Handwritten Chinese Trajectories Prediction with an Improved Flat Functional-link Neural Networks and Kalman Filter^{*}

Duanduan Yang¹, Lianwen Jin², Lixin Zhen³, Jiancheng Huang⁴

(1,2 South China University of Technology, Guangzhou 510640, China;

3,4 Motorola China Research Center, Shanghai 200002, China)

Abstract: This paper introduces the application of the Flat Functional-link Neural Network (FFNN) to predict handwritten Chinese moving trajectories. To solve the prediction problem of a non-stationary time series, conventional neural networks need a lot of time and samples to train, but FFNN can solve this problem very well. Considering the structure of Chinese characters, the paper makes some improvement for FFNN and chooses the appropriate training samples, and promising experimental results have been obtained. Furthermore a comparison is performed between the predictions of the Flat NN and a Kalman filter. Experiments suggest that the improved FFNN predictor works better for the prediction of trajectories of handwritten Chinese characters.

Key words: Time Series Prediction; Neural Networks; Kalman Filter

1. Introduction

With the computer becoming necessary for our lives, how to communicate with computers conveniently, naturally and adequately has become a research focus^[1]. It is very attractive that we can communicate with computers by our favorite means, such as handwriting. A kind of such system named a camera-based human-computer interface (HCI) system for

handwriting acquisition has been presented by Mirio.E.Munich and Pietro Perona^[2], which can be illustrated as Fig.1^[2].

One critical factor in such kind of HCI is the ability of the machine to identify the position of the pen tip quickly and efficiently. In order to perform real-time acquisition and tracking, a predictor is often necessary to be used to provide the prediction of the pen tip position for reducing the searching area and saving computing time.

If the two-dimensional position of pen tip on the i th frame is denoted as p_i that can be localized by Pen Tip Detector and the prediction of pen tip position on the $(i+1)$ th frame is denoted as $\hat{p}_{i+1}$, the function of predictor can be denoted as the following equation:

$$\hat{p}_{i+1} = f(p_i, p_{i-1}, \dots, p_{i-k+1}) \quad (1)$$

Where k denotes prediction step-length, f is a prediction function, p_i and $\hat{p}_{i+1}$ are given by

$$p_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \text{and} \quad \hat{p}_{i+1} = \begin{bmatrix} \hat{x}_{i+1} \\ \hat{y}_{i+1} \end{bmatrix}, \quad \text{where} \quad x_i, y_i$$

denote the pen tip position in X, Y ordinate respectively on the i th frame.

^{*} **Acknowledgement:** This project is supported by Motorola Human Interface Lab Research Foundation(No.D84110), NSFC(No.60275005), GDNSF(No.2003C50101, 011611).

Duanduan Yang, PhD candidate; main research fields: image processing, pattern recognition, neural network; E-mail: ddyang@scut.edu.cn.

Lianwen Jin, PhD, professor; main research fields: character recognition, image processing, neural network, computer vision, human-machine interface.

Lixin Zhen, researcher of Motorola China Research Center.
Jiancheng Huang, researcher of Motorola China Research Center.

Fig.1 A Typical Camera Based HCI System

Kalman filter is a set of mathematical equations that provides an efficient computational solution of the least squares method. As a recursive estimate scheme, Kalman filter is powerful in estimating the future states^[3] and suitable as a linear predictor^[2]. However, the non-linear problem calls for the use of non-linear models. Neural Networks (NNs) have been frequently used to predict all sorts of functions and time series in many real world applications as non-linear methods, such as control, medical treatment, bankruptcy estimation and fire alarm forecast^[4,5]. Some of them, such as Multi-layer Perception (MLP), require complex non-linear learning algorithms (such as BP), which may lead to problems such as slow convergence and multi-minimum error surface^[6-8]. Other NN models, such as Recurrent Neural Networks (RNNs), also require some complex learning algorithms to update the network parameters, like the Real-Time Recurrent Learning algorithm (RTRL)^[9] whose computation complexity is $O(N^4)$, where N is the total number of nodes of the RNNs^[10]. Opposed to these networks, Flat Functional-linked Neural Networks (FFNNs) which take over the drawback of a long and complex learning process with the advantage of learning only one set of weights^[11] are able to achieve “one shot” training^[12]. FFNN can be trained using the rapid learning and dynamic stepwise updating algorithm^[11]. As a linear updating method, it can real-time modify the weight matrix rapidly to ensure the precisely predicting a time series.

