Systems. The Bulk Agent approach is backed by philosophical and cosmological finding. Therefore we present the third section to elaborate cosmological background and the fourth section contains various philosophical inspirations. Fifth section presents our approach as an extension to our previous research paper, which contains, in detail architectural model on Bulk Agent Approach. In this paper, we elaborate our experiment with statistical analysis with the calculated success rate of our approach in a Multi Agent Marketplace. As the sixth section we present the implementation details of our research with a conflict monitoring tool. Conflicts on our multi agent application are evaluated based on observational as well as statistical data collected in collection experiments. Results of our evaluation were discussed in the Seventh section. As the eighth and the last section of this paper we conclude with a summary of our findings with future research plans.

2. REVIEW ON CURRENT TRENDS IN CONFLICT RESOLUTION

There are two types of conflicts in literature which are categorized based on the cause of each of them. Conflicts can be occurred due to limited resource which have to be shared among agents, i.e. agents has to compete with each other to win and get the control of resources, as they are crucial to achieve their goals. Therefore, conflicts due to resources are called Resource Conflicts [9]. Conflicts can also be observed in multi agent environments, when the agent's knowledge or the ontology has gaps in between, i.e. when agents are communicating based on different perceived knowledge, resulting Knowledge Conflicts [9].

To overcome such conflicting situations Argumentation, Negotiation and Coordination play a major role in multi agent systems. On the other hand, deciding on "How" and "When" to communicate, is a more important aspect in multi agent conflict resolution studies [7, 2, 4, 9, 8, 5]. Among various conflict resolution techniques explained in literature, Argumentation-based Negotiation (ABN) has shown promising results when the conflicts occurs as a result of resource limitations [9].

A Argumentation Based Negotiation

At the early stages of literature on multi agent conflict resolution techniques, Heuristic-based and Game-based approaches were introduced [13]. However with the increase of complexity, information and resource demand in a multi agent systems, negotiation becomes an essential feature to overcome conflicting situations. Hence, ABN became a fast immerging technology for conflict resolution [2, 4]. Based on a ring structured multi agent system, it has been shown that the ABN approach has more potential in handling the flexibility and the demand on negotiation[13]. Moreover, there are three types arguments in ABN approach.(i). Reward (A, B, P, Q) denotes that if negotiator B realizes P, negotiator A will give it rewards Q. (ii). Threat (A, B, P, Q) denotes that if negotiator B does not Realize P, negotiator A will give it threat Q. (iii). Appeal (A, B, P, Q, R) denotes that negotiator A desires Negotiator B realizes P, but not Q for reason R. Based on these argument types, agents can exchange proposals with the intension of resolving conflicts. Moreover, meta-information should also be attached with each proposal to justify and convince the proposal [9]. Each agent would bring proposals in favor of itself. However the conflicts can be resolved when agents agree on one proposal with the help of inbuilt self-compromising ability.

Each argument in resolving a conflict has to consume some level of computational time and resources. Taking necessary means to minimize such cost, is much more important than the arguments them-self. Therefore, it is necessary to analyze and avoid (finding an alternative means) conflicts by possible ways. For the same purpose, Conflict Evading (finding an alternative means) and Re-planning (modifying the intended cause of actions) has been introduced in the literature as an alternative to ABN approach [9]. Moreover, It has been identified that the selective argumentation is effective than the indiscriminate argumentation [9]. However, when there is a resource conflict, it is necessary to have abundant resources to utilize the evading or re-planning approaches, whereas, ABN approach shows more effective means in resolving conflicts when the resources are constrained [9]. So the best approach suggested is “argue only if the evading is not possible”. But, these approaches have been experimentally proven only in agent environments with no social structure, so that all agents communicate peer-to-peer [9]. So, Analysis on social relationships and structures among agents is yet to be studied further, in the context of conflict resolution.

Arguments as well as re-planning could sometimes continue indefinitely without a promise of a resolution. As Kraus, Sycara and Evenchik have proposed [5], when the arguments and negotiations consumes long durations, it should be stopped and start re-planning. But, after consuming large computational resources, it could be a huge loss for the multi agent system to abandon or disregard all the arguments and negotiations. Hence, the question is "Is it worth to argue?". In a situation where solution is time critical arguments could makes the system fail. As a solution, some level of tactics needs to be defined such as Last Minute Tactic in the system. But these tactics are based on some level of assumptions, as the tactic can be used only if the environments supports in various ways such as availability of information and deadline for agents [8]. Hence, in a situation where assumptions are invalid tactics may not be useful enough.

