USING AUTOENCODERS FOR FEATURE ENHANCEMENT IN MOTOR IMAGERY BRAIN-COMPUTER INTERFACES

Mahmoud A. Helal¹, Seif Eldawlatly², Mohamed Taher³
Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University
¹ 6434@eng.asu.edu.eg ² seldawlatly@eng.asu.edu.eg ³ mohamed.taher@eng.asu.edu.eg

ABSTRACT
Motor imagery is currently one of the main applications of Brain-Computer Interface (BCI) which aims at providing the disabled with means to execute motor commands. One of the major stages of motor imagery systems is reducing the dimensions of the input data and enhancing the features prior to applying a classification stage to recognize the intended movement. In this paper, we utilize autoencoders as a powerful tool to enhance the input features of the band power filtered electroencephalography (EEG) data. We compare the performance of the autoencoder-based approach to using Principal Component Analysis (PCA). Our results demonstrate that using autoencoders with non-linear activation function achieves better performance compared to using PCA. We demonstrate the effects of varying the number of hidden nodes of the autoencoder as well as the activation function on the performance. We finally examine the characteristics of the trained autoencoders to identify the features that are most relevant for the motor imagery classification task.

KEY WORDS
Neural Rehabilitation, Pattern Recognition and Soft Computing Techniques, Brain-Computer Interface

1. Introduction
Brain Computer Interface (BCI) incur a variety of promising applications for patients suffering from motor disabilities [1]. It provides solutions ranging from controlling wheelchair movement to communicating with spelling devices. BCI represents a tool that enables users to interact with the outside world and a chance to be part of it.

One of the major applications of BCI is Motor imagery (MI) which aims at translating subject's EEG activity to motor commands [2]. This is typically done by detecting the changes in frequency-band power of the electroencephalogram (EEG) signals recorded over the motor cortex area. For instance, right and left hand movements can be characterized by observing the increase in the power of Mu, Beta and Gamma frequency bands or the decrease in the power of the same frequency bands which are termed as event-related synchronization (ERS) and event-related desynchronization (ERD), respectively [3].

A multitude of methods and algorithms have been proposed to be used in motor imagery recognition systems. Such system is typically composed of multiple stages including signal pre-processing, feature extraction, dimensionality reduction and, finally, a classification stage. Methods such as wavelet and Fourier transform were used for feature extraction [4]. For dimensionality reduction and feature enhancement, methods such as Common Spatial Pattern (CSP) and independent component analysis (ICA) were also proposed [5]. Support vector machine (SVM) and linear discriminant analysis (LDA) were used in classification [6].

In this paper, we demonstrate the use of an autoencoder-based technique for dimensionality reduction and feature enhancement. An autoencoder (AE) is a simple neural network that consists of three main layers: input layer, hidden layer and output layer [7]. The goal of the autoencoder is to train the network with the target values set to be equal to the inputs through a reduced number of dimensions. AEs are currently in wide use in deep learning applications. They have been utilized in BCI for feature extraction and classification [8, 9]. We demonstrate the use of AEs in reducing and enhancing features extracted from the frequency-domain representation of EEG motor imagery data. We compare the performance of the proposed approach to using principle component analysis (PCA) for feature enhancement.

2. Methods
The block diagram for the system, as depicted in Figure 1, consists of 5 main elements as detailed below.

2.1 Dataset
The dataset used in this paper is BCI competition IV 2a which consists of four motor imagery tasks: left hand,

Figure 1. Overview of the proposed approach.
right hand, both feet, and tongue movements recorded from 9 subjects [10]. The dataset was recorded in two sessions. Each session consists of 6 runs with a total of 48 trials/run resulting in a total of 288 trials per session. We used one session to train the proposed approach and the other session for testing.

### 2.2 Pre-processing

EEG signals that correspond to the motor imagery task were extracted from the dataset by extracting data in the interval 3 to 6 sec of each trial. Signal pre-processing was then applied to remove signal artifacts by applying whitening, Common Average Reference (CAR), and Z-Score normalization to raw data before any further operation. In addition, down sampling is applied to the pre-processed signal to speedup subsequent processing. This resulted in a feature vector of 157 samples per trial for each EEG channel.