In this paper, some improvements have been made

in applying FFNN for predicting handwriting Chinese character trajectory. A comparison of the performance of FFNN and a Kalman filter has been carried out. The experimental results show that the improved FFNN achieves better prediction performance than that of the original FFNN and Kalman filter.

2. A Brief Introduction of FFNN

FFNN has been discussed by Y. H. Pao in references [13] and [14], the structure of which is illustrated in Fig.2.

Fig.2 A Flat Functional-link Neural Network

Fig.3 A Linear Formulation of Functional-link Network

In general, the output-input function of FFNN is given by

$$Y = [X|f(XW_h + \mathbf{b}_h)]W \quad (2)$$

where Y is the output matrix, X is the input matrix, $\mathbf{b}_h$ is the bias, f is a nonlinear function such as sigmoid function, W_h is the weight matrix between the input and the enhancement nodes that is randomly generated, $|$ denotes a combination of two matrix. W is the weight matrix of the network. If an expand-input matrix is defined as

$$A = [X|f(XW_h + \mathbf{b}_h)] \quad (3)$$

then FFNN can be illustrated by Fig.3 and the matrix W can be regarded as the weighting matrix between expand-input layer and output layer.

Let X denotes a vector composed of $\{p_i, p_{i-1}, \dots, p_{i-k+1}\}$.

$$X = [x_i, y_i, x_{i-1}, y_{i-1}, \dots, x_{i-k+1}, y_{i-k+1}] \quad (4)$$

and Y denotes a prediction of p_{i+1} :

$$Y = [\hat{x}_{i+1}, \hat{y}_{i+1}] \quad (5)$$

then FFNN could be treated as a predictor applied in the HCI mentioned in the introduction section. More details of FFNN can be found in the reference [11]

2.1 Learning Algorithm of FFNN

If the N moving trajectories points of a handwriting character are treated as training samples, input matrix X can be constructed by building a $(N-k) \times (2k)$ matrix whose i th rows consist of $[x_i, y_i, x_{i+1}, y_{i+1}, \dots, x_{i+k-1}, y_{i+k-1}]$, and output matrix will be produced by

$$\begin{bmatrix} x_k, x_{k+1}, \dots, x_N \\ y_k, y_{k+1}, \dots, y_N \end{bmatrix}^T. \quad \text{If more than a}$$

handwriting character are treated as training samples, the input matrix or output matrix could be denoted as the combination of $\{X_1, X_2, \dots, X_L\}$ or $\{Y_1, Y_2, \dots, Y_L\}$

$$X = [X_1^T | X_2^T | \dots | X_L^T]^T \quad (6)$$

$$Y = [Y_1^T | Y_2^T | \dots | Y_L^T]^T \quad (7)$$

where X_i , Y_i are respectively the input matrix and the output matrix from the i th handwriting character.

After the input and output have been constructed, the learning algorithm of FFNN can be described as the following steps:

Step 1 Initialize the expand input matrix A , $A_0 = X$

Step 2 Minimize $\|A_n W_n - Y\|_2$, get W_n

Step 3 If mean-squared error criterion is not satisfied, add an enhancement node, update the new expand input matrix $A_{n+1} = A_n | a$, (adding an enhancement node is equivalence to adding a new column to the expanding matrix) and go to Step 2; otherwise, stop.

According to the Rapid Learning algorithm of FFNN^[6], W matrix can be updated

$$\text{using } W_{n+1} = \begin{bmatrix} W_n - db^T Y \\ b^T Y \end{bmatrix}$$

$$\text{where } b = \begin{cases} c^+ & \text{if } \dots c \neq 0 \\ (1 + d^T d)^{-1} d^T A_n^+ & \text{if } \dots c = 0 \end{cases},$$

$$c = a - A_n d, \text{ and } d = A_n^+ a.$$

As we can see, the weight matrix W of FFNN can be rapidly updated when the Rapid Learning algorithm is applied.

