

Fast and Robust Method for Dynamic Gesture Recognition Using Hermite Neural Network

Wensheng Li, Chunjian Deng

Zhongshan Institute, University of Electronic Science and Technology of China, Zhongshan, China

Email: lws7166@126.com

Abstract—Due to its shortcomings such as slow convergence rate and low recognition accuracy, the traditional BP neural networks perform poorly in dynamic gesture recognition, especially for online gesture training. In this paper, a novel adaptive Hermite neural networks algorithm for dynamic gesture recognition was proposed. At first, a three-layer feed-forward neural network, of which hidden layer neurons are activated by a group of Hermite orthogonal polynomial functions was constructed. Based on its special structure, a method to determine the network weights directly was introduced, and a novel algorithm to determine the optimal number of hidden nodes adaptively was proposed. Then a rapid method of fingertips tracking was put forward to get the trajectory of dynamic gesture. At last, dynamic gesture was recognized through the trained Hermite neural network. Experiment results show that Hermite neural network can enhance the speed and precision of network training, improve the learning speed and recognition accuracy, and has good robustness and generalization ability.

Index Terms—Hermite neural network, adaptive structure determination, trajectory of gesture, dynamic gesture recognition

I. INTRODUCTION

Gesture recognition can provide a more natural and intuitive method for human-computer interaction compared with keyboard and mouse[1]. Dynamic gesture recognition based on machine vision captures hand gestures with a camera, and determines the interaction semantics of the gesture by calculating the correlation coefficient between the trajectory of gestures and the given templates[2-5].

Dynamic gesture recognition system is a complex nonlinear dynamic system, of which the mapping function between input(fingertip trajectory) and output(gesture category) is difficult to determine. In order to implement dynamic gesture recognition, we must produce an effective solution for the complex nonlinear system identification first, that is, determine the model structure and model parameters for dynamic gesture recognition system through a set of experimental data[6].

Due to its parallelism, adaptability, and approximation ability for nonlinear systems, neural network is widely used in fields such as identification and control of nonlinear system and pattern recognition. Until now, most of the neural networks algorithms for system

identification and pattern recognition are based on multilayer feed-forward networks such as multilayer perceptron(MLP) trained with BP algorithm[7].

But BP algorithm has some inherent shortcomings such as slow convergence, danger of over fitting and lack of theoretical guidance for the determination of hidden node number, etc. Therefore, many improved algorithms, which can be divided into two categories, have been proposed [7]: The first is based on the improvement of the standard gradient descent method (such as the conjugate gradient method, etc.); the second is based on numerical optimization (such as LM method). However, most of these algorithms can not meet the situation of online gesture training for different users or new gestures because they focus on the improvement of iterative rules in network training, and can not avoid the lengthy iterative training completely.

Subsequently, some researchers began to apply orthogonal polynomial functions to construct neural networks[8-12]. It was investigated that multivariate polynomial functions with n-order can be approximated with any precision through a three-layer feed-forward neural network, and the number of hidden nodes for the constructed networks only depends on the order and dimension of approximated polynomial[8]. Based on this, [9-12] proposed feed-forward neural networks activated with Hermite or Chebyshev orthogonal basis for single-input single-output(SISO) case. A pseudo-inverse based weight-direct-determination method was derived for the optimal weights from hidden layer to output layer and an adaptive algorithm was also presented for the optimal number of hidden-layer neurons according to the precision requirement[12]. All of these greatly improved the speed of network training. But unfortunately, All of these method are suitable for approximation of functions of one variable, but not suitable for dynamic gesture recognition which is multi-input, multiple output(MIMO).

In this paper, we propose a MIMO Hermit neural network of which the hidden neurons are activated by a group of Hermite orthogonal polynomial functions and apply it to dynamic gesture recognition. Simulation results show that Hermite neural network can enhance the speed and precision of network training, improve the learning speed and identification accuracy of gesture recognition and has good robustness and generalization ability.

