Prediction of the Attention Area in Ambient Intelligence Tasks

Jawad Shafi, Plamen Angelov and Muhammad Umair

Abstract With recent advances in Ambient Intelligence (AmI), it is becoming possible to provide support to a human in an AmI environment. This paper presents an Adaptive Neuro-Fuzzy Inference System (ANFIS) model based scheme, named as prediction of the attention area using ANFIS (PAA_ANFIS), which predicts the human attention area on visual display with ordinary web camera. The PAA_ANFIS model was designed using trial and error based on various experiments in simulated gaming environment. This study was conducted to illustrate that ANFIS is effective with hybrid learning, for the prediction of eye-gaze area in the environment. PAA_ANFIS results show that ANFIS has been successfully implemented for predicting within different learning context scenarios in a simulated environment. The performance of the PAA_ANFIS model was evaluated using standard error measurements techniques. The Matlab® simulation results indicate that the performance of the ANFIS approach is valuable, accurate and easy to implement. The PAA_ANFIS results are based on analysis of different model settings in our environment. To further validate the PAA_ANFIS, forecasting results are then compared with linear regression. The comparative results show the superiority and higher accuracy achieved by applying the ANFIS, which is equipped with the capability of generating linear relationship and the fuzzy inference system in input-output data. However, it should be noted that an increase in the number of membership functions (MF) will increase the system response time.

J. Shafi (✉) · P. Angelov
Data Science Group, School of Computer and Communication,
Lancaster Univeristy, Lancaster, UK
e-mail: j.shafi@lancaster.ac.uk; jawadshafi@ciitlahore.edu.pk

P. Angelov
e-mail: p.angelov@lancaster.ac.uk

P. Angelov
Chair of Excellence, Carlos III University, Madrid, Spain

J. Shafi · M. Umair
Department of Computer Sciences, COMSATS Institute of Information Technology,
Lahore, Pakistan
e-mail: mumair@ciitlahore.edu.pk

V. Sgurev et al. (eds.), *Innovative Issues in Intelligent Systems*,
Studies in Computational Intelligence 623, DOI 10.1007/978-3-319-27267-2_2

1 Introduction

The Ambient Intelligence (AmI) is aiming to provide pervasive support to humans. The visual attention is one of the primary tools to figure out the user attention area on a display [10, 17, 34]. It is a key challenge to figure out the appropriate representation and awareness of a human's attention states, some of research work relevant to human attention are discussed in [1, 2, 9, 30]. "Mindreading" may address different types of human states [14] e.g. intention, attention, emotions. More dedicated services can be provided in AmI environment, if the user attention area can be determined. Some research work have been already done but that is purely based on using special type of tracking devices [11, 26]. The uses of such devices in an application make it relatively expensive. This paper adopts non-linear regression prediction technique, using web camera coordinates with a combination of hand controlling coordinates. This makes it cost effective, reusable, robust and with better accuracy to figure out visual attention area on visual display.

To provide personalized services in an AmI environment, the system has to be able to make inferences based on the interaction with the user. To achieve this we apply computational intelligence (CI). CI are commonly used for learner modelling because of the complexity of relationships, which are difficult to represent. One research [36] listed the main issues related to the use of machine learning techniques for modelling purposes as: the need for large data sets, the need for labelled data and computational complexity.

This paper present methods for identification of the visual attention area on a display. The main objective is to adopt a suitable learning method for eye-gaze identification. We used Adaptive Neuro-Fuzzy Inference System (ANFIS) based on the acquired data from the environment in an application. Section 2 provides a brief review of the work which is based on ANFIS. Section 3 presents the structure of the proposed ANFIS based method. Section 4 presents the results and discussion of the proposed system. Finally, Sect. 5 presents concluding comments about future developments related to this work.

2 Related Work

There are several studies in the literature which deal with the detection of visual attention. The results of [33] are based on a neural network for classification of posture and a Hidden Markov Model for recognizing the state of interest. Another approach in [16] is based on the creation of computing and communication systems that can detect and reason about the human attention by fusing the information received from multiple sources. Another study based on neuro-fuzzy approach is reported in [7] to infer the attention level of a user in front of a monitor using a simple camera but this is based on combining eye gaze and head pose information, in a non-intrusive environment. These models use two inputs together with other biometrics to infer the user's attention.

There are other works in literature around this issue but they are all based on head pose estimation and also use more than one camera or extra equipment [25, 28]. In [11, 26] the authors suggested the gaze detection of visual attention but, as mentioned above, that is based on the use of an eye tracking device.

The Adaptive-Neuro Fuzzy Inference System is a hybrid system that combines the potential benefits of both, the artificial neural network (ANN) [15] and fuzzy logic (FL) [23] methods. ANFIS is being employed in numerous modelling and forecasting problems [21] and in different domains, e.g. chemical and biological engineering [6], renewable energy [27], energy economics [29], stock market [12], cancellation of EMG signals [35], time series prediction [37], speech recognition and signal processing [13] and industrial management [5]. To the best of our knowledge no body have used ANFIS to predict the user attention area in AmI so far.

3 Traditional Architecture

One of the widely used hybrid intelligent systems is the neuro-fuzzy combination. These systems take advantage of human-like ambiguity, interpretability, transparency, and ability to model non-linear vagueness on data in an environment using fuzzy logic [23]; also it has a flexible structure and superior capability of a neural network [15]. Modern neuro-fuzzy systems are multilayer feed-forward neural networks. In these systems the fuzzy rules are trained by the learning algorithm implemented and applied on a neural network [8]. Fuzzy logic does not incorporate any learning mechanism, while neural networks appeared as a black box approach; do not have explicit knowledge representation. A typical neuro-fuzzy system such as the Adaptive neuro-fuzzy inference system (ANFIS) introduced by Jang in 1993 [18] integrates both neural networks and fuzzy logic principles; it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate non-linear functions [3]. Hence, ANFIS is considered to be a universal estimator [21]. There are few characteristics that enable ANFIS to achieve a success [19]:

- It makes the complex system behaviour more refined using fuzzy IF-THEN rules.
- It is easy to implement.
- It enables fast and accurate learning.
- It is easy to incorporate both linguistic and numeric knowledge for problem solving.

3.1 Takagi Sugeno Fuzzy Inference System (TS-FIS)

TS models are a powerful practical engineering tool for modeling and control of complex systems [4]. The output of a TS model can be linear or a constant [22]. As ANFIS is based on Takagi–Sugeno fuzzy inference system [18], so the rules are of the following type:

$$\Re_j : IF(x_m is \aleph_{j1}) AND \ldots AND(x_m is \aleph_{jp}) THEN(x_g = a_{j0} + a_{j1}x_1 + \ldots + a_{jn}x_n);$$
$$j = \{1, R\}$$

where $\Re_j$ denotes the jth fuzzy rule; R is the number of fuzzy rules; x_m is the input variable, $\aleph_{jp}$ denotes the antecedent fuzzy sets, $p = \{1, n\}$; x_g is the output of the jth linear subsystem; a_{jl} are its parameters, $l = \{0, n\}$. In ANFIS, the output linear membership functions of the consequent part of TS-FIS are automatically adjusted. TS-FIS consists of inputs, output(s), set of predefined rules and a defuzzification method. Aggregation is employed to unify the outputs of all the rules resulting into a single fuzzy set. Hence, the final output of the system is the weighted average of all the rule partial outputs.

