ISSN: 2664-9640

Research on Power Grid Reactive Voltage Optimization Method Based on Deep Deterministic Policy Gradient

Gang Li [1,2], Zhiyang Wang [1]

[1]Department of Computer, North China Electric Power University, Baoding 071003, Hebei Province, China;

[2]Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, Baoding 071003, Hebei Province, China;

Abstract

Voltage control is one of the important research contents and guarantee for the safe operation of power system. In this paper, a reactive power and voltage optimization control strategy based on Deep Reinforcement Learning (DRL) technology is explored. A deep deterministic policy gradient (DDPG) algorithm is established on the base of reactive power and voltage optimization model and control characteristics of the power system, arming at minimizing the network loss and the overall deviation of the grid voltage. Target network is added in the algorithm. The feasibility and effectiveness of the algorithm is demonstrated and proved with simulations by applying the constraints training model of the power system.

Keywords

Deep Reinforcement Learning; DDPG; Reactive power; Voltage.

1. Introduction

China has put forward the goals of "carbon neutrality" and "carbon peaking", and further proposed the future development guideline for building a new power system with renewable energy being the main part[1], aiming at coping with the aggravation of environmental pollution and greenhouse effect caused by fossil energy consumption. A high proportion of new energy access will become the basic feature and development form for the power system[1,2]. However, the output of the power generator of new energy forms, such as wind power and photovoltaics, has the character of volatility and uncertainty, which has a great impact on grid voltage and reactive power distribution.

Voltage is an important indicator to measure the power quality in the power system, and the voltage level of the power system generally depends on the reactive power balance of the system. Reactive power optimization is usually regarded as a multi-variable, multi-constraint nonlinear optimization problem [3]. Reactive power optimization methods based on operations research such as Newton's method [4], quadratic programming method [5] and other methods have the demerits of difficult and long solving time. Heuristic algorithms such as Genetic algorithms (GA) [6], Particle Swarm Optimization (PSO) [7], Simulated Annealing (SA) [8], etc. are also applied to this issue. These kinds of algorithms have the character of faster speed, but hampered by problems of susceptibility to fall into local optimum and poor convergence compared with the methods based on operations research.

ML-based (Machine Learning) reactive power and voltage optimization schemes have been preliminarily applied in power systems with the gradual development of ML technology. A reinforcement learning model was built with the goals of minimizing the voltage deviation of the dominant node in the partition and the reactive power output ratio of the generator, and

ISSN: 2664-9640

solves the multi-objective coordinated secondary voltage control[9]. The method of discretization of continuous problems has applied, and the Q-learning method has been used to the reactive power control issue of the nine-zone schematic diagram, achieving good results[10]. Compared with traditional methods, the ML-based optimization scheme does not rely on environmental modeling, and does not rely solely on the accurate prediction of the supply side and the demand side with randomness [11], but there are drawbacks of excessive dimensions of action and sparse actions.

Under the above background, this paper establishes a deep reinforcement learning model for reactive power and voltage optimization control, which integrates voltage deviation and network loss as the objective function, and analyzes the state space and action space under the background of the power system, and realizes the interaction between the power grid and the deep reinforcement learning agent. DDPG algorithm is used to train the model, and the superiority of the proposed algorithm and the model is proved by simulation.

2. DeepReinforcementLearning

2.1. Reinforcement Learning

Reinforcement Learning (RL) is one of the paradigms and methodologies of machine learning. It adopts the "try and fail" mechanism in human and animal learning, and emphasizes learning in interaction with the environment, and uses evaluative feedback signals to achieve the optimization of decision-making. In reinforcement learning, the agent (Agent) obtains the state (State) by observing the environment (Environment), calculates the action (Action) that needs to be performed in the current state by the policy (Policy), and corrects the policy according to the obtained reward (Reward). The purpose of completing the target can finally be achieved by repeating the above process. The key process of reinforcement learning is shown in Figure 1:

