

## Research Article

# Decode Brain System: A Dynamic Adaptive Convolutional Quorum Voting Approach for Variable-Length EEG Data

Tao Xu,<sup>1</sup> Yun Zhou ,<sup>2</sup> Zekai Hou,<sup>1</sup> and Wenlan Zhang<sup>2</sup>

<sup>1</sup>School of Software, Northwestern Polytechnical University, Xi'an 710072, China

<sup>2</sup>School of Education, Shaanxi Normal University, Xi'an 710062, China

Correspondence should be addressed to Yun Zhou; zhouyun@snnu.edu.cn

Received 6 November 2019; Revised 13 January 2020; Accepted 22 February 2020; Published 28 March 2020

Guest Editor: Yuan Yuan

Copyright © 2020 Tao Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The brain is a complex and dynamic system, consisting of interacting sets and the temporal evolution of these sets. Electroencephalogram (EEG) recordings of brain activity play a vital role to decode the cognitive process of human beings in learning research and application areas. In the real world, people react to stimuli differently, and the duration of brain activities varies between individuals. Therefore, the length of EEG recordings in trials gathered in the experiment is variable. However, current approaches either fix the length of EEG recordings in each trial which would lose information hidden in the data or use the sliding window which would consume large computation on overlapped parts of slices. In this paper, we propose TOO (Traverse Only Once), a new approach for processing variable-length EEG trial data. TOO is a convolutional quorum voting approach that breaks the fixed structure of the model through convolutional implementation of sliding windows and the replacement of the fully connected layer by the  $1 \times 1$  convolutional layer. Each output cell generated from  $1 \times 1$  convolutional layer corresponds to each slice created by a sliding time window, which reflects changes in cognitive states. Then, TOO employs quorum voting on output cells and determines the cognitive state representing the entire single trial. Our approach provides an adaptive model for trials of different lengths with traversing EEG data of each trial only once to recognize cognitive states. We design and implement a cognitive experiment and obtain EEG data. Using the data collecting from this experiment, we conducted an evaluation to compare TOO with a state-of-art sliding window end-to-end approach. The results show that TOO yields a good accuracy (83.58%) at the trial level with a much lower computation (11.16%). It also has the potential to be used in variable signal processing in other application areas.

## 1. Introduction

The human brain system is a complex and dynamic system, which is difficult to unravel and understand [1]. How to decode brain activity is always a challenge for researchers. For understanding the brain without invading it, electroencephalogram (EEG) [2], functional near-infrared spectroscopy (fNIRS) [3], and functional magnetic resonance imaging (fMRI) [4] have been used commonly to measure brain activity with their own benefits. Using EEG, the electrical activity of the brain is recorded and monitored. And fNIRS monitors the brain through hemodynamic responses associated with neuron behavior. Both of them have a good temporal resolution and have been used to decode the

states of the brain over time. Complexity driven methodology provides a powerful approach to understand the dynamic connections of different brain regions [5] and to explore different kinds of brain activities. As a measure with high spatial resolution, fMRI measures brain activity by detecting changes associated with blood flow and provides data sources for such methodology.

Over the last two decades, EEG has become a low cost and feasible technology to dig into brain activities, which is used for diagnosing disease [6], analyzing emotion [7], and controlling specific machines [8]. To process EEG data and decode the patterns hidden inside the brain, machine learning techniques have been increasingly used and played a vital role in EEG-based research and application areas [9].

Currently, there are two types of methods for classifying EEG data, that is, the conventional type and the end-to-end type. In the conventional type, EEG data are filtered in the time, frequency, or spatial domain for extracting features. The goal of these features is for building a classifier and a model. Among the methods in the conventional type, support vector machine (SVM) has been used widely in EEG signal processing [10, 11], because of a good performance on EEG data, the amount of which is relatively small compared with the image or audio datasets. In 1994, Tsoi et al. [12] attempted to use artificial neural networks (ANNs) to find people suffering from psychiatric disorders based on EEG. In recent years, deep learning networks have appeared to be effective for EEG signal classification [13], given the sufficient training data available. The end-to-end approach builds a classifier through the raw EEG data without any handcrafted features. It replaces multiple steps with just a single neural network, which is essential for decoding the brain but not clear how to select features.

Learning is one of the most important brain activities for human. Recognizing cognitive states is the prerequisite of learning intervention, offering an opportunity to improve learning experience and outcomes. Therefore, to understand and recognize the process of human cognitive states in learning is an attractive and significant challenge for researchers. Compared with facial expression and behaviors, physiological measures can directly reflect the intrinsic mental states of human beings which involve neural activities of the brain. As one of the physiological measures, EEG has a good temporal resolution and could reflect the changes in emotions over time.

To understand and recognize the process of human cognitive states, one key issue is supposed to be addressed. It is how to process the input and output data in variable length. In a real learning environment, test item differences and individual differences result in input in variable length. First, in terms of test items, the learner reacts differently, showing differences in thinking and answering time. Second, due to individual differences, learners have different thinking times when facing the same test item, and it is also reflected in the difference in answering time. Therefore, the length of EEG recordings in trials gathered in the experiment is usually variable.

