International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Predicting Mouse Cursor Target using Backpropagation Neural Network

Pradeep V, Jogesh Motwani

Abstract: ***Many solutions were proposed in the past decades to assist the people with disability in the movement to interact with personal computers. Various results were proposed to simulate the mouse cursor movement and click operations through facial expressions captured by the camera. Tracking and converting accurately the facial expression of the user to the mouse operation is still acknowledged as a research challenge and opportunity. The proposed system introduces a prediction of items to be selected by the user in the GUI based system applying the backpropagation neural network techniques to improve the performance of the overall selection process.***

Keywords : assistive technology, backpropagation neural network, Camera mouse, hands-free computing, people with disability, predicting mouse cursor target

I. INTRODUCTION

About 5.4 million persons in India are having a disability in movement as per the Census report 2011 [1]. Their inability to use the standard input devices of personal computers makes them away from the world of Information Technology. Many mouse replacement solutions were proposed in the past decades to assist them in interacting with personal computers. Hutchinson et al. [2] have designed a system that relies on special hardware and software designed specifically for motor disabled persons. The solutions developed by Gips et al. [3], Lacourse et al. [4], Chen et al. [5], Evans et al. [6] and Barreto et al. [7] require special hardware for the users to wear on the head or face to operate the computer. Kocejko et al. [8] and Lupu et al. [9] have controlled the mouse cursor by tracking the eye gaze movements of the user. Betke et al. [10], Epstein et al. [11], Nabati et al. [12], Chareonsuk et al. [13], Varona et al. [14], Bian et al. [15], Gorodnichy et al. [16], Gyawal et al. [17] and Morris et al. [18] have avoided the overhead of using and head-mounted devices and high-cost hardware system by capturing the user's head motions through web camera to control the mouse pointer. Fathi et al. [19] achieved this by tracking the eye movement whereas Sugano et al. [20], Sambrekar et al. [21] and M. Nasor et al. [22] have attempted tracking the eye gazes. Betke et al. [10], Nabati et al. [12], Chareonsuk et al. [13], Varona et al. [14], Bian et al. [15], Gorodnichy et al. [16], Fathi et al. [19], Hegde et al. [23] and Arai et al. [24] have developed the camera-based mouse replacement solutions for implementing mouse click events such as single and double-clicking and dragging. Tracking and converting accurately the facial expression of the user to the mouse operation is still acknowledged as a research challenge and opportunity.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
Pradeep V*, Research Scholar, Department of CSE, Channabasaveshwara Institute of Technology, Tumkur, India. Visvesvaraya Technological University, Belagavi, India. Email: writetopv@gmail.com
Jogesh Motwani, Professor, Department of CSE, Channabasaveshwara Institute of Technology, Tumkur, India. Visvesvaraya Technological University, Belagavi, India. Email: jogeshmotwani@gmail.com

II. METHODOLOGY

A. Neural Network Forward Pass

The proposed system introduces a prediction of items to be selected by the user in the GUI based system applying the backpropagation neural network techniques to improve the performance of the overall selection process. The neural network is designed with four input neurons, four hidden neurons and two output neurons. The basic structure of the proposed neural network system is shown in Fig. 1.

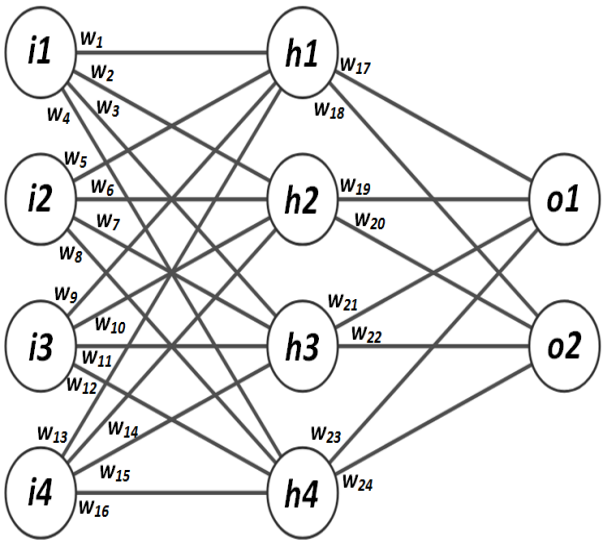

Fig. 1. The basic structure of the proposed Neural Network

The input layer of the neural network contains the following four input neurons as depicted in Fig. 2.