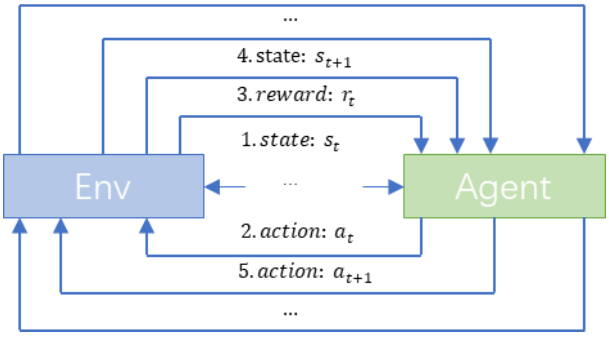

Figure 1: The key process of reinforcement learning

In the figure, s_t and s_{t+1} represent the environmental state observed by the agent at the time t and $t+1$respectively; a_t and a_{t+1} represent the action calculated by the agent according to the state s_t and s_{t+1}; r_t represents the reward for the action; the policy function π is a probability density function conditioned on the state s. Therefore, the calculated expression for a_t is written as:

$$a_t = \pi(a \mid s_t) \tag{1}$$

The expression for calculating the state at the next moment s_{t+1} is written as:

$$s_{t+1} = p(s \mid s_t, a_t) \tag{2}$$

where the environment transfer function p represents the probability density function of the transition from state s_t to s_{t+1} in the case of state s and action a.

ISSN: 2664-9640

2.2. Discounted Reward

The process of reinforcement learning can be abstracted as Markov Decision Process (MDP). The above process starts from the state s_0 and ends at the s_{end}, its Markov process is shown in Figure 2:

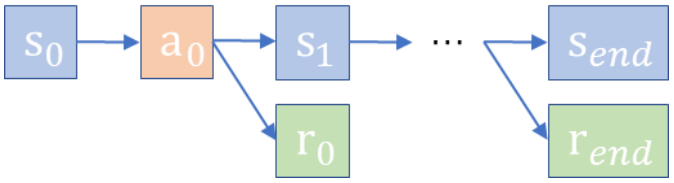

Figure2: Reinforcement learning Markov process

The overall evaluation of the action a_t at moment t should not only consider its current reward, but also its impact on the subsequent Markov chain trajectory. Therefore, define the reward u_t of the action a_t at moment t as:

$$u_t = r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \ldots = \sum_{i=0}^{+\infty} \gamma^i \cdot r_{t+i} \tag{3}$$

where γ is the discount rate, ranging $[0,1)$, u_t is also known as the discount return. The impact of the current action on subsequent rewards diminishes over time, and the discount rate reflects the degree of attenuation. The reward r_t has practical significance, so it is bounded by the lower bound m and upper bound M:

$$m \le r_t \le M \tag{4}$$

So:

$$\begin{cases} u_t = \sum_{i=0}^{+\infty} \gamma^i \cdot r_{t+i} \le M \cdot \sum_{i=0}^{+\infty} \gamma^i = M \cdot \frac{1}{1-\gamma} \\ u_t = \sum_{i=0}^{+\infty} \gamma^i \cdot r_{t+i} \ge m \cdot \sum_{i=0}^{+\infty} \gamma^i = m \cdot \frac{1}{1-\gamma} \end{cases} \tag{5}$$

And the lower bound and upper bound:

$$m \cdot \frac{1}{1-\gamma} \le u_t \le M \cdot \frac{1}{1-\gamma} \tag{6}$$

u_t has practical physical meaning, the discount rate and its range of values guarantee the boundedness of u_t.

2.3. Optimal Action Value Function

The action value function $Q_\pi(s_t, a_t)$ expresses the expectation of the reward based on the action a_t of the policy function π in the state s_t, denoted as:

$$Q_\pi(s_t, a_t) = E[u_t \mid s_t, a_t] \tag{7}$$

The optimal action π' is the strategy that maximizes the action value function, and the optimal action value function is the action value function under the optimal strategy π', denoted as:

$$Q'(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t) \tag{8}$$

In reinforcement learning, the strategy function π determines the distribution of actions in a specific state, the action value function Q_π is used to evaluate performance of performing actions a_t in the current state s_t under a specific strategy π, and the optimal action value function Q' can evaluate the performance of actions a_t in a specific state s_t. The reinforcement

ISSN: 2664-9640

learning by optimizing the optimal action value function is the reinforcement learning based on the optimal action value.

