Double Deep Q-Network Decoder Based on EEG Brain-Computer Interface

REN Min, XU Renyu, ZHU Ting
(Southwest Jiaotong University, Chengdu 611756, China)

DOI: 10.12142/ZTECOM.202303002
https://link.cnki.net/urlid/34.1294.TN.20230907.1041.002, published online September 7, 2023
Manuscript received: 2023-06-08

Abstract: Brain-computer interfaces (BCI) use neural activity as a control signal to enable direct communication between the human brain and external devices. The electrical signals generated by the brain are captured through electroencephalogram (EEG) and translated into neural intentions reflecting the user's behavior. Correct decoding of the neural intentions then facilitates the control of external devices. Reinforcement learning-based BCIs enhance decoders to complete tasks based only on feedback signals (rewards) from the environment, building a general framework for dynamic mapping from neural intentions to actions that adapt to changing environments. However, using traditional reinforcement learning methods can have challenges such as the curse of dimensionality and poor generalization. Therefore, in this paper, we use deep reinforcement learning to construct decoders for the correct decoding of EEG signals, demonstrate its feasibility through experiments, and demonstrate its stronger generalization on motion imaging (MI) EEG data signals with high dynamic characteristics.

Keywords: brain-computer interface (BCI); electroencephalogram (EEG); deep reinforcement learning (Deep RL); motion imaging (MI) generalizability

Citation (Format 1): REN M, XU R Y, ZHU T. Double deep Q-network decoder based on EEG brain-computer interface [J]. ZTE Communications, 2023, 21(3): 3 - 10. DOI: 10.12142/ZTECOM.202303002
Citation (Format 2): M. Ren, R. Y. Xu, and T. Zhu, "Double deep Q-network decoder based on EEG brain-computer interface," *ZTE Communications*, vol. 21, no. 3, pp. 3 - 10, Sept. 2023. doi: 10.12142/ZTECOM.202303002.

1 Introduction

Brain-computer interface (BCI) offers the possibility of direct communication between the human brain and external devices that perform operational tasks[1]. Researchers capture the electrical signals generated by the brain through the electroencephalogram (EEG) and convert them into neural intentions, which will be correctly decoded and used to control external devices, such as wheelchairs, robotic arms, and automatic vehicles[2 - 4]. Among other things, the correct decoding of the neural intention is a crucial step toward this goal, and the correct interpretation of brain activity can provide the external device with the required commands so that it can perform expected tasks. Within the different EEG systems, the motor imagery (MI) BCI[5 - 6] is a very flexible EEG paradigm, which can be used to distinguish between different intracerebral instructions and to control external devices to execute commands by "what is in mind".

Many methods based on traditional machine learning have been used for MI decoding and feature extraction. Among them, filter bank common spatial patterns (FBCSP)[5, 7] based on the characteristics of common spatial patterns (CSP) have achieved good performance. And some researchers have investigated an improved feature extraction method based on CSP to further improve the performance of BCI system[8 - 9]. In addition, linear discriminant analysis (LDA), support vector machines (SVM), etc., are used to find a projection or hyperplane to separate different categories by analyzing feature distribution[10 - 11]. Due to the limited spatial resolution, low signal-to-noise ratio (SNR), and high dynamic characteristics of MI, as well as the existence of a large amount of noise in EEG signals, the extraction of robust features from EEG data is a crucial step for the successful implementation of BCI. In recent years, the success of deep learning methods has alleviated the need for manual feature extraction to a large extent. As a result, many scholars have explored the application of deep learning in EEG signals. For example, the multi-layer perceptron (MLP) was used to correctly classify EEG signals[12]. Since convolutional neural networks (CNNs) can perceive multiple small domain features with the convolution process proceeding layer by layer and can automatically extract rich features to obtain a depth representation, many studies have tried to use CNNs into BCI to build end-to-end EEG decoding models and achieved good performance[13 - 14].

Supervised learning is a popular paradigm to implement BCI, but it requires an explicit supervised signal to learn.

Even so, frequent calibration (retraining) is necessary due to the plasticity of the brain. Therefore, some scholars have focused on developing an adaptive BCI architecture that allows interaction with a dynamic environment[15-17], where BCI users learn by trial-and-error to adjust their brain activity to the decoder by observing how the external device performs the task (using feedback information). Among them, reinforcement learning (RL)[18] is the general framework that makes the system adapt to the new environment. It is an interactive learning paradigm that can improve policies through constant interaction with the environment, aiming to learn the best mapping relationship from the environmental state to the action. Thus, an RL-based BCI framework is explored, which provides a general framework for constructing dynamic mappings from neural intentions to actions adapted to changing environments, requiring only a scalar signal (reward) feedback from the environment to strengthen the decoder to complete the task, rather than a specific permanently available supervisory signal[19]. At the same time, an RL-based BCI architecture is a more reasonable learning solution to those patients unable to produce precise limb movements. In this case, they only need to understand which action will yield the greatest return when reaching the goals in their environment.

Multiple studies have shown that RL can be used in the rat EEG signal[20-21] and neuronal activity to control the basic BCI system[19, 22]. DIGIOVANNA et al.[19] first proposed a Q(λ)-learning algorithm with the temporal difference (TD) error in an RL-based BCI paradigm, which experimentally trained rats to control prostheses in a two-target selection task. Furthermore, SANCHEZ et al.[23] applied Q(λ)-learning to predict one-step actions, extending the RL-based BCI framework to primates performing center-out tasks. In addition, the BCI paradigm using RL has been successfully applied to closed-loop experiments of intracortical signals in monkeys[24-25]. BAE et al.[26] combined the kernel temporal differences (KTD) (λ) algorithm with the Q-learning algorithm to obtain a reinforcement learning-based neural decoding algorithm (Q-KTD), and the feasibility of this method for BCI decoding was demonstrated in a center-out extension task of intracortical signals in monkeys. THAPA et al.[27] further investigated the applicability and feasibility of Q-KTD in an EEG-based BCI system, demonstrating that the Q-KTD algorithm can correctly learn the mapping between neural intentions in EEG signals and external device control commands. However, there are still some challenges in EEG-based RL interface using Q-KTD: 1) The number of kernel units increases with the number of samples; 2) the curse of dimensionality limits the decoding capability of the Q-KTD algorithm; 3) the Q-KTD decoding technique based on Q-Learning has a generalization problem and requires a long training time.