- (*i1*, *i2*), the x and y position of the cursor of the previous selection event.
- (*i3*, *i4*), the difference in x and y positions of the cursor in the nth frame from (*i1*, *i2*) as -1 or 1 or 0, where '1' denotes positive value, '-1' denotes negative value and '0' denotes no change.

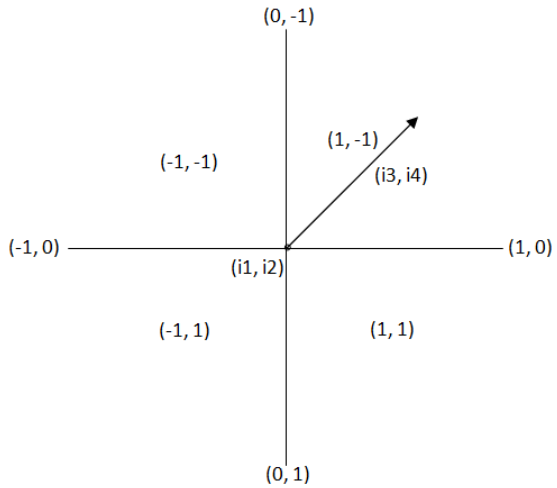

Fig. 2. Input neurons (i1, i2, i3 and i4) of the proposed Neural Network

Number of Hidden layers is selected as suggested by Heaton [25]. The hidden layer contains four neurons that use activation function as follows to calculate the net inputs.

$$in_{h1} = w_1 \times i1 + w_5 \times i2 + w_9 \times i3 + w_{13} \times i4 \quad (1)$$

$$in_{h2} = w_2 \times i1 + w_6 \times i2 + w_{10} \times i3 + w_{14} \times i4 \quad (2)$$

$$in_{h3} = w_3 \times i1 + w_7 \times i2 + w_{11} \times i3 + w_{15} \times i4 \quad (3)$$

$$in_{h4} = w_4 \times i1 + w_8 \times i2 + w_{12} \times i3 + w_{16} \times i4 \quad (4)$$

where $w_i, 1 \leq i \geq 16$ are the weights as shown in the Fig. 1.

The output values of the hidden neurons are calculated using the sigmoid function as follows.

$$out_{h1} = \frac{1}{1+e^{-in_{h1}}} \quad (5)$$

$$out_{h2} = \frac{1}{1+e^{-in_{h2}}} \quad (6)$$

$$out_{h3} = \frac{1}{1+e^{-in_{h3}}} \quad (7)$$

$$out_{h4} = \frac{1}{1+e^{-in_{h4}}} \quad (8)$$

The output layer contains two neurons that use the activation function as follows to calculate the net input.

$$in_{o1} = w_{17} \times out_{h1} + w_{19} \times out_{h2} + w_{21} \times out_{h3} + w_{23} \times out_{h4} \quad (9)$$

$$in_{o2} = w_{18} \times out_{h1} + w_{20} \times out_{h2} + w_{22} \times out_{h3} + w_{24} \times out_{h4} \quad (10)$$

where $w_i, 17 \leq i \geq 24$ are the weights as shown in Fig. 1. The output value is calculated using the sigmoid function as follows.

$$out_{o1} = \frac{1}{1+e^{-in_{o1}}} \quad (11)$$

$$out_{o2} = \frac{1}{1+e^{-in_{o2}}} \quad (12)$$

which is the predicted x and y positions of the user's next mouse selection in the computer monitor screen.

B. Neural Network Back Propagation

The goal of backpropagation is to optimize the weights so that the neural network can learn how to correctly map the arbitrary inputs to outputs. The backpropagation initially processes the output layer and then works backwards processing the hidden layer applying appropriate changes in the weights [26].

