Improving EEG-based BCI Neural Networks for Mobile Robot Control by Bayesian Optimization

Takuya Hayakawa and Jun Kobayashi
Department of Systems Design and Informatics, Kyushu Institute of Technology,
Kawazu 680-4, Iizuka, 820-8502, Japan

E-mail: jkoba@ces.kyutech.ac.jp
lab.jkoba.net

Abstract

The aim of this study is to improve classification performance of neural networks as an EEG-based BCI for mobile robot control by means of hyperparameter optimization in training the neural networks. The hyperparameters were intuitively decided in our preceding study. It is expected that the classification performance will improve if you determine the hyperparameters in a more appropriate way. Therefore, the authors have applied Bayesian optimization to training the EEG-based BCI neural networks and achieved the performance improvement.

Keywords: brain computer interface, electroencephalography, neural network, hyperparameters, Bayesian optimization, mobile robot control

1. Introduction

Brain Computer Interface (BCI) is a promising technology that provides means of direct communication through your brain. Since brain activities bring about perception, recognition, and sensory-motor functions in human beings, BCI based on a state of your brain has potential to be applicable to support many kinds of human activities. A lot of studies on BCI have used non-invasive brain activity measurement methods, especially Electroencephalography (EEG) because of its superior time resolution and ease of use. Furthermore, portable and low-cost EEG measurement devices have become readily accessible lately.

EEG-based BCI have already applied for assisting handicapped people and augmenting human capability[1,2]. R. Single et al. developed a SSVEP-based BCI for controlling a wheelchair using multi-class SVM[3,4]. J. Meng et al. experimentally investigated a noninvasive BCI for reach and grasp task of robotic arm[5]. In these studies, the subjects did not directly imagine desired behavior of a controlled object.

As an EEG-based BCI for mobile robot control, the authors are developing a neural network (NN) for EEG signal classification. In our earlier studies[6–8], we instructed a subject to imagine an arrow representing a desired behavior of a controlled object with closed eyes for removing visual stimuli unrelated to experiments. In addition to the experiments with the subjects closed their eyes, we conducted an experiment to confirm open eyes influence on constructing an EEG-based BCI NN[9]. Considering practical use of BCI, actually imagining a desired motion of a controlled object with open eyes would be a more appropriate way for controlling a mobile robot. However, none of the constructed NNs achieved practical performance.

The authors conjectured that one of the reasons for the insufficient results was due to hyperparameter settings in training the NNs. The hyperparameters were intuitively set in the preceding studies. It is expected that the classification performance will improve if you

Takuya Hayakawa, Jun Kobayashi

determine the hyperparameters in a more appropriate way. Therefore, the study described in this paper aims at improving the performance of the EEG-based BCI NNs for mobile robot control by hyperparameter optimization. For optimization, the authors adopted Bayesian approach that optimizes hyperparameters for NN training by determining the hyperparameters to be next verified on the basis of results already checked. In consequence, the method can find the optimal hyperparameters efficiently.

In this paper, the authors present experiments introducing Bayesian optimization using the EEG signals recorded in our previous study. The experimental results demonstrated that Bayesian optimization of hyperparameters improved the classification rate of the EEG-based BCI NNs for mobile robot control.

2. EEG Signal Classification Using Stacked Autoencoder

This section describes the structure of the multilayered NNs used in the preceding study[9]. Stacked Autoencoder (SAE) was employed as an EEG-based BCI NN for mobile robot control in the study. SAE is a multilayered NN initialized by stacking encoder layers of pretrained Autoencoder (AE), which is a three layered NN shown in Fig. 1.
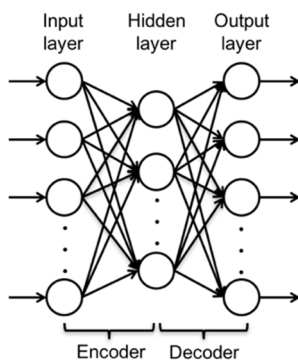

Fig. 1. Autoencoder (AE)

A typical AE has equal-sized input and output layers, and a less-sized hidden layer. The whole of AE is trained in order that it can yield an output signal equal to an input one. Since the hidden layer has less nodes than the input layer, significant information is extracted from the input signal through the encoder part, which is between the input and hidden layers, for restoring the signal through the hidden and output layers; namely AE is a NN for dimensionality reduction. G. E. Hinton and R. R. Salakhutdinov revealed that AE surpasses Principal Component Analysis (PCA) for dimensionality reduction[10].

SAE is a way to construct a multilayered NN avoiding vanishing gradient in backpropagation. As stated above, an initial multilayered NN is prepared by stacking encoders of pretrained AEs in cascade, and the whole NN is trained again using the same input signals and target ones. The last training is called finetuning.