To overcome the above problems, this paper proposes to use double deep Q-network (DDQN), a deep reinforcement learning algorithm, to decode EEG. DDQN[28], as a variant of deep Q-networks (DQN)[29], uses a neural network to approximate the value function and takes into account the generalization while dealing with high-dimensional inputs. In addition, the dual-value network architecture of DDQN can effectively suppress the influence of overestimation of action values on the decision-making process and is robust to EEG signals that may have random interference.

In section 2, this paper introduces the DDQN algorithm and the basic paradigm based on reinforcement learning brain-computer interface. In section 3, the EEG decoder based on DDQN is described and the network structure diagram is given. In section 4, the feasibility and advantages of DDQN for EEG signal decoding are verified by comparative experiments. In section 5, this paper is summarized, and the prospect of future research is discussed.

2 Preliminary

This paper mainly adopts DDQN to perform the end-to-end decoding operation of EEG, and the related concepts and basic knowledge are introduced as follows.

2.1 Reinforcement Learning

RL is a learning framework for dealing with sequential decision problems, which can usually be modeled as a Markov Decision Process (MDP) that can be represented by a five-tuple (S, A, P, R, γ), where:

1) S denotes the state space and $s_t \in S$ represents the state of the agent at the moment t;

2) A denotes the action space and $a_t \in A$ represents the action executed by the agent at time t;

3) $P: S \times A \times S \rightarrow [0, 1]$ denotes the state transition probability, and $P(s_{t+1} | s_t, a_t)$ denotes the probability that the agent executes the action a_t in state s_t to the next state s_{t+1};

4) $R: S \times A \rightarrow R$ denotes the reward function and $R(s_t, a_t)$ represents the immediate reward obtained by the agent by executing the action a_t in the state s_t;

5) $\gamma \in [0, 1]$ is the discount factor used to balance immediate and delayed rewards.

The action selection of an agent in reinforcement learning obeys the policy π, which is expressed as the mapping relationship $\pi: S \rightarrow A$ between the state and the executable action of the agent. RL algorithms can be classified into two categories, policy-based and value-based methods. In the value-based RL method, the policy will not be updated explicitly, but a value table or value function is maintained, and new policies are derived from this value table or value function. The state-action value function is $Q^{\pi}: S \times A \rightarrow R$. $Q^{\pi}(s_t, a_t)$ represents the expected cumulative reward obtained by the agent executing action a_t in state s_t and following the current policy π until the end of the episode, which can be expressed as:

$$Q^{\pi}(s_t, a_t) = E_{\pi}\left\{\sum_t \gamma^t R(s_t, \pi(s_t)) \mid s_t = s,\ a_t = a\right\}. \quad (1)$$

The ultimate goal of the agent is to learn an optimal policy π^*, and the value function obtained under the optimal policy satisfies the Bellman optimality equation: $V^*(s_t) = \max_{a \in A} Q^*(s,a)$, i.e., the optimal value of the state is equal to the expected cumulative reward obtained by taking the optimal action in that state, and the optimal policy can be obtained from $\pi^* \in \operatorname{argmax}_a Q^*(s,a)$.

Q-learning is an RL algorithm based on value iteration, which directly estimates the optimal state-action value function Q^*. It updates the Q-function by the following rule:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\Big(R(s_t,a_t) + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\Big), \tag{2}$$

where s_{t+1} represents the next state reached by the agent executing action a_t in state s_t, and $\alpha \in [0,1]$ represents the learning rate. A common approach to deriving a policy based on the Q-function is the ε-greedy policy, which selects an action greedily with the probability of $1-\varepsilon$ based on the Q-function and performs any action randomly with the probability of ε. This facilitates the exploration of the agent in the environment and avoids falling into a local optimum.

2.2 Double Deep Q-Networks

When the state space is large or continuous, it is impractical to directly use the tabular Q-function for storing the values of all state-action pairs. A common solution is to approximate the Q-function using a function approximator, e. g., $Q(s_t,a_t) \approx Q(s_t,a_t,\theta)$, where $Q(s_t,a_t,\theta)$ represents the parametrized approximation of the Q-function. Specifically, DQN is a method that approximates the state action value function by the Q-learning algorithm through a neural network. In DQN, deep learning and reinforcement learning are combined through a convolutional neural network to approximate the state action value function, and high-dimensional states can be input, which solves the dimensional disaster problem faced by traditional Q-learning.