The error in the value of the output neurons is calculated using the squared error function as follows.

$$E_{o1} = \frac{1}{2}\left(target_{o1} - out_{o1}\right)^2 \quad (13)$$

$$E_{o2} = \frac{1}{2}\left(target_{o2} - out_{o2}\right)^2 \quad (14)$$

where (out_{o1}, out_{o2}) are the predicted x and y positions and $(target_{o1}, target_{o2})$ are the actual x and y positions of the user's next mouse selection in the computer monitor screen.

The total error for the neural network is the sum of these errors:

$$E_{total} = E_{o1} + E_{o2} \quad (15)$$

Applying (14) in (15),

$$E_{total} = \frac{1}{2}\left(target_{o1} - out_{o1}\right)^2 + \frac{1}{2}\left(target_{o2} - out_{o2}\right)^2 \quad (16)$$

During backpropagation, each weight in the network is updated such that the predicted value comes closer to the target value, thereby minimizing the error.

The partial derivative of E_{total} with respect to the weight w_{17}, $\frac{\partial E_{total}}{\partial w_{17}}$ is the amount of change required in the weight w_{17}.

By applying the chain rule,

$$\frac{\partial E_{total}}{\partial w_{17}} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial in_{o1}} * \frac{\partial in_{o1}}{\partial w_{17}} \quad (17)$$

From Eq. (16),

$$\frac{\partial E_{total}}{\partial out_{o1}} = out_{o1} - target_{o1} \quad (18)$$

From Eq. (11),

$$\frac{\partial out_{o1}}{\partial in_{o1}} = out_{o1}(1 - out_{o1}) \quad (19)$$

From Eq. (9),

$$\frac{\partial in_{o1}}{\partial w_{17}} = out_{h1} \quad (20)$$

Applying Eq. (18), Eq. (19) and Eq. (20) in Eq. (17),

$$\frac{\partial E_{total}}{\partial w_{17}} = \left(out_{o1} - target_{o1}\right) * out_{o1}(1 - out_{o1}) * out_{h1} \quad (21)$$

w_{17} is updated as follows to minimize the error.

$$w_{17} = w_{17} - \eta * \frac{\partial E_{total}}{\partial w_{17}} \quad (22)$$

where η is the rate of learning.

Similarly, the weights, w_{18} to w_{24} are updated as follows.

$$w_i = w_i - \eta * A * B \quad (23)$$

where

$$A = \left(out_{oj} - target_{oj}\right) * out_{oj}(1 - out_{oj}), \ where\ j = \begin{cases} 1 & if\ i\epsilon\{17,19,21,23\} \\ 2 & if\ i\epsilon\{18,20,22,24\} \end{cases}$$

$$B = out_{hj}, \ where\ j = \begin{cases} 1 & if\ i\epsilon\{17,18\} \\ 2 & if\ i\epsilon\{19,20\} \\ 3 & if\ i\epsilon\{21,22\} \\ 4 & if\ i\epsilon\{23,24\} \end{cases}$$

Next, the backwards pass is continued by updating the weights leading into the hidden layers i.e., calculating the new values for w_1 to w_{16}.

The partial derivative of E_{total} with respect to the weight w_1, $\frac{\partial E_{total}}{\partial w_1}$ is the amount of change required in the weight w_1.

By applying the chain rule,

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial in_{h1}} * \frac{\partial in_{h1}}{\partial w_1} \quad (24)$$

Applying Eq. (15) in Eq. (24),

$$\frac{\partial E_{total}}{\partial w_1} = [\frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}] * \frac{\partial out_{h1}}{\partial in_{h1}} * \frac{\partial in_{h1}}{\partial w_1} \quad (25)$$

Expanding $\frac{\partial E_{o1}}{\partial out_{h1}}$ and $\frac{\partial E_{o2}}{\partial out_{h1}}$, in Eq. (25),

$$\frac{\partial E_{total}}{\partial w_1} = [\frac{\partial E_{o1}}{\partial in_{o1}} * \frac{\partial in_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial in_{o2}} * \frac{\partial in_{o2}}{\partial out_{h1}}] * \frac{\partial out_{h1}}{\partial in_{h1}} * \frac{\partial in_{h1}}{\partial w_1} \quad (26)$$