2.4. Deep Reinforcement Learning

In complex scenarios, the optimal action-value function Q' is difficult to build and optimize manually. Deep reinforcement learning can well fit the characteristics of any function by a large number of training with the help of deep neural network, and the use of deep neural network to fit the optimal action value function can solve the problem that the function is difficult to construct and train.

Classical deep reinforcement learning methods can better solve the problem of control strategy selection in low-dimensional discrete action spaces. For continuous control problems, traditional methods generally use the method of continuous space discretization, which transforms the original problem into a control problem in discrete space and then uses classical deep reinforcement learning methods to solve it.

However, the discretization of continuous space has many problem including the difficulty in defining the step size of the discretization, the explosion of the dimensions in the action space and the state space after the discretization. Therefore, the discretization method is generally only suitable for problems with small degrees of freedom. For large-scale control problems of power grids, the discretization method is obviously difficult to apply. The deep reinforcement learning algorithm in the continuous action space represented by the deep deterministic policy gradient can better adapt to and solve the control problem of the power grid.

3. Modeling of Reactive Power and Voltage Optimization Scheme

3.1. Reactive Power and Voltage Optimization Model

The grid reactive power and voltage optimization needs to consider both the reactive power optimization and the voltage optimization under the condition of ensuring the constraints of the power system. The goal of reactive power optimization is to reduce active power losses in the grid:

$$g_p = \sum_{i=1}^{l} P_{loss_i} \tag{9}$$

where g_p represents the active power loss of the grid, P_{loss_i} represents the active power loss of the i *th* link line. The goal of voltage optimization is to minimize the overall deviation of the grid voltage, i.e., to maximize the accuracy of the grid voltage. The bus voltage deviation is defined as:

$$\Delta v = \frac{|V_{base} - V|}{V_{base}} \tag{10}$$

where V_{base} denotes the standard voltage of the current bus, V denotes the actual voltage of the current bus, and δ denotes the voltage deviation of the bus, respectively. Then the bus voltage accuracy is:

$$\upsilon = 1 - \Delta v \tag{11}$$

So, grid voltage accuracy:

$$g_v = \sqrt[n]{\prod_{i=1}^{n} \upsilon_i} \tag{12}$$

The constraints of the power grid include bus voltage constraints, link line power constraints, generator output constraints, and power flow constraints, expressed as:

ISSN: 2664-9640

$$\begin{cases} U_{\min} \le U \le U_{max} \\ S_{L\min} \le S_L \le S_{Lmax} \\ S_{Gmin} \le S_G \le S_{Gmax} \\ G(T_d) = 0 \end{cases} \tag{13}$$

3.2. Deep Deterministic Policy Gradient Algorithm

Deep Deterministic Policy Gradient is an improved Deep Q-Learning (DQN) based on the Deterministic Policy Gradient (DPG) method, which estimates the optimal policy function and value function by a deep neural network to solve continuous actions Control problems in space[12]. DDPG is a deep reinforcement learning strategy based on the Actor-Critic method, which includes two parts of the deep neural network, the Policy Network and the Value Network.

3.2.1. Algorithm Process

The policy network guides the Agent to choose actions:

$$a = \pi(s;\theta) \tag{14}$$

where θ is the neural parameter of the policy network, s is the grid state observed by the agent, a represents the action selected by the policy function π. Unlike traditional deep reinforcement learning that selects actions based on probability distributions, the action a here is deterministic, i.e., it's non-random outputsaction vector based on state s and network parameter θ.

The value network evaluates the performance the actions taken by the agent in the current state, expressed as:

$$v = q(s,a;\omega) \tag{15}$$

where ω is the value network neural parameter, q is the value network function, and v denotes the value network's evaluation of the selected action a in the state s.