In the traditional Q-learning algorithm and DQN algorithm, directly selecting the action with the maximum Q value may cause the Q-value overestimation problem, which leads to over-optimistic estimation. DDQN[28] separates action selection and action value evaluation to avoid the overestimation problem. Like DQN, DDQN has two important ideas: the target network and experience replay mechanism. At each time step t in DDQN, the agent executes action a_t in current state s_t based on the current policy, receives the reward $R(s_t,a_t)$, and transforms to the next state s_{t+1}. The transition $(s_t,a_t,R(s_t,a_t),s_{t+1})$ is added to the experience pool D. The neural network parameters are continuously updated by a gradient descent minimization loss function. The neural network parameters are continuously updated by minimizing the loss function through gradient descent, where the loss function is expressed as the mean square error between the target value and the evaluated value, which is defined as:

$$L(\theta) = E_{(s_t,a_t,R(s_t,a_t),s_{t+1})}[\left(y^{\mathrm{DQN}} - Q(s_t,a_t;\theta)\right)^2], \tag{3}$$

where the target value y^{DQN} is defined as:

$$y^{\mathrm{DQN}} = R(s_t,a_t) + \gamma \max_{a'} Q(s_{t+1},a';\theta^-). \tag{4}$$

In Eqs. (3) and (4), θ denotes the online network parameters, θ^- denotes the target network parameters. $Q(s_t,a_t;\theta)$ denotes the online network output, and $Q(s_t,a_t;\theta^-)$ denotes the target network output, which is used to calculate the target value, where the target network has the same structure as the online network, except that its parameter values are replicated from the online network without τ steps, and the parameter representations of the target network remain unchanged during τ time steps.

The idea of DDQN is to decouple the action of selecting the maximum value in the target value and evaluating the value of the action, thus avoiding the problem of overestimation. DDQN uses the target network in DQN as the network for evaluation, without having to introduce an additional network. Therefore, in DDQN, the action is selected using the current Q-network, and then its value is evaluated using the target network. Its target value y^{DDQN} is:

$$y^{\mathrm{DDQN}} = R(s_t,a_t) + \gamma Q\Big(s_{t+1}, \arg\max_a Q(s_{t+1},a;\theta);\theta^-\Big). \tag{5}$$

The difference between DDQN and DQN is that the selection of the optimal action in DDQN is based on the online network Q with parameter θ, whereas the selection of the optimal action in DQN is based on the target network with parameter θ^-. VAN HASSELT et al.[28] have experimentally demonstrated that compared with DQN, DDQN can effectively reduce overestimation and obtain more stable learning.

2.3 Reinforcement Learning Brain Computer Interfaces

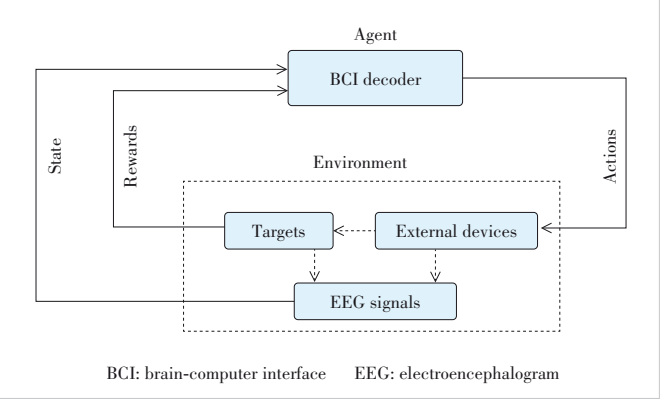
In recent years, RL has become a significant research interest in artificial intelligence. Through trial and error, the RL agent must discover which actions yield the maximum expected reward. Thus, the RL-based BCI attempts to allow BCI control algorithms to learn to complete tasks from interactions with the environment rather than explicit training signals. In fact, for many patients using BCI, the only signals available are their internal brain intention to complete the motor task and external feedback after completing the task, as opposed to specific supervised signals. The RL-based BCI attempts to learn a control policy by which, at any time t, the neural decoder observes a neural state $s_t \in S$, and the neural decoder outputs an action $a_t \in A$ based on the current policy, which

generates a control signal to the external device. After the external device completes the action, the neural decoder receives a feedback signal R_t. In future tasks, the neural decoder uses this feedback to continuously adjust the policy, which learns the optimal function mapping of the neural state to the action directly. The decoding structure is shown in Fig. 1.

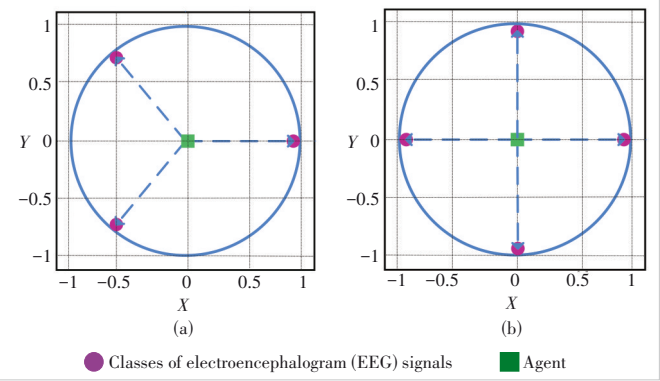
3 DDQN-Based EEG Decoder

An EEG signal decoder based on the Q-KTD RL algorithm provides the possibility of continuous learning of BCI, but its generalization is not negligible for a continuously useful decoder. Therefore, we try to use DDQN to decode the EEG signal correctly in this paper.

For the decoding task of EEG signals, a BCI decoder is considered a reinforcement learning agent, and the decoding of EEG signals is modeled as a common center-out task for BCI, associating the class of MI data with a specific direction, modeled as a single-step reinforcement learning problem, as shown in Fig. 2. A reinforcement learning environment located at the center of the origin (0, 0) with a radius of 1 is set up. In the center-out task, the reinforcement learning agent (the green square in Fig. 2) is located at the center of the origin (0, 0) at the beginning of each trial. By decoding each trial's MI data, the BCI decoder generates a specific action (one of up, down, left, and right), and the agent (located at the origin position (0, 0)) moves a distance of length 1 in the corresponding direction to a corresponding location (one of the purple circles in Fig. 2), and then receives an immediate reward based on the location reached by the agent. This paper uses a double deep Q-networks algorithm to train the agent to obtain a BCI decoder to decode EEG signals correctly.



