

EEG based Artificial Learning of Motor Coordination for Visually Inspired Task using Neural Networks

Shreyasi Datta^{1,a}, Anwesh Khasnobish^{2,b}, Amit Konar^{1,c}, D. N. Tibarewala^{2,d}

¹Dept. of Electronics and Telecommunication Engineering, ²School of Bioscience and Engineering
Jadavpur University, Kolkata, India

^ashreyasidatta@gmail.com, ^banweshakhasno@gmail.com,
^ckonaramit@yahoo.co.in, ^dbiomedju@gmail.com

Atulya K. Nagar

Mathematics and Computer Science Department
Liverpool Hope University
United Kingdom
nagara@hope.ac.uk

Abstract— Damage in parietal and/or motor cortex of the brain can lead to inability in proper visuo-motor coordination, hampering movement planning and execution. The objective of this work is to predict joint coordinates of hand by sequential prediction of the parietal and motor cortex Electroencephalogram (EEG) features from their occipital counterparts using artificial neural networks (ANNs). EEG signals during hand movement execution are acquired from occipital, parietal and motor cortical regions and the joint coordinates of hand are acquired using Kinect sensor. The acquired EEG signals are preprocessed followed by extraction of wavelet features and selection of the best features using Principal Component Analysis. The EEG features originating from one brain region are mapped to the features of another brain region using regression analysis on artificial neural networks with Back Propagation learning. The mapped motor cortical EEG signals are finally used to predict the hand joint coordinates using Back Propagation learning based ANN. The performances of various wavelet adaptation techniques for Back Propagation learning are evaluated. Regression analysis results indicate that Levenberg-Marquardt optimization based weight adaptation performed best in terms of mean squared error, slope of the best linear fit and correlation coefficient between the original values and predicted results.

Keywords—artificial neural network; back propagation learning; electroencephalogram; regression analysis; visual-motor co-ordination

I. INTRODUCTION

Brain Computer Interfacing (BCI) technology [1] has significant contributions in the development of rehabilitative aids for assistance in neuro-motor disabilities. The use of Electroencephalogram (EEG) in BCI is popular because of the simple acquisition and processing, non-invasiveness, high temporal resolution and ease of real time implementations associated with EEG signals. Instances of EEG based BCI study include decoding mental states and cognitive activities [2-3], prosthetic and robotic control through motor imagination

[4-5], emotion recognition [6], object shape recognition from visual and tactile exploration [7] to mention a few.

Patients suffering from Alzheimer's disease, Optic ataxia, Balint's Syndrome and other brain diseases usually have reduced parietal lobe and/or motor cortex functioning [8-10], prohibiting them to correctly control their motor movements. The present work, attempts to provide an artificial pathway to perform the task of visuo-motor coordination in patients with damage in the areas of brain involving decision making and/or coordination related to visual stimuli and corresponding motor actions; namely the parietal and motor cortex regions. Visuo-motor coordination is obligatory for carrying out any motor execution task from visual stimulus. For instance, when a person tries to catch a ball thrown at him by another person, first the visual stimulus of the ball being thrown at him is processed by the occipital lobe of the brain. Then the decision regarding how to position his arms and fingers to catch the ball is made in the parietal region. Finally, according to the decision made, the motor cortex executes the movements of the hands. The occipital, parietal, and motor cortex controls the visuo-motor coordination tasks in a synchronized manner as shown in Fig. 1. In case of damage to any of these brain regions by some accident, trauma, aging or brain ailments, it is difficult for a person to make proper visuo-motor coordination. Our objective is to develop a technology to circumvent the damaged parietal and/or motor cortex to accomplish motor execution only from the occipital EEG signals. Previous works in BCI based rehabilitation include use of artificial limbs, wheelchairs, virtual games controlled by the brain signals of patients [11-14]. However, to the best of the authors' knowledge, no significant work is yet present that aims to design an artificial system capable of bypassing the natural visuo-motor coordination.

The problem can be briefly stated as formulation of a mapping between two regions of the brain. A supervised neural learning or regression algorithm is a good choice for such mapping. In this paper we propose the use of regression analysis on an Artificial Neural Network (ANN) to determine

such mapping. EEG signals from the occipital, parietal and motor cortex regions of a healthy individual for a visuo-motor coordination related task are acquired and subjected to feature extraction. The joint coordinates of the hand during movement execution are also captured using a Kinect Sensor [15]. A three-level ANN based linear regression analysis is done, the first level ANN is trained to predict parietal EEG features from occipital EEG features, the second level ANN is trained to predict motor-cortex EEG features from parietal EEG features while the final level ANN is trained to predict joint coordinates from the motor-cortex EEG features. All ANNs are trained on the principle of Back Propagation Learning [16-17] while the performance is evaluated by varying the weight adaptation techniques of the neural networks.

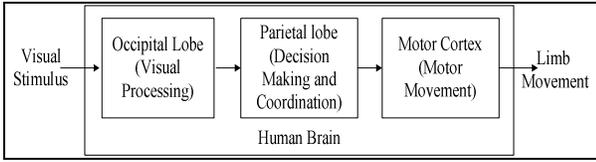


Fig. 1. Movement Execution from Visual Stimuli in Human Brain

The rest of the paper is structured as follows. The methodology followed in the work along with the tools and techniques are explained in Section II. Section III describes the experiments conducted. The results are discussed in Section IV. Finally in Section V conclusions are drawn and the future scopes of work are stated.

II. METHODOLOGY

This section describes the tools and techniques used in this paper and also provide the outline of the work.

