Reinforcement Learning Optimized Intelligent Electricity Dispatching System

Xinhang Xiao
Sichuan Tianfu New Area Electric Power Supply Company, State Grid Corporation of China, Chengdu, China
ceongshau@gmail.com

Abstract—With the rapid development of artificial intelligence, new changes are coming to all walks of life including traditional manufacturing industry, bio-pharmaceutical industry, electric-power industry, etc.. For electric-power industry, many machine learning algorithms have been used to achieve intelligent dispatch of electricity, intelligent electricity equipment fault diagnosis and more optimized customer management, among which the automation of power network is one of the most important issue ensuring safe operation of electricity grid. In the recent years, intelligent dispatch method based on big data analysis has been used in intellectual scheduling of electricity and has got significant promotion. However, the intelligent dispatch on big data analysis needs tremendous history data that is sometime not that easy to get. Moreover, once a small probability event happens, the big data based intelligent dispatch method may lose efficacy, thus in this paper we proposed a new intelligent dispatch model based on the reinforcement learning that is more robust, safer and with higher efficiency. Finally, according to the empirical study result, our model outperforms the traditional methods, where the average economic benefit in the test area increases more than 25%, fluctuation of distribution is more stable than before and the carbon emission decreases about 30%.

1. INTRODUCTION
In the recent years, with the development of the energy crisis, safety and stable issues have become a crucial point in the electricity grid. One important and necessary measure to realize power network safe operation is making regional power network dispatching management better. New technologies developed from the artificial intelligence domain offers traditional dispatching management better way to achieve a safer, more stable and more efficient automatic manage system.

The power grid dispatching and control department is a manage center that contains numerous data, rules and experts’ experience. The traditional methods for the dispatching system normally rely on the dispatching experience and manual analysis. However, with the significant increasing of the scale of data and data categories, the relation between data and dispatching goal is becoming much more complicated than before, leading to a much more tedious work for the dispatching managers. Meanwhile, higher requirement for dispatching managers of their official knowledge and experience is needed. Moreover, the traditional experience related dispatching method is inefficient and cannot take the long-term benefits into account. Under this background, the using of artificial intelligent methods to solve electricity dispatching problem becomes a necessary the most obvious thing.

Machine learning algorithm is somehow the core part of artificial intelligence domain that can be used to analysis grid feature, get user portrait, electricity equipment fault diagnosis and intelligent
dispatch for power grid. Reinforcement learning (RL) is an area of machine learning that concern with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Similar with the learning process that our human beings do, reinforcement learning rewards desired behaviors and punishes undesired ones so that the agent can be better and better.

Reinforcement learning is a young research domain that is mainly being studied in the recent ten years, but it developed in a tremendous speed and has been used in many domains such as resources management in computer clustering, traffic light control, robotics, games, web system configuration etc.. In [1], the authors showed how to use RL to automatically learn to allocate and schedule computer resources to waiting jobs, with the objective to minimize the average job slowdown. In paper [2], researcher tried to achieve automatic control of traffic lights more intelligent and solve the congestion problem by using the RL method. In robotics domain, RL is now playing a significant role because of the excellent learning style of RL and the extreme complexity of intelligent robots, as shown in [3][4][5]. The reason that RL is so famous these days can be owed a lot to its success in games such as the well-known AlphaGo by Google [6] and Dota2 by OpenAI [7], where the computer completely defeated the best human beings. Especially the success of AlphaGo, it gave us human beings an important enlightenment that machine can be trained as smart as human and even with better performance. So if we put its training method, the reinforcement learning method, into medical domain, electricity grid domain and those that closely linked with human life, it is obvious that there will be a tremendous progress.

In this paper, we will introduce the using of RL into electricity dispatching. By considering lots of aspects that can influence the usage of electricity, we defined and trained a RL network to learn the most efficient electricity dispatching decision. Then, the empirical study result shows that the using of RL method in power dispatching can significantly improve the power dispatch efficiency and reduce the waste of electricity resources.

2. RELATED WORK
In this section, we will introduce the basic knowledge of RL, including the definition of RL and how it works.

