A Stacked Autoencoders Approach for a P300 Speller BCI

Hamed Ghazikhani
Department of Computer Engineering
Ferdowsi University of Mashhad (FUM)
Mashhad, Iran
hamed.ghazikhani@mail.um.ac.ir

Modjtaba Rouhani
Department of Computer Engineering
Ferdowsi University of Mashhad (FUM)
Mashhad, Iran
rouhani@um.ac.ir

***Abstract*—This paper addresses a new approach through detecting the P300 and its application to the BCI speller systems. This research employed stacked autoencoders which is based on many autoencoders and a classifier that is regularly a Softmax. This deep structure, decrease the dimension of the data and eventually, the reduced features of the last autoencoder are passed to the Softmax classifier. Subsequently, the parameters of the network would be ameliorated through a fine-tuning phase. Chebyshev Type I, is employed for filtering the EEG signals and using them as an input to the deep neural network. Hyperparameters such as the number of neurons and layers are attained empirically. Therefore, the final structure of the proposed network is 420-210-100-50-20-10-2. To analyze the suggested structure, the second dataset of the third BCI Competition is employed. According to the results, this approach can willingly enhance the character recognition in the BCI speller systems. Thus, the best accuracy percentage according to this research, in an average manner, is 91.5% of both A and B subjects. Consequently, according to the achievements, this method can be comparable to the other state-of-the-art algorithms and, therefore, can improve the recognition rate in the BCI industry.**

***Index Terms*—Brain-Computer Interface (BCI), Event-Related Potential (ERP), P300, Deep Learning, Stacked Autoencoders**

I. Introduction

A Brain-Computer Interface (BCI) is a system which will help a person to control a device by using his brain signals. This system is the only way for those who suffer from the severe motor disabilities like spinal cord injuries, or amyotrophic lateral sclerosis (ALS) [1]–[5]. Through the non-invasive methods of recording the brain activities, the Electroencephalography (EEG) is a common way for the BCI systems which records the activities of the brain by a set of electrodes that are placed on the scalp using the International 10-20 system which is shown in Fig. 1. This method has several advantages such as easiness in use, and availability of equipment which practically convince the researchers to utilize this solution in most of the BCI systems [5]. Event-related potentials (ERPs) are sort of brain signals that will appear in special conditions. These signals are particular responses which are inadvertently and naturally happen to the mind. ERPs have several components which they have a specific meaning. One of the most important components of the ERPs is P300. This wave is a positive peak which will occurs 300 milliseconds after a specific external stimulus. There are a few types of stimulus which can be auditory, visual or somatosensory. So, an important type of BCIs, which have been used a lot in recent researches, is based on the P300 component of the ERP [6]–[8]. Consequently, this system is called P300-based BCI. Hence, the issue is to classify the P300 among all waves and components of brain signals. As it is known, this step is difficult enough because of the noisy data, movements of the subject, fluctuations in the power line, etc. [6]. In this paper, the purpose is to improve the detection of P300 waveform with higher accuracy in a P300 speller BCI.

P300 detection is a machine learning task. Accordingly, this waveform can be recognized by doing pre-processing, feature extraction, feature selection, and an appropriate classification method. Most of the works in the pre-processing section is to improve the signal-to-noise ratio. A common pre-processing method is bandpass filtering that a specific range of frequencies which are used to find the P300 component would be held. Moreover, averaging is a general method while needing to reduce the noise [9]. There are lots of different feature extraction methods in time and frequency domains, such as PCA, ICA, Wavelet transform, Genetic algorithms, etc. [1].

