PARALLEL ARTIFICIAL NEURAL NETWORK APPROACHES FOR DETECTING THE BEHAVIOUR OF EYE MOVEMENT USING CUDA SOFTWARE ON HETEROGENEOUS CPU-GPU SYSTEMS

Norma Alias*, Husna Mohamad Mohsin, Maizatul Nadirah Mustaffa, Siti Hafilah Mohd Saimi, Ridhwan Reyaz

Ibnu Sina Institute, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

*Corresponding author
normaalias@utm.my

Graphical abstract

Abstract

Eye movement behaviour is related to human brain activation either during asleep or awake. The aim of this paper is to measure the three types of eye movement by using the data classification of electroencephalogram (EEG) signals. It will be illustrated and train using the artificial neural network (ANN) method, in which the measurement of eye movement is based on eye blinks close and open, moves to the left and right as well as eye movement upwards and downwards. The integrated of ANN with EEG digital data signals is to train the large-scale digital data and thus predict the eye movement behaviour with stress activity. Since this study is using large-scale digital data, the parallelization of integrated ANN with EEG signals has been implemented on Compute Unified Device Architecture (CUDA) supported by heterogeneous CPU-GPU systems. The real data set from eye therapy industry, IC Herbz Sdn Bhd was carried out in order to validate and simulate the eye movement behaviour. Parallel performance analyses can be captured based on execution time, speedup, efficiency, and computational complexity.

Keywords: High-Performance Computing, Artificial Intelligence, Eyes Movement Behaviour, Heterogeneous CPU-GPU System

1.0 INTRODUCTION

Eyes problem is a very important issue since we use it for every work and task in a daily life. Aging could also be a factor for problems related to eye muscles. Some people have loss of good condition of their muscle of eyes. Information and research related to eye muscles are still lacking. The normal human eye conditions included all eye movement found that the human cornea temperature is 36.5℃ to 37℃ at ambient temperatures between 32℃ and 34.5℃ [1]. Thus, the intraocular pressure is distributed evenly through the eye at the average 15 mm Hg. However, the eyes working are still unknown even the model can be made and can describe some phenomena to obtain a normal temperature of the eye.

Some of the ways that can be used to measure eye and brain activity are by using detection of electrochemical signals in blood flow. This study discusses conditions of eyes based on brain activity through electroencephalogram (EEG) which is known as neural signal [2]. Thus, the integrated of ANN with

EEG are able to obtain the relation between eye movement behaviour and brain activity, since the brain activity can differ when an eye is moving. Before ANN is implemented, proper or good weights must be set. The process of setting the weights is called training or learning [3]. The aim of this paper is to train and predict the eye movement with stress activity using the integrated ANN with EEG digital data signal. The measurement of eye movement is based on eye blinks close and open, moves to the left and right as well as eye movement upwards and downwards. Basically, EEG is an illustration of overall excitation of the brain. EEG sensor device will detect the amplitude and record the eye movement and will be converted from digital to analogue data.

Parallelization is required since the training of large-scale EEG data consume much time [4]. Also stated that parallelization of the training task is a natural reply to the issue of long training times. Parallelization approach can reduce the task [8]. In this paper, the integrated of ANN with EEG signals and parallel algorithms will be applied. The methodologies involve many communications between the nodes. For the small data time, the cost for communication is greater than computation. To solve this problem, there were three alternatives. First of all, it is needed to use a big data set to fulfill the problem of dividing computation and communication is less than one. Next, use domain decomposition method to partition the big data into a subdomain. In addition, the implementation of a heterogeneous CPU-GPU system is an alternative to reduce the communication time.

The rest of the paper has been organized as follows. Section II provides an overall description on the integrated ANN with EEG signals and section III outlines the parallelization techniques implement in the heterogeneous CPU-GPU system. In section IV the parallel performance analysis is presented along with interpretations based on a real data set of eye therapy. Finally, conclusions are outlined in section V.