Expanding $\frac{\partial E_{o1}}{\partial in_{o1}}$ and $\frac{\partial E_{o2}}{\partial in_{o2}}$, in Eq. (26),

$$\frac{\partial E_{total}}{\partial w_1} = \left[\frac{\partial E_{o1}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial in_{o1}} * \frac{\partial in_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{o2}} * \frac{\partial out_{o2}}{\partial in_{o2}} * \frac{\partial in_{o2}}{\partial out_{h1}}\right] * \frac{\partial out_{h1}}{\partial in_{h1}} * \frac{\partial in_{h1}}{\partial w_1} \quad (27)$$

From Eq. (13),

$$\frac{\partial E_{o1}}{\partial out_{o1}} = out_{o1} - target_{o1} \quad (28)$$

From Eq. (9),

$$\frac{\partial in_{o1}}{\partial out_{h1}} = w_{17} \quad (29)$$

From Eq. (14),

$$\frac{\partial E_{o2}}{\partial out_{o2}} = out_{o2} - target_{o2} \quad (30)$$

From Eq. (12),

$$\frac{\partial out_{o2}}{\partial in_{o2}} = out_{o2}(1 - out_{o2}) \quad (31)$$

From Eq. (10),

$$\frac{\partial in_{o2}}{\partial out_{h1}} = w_{18} \quad (32)$$

From Eq. (5),

$$\frac{\partial out_{h1}}{\partial in_{h1}} = out_{h1}(1 - out_{h1}) \quad (33)$$

From Eq. (1),

$$\frac{\partial in_{h1}}{\partial w_1} = i_1 \quad (34)$$

Applying Eq. (28), Eq. (19), Eq. (29), Eq. (30), Eq. (31), Eq. (32), Eq. (33) and Eq. (34) in Eq. (27),

$$\frac{\partial E_{total}}{\partial w_1} = [(out_{o1} - target_{o1}) * out_{o1}(1 - out_{o1}) * w_{17} + (outo2 - targeto2)*outo2(1-outo2)*w18*outh1(1-outh1)*i1 \quad (35)$$

w_1 is updated as follows to minimize the error.

$$w_1 = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} \quad (36)$$

where η is the rate of learning.

Similarly, the weights, *w2* to *w16* are updated as follows.

$$w_i = w_i - \eta * [A * B + C * D] * E * F \quad (37)$$

where

$$A = (out_{o1} - target_{o1}) * out_{o1}(1 - out_{o1})$$

$$B = \begin{cases} w_{17} & if\ i\epsilon\{1,5,9,13\} \\ w_{19} & if\ i\epsilon\{2,6,10,14\} \\ w_{21} & if\ i\epsilon\{3,7,11,15\} \\ w_{23} & if\ i\epsilon\{4,8,12,16\} \end{cases}$$

$$C = (out_{o2} - target_{o2}) * out_{o2}(1 - out_{o2})$$

$$D = \begin{cases} w_{18} & if\ i\epsilon\{1,5,9,13\} \\ w_{20} & if\ i\epsilon\{2,6,10,14\} \\ w_{22} & if\ i\epsilon\{3,7,11,15\} \\ w_{24} & if\ i\epsilon\{4,8,12,16\} \end{cases}$$

$$E = out_{hj}(1 - out_{hj}), \ where\ j = \begin{cases} 1 & if\ i\epsilon\{1,5,9,13\} \\ 2 & if\ i\epsilon\{2,6,10,14\} \\ 3 & if\ i\epsilon\{3,7,11,15\} \\ 4 & if\ i\epsilon\{4,8,12,16\} \end{cases}$$

$$F = i_k, \ \ where\ k = \begin{cases} 1 & if\ i\epsilon\{1,2,3,4\} \\ 2 & if\ i\epsilon\{5,6,7,8\} \\ 3 & if\ i\epsilon\{9,10,11,12\} \\ 4 & if\ i\epsilon\{13,14,15,16\} \end{cases}$$

At this instant, all the weights will be updated. Repetition of this process will result in the reduction of error in every iteration [27].