▲Figure 1. Reinforcement learning (RL)-based BCI decoding structure



▲Figure 2. (a) Classical dataset and (b) BCI Competition IV-2a (BCI-2a) dataset are set to a center-out task. The center is located at the origin (0, 0), represented by a green square, and each class target is a purple circle

The state vector of DDQN is the EEG signal, and the agent takes action based on the current state. The optional action of the agent is the same as the label set of the EEG signal. According to the label information of the EEG signal, the feedback from the environment can be received, and the reward of the environment feedback contains two values of −1 and 1. If the current action performed by the agent is consistent with the label of the EEG signal, the environment feeds a positive reward value. Otherwise, the environment provides a negative value to the agent. The pseudocode of the algorithm is given by Algorithm 1. Moreover, we give the network architecture of the DDQN-based EEG signal decoder in Fig. 3.

Algorithm 1. DDQN-based EEG decoding

Input: the empty replay buffer D, initial network parameters θ, copy of $\theta\ \theta^{-}$, EEG signal sequences X, the training batch size N_b, explore probabilistic decay frequency N_ε, and target network replacement frequency N^{-}.
Output: action a_t
For episode=1 to M **do**
  Randomly initialize EEG signal sequences X
  If episode mod $N_\varepsilon = 0$
    $\varepsilon = \varepsilon \times$ RLepsilonDecayRate
  End if
  For t=0 to T **do**
    Set state $s_t \leftarrow X$ and select action a_t based on the ε-greedy policy
    Execute action a_t and observe reward r_t
    Store (s_t, a_t, r_t, s_{t+1}) in D
    Sample a minibatch of N_b tuples $(s, a, r, s') \sim \mathrm{Unif}(D)$
    Construct target values, one for each of the N_b tuples:

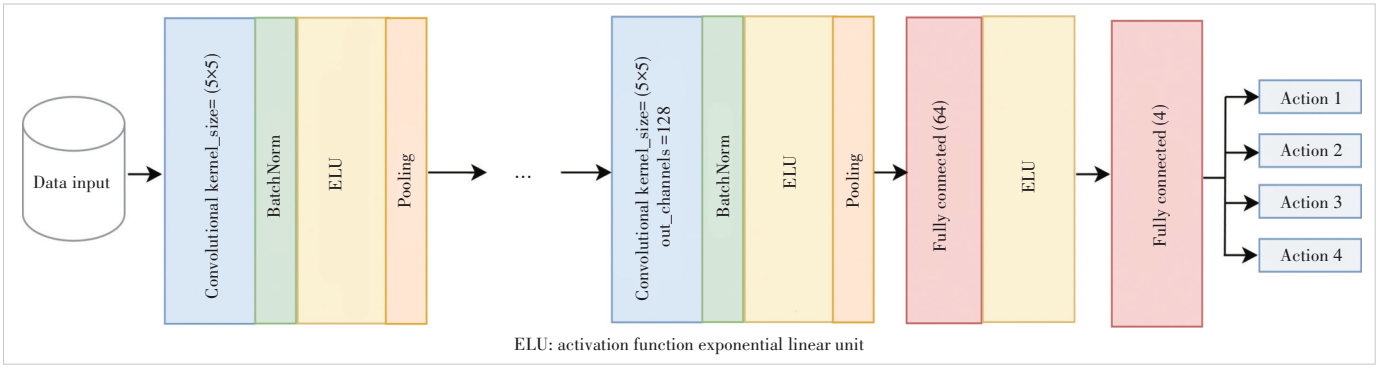
$$y_j^{\mathrm{DDQN}} = \begin{cases} r_j, & \text{if } s_{j+1} \text{ is terminal} \\ r_j + \gamma Q\left(s_{t+1}, \arg\max_a Q\left(s_{t+1}, a\,;\theta\right);\theta^{-}\right), & \text{otherwise} \end{cases}$$

    Do a gradient descent step with loss $\| y_j^{\mathrm{DDQN}} - Q(s_j, a_j\,;\theta)\|^2$
    Replace target parameters $\theta^{-} \leftarrow \theta$ every N^{-} steps
  End
End

4 Experimental Analysis

4.1 Experimental Data

We conducted experiments on two publicly available data sets: Nature's Scientific Data[30] and BCI Competition IV-2a



▲**Figure 3. Network structure of electroencephalogram (EEG) signal decoder based on double deep Q-network (DDQN)**

(BCI-2a)[31].

1) Experiment on Nature's Scientific Data: 21 electrodes were used to record EEG data from 13 healthy subjects, including 8 males and 5 females. This data set provided five different BCI paradigm data sets to imagine the movement of different body parts, such as the left hand, the right hand, or different finger movements. Among these available EEG data sets, we considered the first classical motor imagination dataset (Classical, CLA). The CLA dataset consisted of three types of motor imagination data using EEG signals, corresponding to left-hand movement, right-hand movement, and maintaining neutrality respectively. That is, the participants did not imagine anything. Six subjects in the CLA data were considered, specific to A to F subjects, since the "CLASubjectF1509163 StLRHand" data file contained only two category labels and was therefore not considered. The sampling frequency of the EEG signal was 200 Hz, and each collected data file contained 15 min sessions. Each 15-minute session included 300 trials. The total time of each trial was 3 s, which started with a one-second motion MI cue, and each trial lasted 1.5 – 2.5 s.

2) Experiment on BCI Competition IV-2a: This dataset is a publicly available dataset for BCI Competition IV, which is described in detail by TANGERMANN et al.[31] for the data characteristics of the competition. The BCI-2a dataset, which recorded EEG data from nine subjects using 22 electrodes, provided four different types of motor imagination data: left-hand movement, right-hand movement, exercise of both feet, and tongue movement imagination. The EEG signal was sampled at a frequency of 250 Hz, and the data for each subject consisted of two files, each consisting of six EEG recording blocks containing 48 trials, for a total of 576 trials.