2.2 Prediction Algorithm of FFNN

When a new input pattern is given to the network, the weight matrix should be updated. The prediction algorithm of FFNN is briefly described by the following steps.

1) On the i th frame, after the pen tip position p_i is got by the Pen Tip Detector, construct a new pattern $s = [x_i, y_i, x_{i-1}, y_{i-1}, \dots, x_{i-k+1}, y_{i-k+1}]$, through W_h , and get a new expanding pattern

$a^T = [s \mid s \times W_h]$. Update A by the following equation:

$$A_{n+1} \triangleq \left\{ \frac{A_n}{a^T} \right\} \quad (8)$$

2) Predict the target of the new pattern

$$\hat{p}_{i+1} = a^T * W_n \quad (9)$$

3) On the $(i+1)th$ frame searching in the window centered by $\hat{p}_{i+1}$, find the pen tip position p_{i+1} by the Pen Tip detector and update target vector Y by:

$$Y_{n+1} \triangleq \left\{ \frac{Y_n}{p_{i+1}^T} \right\} \quad (10)$$

4) Update W through

$$W_{n+1} = W_n + (p_{i+1}^T - a^T W_n) b; \quad (11)$$

$$\text{where } b = \begin{cases} (c^T)^+ & \text{if } \dots c \neq 0 \\ (1 + dd^T)^{-1} A_n^+ d & \text{if } \dots c = 0 \end{cases},$$

$$c^T = a^T - d A_n, \text{ and } d^T = a^T A^+.$$

5) After above updating process, if mean-squared error criterion is not satisfied, go to the learning algorithm, otherwise process the next frame.

More details of the algorithm can be obtained from the reference [11]. From Equation (11), it can be seen that the matrix W is updated with a simple linear computation when a new pattern is added to the network.

3. Improvement of FFNN be Applied in Handed Writing Trajectories

To apply the Neural Networks to solve time series prediction problems, historic data-information of the time series is usually used to train the Neural Network^[11,15,16]. From experiments we can find that it is not feasible to solve the problem of predicting the hand written Chinese trajectories for the period of writing is short.

On the other hand if we use the trajectories of a handwriting character as train samples, there will be a problem that which character's trajectories should be used, since the vocabulary of handwriting Chinese characters is very large. If we use the trajectories of all Chinese characters, the input matrix will be huge, which leads to cost much more time being involved to train the network and the ill-condition problem^[11].

As it's known, a Chinese character consists of four basic strokes (Fig.4): horizon, vertical, left slope stroke and right slope stroke. From the experiment we have found that using the four trajectories of basic strokes to train the network was more efficient. To ensure the accuracy of predicating the trajectory of pen tip during the stage of pen up, training samples also include the trajectories of some simple curve-strokes like Fig.5.

Fig.4 The Trajectories of Handwriting Horizon Stroke

Fig. 5 The Trajectories of Basic Curves

Fig.6 and Table 1 show the prediction results from

FFNN trained by the historic data or by the moving trajectories of basic handwriting Chinese strokes. In Fig.6 circles represent the real pen tip positions and the asterisks represent the predicted points of real and corresponding predicted is lined by dash line.

Table 1 The Comparison of Different Training Samples
(unit: pixels)

Chinese character	华		大	
Train samples	Historic data	Strokes data	Historic data	Strokes data
Max error	52.42	21.63	40.22	16.80
Average error	20.49	8.20	16.71	6.54

(a) Predicted Results by Training with Historic Data

(b) Predicted Results by Training with the Trajectories of Handwriting the Strokes

Fig.6 The Comparison of Different Training Sample

From Fig.6 (a) it can be seen that about half of trajectories are not predicted which are used as training samples for the conventional FFNN. On the contrary, using the trajectories of writing Chinese strokes to train the network ensures the almost every point can be predicted, and can get much more precise prediction comparing with conventional FFNN method. Table 1 suggests that our method is more

suitable for predicting the trajectories of handwriting Chinese characters.