B Shared Global Ontology vs Knowledge Conflicts

How the multi agent knowledge should be modeled in a system is mainly a design problem. But knowledge on the agent society has to be a dynamic ontology, which should continuously changing based on the social changes. On the other hand, due to lack of knowledge, motivation and/or less or no capacity to work with social influences create conflicts in multi agents societies [11].When the environment and system complexity increases, social influences from various factors could also be increased on agents. False understanding on those influences could trigger conflicts, and that could increase the risk of system failures. Based on ABN approach such conflicts can be resolved, only if it has following four features available in the multi agent environment. (i) a schema to reason in social settings; (ii) a mechanism to identify a suitable set of arguments; (iii) a language and a protocol to exchange these arguments; and (iv) a decision making functionality to generate such dialogues [11]. According to the third and fourth requirements, it is clear that global shared ontology is required to define the social schema and the state of the immerged society. Modeling social influences and a resistance to failures, based on a global shared ontology would reduce argument passing and it also helps the fast recovery from conflicts. However, shared global ontology would not solve all the problems. It's still required to further design and implement a strong agent structure which supports and smoothen the flow of social and environmental knowledge, without falling into the misconception of anarchic system [7].

3. COSMOLOGICAL INSPIRATION

We can consider the whole universe as a great multi agent system, which contains no visible conflicts. Each planet, star or galaxy including the planet earth and its humans, share the same universal phenomena with no observable conflicts. It is worth to study how the complexity of such a vast system manages in the context of complexity science and its technologies, such as Multi Agent Technology.

According to the Brane Cosmology, gravity defines the direction to the success in the universal evolution, starting from the origin of time. As explained in String theory and its extension M-theory, particles are confined on a hyper-surface (called Brane) embedded in a higher dimension (called Bulk) [10]. We are living in a world of 4 dimensions, where first 3 dimensions define the space which floats on the 4th dimension called time. To analyze the universe, mathematical concept called Brane defined in the theoretical physics. A particle in a world of P dimension is called a P-brane which are compacted to its dimension, while restricting to its higher dimensions [2, 6, 10]. For us, 5th or higher dimensional phenomena such as gravity belongs to the bulk of our Brane. In the same way gravity in our dimension is much weaker than the gravity on lower dimensions. Due to this difference, we perceive the space and time continuum in our universe, much differently than the lower dimension. That makes our sun to keep its nuclear reactions continue for millions of years, providing enough time for us to evolve from a single cell to humans, before it explodes. In other words gravity in the universal extra dimension defines the success of the overall universe. How we map this concept of brane cosmology for our conflict resolution strategy is explained in the fifth section of this paper.

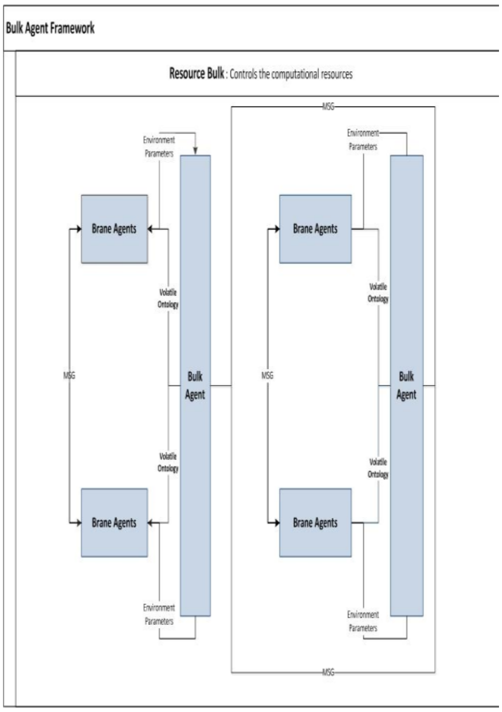
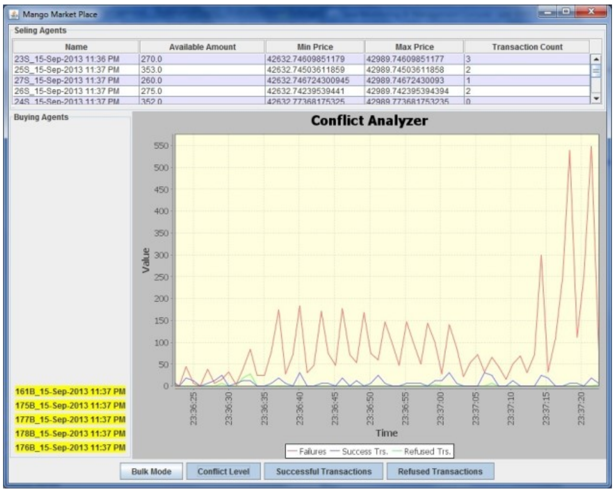