Next is classification methods which most of the state-of-the-art methods in the field of P300 speller BCI are based on conventional ones. In the last decade, researchers have applied the machine learning algorithms to detect the events more accurately. One of the most powerful machine learning algorithms which are a binary classification method is SVM (Support Vector Machine). Using SVM attains many successful results in the field of ERP extraction and in general in EEG signals [10]. Very famous research is what Alain et al. [11] did. They divided the whole data into 17 parts and then train 17 SVMs on each part. It is obvious that they have used a very elaborate method, although, they became the winner of the BCI Competition III. Besides, LDA algorithms have been used a lot in this domain for example in BCI systems that work in motor imagery, P300 speller, etc. [12]. Farwell and Donchin implemented the stepwise discriminant analysis (SWDA) algorithm using the peak picking and covariance analysis of the data [7]. There is also some research about the comparison of the classification methods, nevertheless, they did not show a perfect result on the same data set to help the researchers to have better evaluation

978-1-5386-9569-2/18/$31.00 ©2018 IEEE

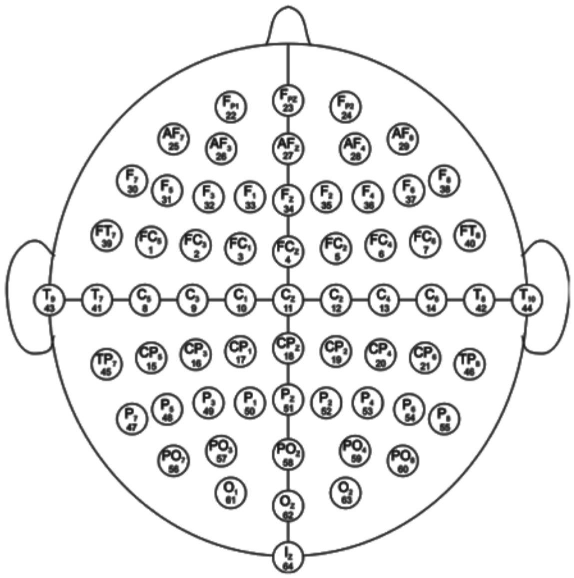

Fig. 1: 64 channel EEG using International 10-20 system [5].

and understanding of the problems. According to one study in conventional methods, SWDA and Fisher's linear discriminant (FLD) provided much better than other prominent algorithms such as Pearson's correlation method (PCM), a linear support vector machine (LSVM), and a Gaussian kernel support vector machine (GSVM). Consequently, they have achieved that most of the errors are related to some factors such as human error, adjacency effect, or mental fatigue [8].

The nature of the EEG signals is full of noise. This characteristic is because of the situation that the electrodes get the data on the scalp. While it has a lot of noise the traditional models are suitable for other problems that do not suffer huge amounts of outliers. The conventional feature extraction algorithms cannot avoid the manual modification of the parameters for the training model under different signal-to-noise ratios. Furthermore, the existing methods of the feature extraction could be failed because of the needing to an expert would be required. However, due to the variability of the ERP waveform, differences in the trigger source phases of the recorders, and the presence of various noise interference, the research studies regarding automatic feature extraction and selection methods remain challenging. Hence, Because of the characteristic of the EEG signals and noisy recording environment, the EEG signals may often be neglected if no proper denoising algorithms or ERP detection techniques are applied. According to the above explanations, an algorithm is needed to handle the noise and the feature extraction while it is learning from the representation of the data.

Recent development in the field of deep learning has opened new research possibilities regarding computer vision and natural language processing. Deep learning as one of the most crucial Machine Learning algorithms is promising in various domains which the model is learning a new representation of the data [13]. Accordingly, Deep learning models are suitable for complex and high dimensional data. These models are also applied to the BCI systems and provided successful results in this field. Arnold et al. have presented a deep learning category as follows [14]:

- Deep belief networks
- Stacked Autoencoders
- Deep kernel machines
- Deep Convolutional Networks