3.2 Artificial Neural Network (ANN)

ANN is a structure of interconnected neurons arranged in a systematic manner to perform some computing task. It is widely used due to its capability of learning using the training data. Its architecture usually consists of input hidden and output layers. Each layer is composed of several processing neurons. Input layer includes the inputs, data values obtained from training. Hidden layer processes the inputs into the hidden layer. The training algorithm which is widely used in ANN is called error back-propagation algorithm [24].

3.3 Back-Propagation Algorithm

Back-propagation is a machine learning feed forward, multilayer network parameter optimization based algorithm for supervised mode of learning [24]. The number of input, hidden, output layers and neurons in each layer depend on the application requirements. The main objective of back-propagation learning algorithm is to adjust the values of weights in the training data set in such a manner so as to get the same value as the correct output value of the network using the validation data set. This process is shown in Fig. 1.

In the forward pass, input weights are injected to subsequent layer. The activation function is implemented to generate the weights for the next layer.

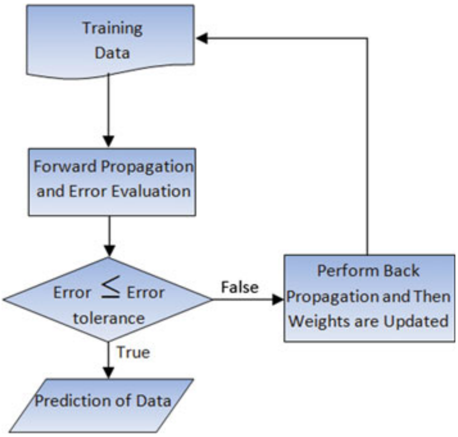

Fig. 1 Back propagation in ANFIS

Finally, the output layer is ready to generate some output value. The generated and original values of the output are utilized to derive the error which is propagated further back to the input layer. The process will continue until the error is less than a pre-defined error tolerance and the network is ready to be used.

3.4 ANFIS Architecture

It maps the first order Takagi-Sugeno Fuzzy Inference System in multilayer feed-forward adaptive neural network to enhance performance with the above mentioned attractive features. The first order Takagi-Sugeno fuzzy model's inference mechanism and defuzzification process is shown in Fig. 2, [18]. It has five-layer architecture.

Fuzzy If-Then rules are used to describe in ANFIS [18] a system for example by two rules:

R_1 = IF x is C_1 and y is D_1, then $f1 = p_1x + q_1y + r_1$
R_2 = IF x is C_2 and y is D_2, then $f2 = p_2x + q_2y + r_2$

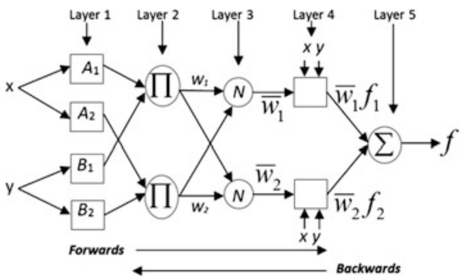

Fig. 2 ANFIS 5 layer architecture, with forwards and backwards propagation

where x and y are the inputs, C_i and D_i are the fuzzy sets f_i are the outputs within the fuzzy region specified by the fuzzy rule, and p_i, q_i, and r_i are the design parameters of the output/consequent part that are determined during the training process. Each layer has a specific functionality.

Layer 1, consist of node function with adaptive parameters as follows:

$$o_j^1 = \mu_{Cj}(x), \quad j = 1, 2 \tag{1}$$

where x is input value, o_j^1 is membership value of fuzzy variable μ_{Cj}, Cj shows constant coefficients and j is the number of rules. Output of this layer is a fuzzy membership grade of the input, which can be described with Gaussian membership function as $\mu_{Cj}(x)$:

$$\mu_{Cj}(x) = \exp\left[-\left(\frac{x - c_j}{a_j}\right)^2\right], \tag{2}$$

where a_j and c_j are the parameters of the membership function.

In layer 2, fuzzy AND operators are used, to fuzzify the inputs. The output of this layer is represented as:

$$o_j^2 = w_i = \prod_{j=1,2} \mu_{Cj} = \mu_{Cj}(x) * \mu_{Dj}(y), \tag{3}$$

Layer 3, consists of fixed nodes, which are used to calculate the normalized firing strength. The output of this layer can be represented as:

$$o_j^3 = \bar{w}_j = \frac{w_j}{w_1 + w_2}, \quad j = 1, 2 \tag{4}$$

In layer 4, the nodes are adaptive. The output of each node in this layer is simply the product of the normalized firing strength and for a first order Takagi-Sugeno model. The output of this layer can be described as:

$$o_j^4 = \bar{w}_j f_j = \bar{w}_j(p_j x + q_j y + r_j), \quad j = 1, 2, \tag{5}$$

where $\bar{w}$ is the output of the previous layer, and p_i, q_i, and r_i are the consequent parameters. In the last layer of ANFIS, it performs the summation of all incoming signals from the previous layer 4 as:

$$o_j^5 = \sum\nolimits_j \bar{w} f_j = \frac{\sum_j w_j f_j}{\sum_j w_j}. \tag{6}$$

3.5 *Hybrid Learning Algorithm (HLA)*

This algorithm is a combination of the gradient descent and the least squares methods which are used to minimize the error in the learning stage. The HLA consist of two passes known as forward and backward pass. In the forward pass, the info flows forward until o_j^4 and the consequent parameters are determined by the least square approach. In the backward pass, the error signals propagate backward and the premise parameters are updated using gradient descendent approach. This process is shown in Fig. 3. Hence, HL approach is much faster by reducing the search space dimensions of the [18].

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_1}{w_1 + w_2} f_2 \tag{7}$$

$$= \bar{w}_1 f_1 + \bar{w}_2 f_2 \tag{8}$$

$$= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \tag{9}$$

where p_1, q_1, r_1, p_2, q_2, and r_2 are the linear consequent parameters.

For the optimal values of these parameters, the least square method is used. When the premise parameters are not fixed, the search space becomes larger and the convergence of the training becomes slower. In the forward pass least square method is used to optimize the consequent parameters. In the backward pass, the error signals propagate backward and premise parameters are updated with the help of gradient descent used to optimize the premise parameters method by keeping consequent parameter fixed. Parameters updating rule is given as [18]:

$$a_{jl}(t+1) = a_{jl}(t) - \varpi . \frac{\partial E}{\partial a_{jl}} \tag{10}$$

where a_j are the adaptive parameters. ϖ is the learning rate for parameter a_{jl}, gradient is obtained by using the chain rule as [31]:

$$\frac{\partial E}{\partial a_{jl}} = \sum_{j=1}^{l+1} \frac{\partial E_j}{\partial \mu_{A_{jl}}} . \frac{\partial \mu_{A_{jl}}}{\partial a_{jl}} \tag{11}$$

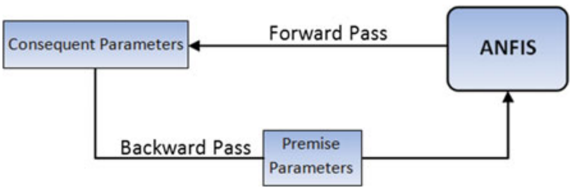

Fig. 3 Forward and backward pass in the hybrid algorithm

The consequent parameters which are found during the forward pass, are used to calculate the output of ANFIS. The output error is used to adapt the premise parameters by means of a standard back propagation algorithm. It has been proven that this hybrid algorithm is highly efficient in training the ANFIS systems [19].