### 2.3 Feature Extraction

Feature extraction aims to extract information out of the pre-processed signal. In this work, we use the band-power method which consists of three stages: First, apply 4th order Butterworth infinite impulse response (IIR) band-pass filter to the input signal using the frequency bands 8-15 Hz, 12-21 Hz and 20-30 Hz [11]. Second, we square every sample in the output signal \( x(t) \) to obtain the time course of power \( p(t) \)

\[
p(t) = x^2(t), \tag{1}
\]

Third, the average power of previous samples in a window of size \( w \) is computed as

\[
p(n) = \frac{1}{w} \sum_{k=0}^{w} p(n-k), \tag{2}
\]

The final feature values used are equal to \( \log_{10}(p(n)) \).

### 2.4 Dimensionality Reduction Methods

The main contribution of this work is in enhancing the extracted features prior to applying the classification stage. To achieve this goal, we introduce an autoencoder-based approach and compare it to using Principal Component Analysis (PCA). Figure 2 illustrates the block diagram of the proposed approach. For each trial, we form a matrix of size 22 channels x 157 samples. Computing the power in the aforementioned three bands for each channel results in 3 matrices, each of size 22 x 157. We then form 157 vectors where each vector is of length 66 which corresponds to the filtered data of the 22 channels in the 3 frequency bands. We then perform dimensionality reduction to each of the 157 vectors. A classifier is then trained for each of the 157 samples. By applying a 10-fold cross-validation, we identify which of the 157 classifiers achieves the highest accuracy. When classifying input test data, this identified classifier is used.

#### 2.4.1 Principal Component Analysis

Principal component analysis (PCA) is a classical technique in dimensionality reduction [12]. The main goal of PCA is to project data on the minimum number of principal components that represent the maximum amount of variance in the data. The projected data thus take the form

\[
z_k = x \cdot w_k, \tag{3}
\]

where \( x \) is the data vector after x-score and \( w \) is first \( k \)-component which is calculated from the eigenvectors of the covariance matrix of the input data and sorted in descending order according to the corresponding eigenvalues.

#### 2.4.2 Autoencoder

An autoencoder (AE) aims at reconstructing its input on its output nodes [7]. This is achieved via a hidden layer of a number of nodes that is less than the number of inputs; thus, performing dimensionality reduction. An AE can be divided into two stages as shown in Figure 3: an encoder stage followed by a decoder stage, where the encoder aims to represent the input data using reduced dimensions. The output of each node of the hidden layer is represented by

\[
h = s(w \cdot x + b), \tag{4}
\]

where \( x \) is input data after pre-whitening and \( s \) is an activation function (linear or sigmoid). The decoder aims to reconstruct data from the compressed data. Therefore, to use autoencoder for dimensionality reduction, the output of the encoder stage is used as a representation of the input.
2.5 Classification

To identify which of the four motor imagery movements corresponds to each trial, Linear Discriminant Analysis (LDA) is utilized in the classification phase [13]. LDA aims to maximize separation between projected input vectors while minimizing the within-class-variance, which could be achieved by identifying the weight vector \( w \) that maximizes \( J(w) \) which takes the form

\[
J(w) = \frac{w^T S_b w}{w^T S_w w},
\]

where \( S_b \) is the between-class scatter matrix defined as

\[
S_b = \sum_{k=1}^{m} n_k (\mu_k - \mu)(\mu_k - \mu)^T,
\]

where \( m \) is number of classes, \( n_k \) is number of samples in class \( k \), \( \mu_k \) is mean of class \( k \) and \( \mu \) is mean of all samples. \( S_w \) is the within-class scatter matrix defined as

\[
S_w = \sum_{k=1}^{m} \sum_{x \in k} (x - \mu_k)(x - \mu_k)^T.
\]

To classify the input data into one of the four classes, four LDA classifiers are trained. Each classifier is trained to classify one of the four classes (left, right, both feet or tongue movement) versus the other classes. In the testing phase, for any input data, the label of the classifier that results in the highest value for the discriminant function is considered as the recognized class.

2.6 Evaluation

Both classification accuracy and Cohen’s kappa value were used in the evaluation of the approach [14]. Classification accuracy represents the percentage of correctly classified trials, while Cohen’s kappa value (or kappa for short) is calculated as

\[
k = \frac{P_o - P_e}{1 - P_e},
\]

where \( P_o \) is the classification accuracy and \( P_e \) is the hypothetical accuracy of a random classifier on the same data.

