The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

Smart Grid Optimization by Deep Reinforcement Learning over Discrete and Continuous Action Space

Tomah Sogabe[1,2,3], Dinesh Bahadur Malla[2], Shota Takayama[2], Katsuyoshi Sakamoto[2], Koichi Yamaguchi[2], Thakur Praveen Singh[3], Masaru Sogabe[3]

[1]Info-Powered Energy System Research Center, [2]Department of Engineering Science,
[1,2]The University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan
[3]Technology Solution Group, Grid Inc., Kita Aoyama, Minato-ku, Tokyo, 107-0061, Japan

Abstract — Energy optimization in smart grid has gradually shifted to agent-based machine learning method represented by the state of art deep learning and deep reinforcement learning. Especially deep neural network based reinforcement learning methods are emerging and gain popularity to for smart grid application. In this work, we have applied the applied two deep reinforcement learning algorithms designed for both discrete and continuous action space. These algorithms were well embedded in a rigorous physical model using Simscape Power Systems™ (Matlab/Simulink™ Environment) for smart grid optimization. The results showed that the agent successfully captured the energy demand and supply feature in the training data and learnt to choose behavior leading to maximize its reward.

1 Introduction

Energy grid system containing renewable energy resources(RES) such as photovoltaic energy, wind power as well as hydropower have been considered as alternative power supply configuration. It is renovating conventional grid systems, aiming at reducing the emission of CO_2 while mitigating the global warming. A decentralized energy system is more robust and resilient against the unexpected natural disasters, which are frequently occur in countries such as Japan. However, due to the intermittent nature of RES, a mismatch between electricity supply and demand is often encountered and causes instability and limit of power output. As an effective approach to these challenges, smart grid has been proposed and has shown great technological innovation towards intelligent, robust and functional power grid [1][2].

Smart grid evolves energy transmission among different sub-smart grid utilities, which finally contribute to the efficient energy management ecosystem of energy storage, energy supply, balanced load demand over large scale grid configuration. Construction of efficient smart grid system is in principle a control optimization mathematical problem. A wide range of methods have been proposed to tackle this challenge including linear and dynamic programming as well as heuristic methods such as PSO, GA, game or fuzzy theory and so on[3]. In the recent years, studies on energy optimization in smart grid has gradually shifted to agent-based machine learning method represented by state of art deep learning and deep reinforcement learning. Especially deep neural network based reinforcement learning methods are emerging and gain popularity to for smart grid application[4][5].

In this work, we focus on the following issues and tasks:

(1) Different from previous reports, we have developed our deep reinforcement learning algorithm embedded in a rigorous physical model using Simscape Power Systems™ for smart power grid optimization. All the parameters used in smart grid represents the realistic electric circuits and detailed fluctuation regarding the voltage, frequency and phase can be therefore fully revealed, which are not available in previous reports where the constructed smart grid system could not output sufficient information.

(2) For RL, model-free off-policy deep Q-learning suing Matlab™ is developed. DQN is suited for addressing continuous state space and discrete action space. Here we have focused on the discrete action control designed for switching the grid power supply/sell and battery charge /discharge.

(3) For continuous state and continuous action space, we

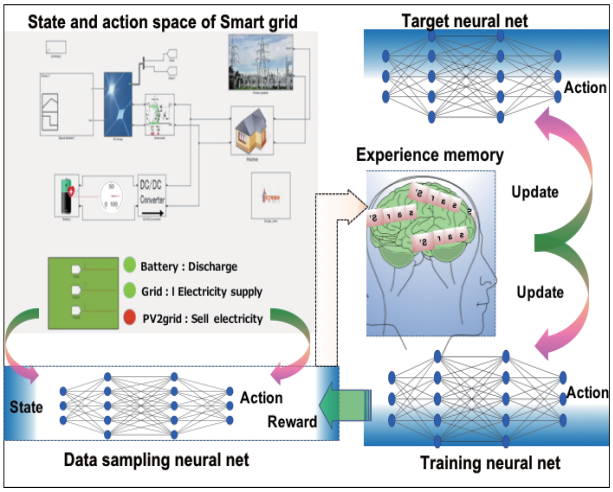

Fig.1. Sketch of smart grid optimization using deep neural network based reinforcement learning algorithm.