Defines an empirical transition of deep reinforcement learning, which includes the state s_t at moment t, the action a_t, the reward r_t and the state s_{t+1} at moment $t+1$:

$$(s_t, a_t, r_t, s_{t+1}) \tag{16}$$

For each transition, use Temporal-Difference (TD) to update the network parameters θ and ω:

For the transition at moment t, calculate the current action value q_t:

$$q_t = q(s_t, a_t; \omega) \tag{17}$$

The action at the moment is expressed as:

$$a'_{t+1} = \pi(s_{t+1};\theta) \tag{18}$$

The action value at moment $t+1$ is:

$$q'_{t+1} = q(s_{t+1}, a'_{t+1}; \omega) \tag{19}$$

TD-Target is expressed as:

$$f_t = r_t + \gamma \cdot q_{t+1} \tag{20}$$

TD-error is the differences in the evaluation of the same action, expressed as:

$$\delta_t = q_t - f_t \tag{21}$$

The smaller the value, δ_t the more accurate and stable the evaluation of the evaluation network is.

Update the network parameters ω of the value network using the TD gradient descent method:

$$\omega \leftarrow \omega - \alpha \cdot \delta_t \cdot \frac{\partial\, q(s_t, a_t; \omega)}{\partial\, \omega} \tag{22}$$

ISSN: 2664-9640

Using the method of DPG policy gradient ascent to update the network parameters θ of the policy network:

$$\begin{cases} g = \dfrac{\partial\, q\left(s,\pi\left(s;\theta\right);w\right)}{\partial\ \theta} = \dfrac{\partial\ \pi\left(s;\theta\right)}{\partial\ \theta} \cdot \dfrac{\partial\, q\left(s,a;w\right)}{\partial\ a} \\ \theta \leftarrow \theta + \beta \cdot g \end{cases} \tag{23}$$

where β is the learning rate of the policy network.

The update process of network parameter is shown in Figure 3:

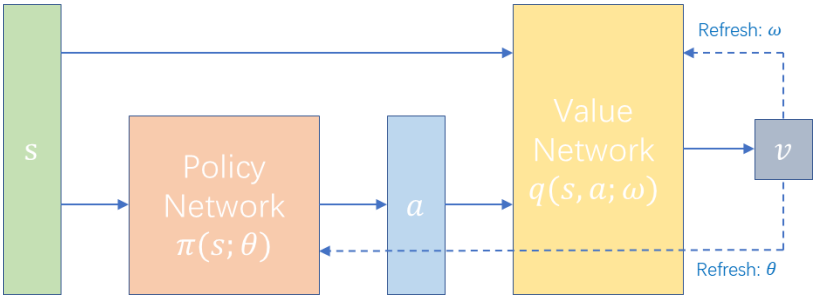

Figure3: Parameter update process

3.2.2. Algorithm Optimization

The TD algorithm adopts a bootstrapping method in updating the policy network and the value network, which can cause the repeated propagation of bias in the network. Therefore, the Target network is used to mitigate this phenomenon. The Target network has the same network structure as the original network. Denote the network parameters of the Target policy network as θ^{-}, and the network parameters of the Target value network as ω^{-}, respectively . Then the calculation formula for q_{t+1} is revised to:

$$\begin{cases} a'_{t+1} = \pi(s_{t+1};\theta^{-}) \\ q_{t+1} = q(s_{t+1},a_{t+1};\omega^{-}) \end{cases} \tag{24}$$

The strategy of regular update policy is adopted for θ^{-} and ω^{-}, and the weighted average of the parameters of the main network and the parameters of the Target network is used as the new Target network parameters, which can mitigate the deviation caused by bootstrapping to a certain extent:

$$\begin{cases} \omega^{-} \leftarrow \tau \cdot \omega + (1-\tau) \cdot \omega^{-} \\ \theta^{-} \leftarrow \tau \cdot \theta + (1-\tau) \cdot \theta^{-} \end{cases} \tag{25}$$

Experience Replay can greatly improve the training speed and effect of DDPG. The experience in each transition can be learned repeatedly, and the utilization of the transition can be improved by establishing an experience pool to save part of the transition. In addition, using experience pooling and non-uniform sampling can break the correlation between transition sequences and improve the learning effect of key states, therefore the final training effect can be improved.