A. Feature Extraction

For abstracting the relevant parameters of EEG related to the current task, EEG signals are subjected to feature extraction. After experimental trials with some commonly used EEG based features, Wavelet transform [18-19], which provides both frequency and time-domain analysis at multiple resolutions, is selected for feature extraction as it provides the best results on an average, while overcoming the limitations imposed by Short Time Fourier Transform. In Discrete Wavelet Transform signals are passed through high and low pass filters in several stages. At each stage i , each filter output is down sampled by two to produce the approximation coefficient A_i and the detail coefficient D_i . The approximation coefficient is then decomposed again, to get the approximation and detail coefficients of the subsequent stages. In the present work, Daubechies order 4 (Db4) mother wavelet is used for discrete wavelet transform.

B. Feature Selection

In order to reduce the dimensions of the EEG feature space thereby selecting only the best features, feature selection is performed using Principal Component Analysis (PCA) [20-21]. PCA is an orthogonal linear transformation that transforms the input data into its Eigen space such that the elements of the transformed data are uncorrelated. We need to extract the first

d principal components as the d best features. The Eigen space is so arranged that the Eigen vectors occur in decreasing order of Eigen values. The first Eigen vector has the direction of the largest variance of data and is the first principal component $PC1$ of the dataset. The second Eigen vector has the direction orthogonal to $PC1$ that has maximum variance and is the second principal component $PC2$. The process is continued to determine the higher principal components. Thus, d best features are selected from a total of D features by taking the first d components of the transformed feature space.

C. Back Propagation Learning based Artificial Neural Networks for Regression Analysis

The Back Propagation learning algorithm [11], [16-17], [22] is one of the most popular supervised learning algorithms to train an artificial neural network. It employs a feed-forward topology of neurons, as shown in Fig. 2, each layer with a number of neurons. The neurons in the intermediate and output layers receive weighted signals from those of the previous layer, which are summed up and then passed on to a non-linearity. The error obtained at the output layer is given by (1) where $target_r$ and Out_r denote the target and output produced at the r^{th} neuron of the output layer.

$$E = (1/2) \sum_r (target_r - Out_r)^2 \quad (1)$$

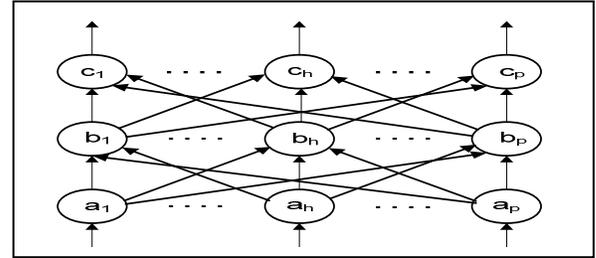


Fig. 2. A 3-layered feed forward neural network where a_i 's, b_i 's and c_i 's denote neurons in the input, intermediate and output layers respectively

The steps of Back-propagation learning in the neural network are:

1. Initialize $i := 1$
2. For the input at the i^{th} instance to the neural network, compute the outputs by a forward pass
3. Compute the error vector E_i at the output layer by taking the difference of each component of the target vector and that of the obtained output vector, i.e., $E_{ij} = T_{ij} - O_{ij}$, for all j , where E_{ij} , T_{ij} and O_{ij} denote the j^{th} component of the i^{th} error vector, target vector and output vector respectively.
4. Repeat steps 2 and 3 for $i=1$ to n where n is the number of input instances.
5. Determine the Root Mean Square (RMS) value of error (ERROR), whose j^{th} component is given by

$$(ERROR)_j = [\sum_{i=1}^n E_{ij}^2 / n]^{1/2} \quad (2)$$

6. Back propagate the RMS error components of the last layer to the preceding layers and adapt the weights of the network according to some algorithm starting with the last layer.

Repeat steps (2-6) until $\sum_j (ERROR)_j^2$ is sufficiently small.

The weight adaptation of the neural network is thus an important factor determining the efficiency of its training. There are several well known algorithms for weight adaptation. Some of these, used in this paper have been briefly stated below.

1) Gradient Descent Search

Newton's Gradient Descent Search (GDS) [22] follows the weight adaption rule (3-4) using the error function (1) which is a function of weights of the interconnection.

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (3)$$

$$w = w + \Delta w \quad (4)$$

where w are the weights of the network, E is the error function, and η is the learning rule ($0 < \eta < 1$). In our work, experimentally we have considered η to be 0.01 and the minimum value of the error gradient reaching which the training of the network is stopped is taken as 1.0e-6.

In order to overcome the problem of the network getting stuck at a local minimum, a momentum (mc) term can be added with the general gradient descent search principle [22]. The momentum helps to slide over through the local minimums in the error surface and reach the global minimum. Such a weight adaptation algorithm given by (5) is referred to as gradient descent search with momentum (M-GDS), where $\Delta w_{previous}$ denotes the previous change in weight. The constant momentum used in the present work has been varied from 0 (no momentum) to 1 (high value of momentum) and found that a value of 0.9 improves results.

$$\Delta w = mc(\Delta w_{previous}) + \eta(1 - mc) \frac{\partial E}{\partial w} \quad (5)$$

2) Levenberg-Marquardt Optimization

The simple form of gradient descent learning can be stated as (6), where W_i and W_{i+1} denote weights of the network at two consecutive instants, η (> 1) is the learning rate and d is the derivative of the error function.