Figure 1: Bulk Agent Framework


Mango Market Place

Seling Agents

Name	Available Amount	Min Price	Max Price	Transaction Count
23S_15-Sep-2013 11:36 PM	270.0	42632.74609851179	42989.74609851177	3
25S_15-Sep-2013 11:37 PM	353.0	42632.74503611859	42989.74503611858	2
27S_15-Sep-2013 11:37 PM	260.0	42632.746724300945	42989.74672430093	1
26S_15-Sep-2013 11:37 PM	275.0	42632.74239539441	42989.742395394394	2
24S_15-Sep-2013 11:37 PM	352.0	42632.77368175325	42989.773681753235	0

Buying Agents

161B_15-Sep-2013 11:37 PM
175B_15-Sep-2013 11:37 PM
177B_15-Sep-2013 11:37 PM
178B_15-Sep-2013 11:37 PM
176B_15-Sep-2013 11:37 PM

Figure.2: Multi Agent Marketplace Monitoring Tool.

Similarly, we conceptually model an agent structure with their responsibilities that could be defined based on the dimensions that agents operate.

4. PHILOSOPHICAL INSPIRATION

Laws governing particulars could never account for the organization principles of a higher entity, as explained by Michael Polanyi [3]. He further pointed out that "success of an immerged system cannot be explained from some of its parts". In the same way, "We cannot see parts of a system until we see a failure" as explained by Heidegger's concept of Breakdown. Such a philosophical definition of a success of a system can be best experimented based on our approach on bulk agents which defines the direction of the success of overall system.

"*The significant problems we face cannot be solved at the same level of thinking we were at when we created them*", as explained by Albert Einstein.

It means that, it is necessary to observe from a higher context so that the bigger picture of the problem can be perceived which resembles the solution.

On the other hand, Buddhist philosophy defines the concept of *Vipassana* meditation, which can be used to train our mind to perceive the suffering as a observation from the perspective of a third party [15, 16], just to come out from the context of suffering and to understand the big picture or the truth as defined in four noble truths. All these philosophical concepts motivates us to find a solution, based on a higher dimensional agents (Bulk Agents) which perceive the environments and its resources from a higher perspective, so that it can define the direction of the success of the overall system.

5. BULK AGENT APPROACH

If we consider a multi agent system as a universe with multi dimensions, universal agents can have lower dimensions as well as higher or extra dimensions. Results of higher dimensional changes could influence on lower dimensions, whereas each lower dimensional changes have at least a minimal influence on the higher dimension. Such concept can be implemented by defining an agent called Bulk Agent to represent the universal extra dimensions.

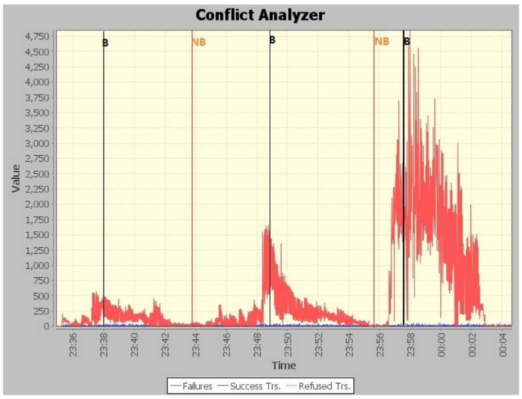


Figure 4: Conflict analyzer graph

In most multi agent systems, social influences can be modeled based on a bulk agent, which helps, shows or direct the success of overall system. It should be noted that, according to the multi agent problem in hand, it is necessary to model the higher dimension in the multi agent systems which controls or maintain the rules, universal to the lower agents. To represents the global knowledge, as well as the knowledge on how the social influence based on globally shared ontology can be generated, motivate us to define two ontological concepts as explains below.

