International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Dynamic Neural Network Based Human Recognition through Mouse Movement

K. Nagi Reddy
Department of CSE, RRSCET
Medak (Dist), Telangana, India, 502 300

Dr. C. D. Naidu
Director, PG Studies
VNR VJIET, Hyderabad, India, 500 090

Dr. K. Suvarchala
Department of CSE, ST. Mary's Group
Hyderabad, Telangana, India, 508 284

***Abstract* – In this paper we present a system to identify a person through mouse movement using Hopfield based Dynamic Neural Network. The DNN has a composite structure wherein each node of the network is a Hopfield network by itself. The DNN with reuse is more suitable to accomplish the present task as it is having the properties like associative memory with 100% recall, large storage capacity, avoiding spurious states and converging only to user specified states. Since the DNN with reuse operates with binary data while the information in Mouse Movement Database is in decimal form, the decimal digits are mapped a binary string called descriptor. The network is trained by descriptors (binary strings), which are obtained through the steps indexing through thresholding. These descriptors are used as exemplar patterns to store them in the DNN with reuse by training. The associative memory property of the network makes it to recall a closest pattern of the given query pattern among the memorized patterns. The experimental results are reported to corroborate that high level of precision can be obtained for efficient recall of inexact queries using the DNN with reuse.**

***Keywords* – Hopfield Network, Dynamic Neural Network, Reuse, Associative Memory, Thresholding, Exemplar Patterns.**

I. INTRODUCTION

System administrators are quite often required to know whether the person using the system is an authorized or a genuine user or not. In the present task we are trying to identify a person through the mouse movement. In this case we are assuming that each individual has his/her own way (style) of moving a mouse initially when a person logs on to a system. Based on this assumption we have created a large database of mouse movement recordings of different persons.

The database of mouse movement is trained by the Dynamical Neural Network with Reuse. A query mouse movement is presented to the network as a test pattern and trying to match with the mouse movement data already memorized by the network. The process of creating mouse movement database involves i) Recording of movement sequences ii) The movement sequences are to be converted into a binary string or descriptor by thresholding. The binary string or descriptor of the query consists of unknown mouse movement data and is presented as test pattern. The network matches to the nearest memorized pattern to the query from the database.

It is assumed that the mouse movement operation on the screen is assumed as a window of size 640x480 pixels. It is to be noted that the starting position of the mouse movement is generally the center of the window and the ending point of movement will be different for different people at different instances of time. Taking the above said facts into consideration, it is very difficult to have different binary strings or descriptors for a particular person at different instances of time. Hence the following restrictions are placed on the movement of mouse on the screen of size 640x480 pixels for the purpose of creating mouse movement database.

1.The starting position of the mouse on the screen is determined depending upon the name of the person through the Hashing Method.

2.Terminating point of the mouse movement is to one of the Icons (Buttons) created on the window of the screen.

II. DYNAMIC NEURAL NETWORK (DNN)

The DNN is a composite structure wherein each node of the network is a Hopfield network by itself [6]. The Hopfield network employs a new learning rule[7] and hence converges to the user-specified stable states without allowing any spurious states. The network is dynamic as it changes its architecture during the process. The DNN contains a set of basic nodes which are identical Hopfield networks. However, during the learning phase, the nodes acquire different synaptic weights. The basic nodes are grouped together in a hierarchical organization. Each group has a designed basic node called the leader. When some pattern is presented to DNN, it is presented to all the basic nodes at the lowest level of the hierarchy of nodes. Each node reaches its own stable state based on the common input and individual synaptic weights. These nodes transmit their stable states to their respective leader. Here, the DNN adopts a pruning mechanism and retains only the leader nodes. These leader nodes are treated as the basic node of the next level of the hierarchy and they send the resulting states to the leader nodes at the next level of the hierarchy after reaching the stable state. The process proceeds in this way till the whole network reaches a single stable state. In one cycle, the available basic nodes carry out the state transition function with the given synaptic matrices and in the next cycle these nodes communicates [9] among themselves to change the synaptic weights. At this stage the network is pruned to retain only the leader nodes of the current level of the hierarchy. In [6] it is shown that by this process the network can accomplish very efficient associate recall without any spurious states or any sort of memory limitations, which are two main drawbacks of the Hopfield model of the neural networks. However, the DNN model overcomes the above drawbacks by retaining back the pruned nodes and reusing them in subsequent iterations [5].