In the training phase of the classifiers, 10-fold cross-validation is performed in which 80% of the training trials were randomly selected for training while the remaining 20% were used for validation. The cross-validation phase aimed at identifying which of the 157 samples per trial depicted in Figure 2 achieves the highest validation kappa. The identified sample and its corresponding classifier were then used in the test phase. In addition, the cross-validation phase was used to identify the optimal number of hidden nodes of the AE and the optimal number of principal components for the PCA method for each subject.

3. Results

3.1 Autoencoder versus PCA using Optimal Parameters

We first compared the performance of the proposed AE-based approach to using PCA for dimensionality reduction. Figure 4 shows the kappa value obtained for each subject using both approaches. In this analysis, subject-dependent optimal parameters were used for each method (number of hidden neurons in AE and number of principal components in PCA). The figure indicates that using AE achieves slightly better performance compared to using PCA with an across-subject average kappa of 0.66 for AE and 0.65 average kappa for PCA. However, a significant increase in kappa was observed for the testing dataset where using AE achieved an average kappa of 0.55 (classification accuracy of 66%) while using PCA achieved 0.52 (classification accuracy of 64%).

3.2 Performance for Different Number of Dimensions

To examine the effect of the number of reduced dimensions on the performance, we compared the performance of AE to that of PCA for the same number of reduced dimensions. Figure 5 illustrates the performance for reduced dimensions in the range of 45 to 65 dimensions with a step of 5 averaged across subjects. Results demonstrate that the best performance could be achieved using 45 dimensions with a kappa of 0.56 (67% classification accuracy) compared to 0.52 (64% classification accuracy) using PCA. The large standard deviation can be attributed to the differences in the performance across subjects as previously illustrated in Figure 4. In addition, as the number of reduced dimensions increases above 55, the performance deteriorates. This suggests that the best number of reduced dimensions should be less than 55. The analysis also reveals that using AE is consistently better than using PCA for dimensionality reduction.

![Figure 4](image4.png)

Figure 4. Validation results when no dimensionality reduction is performed, using PCA and using AE (mean ± std).
3.3 Performance of Linear versus Sigmoid Activation Functions

We next examined the performance as we vary the activation function of the hidden layer of the AE. Figure 6 shows the performance when sigmoid activation function is used compared to linear activation function averaged across subjects. Results indicate that using sigmoid activation function achieves better performance compared to linear activation function which achieves a kappa of 0.53. This kappa value is close to that obtained using PCA which is consistent with other reports that indicated that an AE with linear activation function is similar to PCA [15].

3.4 Analysis of AE Weights

To better understand which features were more significant in the classification process, we examined the weights of the encoding stage of the autoencoder. We first computed the absolute value of the weights of the classifier and identified the AE hidden neuron that has the highest value. This represents the most important hidden neuron (reduced dimension) in the classification stage. We then computed the absolute value of all the weights connecting the 66 input features (corresponding to 22 channels x 3 bands) to this hidden neuron. These values were then averaged across all AEs trained for each subject. Figure 7 illustrates the values for all 22 channels. Results indicate that channels 9, 10 and 11 are the most important channels given that they have the highest absolute value of the weights. These channels correspond to EEG channels C1, C2 and C3 which cover the left, central and right areas of the motor cortex, respectively. This is expected given that EEG dynamics related to motor imagery are more prominent in these locations.

4. Conclusion

We demonstrated the utility of autoencoders in reducing the dimensions of EEG data in motor imagery tasks. Our results suggest that using autoencoders with sigmoid activation functions could achieve better performance compared to using linear PCA. Results also revealed that as the number of reduced dimensions increases beyond 55 dimensions, the accuracy decreases with best performance achieved using 45 dimensions. In addition, the weights analysis of the autoencoder indicated that central channels around the motor cortex area are the most critical channels in the feature enhancement process. Compared to the results achieved in the BCI competition IV, the performance obtained using our approach with kappa of 0.56 comes in second place, where the approach ranked first achieved a kappa of 0.57. This work could be extended by using features other than band power features such as wavelet features. In addition, employing deep autoencoder architectures (stacked and denoising autoencoders) could provide more insight into how to combine and enhance the input features for better classification accuracy. This could also enhance the significance of the results obtained using the autoencoder approach compared to linear PCA.

References