The parameter update process includes Target network and experience pool, which is shown in Figure 4:

ISSN: 2664-9640

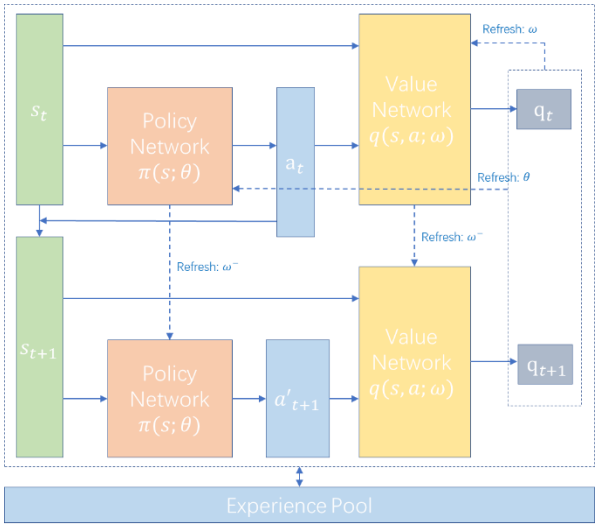

Figure4: The update process containing Target and experience pool parameters

3.2.3. States and Actions

The observation and control process of the power system contains two types of data, i.e., continuous and discrete, so the control problem of the power system can be abstracted into the control problem of the hybrid space. The traditional processing method is to encode discrete variables in the form of one-hot codes and concatenate them with normalized continuous variables. The one-hot code can cause the dimension expansion of the data, which is not conducive to the convergence and stability of the algorithm. This problem can be avoided by using discrete variables continuous.

The set of values of the discrete variable is expressed as:

$$A=\{a_0,a_1,...,a_n\} \tag{26}$$

Define the forward mapping function:

$$f(a)=\frac{a-A_{min}}{A_{\max}-A_{min}}\cdot k+\frac{1-k}{2} \quad a\in A \tag{27}$$

where A_{max} and A_{min} are the maximum and minimum values of the set A, respectively, $k\in(0,1]$ indicating the length of the mapping interval, and $f(a)$ can be assembled into an interval $[\frac{1-k}{2},\frac{1+k}{2}]$ of length k.

The reverse mapping function follows the proximity principle and is constructed as:

$$a=\arg\min_a(|v-f(a)|) \tag{28}$$

where v is the value to be reverse mapped, a is the discrete variable obtained. This means can be used to map continuous variables such as static compensators, some new energy power supplies, and transformer taps back to data containing discrete variables:

The state observed by the agent from the grid environment at time t is denoted as:

$$s_t=\{B_t,L_t,G_t,W_t\} \tag{29}$$

where B_t represents the electrical quantity information of the busbar at the time t: busbar voltage, voltage phase angle, and busbar admittance; L_t denotes the electrical quantity information of the link line at the time t: link line impedance, starting busbar injection power, and ending busbar injection power; G_t denotes Power information, including reactive power supply and new energy equipment; W_t indicates the power information of the equivalent load.

ISSN: 2664-9640

The agent obtains the action a_t to be taken according to the state s_t at moment t by policy function π, including the following information:

$$a_t = \{G'_t, T'_t\} \tag{30}$$

where G'_t represents the controllable part of the generator, T'_t denotes the tap selection of the transformer. For traditional generator sets and new energy power generation equipment, only active power output and reactive power output of the controllable part are generally considered, whereas for reactive power compensation equipment, only its reactive power output is considered.

3.2.4. Reward function

The environment gives corresponding rewards to the different actions of the agent, and gives sufficient punishment to the actions that violate the constraints. According to the analysis results in Section 2.1, a reward function based on minimum active power loss and maximum voltage accuracy is constructed, if the action does not violate the constraints. When an action violates a constraint, a penalty value is generally given, then:

$$r = \begin{cases} \eta \cdot g_v \cdot \dfrac{S_{base}}{g_p} \\ -4 \end{cases} \tag{31}$$

where S_{base} is the reference power of the grid, and η is the adjustment coefficient, which is used to control the value of the reward.