Contact: Tomah Sogabe, i-PERC, The University of Electro-communications, sogabe@uec.ac.jp

have self-developed a Gibbs deep policy gradient algorithm, in which we have hybridized the latest deep deterministic policy gradient with the deep actor-critic stochastic policy gradient.

2. ALGORITHM AND MODEL

(i) **Deep Q-Learning (DQN):** A general model which describes the main framework is given as follows. In this

Algorithm: Model-free off-policy Deep Q-learning with Experience Replay

Load the data from data directory for Matlab® simulation
Open Simulink® model
Initialize Simulink® model with initial parameter {first action}
Initialize replay memory D to capacity N
Initialize weight with random weights
Initialize action with zeros (in this simulation 7 actions)
for episode = 1:M do
  Initialise state s = {*starting Value*}
  for t = 1: T do
    set starting and end simulation time for Simulink®
    set update parameter
    start simulation for data which are output of simulation
    simulation out = [Photovoltaic power generation,
                      Battery State of charge(SOC)…
                      Load consumption,
                      DC power generator]
    State= simulation out
    forward state to Deep Q-learning network
    with probability select a random action a_t otherwise select $a_t = max_a Q^*(\varphi(s_t), a; \vartheta)$
    Deep Q-learning network return action index
    Where
    1 = switch from Photovoltaic array to DC power generator
    2 = switch from Photovoltaic array to Load
    3 = switch from DC power generator to Load
    4 = switch for battery charge
    5 = switch from Photovoltaic array to DC power generator
        & switch from Photovoltaic array to Load
    6 = switch from Photovoltaic array to DC power generator
        & switch for battery charge
    7 = switch from DC power generator to Load
        & switch for battery charge
    set action in the Simulink model
    1 = "ON"
    0 = "OFF"
    simulate Simulink Model
    get simulation data from model
    set s_{t+1} = {Next *state*} from getting simulation data from model
    execute action a_t in emulator and observe reward r_t
    store transition (s_t, a_t, r_t, s_{t+1}) in D
    set $s_t = s_{t+1}$
    until s_{t+1} is terminal
    sample random minibatch of transitions (s_t, a_t, r_t, s_{t+1}) from D

$$y_j = \begin{cases} r_j & for\ -\ and\ +\ reward \\ r_j + \gamma max_a Q(s_{j+1}, a^0; \vartheta) & for\ 0\ \ reward \end{cases}$$

    Perform a gradient descent step on $(y_j - Q(\varphi(s_t), a; \vartheta))^2$ for weight upda
  end for
end for

sketch, we adopted deep Q-learning algorithm as an example to illustrate the learning principle: physical model of smart grid simulation environment based on Simscape Power Systems™ was constructed. The state space is always continuous and action space is set either discrete or continuous for off-policy Q learning and deep policy gradient algorithm respectively. A detailed operation flow is given as follows by the form of pseudo-simulation code:

(ii) **Gibbs Deep Policy Gradient (GDPG)**: Deterministic policy is in theory efficient at the late stage of simulation because the policy distribution is less variant and more deterministic. Policy gradient is usually formulated as follows, where η is the policy object function; θ is the function approximation parameter (in neural network, it is the weight w); s and a correspond to the state and action $Q_\pi(s,a)$ is the state-action function under certain policy $\pi(a|s,\theta_p)$ and is the :

$$\nabla\eta(\theta_p) = E\left[Q_\pi(s,a)\,\nabla_{\theta_p} log\ \pi(a|s,\theta_p)\right]$$

policy distribution function. David et al has shown that if the policy is treated as deterministic, the above equation can be reformed as: [6]

$$Q_\pi(s,a)\,\nabla_{\theta_p} log\ \pi\left(a\middle|s,\theta_p\right) = \nabla_a Q(s,a) \qquad (2)$$