In the process of FFNN, sometimes too many enhancement nodes are needed to add to the networks. This will cause the small singular value of expand input matrix A that will result in very large value in the weighting matrix. As a result, any noise in the new data will be amplified. To solve the problem, the

following methods are proposed in this paper:

1) During the prediction stage, the trajectories of handwriting strokes are used to train the networks instead of using the trajectories of handwriting Chinese characters. By this way, the number of enhancement nodes of the networks could be reduced to about 8 times by using the trajectories of one character. As we know, probably there are 3,755 daily-used Chinese characters. It is unworkable to use the trajectories of all handwritten Chinese characters to train the network. From experiments it can be found that the network trained by the trajectories of Chinese basic strokes can build a good predictor for handwritten Chinese trajectories since the train samples almost include all the information of handwriting Chinese characters.

2) During the prediction stage, we set the upper bound for enhancement nodes. If the number of enhancement nodes reaches the upper bound, the weight matrix is resetted to the original state just after training. The number of enhancement nodes will also reset to the original number. This method is equivalent to divide the character to several segments, avoiding too many enhancement nodes being added to the network. Table 2 compares the prediction results of setting the upper bound of enhancement nodes with those who are without upper bound. From table2 it can be seen that the predictions are much more accurate after setting the upper bound because it avoids the enhancement nodes increasing ceaselessly.

Table 2 The Comparison with or without Upper Bound of the Enhancement Nodes (unit: pixels)

Chinese character	华		南	
With or without upper bound	Yes	No	Yes	No
Maximal error	21.63	71.08	34.12	109.06
Mean error	8.20	15.62	10.04	16.574

4. The Predictor Based on Kalman Filter

The prediction for the future position of pen tip can be obtained from the state estimations, like Kalman filter^[2,16].

Under the constant speed assumption, let $\{\Delta x_i, \Delta y_i\}$ denote the pixel displacements in x and y direction at the i^{th} frame. Then in $i+1^{th}$ frame the pen tip position is given by

$$x_{i+1} = x_i + \Delta x_i, \quad (13)$$

$$y_{i+1} = y_i + \Delta y_i \quad (14)$$

If we introduce the state and measurement vectors:

$$s_i = \begin{bmatrix} x_i \\ y_i \\ \Delta x_i \\ \Delta y_i \end{bmatrix}, \quad z_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad \text{the Stated equation and}$$

measurement equation can be respectively denoted as

$$s_{i+1} = F s_i + G u_i \quad (15)$$

$$z_{i+1} = H s_i + v_i \quad (16)$$

with

model

parameters

$$F = \begin{bmatrix} 1010 \\ 0101 \\ 0010 \\ 0001 \end{bmatrix}, \quad G = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}, \quad H = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \quad \text{where}$$

u_i and v_i denote two uncorrelated zero mean white process and measurement noises that satisfy

$$E \left(\begin{bmatrix} s_0 \\ u_i \\ v_i \end{bmatrix} \begin{bmatrix} s_0 \\ u_i \\ v_i \end{bmatrix}^T \right) = \begin{bmatrix} \Pi_0 & 00 \\ 0Qd_{ij} & 0 \\ 00Rd_{ij} \end{bmatrix} \quad (17)$$

The covariance matrices are chosen as

$$Q = \begin{bmatrix} 3000 \\ 0300 \\ 0010 \\ 0001 \end{bmatrix}, \quad R = \begin{bmatrix} 30 \\ 03 \end{bmatrix}. \quad \text{These choices are}$$

determined as follows. Assume initially the large uncertainties in X and Y locations, say of 3 order pixels. And the small uncertainties in the displacement, say of 1 order pixel. Then, those values are checked as optimality by the testing of whiteness of the result innovation process of Kalman filter following the

method of Mehra^[17], the choice of Q, R meet Mehra's 95% confidence whiteness test^[16].