4.2 Experimental Methods

The collection of the CLA data set is to extract the brain imagination data of 21 channels continuously for 0.85 s at a sampling frequency of 200 Hz, starting from the action stimulus for each subject[30]. For the BCI-2a dataset, some researchers have tried to use a relatively large window (about 3 s to 4 s) for their studies[32 – 33], but a relatively small window is more realistic for online BMI[27]. Therefore, the proposed method uses a 0.85-second EEG window to decode subjects' motor images. We first cropped the 0.85-second EEG signal at [0, 0.85] after the beginning of the trial for CLA and at [2, 2.85] after the beginning of the visual cue for BCI-2a. The CLA data consists of 21 channels, where the number of samples in each channel is 170 (the same as sampling frequency 200 Hz × 0.85 s), and the BCI-2a data set has a total of 22 channels, in which the number of samples per channel is 213 (about sampling frequency 250 Hz × 0.85 s).

For the CLA dataset, we evaluated the performance of the EEG signal within 0.85 s by dividing it into a training set and a test set, respectively, and as in Ref. [27], we executed 10 Monte Carlo trials at 100 episodes, and in each trial, the sequences of the trials were randomized. The performance was observed on the training set based on the success rate of the trials, which was calculated as the ratio of the number of successful trials at each step to reach the specified goal to the total number of trials considered. For the BCI-2a dataset, the performance was evaluated directly based on the existing training and test sets of each subject. In DDQN, we adopted the ε-greedy method for the exploration strategy, where the exploration probability of the intelligence was set to $\varepsilon = 0.1$ at the beginning of the trial and decays every 20 episodes, with each exploration probability decaying to half of the original one, so that the agent would be more inclined to be exploited as the trial progressed.

4.3 Feature Extraction

In order to illustrate the advantages of DDQN for EEG decoding, this paper compares the DDQN algorithm with classical supervised learning algorithms (the SVM, decision tree, and random forest) and the Q-KTD algorithm based on traditional reinforcement learning. In order to make the experiment more convincing, we follow the feature extraction approach introduced in Ref. [27] to construct Features 1 and 2, which is constructed as follows :

Feature 1: In order to obtain the complete information held in the EEG data, Feature 1 was extracted by first cropping the EEG signal data for 0.85 s, and then concatenating each channel of the cropped data as one motor imagery state vector for

each experiment. For example, for the BCI-2a dataset which has 22 channel numbers, each channel has 213 samples, so the size of the motion imagery state vector in each experiment is 4 686 (equal to 213 samples × 22 channels). Alternatively, for the CLA data, which has 21 channels, each channel contains 170 samples, so the size of the state vector in each experiment is 3 570 (equal to 170 samples × 21 channels).

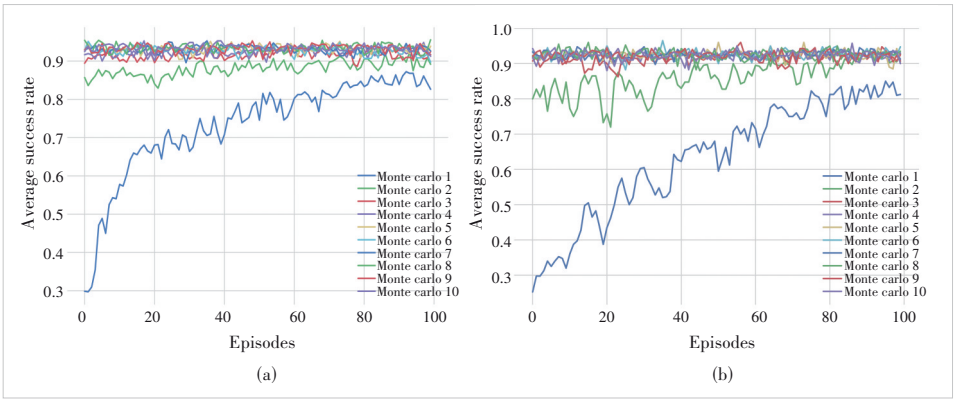
Feature 2: It has been proved that EEG signals contain frequency information[34], therefore, this paper adopts the same way as Ref. [27] to extract the frequency information as Feature 2. For the dataset BCI-2a and CLA dataset, firstly, the fast Fourier transform is performed on the EEG data with a cropped duration of 0.85 s, and then the selected complex frequency components correspond to the real and imaginary values, respectively. For the BCI-2a data, the real and imaginary values of the transformed 0 – 15 Hz were used for frequency classification, yielding a 550 dimensional feature state vector for each experiment. For the CLA dataset, using the frequency components between 0 – 5 Hz, a complex frequency feature with 5 dimensions of real values and 4 dimensions of imaginary values for each channel is obtained, and the components for each channel are concatenated to obtain a feature state vector with a dimension of 189 (equal to 9 × 21 channels).