2.0 METHODOLOGY

EEG signal is an illustration of overall excitation of the brain. Data that has been used in this paper is EEG digital data signals of three conditions of the eye including eye close and open, eyes are moved to the left and right, eye movement upwards and downwards. The methodologies of the prediction and visualization of eye movement behaviour are as follows:
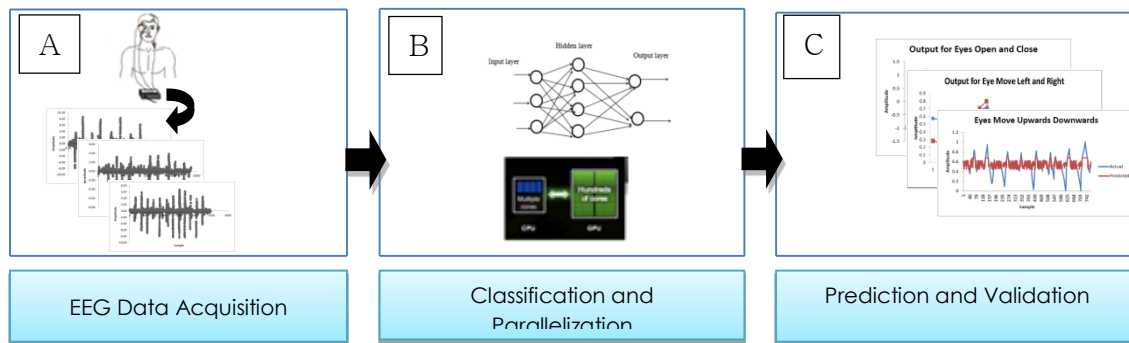

Figure 1 Methodologies involve in prediction and visualization of eye movement behaviour

2.1 EEG Data Acquisitions

For this research, EEG data was taken from eye therapy industry, IC Herbz Sdn Bhd. It is a representation of overall excitation of the brain. The pattern changes significantly between states of sleep and wakefulness. The EEG signal can be divided into five frequency bands. There were beta, alpha, theta, delta and gamma [2]. The range of frequency for every band differ from when the person in awake. The EEG graph was investigated. Three different states of eyes including normal eye moves upwards and downwards, eye closure and eye opening, and the eyes moved to the left and right with voluntary eye blinking condition [9].

2.2 Classification and Digital EEG

Data of EEG signal are limited to three conditions of eyes which are normal EEG of an eye when relax, eye closure and opening as well as eyes move to the left and right. In this paper, ANN has been used for classification and provides the characteristic feature of generalization. The type of neural network that we used is backpropagation neural network [10]. Back-propagation is a method of training in an artificial neural network [11]. Basically, this study applies feed forward architecture of multilayers, sigmoid function as activation function [12] and back-propagation algorithm as the learning algorithm in training process. For backpropagation algorithm, it needs a large set of complete record and with the target variables. The training is done through the network for every

observation. On the other hand, parallelization of ANN is carried out to overcome computation of a large sparse EEG data set using ANN training model [13]. The parallelization of integrated ANN with EEG signals has been implemented on the heterogeneous CPU-GPU systems supported by Compute Unified Device Architecture (CUDA).

2.3 Prediction and Validation

ANN can predict the output of eyes for future precaution or prevention from any problem related to the eyes. The real data set from eye therapy industry, IC Herbz Sdn Bhd was carried out in order to validate and simulate the eye movement behaviour. Parallel performance analyses are conducted to evaluate the parallel performance of the parallel algorithm. The graphs of run execution, speedup, efficiency, computational complexity will be captured and analyse. Some numerical results of sequential and parallel ANN will be presented in the next section.