The way that deep methods work is that the layers behave separately, and they use the greedy training methods. Hence, one layer would be trained after the previous trained layer and this process will be continued until all the data is encoded. Now, it is time to do the fine-tuning method which is a supervised mean for the whole trained model to adjust the weights more precisely [15]. Cecotti et al. were among the first research that evaluated the CNN to extract the correct P300 in a P300 speller BCI [5]. They had used two convolutional layers respectively for learning spatial and temporal filters. Furthermore, they had combined CNNs to check how the model will work. Accordingly, Liu et al. have recently published a paper which has implemented a new CNN approach based on batch normalization. According to them, their model can be used to strengthen the P300 speller BCI systems [16]. Sobhani reported that Deep Belief Network would also be promising in BCI [17]. He had mentioned many accuracy percentages related to deep models and the other methods. Hence, with his results and discussions, he helped the scientists to strongly evaluate the deep learning. Vareka et al. published a paper which explores the Stacked Autoencoders for the P300 detection [15]. According to their talking, they have their own data set which it is available on the internet. Next, they simply pass the raw data to the Stacked Autoencoders which was the first approach of applying the Stacked Autoencoders to an ERP signal. Consequently, they have reported a 69.2% of accuracy in comparison with MLP and LDA.

As it is mentioned, recent researchers are focusing on learning from data in a deep neural network (DNN) structure. While shallow ones cannot handle the correct feature, DNNs are trying to learn from the database on their representation. These models provide the correct features with unsupervised methods and perform a non-linear dimensionality reduction. Finally, the backpropagation method would be run to do the fine-tuning operation on the whole network [18]. Similarly, in this paper, Stacked Autoencoders for the feature reduction and classification part is used, however to the best of the author's knowledge, it is the first time that a Stacked Autoencoders is applied to a P300 speller BCI.

The paper is organized as follows: In Section II the P300 speller paradigm is described. Stacked Autoencoders are presented in Section III. The database, pre-processing, and character recognition are described in Section IV. Finally, the results and the discussions are detailed in Section V.

Fig. 2: The P300 speller paradigm [19].

II. P300 Speller Paradigm

The P300 speller paradigm and totally Oddball paradigm at first was presented by Farwell and Dochin [7]. A subject will face a 6 by 6 matrix of various characters (Fig. 2) and should focus on one of the characters at a time. All the rows and columns are randomly intensified at a rate of 5.7Hz. When the user concentrates on one of the characters in a row or column which is flashed, ERP is recognized and a positive peak of voltage in 300 ms after the stimulus will appear in the EEG signal. That is why this waveform is called P300. Most of the times, this process will be repeated many times to achieve the better P300. Thus, through 12 flashed rows and columns there are only 2 which contains the target. By combining these rows and columns the character which the user had focused on would be attained [19]. it is known that there are 2 targets out of 12 intensifications and 10 non-targets. This concept is a binary classification, which is P300 and non-P300. Therefore, it is crucial that a machine learning algorithm would be chosen which could improve the accuracy of detecting the P300. BCI Competition III Dataset II has been utilized in this project which will be completely described in Section IV-A.

III. Stacked Autoencoders

A simple autoencoder is a two-layer unsupervised neural network (see Fig. 3). Usually, autoencoders are used to reduce the input dimension and elicit the useful features. Moreover, they are feedforward neural networks that are trained to learn the smaller code of the data [20]. To describe more, they compress the input data into a latent space representation and then reconstruct the output from this representation [21]. As a result, the number of neurons in the output (decoding) is equal to the input. This network includes two important parts:

1) Encoder which will compress the input data to a lower latent space representation. The encoding function could be shown as $h = f(x)$.
2) The Decoder part of the neural network which has to reconstruct the input from the latent space representation. The decoding function could be represented as $r = g(h)$ [21], [22].

Fig. 4 demonstrates architecture of an Autoencoder. Therefore, the above explanations can be formulated as follows:

$$g(f(x)) = g(h) = r \implies r \approx x \tag{1}$$

Where $f(\cdot)$ plays the role of a function in encoding layer and $g(\cdot)$ is the function that decode the input. So, it is obvious that the output r is wanted to be as similar as possible to the input x.

Sparse Autoencoders are a sort of regularized autoencoders which are usually used to learn the new representation for other problems such as classification [21]. In fact, by adding sparsity constraint the neural network is enforced to learn the fruitful information in data. Softmax regression [24] is a method for classification issues which generalizes logistic regression to classification problems. Subsequently, a deep network can be built from the combination of some Sparse Autoencoders and a Softmax classifier to do a complex classification concept. Thus, this network as a whole is then called Stacked Autoencoders (Fig. 5).