4.4 Experimental Results and Analysis

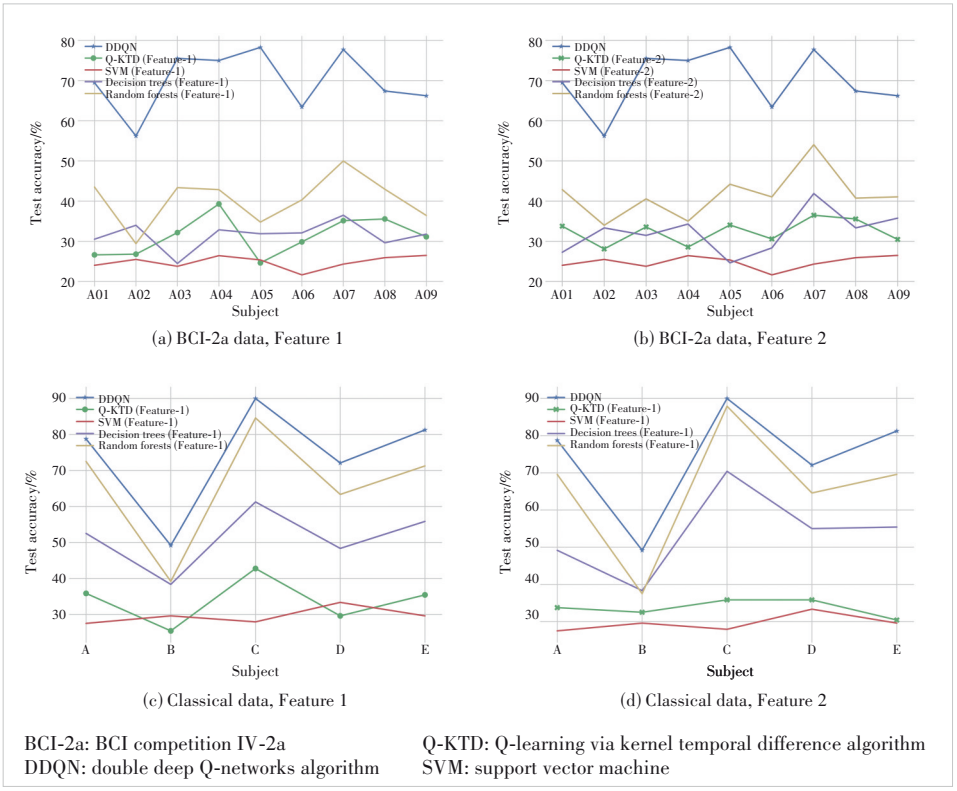
Firstly, the reinforcement learning agent is trained on the CLA and BCI-2a training sets, respectively, and its learning curve is observed. Fig. 4 shows the learning curves of the first subject on each of the two datasets. The learning curves show that the DDQN algorithm can correctly learn the correct mapping of EEG signals to actions directly in the MI center-out task with full learning by the agent as the experiment progresses.

Secondly, the generalization effect of the DDQN algorithm on the test set was compared with that of the traditional supervised learning algorithms (the SVM, decision tree, and random forest) and the Q-KTD algorithm. The result was shown in Fig. 5. For the same EEG data set, under different feature extraction, the generalization effect of DDQN algorithm on EEG decoding was significantly better than Q-KTD algorithm. Moreover, due to the small sample size and high dynamic characteristics of EEG signals, the classification performance of traditional supervised learning algorithms is poor, compared with DDQN-based EEG decoding.

Thirdly, the running time of DDQN and Q-KTD algorithms



▲ **Figure 4. Learning curves of double deep Q-network (DDQN) on (a) Classical (CLA) dataset and (b) BCI competition IV-2a EEG data (BCI-2a) dataset, where each color represents the average success rate of 10 Monte Carlo trials**



BCI-2a: BCI competition IV-2a
DDQN: double deep Q-networks algorithm
Q-KTD: Q-learning via kernel temporal difference algorithm
SVM: support vector machine

▲ **Figure 5. The generalizability of DDQN is compared with other classical algorithms for decoding based on Feature 1 (left) and Feature 2 (right) on the (a)(b)BCI-2a dataset and (c)(d) Classical (CLA) dataset, respectively**

is compared when training the agent, and the learning time of the agent under the two datasets is shown in Tables 1 and 2. According to the results, it is clear that the agent using DDQN learns much faster compared with the Q-KTD algorithm using Feature 1, and the learning time is not much different from the Q-KTD algorithm using Feature 2. However, the generalization of DDQN is much better than the Q-KTD algorithm.

Finally, in order to reflect the stability of decoding based on the deep reinforcement learning algorithm, we conducted 10 repeated experiments under different random seeds, and described their mean and standard deviation. The experimental results are shown in Fig. 6.

5 Conclusions

This paper investigates the applicability and feasibility of the deep double Q reinforcement learning algorithm in the brain-computer interface. We use two different EEG signal datasets and evaluate the performance of DDQN on both the datasets. The experimental results show that DDQN performs well in the correct decoding of EEG signals and has better generalization. This indicates that deep reinforcement learning can learn the correct decoding of EEG signals through feedback signals and has better generalization than the Q-KTD reinforcement learning algorithm. In the future, we will investigate further applications of deep reinforcement learning in EEG signals.

▼ Table 1. Comparison of learning time of DDQN and Q-KTD algorithm agent on classical (CLA) dataset