3.6 ANFIS for Visual Attention Area Modelling

Neuro-Fuzzy technique has been used for intelligent modelling and control of uncertain systems. It is based on the input/output data pairs collected by the system. ANFIS is selected to predict the value of eye-gaze coordinates on the visual display. We need to apply several steps to train the model using ANFIS. These steps include: to define input/output values, to define fuzzy sets for the input values; to define fuzzy rules; and to create and train the neural network.

To implement and test the ANFIS, a development tool is required. Matlab TS fuzzy logic toolbox (FLT) [20] was selected as the development tool. This tool provides an environment to build and evaluate fuzzy systems using a graphical user interface (GUI). It consists of a FIS editor, the rule editor, a membership function editor, the fuzzy inference viewer, and the output surface viewer.

The FIS editor displays general information about a fuzzy inference system. The membership function editor is the tool that displays and edits the membership functions associated with all input and output variables. The rule editor allows the user to construct the rule statements. The rule viewer allows users to interpret the entire fuzzy inference process at once. The ANFIS editor GUI menu bar can be used to load a FIS training initialization and save the trained FIS.

3.7 Learning in ANFIS Using a Simulated Game

The basic purpose of our environment is to predict the value of the eye-gaze using the ordinary web camera. Using the mouse coordinates we want to construct a system that learns from x and y coordinates. Based on this learning it will adapt predicted eye-gaze coordinates of the visual attention area in the environment. This model of learning is a three-step process: first, we gather the data which will be used for learning, then create the learning model based on the data which we gather in first step, and finally adapt the learning model and deploy it in the environment.

Selection of the input have high influence on the prediction model. For this reason, it is of great importance to collect significant inputs and then design the system based on them. We need to adopt such a technique from which we can get maximum advantage. A suitable technique, that is based on the assumption that the ANFIS model is with the smallest Root Mean Square Error (RMSE) using a small number of data [32]. In our model only two inputs (x_g,y_g) are selected for the learning purpose.

3.8 ANFIS Training in the Environment

This process starts after receiving the training input/output data set. Two vectors are used to train the ANFIS, as training data is a set of input and output vectors. Training data set is used to find the parameters for the membership function. Also, in this process a threshold value for the error between the actual and desired output is defined. If this error is greater than the threshold value then the parameters are updated using the gradient decent method. It continues to change the premise parameters until the error value becomes less then the threshold value. The checking data set is then used to compare the model with the data [18].

ANFIS training rules use hybrid learning with combination of least square and gradient descent methods to determine the consequent parameters. The aim of using ANFIS is to identify the visual attention area on display. The further goal is to achieve the best performance possible; this is totally dependent on how we create a suitable training data-set to train the Neuro-Fuzzy system. Considering this reason, we need to gather training data from different users with different age and from male and female and on different display sizes and with different camera resolutions.

3.9 ANFIS Training in Matlab

We have used ANFIS Matlab toolkit [20] for this purpose. The first step is to prepare the training data in Matlab. ANFIS training uses the 'anfis.m' function. Evaluation of the system compared to the desired output is conducted using the 'evalfis.m' function. The data-set must be in a matrix form before we input it to the 'anfis.m' function. Where each column is used to represent the number of input and rows represent data-sets. The matrix contains as many input columns as needed but the last column must be the output.

As in traditional approaches you need to decide or ask expert's to create the membership function, or use command 'fismat.m', which provide help in the creation of the initial set of membership functions. After the creation of membership functions, the system training begins. We input the training data we defined using the command 'anfis.m'.

After the training process, the final training membership functions and training error are produced, Fig. 4. The checking data-set can be used in conjunction with the training data-set to improve accuracy. We can use as much data to train the system as needed, even ANFIS can be used with single training data-set but input of checking data increases the chance that the system quality will increase. Figure 4 illustrates the trend of errors using hybrid learning algorithms of ANFIS. These depict that error monotonically decreases with an increase in number of epochs.

As soon as the system training is complete, the process of evaluation of the system begins. We can enter the input data sets into our trained fuzzy system, but this

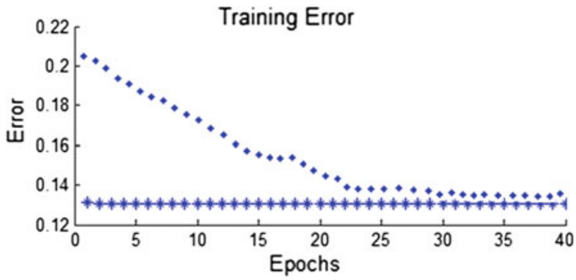

Fig. 4 Trend of errors with ANFIS

data-set do not include the output column in the matrix. The function 'evalfis.m' represents the output of the ANFIS system in our developed environment. There are different measurement parameters to test the effectiveness and efficiency of the system; few of them are Root Mean Square

Error (RMSE), Mean Average Error (MAE), Mean Absolute Percentage Error (MAPE), Normalized Mean Square Error (NMSE), coefficient of determination (R^2) and Coefficient of Correlation (R) between the desired learner contexts and the learning content format as the system input/output. We will discuss them in latter section. Once the ANFIS is trained, we can test the system against different sets of data values to check its functionality.

The system training methodology is summarized in Fig. 5. For each time instant we have two separate input parameters which control the eye-gaze x and mouse x coordinates (x_g, x_m) and one output coordinate $\tilde{x}_g$. At the same time, other two input parameters were control the eye-gaze y and mouse y coordinates (y_g, y_m) and one other output coordinate $\tilde{y}_g$.

Further, 'genfis1.m' function is used to generate an initial single output FIS matrix from training data. First, we need to initiate our model with default values for membership function numbers "13*13" and further we need to define the type of membership functions (these may be Gaussian curve, Generalized bell-shaped and Triangular-shaped; with the following commands 'gaussmf.m', 'gbellmf.m' and 'trimf.m'). There are few others as well but in our model we are just using these three.

3.10 ANFIS Structure and Takagi–Sugeno fuzzy inference

The conditions that determine the learning content format depend completely on experts. Each of the two input conditions is represented by the following term sets. Display is divided in four grids, Each of x_g, y_g, x_m and y_m are shown in Fig. 6 representing the following linguistic terms for defining the membership functions, upper-left-corner (ULC), Lower-Left-Corner (LLC), Upper-Right-Corner (URC), Lower-Right-Corner (LRC), Upper-Mid_Centre (UMC), Lower-Mid_Centre (LMC), Centre (C), Right-Centre (RC), Left-Centre (LC), Grid-1-Center (G1C), Grid-2-Center (G2C), Grid-3-Center (G3C) and Grid-4-Center (G4C).