2.4 Implementation And Outcome

Training data of ANN involves many communications between the nodes. For the small data time, the cost for communication is greater than computation. To solve this problem, there were three alternatives. First of all, it is needed to use a big data set to fulfill the problem of dividing computation and communication is less than one. Next, use domain decomposition method to partition the big data into a subdomain. In addition, the implementation of a heterogeneous CPU-GPU system is an alternative to reduce the communication time.
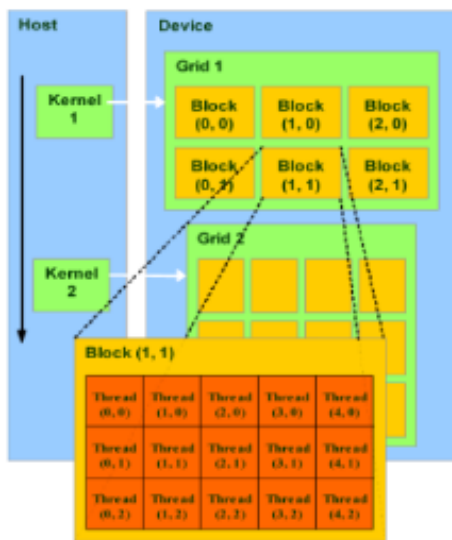
(2a)
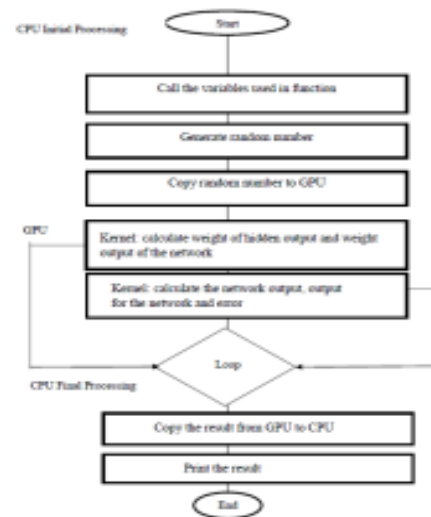
(2b)

Figure 2 CPU-GPU Implementation

Figure 2a illustrates thread and block definition. Every thread in thread block has their identification number called thread ID, as well as every block in the grid, has its block ID. The thread can communicate with other thread in a block by shared memory [14]. Figure 2b shows the flow for GPU implementation. The process starts with generating the random number. Random number has been generated before used it for initial weights process. The first part is in host or CPU where all variables were defined. Next, GPU starts with a kernel. In order to allocate space for the device, the variables are defined using CUDA API, *cudaMalloc*. There were two types of weight that need to find in this study. There were the weights of hidden output and weights of input hidden. The data has been loaded in this network. In addition, the kernel function calls grid and block variables and it using three angular brackets <<<grid, block>>>. After running the kernel, the result is copied back to host [15] using CUDA API cudaMemcpy. The result will be displayed after finish all the iteration.

3.0 RESULTS AND DISCUSSION

The results and accuracy using the artificial neural network (ANN) to detect eye movements are discussed.

3.1 Accuracy of Eye Movement

As stated earlier, there are three conditions of eyes used in this study. Every condition has a different number of data. A number of data for normal condition of eye movements to the left and right are 753. However, only 527 numbers of data were trained for this condition. The number of data training that is used to train every condition is 70 percent of the total number of data. For condition of eyes closed and opened condition, the total number of data is 994. The number of data used for training is 696 data which is 70 percent of the total number of data.

| N=20	Digital Data, Z_n	ANN, T_n	MSE $(Z_n-T_n)^2$
i = 0	-0.4729	-1.0024	0.2804
i=1	-0.4096	-0.761	0.1235
i=2	0.0339	0.1262	0.0085
i=3	0.5405	1.1424	0.3622
i=4	0.4774	0.9836	0.2562
i=5	0.3508	0.7402	0.1516
i=6	0.161	0.578	0.1739
i=7	0.0345	0.071	0.0013
i=8	0.098	0.176	0.0061
i=9	0.2883	0.5424	0.0646
i = 10	0.3514	0.7234	0.1384
i=11	0.1616	0.3434	0.0331
i=12	-0.0283	-0.0374	0.00008
i=13	-0.2811	-0.457	0.0309
i=14	0.0986	0.0334	0.0043
i=15	0.2254	0.4434	0.0475
i=16	0.3525	0.7172	0.1330
i=17	0.2262	0.479	0.0639
i=18	0.1632	0.2962	0.0177
i=19	0.3533	0.7624	0.1674
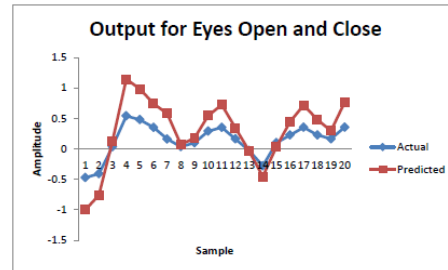
Figure 3 The actual and predicted graph of eyes open and close