(i). Volatile Ontology (Ontology of the Bulk Agent immerged from the overall brane agents).

(ii). Concrete Ontology (Ontology of the Bulk Agent which has no impact from the brane agents).[17]

Volatile Ontology will be generated by the bulk agent and it will be shared among brane agents. Volatile Ontology may contain the social knowledge based on the so called big picture of the society. To generate such ontology and show the direction of the success to other agents, it is necessary to have an ontology inbuilt in to the Bulk Agent. That ontology may not be changed based on the brane agent's actions or influences. Therefore, such ontology is called concrete ontology. It is important to note that the volatile

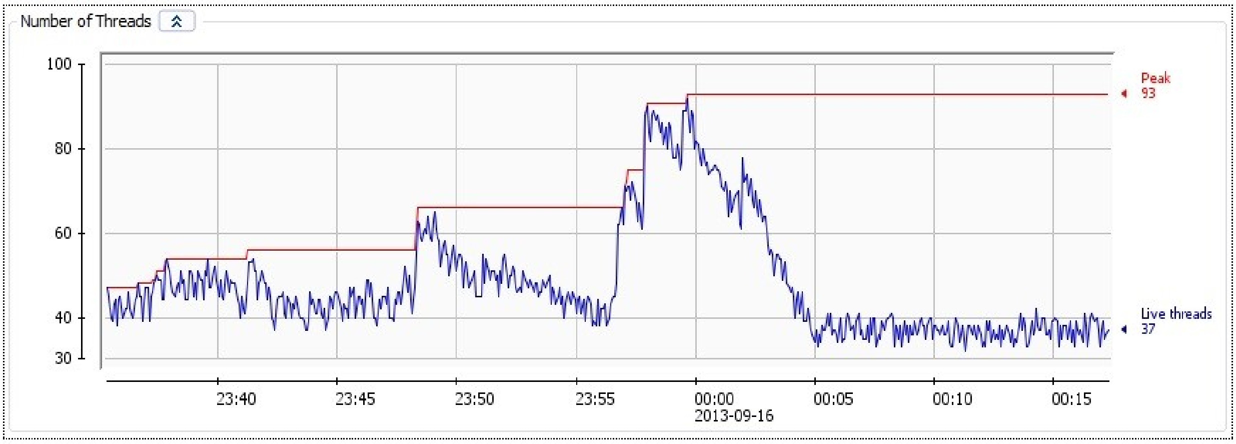


Figure 3. Thread usages in the multi agent marketplace

ontology should not make bulk agents a ruler of the society, or the dictators, as it would abuse the very meaning and the ability of multi agent concept to handle complex requirements and uncertainty. This model helps us to define an ontology which shows more power of resistance in conflicting situations than in non-conflicting situations, so that the conflicts can be resolved fast and easy.

In other words, agents can argue not only based on a localized knowledge of agent's operations, but also the global influences and direction of the success, which again is volatile to the emergent effect. So the main concept of this research is to define agents as brane and bulks and define their ontology in such a way that, the complexity of the overall system can be reduced drastically as the conflicts can be better avoided or resolved at the higher dimensions. As shown in the Bulk Agent Architecture [17] in Figure 1, multiple bulk agents can be formulated and allowed to communicate while keeping their own ontology unique for its brane environments. This approach has been successfully implemented in a Multi Agent Marketplace where supplier agents and buyer agents compete each other for a better price [17].

6. IMPLEMENTATION

Our multi agent marketplace is implemented in the *JADE* multi agent framework. As shown in the Figure 2, buyer agents in a given moment of time is listed in yellow color in the left side panel whereas the sellers with their selling price range is shown in the grid at the top. The conflict analyzer graph shows the rate of successful (or useful) and failed (or irrelevant) transactions (or communication) in blue and red colors respectively.