If we introduce the following predicted and filterer estimates of state vectors as

$\hat{s}_i$: the estimate of s_i given $\{z_0, z_1, z_2, \dots, z_{i-1}\}$

$\hat{s}_{i|i}$: the estimate of s_i given $\{z_0, z_1, z_2, \dots, z_i\}$

and the corresponding error variance is given by:

$$P_i = E(s_i - \hat{s}_i)(s_i - \hat{s}_i)^T \quad (18)$$

$$P_{i|i} = E(s_i - \hat{s}_{i|i})(s_i - \hat{s}_{i|i})^T \quad (19)$$

The $\{\hat{s}_i, \hat{s}_{i|i}\}$ can be constructed recursively as the following equations^[2]:

$$\hat{s}_{i+1} = F\hat{s}_{i|i}, \quad i \geq 0 \quad (20)$$

$$\hat{s}_{i+1|i+1} = \hat{s}_{i+1} + P_{i+1}H^T R_{e,i+1}^{-1} e_{i+1} \quad (21)$$

where

$$P_i = FP_{i|i}F^T + GQG^T \quad (22)$$

$$R_{e,i+1} = R + HP_{i+1}H^T \quad (23)$$

$$P_{i+1|i+1} = P_{i+1} + P_{i+1}H^T R_{e,i+1}^{-1} HP_{i+1} \quad (24)$$

and with initial conditions

$$\hat{s}_{0|0} = P_{0|0}^{-1}H^T R^{-1}s_0 \quad (25)$$

$$P_{0|0} = (\Pi_0^{-1} + H^T R^{-1}H)^{-1} \quad (26)$$

Equations (20)-(24) are known as the time- and the measurement-updating equations of Kalman filter. Equation (20) is used to get the prediction, while other equations are used to update the corresponding parameters.

5. Prediction Performance of FFNN Against Kalman Filter

A comparison of prediction performance between FFNN and Kalman filter is given in the paper. The errors of prediction are defined as the Euclidian distance between the real values and the prediction values. This paper uses FFNN and Kalman filter to

predict 11 time series data that are caused by the trajectories of 11 handwritten Chinese characters. Table 3 and Fig.7 show the comparison of the performance between Kalman filter and FFNN in the prediction of the trajectories of the same Chinese written by different people. Table 4 and Fig.8 give more results.

As it is seen, FFNN has obtained better performance in the aspect of mean square error, though the maximal error from FFNN is sometime bigger than that from Kalman filter. Whatever, the overall performance of FFNN is much better than that of the Kalman Filter when applied in predicting the trajectories of handwritten Chinese.

Table 3 The Performance of Kalman Filter and FFNN

(Unit: pixel)

Series number (华)	Kalman filter model		FFNN model	
	Maximal error	Average error	Maximal error	Average error
1	21.71	8.63	22.78	5.86
2	21.56	10.48	23.41	7.14
3	22.59	10.29	18.42	6.68
4	47.91	18.52	36.49	14.52
5	15.77	5.04	12.13	3.81
overall	47.91	10.59	36.49	7.60

Table 4 The Performance of Kalman Filter and FFNN

(Unit: pixel)

Sample	Kalman filter model		FFNN model	
	Maximal error	Average error	Maximal error	Average error
华	28.96	12.29	21.60	8.20
南	30.48	12.84	34.12	10.04
理	33.98	16.20	31.03	12.54
工	23.72	12.80	14.80	6.89
大	22.06	10.32	16.80	6.54
学	25.36	10.97	32.51	8.61
overall	33.98	12.23	34.12	8.80

As it is usually pointed out^[16], a central premise in the Kalman filtering formulation is that it assumes that the underlying model parameters $\{F, G, H, R, Q\}$ are accurate. When the parameters R, Q of the Kalman filter model are reasonably assumed, the performance of Kalman filter is similar to that of FFNN, like the fifth row of table 3. When the assumption is not satisfied, the performance of the filter can deteriorate

appreciably, like the fourth row of Fig.7. FFNN has the merit that it is not sensitive to the initial parameter selection. However, we have also noticed from experiment that when the writing speed of the predicted character is quite different from the writing speed of the train sample, the performance of FFNN might become worse.

6. Conclusions

In this paper, an improved FFNN predictor is

proposed to predict moving handwritten Chinese trajectories, the results of experiments indicate that FFNN can precisely and real-time predict the time series of handwritten trajectory very well. In some situations it is even more powerful than Kalman Filter. The disadvantage of FFNN is that more complex matrix computations are required than Kalman filter. The improved FFNN proposed can be also applied to other time series prediction and object tracking, which may worth our further study.