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

III. Hashing Method

In order to fix the co-ordinates (x, y) of the starting position of the mouse movement of each individual person, Hashing Technique [2,3] is used. In this procedure the name of the person whose mouse movement is recorded is taken into consideration for fixing the co-ordinates of the starting position of the mouse movement. The number of characters in the name of the person is divided into two halves, first half characters contribute to x co-ordinate and the second half characters contribute to y co-ordinate on the screen. The ASCII values of the characters in the name of the person are considered to compute the co-ordinates of the starting point of the mouse movement. To further ease the understanding of the Hashing method, the steps involved in it are compiled in the form of an algorithm and is presented in figure 1

Input : W_i : Name of the person, l: length of the world W_i.
Output : (x, y) Co-ordinates of the initial position of the mouse movement of W_i.
Procedure: Hash (l, W_i: in; x, y: out)

1. x=0; y=0; /* Initial co-ordinates of mouse movement */
2. for i=1 to l/2 do
3. x = x * 2 + ASCII(W_i)
4. end do
5. for i = l/2 + 1 to l do
6. y = y * 2 + ASCII(W_i)
7. end do
8. x = x mod 640
9. y = y mod 480
10. End

Fig.1. The Algorithm for Hashing Method

A case is taken in which the name of the person is JASWANTH containing eight characters. The first four characters contribute to x co-ordinate and next four characters to y co-ordinate.

ASCII(J) = 74 ASCII(A) = 65
ASCII(S) = 83 ASCII(W) = 87
ASCII(N) = 78 ASCII(T) = 84
ASCII(H) = 72

x = 0*2+ASCII(J) = 0* 2+74 = 74
x = 74*2+ASCII(A) = 148+65 = 213
x = 213*2+ASCII(S) = 426+83 = 509
x = 509*2+ASCII(W) = 1018+87 = 1105

Hash value of "JASW" = 1105 mod 640 = 465 and similarly the hash value of "ANTH" = 1072 mod 480 = 112. Thus co-ordinates of the starting position of the mouse of JASWANTH is fixed at (x, y) = (465,112) with respect to name of the person.

IV. Our Approach

The large database of mouse movement of different people is created. As an example the mouse movement of different persons by name JASWANTH and PRASAD on the screen is given in figure 2. The points in the path of mouse movement are taken at an interval of 5 msec duration amounting to 50 sample points. The point to point slope is evaluated along the line of mouse movement, which comes out to be as 50 sample slope values [4]. The slope values are read as S_i

S_1	S_2	S_3	…	S_{50}

4.1. Thresholding

As the neural network handles a binary data, the decimal values of slopes are to be mapped into a binary string through thresholding method. Thresholding is done by comparing the average slope value with the individual slope values. Average slope value is the ratio of sum of all 50 slope values and number of samples. Each slope value is compared with the average value and is replaced by 1's and 0's, if greater and less than the average value, respectively. Through this method, each data set in the data base is mapped to a descriptor.

And hence a Descriptor or a binary string of 50 bits is obtained. The computational steps involved in obtaining a descriptor or a binary string are presented as an algorithm in Figure 3. As explained above the binary strings of all the mouse movements of different people in the database are obtained.