The conventional classifiers require input data in fixed length, which makes some current approaches attempt to limit all trials of EEG recordings in a fixed time. Forcing input length to be the same is not the best solution, which would result in the loss of information hidden in the data. Some approaches [14–18] employ the fixed-length sliding time window to traverse different trials of EEG recordings in different lengths. For example, in [14], Xu et al. proposed the application of MW-TFA techniques applying a set of sliding windows instead of a single window to process EEG data. The results demonstrated MW-TFA techniques as a useful tool to estimate the TF distribution. In [11], Liu et al. proposed a fractal dimension-based algorithm of quantification of basic emotions, leveraging the sliding window. The benefit of the usage of the sliding window is that this method enables real-time processing. To recognize a discrete

emotion in real time, Liu et al. [17] used a short-time Fourier transform (STFT) with a sliding time window approach for feature extraction and normalization based on time-frequency (TF) analysis. The results showed an advantage over the existing state-of-the-art real-time emotion recognition systems from EEG signals in terms of accuracy. For emotion classification, Wang et al. [18] used a time window without overlapping to process EEG data and tested the effect of window size. However, the sliding time window method that processes overlapping parts between slices consumes large computation on repeatedly calculating overlapped parts of slices. A time window without overlapping may miss information hidden in EEG data.

To achieve a low computation but still a good performance when processing EEG data in variable length, we propose TOO (Traverse Only Once), a new approach for processing variable-length EEG trial data in this paper. TOO is a convolutional quorum voting approach that takes into account the input data in variable length and reduces the unnecessary computation. First, the main idea of TOO is to build a pure convolutional model to recognize two classes of cognitive states by traversing the EEG trial only once. To avoid the computation on overlapped parts of data slices, the model is built by a convolution with rectangle size kernels, which is a convolutional implementation of the sliding time windows. Second, to process input and output data in variable length, we adopt the  $1 \times 1$  convolutional layer to replace the fully connected layer. Each output cell generated from the  $1 \times 1$  convolutional layer corresponds to each slice created by a sliding time window, which reflects changes in cognitive states. Third, we use quorum voting to process output cells, which determine the cognitive state representing the entire single trial.

TOO has several benefits over existing machine learning approaches for processing EEG data. First, TOO is able to process variable-length EEG trial data with a convolutional implementation of the sliding time windows. It provides an adaptive model for trials of different lengths. Second, it traverses only once the entire EEG trial data to predict class probabilities and supports end-to-end learning, i.e., learning from the raw data of channels as input directly. Due to limitations in the study of human brains and cognitive activities, it is unclear which underlying essence from EEG data should be extracted as features for a specific cognitive state classification using conventional methods. However, end-to-end learning is able to map raw data directly to objectives without handcrafted features. Third, through evaluation, TOO has been proved to generate classifications with good performance and low floating-point operations (FLOPs) in computation. It avoids highly duplicated computation, unlike sliding time window techniques. Finally, TOO obtains not only the local classification results reflecting the changes in the cognitive states but also the class probability representing the state of the entire trial with quorum voting technique.

To verify this approach, we implement an experiment, in which we use Raven's progressive matrices (RPMs) [19] to elicit two cognitive states and collect EEG data. The results show that TOO is effective and requires low computation. Compared

with a state-of-art sliding window end-to-end approach, TOO yields good accuracy (83.58%) with a much lower FLOPs in computation (11.16%). It also has the potential to be used in variable signal processing in other application areas.

## 2. The Convolutional Quorum Voting Approach

Inspired by YOLO [20], we propose a convolutional quorum voting approach that traverses entire EEG trial data only once to predict class probabilities. As shown in Figure 1, the whole recognition approach has three following main parts: convolutional implementation of the sliding time windows,  $1 \times 1$  convolutional layer, and quorum voting.

*2.1. Convolutional Implementation of the Sliding Time Windows.* The first part of the TOO is a convolutional implementation of the sliding time windows. We use convolution to compute the entire EEG trial input in one go to take over the sliding time windows, which is convolutional implementation of the sliding time windows. Figure 2(a) shows a sliding time window approach that segments the entire EEG trial data into several windows and computes each window of the input one at a time. Compared to sliding time window approaches, convolutional implementation shares mutual computations of overlapping so to avoid duplicated computation. Convolution, in deep learning, is considered as a cross-correlation. It performs an element-wise product with the “window” (or the kernel), followed by summing up the results into a single output. To produce the next results, it slides the window through the input and performs the same operations. Convolution is shift-invariant and linear, and all weights of the “window” can be shared.

When implementing convolution in a sliding way (see Figure 2(b)), the difficulty is how to ensure that the output size equals the number of the sliding window (see the bottom of Figure 2). For convolution, the size of the output of one convolutional layer depends on the size of the filter and stride. The EEG input, collecting from multiple channels, in our example, i.e., 8 channels, looks like an extreme long narrow band, in which the length (sampling rate multiples one trial time) is much more than the width (numbers of channels). Its size differs from common deep learning tasks, like image processing. Thus, we design a rectangle convolutional kernel in TOO, instead of typical square kernel, to meet the requirements from this feature of EEG data. The structure of model for EEG data can be customized according to the width and length of the kernel and stride.