$$W_{i+1} = W_i - \eta d \quad (6)$$

A weight update rule based on quadratic approximation, an improvement over the simple gradient descent learning is given by (7),

$$W_{i+1} = W_i - H^{-1} d \quad (7)$$

where H denotes an approximation to the Hessian of the error matrix. However quadratic approximation assumes the error function to be a linear function of weights which is true only near a minimum. The Levenberg technique involves blending of the above two techniques. The gradient descent method is used until we approach a minimum, and then quadratic rule is followed. Let λ be a blending factor that determines the mix between gradient descent and quadratic approximations. The update rule is

$$W_{i+1} = W_i - (H + \lambda I)^{-1} d \quad (8)$$

where I is the identity matrix. If λ is too small then it approaches quadratic approximation and if λ is too large then it approaches gradient descent learning. If by doing an update of weights using the above approach, the error is increased, the weights are reset and λ is increased by a significant factor as increase in error indicates that we are not near a minimum. If error is decreasing, that is we are getting closer to minimum, then we decrease λ . If λ is high and we are essentially doing gradient descent, to get benefit from the Hessian matrix, we should move further in directions in which the gradient is smaller. Using this concept, the Levenberg-marquardt (LM) optimization [23-24] technique for weight adaptation can be stated as (9), where $Diag [H]$ denotes the diagonal of the Hessian.

$$W_{i+1} = W_i - (H + \lambda diag[H])^{-1} d \quad (9)$$

The initial value of λ is taken as 0.001 in the present work and the increase and decrease factors of λ are taken as 10 and 0.1 respectively. The minimum value of the error gradient reaching which the training of the network is stopped is taken as 1.0e-6.

3) Bayesian Regularization

One approach to improve generalization in neural networks is to add constraints with the typical mean squared error objective function. The merit of such generalization lies in improving the degree of smoothening of the objective function, and such process is referred to as regularization.

Bayesian regularization provides neural network training in a Bayesian statistical framework. This framework assumes the network weights to be random variables and attempts to maximize the conditional probability of the weights when the data is given using Bayes' rule. If E_w be the sum of the square of network weights necessary for regularization and E_D be the squared error norm, the problem is to minimize the objective function $F(w) = \beta E_D + \alpha E_w$, [32] where α and β are

parameters of the objective function that depend on the variance of the weights and the measurement noise respectively.

The main steps of Bayesian regularization [24-26], [32] (BR) with Gauss-Newton approximation to Hessian matrix is given below.

1. Initialize $\alpha=0$ and $\beta=1$ following Nguyen-Widrow initialization method.
2. Execute one step of Levenberg-Marquardt algorithm to minimize $F(w)$.
3. Compute the effective number of parameters $\gamma = N - 2\alpha \text{tr}(H^{-1})$, where N is the total number of parameters in the network, $H = \nabla^2 F(w) = 2\beta J^T J + 2\alpha I_N$, where I_N is the identity matrix of $N \times N$, and J is the Jacobian of the training set errors.
4. Compute α and β by $\alpha = \gamma / 2E_w(w)$ and $\beta = (n - \gamma) / 2E_D(w)$, n depends on the number of samples and the network model like the number of layers and neurons in each layer.
5. Repeat through step 2 until convergence in $F(w)$ is observed.

In BR the blending factor of the LM rule is taken as 0.005 with the increase and the decrease factors being 10 and 0.1 respectively in the present work. Training is carried till a minimum error gradient of $1e-10$ is achieved.

4) Scaled Conjugate Gradient

Conjugate Gradient methods use the second derivative of the error function and find a better way to the minimum error than first order techniques, at a higher computational cost. It is called ‘conjugate’ because, unlike the normal gradient descent that proceeds in the direction of the gradient, it proceeds in the conjugate direction to the directions of the previous steps. Conjugate gradient method use the principle of line search that is computationally expensive. In Scaled Conjugate Gradient [27] (SCG) method, the expensive line-search and computation of Hessian of the error function is avoided using two parameters, σ , that determines change in weight for second derivative approximation and λ , that controls the indefiniteness of the Hessian. In the present work these values have been taken as $5.0e-5$ and $5.0e-7$ respectively. The minimum value of the error gradient reaching which the training of the network is stopped is taken as $1.0e-6$.

D. Overview of proposed methodology

The schematic of entire course of work is presented in Fig. 3, Fig. 4 and Fig. 5. The techniques of EEG signal acquisition and pre-processing as well as Kinect data acquisition are described in Section III. The first, second and third level Back Propagation neural networks (BPNNs) are termed BPNN1, BPNN2 and BPNN3 henceforth. Fig. 3 illustrates the training of BPNN1 and BPNN2 by the EEG features of the occipital-parietal and parietal-motor cortex regions respectively. The

occipital EEG features are given as input to BPNN1 and the corresponding set of parietal EEG features is set as target outputs. The weights of the network are adapted until it learns to predict the parietal EEG features from that of occipital EEG.

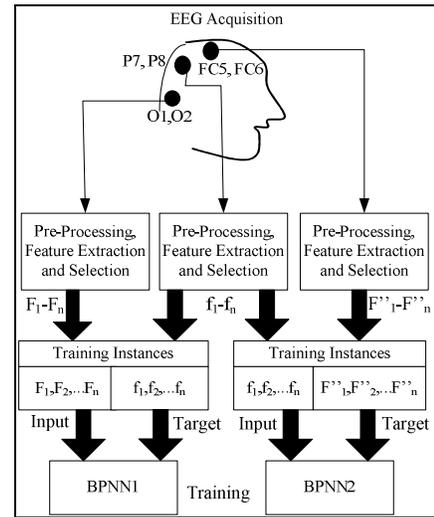


Fig. 3. Training phase to determine coordination between Occipital to parietal and parietal to motor cortex EEG features, Occipital Parietal and Motor Cortex EEG channels are denoted by O1, O2; P7, P8 and FC5 and FC6 respectively