IV. Method

The proposed method consists of three main parts:

1) Pre-processing: It is crucial that an appropriate amount of data should be used to have more accurate information and analysis to pass to the Machine learning model.

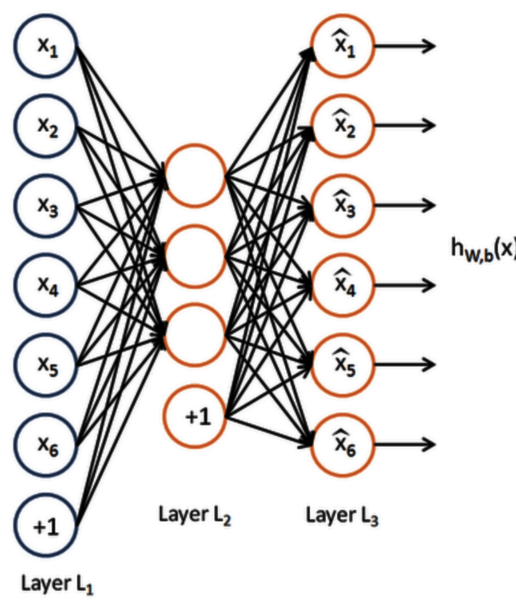

Fig. 3: A simple Autoencoder, is an unsupervised neural network that tries to learn a function $h_{W,b}(x) \approx x$. The output layer (Layer L_3) is as the same as the input layer (Layer L_1) [23].

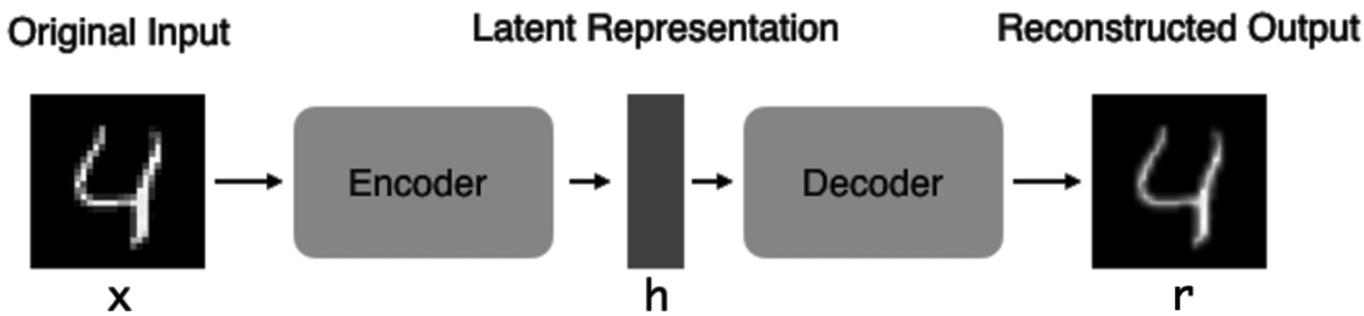

Fig. 4: The architecture of an Autoencoder. An application in the image processing field [21].

2) Using Stacked Autoencoders: This step is important enough that will discuss how to train a Stacked Autoencoders for a P300 classification issue which will disclose whether there is a P300 or not.
3) Character Recognition: In this section, the trained Stacked Autoencoders is used in order to do the character recognition in the test data set.

A. Database

In this data set, there are two different subjects that separately did the P300 speller paradigm. For each subject, there are two datasets that are consisted of the training set and a test set. The training set is composed of 85 characters and there are 100 characters in the test set. There are 15 repetitions among all 12 rows and columns. Hence, 180 trials ($15 \times 12 = 180$) is available. However, there are only 30 trials out of 180 trials that refer to a target stimulus that attains P300. So, it is needed to prepare the data to be labeled once in order to train the Stacked Autoencoders model. In the test set the targets are not identified, thus, the learner model should do this and detect the correct position of the P300 waveform from the training set [19].