4. Simulation

4.1. Introduction to Calculation Case

The calculation example is based on the IEEE 10-Machine 39-Bus model, with reference voltage 345kV, and the reference power 100MVA. On the basis of this model, the generator G8 in the modified model is a photovoltaic power source PV1 with a maximum active power of 400MW, and a TSC group reactive power compensation device T1 is added for the photovoltaic power source. The device is divided into seven gears, each gear is 10MVar, the total capacity of the device is 70MVar. In addition, a set of reactive power compensation devices T2 of the same specification is additionally supplemented at the bus bar 8 . The modified model topology is shown in Figure 5:

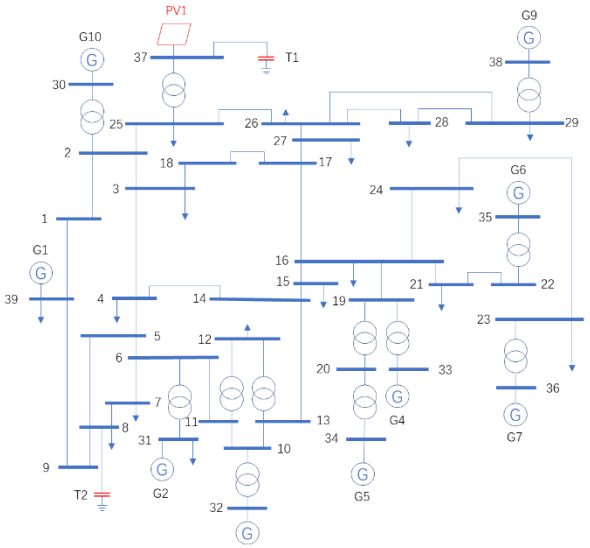

Figure5: Model topology

ISSN: 2664-9640

4.2. Training Parameters and Process

The environment part in the deep reinforcement learning adopts the power flow simulation program based on the secondary development of PyPower, and the Gaussian noise is used to simulate the load change. The agent part uses Pytorch to build the policy network, the value network and the two corresponding Target networks. The hyperparameters of deep reinforcement learning are shown in Table 1:

Table 1: DRL hyperparameters

Parameter Name	Policy Network	Value Network
number of hidden layers	7	5
neurons in each hidden layer	1024	512
learning rate	0.001	0.001
Target network update rate	0.05	0.05
discount factor	0.98	
Reward function adjustment factor	0.05	
epoch_size	5000	
batch_size	2048	
stepperepoch	1000	
experience pool size	100000	
others	Dropout and regularization parameter	

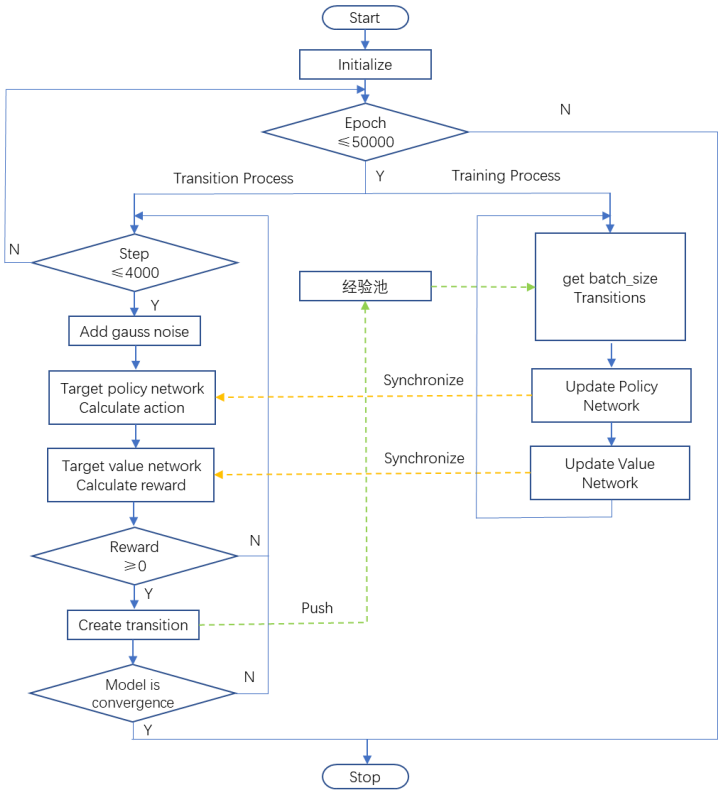

Figure6: Multi-process model training process

ISSN: 2664-9640

In the training process, the Transition in the experience pool is used to train the policy network and the value network, and the Transition generated by the Target network is put into the experience pool, repeat this process until the network converges or when the specified number of rounds of training is completed. Since the coupling between the generation of Transition and the training of the model is loose, the multi-process technology can be used to run the generation and it can be used in different processes to improve the training efficiency of the model. The multi-process model training process is shown in Figure 6.

