Deep Reinforcement Learning for Power System Service Restoration

Team members: Felipe Buchbinder, Shota Takeshima, Young Kyung Kim

Plan

- Unmet Need: Improving model performance compared to original paper.
- Hypothesis/Goal/Aim: Our goal is to improve model. We have ~~few~~ hypothesis. We believe through using more advanced techniques as auto-encoder and recurrent neural network we would able to improve original model. We also believe that through incorporating volatility into reward function will improve original model.
- Objectives with (short, one-sentence) justification: We aim to use reinforcement learning to build a model that can assist actors managing a hot or cold water system to maintain the stability of power generation and distribution, especially for efficiency in cost. Through building the model we are expected to write paper and submit to the conference.
- Audience: researchers
- Style: Academic paper(planning to submit the paper to conference)

Abstract

to be added, since we don't have results

1 Introduction

1.1 Motivation and importance of the problem

Electrical power grids are an essential, yet extremely complicated infrastructure of city life. The task of maintaining or restoring the stability of power generation and distribution (in the event of a targeted attack or unintentional failure) is therefore a very important one. Many methods have been developed to address this problem, all of which required manual planning or interruptions. However, there are limitations to manual planning, as humans have limits on processing the huge amount of information needed to model such large and complex systems as the power grids of modern cities. Thus, scholars have started to look into reinforcement learning models that would identify the patterns of the environment to make a optimal decisions for maintaining or restoring the stability of power generation. Considering the development of various different techniques in reinforcement learning, there is room for more improvement in building a better model that would make optimal decisions. This paper makes a contribution to extant literature by providing one such model.

2 Literature Review

2.1 Literature in Energy

- Literature in Power grids and issues in assuring stability of generation and distribution. Smart cities, and a view of the future.
- City Learn (simulation environment)
- Fusing TensorFlow with building energy simulation for intelligent energy management in smart

cities
- How energy price ~~changes~~ from changes in demand or supply.

2.2 Literature in Reinforcement Learning

- Reinforcement learning for demand response: A review of algorithms and modeling techniques
- How different field used reinforcement learning to insight on application wise. For example, finance used used moving variance to capture volatility of a stock which was used as a reward for reinforcement learning.
- Different reinforcement learning techniques as soft-actor critiques or multi-task learning.

3 Experimental results

3.1 Data

3.1.1 Simulator

- Describe the environment(especially about what kinds of data are generated from the environment)

CityLearn is an OpenAI Gym environment for the easy implementation of RL agents in a multi-agent demand response setting to reshape the aggregated curve of electrical demand by controlling the energy storage of a diverse set of buildings [1]. Its main objective is to facilitate and standardize the evaluation of RL agents such that it enables benchmarking of different algorithms. CityLearn includes energy models of air-to-water heat pumps, electric heaters, and the pre-computed energy loads of the buildings, which include space cooling, dehumidification, appliances, domestic hot water (DHW), and solar generation. We try to implement a new reinforcement algorithm (RL) into CityLearn to control the storage of DHW and chilled water. The RL agents send their control actions hourly and receive a set of states and rewards in return. Indoor temperatures in the buildings do not change, as the environment automatically constraints the actions of the controllers and ensures that the energy supply devices are large enough to guarantee that the energy demand of the buildings is always supplied. The RL agents can decide how much cooling or heating energy store or release at any given time. A backup controller integrated in CityLearn guarantees that the energy supply devices prioritize satisfying the energy demand of the building before storing any additional energy. The simulator allows us to configure the building properties and their number to be controlled by the agent(s). In addition, it also provides four climate zone setting to evaluate if the agent can works generally in several environments. As Figure 1 shows, the climate properties, such as temperature inside/outside the buildings differ by the climate zone.