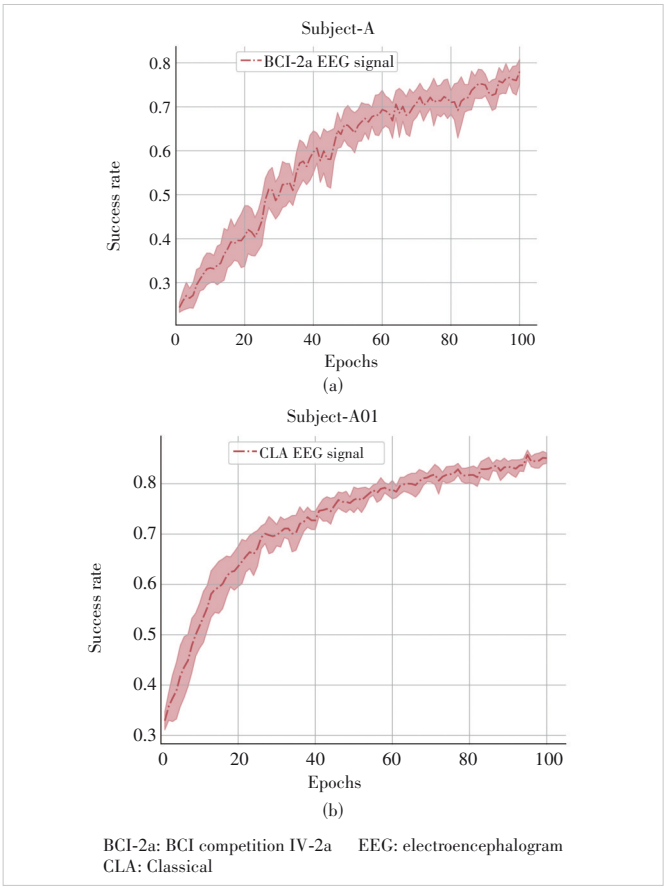
Subject	Algorithm		
	DDQN	Q-KTD_Feature 1/min	Q-KTD_Feature 2/min
A01	18.69	103.14	10.72
A02	9.3	117.8	10.94
A03	19.26	103.1	11.05
A04	16.55	79.73	8.82
A05	19.95	112.97	10.67
A06	14.96	54.86	6.43
A07	17.16	108.93	11
A08	18.11	103.73	10.53
A09	16.71	87.46	8.44

DDQN: double deep Q-networks algorithm
Q-KTD: Q-learning via kernel temporal difference algorithm

▼ Table 2. Comparison of learning time of DDQN and Q-KTD algorithm agent on BCI-2a dataset

Subject	Algorithm		
	DDQN	Q-KTD_Feature 1/min	Q-KTD_Feature 2/min
A	25.94	229.93	11.64
B	25.31	232.93	11.69
C	22.59	235.17	12.62
D	25.6	234.75	13.12
E	25.35	233.66	12.62

BCI-2a: BCI competition IV-2a
DDQN: double deep Q-networks algorithm
Q-KTD: Q-learning via kernel temporal difference algorithm



BCI-2a: BCI competition IV-2a EEG: electroencephalogram
CLA: Classical

▲ Figure 6. Results of experiments on (a) BCI-2a and (b) CLA under 10 different random seeds

References

[1] WILLETT F R, AVANSINO D T, HOCHBERG L R, et al. High-performance brain-to-text communication via handwriting [J]. Nature, 2021, 593(7858): 249 – 254. DOI: 10.1038/s41586-021-03506-2
[2] CRUZ A, PIRES G, LOPES A, et al. A self-paced BCI with a collaborative controller for highly reliable wheelchair driving: experimental tests with physically disabled individuals [J]. IEEE transactions on human-machine systems, 2021, 51(2): 109 – 119. DOI: 10.1109/THMS.2020.3047597
[3] SCHWARZ A, HÖLLER M K, PEREIRA J, et al. Decoding hand movements from human EEG to control a robotic arm in a simulation environment [J]. Journal of neural engineering, 2020, 17(3): 036010. DOI: 10.1088/1741-2552/ab882e
[4] SONG Y H, WU W F, LIN C Q, et al. Assistive mobile robot with shared control of brain-machine interface and computer vision [C]//4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2020: 405 – 409. DOI: 10.1109/ITNEC48623.2020.9085096
[5] ANG K K, CHIN Z Y, WANG C C, et al. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b [J]. Frontiers in neuroscience, 2012, 6: 39. DOI: 10.3389/fnins.2012.00039
[6] TONIN L, CARLSON T, LEEB R, et al. Brain-controlled telepresence robot by motor-disabled people [C]//Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2011: 4227 – 4230. DOI: 10.1109/IEMBS.2011.6091049
[7] GROSSE-WENTRUP M, BUSS M. Multiclass common spatial patterns and information theoretic feature extraction [J]. IEEE transactions on biomedical engineering, 2008, 55(8): 1991 – 2000. DOI: 10.1109/TBME.2008.921154
[8] JIN J, XIAO R, DALY I, et al. Internal feature selection method of CSP based on L1-norm and Dempster-Shafer theory [J]. IEEE transactions on neural networks and learning systems, 2020, 32(11): 4814 – 4825. DOI: 10.1109/TNNLS.2020.3015505