and if the action a is approximated as policy action function:

$$a = u(s|\theta^\mu) \qquad (3)$$

using the chain rule $\nabla_a(s,a)$ can be further extended as:

$$\nabla_{\theta^a} Q(s|\theta^a)\nabla_{\theta^\mu} u(s|\theta^\mu) \qquad (4)$$

and then policy parameter θ^μ is updated as the usual gradient decent:

$$\theta^\mu = \theta^\mu + \alpha \cdot \nabla_{\theta^a} Q(s|\theta^a)\nabla_{\theta^\mu} u(s|\theta^\mu) \qquad (5)$$

However, implementing the deterministic policy at the early simulation stage will inevitably cause high variance and slow convergence because the policy is far from optimal policy so the policy distribution is fairly stochastic and less deterministic with high bias. The hybridized algorithm is designed in such a way that both the advantage of deterministic and stochastic policy is assimilated thus a stable learning profile with fast convergence can be achieved.

(iii) **Neural Network Model**: In this work, we use multilayer neural network including four hidden layers to approximate the

$$\Delta\theta^\mu = \begin{cases} \alpha \cdot \nabla_{\theta^a} Q(s|\theta^a)\nabla_{\theta^\mu} u(s|\theta^\mu), & Step_t \\ \alpha \cdot \{r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\,\nabla_{\theta^\mu} u(s|\theta^\mu), & Step_{t+1} \end{cases}$$

state-action value function. The activation function is fixed at hyperbolic-tangent function and epsilon-greedy algorithm is utilized to enhance the exploration in the case of DQN for discrete action and re-parameterization. These techniques were used when using GDPG for continuous action space.

3. RESULTS

Here we present one representative simulated results by employing DQN algorithm to optimize four discrete action controls: (1) Grid on/Battery off and (2) Grid