III. EXPERIMENT

The system is tested in the Windows 7 Professional 32-bit Operating System with 1366 x 768 Screen Resolution with Landscape orientation. Selection area in the monitor is assumed as 40 square pixels. Five samples of two consecutive user selection areas are shown in Fig. 3., where j_i is the first selection area and j_o is the next selection area for $1 \leq j \leq 5$. The x and y position of the cursor in the area j_i are the *i1* and *i2* neuron values of the input layer as shown in Fig. 1. The x and y position of the cursor in the area j_o are the $target_{o1}$ and $target_{o2}$ values as shown in Eq. 13 and Eq. 14.

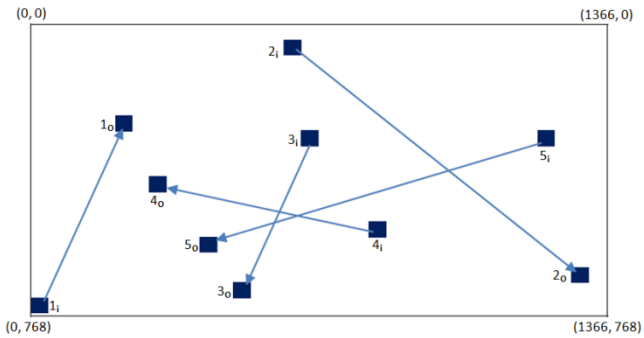

Fig. 3. Five samples of two consecutive user selection areas

The pixel area of square regions j_i and j_o of Fig. 3 are shown in Table I.

Table- I: Pixel area of square regions j_i and j_o of Fig. 3.

Sample (j)	Selection area							
	j_i				j_o			
	x_{min}	x_{max}	y_{min}	y_{max}	x_{min}	x_{max}	y_{min}	y_{max}
1	0	40	720	760	200	240	240	280
2	600	640	40	80	1280	1320	640	680
3	640	680	280	320	480	520	680	720
4	800	840	520	560	280	320	400	440
5	1200	1240	280	320	400	440	560	600

Initial values are assigned randomly to all the weights. For each sample shown in Table 1, the *i1* and *i2* neuron values of the input layer are generated as the x and y position of the cursor within the range specified for the area j_i. The *i3* and *i4* neuron values of the input layer for each sample are fixed as shown in Table 2.

Table 2: Fixed values of i3 and i4 for each sample of Table 1.

Sample	1	2	3	4	5
i3	1	1	-1	-1	-1
i4	-1	1	1	-1	1

The $target_{o1}$ and $target_{o2}$ are generated as the x and y position of the cursor within the range specified for the area j_o for each sample. The total error for the neural network is calculated as shown in Eq. 15. and the weights are updated as shown in Eq. 23 and Eq.37 to minimize the error.

IV. RESULT

The output layer neuron values are predicted for the sample values of Table 1 for multiple passes. The predicted cursor positions of each sample j during the first pass with initial random weights are shown in Fig. 4. The errors are visible which are the difference between the actual cursor position region in the area j_o and the predicted cursor position j_p. Fig. 5 shows that the error is minimized after 5 passes. After 10 passes, the predicted cursor position is in the actual selection for samples 1 and 3 (Fig. 6); after 20 passes for sample 2 (Fig. 7); after 35 passes for sample 4 (Fig. 8); and after 65 passes for sample 5 (Fig. 9). The predicted cursor positions j_p of Fig. 4 to Fig. 9 are summarized in Table 3.