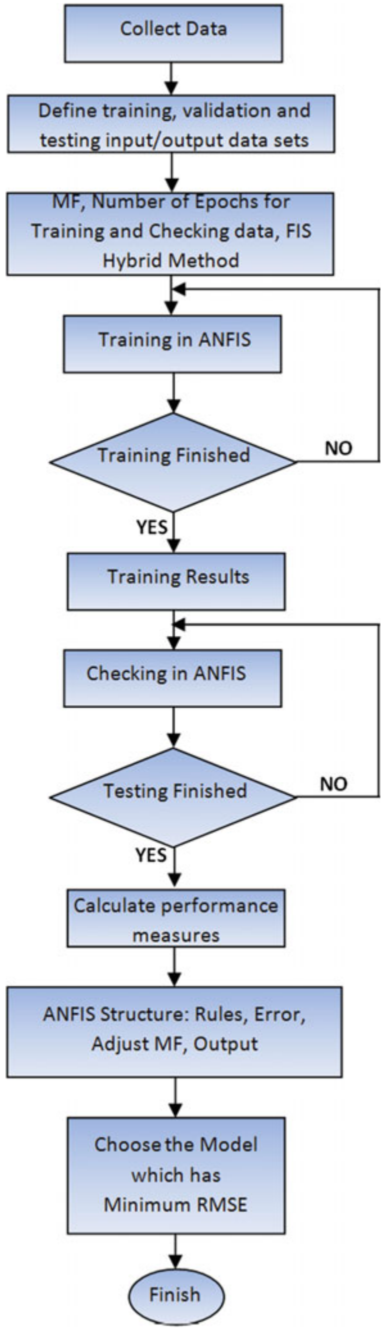

Fig. 5 Flowchart of the PAA_ANFIS

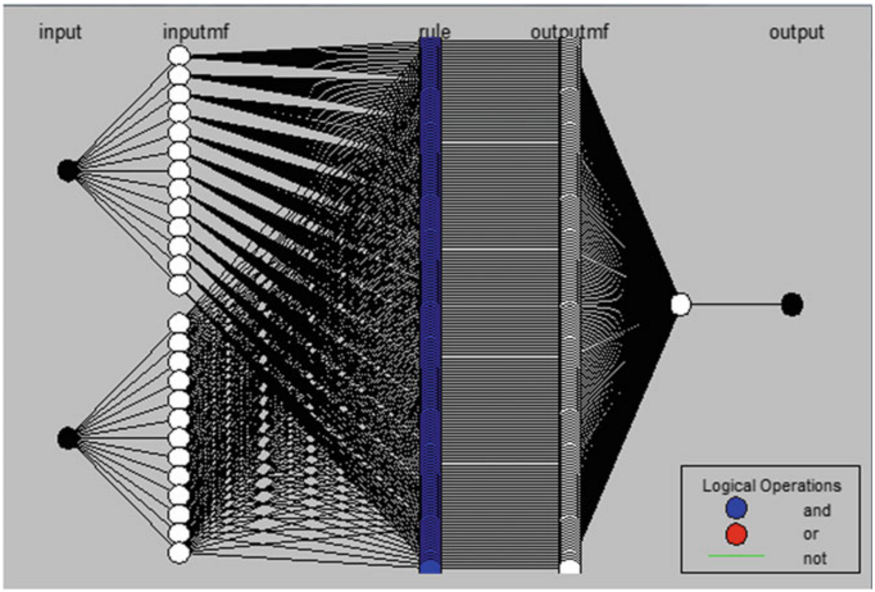

Fig. 6 ANFIS structure with two inputs, one hundred sixty nine rules and one output

In our case, the number of membership functions for each input is 13, so the total numbers of membership functions used is 52. The generated TS-FIS structure contains 169 fuzzy rules. Each rule has one output membership function which is of type “linear”. ANFIS has a multi-input single output (MISO) structure. Membership function parameters adjustment is done through the use of gradient based method, which reflects how well different set of training data are modelled; as this data represent different conditions in its parameters.

Several experiments have been conducted to assess whether the proposed model with combination of ANFIS has produced acceptable result. ANFIS network training involves mapping inputs through input membership functions and mapping output through output membership functions. The parameters associated with each membership function will keep changing throughout the learning process, but are fixed afterwards.

The process of training, in ANFIS begins by specifying the number of sets of each input variable and shape of their membership functions. Then, all the training data passes through the Neural Network (NN), to adjust the input parameters to find the input/output relationships. Another reason for passing the data in NN is that the error can be minimized.

In this paper, an ANFIS model based on both Neural Network (NN) and Takagi–Sugeno (TS) fuzzy inference system, has been developed to adapt learning content formats for our gaming environment. The experiments were divided into two ANFIS structures to demonstrate learning activities in different contexts. For this purpose, computer gaming environment using Matlab statistical validation indexes were used for determining the performance of the best proposed model. Once the input/output have been recorded it is necessary to validate the quality of the model results after training and before deploying it into the environment.

We trained the system with a variety of different settings such as dataset sample, epoch number, membership function type and number, and number of inputs to achieve the best performance. These results are discussed in the last section

3.11 Proposed Model for Eye-Gaze Prediction

The steps involved in the implementation of the proposed model using ANFIS include:

Step1: Collecting data from the environment.
Step2: Determining training, validation and testing input- output sets.
Step3: Generating initial first order TSK fuzzy inference system.
Step4: Extracting rules.
Step5: Tuning antecedent and consequent parameters by ANFIS hybrid learning algorithm. After setting training parameters, learning algorithm will be continued to reach the specified error goal or maximum number of epochs.
Step6: Calculating performance measures.
Step7: Selecting the model with the minimum validation error.
Step8: Running the model.

By using TS-FIS and neural methodology in form of HLA, we generated the optimal ANFIS structure. This procedure establishes the first order TSK fuzzy inference system that models the data behaviour in a better way. The flowchart of our proposed model is shown in Fig. 5.

4 Results and Discussions

Many experiments were conducted, and data were divided into two separate sets; one for the training and other for the checking. The training data set was used to train the ANFIS, whereas the checking data set was used to verify the accuracy and the effectiveness of the trained ANFIS model, so it can be adopted in the environment. In order to find out the optimal solution, that can address the requirements of our problem of identifying the visual attention area with the help of eye-gaze. We need to consider the numbers of important factors which can affect the system efficiency before we deploy them in the environment.

Some of the aspects are taken into account in relation to ANFIS system training such as: number of membership functions, type of membership functions, training data samples and epoch number which generated different MAE and MASE.