II. MODEL OF HERMITE NEURAL NETWORK

A. Hermite orthogonal polynomials

At first, we introduce the definition about Hermite orthogonal polynomial and its related properties.

Definition 1. The Hermite polynomials could be defined as follows:

$$\varphi_{i-1}(x) = (-1)^{i-1} e^{x^2} \frac{d^{i-1}(e^{-x^2})}{dx^{i-1}}, \quad 0 \leq x < +\infty \quad (1)$$

We can easily see that the Hermite polynomials defined above can be recursively generated by the following formula:

$$\begin{aligned} \varphi_0(x) &= 1 \\ \varphi_1(x) &= 2x \\ \varphi_i(x) &= (2x\varphi_{i-1}(x) - 2(i-1)\varphi_{i-2}(x)), \quad i = 2, 3, 4, \dots \end{aligned} \quad (2)$$

In particular, for any $i > 0$, we notice that $\varphi_i(x)$ is an algebraic polynomial of degree $i-1$ with respect to x .

Using well-known trigonometric relations, we have:

$$\int_{-\infty}^{\infty} \varphi_i(x)\varphi_j(x)e^{-x^2} dx = \begin{cases} 0 & i \neq j \\ 2^i i! \sqrt{\pi} & i = j \end{cases} \quad (3)$$

which shows the orthogonality of the Hermite polynomials with respect to weight function $\rho(x) = e^{-x^2}$ over interval $[-\infty, \infty]$.

Based on polynomial interpolation and approximation theory, we can use generalized polynomial $\varphi(x)$ to interpolate and approximate an unknown nonlinear function $\Phi(x)$, and for the approximation of $\varphi(x)$ to $\Phi(x)$, we have the following definition and theorem.

Definition 2 (Least Square Approximation) Suppose $\Phi(x)$ and all functions in $\{\varphi_j(x)\}_{j=0}^{h-1}$ is continuous over $[a, b]$, and $\{\varphi_j(x)\}_{j=0}^{h-1}$ are linearly independent functions, $\rho(x)$ is a weight function. Determine the coefficients w_0, w_1, \dots, w_{h-1} for generalized polynomial $\varphi(x) = \sum_{j=0}^{h-1} w_j \varphi_j(x)$ to make $\int_a^b [\Phi(x) - \varphi(x)]^2 \rho(x) dx$ be the minimum, the resulting function $\varphi(x)$ is called least square approximation function for $\Phi(x)$ in $[a, b]$ with respect to weight function $\rho(x)$.

Theorem 1 (Existence and uniqueness of least square approximation) Suppose $\Phi(x)$ is continuous over $[a, b]$, then its least square approximation $\varphi(x)$ must exist and be unique.

B. Hermite Neural Network Model

The model of Hermite neural network is shown in Figure 1. The model consists of input layer, hidden layer and output layer, where the input layer has n nodes with $X = (x_1, x_2, \dots, x_n)^T \in R^n$ as the input vector, the output layer has m nodes with $Y = (y_1, y_2, \dots, y_m)^T \in R^m$

as the output vector. There are h hidden neurons which adopt a set of Hermite polynomials as activation functions. $W \in R^{h \times m}$ is the weight matrix from hidden layer to output layer, where w_{jk} is the connecting weight from j^{th} node in hidden layer to k^{th} node in output layer. Σ in the output layer nodes means that linear activation functions are used in the output layer neurons.

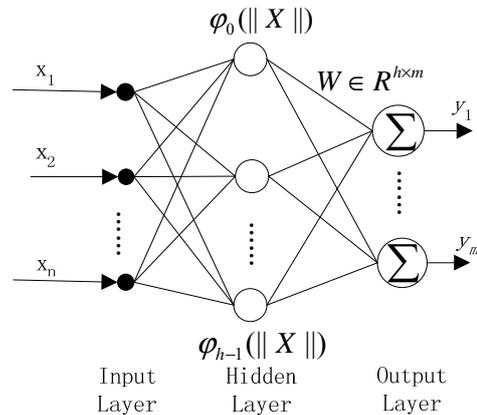


Figure.1. The model of Hermite neural network

Assuming there are P input vectors (X^1, X^2, \dots, X^P) as learning samples and P output vectors (D^1, D^2, \dots, D^P) as instructor signals. In order to build the functional mapping between input and output, Hermite Orthogonal basis functions are used to interpolate and approximate. The interpolation functions are defined as a linear combination of Hermite basis functions:

$$F_k(X) = \sum_{j=1}^h w_{jk} \varphi_j(\|X\|) \quad k = 1, 2, \dots, m$$

(4) where $\|X\|$ is a linear mapping $R^n \rightarrow R$.

C. Determination of Weights

Now, we discuss the iterative method and direct-determination method to obtain the weights of the network.

Theorem 2 The weights of Hermite neural network as shown in Figure 1 can be derived from the iterative formula as follow:

$$W(k+1) = W(k) - \eta \Phi^T(\Phi W(k) - D) \quad (5)$$

where

$$\Phi = \begin{bmatrix} \varphi_0(\|X^1\|) & \varphi_1(\|X^1\|) & \dots & \varphi_{h-1}(\|X^1\|) \\ \varphi_0(\|X^2\|) & \varphi_1(\|X^2\|) & \dots & \varphi_{h-1}(\|X^2\|) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(\|X^P\|) & \varphi_1(\|X^P\|) & \dots & \varphi_{h-1}(\|X^P\|) \end{bmatrix} \in R^{P \times h}$$

and $k = 0, 1, 2, \dots$, learning rate η should be small enough to ensure iterative convergence.

Just like the traditional BP neural network training methods, the weight iteration formula must get the optimal weights through lengthy iterative training. Benefit from the special structure of Hermite neural network, we can determine optimal weights of the neural network directly; i.e., the following theorem.

Theorem 3 (direct-weights-determination) The neural-weights of Hermite neural network could be obtained directly as formula:

$$W = (\Phi^T \Phi)^{-1} \Phi^T D \quad (6)$$

Where $\text{pinv}(\Phi) = (\Phi^T \Phi)^{-1} \Phi^T$ is the pseudo inverse of Φ .

Proof: According to Theorem 2, that is, $W(k+1) = W(k) - \eta \Phi^T (\Phi W(k) - D)$. Since the learning rate $\eta > 0$, so when the network reaches steady state (i.e., k is large enough), we will have $W(k+1) = W(k) = W$, which satisfy $\Phi^T (\Phi W - D) = 0$. Then we can use pseudo-inverse matrix and formula (6) to determine the optimal weights directly.

The Pseudo-inverse based direct-weight-determination for Hermite neural network avoids the lengthy iterative training, but can also achieve least square approximation for the mapping function between input and output.

D. Adaptive Algorithm for Optimal Hidden-Layer

As we know, determination of the number of hidden neurons for neural network hidden layer is in the lack of theoretical guidance, and there is still no good method to determine network structure. Benefiting from the special structure of Hermite neural network, the optimal number of hidden neurons can be determined adaptively.

Given the target precision $\epsilon > 0$ for the network, we can determine the minimum of hidden nodes to meet the requirement of precision adaptively as follows:

1) Initialize the number of hidden layer nodes as $h = \sqrt{n + m} + 1$, and set the target precision, such as $\epsilon = 0.005$;

2) Set the maximum of hidden nodes, such as $MaxHideNode = \sqrt{n + m} + 20$;

3) If $h > MaxHideNode$, it shows that the target precision can not be reached within $MaxHideNode$, and terminate the algorithm; otherwise, jump to step 4);

4) Use weights-direct-determination method to solve network optimal weights and then compute the current precision MSE(Mean Square Error);

5) if the $MSE \leq \epsilon$, it shows that the minimum of hidden layer nodes as h meeting the target precision requirement has been found, and terminate the algorithm. Otherwise set $h = h + 1$, jump to step 3).