Reinforcement learning is the science of decision making. It is about learning the optimal behavior in an environment to obtain maximum reward. This optimal behavior is learned through interactions with the environment and observations of how it responds, similar to children exploring the world around them and learning the actions that help them achieve a goal.

Reinforcement learning focuses on training an algorithm following the cut-and-try approach, its working process shows in Figure 1 bellow. The agent refers to algorithm that evaluates a current situation (state). Then the agent takes an action, the action will go through the environment and lead to a certain result, after this interaction the environment will output a feedback which is exactly the reward. The feedback can be positive and negative so the reward can be a positive reward or negative reward respectively. When the reward is a positive one, it means that the agent had a right decision in the last step and the environment encourages it to learn in this direction, while the negative reward means the opposite one. Finally, after many rounds of study iterations, the agent can get a maximized reward and get close to reach its best. The working process of RL is just like us human beings, we learn again and again to be the best we can be. In conclusion, agent is an assumed entity which performs actions in an environment to gain some reward; environment (e) is a scenario that an agent has to face; reward (R) is an immediate return given to an agent when he or she performs specific action or task; state (s) refers to the current situation returned by the environment; policy (π) means a strategy which applies by the agent to decide the next action based on the current state; value (V) is expected long-term return with discount, as compared to the short-term reward; value function specifies the value of a state that is the total amount of reward and can be understood as an agent which should be expected beginning from that state.
Reinforcement learning can be implemented into three approaches, value-based, policy-based, and model-based. In a value-based RL method, we try to maximize the value function $V(s)$, in which the agent is expecting a long-term return of the current states under policy $\pi$. In a policy-based RL method, we try to come up with such a policy that the action performed in every state helps to gain maximum reward in the future. While in model-based method, it is needed to create a virtual model for each environment and the agent learns to perform in that specific environment, which is modeled as a tuple $<S,A,P,R,r>$ of Markov Decision Process.

### 3. ALGORITHM

#### 3.1 Markov Decision Process

There are two important algorithms in RL that are the fundamental ones, Markov Decision Process (MDP) and Q-learning. Basic reinforcement is modeled as an MDP. An MDP is a discrete-time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker [8]. Additionally, the MDP is the foundation of Q-learning.

An MDP is a tuple $<S,A,P,R,r>$, where:
- $S$ is a finite set of states,
- $A$ is a finite set of actions,
- $P$ is a state transition probability matrix, and
- $R$ is a reward function,
- $\gamma \in [0, 1]$ is a discount factor.

#### 3.2 Q-learning

The other important learning model in RL is the Q-learning, which is a model-free algorithm with the purpose to learn the value of an action in a particular state. The so-called model-free is because it does not require a model of the environment and can handle problems with stochastic transitions and rewards without requiring adaptations [9]. To introduce Q-learning, we first need to introduce the Q-table. Q-table is a fancy name for a simple lookup table where we calculate the maximum expected future rewards for action at each state, basically, this table could guide people to the best action at each state. In Q-learning, the final purpose is to find out the best Q-table. Given an action $a_t$ and state $s_t$, the Q-value function is:

$$Q^\pi(s_t, a_t) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | s_t, a_t],$$

where $\gamma$ varies from 0 to 1. When $\gamma$ is close to 0, the cumulative reward considers more about immediate benefits, while $\gamma$ approximates 1, the long-term benefit will be thought more. Equation (1) can be then wrote to:

$$Q^\pi(s_t, a_t) = E[R_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) | s_t, a_t]$$

The optimal state-action value function is $Q^*(s,a) = \max_\pi Q^*(s,a)$ and the expansion formula on expectation is:
\[ Q^*(s, a) = \Sigma P(s'|s, a) \left( R(s, a, s') + \gamma \max_{a'} Q^*(s', a') \right) \] (3)

After \( k \) times of iterations we will get the final Q-table showing in equation 4 below. In Q-learning, the core of learning process is looking for the optimal Q-table, where each action and state of Q-learning corresponds to a Q value that is the exact corresponding reward value and all these values make up the Q-table. In other words the Q-table is the map of Q-learning.