As shown in the Figure 2, we have implemented the conflict monitoring tool to monitor successful transactions (blue colored graph) and amount and level of conflicting situations (red colored graph). The conflict analyzer graph is used to obtain the statistic figures to calculate the overall success of the system. In our multi agent marketplace, to maintain the competitive advantage, while avoiding conflicts, knowledge on the overall system is essential. Such knowledge gap can be better bridged by the volatile shared ontology of the bulk agent when needed.

We have evaluated the effectiveness of the bulk agent approach based on the number of successful transactions and the amount of communication. In order to analyze the level of conflicting situations, statistical figures were counted based on failed communication or the communication effort that has taken to avoid arguments, and the cost over benefit of agents. Moreover consumption of computational resources by the overall system with and without the Bulk Agent approach has been separately evaluated.

Our application monitoring tool can be used to enable or disable the Bulk Agent in the agent environment so that we can compare conflicting levels as well as the computational usages of the overall system. We have used the JConsole to monitor the computational resource consumption. There are three overlapping charts in the Conflict Analyzer graph to indicate the conflicting levels, successful transactions and refused transactions. Refused transactions were calculated, based on how many times a transaction has been abundant in-between due to the demand fluctuations and the uncertainty. In other words, by the time a buyer is ready to pay for a pre-defined price, seller could sell the same item to another buyer, so that the first buyer's transaction is refused. Analysis on refused transaction count is necessary to measure the amount of uncertainty in the market.

7. EXPERIMENT AND EVALUATION

In our experiment, initially we let the system to run for some time till it reaches o the equilibrium state. Then we introduce the bulk agent to the market and then observe the pattern of conflict reduction. When it reaches to the minimum conflicting level, we removed the effect of the bulk agent. In each such iteration rapid increase in the conflicting level has been observed. We did this experiment multiple times and collected statistic figures to identify the effectiveness of the Bulk Agent Approach in the Multi Agent Marketplace that we have implemented.

Experimental results in the conflict analyzer graph is shown in the Figure 4. Each black line indicate the time that we have enabled the Bulk Agent mode whereas the long vertical red line indicates the time that we have disabled the operations of the Bulk Agent. Accordingly, it indicates higher amount of conflicts when the effect of the bulk agent is not present. But the system goes down to the minimal level of conflicts, whenever the bulk agent is introduced to the market. It can be seen that the average conflicting level decreases down to 50, regardless of how large the initial conflicting level. In other words more than 90% of conflicts can be avoided based on the Bulk Agent Approach in our Multi Agent Marketplace.

In JADE, message passing is facilitated using asynchronous threads. Therefore, the amount of live threads indicates the level of communication overhead. Each time the bulk agent is introduced to the market, it has been observed drastic reduction of unnecessary communications. As shown in the Figure 3, it can also be seen that amount of parallel threads that always stabilizes at the average of 40 when the bulk agent is in action.

8. CONCLUSION

This is our second paper based on the Bulk Agent approach in conflict resolution [17]. We have presented some statistical analysis based on our experimental result which has shown a drastic conflict reduction by 90% within the average of 5 to 6 seconds. Moreover the reduction of communication overhead makes the system lightweight while maintaining a better agent society with the introduction of Bulk Agents. Moreover these statistical analysis shows that, our novel approach would solve most of the design and implementation challenges of multi agent application as it consumes minimal amount of threads with the promise of lower communication overhead.

However, it should be noted that the right selection of knowledge in the volatile ontology is the main secrete behind the success of the overall system. Hence it is important to encourage further studies on structures and best practices in defining volatile and concrete ontology.

This paper is presented as a solution to avoid costly conflicting operations in multi agent systems and we don't clam this as another multi agent problem solving algorithm. However, there is a potential of improving this approach as a problem solving algorithm as the next phase of our research.

REFERENCES

[20] J. Azilawati, C. Mei Lin Ho and Y. San Chee, "Argument-based negotiationandconflictresolutionthroughenactiverolepl ayinSecond Life," Proceedings of the15thInternationalConference onComputers in Education(pp. 561–568).Amsterdam:IOS Press.

[21] M.Georgeand J.R. Minkel–ARecycledUniverse,2012, PP 1 – 20.

[22] H.Prosch, Michael Polanyi: ACritical Expositio,1986,SUNY Press.