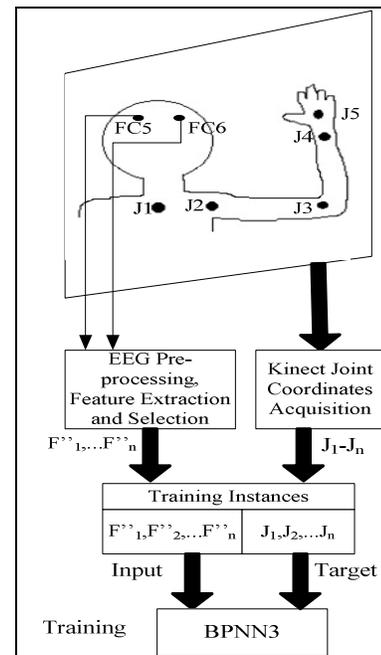


Fig. 4. Training phase to generate joint coordinates from Motor Cortex

All the techniques of weight adaptation previously discussed are applied to obtain a set of trained BPNNs for each algorithm. Similar procedure is carried out for BPNN2 as well. In Fig. 4 the training of BPNN3 taking the motor cortex EEG features as the input and the joint co-ordinate features as the targets has been illustrated. 5 significant joints spanning from the shoulder to the palm of a hand are considered in this present work. Each joint is represented by 3-dimensional joint

co-ordinates. The complete flow for testing the proposed system using the three trained BPNNs is illustrated in Fig. 5. The predicted parietal EEG features (Y_{PAR}) obtained from trained BPNN1 are supplied to BPNN2 from where the predicted Motor Cortex features (Y_{MC}) are obtained. Y_{MC} when supplied to the trained BPNN3 produce the predicted joint co-ordinates. The process carried out for predicting the co-ordinates of each joint separately using BPNN3s trained in accordance.

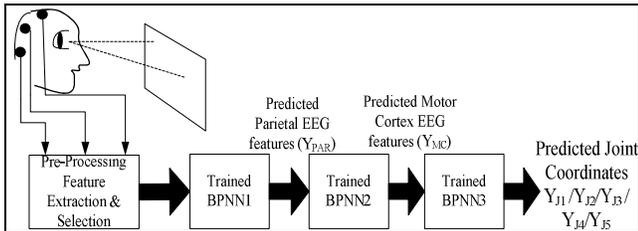


Fig. 5. Junction coordinates generation from predicted motor cortex EEG features

III. EXPERIMENTS

A. Experimental Paradigm

1) EEG Acquisition and Pre-processing

Experimental data is acquired from 8 right handed subjects, 3 male and 5 female, in the age group 25 ± 5 years, with their consent, over a period of 5 days. EEG is acquired using a 14 channel wireless Emotiv headset [28] which has a sampling rate 128Hz. The placement of electrodes follows the standard 10/20 system of electrode placement [11] and is shown in Fig. 6(a). EEG signals from occipital, parietal and motor cortex are further processed, acquired from three electrode pairs, i.e. O1-O2; P7-P8 and FC5-FC6 (Fig. 6(a)). The final joint co-ordinates are generated from the predicted FC5-FC6 EEG signals. Though more electrodes can produce better results, we consider only these three electrode pairs as we have to demonstrate the processing according to the natural neural pathway.

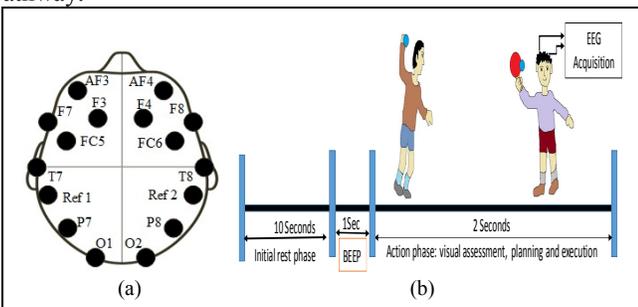


Fig. 6. (a) Electrode Placement for EEG Acquisition (b) Experimental Paradigm

Data acquisition is done following the paradigm illustrated by Fig. 6(b). There is an initial rest phase of 10 seconds. After a beep sound to declare the start of acquisition phase a person throws a ball towards the subject who is standing at a fixed spot at an appropriate distance. The subject watches the ball, plans his/her action and ultimately hits the ball with the bat. During this time EEG is acquired from him/her. The process

approximately takes 2 seconds of time. This throw and hit process is continued for 10 times, on each subject.

EEG has a wide frequency spectrum ranging from distinct bands [11]. The stimuli considered in the experiments are known to produce significant response limited to the frequency range of 4-30 Hz. To extract the EEG signals in the desired frequency range and thereby eliminate the other frequencies, an Elliptical Band pass filter of order 6 with 1dB passband ripple and 50 dB stopband ripple in the bandwidth 4-30Hz has been used. Spatial filtering by Common average referencing [11] has been performed on the filtered EEG signals to remove the interference in between channels. For each EEG channel, all the channels equally weighted are subtracted to eliminate the commonality of that channel with the rest and preserve its specific temporal features.

2) Joint Co-ordinate Data using Kinect Sensor

The Kinect [15], is a sensor device with set of IR and RGB camera that appears as a long horizontal bar with a motorized base as shown in Fig. 7(a). It detects the 3D image of an object and tracks the skeleton of the person standing in front of it within a finite amount of distance. The Kinect sensor with the help of the corresponding Software Development Kit (SDK) senses the skeleton and the body postures irrespective of the color of the skin or the individual's dress. In Fig. 7(b) a skeleton generated by the Kinect has been shown. The Kinect Sensor produces the human skeleton represented by twenty body joints in the 3-D space. Out of these twenty joints, 5 joints of the right hand are useful for the present work, denoted by yellow circles and marked, as there is no significant information from the rest of the body parts for hitting a ball by a right handed subject while standing at a fixed position. These joints are shoulder center (J1), shoulder right (J2), elbow right (J3), wrist right (J4) and hand right (J5).