\[ Q_{k+1}^*(s, a) = \Sigma P(s'|s, a) \left( R(s, a, s') + \gamma \max_{a'} Q_k^*(s', a') \right) \] (4)

Generally, the Q-table uses the Bellman equation to update \( Q(s, a) \) and get the best one, the process can be summarized as follows:
1. Initialize Q-values arbitrarily for all state-action pairs.
2. For life or until learning is stopped.
3. Choose an action \( a \) in the current world state \( s \) based on current Q-value estimates \( Q(s, \cdot) \).
4. Take the action \( a \) and observe the outcome state \( s' \) and reward \( r \).
5. Update \( Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \).

4. MODEL

In this section we will introduce our model, including its mathematical structure, parameters, input and output.

Our model is based on the Q-learning method. As introduced in the last section we first need to determine what are the states, actions and design the reward. We choose a large residential quarter level electricity dispatching system as example, discretize the dispatching process that is originally a continuous process. Energy storage in distribution network, load rate of electricity, time and battery capacity are simulated as the states of the system; charge, discharge, electric energy conversion and the increase and decrease of dispatching are set as the actions. Once one of the actions \( a_t \) happens, the related reward will be transmitted from the environment to the formal state \( s_t \) that will lead the state update to \( s_{t+1} \).

As we have time as one of the system state, we can discretize the dispatching process into 24 parts, of which one hour is a segment. Assume the energy storage in distribution network, load rate of electricity and battery capacity have \( L, M, N \) segments respectively, then we have \( lmn \times 24 \) states, which means that our state set \( S \) is \( S = \{s_1, s_2, ..., s_{lmn \times 24} \} \).

After setting the set of states in our model, we need to determine action set, where each action \( a_t \) has a interaction with the environment and will determine the state’s transformation \( s_t \rightarrow s_{t+1} \). The design of actions is shown in the table bellow:

| TABLE 1 POLICY SETTING |
|------------------------|
| Policy                 | Value a |
|------------------------|---------|
| Charge by thermal power generation | +1 |
| Charge by clean energy (wind, sun) | +2 |
| Energy storage system→distribution | +1 |
| Distribution→energy storage system | -1 |
| Distribution network→ordinary use | -1 |
| Distribution network→commercial use | -2 |
| Leave unused | -1 |

From the table we can see that we give positive feedback to those policies that increase the amount of distribution network and negative feedback to actions that output electricity to users. When electricity is generated by clean energy like wind, sun light and hydropower, we give larger action feedbacks in order to encourage the use of clean energy.
Once the states and actions are determined, we begin the training of our model, based on the Q-learning method, we continuously update the Q-table after every iteration and finally the Q values can infinitely approaches convergence after thousands of training episodes. The training process is showing bellow:

1. Initialize environment.
2. Initialize the original Q-table according to the current environment.
3. Based on the environment including number of residents, season, current electricity prize, transmission and distribution cost and many other variables to calculate the final reward and economic benefits. Then the reward or economic benefits will be put into the R-table which refers to the reward that the certain action caused.
4. Under the current state $S_t$, for example current time, storage in distribution network, determine which action can be chose.
5. Using Q-learning algorithm, combined with the greedy method, choose the corresponding policy and then update Q-table.
6. If Q- I s convergent, end the iteration and get the final Q-table; if not, then continue step 3.

Pseudo-code of the training process of our model is showing bellow:

```
Algorithm 1
Algorithm parameters: $\alpha \in (0, 1]$, small $\epsilon > 0$
Initialize $Q(s, a)$, for all $s \in S, a \in A$
1: for each episode do
2: Initialize $S$
3: for each step of episode do
4: Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
5: Take action $a$, observe $r, s'$
6: end for
7: $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
8: $s \leftarrow s'$
9: Until $S$ is terminal
10: end for
```

5. **EMPIRICAL STUDY**

In the simulation experiment, we choose a distribution network with the capacity of 5000 kWh, we set energy storage in distribution network, load rate of electricity and battery capacity have 5 segments, which means that we have $5^3 \times 24 = 3000$ actions. Policies for each action obeys the policy setting in the last section. We set iteration upper limit as $8 \times 10^4$.