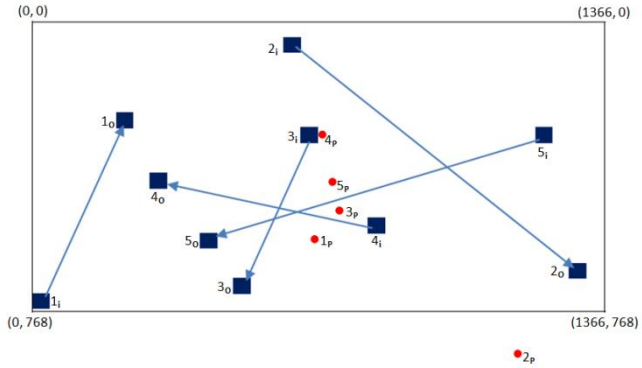

Fig. 4. Predicted cursor positions j_p during the first pass

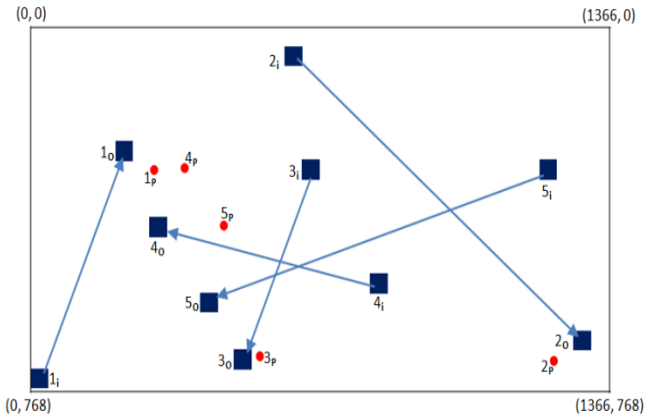

Fig. 5. Predicted cursor positions j_p after 5 passes

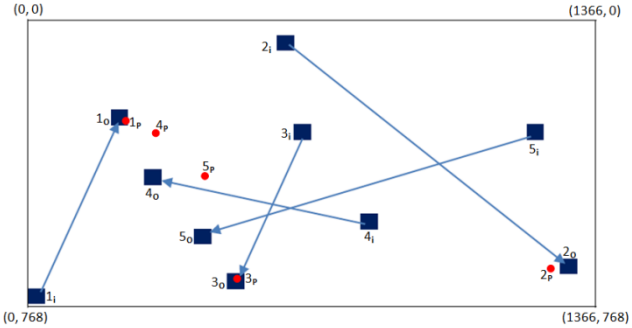

Fig. 6. Predicted cursor positions j_p after 10 passes

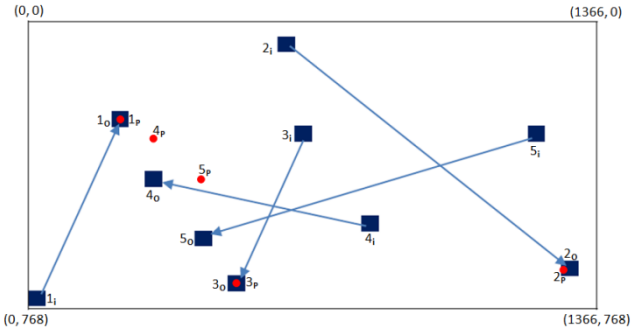

Fig. 7. Predicted cursor positions j_p after 20 passes

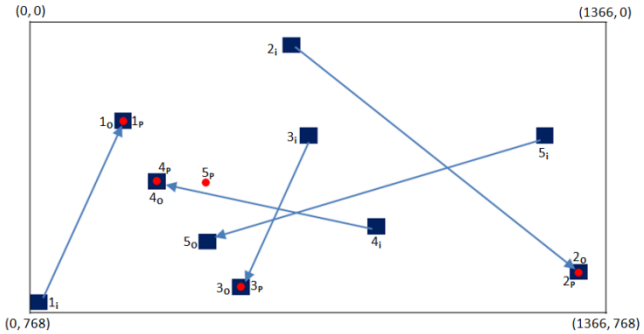

Fig. 8. Predicted cursor positions j_p after 35 passes

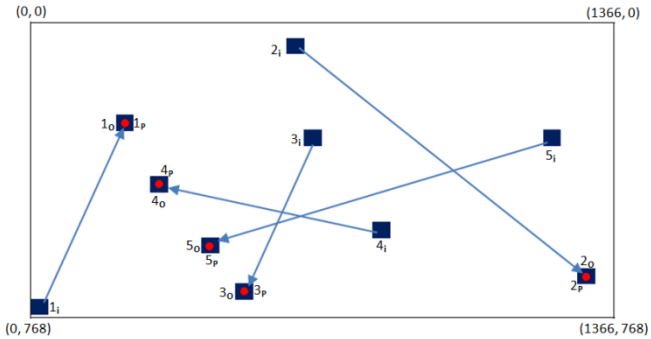

Fig. 9. Predicted cursor positions j_p after 65 passes

Table 3: Summary of predicted cursor positions j_p of Fig. 4 to Fig. 9.