Figure 3 shows the graph plotted from 20 outputs of ANN. The blue line is actual data or digital EEG data while the red line is predicted value for eyes open and close condition. The graph can be described having spike patterns for predicted value. At iteration 3, the value for predicted of ANN is 1.1424 where it is at peak. At iteration 4 then the graph drop sharply. For all the points, the lowest point is at iteration 0 with value -1.0024. For accuracy, the total mean square error is 0.10. As a conclusion, both patterns of the graph has a spiking pattern and if for the next number of a sample can predict that the pattern will also go up and down value. This means that for eyes condition, blinking of the eyes increases as the amplitude increase in EEG graph.

| N=20	Digital Data, Z_n	ANN, T_n	MSE $(Z_n-T_n)^2$
i = 0	0.5721	0.2814	0.0845
i=1	0.5625	0.2595	0.0918
i=2	0.5529	0.2327	0.1025
i=3	0.5817	0.2993	0.0798
i=4	0.6105	0.3466	0.0696
i=5	0.6296	0.3926	0.0562
i=6	0.6680	0.5857	0.0068
i=7	0.6872	0.7047	0.0003
i=8	0.7111	0.8000	0.0079
i=9	0.6536	0.4964	0.0247
i = 10	0.6392	0.4265	0.0452
i=11	0.6200	0.3666	0.0642
i=12	0.6200	0.3666	0.0563
i=13	0.6296	0.3924	0.0672
i=14	0.6152	0.3559	0.0697
i=15	0.6104	0.3465	0.0734
i=16	0.6008	0.3299	0.0643
i=17	0.6199	0.3664	0.0249
i=18	0.6535	0.4957	0.0181
i=19	0.6583	0.5237	0.0512
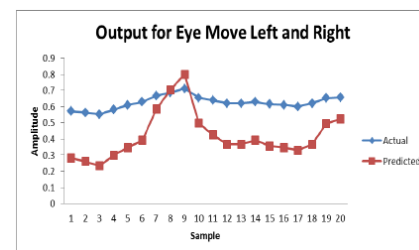
Figure 4 The actual and predicted graph of eyes move to left and right

Figure 4 shows that at iteration $i=8$, the value for predicted of ANN value is -0.8000 where it is at peak. The next iterations, the value is drop. The lowest point for predicted value is at iteration $i=2$ with value 0.2327. Accuracy for this condition is 0.23. From this predicted graph, for the next number of sample, the amplitude may have a spiking pattern as having a high number of sample.

| N=20	Digital Data, Z_n	ANN, T_n	MSE $(Z_n-T_n)^2$
i = 0	0.5678	0.6100	0.0018
i=1	0.5756	0.5624	0.0001
i=2	0.5914	0.4732	0.0140
i=3	0.5441	0.5300	0.0001
i=4	0.5362	0.4717	0.0042
i=5	0.5204	0.4450	0.0057
i=6	0.5440	0.5294	0.0002
i=7	0.5519	0.5885	0.0013
i=8	0.5677	0.6105	0.0018
i=9	0.5755	0.5631	0.0001
i = 10	0.5519	0.5884	0.0013
i=11	0.5439	0.5290	0.0002
i=12	0.5439	0.5288	0.0002
i=13	0.5755	0.5637	0.0001
i=14	0.5833	0.5057	0.0060
i=15	0.5754	0.5640	0.0001
i=16	0.5597	0.6198	0.0036
i=17	0.5478	0.5598	0.0001
i=18	0.5281	0.4420	0.0074
i=19	0.5085	0.4742	0.0012
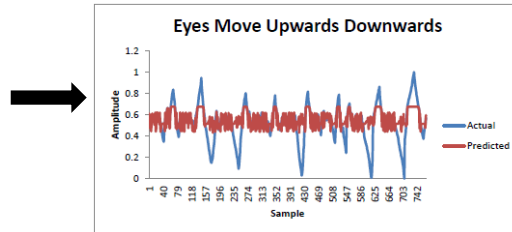
Figure 5 The actual and predicted graph of eyes move upwards downwards

Based on Figure 5, it is the output for eye move upwards and downwards condition. Accuracy for this condition is 0.14. From this predicted graph, for the next number of sample, the amplitude may have spike pattern as having a high number of sample.

3.2 Parallel Performance Analysis

Basically, the execution time between CPU and GPU is compared. The numbers of threads per block that have been tested are different. Table 1 indicates the execution time between CPU and GPU for eyes moves left and right condition. For parallel performance, aspects consider in this study are speedup, efficiency and effectiveness [5], [6], [7]. Figure 6 shows the execution time for eyes close open, eyes move left and right and eyes moves upwards and downwards.