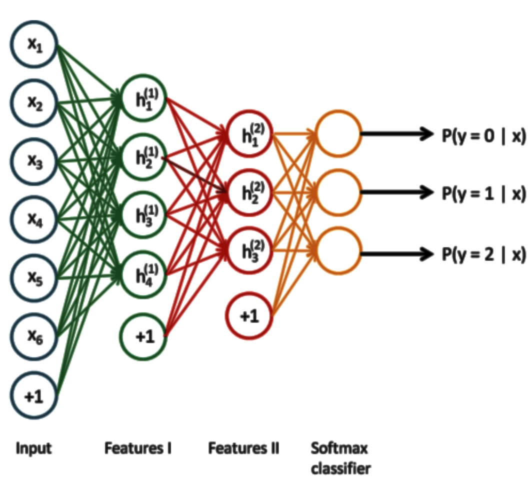

Fig. 5: Stacked Autoencoders [23].

B. Pre-processing

As it is known that the P300 component is occurred between 0 ms and 1000 ms of the EEG signal and will appear 300 ms after the stimulus, then the EEG simply preprocessed between 0 ms and 600 ms to the start of the stimulus. it is known that this amount of time is sufficient for the required data for a P300 classification problem. Then, the 4th-order bandpass Chebyshev Type I is used in order to filter the EEG signal. Furthermore, the cut-off frequency between 0.1 Hz and 20 Hz is set. Subsequently, some copies of the target stimulus were added to fix the imbalanced problem of the data. Then, the corresponding number of target and non-target remained. Consequently, the training set consists of 25500 features, which each of the classes includes 12750 of the features. Next, passing these features to the Stacked Autoencoders in order to do the classification of the P300 will be described.

C. Using Stacked Autoencoders

For the Stacked Autoencoders of a P300 classification, a data set D with N labeled trials: $D = \{x_i, y_i\}_{i=1}^{N}$ is provided, Where x_i is the feature vector and $y_i \in \{-1, +1\}$ is the corresponding label pointing the two classes. Here $y_i = +1$ refers to the existence of P300 of a stimulus that is expected and $y_i = -1$ refers to the absence of P300. In this part, the filtered EEG data were passed directly to the Stacked Autoencoders. At first, the feature vectors were shuffled to know that the Stacked Autoencoders is learning in a random condition. In brief, the training set in both unsupervised pre-training and supervised fine-tuning is used. Additionally, any of the weights after training process and the testing process were not changed.

At first, to have a better examination of the classification issues, 30% of the features for test and 70% for the training were selected. This was done to make sure how the model is learning the P300 stimulus and whether the parameters could be improved. After that the appropriate parameters and the model were found, all the training set were passed together for a better learning rate and accuracy.

This project was implemented using the Matlab Neural Network Toolbox [25]. The parameters of the Stacked Autoencoders such as the number of layers, number of neurons and iteration were set empirically. At first, the work started

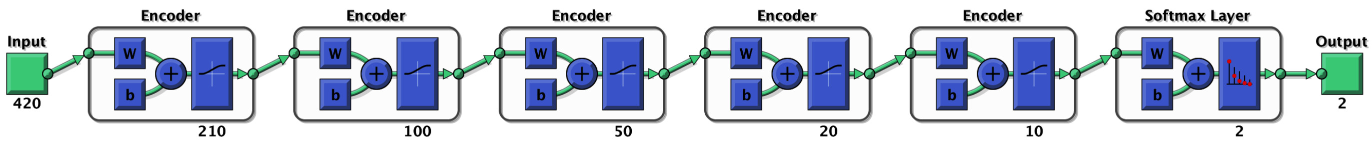

Fig. 6: The architecture of the proposed Stacked Autoencoder.

with two layers and then increase the number to reach the better accuracy. Then, check the iteration and at first set it to 1000 then subsequently examined that the model is going to face the over-fitting. Hence, the number of iterations were decreased to the optimum number. Eventually, the number of iterations was set to 300 and the optimum number of layers is as follows:

1) The input vectors include 420 features. Then these inputs were passed to the first Autoencoder layer.
2) The first Autoencoder layer with 210 was trained. Then it was shortened the 420 to 210 features.
3) The next Autoencoder was built with 100 neurons.
4) The 100 features were passed to 50 neurons of the next Autoencoder.
5) And then 20 neurons trained on the next Autoencoder.
6) Finally the last Autoencoder with 10 neurons was in hand.