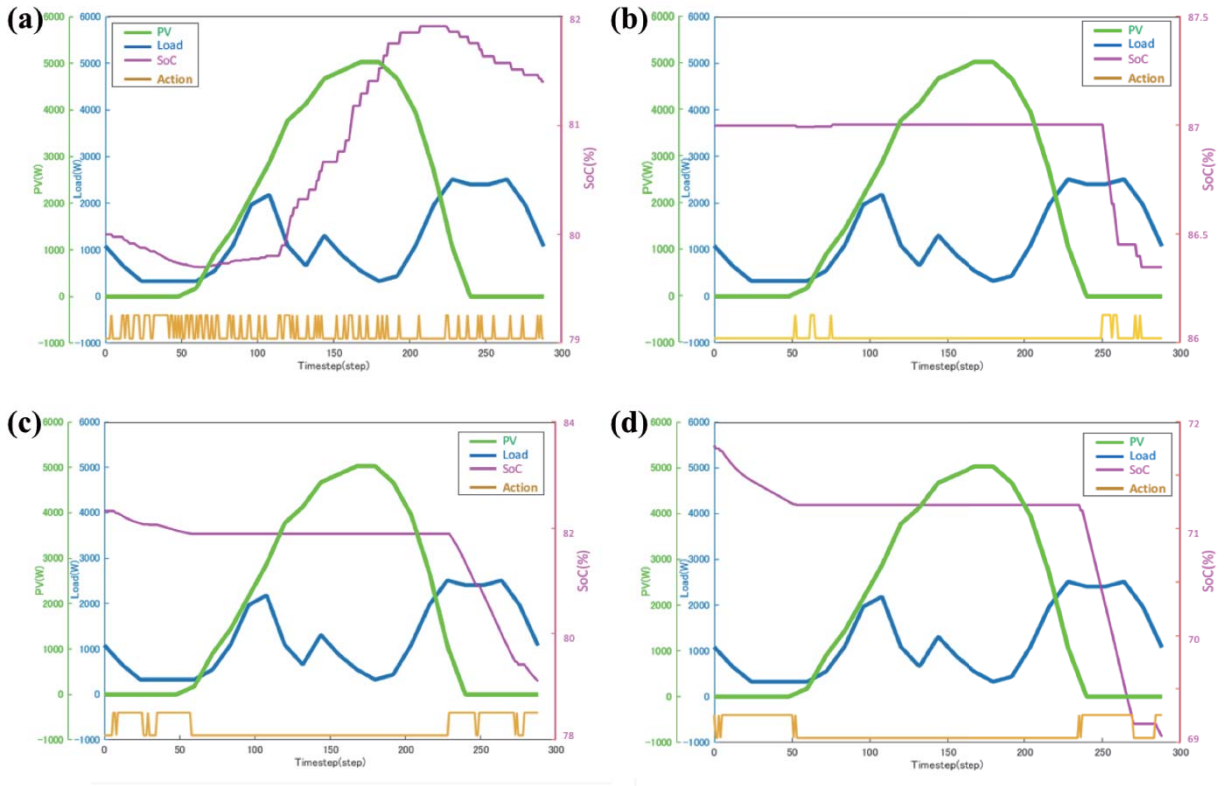

Fig.2. Agent training results using the DQN algorithm. (a) Early training stage; (b) Middle training stage; (c) and (d) latter training stage.

off/Battery on. Action pair (1) was designed to deal with situation when the PV power exceeds the load demand and is able to get the reward by selling the electricity to Grid. On the other hand, Action pair (2) was designed for agent to learn to charge the battery during daytime for the discharge when the PV power is off during night time. The load demand can be supplied through either the grid or the battery and thus the agent has to learn how to control these actions to maximize the reward designed in advance. As shown in Fig.2 the action profile varied during learning process under the preset reward function. At the early stage, the agent showed random actions, as shown in Fig.2(a), where a frequent switch on/off was observed during training. In the middle of training, the agent gradually grasped the inherent feature between demand and supply and learnt how to choose stable action to maximize its reward. This can be seen in Fig.2(b) where constant on or off period were extended and the action profile experiences less noise than Fig.2(a). The agent made a decision to sell its surplus PV power to grid during the day time when its SoC remains high. In the latter training stage as shown in Fig.2(c) and (d), the agent successfully learnt to discharge its battery power during night instead buying electricity from the grid. More detailed results regarding the comparison between discrete action and continuous action will be presented at the conference. Deep insights related how to design various reward functions to train the agent for the different targets posed by smart grid energy system would also be presented.

4. CONCLUSION

We present here a deep reinforcement learning method applied for smart grid optimization. From the preliminary simulation results, the agent was able to catch the feature involved in the balance of load demand, PV power surplus and battery discharge/charge as well as grid integrate. The agent successfully learnt how to tune its action profile to maximize the reward function during training. More detailed results regarding to the comparison between DQN and GDPG and the key role played by reward function will be given at the conference.

REFERENCES

[1] R. H. Khan and J. Y. Khan, "A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network," *Computer Networks*, vol. 57, no. 3, pp. 825-845, 2013.

[2] H. E. Brown, S. Suryanarayanan, and G. T. Heydt, "Some characteristics of emerging distribution systems considering the smart grid initiative," *The Electricity Journal*, vol. 23,no. 5, pp. 64 -75, 2010.

[3] M. R. Alam, M. St-Hilaire, and T. Kunz, "Computational methods for residential energy cost optimization in smart grids: A survey," *ACM Comput. Surv.*, vol. 49, pp. 22-34,Apr. 2016.

[4] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," *Sustainable Energy, Grids and Networks*, vol. 6, pp. 91-99,

2016.

[5] V François-Lavet, Q Gemine, D Ernst, R Fonteneau, "Towards the minimization of the levelized energy costs of microgrids using both long-term and short-term storage devices, "*Smart Grid: Networking, Data Management, and Business Models*, P295-319,2016

[6] S. David, L. Guy, H. Nicolas, D. Thomas, W. Daan, and R.Martin. "Deterministic policy gradient algorithms," *ICML*, 2014