[23] L.Hsairi,K.Ghedira,M.A.AlimiandA.Benabdelhafid,"R esolution of conflicts via argument based netotiation: extended enterprise case," Service Systemsand Service Management, 2006International Conference on,vol.1,no., pp.828-833, 25-27Oct. 2006

[24] S.Kraus,K.SycaraandA.Evenchik,"Reaching agreementsthrough argumentation," ArtificialIntelligence,104(1998)1–69

[25] M.Gogberashvili,Ourworldasanexpandingshell,Institut eofPhysics, Tamarashvili,st.Europhys,2008, LettVolume 49,Number 3.

[26] M. Wooldridge and N. R. Jennings, "Pitfalls of Agent-Oriented Development," Proc. Second Int'l Conf. Autonomous Agents,ACMPress,NewYork, May1998, pp. 385–391.

[27] H. Nabila, D. Yannis and M. Pavlos, "Tactics and concessions for argumentation-based negotiation,"FourthInternationalConferenceon ComputationalModels of Argument, 2012,Vienne:Autriche.

[28] N. C. Karunatillake and N. R. Jennings,"Is it worth arguing?," ArgMAS, 2004, pp. 234-250

[29] B.PhilippeandC.vandeBruck,"Cosmologyand brane worlds: a review,"CenterforMathematical Sciences,Class.Quant.Grav,2003, pp. 201-232.

[30] N. C. Karunatillake, N. R. Jennings, I. J. Rahwan andT. Norman,"Arguing and negotiating in the presence of social influences," 4thInternationalCentralandEasternEuropeanConferenc e on Multi-Agent Systems(CEEMAS'05),Budapest, Hungary,2005, pp. 223-235.

[31] N.C. Karunatillake, N. R. Jennings, I. Rahwan and S. D. Ramchurn, "Managing social influences through argumentation-based negotiation," ThirdInternationalWorkshoponArgumentationinMulti-AgentSystems (ArgMAS 2006),Hakodate,Japan, 2006, pp. 35-52

[32] K.Huang,D.A.Cartes,S.K.Srivastava,"Amulti agentbasedalgorithm forring-structuredshipboardpower systemreconfiguration,"Systems, ManandCybernetics, IEEEInternationalConferenceon,vol.1,2005, pp.530-535.

[33] B.McInnes,OfficialNUS,National University ofSingapore,Webpage, http://www.math.nus.edu.sg

[34] C.S.Anderson,PainanditsEndings:Thefournoble truthsinTheravada Buddhistcanon, 1999,vol. 1.

[35] Nyanaponika Thera, The heart of Buddhist meditation, vol. 1, 1992.

[57] P.M.Gunathilaka, A.S.Karunananda, "Exploiting Bulk Agent Approach for Conflict Resolution in Multi Agent Systems", Asia Modelling Symposium 2013 (AMS2013), Seventh Asia International Conference on Mathematical Modelling and Computer Simulation, Kuala Lumpur, Malaysia, 2013, In press.

Bone Crack Detector based on X-Ray using Fuzzy Logic and Neural Network

K.A.W.P. Abesinghe, S.D.H.S.Wickramarathna
Uva Wellassa University, Badulla, Sri Lanka

harshani@uwu.ac.lk

***Abstract* - Large number of X-Ray images are analyzed by doctors in hospitals daily to identify various diseases and anomalies in human body. One particular area is identification of problems in bones in that sense, based on X-Ray images doctors are going to predict the problem in that bone such as bone crack, damage etc. These kinds of manual inspection of X-Ray images consume a lot of time and the process itself is monotonous made during the inspection.**

As a solution to these problems here we introduce Computer-assisted decision-making system to detect cracks in bones which are visible in X-Ray image using Fuzzy Logic and Neural Networks.

1. INTRODUCTION

An x-ray (radiograph) is a medical test that helps doctors inspect and treat medical conditions with bone damage. Bones contain calcium element and calcium has high atomic number. Therefore calcium absorbs X-Rays efficiently than other elements in the bone structure. Small amount of X-Rays reaches the shadows of the bones and make them clearly visible on the radiograph [3].