Sample (j)	Predicted cursor positions j_P											
	Pass 1		After 5 passes		After 10 passes		After 20 passes		After 35 passes		After 65 passes	
	x	y	x	y	x	y	x	y	x	y	x	y
1	668	574	289	304	239	267	220	259	222	259	219	258
2	1167	934	1237	697	1266	662	1287	659	1298	660	1300	658
3	730	499	531	693	504	695	498	700	503	699	499	699
4	690	299	348	299	309	298	298	302	303	420	299	418
5	720	418	454	423	424	417	418	420	423	419	419	578

V. CONCLUSION

This paper is focused on introducing a prediction of items to be selected by the user in the GUI based system to improve the performance of the overall selection process. The backpropagation neural network technique is chosen as it is an algorithm widely used in the training of the feed-forward neural networks. The system performance and learning rate can be improved using complex neural network architectures or using different neural network techniques.

REFERENCES

1. Social Statistics Division, Ministry of Statistics and Programme Implementation, Government of India, New Delhi (2017) Disabled Persons in India, A statistical profile 2016, http://www.mospi.gov.in.
2. Hutchinson, T.E., White, K.P., Martin, W.N., Reichert, K.C. and Frey, L.A. (1989) Human–computer interaction using eye-gaze input, IEEE Transactions on Systems, Man, and Cybernetics, 19(6), 1527-1534.
3. Gips, J., Olivieri, C.P., and Tecce, J.J. (1993) Direct control of the computer through electrodes placed around the eyes, Proceedings of Fifth International Conference on Human Computer Interaction, Orlando, FL. In M. J. Smith and G. Salvendy (eds.) Human-Computer Interaction: Applications and Case Studies, Orlando, Florida, 8-13 August, pp. 630-635, Elsevier, Amsterdam.
4. LaCourse, J.R. and Hludik, F.C. (1990) An eye movement communication-control system for the disabled, IEEE Transactions on Biomedical Engineering, 37, 1215-1220.
5. Chen, Y.L., Tang, F.T., Chang, W.H., Wong, M.K., Shih, Y.Y. and Kuo T.S. (1999) The new design of an infrared-controlled human-computer interface for the disabled, IEEE Transactions on Rehabilitation Engineering, 7(4), 474-481.
6. Evans, D.G., Drew, R. and Blenkhorn, P. (2000) Controlling mouse pointer position using an infrared head-operated joystick, IEEE Transactions on Rehabilitation Engineering, 8(1), 107-117.
7. Barreto, A.B., Scargle, S.D. and Adjouadi, M. (2000) A practical EMG-based human-computer interface for users with motor disabilities, Journal of Rehabilitation Research and Development, 37(1), 53-64.
8. Kocejko, T., Bujnowski, A. and Wtorek J. (2009) Eye-Mouse for Disabled. In: Hippe Z.S., Kulikowski J.L. (eds) Human-Computer Systems Interaction. Advances in Intelligent and Soft Computing, vol 60. Springer, Berlin, Heidelberg.
9. Lupu, R.G., Bozomitu, R.G., Păsărică, A. and Rotariu, C. (2017) Eye Tracking User Interface for Internet Access Used in Assistive Technology, Proceedings of E-Health and Bioengineering Conference (EHB), Sinaia, Romania, pp. 659-662, 22-24 June, IEEE.
10. Betke, M., Gips, J. and Fleming, P. (2002) The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 10(1), 1-10.
11. Epstein, S., Missimer, E. and Betke, M. (2014) Using kernels for a video-based mouse-replacement interface, Personal and Ubiquitous Computing, 18(1), 47-60.
12. Nabati, M. and Behrad, A. (2015) 3D Head pose estimation and camera mouse implementation using a monocular video camera, Signal, Image and Video Processing, 9(1), 39-44.
13. Chareonsuk, W., Kanhaun, S., Khawkam, K. and Wongsawang, D. (2016) Face and Eyes mouse for ALS Patients, Proceedings of Fifth ICT International Student Project Conference, Nakhon Pathom, Thailand, pp. 77-80, 27-28 May, IEEE.