Fig 7 The Prediction of Result of Kalman Filter and FFNN, Left: the Result from FFNN, Right: the Result from Kalman Filter, the First Row Corresponds to the Fourth Row of Table 3, and the Second Row Corresponds to the Fifth of Table 3

Fig 8 The Prediction Results of Kalman Filter and FFNN, Left: the Result from FFNN, Right : the Result from Kalman Filter, the First Row Corresponds to the Third Row of Table 4, the Second Row Corresponds to the sixth of table 4

References:

- [1] Johnson.H, *Relationship between User Models in HCI and AI*, Computers and Digital Techniques, IEEE Proceedings, Volume: 141, Issue: 2, Mar.1994: pp. 99-103.
- [2] Mirio.E.Munich, Pietro Perona, *Visual Input for Pen-Based Computers*, IEEE Transaction On Pattern Analysis and Machine Intelligence, VOL 24, No.3, March 2002: pp.313-328.
- [3] Kalman,R.E, *A New Approach to Linear Tilter and Prediction Problem*, Transaction of the ASME-Journal of Basic Engineerings, March 1960: pp. 35-45.
- [4] Charalambous, C,Charitou, A, Kaourou, F, *Comparative Analysis of Artificial Neural Network Models: Application in Bankruptcy Prediction*, IJCNN '99, International Joint Conference on Neural Networks, vol.6, July 1999: pp.3888-3893.
- [5] Ahmed Rubaai, Raj.Kotaru, M.David.Kankam, *Online Training of Parallel Neural Network Estimators for Control of Induction Motors*, IEEE Transactions on Industry Applications, Vol. 37, No. 5, September/October 2001.
- [6] S.Chen,S.A.Billings, C.F.N.Cowan, P.M.Grant, *Parallel Recursive Prediction Error Algorithm for Training Layer Neural Network*, International Journal of control, Vol.51, No.6, 1990: pp.1215-1228,
- [7] D.R.Hush, B.G.Horne, *Progress In Supervised Neural Network*, IEEE Signal Proc. Mag., January 1993: pp.9-39.
- [8] P.J.W.Rayer, *A New Connectionist Model Based on A Non-linear Adaptive*, Proc. ICASSP, 1989: pp.1191-94.
- [9] R.J.Williamnce, D.Zipser, *A Learning Algorithm for Continually Running the Neural Recurrent Networks*, Neural Computation, Vol.1, No.2, 1989: pp.270-280.
- [10] S.Haykin, *Neural Networks- A Comprehensive Foundation*, MacMilian,1994.
- [11] C.L.Philip.Chen, John.Z.Wan, *A Rapid Learning and Dynamic Stepwise Updating Algorithm for Flat Neural Networks and the Application to Time-series Prediction*, IEEE Transaction On Systems, Man, And Cybernetics—PART B: Cybernetics, Vol.29, No.1, February 1999: pp.62 – 72.
- [12] C. L. P. Chen, *A Rapid Supervised Learning Neural Network for Function Interpolation and Approximation*, IEEE Transaction , Neural Networks, Vol. 7, Sept. 1996: pp. 1220-1230.
- [13] Y. H. Pao, Y. Takefuji, *Functional-Link Net Computing, Theory, System Architecture, and Functionalities*, IEEE Computer, Vol. 3, 1991: pp.76-79.
- [14] Y. H. Pao, G. H. Park, and D. J. Sobajic, *Learning and Generalization Characteristics of the Random Vector Functional-link Net*, in Neuro-computing, Amsterdam, The Netherlands: Elsevier, Vol. 6, 1994: pp.163-180.
- [15] Sathyanarayan.S.Rao,Viswanath.Ramamurti, *A Hybrid Technique to Enhance the Performance of Recurrent Neural Networks for Time Series Prediction*, IEEE International Conference on Neural Networks, Vol.1, April 1993: pp 98-103.
- [16] Dominguez.S.M, Keaton.T, Sayed. A.H, *Comparison of Robust Estimation and Kalman Filtering Applied to Fingertip Tracking in Human-machine Interfaces*, Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, Vol.1, Nov. 2001: pp.342-346.
- [17] R.k.Mehara, *On the Identification of Variances and Adaptive Kalmna Filtering*, IEEE Transcations on Automatic Control, AC-15,1970: pp.175-183.

(Editors: Norry, Ivan, Xining Xu)