The above algorithms can determine the minimum of hidden nodes speedily, and obtain the most simplified model structure of the neural network.

III. ACQUISITION AND RECOGNITION OF DYNAMIC GESTURE

A. Extraction of gesture trajectory

To implement dynamic gesture recognition based on machine vision, the first work to do is to detect and track fingertips in real time, and obtain the gesture trajectory.

To simplify the tracking algorithm and improve the tracking efficiency, we propose an efficient method to detect and track fingertips based HSV color space as follows.

1) In order to reduce the effect of illumination, we employ H and S of HSV for learning the probability distribution of fingertip colors. A 2D color histogram is computed as the result of on-line training of fingertip colors. The 2D color histogram can be used as the basis of target fingertip detection and tracking;

2) Capture a video image, and convert the image from RGB to HSV;

3) Compute color probability distribution $I(x, y)$ of the target through back-projection on the 2D color histogram;

4) Transform $I(x, y)$ into a binary image $B(x, y)$:

$$B(x, y) = \begin{cases} 0 & I(x, y) < p \\ 1 & I(x, y) \geq p \end{cases} \quad (7)$$

where p is a given threshold.

5) Determinate the target region of fingertips through edge detection;

6) Determinate the centroid positions (x_c, y_c) of target as follows:

$$M_{00} = \sum_x \sum_y B(x, y)$$

$$M_{10} = \sum_x \sum_y xB(x, y)$$

$$M_{01} = \sum_x \sum_y yB(x, y)$$

$$x_c = M_{10} / M_{00}, y_c = M_{01} / M_{00} \quad (8)$$

7) Track the centroids of the fingertips through MDF(Minimum Distance First), and go to step 2).

B. Model of dynamic gesture

Dynamic gestures are divided into two categories: four-finger-gesture (gesture with both hands) and two-finger-gesture (one hand gesture). For the four-finger-gestures such as zoom in, zoom out, rotation, etc, we can implement gesture recognition by analyzing the relative position of fingers of the left and right hand. In this paper, we focus on the recognition of two-finger-gesture.

A two-finger-gesture is drawn by two fingers (thumb and index finger) of one hand. It starts when detecting the two fingertips is getting closer and the distance between them is less than the given threshold, and ends when detecting the two fingertips is getting farther away and the distance between them is greater than the given threshold. Two-finger-gestures can be defined by the user. In this paper 16 gestures were defined as shown in Table 1. The rough dot in the stroke indicates the start of a dynamic gesture.

TABLE I
The List of two-finger dynamic gestures

Gesture	Interaction semantics	Gesture stroke
Digits	Input digital number	0 1 2 3 4 5 6 7 8 9
Confirm	Confirm/Dial	✓
Cancel	Cancel/Cancel the call	✗
Forward	Forward/Scroll left	→
Backup	Backward/Scroll right	←
Pagedown	Pagedown/Next sheet	↘
PageUp	Pageup/Previous sheet	↙

So far, we have obtained a set of point for a two-finger-gesture (the mid-point of two fingertips)by fingertip tracking. If the set has more than 15 points, find the two points with minimum span, and replace them with their midpoint, repeat the operation until the number of the point in the set is 15. Then obtain 14 vectors by calculating the difference between two adjacent points and doing normalization, finally we can get the vector combination (28 components) of a two-finger-gesture. For example, the corresponding vector for number “6” gesture strokes as shown in figure 2 is:

{-0.93,0.36,-0.72,0.69,-0.47,0.88,-0.40,0.92,-0.26,0.97,
-0.07,1.00,0.11,0.99,0.65,0.76,0.98,0.18,0.86,-0.51,0.46,
-0.89,-0.49,-0.87,-0.98,-0.18,-1.00,0.00}.



Figure.2. The trajectory of gesture “6” and its sample points

C. Dynamic gesture recognition based on Hermite neural network

1) The design of Hermite neural network structure

The number of input nodes is set to 28, which is corresponding to the 28 components of gesture vector. The number of hidden nodes is set to 12 by means of the adaptive method discussed above to meet the requirement of precision 0.005. The number of output nodes must be compatible with the types of dynamic gestures to be recognized. As there are 16 types of gestures to be identified, we set the number of output nodes to 16.