4.3. Example simulation and result analysis

The above model is preset to 5000 rounds of simulation, the cumulative reward value of each round is shown in Figure 7, and the loss curve of the evaluation network of the model is shown in Figure 8. During the model training process, the cumulative reward value generally shows an upward trend. Episode reward has two relatively obvious fluctuations near 2000 and 3200, and there are also obvious spikes at the corresponding position of the loss curve, indicating that the algorithm has detected a new optimization direction at this time, which resulted in a larger update of the model. After about 4000 rounds of operation, the cumulative reward is basically positive and the loss of the evaluation network is basically stable. At this time, the model is considered to have converged.

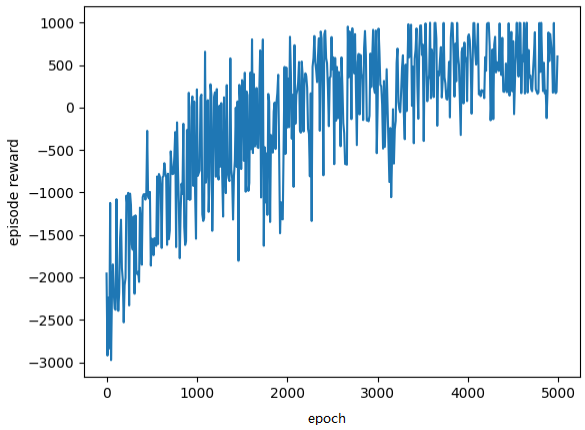

Figure7: Model cumulative reward curve

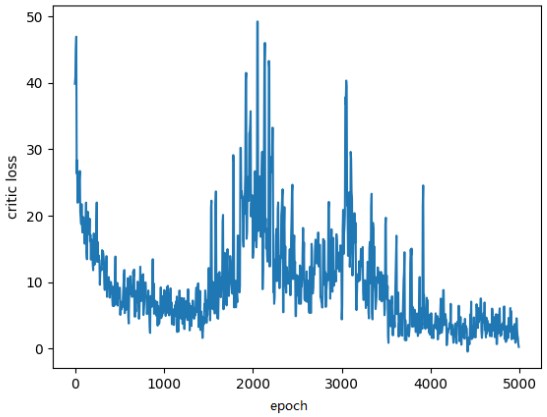

Figure8: Model evaluation network loss curve

In order to verify the effectiveness of the optimization scheme in this paper, the advantages and disadvantages of the optimization scheme in this paper and the reactive power optimization algorithm based on particle swarm are compared in the case of the topological structure of the power grid and the same electrical parameters. Each method extrapolates back 20 steps, each

ISSN: 2664-9640

step uses Gaussian noise to simulate load changes, and compares the active loss inFigure 9 and voltage accuracy in Figure 10.