The output of each layer depends on the kernel size and stride. Through adjusting the number of layers, kernel size, and stride, we can build networks that produce desirable output dimension (see the bottom of Figure 2). The output of each layer is expressed as

$$\begin{aligned} O_w &= \frac{(I_w - K_w + 2 \times P)}{K_s} + 1, \\ O_h &= \frac{(I_h - K_h + 2 \times P)}{K_s} + 1, \end{aligned} \quad (1)$$

where  $O_w$  and  $O_h$  represent the width and height of output, respectively;  $I_w$  and  $I_h$  represent the width and height of input, respectively;  $K_w$  and  $K_h$  represent the width and height of the kernel, respectively;  $P$  is the padding size; and  $K_s$  is the kernel stride.

*2.2. Converting Fully Connected (FC) Layer to  $1 \times 1$  Convolutional Layer.* The second part of the TOO is the  $1 \times 1$  convolutional layer. The typical convolutional neural networks contain two parts: feature extraction and classification. The convolutional layers serve feature extraction from data. After the output of the last convolutional layer is being flattened, the fully connected (FC) layer is added to classify data into different categories. This structure makes that the whole net can be trained from end to end by backpropagation. However, it has a birth defect that the size of output and input must be fixed due to the FC layer. The FC layer requires every neuron to connect to all neurons on the other side (see Figure 3(a)). If the size of output or input varies, the net has to be changed at the same time.

In mathematics, the  $1 \times 1$  convolutional layer equals a fully connected layer. Convolutional kernels, which share parameters in the whole neuron layer, can be connected to a local region. Therefore, the functional form between two layers is identical, which is not changed by the structure of the net (see Figure 3(b)).

To meet the requirement of processing variable length of input and output, we use  $1 \times 1$  convolution to replace the FC layer in the end part of the networks in TOO (see Figure 3). The main difference between FC layer and  $1 \times 1$  convolutional layer is that the former requires a fixed-size input, while the latter can process input and output in variable sizes with the shared same parameters (weights).

In TOO, the output size depends on the length of EEG data of a trial, the length of the sliding time window, and the stride of the sliding time window. Each output cell generated from the last layer represents one cognitive state which is equal to the classification result created by a single sliding window. The number of final output cells is expressed as

$$N = \frac{L_t - L_w}{S} + 1, \quad (2)$$

where  $N$  represents the number of final output cells,  $L_t$  is the length of a trial,  $L_w$  is the length of a sliding window, and  $S$  is the stride of the sliding time window.

In TOO, the structure of models is dynamic adaptive, but the convolutional kernel size is immutable. TOO model design enables end-to-end training for every EEG data in different lengths simultaneously.

*2.3. Quorum Voting.* The last part of the TOO is quorum voting. With the quorum voting part, TOO provides the opportunities to obtain a global result as well as a local result. In some cases, it is important to know not only a global cognitive state but also the changes in such a cognitive state for many researchers, especially psychologists. Recognizing the changes in cognitive states during the learning process

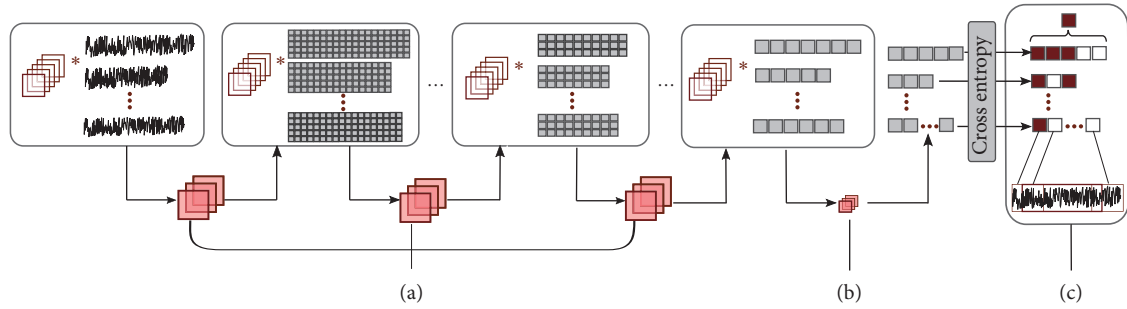


FIGURE 1: The TOO approach: (a) convolutional implementation of sliding time windows; (b)  $1 \times 1$  convolutional layer; (c) quorum voting.

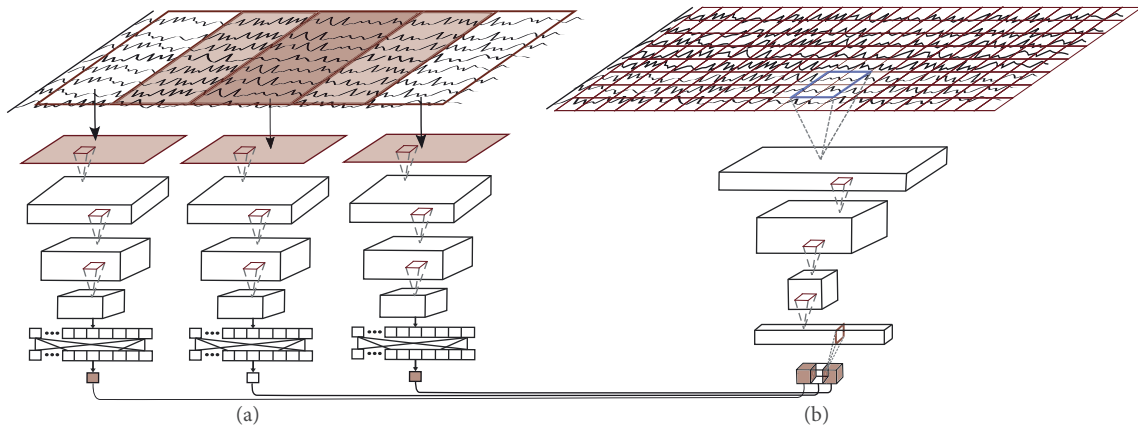


FIGURE 2: From sliding time window approach to convolutional implementation of the sliding time windows. (a) The EEG data are segmented into three pieces by sliding time windows. These pieces are predicted with class probabilities, respectively. (b) The same EEG data are processed by a pure convolution and obtain directly three output cells which are corresponding to three pieces generated by sliding time window in (a).

helps researchers investigate cognitive states at a fine-grained level. In other cases, researchers are also very concerned with the cognitive state in a relatively long period.