2) Neural network training

We use 16X5 training samples for 16 types of gestures, that is, we use five training samples for each gesture.

For the input vectors of training samples, we can be obtained by the aforementioned method, and for the expected output vectors of training samples (teacher signal), as the gestures are divided into 16 types, so we use 16 unit vectors with length of 16 to represent these 16 gestures. An unit vector of which the mth element is 1 and the remaining elements are all 0 can be seen as the output of the mth gesture. For example, the expected output of the first gesture “0” would be (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), and expected output of the fourteenth gestures “backward” would be (0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0).

The optical weight matrix W for Hermite neural network can be obtained through equation (6), and the trained Hermite neural network can be used for dynamic gesture recognition.

3) Identification of dynamic gesture

When the vector corresponding the dynamic gesture to be tested is input into the trained Hermite neural network, the gesture can be identified through the output of the network: the closer the output of the network approaching the predefined output of the Mth gesture, the more likely the input gesture is the Mth gesture.

For example, the output of the network corresponding a input gesture is (0.01,0.00,0.00,0.00,0.01,0.00,0.00,0.00,0.00,0.00,0.99,0.00,0.00,0.00,0.01,0.01), the maximum component is the 11th components, so we can determine that it is the gesture of No. 11, i.e. the gesture of “confirm”.

If the largest component of output vectors is less than a given threshold (e.g. 0.5), we can determine that the gesture is not a pre-defined gestures, that means gesture recognition fails.

IV. EXPERIMENT RESULTS AND ANALYSIS

In order to verify he feasibility and effectiveness of dynamic gesture recognition based on Hermite neural network, both traditional BP network which are trained by GDJ and LM algorithm and Hermite network of which weights are determined directly are used for

dynamic gesture training and recognition for same samples.

A. Network training test

It can be seen from Table 2 that Hermite network is superior to BP neural network not only in training speed, but also in network precision. For Hermite network, the weights can be determined directly and the training time is

shortened to 0.57s, while for BP network trained by GDX and LM algorithm used in, the average training times were 15.3s and 6.5s. At the same time, The MSE of BP network trained by GDX and LM (setting the maximum of iterations to 3000 and target network performance to 0.005) are 0.0049925 and 0.0042041, while the MSE of Hermite networks is 0.0034623.

Table II
Comparison of training results using Hermite network and traditional BP network

Method	Training process	runtime(s)	MSE
BP(GDX)	2213 iterations	15.30	0.0049925
BP(LM)	197 iterations	6.50	0.0042041
Hermite Network	One step	0.57	0.0034623

B. Gesture recognition test

It can be seen from test result that Hermite neural network has novel robustness and generalization ability for dynamic gesture recognition to tolerate a certain range of gesture input differences. Part of the results are shown in Figure 3.

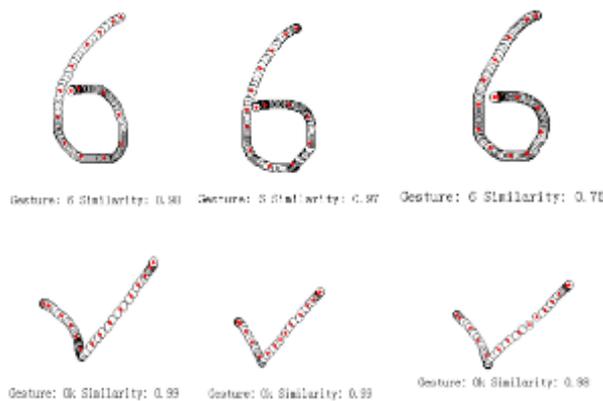


Figure.3. Robustness of Hermite network to gesture input

For each dynamic gesture, 200 samples were tested by Hermite neural network and BP neural networks trained by GDX and LM methods. The average accuracy of gesture recognition with BP trained by GDX and LM were 89.7% and 91.9%, while the average accuracy with Hermite neural network reached an average of 94.7%.