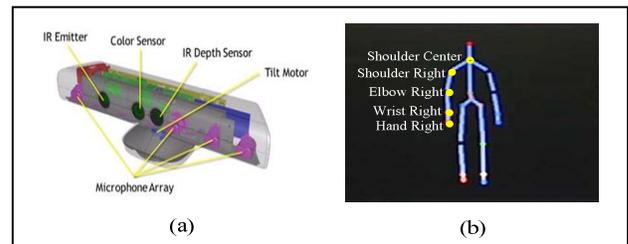


Fig. 7. (a) The Kinect Sensor (b) Full Body Skeleton acquired from the Kinect

The Kinect produces skeletons at a rate of 30 frames/second, and from each frame 3 coordinates specifying each of the 5 joints. Therefore for an instance of 2 seconds duration for each joint, $2 \times 30 \times 3 (=180)$ coordinate points are obtained that are taken as a single instance of joint co-ordinates for a similar 2 seconds EEG feature space of wavelet coefficients. All data after acquisition is normalized for each joint in the range $[-1, 1]$.

B. Feature Extraction and Selection

Wavelet Approximate (A3) and Detail (D3) Coefficients of level 3 decomposition with Daubechies order 4 mother wavelet has been used as EEG features according to the explanation in Section II. The order of decomposition is chosen by the fact

that the EEG signals are acquired at 128Hz and as the region of interest is 4-30Hz the third level approximation and detail coefficients provide the best estimate of the EEG signals. Level 2 Approximate Coefficients (A2) could also have been used but the use of A3 and D3 together produce a larger feature space and hence a better representation. Table I provides the decomposition of the EEG signal of sampling frequency 128 Hz into respective frequency bands during wavelet decomposition.

The time window for EEG extraction in all the three levels is kept at 2s. Experiments are conducted by taking smaller window lengths and concatenating the features obtained, however such methods provide no significant performance improvement but increase the computational complexity. The dimension of each feature vector for an EEG instance of 2 seconds duration is 276. However, the dimension of an instance of Joint co-ordinate data obtained from 2 seconds duration for each joint has dimensions 180. In order to keep parity in the dimensions and reduce the feature size of EEG data, the 180 best EEG features have been selected in each experiment by PCA. The feature space is normalized in the range [-1,1].

TABLE I. DECOMPOSITION OF EEG SIGNAL OF 128 HZ SAMPLING FREQUENCY

Frequency Range (Hz)	Wavelet Coefficient
64-128	D1
32-64	D2
16-32	D3
0-16	A3

C. Regression Analysis using BPNN

The Neural Networks are implemented varying the number of neurons in the intermediate layer from 3 to 12 and the best average performance is noted for 10. Back propagation learning is used with different weight adaptation techniques discussed before. In the training phase, data is crossvalidated to provide each neural network with 70% training data and 15% data each for validation and testing purposes. The performance is determined from linear regression analysis [29-30] to predict the respective targets from the corresponding input values. The slope (m) and the y-intercept of the best linear regression relating targets to network outputs (b), the mean squared error (MSE) and correlation coefficient (R) between the targets and the outputs are evaluated. For a perfect fit between the outputs and the targets, the slope would be 1, the y-intercept would be 0 and the correlation coefficient would be 1. The value of the correlation coefficient determines how well the variations in the inputs are correctly produced in the targets. The value of the MSE should be as low as possible, and less than 1.

IV. RESULTS AND DISCUSSIONS

The results of regression analysis for mapping the features of the occipital → parietal, parietal → motor cortex and finally motor cortex → joint co-ordinates have been described here.

Fig. 8 (a), (b) and (c) graphically illustrate the output provided by regression analysis of the neural networks BPNN1, BPNN2 and BPNN3 for a particular case using LM

optimization for weight adaptation. The network outputs (predicted EEG features) are plotted versus the targets (actual EEG features) as open circles. The best linear fit is indicated by a dashed line. The perfect fit (output equal to targets) is indicated by the red solid line. Here, it is difficult to distinguish the best linear fit line from the perfect fit line in (a) and (b) because the fit is so good as indicated by the values of the slope ($m=0.9396$ for BPNN1 and $m=0.9327$ for BPNN2), the y-intercept ($b= -0.0426$ for BPNN1 and $b= -0.0441$ for BPNN2) and the correlation coefficient ($R=0.9600$ for BPNN1 and $R=0.9638$ for BPNN2). The performance of the BPNN3 is not as good, with $m=0.46$, $b=-0.097$ and $R=0.7135$.

A single execution of any algorithm takes the data from the i^{th} day of each subject for training and each of the data from the other days (total 5days) for testing separately and computes the mean over days for a subject and then over all subjects. Tables II, III, IV and V provide the performances in terms of Mean Squared Error (MSE), slope (m) and the y-intercept (b) of the best linear fit and correlation coefficient (R) respectively. In each Table, the mean results are reported for 50 independent executions of each algorithm, along with the standard deviation in parenthesis. J_i denotes the Joint index according to Section III A 2.