   According to the above information, the 5 Autoencoders were combined all together in order to reduce the features from 420 to 10. Furthermore, other parameters such as L2WeightRegularization and SparsityRegularization were set according to the Matlab default settings [25].

   Afterward, the 10 feature vectors were passed to a Softmax classifier with 200 iterations. As a result, a Stacked Autoencoders with this structure: 420-210-100-50-20-10-2 was achieved. The next task was the fine-tuning process using the Backpropagation algorithm. The Stacked Autoencoder is shown in Fig. 6.

D. Character Recognition

The proposed Stacked Autoencoders of a P300 classification is ready to be applied to its main task which is the recognition of the character in the test set. As it is known, each character has 180 trials that only 30 of the 180 are the targets. And, the data set has repeated this process 15 times for certainty and providing more accurate characters. The feature vectors were passed one by one to the trained model that the model would tell whether it is P300 stimulus or not. Next, the model gives a score about the percentage of its guess of the target and non-target. Afterward, the overall score according to a row or column was calculated. Then by combining the highest scores of the corresponding row or column, the predicted character is attained. Accordingly, Eq. (2) shows how to predict the rows and columns based on the scores:

$$C_i = \sum_{i=1}^{I} SAE(f_{r,c}) \tag{2}$$

Where $SAE(f_{r,c})$ is the proposed Stacked Autoencoders which will provide the scores of the $f_{r,c}$, in i_th row or column of the 15 times repetitions. Consequently, the character would be chosen as follows:

$$\max(C_i) \implies \text{for all the 6 rows and 6 columns.} \tag{3}$$

V. Experiments

In this section, the results and comparison of the proposed method is provided. The suggested method is comparable to the state-of-the-art algorithms that have been worked in the field of P300 speller BCI. The Accuracy, Precision, and Recall are used to evaluate the proposed method related to the P300 classification. However, in this paper, the recognition accuracy which is the main goal in this task, is focused. the proposed model were compared to the Convolutional Neural Network which have been proposed by Cecotti et al. [5]. And then compare with some of the results that are reported by Yoon et al. [26]. Then other state-of-the-art methods such as GBM (Gradient Boosing Method), LDA, SVMLight, BNT (Bayes Network) that are reported in [19] have been compared to assess the performance of the proposed method. For different techniques and algorithms, the performance of the character recognition accuracy is depicted in Table I. According to the Table I, it is shown that for each subject in the data set the accuracy in each iteration is calculated, afterward, the average of the accuracy is presented. The Competition is based on the 5_{th} and 15_{th} repetitions and the winner was who received the most recognized character. A comparison between the proposed method and earlier methods are reported in the Table II. According to the Table II the proposed method is comparabled to the state-of-the-art methods.

VI. Conclusion and Future Works

In this paper, a method in the field of P300 speller BCI is presented. In this method, a deep neural network is used to classify the P300 stimulus. So, the model is Stacked Autoencoders which easily do the feature extraction and

TABLE I: Number of correctly recognized characters in the proposed method

Subjects	**Epochs**					
	1	***2***	***3***	***4***	***5***	***15***
A	14	29	45	49	52	92
B	32	36	49	49	59	91
Mean	23	32.5	47	49	55.5	91.5

TABLE II: Comparison of the proposed method with other techniques

Method	Epochs	
	5	**15**
CNN-3 [5]	57%	88.5%
GSVM [26]	3%	3%
GBM[a]	53%	89.5%
LDA	57.5%	87.5%
SVMlight	54.5%	83.0%
BNT[b]	27.5%	33.5%
Stacked Autoencoders[c]	***55.5%***	***91.5%***

[a]Gradient Boosting Method
[b]Bayes Network
[c]Proposed method

classification with astonishing results. An accuracy of 91.5% was achieved in character recognition that is comparable to the state-of-the-art and other contemporary methods. Next, the proposed method was evaluated and it was discovered that this model could be preferable to the others such as SVM-based models. That is because the Stacked Autoencoders are less prone to over-fitting than other conventional algorithms.

