Predicting of eye trajectories based on the ART Neural network

Jong-Hei Ra, Yoon-Kyung Jung, Moon-Hyun Kim

Department of Electrical & Computer Engineering
SungKyunKwan University
300, Chunchun-dong, Jangan-gu, Suwon-si, Kyunggi-do, Korea

Abstract

In this paper an approach for achieving the robust and fast tracking and the prediction of eye regions from image sequences is proposed. The algorithm of predictive ART is based on ART2 neural network which is an unsupervised learning mechanism capable of self-organizing stable recognition categories in response to analog input patterns. The predictive ART performs on-line incremental learning of input patterns and predicts the non-linear data. Tracking eyes of frontal image as well as eyes of near frame image are demonstrated on a collection of image sequences achieved on the Internet. Some images are at low quality and the image size is too small, but the results are satisfactory. The system can be applied to real-time servo control, intelligent room system, and autonomous robot and preprocessing of face recognition for security.

Keyword : Motion Tacking, Predictive ART, Eye Trajectory

1. Introduction

Existing studies on prediction have used as experiment data mostly obtained in fields of economy, engineering, industry, etc. In fact, prediction systems used in economy fields have been utilized for establishing business plans, predicting stock quotations tendencies, and controlling properties and stock[1]. In 1990, Santa Fe laboratory began to define time series data used in experiments and estimate objectively the performance of various techniques[2]. However the data used in the study, was one-dimensional such as stock index, exchange rates of currency and combined signals of computers, so it is difficult to compare the results of that study with results of this paper, which deals with multi-dimensional data that obtained from image sequences. In general, the statistical method can be used for predicting of linear data, but it is difficult to use non-linear data, which is obtained the real environment. In order to processing the non-linear data, various techniques are proposed such as Kalman Filter, Hidden Markov Model, Recurrent Neural Network, and Multi-layer perceptron, Radial Basis Function Neural Network, Time Delay Neural Network, Genetic Algorithm, Wavelet and Linguistic Fuzzy Regulation. According to the results of Santa Fe experiment, it is show that the performance of a network can be improves when many experimental data are used and the data have many noises, as possible[2][3][4].

Prediction method is classified into one-step prediction and multi-step prediction. One-step prediction is a method to predict the value of time $t+1$ at time t, and then predict the value of time $t+2$ with real input value at time $t+1$. Multi-step prediction is a method to predict all values of $t+1$, $t+2$, $t+3$,.., $t+N$ at time t. Because this method predicts any values based on predicted values over again, error rates increase geometrically. The prediction method, which we propose, detects the next position of moving object from image sequences, which is short-term forecasting and uses the one-step prediction. We use nonlinear data to represent moving object that is described in a vector system with velocity and direction.

The paper is organized as follow. Section 2 describes the predictive ART that tracks eye location in the next frame image and section 3 show the algorithm for predictive ART NN .

Section 4. Describes the stability evaluation of proposed neural network. And then section 5 shows experimental result on image sequences. Finally, Section 6 concludes with a summary and plans for future work.

2. Predicting the motion using neural network

Generally, the method of prediction for time series data such as laser data, synthetic data, stock market data, international exchange rates, voice data, was used the statistical approach, genetic algorithms, and neural network approach[1][2]. The time series predicting systems are widely adopted in the economic and engineering areas. In this paper, we are proposed the predictive ART, which modifies the ART2 neural network that allow real time sampling and predict the non-linear data.

2.1 Pre-processing

Pre-processing for input patterns of neural network determine the quality of the tracking system. The moving information of object in image sequences includes spatio-temporal data, and it is difficult to use directly input to the predictive ART. So, we used to spatial data, which is converted from the spatio-temporal data, for input of predictive ART neural network.

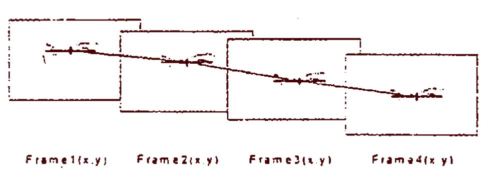

Figure. 1 : Eye trajectory of image sequences

Figure 1 shows the trajectory, which connects the four frames, can be presented as a point on the 8th dimension of view in Figure 2. Because only one point exists on the *8-th* dimension, it can be distinguished from other trajectory that follows different paths. We use the spatial trajectory as an input vector for the ART2 neural network.