X-Ray images are used by doctors to detect the Fractures of broken bones, bone tumors (abnormal growth of the bone cells may be cancer), degenerative bone conditions, Osteomyelitis, Osteomalacia and etc. [1]. Daily, doctors are analyzing thousands of X-Ray images at hospitals, which is a monotonous and a time consuming activity.

X-Ray images are input to the computer via a scanner with transparency adaptor or use a X-Ray illuminator and take the photos of that X-Ray [2].

The main objective of this study is to develop a Computer-assisted decision-making system to detect cracks in bones in X-Ray images. Hence the developed recognition system consists of three components, namely: Image Processing and feature extraction component, Fuzzy Logic based identification component and Neural Network based verification component.

2. METHODOLOGY

The system is developed to detect simple bone structures as well as complex bone structures. Femur bone structure is used to represent simple bone structures and single figure structure is used to represent complex bone structures. Below given figure 1 shows the basic system architecture. C#.net, Emgu CV image processing library and Aforge.net framework are the technologies used to develop the system.

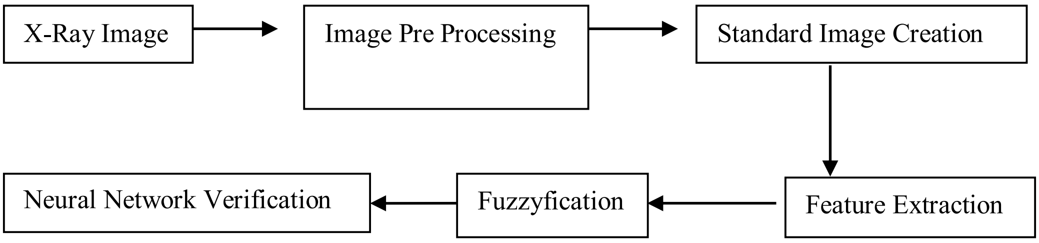


Figure 1: Basic system architecture

2.1 Image Processing Module

Image processing module consists of several sub modules namely; Image Preprocessing, background elimination, crack detection, rough line and smooth line separation and actual crack detection modules. Image Preprocessing is used to enhance the features of the image. Edge based filters are applied to enhance the edges because, edges are an important factor for detecting cracks of in bones in X-Rays. Then system detects the edges of the image using canny edge detector.

After detecting the edges, it eliminates the background of the image. System finally detects the edges that could be cracked or not. The system is capable of detecting actual cracks and also some specific features of the bone such as bends. Those specific features of the bone are smooth lines and cracks are rough lines. Using that specific characteristics system separate crack lines from some other features of the bone. Finally abstract the feature information of crack line for Fuzzyfication.

2.2 Fuzzy Classifier Module

Fuzzy Classifier contains fuzzy inference engine and input output variables. Input variables are information about the crack lines finally detected by image processing module. Output variables are detected as cracks. System is using two fuzzy sets and three fuzzy functions for each fuzzy set. One fuzzy set is Fuzzy Multiplication and other is Fuzzy Variation. Fuzzy functions for each Fuzzy variable are used for following classifications.

- High chance to be a crack
- Mid chance to be a crack
- Low chance to be a crack

Fuzzy Multiplication is calculated by number of times the crack lines change their direction, multiplied by number of rough lines of the crack, detected by the image processing module. Fuzzy Variation is creating a line from start point of crack to end point of crack and taking the variation of the crack line from the already created center line.

1. **RULE 1**: If Fuzzy Multiplication is high and Fuzzy Variation is high then there is a high chance of a crack on the bone
2. **RULE 2**: If Fuzzy Multiplication is mid and Fuzzy Variation is mid then medium chance to be a crack on the bone
3. **RULE 3**: If Fuzzy Multiplication is low and Fuzzy Variation is low, no crack on the bone

The fuzzy rules calculate the output and those outputs send to Neural Network for verification.

2.3 Neural Network Module

Neural Network takes some input from the Fuzzyfication and some parameters are taken from image processing module.

Fuzzyfication parameters are three functions for two of Fuzzy variable returns six outputs and two extra parameters taken from the image processing module. These parameters are

- Bone Length Ratio= actual bone length/ crack length of bone;
- Bone Width Ratio= actual bone width/ crack width of bone;

Eight neurons for input layers, ten neurons for hidden layer and three neurons for output layer were used to train the Neural Network under supervised learning. Output is obtained as a pattern. Finally Neural Network verifies the fuzzy output and correctly says whether the bone is cracked or not.