![Figure 2. Total economic benefit](image)

In Figure 2 and Figure 3, the green line is the variation trend of economic benefit using the reinforcement learning based model.
for dispatching electricity, while the blue line shows the trend of original experience related methods. It is obvious that the RL method based model can significantly improve the economic benefit for the dispatching system.

Figure 4 and Figure 5 shows the fluctuation of the distribution network before and after optimization by our RL based model. From the two figure we can see that the fluctuation of the distribution network is much more stable than before.
Figure 6. Carbon emission index before and after optimization

Finally, Figure 6 bellow shows the carbon emission after 200 episodes of training. The blue line is the carbon emission per day in the test area variation for original method and the green one shows the variation after optimization, in which we can see that our model has a significant improvement with an improvement of about 30% in average.

6. CONCLUSION
With the rapid development of artificial intelligence, many astonishing changes have appeared around our usual lives. Even in some traditional industry areas that have developed for years, new artificial intelligence measures are making them work more sensible, reasonable and much more efficiency with higher economic benefits. Reinforcement learning is a new research branch of artificial intelligence that has been widely used in automatic control domain. In this paper, we discuss the using of reinforcement learning into electricity dispatching system that is relevant to the society and our usual lives. From the empirical study we can see that our reinforcement learning based dispatching method can rational allocate resources, allocate power generation tasks reasonably and improve the economic benefit for power grid, etc. In the next step for further study, we will do more research including design more complicated training networks with deep reinforcement learning methods and use larger electrical network model to simulate our training environment, in order to improve the efficiency of power grid dispatching system further more.

REFERENCES
[1] Mao, Hongzi and Alizadeh, Mohammad and Menache, Ishai and Kandula, Srikanth, “Resource management with deep reinforcement learning,” P Proceedings of the 15th ACM workshop on hot topics in networks, pp. 50–56, 2016.
[2] Arel, Itamar and Liu, Cong and Urbanik, Tom and Kohls, Airton G, “Reinforcement learning-based multi-agent system for network traffic signal control,” IET Intelligent Transport Systems, vol. 4, pp. 128–135, 2010.
[3] Lin, Long-Ji, “T Reinforcement learning for robots using neural networks” 1992
[4] Mahadevan, Sridhar and Connell, Jonathan, “T Automatic programming of behavior-based robots using reinforcement learning,” J Artificial intelligence, vol. 55, pp. 311-365, 1992.
[5] Mahmood, A Rupam and Korenkevych, Dmytro and Vasan, Gautham and Ma, William and Bergstra, James, “Benchmarking reinforcement learning algorithms on real-world robots,” Conference on robot learning, pp. 561–591, 2018.
[6] Silver, David and Schrittwieser, Julian and Simonyan, Karen and Antonoglou, Ioannis and Huang, Aja and Guez, Arthur and Hubert, Thomas and Baker, Lucas and Lai, Matthew and Bolton,
Adrian, et al, “Mastering the game of go without human knowledge,” nature, vol. 550, pp. 354–359, 2017.

[7] Berner, Christopher, et al, “Dota 2 with large scale deep reinforcement learning,” arXiv preprint arXiv:1912.06680, 2019.

[8] Song, Haili and Liu, C-C and Lawarr, Jacques and Dahlgren, Robert W, “Optimal electricity supply bidding by Markov decision process,” IEEE transactions on power systems, vol 15, pp. 618–624, 2000.

[9] Watkins, Christopher JCH and Dayan, Peter, “Q-learning,” Machine learning, vol 8 (3-4), pp. 279–292, 1992.

[10] Song, Haili and Liu, C-C and Lawarr\text{e}e, Jacques and Dahlgren, Robert W, “Optimal electricity supply bidding by Markov decision process,” IEEE transactions on power systems, vol 15, pp. 618–624, 2000.