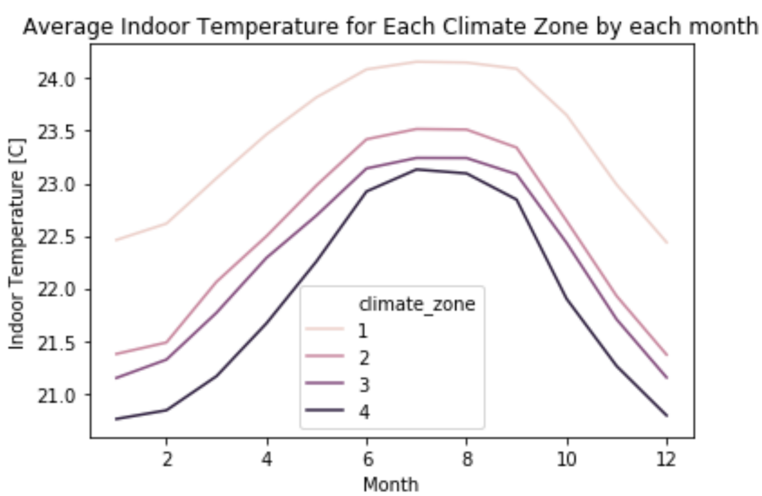

Figure 1: Average indoor temperature setting by climate zone

3.1.2 State, reward, and action for reinforcement learning agent

- Describe about different parameters for the environment(for example explain about region parameter)
- Describe who is the agent. Describe what are the possible actions. Describe how is reward calculated.
- Detail number of simulations (and parameters) actually employed. Brief descriptive statistics

City Learn simulates an environment where a set of N buildings are, at any given time t, in a state that is completely defined by 28 variables. These variables are of the following types:

- Temporal variables as "month", "day", or "hour" • Temperature variables
- Humidity variables
- Diffuse solar radiation variables
- Storage variables(How much energy is stored)

We find it important to highlight that all but the last type of variables are related to the demand for energy. Indeed, the demand for energy varies through time, with temperature, humidity, or with the speed with which energy dissipates. The last kind of variable, however, refers to storage. And storage is ultimately a "managerial" decision: it's how the agent chooses to manage the network. Such choice has implications that deserve our attention: If a building stocks energy today, it is betting that energy today will be cheaper than in the future. Moreover, if it chooses to use the energy it has previously stocked, it is limited by the choice of how much it decided to stock in the past. Hence, the storage variables insert a path dependency on our problem. As for the actions, there are only 2 actions possible for the agent to take - Increasing or decreasing the energy in cooling storage - Increasing or decreasing the energy in domestic hot water storage since each building has its own storage, such 2 actions are, in reality, 2 actions per building. In other words, the agent must decide how much energy will de put in storage (or taken out of storage) for each building. Such a central agent may be a single agent, managing the entire network, or a set of agents, one responsible for each building, which acts independently but can communicate with each other. We have considered a set of one agent per building acting independently. For simplicity, however, we often refer to these agents in a singular form.

3.2 Methods

Description on what part of pipeline improved compared to original paper
- PCA into Auto-encoder
- Gradient boosting decision tree into Recurrent Neural Network
- Improved reward functions through incorporating whether demand curve is flattened or not.

Evaluation of models
– consider different baselines, which will from original paper that has 4 different agents that are trained with different reward function.
- $r_i^1 = min(0, e_i)$
- $r_i^2 = sign(e_i) \times min(0, e_i)^2$
- $r_i^1 = min(0, e_i)^3$
- $r_i^1 = -sign(e_i) \times e_i^2 \times min(0, \Sigma e_i)$

- compare through evaluating these 5 objectives below.
Peak demand (for the entire simulated period)
- Average daily peak demand (daily peak demand of the district averaged over a year)
- Ramping
- 1 - Load factor (which will tend to 0 as the load factor approaches 1)
- Net electricity consumption

3.3 Evaluation of Results

- Compare the results of all models.
- First we are going to separately change one by one. For example check whether auto-correlation will improve separately.
- We are going examine how this change impacts one of the objective. We are going to do this for three methods we have mentioned
- For final result we are going to combine three method we have discussed to see how those three

changed have impact together.
- We also expect some trade-offs from some of the method we add and we are going to probably talk about in this section. Discuss about how to handle the trade-offs if there is any.

4 Conclusion/Discussion

We will mention about the limitation of our model and talk about what kind of different methods we could possibly consider in the future.
We will also mention about limitation in environment.It would be great if we could find more environment to make our model more generable.