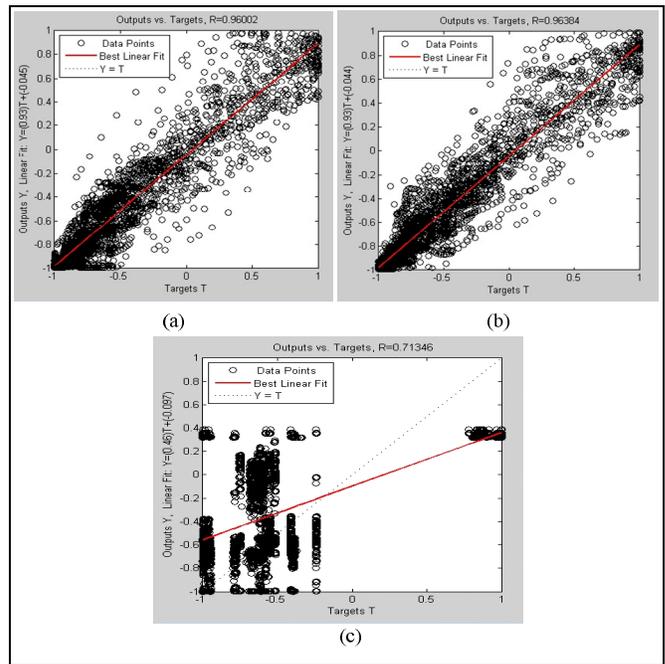


Fig. 8. Results of regression analysis of (a) BPNN1, (b) BPNN2 and (c) BPNN3 (for J_1) for a particular testing sample

From Table II it is observed that the mean values as well as the standard deviations of MSE are significantly low in case of BPNN1 and BPNN2, however, the MSEs increases on BPNN3, which is theoretically justified because BPNN3 tries to predict joint coordinates from EEG features. On an average, LM algorithm performs the best over all the three levels of BPNNs and BR is the second best. From Table III also it is concluded that LM performs best among the rest of the algorithms, in terms of the slope of the best linear fit.

TABLE II. PERFORMANCE OF BPNNs : MSE BETWEEN TARGETS AND OUTPUTS OF THE NEURAL NETWORKS

Algorithm		<i>GDS</i>	<i>M-GDS</i>	<i>LM</i>	<i>BR</i>	<i>SCG</i>
BPNN1		0.565 (0.089)	0.287 (0.063)	0.086 (0.005)	0.045 (0.008)	0.156 (0.012)
BPNN2		0.655 (0.025)	0.501 (0.027)	0.088 (0.012)	0.093 (0.006)	0.323 (0.032)
BPNN3	J1	0.961 (0.148)	0.678 (0.056)	0.523 (0.017)	0.578 (0.023)	0.623 (0.045)
	J2	0.966 (0.090)	0.822 (0.075)	0.536 (0.045)	0.665 (0.019)	0.789 (0.053)
	J3	0.825 (0.155)	0.878 (0.021)	0.632 (0.011)	0.552 (0.025)	0.0534 (0.055)
	J4	0.989 (0.085)	0.753 (0.083)	0.678 (0.026)	0.631 (0.065)	0.725 (0.059)
	J5	0.786 (0.091)	0.891 (0.021)	0.673 (0.034)	0.680 (0.042)	0.721 (0.076)

TABLE III. PERFORMANCE OF BPNNs : SLOPE OF THE BEST LINEAR FIT BETWEEN TARGETS AND OUTPUTS OF THE NEURAL NETWORKS

Algorithm		<i>GDS</i>	<i>M-GDS</i>	<i>LM</i>	<i>BR</i>	<i>SCG</i>
BPNN1		0.753 (0.072)	0.917 (0.035)	0.928 (0.032)	0.938 (0.011)	0.812 (0.064)
BPNN2		0.659 (0.053)	0.851 (0.025)	0.951 (0.014)	0.912 (0.026)	0.764 (0.075)
BPNN3	J1	0.451 (0.065)	0.438 (0.086)	0.625 (0.013)	0.478 (0.173)	0.633 (0.090)
	J2	0.432 (0.055)	0.561 (0.052)	0.586 (0.057)	0.568 (0.051)	0.554 (0.103)
	J3	0.520 (0.153)	0.468 (0.088)	0.479 (0.093)	0.574 (0.085)	0.547 (0.076)
	J4	0.477 (0.059)	0.562 (0.093)	0.671 (0.089)	0.585 (0.075)	0.348 (0.098)
	J5	0.512 (0.091)	0.392 (0.105)	0.468 (0.064)	0.623 (0.145)	0.458 (0.112)

TABLE IV. PERFORMANCE OF BPNNs : Y-INTERCEPT OF THE BEST LINEAR FIT BETWEEN TARGETS AND OUTPUTS OF THE NEURAL NETWORKS

Algorithm		<i>GDS</i>	<i>M-GDS</i>	<i>LM</i>	<i>BR</i>	<i>SCG</i>
BPNN1		-0.068 (0.062)	-0.059 (0.054)	-0.015 (0.087)	-0.024 (0.021)	-0.048 (0.007)
BPNN2		-0.072 (0.054)	-0.065 (0.036)	-0.018 (0.042)	-0.025 (0.067)	-0.035 (0.015)
BPNN3	J1	-0.088 (0.039)	-0.077 (0.108)	-0.074 (0.133)	-0.081 (0.065)	-0.053 (0.073)
	J2	-0.075 (0.112)	-0.086 (0.075)	-0.069 (0.068)	-0.074 (0.232)	-0.089 (0.059)
	J3	-0.091 (0.098)	-0.092 (0.145)	-0.073 (0.077)	-0.085 (0.058)	-0.075 (0.051)
	J4	-0.095 (0.023)	-0.065 (0.086)	-0.075 (0.117)	-0.063 (0.092)	-0.091 (0.062)
	J5	-0.089 (0.056)	-0.081 (0.059)	-0.073 (0.032)	-0.074 (0.137)	-0.082 (0.033)

A close observation of the results of Table IV indicate that the best mean value of the y-intercept is obtained using BR algorithm, followed by LM as an average over all the BPNNs. From Table V it is observed that the maximum mean values of the correlation coefficient occur for LM algorithm.