4.1 Performance Measures

We applied the commonly used performance measures in forecasting problems to compare and evaluate the accuracy of our model. These performance measures are Root Mean Square Error (RMSE), Mean Average Error (MAE) and Coefficient of Determination (R^2) between the desired learner contexts and the learning content. Which are calculated by

$$MAE = \frac{1}{N}\sum_{q=1}^{Q}\left|y_q - \tilde{y}_q\right|, \tag{12}$$

RMSE is defined as the square root of the mean squared error as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{q=1}^{Q}\left(y_q - \tilde{y}_q\right)^2} \tag{13}$$

The difference of the RMSE between observed and predicted values was computed for each trial with different epoch numbers, and the best structure was determined by the lowest value of the RMSE. Coefficient of Determination (R^2) is used, to measure of how well future outcomes are likely to be predicted by the framework. And can be given as:

$$R^2 = 1 - \frac{\sum_{q=1}^{Q}\left(y_q - \tilde{y}_q\right)^2}{\sum_{q=1}^{Q}\left(y_q - \bar{y}_q\right)^2} \tag{14}$$

where N is the number of testing patterns, $\tilde{y}_q$ is the predicted values, and y_q is original value, $\bar{y}_q$ is the average of all original values to Q testing patterns. The value of R^2 lies in between 0 and 1.0. It is used to describe how well a regression line fits a set of predicted data. The value of R^2 near 1.0 indicates that a regression line fits the predicted data well, while value of R^2 closer to 0 indicates a regression line does not fit the predicted data very well. The model was trained for 200 epochs and it was observed that the most of the learning was completed in the first 90 epochs as the root mean square error (RMSE) settles down to almost 0.01785 at 130 th epoch. Figure 7 shows the training RMSE curve for the ANFIS model. After training the ANFIS, it is found that the shape of membership functions (Fig. 8) is slightly modified. This is because of the close agreement between the knowledge provided by the expert and input/output data pairs. Figures 9 and 10, compares the accuracy of eye-gaze and ANFIS prediction values.

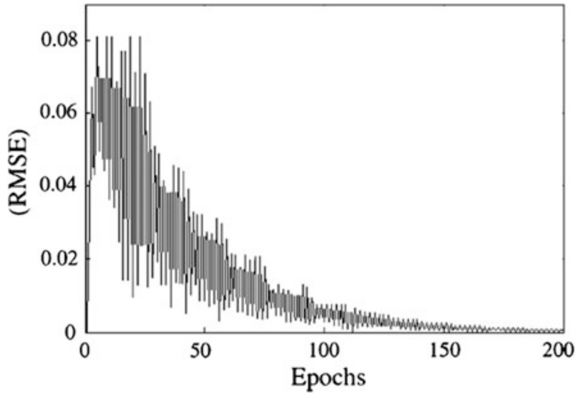

Fig. 7 Training trend of errors with ANFIS

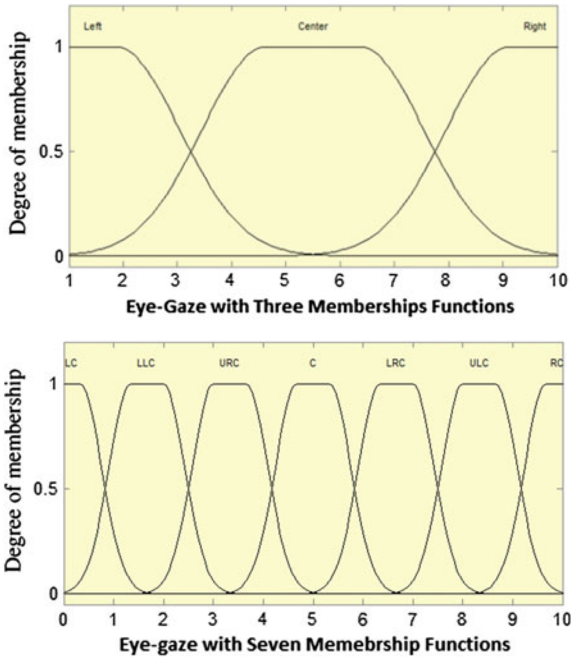

Fig. 8 Membership functions of eye-gaze with three MF's (*Left*) and membership functions of eye-gaze with seven MF's (*Right*)

5 Numerical Results

The first system was structured by selecting two inputs and feeding them into the network given the name as function number (3*3). Two input parameters x_m and x_g were selected. There were three membership functions represented as "Centre, Right, Left". First, we tested our model with default values for membership function number (3*3); membership function types "Bell-shaped, Gaussian, Triangular" are used. These defaults provide membership functions on each of the

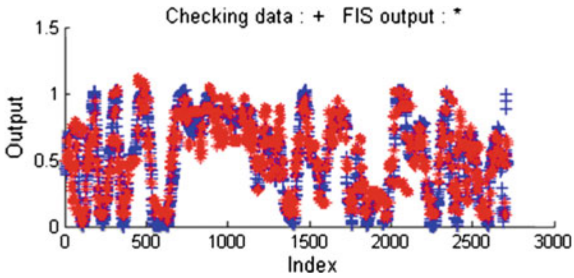

Fig. 9 Modelling performance of estimated (*red*) versus real values (*blue*) using ANFIS in simulated environment

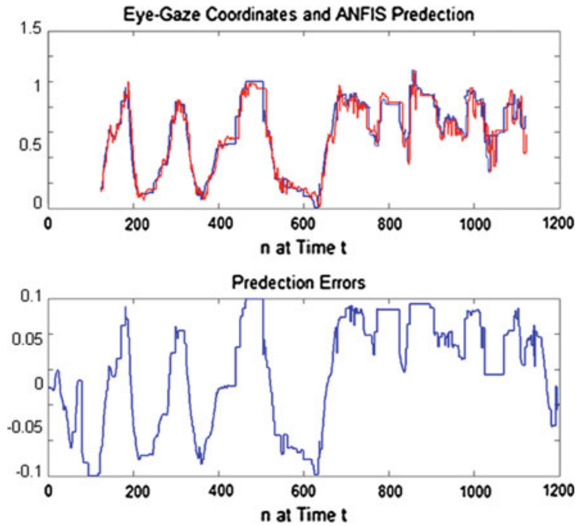

Fig. 10 Prediction error in simulated environment; ANFIS model (*blue line*)

two inputs, six altogether. The generated fuzzy inference system structure contains 9 fuzzy rules. More details are tabulated in Table 1 with different measurement techniques.

Next, we use membership function number (13*13). The first system was structured by selecting two inputs and feeding them into the network. Two input parameters x_m and x_g were selected. The ANFIS has thirteen membership functions represented as "ULC, LLC, URC, LRC, UMC, LMC, RC, LC, G1C, G2C, G3C, G4C, C". Now we have used/consulted a human expert who has selected/suggested these membership functions. Again membership function types "Bell-shaped, Gaussian, Triangular" are used for number (13*13). These human experts provide membership functions on each of the two inputs, twenty-six altogether. The generated fuzzy inference system structure contains 169 fuzzy rules. More details are tabulated in Table 2 with different measurement techniques.