For the future work, using other Autoencoders are considered to evaluate whether they could improve the accuracy of the character recognition. Another idea is that by using other regularization methods make the model more robust to the over-fitting.

References

[1] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update," *Journal of neural engineering*, 2018.

[2] S. Vaid, P. Singh, and C. Kaur, "EEG signal analysis for BCI interface: A review," in *Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*. IEEE, 2015, pp. 143–147.

[3] J. A. Elshout, "Review of brain-computer interfaces based on the P300 evoked potential," 2009.

[4] S. Amiri, R. Fazel-Rezai, and V. Asadpour, "A review of hybrid brain-computer interface systems," *Advances in Human-Computer Interaction*, vol. 2013, p. 1, 2013.

[5] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 433–445, 2011.

[6] A. Setare, A. Rabbi, L. Azinfar, and R. Fazel-Rezai, "A review of P300 SSVEP and hybrid P300/SSVEP brain-computer interface systems," *Brain-Computer Interface SystemsRe-cent Progress and Future Prospects*, 2013.

[7] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[8] R. Fazel-Rezai, B. Z. Allison, C. Guger, E. W. Sellers, S. C. Kleih, and A. Kübler, "P300 brain computer interface: current challenges and emerging trends," *Frontiers in Neuroengineering*, vol. 5, p. 14, jul 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3398470/

[9] L. Vařeka, T. Prokop, R. Mouček, P. Mautner, and J. Štěbeták, "Application of Stacked Autoencoders to P300 Experimental Data," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 187–198.

[10] S. Kundu and S. Ari, "P300 Detection with Brain–Computer Interface Application Using PCA and Ensemble of Weighted SVMs," *IETE Journal of Research*, pp. 1–9, 2017.

[11] A. Rakotomamonjy and V. Guigue, "BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller," *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1147–1154, 2008.

[12] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, no. 2, p. R1, 2007.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/26017442 http://www.nature.com/doifinder/10.1038/nature14539

[14] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An Introduction to Deep Learning," in *European Symposium on Artificial Neural Networks (ESANN)*, ser. Proceedings of the European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, apr 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01352061

[15] L. Vařeka and P. Mautner, "Stacked Autoencoders for the P300 Component Detection," *Frontiers in neuroscience*, vol. 11, p. 302, 2017.

[16] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on Batch Normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[17] A. Sobhani, "P300 Classification using Deep Belief Nets," Ph.D. dissertation, Colorado State University, 2014.

[18] J. Zheng, J. Lu, S. Peng, and T. Jiang, "An Automatic Micro-Seismic or Acoustic Emission Arrival Identification Scheme with Deep Recurrent Neural Networks," *Geophysical Journal International*, no. November 2017, p. ggx487, 2017. [Online]. Available: + http://dx.doi.org/10.1093/gji/ggx487

[19] D. Krusienski and G. Schalk, "BCI Competition III Challenge 2004," 2004. [Online]. Available: http://www.bbci.de/competition/iii/

[20] K. Murphy, *Machine learning: a probabilistic approach*, 2012.

[21] N. Hubens, "Deep inside: Autoencoders – Towards Data Science," 2018. [Online]. Available: https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f

[22] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 37–49.

[23] A. Ng, J. Ngiam, C. Yu Foo, Y. Mai, and C. Suen, "UFLDL Tutorial," 2013. [Online]. Available: http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial

[24] A. Ng, "Cs 229 machine learning course materials," *Stanford University*, 2016.

[25] M. T. M. I. Natick, "Matlab," MATLAB Version 9.3.0 (R2017b) - Neural Network Toolbox, 2017.

[26] K. Yoon and K. Kim, "Multiple kernel learning based on three discriminant features for a P300 speller BCI," *Neurocomputing*, vol. 237, pp. 133–144, 2017.