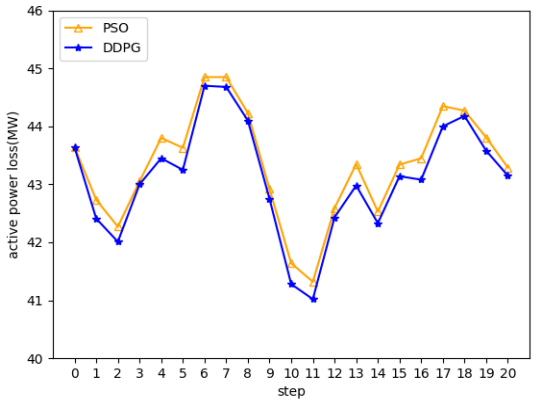

Figure9: Active power loss curve

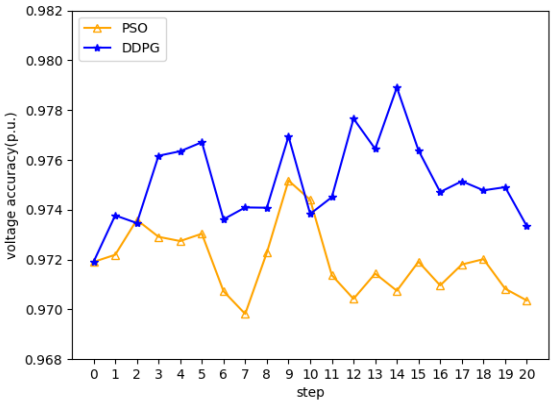

Figure10: Voltage accuracy curve

It can be seen from the figure that under the same simulation starting point, the overall active power loss of the method in this paper is superior to the reactive power optimization algorithm based on particle swarms; the algorithm in this paper has great advantages in the accurate direction of voltage. At the same time, the average duration of the action generated by the algorithm in this paper is 10ms, which is much lower than the 2732ms of the particle swarm algorithm.

5. Conclusion

The scheme designed in this paper solves the grid reactive power and voltage optimization control problem in the hybrid control space. The scheme adopts the deep deterministic policy gradient technology to train the policy network and the value network, and by joining the Target network and the experience playback pool the bootstrapping problem can be mitigate and the experience utilization rate can be improved. The training speed is further accelerated by means of multi-process. Compared with some heuristic algorithms, the scheme in this paper has the characteristics of smaller active power loss and higher voltage accuracy, and has obvious advantages in calculation speed, which has a significant effect on improving the security and reliability of the power grid.

ISSN: 2664-9640

References

[1] WU Zhaoyuan, ZHOU Ming, WANG Jianxiao, et al. Review on Market Mechanism to Enhance the Flexibility of Power System Under the Dual Carbon Target. Proceedings of the CSEE(2022)No.9,p.1-18.

[2] LU Zongxiang, HUANG Han, SHAN Baoguo et al. Morphological Evolution mode land Power Forecasting Prospect of Future Electric Power systems with High Proportion of Renewable Energy. Automation of Electric Power Systems(2017)No.9,p.12-18.

[3] DANG Cunlu,ZHANG Ning,SHAO Chong. Review of Reactive Power Optimization in Power System. Power System and Clean Energy(2014)No.1,p.8-13.

[4] SUN Jian, JIANG Dao-zhuo.A ZBUS POWER FLOW CALCULATION METHOD FOR DISTRIBUTION NETWORK BASED ON NEWTON METHOD.Power System Technology(2004)No.15,p.40-44.

[5] Xu Jianting, Wang Xiuying, Li Xingyuan. SUCCESSIVE QUADRATIC PROGRAMMING METHOD FOR VOLTAGE/REACTIVE POWER OPTIMIZATION IN POWER SYSTEMS. Automation of Electric Power Systems(2001)No.23,p.4-8.

[6] Ma JinTao,L.L.Lai,Yang YiHan. Application of Genetic Algorithms in Reactive Power Optimization. Proceedings of the CSEE(1995)No.5,p.347-353.

[7] LIU Jia , LI Dan , GAO Li-qun , SONG Li-xin.. Vector Evaluated Adaptive Particle Swarm Optimization Algorithm for Multi-objective Reactive Power Optimization. Power System Technology (2004) No.19,p.14-19.

[8] JIA De-xiang ,TANG Guo-qing, HAN Jing .Reactive power optimization of power system based on modified simulated annealing algorithm. RELAY(2004)No.4,p.32-35.

[9] LI Ting, LIU Mingbo. Reduced Reinforcement Learning Method Applied to Multi-objective Coordinated Secondary Voltage Control. Proceedings of the CSEE(2013),No.31,p.130-139.

[10] Diao Haoran, Yang Ming, Chen Fang,Sun Guozhong. Reactive power and voltage optimization control approach of the regional power grid based on reinforcement learning theory. TRANSACTIONS OF CHINA ELECTROTECHNICAL SOCIETY(2015),No.12,p.408-414.

[11] Wang C, Ju P , Lei S, et al . Markov Decision Process-Based Resilience Enhancement for Distribution Systems：An Approximate Dynamic Programming Approach. IEEE Transactions on Smart Grid(2019),No.3,p.2498-2510.

[12] Lillicrap T P , Hunt J J , Pritzel A , et al. Continuous control with deep reinforcement learning[J]. Computer Science, 2015.