Using a fully convolutional network can obtain a main cognitive state as a global result for each trial based on a fully convolution network directly. However, this result only represents an emotional state over a period of time and cannot reflect the subtle changes during this period. Therefore, as shown in Figure 1, we integrate a quorum voting part in TOO. Convolutional implementation of the sliding time windows and  $1 \times 1$  convolutional layer can produce output cells that have the same number as the output generated by a sliding time window. These output cells are essential for cognitive investigation, reflecting the changes in the brain state over time. Then, the third part, quorum voting, works on the output cells and produces a cognitive state representing the entire data, which helps researchers explore the main cognitive state for a long period, at least an entire trial.

Properly handling a possible tie is the prerequisite of victory voting. In this paper, we only take into account the binary classification. To avoid a tie, the total number of output cells should be odd. For our case, we trim the last extra small segments away from the entire data and keep an integral number of seconds. Since most artefacts are caused

by interaction at the end of each trial, this trimming strategy minimizes the loss of EEG data while achieving the total number of output cells to be odd. For example, as shown in Figure 4, the length of EEG in a trial is 6.12 seconds. The length of the sliding window is 4 seconds, and the stride is 0.5 seconds. So the number of output cells calculated by our model is 5, with disposing of 0.12 seconds.

### 3. Experiment, Evaluation, and Results

To distinguish whether the cognitive state of the learner's answering question correctly is a guessing state or an understanding state, we designed and implemented an experiment (see Figure 5(a)). In this experiment, we collected EEG data, time stamps for segmenting data into trials, self-assessment of subjects, and subjects' answers for Raven's progressive matrices (RPMs) [19]. We used the EEG trial data collected from this experiment to verify our TOO approach.

#### 3.1. Experiment

**3.1.1. Subjects.** Twenty-three subjects ranging from 20 to 47 years (mean = 24.48 and SD = 6.36), including 11 females and 12 males, were involved in this experiment. All subjects

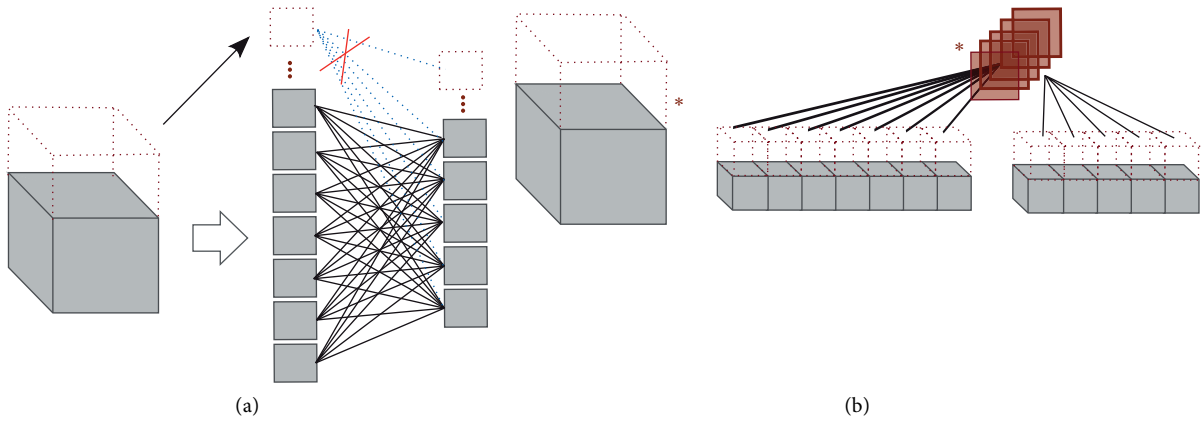


FIGURE 3: Converting FC layer to  $1 \times 1$  convolutional layer. (a) FC layer. (b)  $1 \times 1$  convolutional layer.