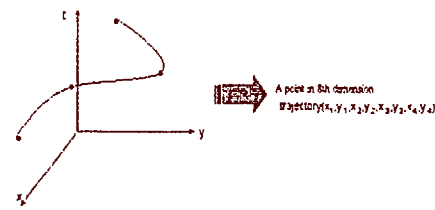

Figure 2 : Converting spatio-temporal data to spatial data

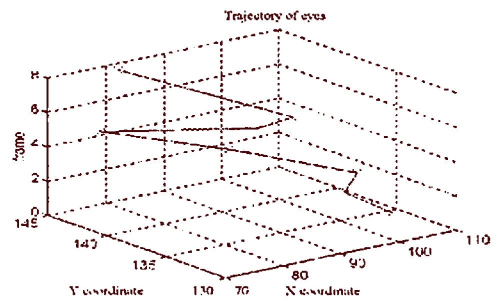

Figure. 3 : Eye trajectory for time

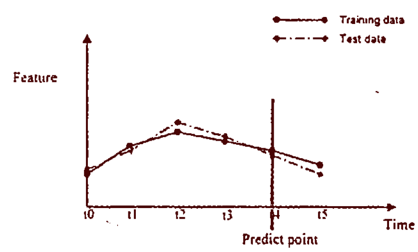

Figure. 4 : Assumption in predictive system

The Figure 3 shows the trajectory for the eye positions, which is an actual data obtained from experiments, in accordance with time, and note that the eye trajectory are non-linear data. We assumed, the next positions of trajectories that follow similar paths are also similar. This is illustrated in Figure 4. If two trajectories follow similar path before the embedding dimension($t4$), the next position of $t5$ will be similar.

If we establish the trajectories of moving object as a point in a higher dimension, the points of trajectories with similar paths are located a very close and can be classified as one class as shown in Figure 5.

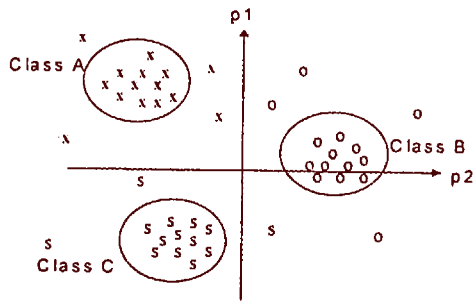

Figure 5 : Clustering of ART

2.1 Structure of predictive ART

The ART NN is an unsupervised neural network, which classifies the class according to their input pattern and is able to create new classes in the real processing as well as in the training processing. In this paper, we suggested predictive ART, which has been modified from ART2 NN. The structure of predictive ART is illustrated in Figure 6. Basically, the ART NN has taken the winner-take-all strategy which changes the weight of the output vector, is the nearest in characters to the input vector. If the output vector exceeds the vigilance value or do not satisfy the hypothesis-testing, the ART NN decides that it is different class with existing classes, and then makes a new class.