Table 1 The scenario$_1$ (membership functions number "3*3")

ANFIS parameter	Scenario$_{(1)}$		Scenario$_{(1)}$		Scenario$_{(1)}$	
Number of inputs	2					
Membership function type	Bell-Shaped		Gaussian		Triangular	
Number of membership function	3*3 (Right, Centre, Left)					
Training data set	20	28	20	28	20	28
Checking data set	40	50	40	50	40	50
Epoch number	10	17	10	17	10	17
Number of nodes	35		35		35	
Number of linear parameters	27		27		27	
Number of non-linear parameters	18		12		18	
Total number of parameters	45		39		45	
Number of fuzzy rules	9					
MAE	0.03923	0.04175	0.01843	0.02888	0.02510	0.03479
RMSE	0.19807	0.20432	0.13576	0.16994	0.15843	0.18652

Table 2 The scenario$_2$ (membership functions number "13*13")

ANFIS parameter	Scenario$_{(2)}$		Scenario$_{(2)}$		Scenario$_{(2)}$	
Number of inputs	2					
Membership function type	Bell-Shaped		Gaussian		Triangular	
Number of membership function	13*13(ULC,LLC,URC,LRC,UMC,C,LMC,RC,LC,G1C, G2C,G3C,G4C)					
Training data set	10	30	10	30	10	30
Checking data set	20	60	20	60	20	60
Epoch number	8	15	8	15	8	15
Number of nodes	195		195		195	
Number of linear parameters	305		305		305	
Number of non-linear parameters	28		18		28	
Total number of parameters	333		323		333	
Number of fuzzy rules	169					
MAE	0.01547	0.05432	0.01325	0.02234	0.02364	0.04587
RMSE	0.12441	0.23307	0.11511	0.14947	0.15375	0.21417

Next, we use membership function number (5*5). The first system was structured by selecting two inputs and feeding them into the network. Two input parameters x_m and x_g were controlled. Has thirteen membership functions represented as "ULC, LLC, URC, LRC, C". Again, membership function types "Bell-shaped, Gaussian, Triangular" are used for number (5*5). These human experts provide membership functions on each of the two inputs, ten altogether. The generated fuzzy inference

Table 3 The scenario$_3$ (membership functions number "5*5")

ANFIS parameter	Scenario$_{(3)}$		Scenario$_{(3)}$		Scenario$_{(3)}$	
Number of inputs	2					
Membership function type	Bell-Shaped		Gaussian		Triangular	
Number of membership function	5*5 (ULC, LLC, URC, LRC, Centre)					
Training data set	60	80	60	80	60	80
Checking data set	120	160	120	160	120	160
Epoch number	10	20	10	20	10	20
Number of nodes	265		265		265	
Number of linear parameters	440		440		440	
Number of non-linear parameters	20		15		20	
Total number of parameters	460		455		460	
Number of fuzzy rules	25					
MAE	0.04923	0.06875	0.02884	0.03888	0.03254	0.04847
RMSE	0.22188	0.26220	0.16982	0.19718	0.18042	0.22016

system structure contains 25 fuzzy rules. More details are tabulated in Table 3 with different measurement techniques.

Next, we use membership function number (7*7). The first system was structured by selecting two inputs and feeding them into the network. Two input parameters x_m and x_g were controlled. Has seven membership functions represented as "ULC, LLC, URC, LRC, RC, LC, C". Again membership function types "Bell-shaped, Gaussian, Triangular" are used for number (7*7). A human expert provides membership functions on each of the two inputs, fourteen altogether. The generated fuzzy inference system structure contains 49 fuzzy rules. More details are tabulated in Table 4 with different measurement techniques.

The ANFIS models are evaluated based on their performance in training and checking sets as shown in Tables 1, 2, 3 and 4. The ANFIS models have shown significant performance variations against the evaluation criteria in terms of data sample, number and type of membership functions. It appears that the ANFIS models are more accurate and consistent in different subsets, where average values of RMSE and MAE are 0.20896 and 0.04367 respectively.

The experimental results shown in the tables above demonstrate that the fuzzy rule base selected by the human expert produces consistent results for the test data used i.e. 10 and 18 in Scenario$_{(1)}$,20 and 62 in Scenario$_{(2)}$, 60 and 80 in Scenario$_{(3)}$ and, finally, 60 and 100 in Scenario$_{(4)}$. However, not all situations are covered by expert's fuzzy rules and some missing rules are detected by ANFIS. Tables 1, 2, 3 and 4 show that all situations for all input attributes are covered by the sets of 9, 169, 25, and 49 rules; however some of the rules have been found to produce illogical decisions because the training sample size was not large enough to cover all possible cases.

Table 4 The scenario$_4$ (membership functions number "7*7")

ANFIS parameter	Scenario$_{(4)}$		Scenario$_{(4)}$		Scenario$_{(4)}$	
Number of inputs	2					
Membership function type	Bell-Shaped		Gaussian		Triangular	
Number of membership function	7*7 (ULC,LLC,URC,LRC,RC,LC,C)					
Training data set	60	100	60	100	60	100
Checking data set	100	210	100	210	100	210
Epoch number	15	30	15	30	15	30
Number of nodes	397		397		397	
Number of linear parameters	900		900		900	
Number of non-linear parameters	45		30		45	
Total number of parameters	945		930		945	
Number of fuzzy rules	49					
MAE	0.05923	0.08675	0.04843	0.07852	0.05963	0.08561
RMSE	0.24337	0.29453	0.22007	0.28021	0.24419	0.29259

The models are evaluated based on their performance measures (RMSE and MAE) in training and checking data-sets. The results shown in Table 1, 2, 3 and 4 reflect that ANFIS model have shown significant performance variations against the evaluation criteria in terms of data sample, number of membership functions, and type of membership functions. Scenario$_{(2)}$, which consists of two inputs and membership function(3*3), has minimum RMSE,MAE and the highest correlation. It appears that the ANFIS models are accurate and consistent in different subsets as the lowest value of RMSE is 0.11511 Scenario$_{(2)}$ and the highest value of the RMSE is 0.29453 Scenario$_{(4)}$.

Both the type and number of MF are important in building the ANFIS architecture. The required number of membership functions is determined through trial and error based on error values. It indicates that each ANFIS model is very sensitive to the type and number of MF. Increasing the number of MF's per input does not necessarily increase the model performance, but usually leads to model over fitting. The RMSE values were used to determine the best MF and epoch number in order to select the best fit model. From these results, the Gaussian membership function with epoch numbers of 8 and 15 was found to be the best fit model with the lowest RMSE value and with epoch 15 and 30 (Scenario$_{(2)}$ and Scenario$_{(4)}$).

Tables 1, 2, 3 and 4 show different approximations regarding different epoch numbers and membership functions (3*3, 13*13, 7*7, and 5*5). It is shown that the number of training samples has influenced ANFIS behaviour by adding new possibilities to produce more acceptable results. For Fuzzy Logics it is of great importance to train the system with different training data that tries to covers different possible data samples. Training is also important to build ANFIS architecture; as they include epoch numbers, the error, the initial step size, and decrease or increase rate of the step size.

In Scenario$_{(1)}$, Scenario$_{(2)}$, Scenario$_{(3)}$ and Scenario$_{(4)}$ different epoch numbers are used. One important point is that increasing the epoch number for a training data set does not necessarily improve the system performance significantly but it helps to overcome the problem of over-fitting. The trend of predicted data using ANFIS is illustrated in Figs. 11 and 12 along x and y axis respectively. Figure 11 illustrates that predicted data using ANFIS is converging more to the regression line. Approximately 91.23 % of the predicted data fits along the regression line for x-axis, while in case of Fig. 12, 89.01 % predicted data is converging to the regression line for y-axis.