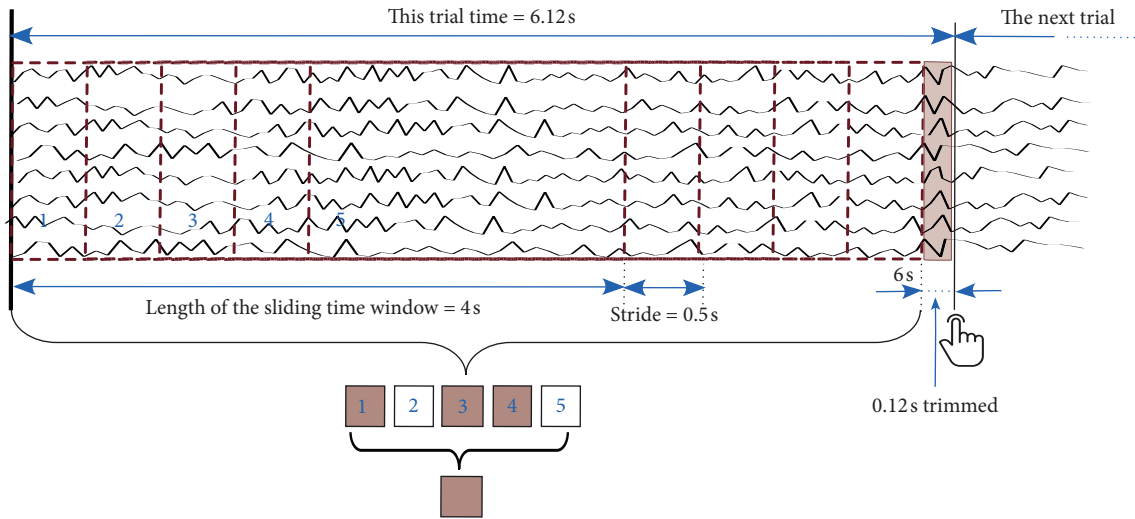


FIGURE 4: Trimming strategy in quorum voting to avoid a tie.

had normal or corrected vision and were right-handed. Most of the subjects (60.87%) had college-level education. All subjects were either studying or working in the university. All subjects have read and signed the ultimate consent form in a single access type version. All data can be shared publicly. Subjects were compensated for their time.

**3.1.2. Stimulus Materials.** For the experiment, we selected Raven’s progressive matrices (RPMs) as the materials to evoke two cognitive states in logic reasoning, including “guessing” and “understanding”. RPM is a nonverbal intelligence test that contains visual geometric design items, the scores of which would not be influenced by culture and knowledge. The task of each test item is to pick up the missing one from six or eight choices based on pattern inference. Choosing an answer is based on the keyboard instead of the mouse to minimize the artifacts caused by electromyography. We screened 48 items of this test with different levels of difficulty. The item difficulty index ranges from 0 to 0.83 with an average of 0.27, which ensured that

the “guessing” and “understanding” states could be elicited. Overall, each subject completed 48 trials of logic reasoning.

The EEG trial data were labeled as “guessing” when the subject answered the RPM item correctly and reported a “confused” state during the answering process. When the subject answered the RPM item correctly and did not feel confused, the trial data were labeled as “understanding”. The incorrect answers were not included in this dataset. To guarantee the quality of ground truth, we combined item answers with subjects’ self-reported data to label EEG signals for classification.

**3.1.3. Procedure.** In the beginning, the tester briefly explained this experiment and asked for the permission of using the recorded EEG data for the research purpose. Every subject read and signed the consent form. Then, each subject had a 150-second of brain resting with watching 10 scenery and had RPM with watching and responding to 48 consecutive test items coded by E-Prime 2.0 [21], wearing EEG headset (see Figure 5(b)). Finally, each subject filled out a

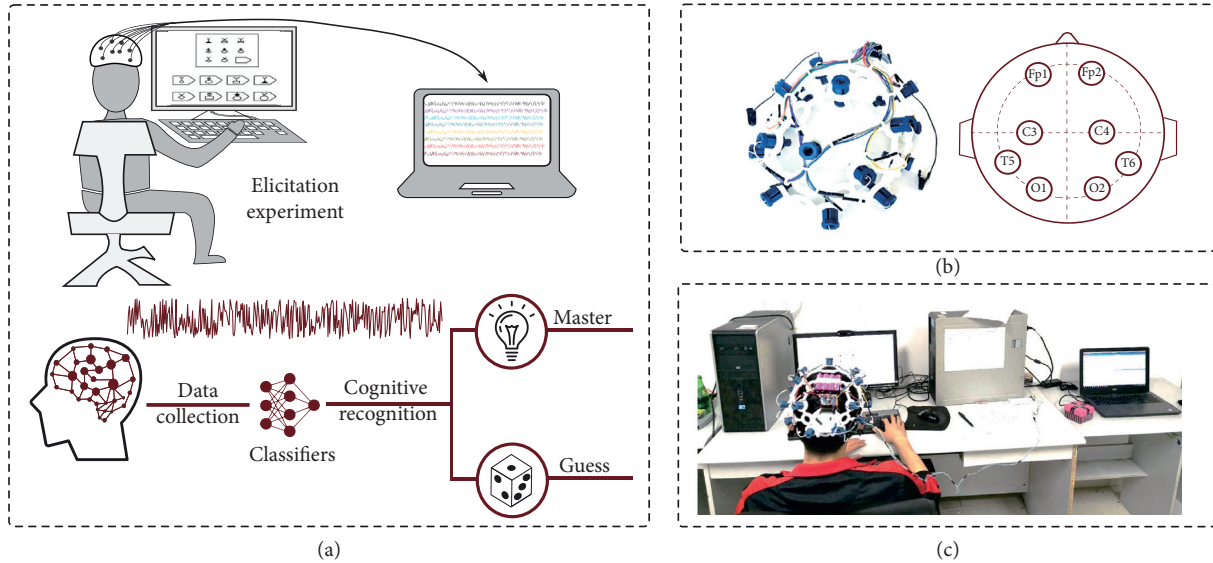


FIGURE 5: (a) The flow of EEG data collection and the recognition of “guessing” and “understanding”. (b) The open BCI acquisition device and 8 channel positions. (c) One of the subjects was watching the stimulus and selecting items in the experiment. At the same time, EEG data were collected and visualized on the screen of another computer.

customized questionnaire to self-report their states after finishing watching. In this process, EEG data were recorded through a laptop, and the stimuli were presented via another computer, with the trigger synchronizing time stamps.