In Table VI we present the statistical significance level (SS) of the difference of the mean of the best two algorithms using

t-test of 25 samples [31]. Here “+” indicates that the t value of 49 degrees of freedom is significant at a 0.05 level of significance by two-tailed test, whereas “-” means the difference of mean is not statistically significant, and “NA” stands for not applicable, covering cases for which two or more algorithms achieve the best accuracy results. The best algorithm is marked in bold. To determine the best algorithm, the mean values of MSE, slope of best linear fit, y-intercept of best linear fit and correlation coefficient should respectively be minimum, maximum, nearer to zero and maximum. These four parameters are averaged over all the three levels of BPNNs over the independent runs. The mean values and standard deviations of the parameters are given in Table VI. LM turns out to be superior to the rest of the algorithms in terms for 3 out of the four parameter cases in a statistically significant manner, while BR has the second best performance.

TABLE V. PERFORMANCE OF BPNNs : CORRELATION COEFFICIENT BETWEEN TARGETS AND OUTPUTS OF THE NEURAL NETWORKS

Algorithm		<i>GDS</i>	<i>M-GDS</i>	<i>LM</i>	<i>BR</i>	<i>SCG</i>
BPNN1		0.782 (0.039)	0.891 (0.073)	0.956 (0.017)	0.975 (0.015)	0.856 (0.010)
BPNN2		0.745 (0.020)	0.882 (0.043)	0.961 (0.019)	0.966 (0.019)	0.899 (0.040)
BPNN3	J1	0.558 (0.028)	0.591 (0.077)	0.853 (0.027)	0.778 (0.020)	0.523 (0.055)
	J2	0.469 (0.097)	0.568 (0.068)	0.738 (0.029)	0.628 (0.031)	0.897 (0.087)
	J3	0.721 (0.235)	0.439 (0.034)	0.832 (0.082)	0.763 (0.126)	0.772 (0.089)
	J4	0.556 (0.065)	0.720 (0.078)	0.765 (0.036)	0.882 (0.058)	0.683 (0.072)
	J5	0.534 (0.072)	0.662 (0.065)	0.822 (0.075)	0.675 (0.034)	0.596 (0.041)

TABLE VI. STATISTICAL SIGNIFICANCE FOR MEAN VALUES OF PARAMETERS OVER ALL BPNNs

Parameter	<i>GDS</i>	<i>M-GDS</i>	<i>LM</i>	<i>BR</i>	<i>SCG</i>	SS
MSE	0.850 (0.032)	0.695 (0.043)	0.432 (0.055)	0.487 (0.078)	0.584 (0.052)	+
Slope best linear fit	0.553 (0.071)	0.598 (0.062)	0.773 (0.023)	0.758 (0.067)	0.588 (0.042)	+
y-intercept of best linear fit	-0.097 (0.052)	-0.085 (0.031)	-0.044 (0.054)	-0.041 (0.005)	-0.059 (0.009)	-
Correlation coefficient	0.623 (0.012)	0.684 (0.033)	0.856 (0.026)	0.824 (0.58)	0.735 (0.042)	+

The present work proposes a three level ANN framework for predicting the joint co-ordinates of the hand from occipital EEG responses. The idea behind such an architecture is to provide an alternative that mimics the natural neural pathway of visual-motor co-ordination that occurs in three stages: occipital→ parietal→ motor cortex→ joint movement. However in an attempt to reduce computational complexity a single level ANN that predicts joint co-ordinates directly from occipital EEG features without using the parietal and motor cortex EEG responses is studied. This ANN is also based on Back propagation learning and different weight adaptation strategies are employed. However the performance of such an ANN produce MSE values greater than 1 in many cases clearly

indicating that a single level ANN is not suitable for this purpose.

V. CONCLUSIONS AND FUTURE DIRECTIONS

The present work proposes a novel technique of predicting the joint coordinates during hand movement in response to a visual stimulus by developing an artificial neural network based strategy. This work finds applications in the development of a rehabilitative aid for circumventing the natural path of visual-motor coordination through the occipital-parietal-motor cortex regions of the brain based on EEG analysis. The neural networks are trained using normalized features and hence in case of abnormal/unavailable parietal and motor cortex responses, these pre-trained neural networks from data of healthy subjects can be used to predict these signals in suitable patients. A number of different weight adaptation techniques for back propagation based learning of the neural networks has been evaluated and compared. Future applications of the work include the movement of a robot arm in real time conditions using the joint co-ordinates generated by the proposed method. Work is being done to implement a closed loop system that will provide the necessary feedback signals to overcome the errors in the movement of the robot arm while being controlled by the EEG of a patient.

ACKNOWLEDGMENT

This study has been supported by University Grants Commission, India, University of Potential Excellence Program (UGC-UPE) (Phase II) in Cognitive Science, Jadavpur University and Council of Scientific and Industrial Research (CSIR), India.