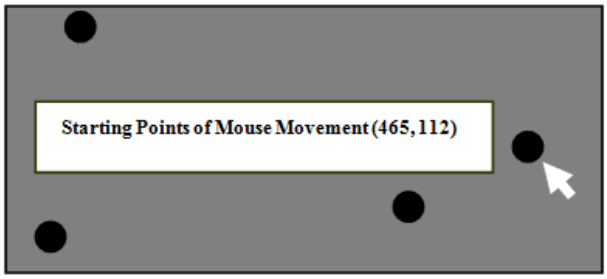

Fig.2. (a) Mouse Movement of JASWANTH

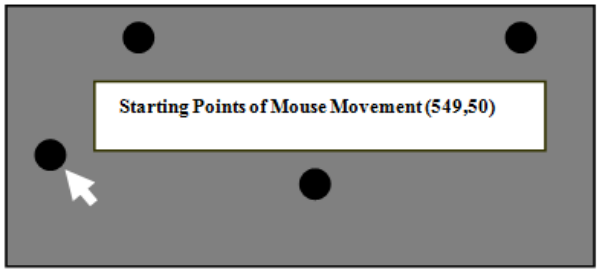

Fig.2. (b) Mouse Movement of PRASAD
Fig.2. Mouse Movement of different persons

4.2. String matching through DNN with Reuse

The need to find an approximate match to a string arises in many practical problems [1]. A mouse movement database matching is such a system wherein to identify a person through a descriptor or a binary string that describes the user name and mouse movement of the person. The query from a user with descriptor[8] or a binary string which may or may not match exactly with that of the data in the database. In this context even if the query from the user matches partially or fully with the

stored pattern, the information related to that person can be retrieved.

The descriptor or a binary string is obtained for each person mouse movement through the algorithm (Figure 3). The descriptor is a bit encoding of the values used to retrieve the record. These descriptors are memorized in DNN with reuse by training. A query descriptor is formed using the same encoding technique. The DNN with reuse retrieves the descriptor of the person which approximately matches with that of the query.

Input: D – Database consisting of K different names $W_1, W_2, W_3, \ldots, W_k$
Output: S – Descriptor of the name (W_k)
Procedure Mouse (D: in; S: Out)

1. for i = 1 to k do
2. Hash (); /* Co-ordinates are evaluated for each W_k */
3. Buttons are created at four different location on the screen
4. line(); /* Mouse movement from initial position to one of the buttons on the screen */
5. Evaluate different slope values for different samples of 5 msec duration from starting position to the terminating position of the mouse.
6. Threshold ();
7. A descriptor (S_i) is obtained after thresholding is done
8. End do
9. End

Fig.3. The Algorithm for Obtaining a Descriptor of the Mouse Movement

The following example illustrates the network behavior for approximate retrieval in case of mouse movement database.

Let us consider a sample mouse movement database which consists of sixteen persons. The names of sixteen persons are given below:

Person	1 Srinivas	9 Shravani
	2 Jaswanth	10 Aruna
	3 Gandhi	11 Manjula
	4 Venkatesh	12 Padmaja
	5 Prabhudev	13 Parvathi
	6 Rambabu	14 Sunitha
	7 Shivaji	15 Sujatha
	8 Krishna	16 Prasad

The descriptor of persons, are obtained as S_i in Figure 4.

Let us assume that the system administrator wants to know whether the user is an authorized user of the system or not based on the mouse movement. The descriptor of the query X is computed through algorithm (Figure 3).

S_1 = 111111111111111111111111111000001110000000001111110
S_2 = 111111111011000000111111111111111111111111111001111
S_3 = 000010000000111111111111000000000000000000000000000
S_4 = 111111111000001110011010001001111001111111111111111
S_5 = 111111100010011101111111111111111011111111110100000
S_6 = 000000000000000110010100000110101110000110000000000
S_7 = 011111111111111111111011100111110100010000000000000
S_8 = 000000000000000011110110001011011111111111111100000
S_9 = 111111111001011010000110111111111010110111110000101
S_{10}= 000000000010110100000111111011000010100000000011101
S_{11}= 111111110011110100000000010000000000000000000000011
S_{12}= 000000000001111000000101110111111110000000000000001
S_{13}= 111111111111111001101100000111001111110110111111001
S_{14}= 000000000111110111101111111000000010101111101000000
S_{15}= 111111111110011110000100001011001111110000011011
S_{16}= 000000000000000000000000010010010011000110000000001

Fig.4. Descriptors of Persons

X = 1110011110110000001111111111111001111111111111000000.