3. Bayesian Optimization of Hyperparameters

Ahead of training a NN, you must determine some parameters, such as number of layers, number of nodes, learning rate, and so on, which are called hyperparameter. The preceding study has not achieved practical performance of the SAEs for EEG-based BCI probably because the hyperparameters were intuitively determined. Optimization of hyperparameters is an essential process to improve performance of NNs.

One of the methods for hyperparameter optimization is grid search, in which combinations of hyperparameters are uniformly arranged in the form of a grid in a hyperparameter space, all of the parameter combinations are verified. Therefore, the optimal hyperparameters are found with a high probability if the grid arrangement in the hyperparameter space is sufficiently dense. In the case of high dimensional hyperparameter space, however, it becomes difficult to conduct verification in the dense grid within a realistic time due to huge number of hyperparameter combinations. In addition, if some of the hyperparameters have little influence on the performance of the NN, calculation time for grid search is wasted on searching in worthless regions of the hyperparameter space.

J. Bergstra and Y. Bengio have shown empirically and theoretically that random search can find optimal hyperparameters more efficiently than grid search[11]. Furthermore, J. Snoek et al. have demonstrated that Bayesian optimization for hyperparameter selection of machine learning algorithms found optimal hyperparameters faster, and outperformed hyperparameter selection by a human expert[12]. The Bayesian approach determines the hyperparameters to be next examined on the basis of results already verified.

First, Bayesian optimization randomly selects hyperparameters and verifies them. Following that, a

region where the optimal hyperparameters are likely to exist are estimated from the previous results and preferentially verified hyperparameter in the region. The estimation of the region to be examined is made by maximizing an acquisition function that includes probability and expected values under the assumption that a function to be optimized follows Gaussian process. In this research, the authors attempted to optimize the hyperparameter of the SAEs for EEG-based BCI for mobile robot control using GPyOpt[13], which is a Python library for Bayesian optimization.

4. Results and Discussion

The authors empirically confirmed effects of Bayesian optimization for tuning hyperparameters on improving the classification performance of the SAEs for EEG-based BCI mobile robot control using the EEG signal dataset obtained in the preceding study. Fig. 2 is a schematic diagram of the apparatus for EEG measurement experiments conducted in the preceding study[9]. In the EEG measurement experiments, two different imagining tasks, named "CLOSED-EYES" and "OPEN-EYES", were given to three subjects as follows.

- CLOSED-EYES: close your eyes and imagine a specified arrow
- OPEN-EYES: watching the mobile robot moving to a certain direction, imagine the robot's motion

Refer to Ref. 9 for the other experimental conditions of the EEG measurement, such as experimental procedure, information of the subjects, and so on.
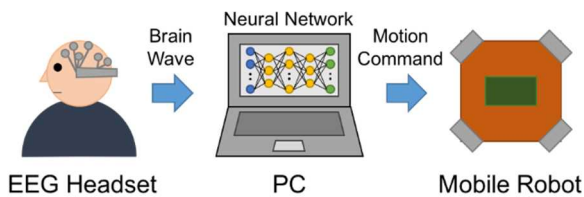

Fig. 2. Experimental apparatus

Chainer[14] (ver. 1.18.0) for implementing the SAEs and GPyOpt[13] (ver. 1.2.0) for Bayesian Optimization were used in the confirmation experiments. Table 1 shows the computation environment for development and execution of the SAEs. Table 2 describes the hyperparameters optimized in the experiments and their

options; the settings enclosed in parentheses are the options chosen in the preceding study[9]. GPyOpt optimized the hyperparameters so that false recognition rate is minimized. The false recognition rate of each SAE was calculated using 5-fold cross validation.

Table 1. Specifications of computation environment for development and execution of stacked autoencoders

CPU	Intel® Core™ i7-6800K CPU @ 3.40GHz
Memory	16GB (DDR4-2133 4GBx4)
Storage	SSD 240GB + HDD 1TB
GPU	NVIDIA GeForce GTX1060 6GB GDDR5
OS	Ubuntu 16.04 LTS

Table 2. Hyperparameters and options

Hyperparameter	Options (setting value in Ref. 9)
Number of hidden layers	1 ~ 3 (3)
Number of nodes in 1st hidden layer	250 or 300 (250)
Number of nodes in 2nd hidden layer	150 or 200 (150)
Number of nodes in 3rd hidden layer	50 or 100 (100)
Iteration of pretraining each AE (epoch)	1000 or 2000 (1000)
Iteration of finetuning (epoch)	5000 or 10000 (10000)
Dropout 50% in pretraining AEs	OFF or ON (ON)
Dropout 50% in finetuning	OFF or ON (ON)

Table 3 and Table 4 show the results obtained in the preceding study[9]. Table 5 and Table 6 report the classification rate percentages of the SAEs calculated in this study with Bayesian optimization. Compared the updated results with the previous ones, it is indicated that we could improve the performance of the SAEs trained with the hyperparameters optimized by Bayesian approach. However, the options of the hyperparameters given to the optimization algorithm were very restricted. Therefore, we would consider further improvement to be possible by revising the hyperparameter options more carefully.