Table 1 Execution time between CPU and GPU in second

| No of Blocks	Execution Time for CPU(second)			Execution Time for GPU(second)		
	Left right	Close Open	Upwards Downward	Left right	Close Open	Upward Downward
5	14.5930	18.9620	19.7020	1.3430	2.9160	2.9300
7	14.5930	18.9620	19.7020	1.1680	2.3330	2.6130
9	14.5930	18.9620	19.7020	1.1110	2.0400	2.3010
11	14.5930	18.9620	19.7020	1.0710	1.7960	1.9690
13	14.5930	18.9620	19.7020	0.7520	1.7590	1.8320
15	14.5930	18.9620	19.7020	0.4950	1.3000	1.7820
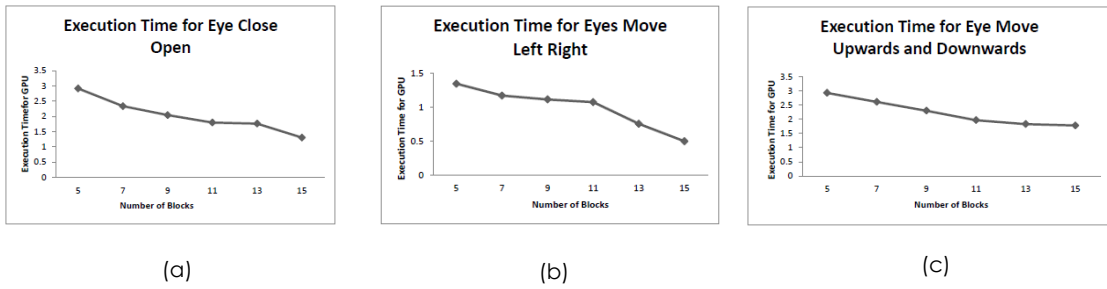
Figure 6 Execution time for a) eyes close open; b) eyes move left and right; c) eyes move upwards downwards

Table 2 shows the analysis of speedup and efficiency for three conditions of eyes. As we can see, speedup increases as the number of blocks increase. This is because high numbers of blocks contribute to speed for a parallel program. While, Figure 7 shows the speedup for eyes close open, eyes move left and right, and eyes move upwards and downwards.

Table 2 Speedup and efficiency for eyes conditions

No of Blocks	Speedup			Efficiency		
	Left right	Close Open	Upwards Downwa-rd	Left right	Close Open	Upward Downward
5	10.8860	6.5027	6.7242	2.1732	1.3005	1.3448
7	12.4940	8.1277	7.5400	1.7849	1.6110	1.0771
9	13.1350	9.2951	8.5624	1.4594	1.0328	0.9514
11	13.6256	10.5579	10.0061	1.2387	0.9598	0.9096
13	19.4056	10.7800	10.7544	1.4927	0.8292	0.8273
15	29.4808	14.5862	11.0561	1.9654	0.1006	0.7371

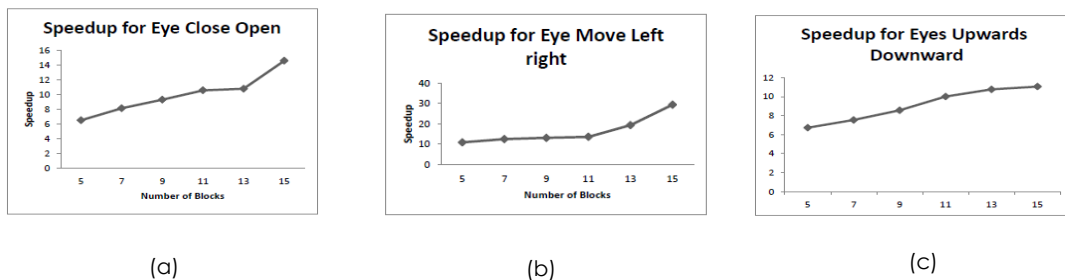**Figure 7** Speedup for a) eyes close open; b) eyes move left and right; c) eyes move upwards downwards
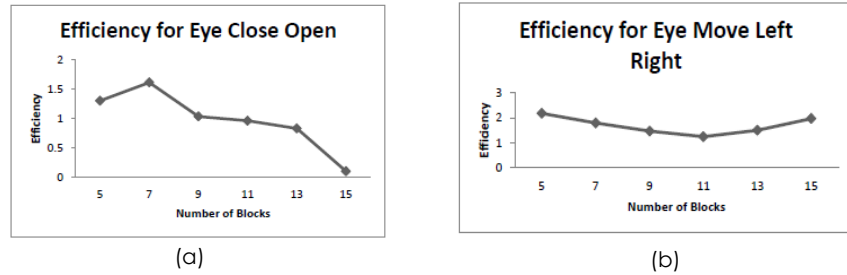
(a)   (b)