**3.1.4. Acquisition Device.** We collected EEG data, time stamps for segmenting data into trials, self-assessment of subjects, and subjects’ answers for RPM. We employed the OpenBCI Cyton board with the 3D printed headset to acquire EEG data. This headset features 8 channels (Fp1, Fp2, C3, C4, T5, T6, O1, and O2) plus 2 references (A1 and A2) based on the 10–20 format (see Figure 5(c)), with the sampling rate of 250. We integrated a trigger function and hardware into the system to help split each subject’s data by trial. The time stamps for segmenting data into trials were obtained here. The stimuli were coded by E-Prime 2.0. Through E-Prime, we gathered subjects’ answers for RPM. After the stage of elicitation, self-assessment reports were filled out by subjects. To make the label accurately reflect the cognitive states, data of self-assessment and answers for RPM were used.

**3.2. Evaluation.** We test with different approaches and compare TOO approach with STQV approach in the work of Xu et al. [22]. The classification tasks are towards two categories: “guessing” and “understanding.”

**3.2.1. Model Design.** STQV is based on the sliding time window approach, which aims at processing variable EEG data. To make comparison possible, we used the TOO to build a model, which had the same number of layers and the number of output cells as STQV. We present the details of the model in Figure 6. This TOO model was composed of six

convolutional layers, a  $1 \times 1$  convolutions layer, and a quorum voting layer. The input width was the number of electrodes, and the input length was the answering time of subject in each trial. The output size corresponded to the number of sliding time windows.

**3.2.2. EEG Dataset and Training.** The EEG data that we collected had the following properties. First, EEG data of each trial varied in length due to individual differences and differences in the difficulty of the test items. Second, the labels of these data were two classes. We created a dataset to store EEG data. We had 1104 trials ( $23 \times 48$ ) of EEG data in total, and EEG data for each trial contained eight channels of data. Only 294 of 1104 trials were used, which were related to the classification of “guessing” and “understanding”. Therefore, 294 trials with the labels “guessing” or “understanding” were filtered to evaluate TOO. Among them, the maximum value of the length of trials was 15 s, the minimum was 5 s, the average was 10.57 s, and the standard deviation was 5.86. 4194 pieces were generated by the sliding time window in STQV. The output size was determined by the number of sliding time windows, as shown in Figure 2. The length of the sliding window is 4 seconds, and the stride is 0.5 seconds. Finally, 4194 output cells were generated by the TOO model.

In this work, we split the data into two sets, that is, training set and test set. Then, we used this train/test split to evaluate the performance of the final model. The training set randomly collected EEG data from 16 subjects, and the test set used the data of 7 subjects. The learning rate was set to  $1.5 \times 10^{-3}$ . Binary cross-entropy was used to compute the loss of model, while the stochastic gradient descent was adopted as the optimization algorithm to reduce loss. Early stopping was used as the method of regularization to avoid overfitting when training model with iterations.

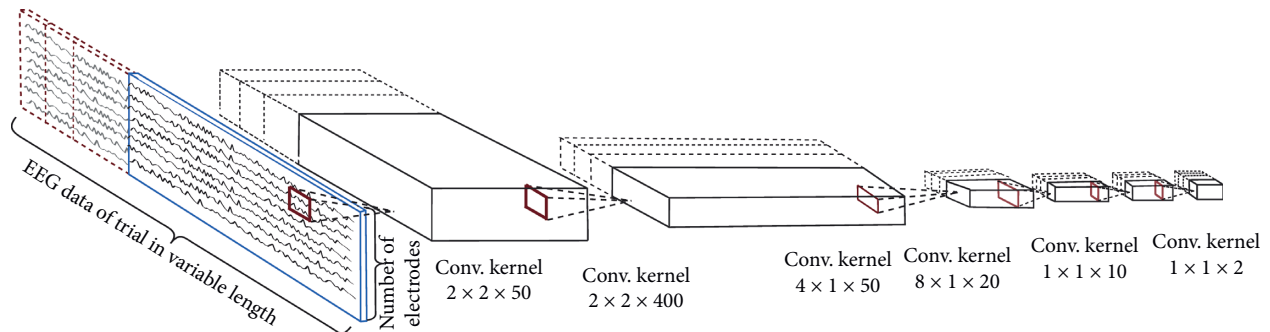


FIGURE 6: Six convolutional layers of the TOO model structure.

TABLE 1: The accuracy of TOO and STQV.