5. Conclusion

This paper presented the performance improvement of the EEG-based BCI neural networks for mobile robot control proposed in the preceding study. The authors

introduced Bayesian optimization in order to find the better hyperparameters for training the neural networks and then experimentally confirmed the expected effect of the method. The authors will design the appropriate options of hyperparameters that considerably ameliorate the classification performance of the EEG-based BCI neural networks for mobile robot control.

Table 3. Classification rate percentages of SAEs trained using 120 samples recorded in CLOSED_EYES without Bayesian optimization[9]

Subject	1st day	2nd day	3rd day	4th day	5th day	Ave.
A	50.00	54.17	58.33	73.33	85.83	64.33
B	47.50	60.83	61.67	55.83	48.33	54.83
C	31.67	40.00	45.00	66.67	31.67	43.00

Table 4. Classification rate percentages of SAEs trained using 120 samples recorded in OPEN_EYES without Bayesian optimization[9]

Subject	1st day	2nd day	3rd day	4th day	5th day	Ave.
A	95.00	45.00	52.50	31.67	67.50	58.33
B	60.00	49.17	53.33	25.83	43.33	46.33
C	54.17	30.00	38.33	59.17	25.00	41.33

Table 5. Classification rate percentages of SAEs trained using 120 samples recorded in CLOSED_EYES with Bayesian optimization

Subject	1st day	2nd day	3rd day	4th day	5th day	Ave.
A	55.00	57.50	66.67	79.17	92.50	70.17
B	50.00	70.00	53.33	66.67	54.17	58.83
C	44.17	49.17	51.77	76.67	33.33	51.02

Table 6. Classification rate percentages of SAEs trained using 120 samples recorded in OPEN_EYES with Bayesian optimization

Subject	1st day	2nd day	3rd day	4th day	5th day	Ave.
A	95.83	48.33	45.83	41.67	75.00	61.33
B	66.67	57.50	60.83	33.33	43.33	52.33
C	60.00	40.00	49.17	76.67	38.33	52.83

References

1. A. Kübler et al., Patients with ALS can use sensorimotor rhythms to operate a brain-computer interface, *Neurology* **64**(10) (2005) 1775–1777.
2. T. P. Luu et al., Gait adaptation to visual kinematic perturbations using a real-time closed-loop brain–computer interface to a virtual reality avatar, *J. Neural Engineering* **13**(3) (2016).
3. R. Singla et al., Influence of stimuli color on steady-state visual evoked potentials based BCI wheelchair control, *J. Biomedical Science and Engineering* **6**(11) (2013) 1050–1055.
4. R. Singla et al., Influence of stimuli color in SSVEP-based BCI wheelchair control using support vector machines, *J. Medical Engineering & Technology* **38**(3) (2014) 125–134.
5. J. Meng et al., Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks, *Scientific Reports* **6**(38565) (2016).
6. K. Tomonaga et al., Experiments on classification of electroencephalography (EEG) signals in imagination of direction using a wireless portable EEG headset, in *Proc. of Int. Conf. on Control, Automation and Systems* (Busan, South Korea, 2015), pp. 1805–1810.
7. S. Wakamizu et al., Experiments on Neural Networks with Different Configurations for Electroencephalography (EEG) Signal Pattern Classifications in Imagination of Direction, in *Proc. of IEEE Int. Conf. on Control System, Computing and Engineering* (Penang, Malaysia, 2015), pp. 477–481.
8. K. Tomonaga et al., Experiments on Classification of Electroencephalography (EEG) Signals in Imagination of Direction using Stacked Autoencoder, *J. Robotics, Networking and Artificial Life* **4**(2) (2017) 124–128.
9. T. Hayakawa and J. Kobayashi, Open Eyes Influence on Electroencephalography Signals for Constructing Neural Network Classifiers as Mobile Robot Control Brain-Computer Interface, in *Proc. of Int. Conf. on Advanced Mechatronic Systems* (Xiamen, China, 2017), ThuM01-03.
10. G. E. Hinton and R. R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, *Science* **313** (2006), pp. 504–507.
11. J. Bergstra and Y. Bengio, Random Search for Hyper-Parameter Optimization, *J. Machine Learning Research* **13** (2012) 281–305.
12. J. Snoek et al., Practical Bayesian Optimization of Machine Learning Algorithms, in *Proc. of Int. Conf. on Neural Information Processing Systems* (Lake Tahoe, Nevada, 2012), pp. 2951–2959.
13. GPyOpt: A Bayesian Optimization framework in Python, http://github.com/SheffieldML/GPyOpt, 2016.
14. Chainer: A Powerful, Flexible, and Intuitive Framework for Neural Networks, https://chainer.org/.