Figure 8 Efficiency for a) eyes close open; b) eyes move left and right

Basically, the efficiency decreases when the numbers of blocks increase. In comparison, the efficiency of eyes moves left and right is higher that efficiency for eyes open and close. Eyes closed and opened conditions have a higher number of data compared to eyes moving to the left and right conditions. This reason might contribute to efficiency factor. Meanwhile, Table 3 shows that the two factors for performance analysis which is effectiveness and temporal performance. Figure 8a shows effectiveness for eyes move left and right. The effectiveness increase as the number of blocks increases. However, for eyes close and open the effectiveness such as in Figure 8b also decreases as the number of blocks increases. This could be due to CPU-GPU memory transfers. As a result, it makes a bandwidth problem and due to the minimum time of the communication inside the GPU and the threads in thread block is not in balance.

Table 3 Effectiveness and temporal performance for eyes conditions for eyes conditions

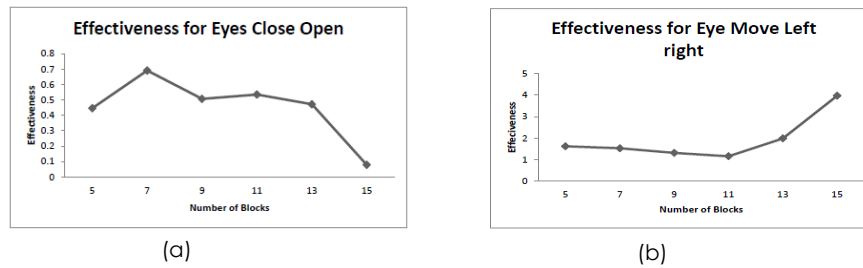| No of Blocks	Effectiveness			Temporal Performance		
	Left right	Close Open	Upwards Downwa-rd	Left right	Close Open	Upward Downward
5	1.6182	0.4460	0.4590	0.7446	0.3429	0.3413
7	1.5281	0.6905	0.4122	0.8562	0.4286	0.3827
9	1.3136	0.5063	0.4135	0.9000	0.4902	0.4346
11	1.1566	0.5344	0.4620	0.9337	0.5568	0.5079
13	1.9850	0.4714	0.4516	1.3298	0.5685	0.5459
15	3.9705	0.0774	0.4136	2.0202	0.7692	0.5612

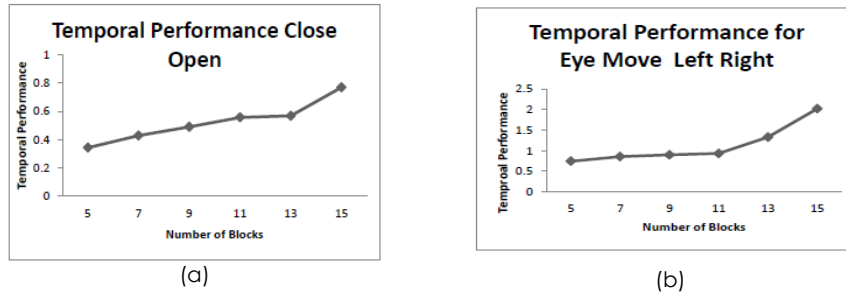**Figure 8** Effectiveness for a) eyes close open; b) eyes move left and right

Figure 9 Temporal Performance for a) eyes close open; b) eyes move left and right

Performance increases as the number of blocks increases, as shown in Figure 9. Lastly, based on the parallel performance, a parallel algorithm is much better than sequential algorithm due to its faster computational time. This means that the parallel algorithm is able to run huge numbers of data while reducing the runtime. The process of training or setting the right weights in artificial neural network model can be done with short time using parallel algorithm compared to a sequential algorithm.