V. CONCLUSION

This paper construct a MIMO Hermite neural network for dynamic gesture recognition. Due to its direct-weights-determination and the adaptive algorithm to determine the number of hidden r nodes, it has overcome such shortcomings as slow convergence and low accuracy which exist in traditional BP networks, and greatly improved the training speed, and can meet the user's requirement of performing online gesture training. At the same time, it improves the network accuracy and generalization ability significantly.

ACKNOWLEDGMENT

This work was supported by a grant from Natural Science Foundation of Guangdong, China (8152840301000009); Science and Technology Planning Project of Guangdong, China (2009B030803031).

REFERENCES

- [1] Todd C. Alexander, Hassan S. Ahmed, Georgios C. Anagnostopoulos, "An Open Source Framework for Real-Time, Incremental, Static and Dynamic Hand Gesture Learning and Recognition". *Human-Computer Interaction, Part II, HCII 2009, LNCS 5611*, 2009: 123-130.
- [2] Jos'e M. Moya, Ainhoa Montero de Espinosa, Alvaro Araujo, Juan-Mariano de Goyeneche, Juan Carlos Vallejo, "Low-Cost Gesture-Based Interaction for Intelligent Environments". *IWANN 2009, Part II, LNCS 5518*, 2009: 752-755.
- [3] Kaustubh, Sumantra, "Hand gesture modelling and recognition involving changing shapes and trajectories, using a Predictive EigenTracker". *Pattern Recognition Letters*, 2007(28): 329-334.
- [4] Hardy Francke, Javier Ruiz-del-Solar, and Rodrigo Verschae, "Real-Time Hand Gesture Detection and Recognition Using Boosted Classifiers and Active Learning". *PSIVT 2007, LNCS 4872*, 2007: 533-547.
- [5] Jiyoung Park and Juneho Yi, "Efficient Fingertip Tracking and Mouse Pointer Control for a Human Mouse". *ICVS 2003, LNCS 2626*, 2003: 88-97.
- [6] Pang Zhonghua, Cui Hong, *System Identification and Adaptive Control with MATLAB Simulation*. Beijing: BeiHang University Press, 2009.
- [7] Martin T. Hagan, Howard B. Demuth, Mark H. Beale. *Neural Network Design*. China Machine Press, 2002.
- [8] Wu Xiaojun, Wang Shitong, Yang Jingyu, Cao Qiyang, "The Study on the Orthogonal Polynomials-based Neural Networks and Its Properties". *Computer Engineering and Application*, 2002, 38(9):25-26.
- [9] Jagdish C. Patra, Alex C. Kot. "Nonlinear Dynamic System Identification Using Chebyshev Functional Link Artificial Neural Networks". *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS*, 2002, 32(4):505-511.
- [10] Mu Li, Yigang He, "Nonlinear system identification using adaptive Chebyshev neural networks". *Intelligent Computing and Intelligent Systems (ICIS)*, 2010: 243- 247.
- [11] ZHANG Yunong, CHEN Yulong, JIANG Xiaohua, ZENG Qingdan, ZOU Ajin, "Weights-directly-determined and Structure adaptively tuned Neural Network Based on

Chebyshev Basis Functions". *Computer Science*, 2009, 36(6): 210-213.

- [12] ZHANG Yunong, CHEN Yangwen, YI Chenfu, LI Wei, "Feed-Forward Neural Network Activated with Hermite Orthogonal Polynomials and Its Weights-Determination Method". *Journal of Gansu Sciences*, 2008, 20(1): 82-86.

Wensheng Li, born in 1966, M.S. associate professor. His current research interest includes embedded software, biometrics, multimedia processing and communication.

Chunjian Deng, born in 1980, Ph.D. associate professor. His current research interest includes His research interest includes information communication an display.