Accuracy	STQV				TOO
	FBCSP + SVM	FBCSP + LDA	FBCSP + NBPW	ConvNets	
Output cells	81.22%	82.04%	82.04%	86.26%	<b>86.00%</b>
Trial	83.71%	82.64%	80.65%	83.58%	<b>83.58%</b>

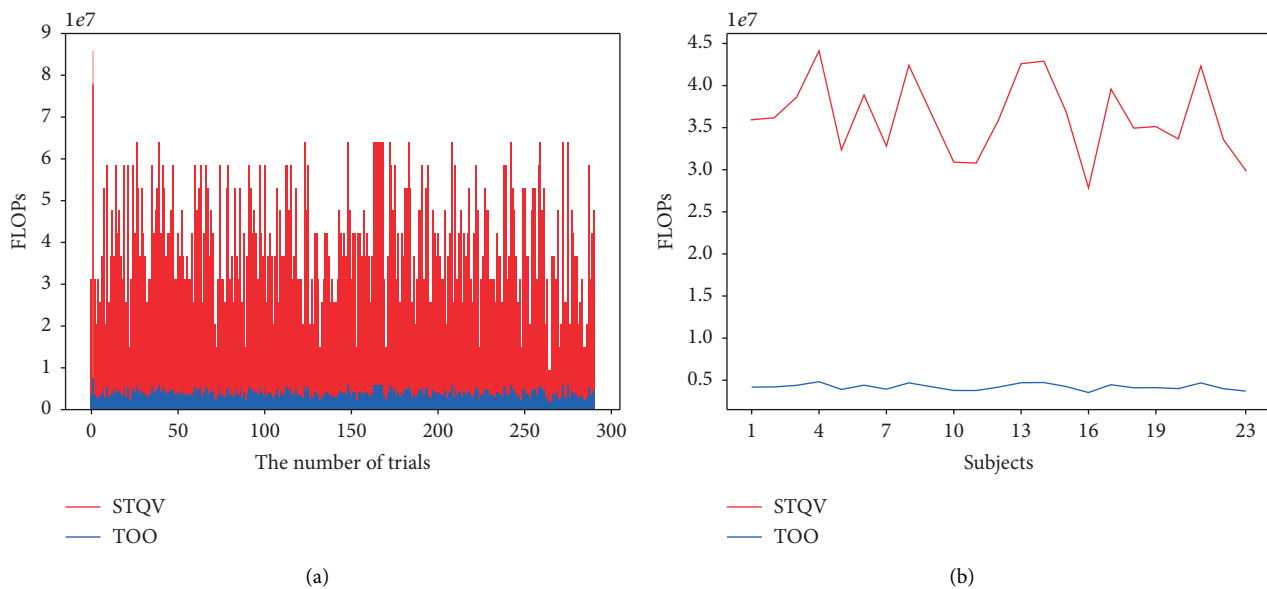


FIGURE 7: (a) The FLOPs charts of trials; (b) the FLOPs charts of subjects.

**3.3. Results.** To test TOO, we calculated the accuracy and computation of the models built by TOO and another approach that used a sliding time window to process the data and then compared these results.

**3.3.1. Accuracy.** First, we compared the performance of TOO and a sliding time-window with quorum-based voting (STQV) approach [22]. STQV is an approach to process the input in variable length. It segments EEG data of variable length into fixed-length pieces, then predicts the class of each piece, and determines the class of each question through voting from pieces. In [22], the STQV approach was implemented with the end-to-end ConvNets and filter bank

common spatial pattern (FBCSP) combining the vector machine (SVM), linear discriminant analysis (LDA), and Naïve Bayesian Parzen window (NBPW) classification algorithms. It employed the time sliding window to process variable data. EEG trial data were first segmented into a number of 4-second time window with a 3.5-second overlapping between two successive time windows. The stride is 0.5 s, and each slice is 4 s. The slices were used as the input for predicting class probabilities by the end-to-end convolution.

We made the comparison not only at the output cell level but also at the trial level. As shown in Table 1, the accuracy of TOO achieves 86.00% at the level of output cells and 83.58% at the level of trial, performing as good as the end-to-end ConvNets with STQV approach.

3.3.2. *Computation.* Besides accuracy, we calculated the floating point operations (FLOPs) [23], using the following function for convolutional kernels:

$$\text{FLOPs} = 2 \times H \times W \times (C_{in} \times K_h \times K_w + 1) \times C_{out}, \quad (3)$$

where  $H$ ,  $W$ , and  $C_{in}$  are the height, width, and the number of channels of the input feature map, respectively;  $K_h$  and  $K_w$  are the kernel width and kernel length; and  $C_{out}$  is the number of output channels.

Different from the sliding time window approach, TOO traverses the entire EEG data of a trial for training and testing; therefore, it reduces duplicated computation greatly. As shown in Table 1, ConvNets show the best performance compared with FBCSP + SVM, FBCSP + LDA, and FBCSP + NBPW. Both ConvNets and TOO are based on convolutional neural networks and use quorum voting. In this work, we compared TOO and STQV with ConvNets (for which we used STQV in short in the next paragraph) and calculated FLOPs for both of them.

As shown in Figure 7, it is obvious that TOO outperforms STQV greatly in computation. It is easy to find that TOO model costs much less computation with good accuracy. The bars indicate the FLOPs of every trial in Figure 7(a), of which the red represents the TOO and the blue represents the STQV. The lines in Figure 7(b) describe the FLOPs of every subject. TOO yields a smaller amount computed than STQV for all trials and all subjects. The FLOPs of all trials on average for STQV is  $4.0959 \times 10^7$ , while for TOO is just  $0.4573 \times 10^7$ . In this test, TOO uses only 11.16% FLOPs of STQV to achieve the same good performance.