4.0 CONCLUSION

In this paper, the integrated of ANN with EEG signals has been implemented on heterogeneous CPU-GPU systems for eye movement behaviour detection. The digital data of three conditions of eyes has been tested. In addition, this study, deal with actual EEG experimental graph and as a result, it gives a big number of data. These makes the data capable of investigating, a using parallel algorithm and CPU-GPU system. This study is about developing a parallel algorithm for an artificial neural network to apply in eye conditions. Integrated ANN with EEG signals has been compared between sequential and parallel algorithms for large-scale digital data. The process of training or process of setting the right weight in an artificial neural network may consume time as having a large number of data. From Section IV, we can see that execution time for GPU is better than CPU. This means that the GPU enable to parallel the large number of data for different conditions of eyes.

Therefore, parallel ANN is an alternative to predict the trend of EEG graph using a huge number of data.

References

[1] L. Kessel, L. Johnson, H. Arvidsson, M. Larsen. 2010. The Relationship Between Body And Ambient Temperature And Corneal Temperature. *Invest. Ophthalmol. Vis. Sci.* 51: 6593-6597.
[2] Mansor, W., Rani, M. S. A., & Wahy, N. 2011. *Integrating Neural Signal and Embedded System for Controlling Small Motor*. INTECH Open Access Publisher, Chicago. 31-43.
[3] Verber, D. 2012. *Implementation of Massive Artificial Neural Networks with CUDA*. INTECH Open Access Publisher. 278-303.
[4] Dahl, G., McAvinney, A. & Newhall, T. 2008. *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks.*
[5] Alias, N., Mokhtar, H., San, Y., Saipol, H. F. S., Mustaffa, M. N., Said, N. M. & Nor, N. M. 2014. High Speed Computation Of Muscle Stress Problem Using Crack Propagation Modelling On DPCS Platform. *Advanced Science Letters*. 20: 1961-1966.
[6] Norma, A., Islam, M. R. 2010. A Review of the Parallel Algorithms for Solving Multidimensional PDE Problems. *Journal of Applied Sciences*. 10: 2187-2197.

[7] Alias, N., Saipol, H., Ghani, A., Mustaffa, M. N. 2014. Parallel Performance Comparison Of Alternating Group Explicit Method Between Parallel Virtual Machine And Matlab Distributed Computing For Solving Large Sparse Partial Differential Equations. *Advanced Science Letters.* 20(2): 477-482.

[8] Ahmad, A. S., Zuliaonto, A., Sanjaya, E. 1999. Design and Implementation of Parallel Batch-mode Neural Network on Parallel Virtual Machine. *Proceedings, Industrial Electronic Seminar, Grahar Institut Teknologi, Surabaya.* 1-5.

[9] Bulling, A., Ward, J. A., Gellersen, H., & Troster, G. 2011. Eye Movement Analysis for Activity Recognition Using Electrooculography. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.* 33(4): 741-753.

[10] Liu, J., and Guo, L. 2013. Implementation of Neural Network Backpropagation in CUDA. *Intelligence Computation and Eolutionary Computation.* 1021-1027.

[11] Scanzio, S., Cumani, S., Gemello, R., Mana, F., Laface, P. 2010. Parallel Implementation of Artificial Neural Network Training for Speech Recognition. *Pattern Recognition Letters.* 1302-1309.

[12] Sharma, V., Rai, S., & Dev, A. 2012. A Comprehensive Study of Artificial Neural Network. *International Journal.* 2(10): 278-284.

[13] Long, L. N., Gupta, A. 2008. Scalable Massively Parralel Artificial Neural Networks. *Journal of Aerospace Computing, Information and Communication.* 5: 1-13.

[14] Wu, Z. Y., Effekharian, A. A. 2011. Parallel Artificial Neural Network using CUDA- enabled GPU for Extracting Hydraulic Domain Knowledge of Large Water Distribution Systems, World Environmental and Water Resources Congress 2011. *Bearing Knowledge for Sustainability.* 79-82.

[15] Rafia I. 2011. *An Introduction to GPGPU Programming- CUDA Architecture.* Malardalen Research Centre.