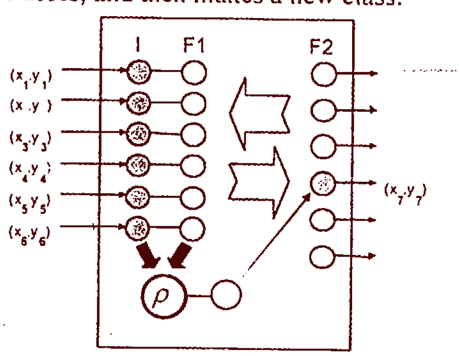

Figure. 6 : Structure of Predictive ART

Figure 6 shows the implemented neural network has a predictive ART structure, which is completely, connect the input *F1* layer with the output *F2* layer. The input *F1* layer detects the similar output neuron after comparing the input pattern with the output pattern and the output *F2* layer activates the corresponding output. the number of input neurons is $N_{property} \times (N_{frame} - 1)$ where $N_{property}$ is the number of attributes featuring the center of the eye and N_{frame} is the number of clustering frames and only N_{frame} of input neuron for more clearly and better understanding represents Figure 6. N_{frame} of input is reduced to $(N_{frame} - 1)$ because input vector consists of the velocities, which represent relative locations of movement. In the ART NN, output neuron means a class corresponding to the input pattern. The weight connected from F_2 layer to F_1 layer, which is each prototype's class, is used to measure the value of similarity between the input vector and its class.

3. Algorithm

3.1 Preprocessing

When the set ***Tr*** represents information about trajectories, which are obtained by the histogram matching method[5], input pre- processing is described as following. It is assumed that the k-th trajectory of the predictive ART neural network is Tr_k $\{(W_k, H_k), (x_1, y_1), (x_2, y_2) \ldots, (x_{N-1}, y_{N-1}), (x_N, y_N)\}$, where N is the number of image frames consisting of the *k-th* trajectory, *W* is the width of the *k-th* trajectory, *H* the height, x_j the x-coordinate of the center location of the eye at the *j-th* frame of the *k-th* input set and y_j the y-coordinate of the center location of the eye at the *j-th* frame of the *k-th* input set. In order to use the trajectory of moving object obtained from the eye-detecting module as the input of the neural network, the ND_k input patterns have been set up using Equation 1.

$$T=N-1$$

$$ND_k \{x_k(1), y_k(1), x_k(2), y_k(2), \ldots, x_k(T), y_k(T), x_k(T+1), y_k(T+1)\}$$

$$Xk(j) = {(xj+1-xi)}/{Wk}, \quad yk(j) = {(yj-1-yi)}/{H_k} \quad (1 \le j \le T) \quad (1)$$

Here, the last frame, $x_k(T+1)$, $y_k(T+1)$ is used to predicting frame and *T* is the embedding dimension for predicting the next image by saving the past data of the trajectory. As the ability of prediction depends on the way of representation for input patterns, the network use velocities between input frames, which is not affected by the absolute location of movement and normalizing the velocity vectors according to the width and height of image for ignoring the image size.

In Equation 1, the velocity vectors are normalized according to the width and ***Tr*** height of image. The first cluster is initialized with the center coordinate of the cluster and the prediction vector using the first input trajectory vector. And then the neural network decides whether each new vector will be assigned to the prior cluster or new clusters will be created according to the distance between the new vector and the existing cluster center. In case a new cluster is created, the neural network use the coordinates of the input vector which is used to create the new cluster and its member of vectors to assign the weight of the new cluster.

3.2 Select the winner neuron

Distance between the central point of cluster and the trajectory of eye, is computed by Equation 2.

$$Di=\sum_{j=1}^{j=T}(\lambda^{T-j})\sqrt{(Wi_j(x)-xk(j))^2+(Wi_j(y)-yk(j))^2} \qquad (2)$$

for D_i = a distance between the prototype of *i-th* output neuron and the input, W_{ij} (x) = a weight between the *i-th* cluster and the x coordinate of *j-th* frame, W_{ij} (y) = weight between the *i-th* cluster and the y coordinate of *j-th* frame, λ = recency weight, weight for the time ($0 < \lambda \leq 1$).

The pattern represents regency weight and weight for the time and ranges from 0 to 1 is 1, this represent that we don't consider weight for the time. Winner neuron is a neuron that has minimum distances between current input neurons and weights of output neurons.

$$C=\{i \mid Di=\min(Dj)\} \qquad (3)$$

In Equation3, the M is number of class and C is class index has a minimum distance that *k-th* input trajectory. We take the vigilance test for selected winner neuron. Vigilance means constant, which control similarity between a certain input value and connected weight. If distance between winner neuron and input trajectory is greater than vigilance, then output neuron is inactivated and other neuron is selected.