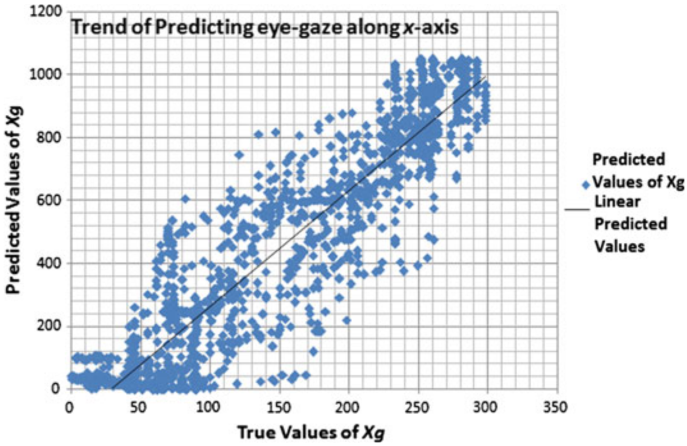

Fig. 11 Trend of predicted x_g using PAA_ANFIS

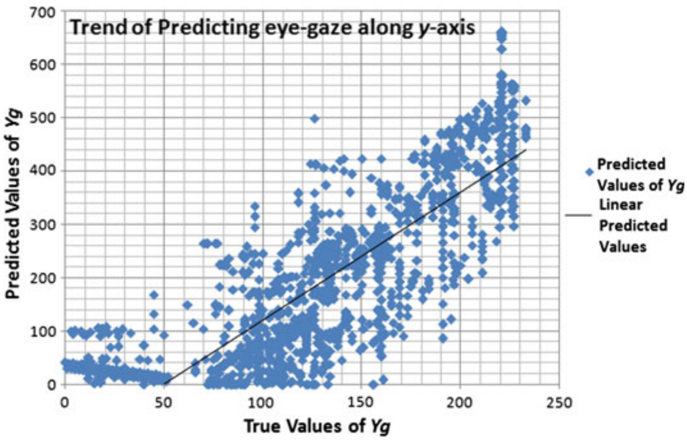

Fig. 12 Trend of predicted y_g using PAA_ANFIS

5.1 Comparative Performance Analysis

The performance analysis and a comprehensive difference of ANFIS and linear regression (LR) are presented in Table 5, on the basis of trend of generated data, fitness of test and predicted data, trend of error and execution time. The results obtained have been compared using 150 training data and 125 testing data pairs.

It is observed that ANFIS result is highly correlated with least RMSE as comparison with LR. Decreasing rate of error is higher because of its hybrid learning algorithm. Mean execution time is also computed which is competitively low. Figures 13 and 14 depict the fitness of generated data over test data. The CORR and RMSE between predicted and test data are shown in Table 6.

Results in the tables are sufficient proof, that ANFIS model performance is, in general, accurate and acceptable, where some rules are covered by human expert training data-sets and the missing rules are extracted by ANFIS. Furthermore ANFIS has capability of generating linear relationship in input-output data. The results of

Table 5 The comprehensive analysis of ANFIS and LR for eye-gaze prediction

Factors	ANFIS	LR
Capturing of fuzziness	Using FIS	Not available
Correlation in test and generated data	High	Less
Coefficient of determination	High	Less
Error rate	Low	High
Execution time	Less	High

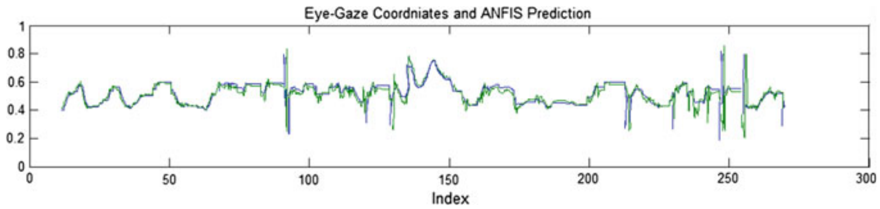

Fig. 13 Trend of prediction of eye-gaze coordinates (*green line*); ANFIS model (*blue line*)

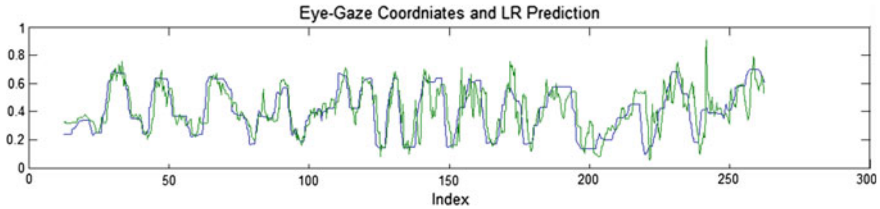

Fig. 14 Trend of prediction of eye-gaze coordinates (*green line*); LR model (*blue line*)

Table 6 Comparative analysis of ANFIS and LR for eye-gaze prediction

	CORR	RMSE
ANFIS	0.9134	0.1038
LR	0.8269	0.4873

the use of ANFIS models demonstrate that ANFIS can be successfully applied to predict the eye-gaze visual attention area in the environment.

6 Conclusions

In this research, we used ANFIS model to predict the user attention area using an ordinary web-camera. We implemented the PAA_ANFIS in developed game to investigate its accuracy. The performance of ANFIS was evaluated using standard error measurements techniques which revealed the optimal setting necessary for better predictability. PAA_ANFIS adopted a hybrid approach that combined the Fuzzy Inference System with the Neural Network in determining a complete fuzzy rule system. Combining a fuzzy inference system into an ANFIS Structure, we take advantage of neural methodology to train a fuzzy system that increases its capability for dealing with nonlinearity and imprecision in addition to its easier implementation.

The PAA_ANFIS approach has successfully solved the problem of prediction. The utilized performance measures confirmed higher forecasting accuracy of our model in comparison to Linear Regression. Hence ANFIS appears as a better choice then linear regression.

Future research on this scenario needs to study new techniques which are evolving and online. These techniques have advantage over ANFIS, that we do not need to ask expert to define membership function. As these memberships functions are even artificial which cannot reflect the real world scenario. So we expect that using an evolving technique will increase the feasibility of developing more effective and predictive systems.

Acknowledgments We gratefully acknowledge Prof. Dr. Plamen Angelov (leads the intelligent System Research Group at School of Computing and Communications Lancaster University) for providing valuable comments and suggestion during span of this research work, without which we have been unable to complete this research.