REFERENCES

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control. *Clinical neurophysiology*," vol. 113, no. 6, pp.767-791, 2002.
- [2] A. Vuckovic, V. Radivojevic, A. C. Chen, and D. Popovic, "Automatic recognition of alertness and drowsiness from EEG by an artificial neural network," *Medical Engineering & Physics*, vol. 24, no. 5, pp. 349-360, 2002.
- [3] A. Gevins, and M. E. Smith, "Detecting transient cognitive impairment with EEG pattern recognition methods," *Aviation, space, and environmental medicine*, vol. 70, no. 10, pp. 1018-1024, 1999.
- [4] G. Pfurtscheller, and C. Neuper, "Motor imagery and direct brain-computer communication," *Proceedings of the IEEE*, vol. 89, no. 7, pp.1123-1134, 2001.
- [5] C. Guger, W. Harkam, C. Hertnaes, and G. Pfurtscheller, "Prosthetic control by an EEG-based brain-computer interface (BCI)," in *Proc. aaate 5th european conference for the advancement of assistive technology*, pp. 3-6, November 1999.
- [6] K. Schaaff, and T. Schultz, "Towards an EEG-based emotion recognizer for humanoid robots," in the 18th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2009, pp. 792-796, September 2009.
- [7] A. Khasnobish, A. Konar, D. N. Tibarewala, S. Bhattacharyya, and R. Janarthanan, "Object Shape Recognition from EEG Signals during Tactile and Visual Exploration," in *Pattern Recognition and Machine Intelligence*, Springer Berlin Heidelberg 2013, pp. 459-464.
- [8] E. Bruner, and H.I. Jacobs, "Alzheimer's disease: the downside of a highly evolved parietal lobe?," in *J Alzheimer's Disease*, vol. 35, no.2, pp. 227-40, 2013.
- [9] M. Rizzo, "Balint's syndrome and associated visuospatial disorders," in *Baillieres Clinical Neurology*, vol. 2, no. 2, pp. 415-437, 1993.
- [10] A. Hasan, T. Bergener, M.A. Nitsche, W. Strube, T. Bunse, P. Falkai, and T. Wobrock, "Impairment of motor cortex responses to unilateral and bilateral direct current stimulation in schizophrenia," in *Frontiers in Psychiatry*, vol. 4, 2013.
- [11] G. Dornhege, *Towards Brain-Computer Interfacing*, MIT Press, 2007.
- [12] A. Nijholt, and D. Tan, "Brain-computer interfacing for intelligent systems," *IEEE Intel. Sys.*, vol. 23, no. 3, pp.72-79, May-June 2008.
- [13] Y. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, "A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair," *IEEE Trans. Neural Sys. Rehab. Eng.*, vol. 20, no. 5, pp. 720-729, Sept. 2012.
- [14] Y. Chae, J. Jeong, and S. Jo, "Toward brain-actuated humanoid robots: Asynchronous direct control using an EEG-based BCI," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1131-1144, 2012.
- [15] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo, "Kinect identity: Technology and experience," *Computer (Long. Beach. Calif.)*, vol. 44, no. 4, pp. 94-96, 2011.
- [16] A. Turnip, H. Keum-Shik and S.S. Ge, "Backpropagation neural networks training for single trial EEG classification," in *29th Chinese Control Conference (CCC)*, pp. 2462-2467, 2010.
- [17] R. Roy, A. Konar, D.N. Tibarewala and R. Janarthanan, "EEG driven model predictive position control of an artificial limb using neural net," in *Third International Conference on Computing Communication & Networking Technologies (ICCCNT)*, pp. 1-9, 2012.
- [18] P. Jahankhani, V. Kodogiannis, K. Revett, "EEG Signal Classification Using Wavelet Feature Extraction and Neural Networks," in *IEEE John Vincent Atanasoff International Symposium on Modern Computing*, pp.120-124, 2006.
- [19] S. Qin, and Z. Ji, "Multi-Resolution Time-Frequency Analysis for Detection of Rhythms of EEG Signals," *IEEE Proceedings on Signal Processing, 11'th International DSP Workshop*, vol. 4, pp. 338-341, 2004.
- [20] K. Mahajan, M.R. Vargantwar, and S.M. Rajput, "Classification of EEG using PCA, ICA and neural networks," *Int. J. Engg. and Advanced Tech.*, vol. 1, no. 1, pp. 80-83, 2011.
- [21] C. Bugli, and P. Lambert, "Comparison between independent component analysis and principle component analysis in electroencephalogram modelling," *Biometrical Journal*, vol. 48, no. 5, pp. 1-16, 2006.
- [22] A. Konar, "Computational Intelligence. Principles, Techniques and Applications", Springer, 2005.
- [23] S. Roweis, "Levenberg-marquardt optimization," *Univ. of Toronto*, 1996.
- [24] M.T. Hagan, and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Trans. on Neural Nets*, vol. 5, no. 6, pp. 989-993, 1994.
- [25] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415-447, 1992.
- [26] D.J.C. MacKay, "A practical Bayesian framework for Backpropagation Networks," *Neural Computation*, vol. 4, no. 3, pp. 448-472, 1992.
- [27] M.F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *IEEE Trans. Neural Networks*, vol. 6, pp. 525-533, 1993.
- [28] <http://www.emotiv.com/>
- [29] D. C. Montgomery, E.A. Peck, and G. Vining, "Introduction to linear regression analysis", Vol. 821, Wiley, 2012.
- [30] O. Rudovic, I. Patras and M. Pantic, "Regression-Based Multi-view Facial Expression Recognition," in *International Conference on Pattern Recognition (ICPR)*, pp. 4121-4124, 2010.
- [31] Pavel Bhowmik, Pratyusha Rakshit, Amit Konar, Eunjin Kim, Atulya K. Nagar, "DE-TDQL: An adaptive memetic algorithm," *IEEE Congress on Evolutionary Computation*, pp. 1-8, 2012.
- [32] F. Dan Foresee, and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *International Conference on Neural Networks*, vol. 3, pp. 1930-1935, 1997.