[9] JIN J, MIAO Y, DALY I, et al. Correlation-based channel selection and regularized feature optimization for MI-based BCI [J]. Neural networks, 2019, 118: 262 – 270. DOI: 10.1016/j.neunet.2019.07.008
[10] FU R R, TIAN Y S, BAO T T, et al. Improvement motor imagery EEG classification based on regularized linear discriminant analysis [J]. Journal of medical systems, 2019, 43(6): 169. DOI: 10.1007/s10916-019-1270-0
[11] LIU Y X, ZHOU W D, YUAN Q, et al. Automatic seizure detection using wavelet transform and SVM in long-term intracranial EEG [J]. IEEE transactions on neural systems and rehabilitation engineering, 2012, 20(6): 749 – 755. DOI: 10.1109/TNSRE.2012.2206054
[12] SAMUEL O W, GENG Y J, LI X X, et al. Towards efficient decoding of multiple classes of motor imagery limb movements based on EEG spectral and time domain descriptors [J]. Journal of medical systems, 2017, 41(12): 194. DOI: 10.1007/s10916-017-0843-z
[13] LIN C T, CHUANG C H, HUNG Y C, et al. A driving performance forecasting system based on brain dynamic state analysis using 4-D convolutional neural networks [J]. IEEE transactions on cybernetics, 2021, 51(10): 4959 – 4967. DOI: 10.1109/TCYB.2020.3010805
[14] AMIN S U, ALSULAIMAN M, MUHAMMAD G, et al. Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion [J]. Future generation computer systems, 2019, 101: 542 – 554. DOI: 10.1016/j.future.2019.06.027
[15] TAYLOR D M, TILLERY S I H, SCHWARTZ A B. Direct cortical control of 3D neuroprosthetic devices [J]. Science, 2002, 296(5574): 1829 – 1832. DOI: 10.1126/science.1070291
[16] GAGE G J, LUDWIG K A, OTTO K J, et al. Naïve coadaptive cortical control [J]. Journal of neural engineering, 2005, 2(2): 52 – 63. DOI: 10.1088/1741-2560/2/2/006
[17] VELLISTE M, PEREL S, SPALDING M C, et al. Cortical control of a prosthetic arm for self-feeding [J]. Nature, 2008, 453(7198): 1098 – 1101. DOI: 10.1038/nature06996
[18] SUTTON R S, BARTO A G. Reinforcement learning: an introduction [M]. Cambridge, USA: MIT press, 2018
[19] DIGIOVANNA J, MAHMOUDI B, FORTES J, et al. Coadaptive brain: machine interface via reinforcement learning [J]. IEEE transactions on biomedical engineering, 2009, 56(1): 54 – 64. DOI: 10.1109/TBME.2008.926699
[20] ITURRATE I, MONTESANO L, MINGUEZ J. Robot reinforcement learning using EEG-based reward signals [C]//IEEE International Conference on Robotics and Automation. IEEE, 2010: 4822 – 4829. DOI: 10.1109/ROBOT.2010.5509734
[21] MATSUZAKI S, SHIINA Y, WADA Y. Adaptive classification for brain-machine interface with reinforcement learning [C]//18th International Conference on Neural Information Processing. ICONIP, 2011: 360 – 369. DOI: 10.1007/978-3-642-24955-6_44
[22] MAHMOUDI B, SANCHEZ J C. A symbiotic brain-machine interface through value-based decision making [J]. PLoS one, 2011, 6(3): e14760. DOI: 10.1371/journal.pone.0014760
[23] SANCHEZ J C, TARIGOPPULA A, CHOI J S, et al. Control of a center-out reaching task using a reinforcement learning brain-machine interface [C]//5th International IEEE/EMBS Conference on Neural Engineering. IEEE, 2011: 525 – 528. DOI: 10.1109/NER.2011.5910601
[24] POHLMEYER E A, MAHMOUDI B, GENG S J, et al. Using reinforcement learning to provide stable brain-machine interface control despite neural input reorganization [J]. PLoS one, 2014, 9(1): e87253. DOI: 10.1371/journal.pone.0087253
[25] MARSH B T, TARIGOPPULA V S A, CHEN C, et al. Toward an autonomous brain machine interface: integrating sensorimotor reward modulation and reinforcement learning [J]. Journal of neuroscience, 2015, 35(19): 7374 – 7387. DOI: 10.1523/jneurosci.1802-14.2015
[26] BAE J, CHHATBAR P, FRANCIS J T, et al. Reinforcement learning via kernel temporal difference [C]//Annual International Conference of IEEE Engineering in Medicine and Biology Society. IEEE, 2011: 5662 – 5665. DOI: 10.1109/IEMBS.2011.6091370
[27] THAPA B R, TANGARIFE D R, BAE J. Kernel temporal differences for EEG-based reinforcement learning brain machine interfaces [C]//44th Annual International Conference of IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2022: 3327 – 3333. DOI: 10.1109/EMBC48229.2022.9871862
[28] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning [C]//AAAI Conference on Artificial Intelligence. ACM, 2016: 2094 – 2100. DOI: 10.5555/3016100.3016191
[29] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning [C]//NIPS Deep Learning Workshop. NIPS, 2013. DOI: 10.48550/arXiv.1312.5602
[30] KAYA M, BINLI M K, OZBAY E, et al. A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces [J]. Scientific data, 2018, 5(1): 1 – 16. DOI: 10.1038/sdata.2018.211
[31] TANGERMANN M, MÜLLER K R, AERTSEN A, et al. Review of the BCI competition IV [J]. Frontiers in neuroscience, 2012, 6: 55. DOI: 10.3389/fnins.2012.00055
[32] QI F, WANG W, XIE X, et al. Single-trial eeg classification via orthogonal wavelet decomposition-based feature extraction [J]. Frontiers in Neuroscience, 2021, 15: 715855. DOI: 10.3389/fnins.2021.715855
[33] MUSALLAM Y K, ALFASSAM N I, MUHAMMAD G, et al. Electroencephalography-based motor imagery classification using temporal convolutional network fusion [J]. Biomedical signal processing and control, 2021, 69: 102826. DOI: 10.1016/j.bspc.2021.102826
[34] KEERTHI KRISHNAN K, SOMAN K P. CNN based classification of motor imaginary using variational mode decomposed EEG-spectrum image [J]. Biomedical engineering letters, 2021, 11(3): 235 – 247. DOI: 10.1007/s13534-021-00190-z

Biographies

REN Min received her BS degree from the School of Mathematics and Information, China West Normal University in 2021. She is currently working toward an MS degree at Southwest Jiaotong University, China. Her research interests include reinforcement learning and its application and data mining.

XU Renyu (ryxu@swjtu.edu.cn) received her PhD degree in mathematics and information science from Paris Saclay University, France in 2017. She is currently a lecturer in the school of mathematics, Southwest Jiaotong University, China. Her current research interests include reasoning with graph theory and machine learning.

ZHU Ting received her BS degree from the School of Mathematics and Software Science, Sichuan Normal University, China in 2021. She is now working toward an MS degree at Southwest Jiaotong University, China. Her research interests include reinforcement learning and deep reinforcement learning.