14. Varona, J., Yee, C.M. and Perales, F.J. (2008) Hands-free vision-based interface for computer accessibility, Journal of Network and Computer Applications, 31(4), 357-374.
15. Bian, J.P., Hou, J., Chau, L.P. and Thalmann, N.M. (2016) Facial position and expression-based human-computer interface for persons with tetraplegia, IEEE Journal of Biomedical and Health Informatics, 20(3), 915-924.
16. Gorodnichy, D.O. and Roth, G. (2004) Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces, Image and Vision Computing, 22(12), 931- 942.
17. Gyawal, P., Alsadoon, A., Prasad, P.W.C., Hoe, L.S. and Elchouemi, A. (2016) A novel robust camera mouse for disabled people, Proceedings of 7th International Conference on Information and Communication Systems, Irbid, Jordan, pp. 217-220, 5-7 April, IEEE.
18. Morris, T. and Chauhan, V. (2006) Facial feature tracking for cursor control, Journal of Network and Computer Applications, 29(1), 62-80.
19. Fathi, A. and Mohammadi, F.A. (2015) Camera-based eye blinks pattern detection for intelligent mouse, Signal, Image and Video Processing, 9(8), 1907–1916.
20. Sugano, Y., Matsushita, Y., Sato, Y. and Koike, H. (2015) Appearance-based gaze estimation with online calibration from mouse operations, IEEE Transactions on Human-Machine Systems, 45(6), 750-760
21. Sambrekar, U. and Ramdasi, D. (2015) Estimation of Gaze for Human Computer Interaction, Proceedings of International Conference on Industrial Instrumentation and Control, Pune, India, pp. 1236-1239, 28-30 May, IEEE.
22. Nasor, M., Rahman, K. K. M., Zubair, M. M., Ansari, H. and Mohamed, F. (2018) Eye-controlled mouse cursor for physically disabled individual, Proceedings of International Conference on Advances in Science and Engineering Technology, Abu Dhabi, UAE, pp. 152-156, 6 Feb-5 Apr, IEEE.
23. Hegde, V.N., Ullagaddimath, R.S. and Kumuda, S. (2016) Low cost eye based human computer interface system (Eye controlled mouse), Proceedings of IEEE Annual India Conference (INDICON), Bangalore, India, pp. 1-6, 16-18 Dec, IEEE.
24. Arai, K. and Mardiyanto, R. (2010) Camera as Mouse and Keyboard for Handicap Person with Troubleshooting Ability, Recovery, and Complete Mouse Events, International Journal of Human-Computer Interaction, 1(3), 46-56.
25. Heaton, J. (2008) Introduction to Neural Networks with Java. Second ed., Heaton Research, Inc, 2008. ISBN: 1-60439-008-5
26. Kriesel, D. (2007) A Brief Introduction to Neural Networks, dkriesel.com
27. Mazur, M. (2015) A Step by Step Backpropagation Example. https://mattmazur.com/ 2015/03/17/a-step-by-step-backpropagation-example.

AUTHORS PROFILE

Pradeep V, received the Master of Engineering in Computer Science and Engineering from Anna University, Chennai, India in 2014 and the Bachelor of Engineering in Computer Science and Engineering from Bharathiar University, Coimbatore, India in 2002. He has teaching experience of 17 years.

Jogesh Motwani, received the Doctor of Philosophy from Indian Institute of Technology, Bombay, India in 1997, the Master of Science from Northern Illinois University, Illinois, US in 1988 and the Bachelor of Technology in Computer Science and Engineering from Indian Institute of Technology, Bombay, India in 1986. He has teaching and research experience of 30 years.