3. 3 Hypothesis-testing

Winner neuron satisfies another condition in ART. That is hypothesis-testing condition. This is caused the following problem. Though Euclidean distance between two trajectories in high dimension is very closed, the similarity among every input element is not guaranteed because difference in every element is not uniformed. So if the difference between every elements of input pattern and top-down expectation, weight in F_1-F_2 layers, is greater than threshold, then we decide that activated prototype of output pattern is different input pattern, and inactivate output neuron. If computed V_i is greater than threshold, then search another neuron. Also, if the neuron satisfies vigilance condition and hypothesis-testing condition, which is not in output neurons, then it is find new neuron.

$$V_i=\sum_{j=1}^{j=T}|Wx_{ij}-x(j)|+\sum_{j=1}^{j=T}|Wy_{ij}-yx(j)| \qquad (4)$$

where, Wx_{ij} is x weight between ith class and jth frame and Wy_{ij} is y weight between ith class and jth frame.

3.4 Learning

Equation 5 is mathematical form of weight learning process between input neuron and winner neuron. In this Equation, convergence velocity is decreased with time. This is done by error between predictive value and actual value of the velocity; divide with number of cluster patterns. Predictive velocity vector follows.

$$Wx_{cj}(k) = Wx_{cj}(1) + \sum_{i=2}^{k-1} \eta \times \frac{x_i(j) - w_{xc}(i)}{count\ (w_c(i))}$$
$$Wy_{cj}(k) = Wy_{cj}(1) + \sum_{i=2}^{k-1} \eta \times \frac{y_i(j) - w_{yc}(i)}{count\ (w_c(i))} \quad (5)$$
$(3 \le k \le datasize)$, $(1 \le j \le M)$

where, η is learning rate which have $0 < \eta \le 1$ and $count(W_c(i))$ is a number of winner neuron pattern.

$$Wx_{c,pred(k)} = x_1(T+1) + \sum_{i=2}^{i=k-1} \eta \times \frac{x_i(T+1) - Wx_{c,pred}(j)}{count\,(W_c(j))}$$
$$Wy_{c,pred(k)} = y_1(T+1) + \sum_{i=2}^{i=k-1} \eta \times \frac{y_i(T+1) - Wy_{c,pred}(j)}{count\,(W_c(j))} \quad (6)$$
$(3 \le k \le datasize)$

Trained weight by Equation 5 operating prototype of class, and trained predictive velocity vector by Equation 6 is saved in related output vector, then it used predictive velocity of next frame when related input pattern is coming.

3.5 Cluster controlling

Existing ART NN, the vigilance is fixed, is not adaptive when trajectory numbers of class category vary. This makes the class which contain many similar trajectories has the same radius of small similar trajectories, so class controlling is needed. In this paper, radius of class is controlled by making category of class vary adaptively to number of trajectory in class, which make vigilance to variable.

$$\rho_c(k+1) = \rho_c(k) + \frac{\rho_c(k)}{\mu}$$

for $\forall_{i=c}$, (7)

$$\rho_i(k+1) = \rho_i(k) - \frac{\rho_i(k)}{k \times \mu}$$

where, μ is number of pattern which are included corresponding cluster and $\rho_i(k)$ is the vigilance *i-th* neuron when *k-th* pattern input is coming

In this paper, we proposed simple control methods that vary vigilance, but this method is not guaranteed optimal clustering because we assumed that two trajectories have the similar path, have similar location in next frame. If we want to predict without this assumption, hierarchical clustering method is needed[7].

4. Stability evaluation of proposed neural network

The important issue in unsupervised neural network is stability and plasticity dilemma. In competitive and Grossberg neural network, this caused by forgetting past input because of recent trained inputs[6]. ART is contrived by Grossberg and Gail Carpenter and overcomes stability and plasticity dilemma. Figure 8 shows neural network converges, so error decreases. This neural network has stability. The result in figure 8 represents the values when 74 sample data with 6 frames and 2 vigilances are trained after 4 iterations.