3. RESULTS

System is able to detect the crack in the long bone X-ray image as well as in single figure. Bones are not in same size and not in same shapes, but the fuzzy module detects the crack in long bone as well as in single figure. Neural Network further clarify that bone contain cracks or not.

Long bone and single figure are also not in same shape and not in same sizes. Single figure contains complex structure rather than long bone.

Some characteristics of bone also sometimes display as cracks of the bone. But fuzzy classifier is eliminating those characteristics. Neural Network verifies that X-Ray contain cracks or not. Output of the system displays in the user interface and marked the crack of the bone.

3. Evaluations

Evaluation of the project can be done in two ways at the beginning. We can take the X-Ray image into system via a scanner with a transparency adaptor or with using X-Ray illuminator machine and camera. Those images first provides to the image processing module for preprocessing, background elimination, crack detection, separate the crack line from smooth lines and finally the actual crack detection. The output of the image processing module, forwards to the input of fuzzy logic detection module. And the output of the Fuzzyfication forwards to the input of the neural network for further verification and neural networks detect the crack of the X-Ray images from non crack X-Ray images.

4. CONCLUSION

System can detects the crack of the bone of X-Ray images in femur bone and also in finger as simple bone structure and complex bone structure. Also system is able to mark intensive places that could be a crack. Fuzzy logic detects the crack and Neural Network verifies the crack correctly.

Further Work:
This system is implemented to detect the cracks in the long bone through X ray image. In addition to this feature the system provides some features that can apply further.

- Bones are different in shape of the bones and can expand this system to detect the crack in different types of bones.
- Detect the crack in the complex bone structure (eg: Hand X Ray image).
- Detect the hole in the teeth and guess amount of plaster wanted to fill that hole.

ACKNOWLEDGMENT

The first author would like to acknowledge the help and guidance given by the employees of the Badulla General Hospital. Their help given in finding X-Rays and other related work contributed largely in making this project a success.

REFERENCES

[1] Hum Yan Chai, Lai Khin Wee, Tan TianSwee, SheikhHussain, “Gray-Level Co-occurrence Matrix Bone Fracture Detection”, Department of Biomedical Instrumentation and Signal Processing of University Technology, Malaysia.

[2]S.K.Mahendran,S.SanthoshBaboo, “An Enhanced Tibia Fracture Detection Tool Using Image Processing and Classification Fusion Techniques in X-Ray Images”,Sankara College of Science and Commerce, Coimbatote, Tamil Nadu, India.

[3] Tian Tai Pengn, “Detection of Femur fractures in X Ray images”, National University of Singapore, Singapore.

Organizing Committee

- Prof. Nalin Wickramarachchi, University of Moratuwa
- Dr. A.P. Madurapperuma, The Open University of Sri Lanka
- Dr. L.S.K. Udugama, The Open University of Sri Lanka
- Dr. (Ms) H.U.W. Ratnayake, The Open University of Sri Lanka
- Prof. A. S. Karunananda, University of Moratuwa
- Dr. (Ms) D.D.M. Ranasinghe, The Open University of Sri Lanka
- Ms. Chandrika Fernando, Sri Lanka Institute of Information Technology
- Mr. Budditha Hettige, University of Sri Jayewardenepura
- Ms. P.T.R. Dabare, The Open University of Sri Lanka
- Ms. Dilini T. Kulawansa, University of Moratuwa

Panel of Reviewers

- Prof. Nalin Wickramarachchi, University of Moratuwa
- Dr. L.S.K. Udugama, The Open University of Sri Lanka
- Dr. (Ms) D.D.M. Ranasinghe, The Open University of Sri Lanka
- Dr. (Ms) H.U.W. Ratnayake, The Open University of Sri Lanka

Technical Session Chairs

- Prof. Nalin Wickramarachchi, University of Moratuwa
- Dr. L.S.K. Udugama, The Open University of Sri Lanka

Panel of Judges– Best paper Award

- Dr. N. Ranasinghe, University of Colombo
- Dr. L. Ranatunga, University of Moratuwa
- Dr.(Ms) Subha Fernando, University of Moratuwa