## 4. Conclusion

In this paper, we propose a convolutional quorum voting approach named TOO to recognize two-class cognitive states with EEG trial data. TOO builds a dynamic adaptive model to process input and output EEG signals in variable length. Due to its nature pure convolution operation, it products classification results by traversing the entire input data only once instead of the sliding time window approaches, which compute overlapped data repeatedly. TOO extremely reduces computation. Results from the evaluation show that TOO achieves an accuracy of 83.58% of classification but costs only around one-tenth of the computation of the state-of-art sliding time window approaches. This research sheds new light on EEG data processing with convolution.

Furthermore, the TOO model has great potential ability to deal with data in variable length in other domains, like speech-based emotion recognition, physiological signal recognition, and sensor-based activity recognition.

## Data Availability

The EEG data used to support the findings of this study may be released upon application to the effective computing and intelligent interaction group in Northwestern Polytechnical University, who can be contacted at xutao@nwpu.edu.cn.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers 61703259 and 61702417) and the Natural Science Foundation of Shaanxi Province (grant number 2019JQ-628).

## References

- [1] H. T. Siegelmann, "Complex systems science and brain dynamics: a frontiers in computational neuroscience special topic," *Frontiers in Computational Neuroscience*, vol. 4, no. Sep, 2010.
- [2] Electroencephalography, Wikipedia, 05-Jan-2020.
- [3] Functional Near-Infrared Spectroscopy, Wikipedia, 03-Jan-2020.
- [4] Functional Magnetic Resonance Imaging, Wikipedia, 01-Jan-2020.
- [5] Q. K. Telesford, S. L. Simpson, J. H. Burdette, S. Hayasaka, and P. J. Laurienti, "The brain as a complex system: using network science as a tool for understanding the brain," *Brain Connectivity*, vol. 1, no. 4, pp. 295–308, 2011.
- [6] J. Jeong, "EEG dynamics in patients with Alzheimer's disease," *Clinical Neurophysiology*, vol. 115, no. 7, pp. 1490–1505, 2004.
- [7] G. Lee, M. Kwon, S. Kavuri Sri, and M. Lee, "Emotion recognition based on 3D fuzzy visual and EEG features in movie clips," *Neurocomputing*, vol. 144, pp. 560–568, 2014.
- [8] A. Kostov and M. Polak, "Parallel man-machine training in development of EEG-based cursor control," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 203–205, 2000.
- [9] F. Lotte, L. Bougrain, A. Cichocki et al., "A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, no. 3, p. 031005, 2018.
- [10] S. Fazli, C. Grozea, M. Danoczy, B. Blankertz, F. Popescu, and K.-R. Müller, "Subject independent EEG-based BCI decoding," *In Advances in Neural Information Processing Systems*, vol. 22, pp. 513–521, 2009.
- [11] T. Xu, Y. Zhou, Z. Wang, and Y. Peng, "Learning emotions EEG-based recognition and brain activity: a survey study on BCI for intelligent tutoring system," *Procedia Computer Science*, vol. 130, pp. 376–382, 2018.
- [12] A. C. Tsoi, D. S. C. So, and A. Sergejew, "Classification of Electroencephalogram using artificial neural networks," *In Advances in Neural Information Processing Systems*, vol. 6, pp. 1151–1158, 1994.
- [13] R. T. Schirrmester, J. T. Springenberg, L. D. J. Fiederer et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
- [14] Y. Xu, S. Haykin, and R. J. Racine, "Multiple window time-frequency distribution and coherence of EEG using Slepian sequences and Hermite functions," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 7, pp. 861–866, 1999.
- [15] T. Mullen, C. Kothe, Y. M. Chi et al., "Real-time modeling and 3D visualization of source dynamics and connectivity using wearable EEG," in *Proceedings of 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2184–2187, Osaka, Japan, July 2013.



- [16] Y. Liu, O. Sourina, M. K. Nguyen et al., “Real-time EEG-based human emotion recognition and visualization,” in *Proceedings of 2010 International Conference On Cyberworlds*, pp. 262–269, Singapore, Singapore, October 2010.
- [17] Y.-J. Liu, M. Yu, G. Zhao, J. Song, Y. Ge, and Y. Shi, “Real-time movie-induced discrete emotion recognition from EEG signals,” *IEEE Transactions on Affective Computing*, vol. 9, no. 4, pp. 550–562, 2018.
- [18] X.-W. Wang, D. Nie, and B.-L. Lu, “Emotional state classification from EEG data using machine learning approach,” *Neurocomputing*, vol. 129, pp. 94–106, 2014.
- [19] J. Raven, J. C. Raven, and J. H. Court, *Manual for Raven’s Progressive Matrices and Vocabulary Scales*, Oxford Psychologists, Oxfordshire, UK, 1998.
- [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, July 2016.
- [21] E-Prime® | Psychology Software Tools, <https://pstnet.com/products/e-prime/>.
- [22] T. Xu, Y. Zhou, Y. Wang, Z. Zhao, and S. Li, “Guess or not?” in *Proceedings of the Extended Abstracts Of the 2019 CHI Conference On Human Factors In Computing Systems*, pp. LBW1210:1–LBW1210:6, New York, NY, USA, 2019.
- [23] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning convolutional neural networks for resource efficient inference,” in *Proceedings of the International Conference on Learning Representations*, Toulon, France, April 2017.