5 Figures

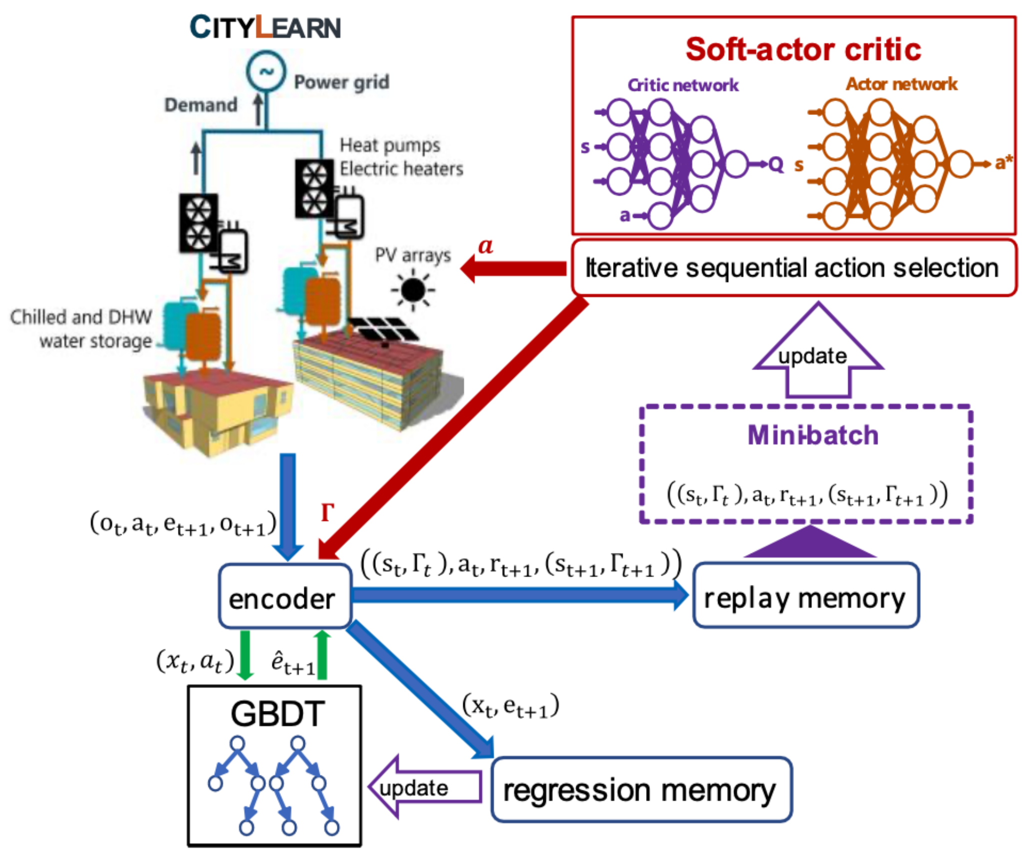

Figure 2 Simulation framework: integration of MARLISA into CityLearn

Figure 2: Going to highlight part of pipelines that we are going to change to improve

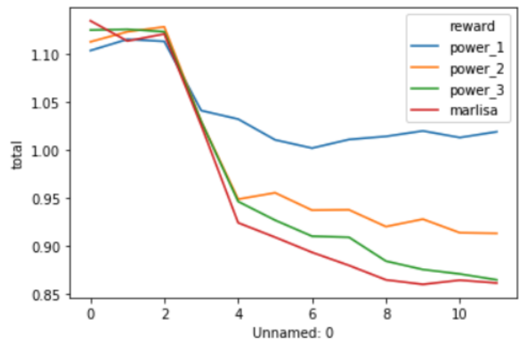

Figure 3: We are going to have other lines that we are going to try to improve. Currently those 4 lines will be a baseline for us

References

[1] José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. Citylearn v1.0: An openai gym environment for demand response with deep reinforcement learning. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '19, page 356–357, New York, NY, USA, 2019. Association for Computing Machinery.