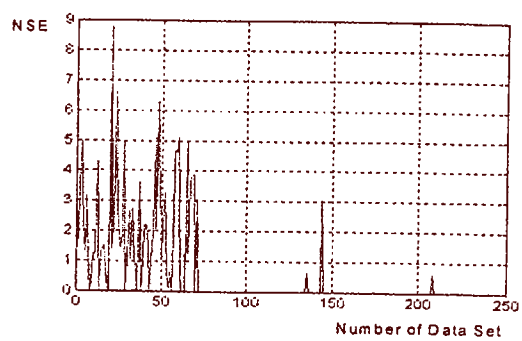

Figure 8 : Stability estimation of predictive ART

5. Experimental result and Analysis

The table 1 shows that we presented the error for recall after the 10-times iteration learning based on the 74 sets of getting eye area using to trace eyes. The formula for calculating error is presented in Equation. 8 that produced using to divide the number of total input patterns into the prediction speed of all input patterns and Euclidean distance between real speeds and predict. In the parenthesis, we presented the number of producing classes after learning. In case that vigilance is 1 and embedding dimension is greater that or equal to 6, error became zero. And this case is understood for good prediction performance because the classes for each data are made up in case of 69 classes for 74 data. In the result according to the table 1, the smaller the value of vigilance and the many previous frame-numbers used prediction, the better the result. The result is variable according to experiment environment as error measure method and data. Because of coincidence the experiment environment, this paper didn't use as comparing data, but we referenced the result of performance estimation of prediction methods operating in Santa Fe.

$$Error = \frac{\sum_{j=1}^{j=k} \sqrt{(x_i(pred) - Wx_{c,pred}(j))^2 + (y_i(pred) - Wy_{c,pred}(j))^2}}{K} \quad (8)$$

Where, K is number of trajectory .

Table1 : Prediction result of moving in Predictive ART

Vigilan	1	2	3	4	5	6	Time (sec/image)
5	0.02250	0.17838	0.57757	0.796503	0.91076	1.16345	0.0081
6	0.00000	0.00856	0.06841	0.168663	0.29164	0.31988	0.0108
7	0.00000	0.00856	0.00878	0.012853	0.05933	0.16187	0.0149
8	0.00000	0.00000	0.01223	0.016523	0.02000	0.02314	0.0189

6. Conclusion

In this paper, we proposed predicting algorithm, based on the predictive ART NN, for the eye tracking system. The Predictive ART of predictive module was adapted at changing the existing ART2 neural network and we operated the unsupervised on-line real-time learning method and so input the moving locus gathered in tracking module and predicted the moving in the next frame. The predictive module is available at the verification module and to reduce the detective area and to discriminate easily success or failure of traced eye location. Later, if we made the enhanced prediction module by merge the tracking module[4] and the prediction module, we will prevent the detecting the failure generating successively based on the wrong detecting template and expect to improve the system performance.

The implementation system has the meaning as tracing the moving which is important information in the image sequences. And it can be applied to the servo control focusing the meeting attendants on the center of camera at the remote meeting or in the intelligent room, or in the security of the shops or banks.

Reference

[1] D.J Kim, "Forecasting Time Series with Genetic Fuzzy Predictor Ensemble", Journal of KISS(B), Software and Applications, vol. 24, no. 2, 1997.2, pp. 193-206

[2] Andreas S. Weigend, Neil A, Gershenfeld, "Results of the Time series Prediction Competition at the Santa Fe Institute", World Congress Neural Networks vol. 4, 1993, pp. 663-670

[3] Jose C. Principle, Jyh-Ming Kuo, "Dynamic Modeling of Chaotic Time Series with Neural Networks", NIPS, 1995

[4] Peter T. Kazlas, Andres S. Weigend, "Direct Multi-Step Time Series Prediction Using TD", NIPS-7, pp 772-782

[5] MH Kim, JH Ra, KI Kang, "Tracking the eye trajectory in dynamic image using adaptive template matching", conf. on KIPS, April, 1998

[6] Asriel U. Levin, "Predicting with Feed-forward Networks", World Congress on Neural Networks vol. 3, 1993, pp. 385-388

[7] G. Bartfai, "Hierarchical Clustering with ART Neural Networks"