References

1. Aarts, E., Harwig, R., Schuurmans, M.: Ambient intelligence. In: Denning, P. (ed.) The Invisible Future, pp. 235–250. McGraw Hill, New York (2001)
2. Aarts, E., Collier, R., van Loenen, E., de Ruyter, B. (eds.): Ambient intelligence. In: Proceedings of the First European Symposium, EUSAI2003. Lecture Notes in Computer Science, vol. 2875, pp. 432. Springer (2003)

3. Abraham, A.: Adaptation of fuzzy inference system using neural learning, fuzzy system engineering: theory and practice. In: Nedjah, N., et al. (eds.), ch. 3, pp. 53–83. Springer, Berlin, Germany (2005)
4. Angelov, P., Filev, D. (2004) An approach to on-line identification of takagi-sugeno fuzzy models. IEEE Trans. Syst. Man Cybern. Part B—Cybern **34**(1), 484–498. ISSN 1094-6977
5. Anifowose, F.A., Labadin, J., Abdulraheem, A.: Prediction of petroleum reservoir properties using different versions of adaptive neuro-fuzzy inference system hybrid models. Int. J. Comput. Inf. Syst. Ind. Manage. Appl. **5**, 413–426 (2013)
6. Areerachakul, S.: Comparison of ANFIS and ANN for estimation of biochemical oxygen demand parameter in surface water. Int. J. Chem. Biol. Eng. **6**, 286–290 (2012)
7. Asteriadis, S., Karpouzis, K., Kollias, S.: A neuro-fuzzy approach to user attention recognition. ICANN'08: Proceedings of the 18th international conference on Artificial Neural Networks, Part I, pp. 927–936. Springer, Berlin, Heidelberg (2008)
8. Berenji, H.R., Khedkar, P.: Learning and tuning fuzzy logic controller through reinforcements. IEEE Neural Netw. **3**(5), 724–740 (1992)
9. Bosse, T., Hoogendoorn, M., Klein, M., Treur, J.: A component-based ambient agent model for assessment of driving behaviour. In: Sandnes, F.E. et al. (eds.) Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing, UIC'08. Lecture Notes in Computer Science, vol. 5061, pp. 229–243. Springer (2008)
10. Bosse, T., Maanen, P., van, P., Treur, J.: Simulation and formal analysis of visual attention. Web Intell. Agent Syst. J. **7**, 89–105 (2009)
11. Bosse, T., Hoogendoorn, M., Memon, Z.A., Treur, J., Umair, M.: An adaptive human-aware software agent supporting attention-demanding tasks. In: Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.) Principles of practice in multi-agent systems, pp. 292–307. Springer, Berlin, Heidelberg, Standard: ISSN: 0302-9743 ISBN: 978-3-642-11160-0 (2009b)
12. Boyacioglu, M.A., Derya, A.: An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul Stock Exchange, expert systems with applications, vol. 37, pp. 7908–7912. Elsevier (2010)
13. Elwakdy, A.M., Elsehely, B.E.: Speech recognition using a wavelet transform to establish fuzzy inference system through subtractive clustering and neural network (ANFIS). Int. J. Circuits Syst. Signal Process. **2**(4) (2008)
14. Gärdenfors, P.: How Homo Became Sapiens: On The Evolution Of Thinking. Oxford University Press, Oxford (2003)
15. Haykin, S.: Neural Networks: A Comprehensive foundation second edition". Pearson Prentice Hall, Delhi India (2005)
16. Horvitz, E., Kadie, C., Paek, T., Hovel, D.: Models of attention in computing and communication: from principles to applications. Commun. ACM **46**(3), 52–59 (2003)
17. Itti, L., Koch, C.: Computational modeling of visual attention. Nat. Rev. Neurosci. **2**(3), 194–203 (2001)
18. Jang, J.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybernet. **23**, 665–685 (1993)
19. Jang, J.S.R., Sun, C.T.: Neuro-fuzzy modeling and control. Proc. IEEE **83**(3), 378–406 (1995)
20. Jang, J.S., Gulley, N.: Fuzzy Logic Toolbox User's Guide, the Math Works Inc (1995)
21. Jang, J.S., Sun, C.T., Mizutani, E.: Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, New Jersey (1997)
22. Jassbi, J.J.: A Comparison of Mamdani and Sugneo Fuzzy Inference Systems for a Space Fault Detection Application, IEEE, pp.1–8 (2008)
23. Klir, G.J.: Fuzzy Sets and Fuzzy Logic, PHI publications, ISBN:81-203-1136-1 (1995)
24. Mashrei, M.A.: Neural network and adaptive neuro-fuzzy inference system applied to civil engineering problems. In: Fuzzy Inference System - Theory and Applications, ISBN 978-953-51-0525-1, Hard cover, 504 pp. InTech (2012)
25. Mao, Y., Suen, C.Y., Sun, C., Feng, C.: Pose estimation based on two images from different views. In: Eighth IEEE Workshop on Applications of Computer Vision (WACV), Washington, DC, USA, IEEE Computer Society 9 (2007)

26. Memon, Z.A., Oorburg, R., Treur, J., Umair, M., de Vos, M.: A Software environment for a human-aware ambient agent supporting attention-demanding tasks. In: Tenth International Conference on Computational Science, ICCS'10, Elsevier, pp. 2033–2042, vol. 1 (2010)
27. Mellit, A., Soteris, A.K.: ANFIS-based modelling for photovoltaic power supply system: A case study. Renew. Energy **36**, 250–258 (2011)
28. Meyer, A., BÄohme, M., Martinetz, T., Barth, E.: A single-camera remote eye tracker. In: Lecture Notes in Artificial Intelligence, pp. 208–211. Springer (2006)
29. Moreno, J.: Hydraulic plant generation forecasting in Colombian power market using ANFIS. Energy Econ. **31**, 450–455 (2009)
30. Riva, G., Vatalaro, F., Davide, F., Alcañiz, M. (eds.) Ambient Intelligence. IOS Press (2005)
31. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: "Learning internal representations by error propagation. In: Rumelhart, D.E., James, L., McClelland (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, ch. 8, pp. 318–362. MIT Press, Cambridge (1986)
32. Shing, J., Jang, R.: Input selection for ANFIS learning. In: Proceedings of IEEE Fifth Int'l Conference Fuzzy Systems (1996)
33. Tarzia, S.P., Dick, R.P., Dinda, P.A., Memik, G.: Sonar based measurement of user presence and attention. In: Ubicomp'09: Proceedings of the 11th international conference on Ubiquitous computing, pp. 89–92. ACM, New York (2009)
34. Turatto, M., Galfano, G.: Color, form and luminance capture attention in visual search. Vision. Res. **40**, 1639–1643 (2000)
35. Vijila, C.K.S., Kumar, C.E.S.: Interference cancellation in EMG signal using ANFIS. Int. J. Recent Trends Eng. **2**(5), 244–248 (2009)
36. Webb, G.I., Pazzani, M.J., Billsus, D.: Machine learning for user modeling. User Model. User-Adap. Inter. **11**(1/2), 19–29 (2001)
37. Zounemat-Kermani, M., Teshnehlab, M.: Using adaptive neuro-fuzzy inference system for hydrological time series prediction. Appl. Soft Comput. **8**, 928–936 (2008)

Springer

Innovative Issues in Intelligent Systems
Sgurev, V.; Yager, R.R.; Kacprzyk, J.; Jotsov, V. (Eds.)
2016, X, 353 p. 168 illus., 65 illus. in color., Hardcover
ISBN: 978-3-319-27266-5