The closest pattern to the test pattern from the mouse movement database is obtained after 5 iterations. The output pattern of the DNN is

S_2 = 111111111011000000111111111111111111111111111001111.

This is descriptor or binary string of the mouse movement of JASWANTH, which is required.

V. Experimental Results

The performance of the DNN with reuse critically depends on the efficiency of descriptor computation and the efficiency of descriptor computation depends on the appropriate method of evaluating the coordinates of the starting position of the mouse movement. In the event of false drop, it may happen that two distinct records (mouse movement data) may correspond to same descriptor. It is

observed that for appropriate choice of the coordinates of the starting position of mouse movement, false drop can be avoided. Experiments were carried out to evaluate the performance of DNN with reuse in the context of mouse movement database retrieval system.

For mouse movement application the collection of mouse movement data (Figure 2) of different people is recorded and is considered for a case study. The staring position of the mouse on the screen is determined using hashing technique. The descriptor of the mouse movement is determined through the algorithm (Figure 3).

The descriptors are memorized in DNN with Reuse by training. The process of matching the mouse movement starts when query descriptor of mouse movement is presented as test pattern. Experiments were carried out by varying the values of N (size of the descriptor) from 10 to 50. Results of the experiments were tabulated as below:

Descriptor length	No. of input queries	No. of accurate retrievals	% of retrievals
10	10	5	50
20	10	5	50
30	10	6	60
40	10	7	70
50	50	37	74

Based on the experiments, it is observed that the percentage of retrieval is dependent on the descriptor size. Hence it is concluded that the DNN with reuse can perform an efficient approximate search if appropriate technique can be developed to find the descriptor. However, if proper descriptor computation scheme is employed the retrieval performance is believed to be improved. Though the performance is degraded due to false drop, around 74% of retrieval makes this method a novel one.

VI. Conclusion

In this paper, matching capability in databases of DNN with reuse is discussed. Experiments were carried out on the performance of the network by considering the case study, namely mouse movement database. It is shown that the neural network based retrieval scheme works efficiently even for very large databases

References

[1] Hall, P.A.V. Dowling, G.R. Approximate String Matching. ACM Computing Surveys, 12,4, 381-409

[2] Knuth, D.E. The Art of Computer Programming Vol. 3: Sorting and Searching. Addision Wesley, Reading, Mass, 1973.

[3] Knuth, D.E., Morris, J.H., Pratt, V.R. Fast pattern Matching strings, SIAM Journal of Computing, 6, 2, 323-350, 1977.

[4] Nidal F. Shilbayeh and Mahmoud Z. Iskandarani. An Intelligent Multilingual Mouse Gesture Recognition System. International Journal of Computer Science 1(3); 346-350, 2005

[5] Arun K Pujari, C.Dhanunjaya Naidu, B C Jinaga. An Adaptive character recognizer for Telugu Scripts using Multiresolution Analysis and Associative Memory. Proceedings of the Indian Conference on Computer vision, Graphics & Image Processing 235-240, 2002

[6] Rao, M.S., Arun K Pujari and sreenivasan. B. A New Neural Network architecture with associate memory, Pruning and order sensitive learning. International Journal of neural Systems 9, 4, 351-370, 1999

[7] Hopfield. J. J. Neural Networks and Physical Systems with emergent collective computational abilities. Proceedings of National Academy of Sciences, USA, 79, 2554-2558, 1982

[8] Athithan, G. A comparative study of two learning rules for associative memory. Pramana Journal of Physics, 45, 569-582, 1995.

[9] Abbott, L.F., Kepler, T.B.Universality in space of interactions for network models. Journal of Physics A Math. Gen., 22, 2031-2038, 1989