Effective House Energy Management Using Reinforcement Learning Technical Documentation

Dawid Czarneta, Jakub Frąckiewicz, Filip Olszewski, Michał Popiel, Julia Szulc

June 3, 2018

Abstract

This is a technical documentation of our university project developed with cooperation and guidance of Rafał Pilarczyk from SAMSUNG. Document contains a short introduction and overview of main concepts, milestones, goals and effects of this project. It is followed by a technical part, that contains a setup guide, code and tests overviews, explanation of used algorithms and concepts (including the Reinforcement Learning part) and details about the development process, providing useful insights, experiences and lessons we have learned the hard way.

1 Introduction

1.1 Goals and milestones

1.2 Results

2 Reinforcement Learning

2.1 What have we used?

2.2 PyTorch Framework

3 Getting Started

3.1 Setup and Requirements

This project was made on Debian based Linux distributions. We strongly recommend it in similar environment.

Firstly, install Python 3.6.5. You can do it via apt package manager from command line

```
$ sudo apt–get install python3.6
```

or by downloading it from producent's page [1] and performing manual installation.

Secondly, go to main project directory and install required Python libraries:

```
$ pip3 install –r requirements.txt
```

From now on, you should be able to use our project by running different source scripts.

[1] `https://www.python.org/downloads/release/python-365/`

3.2 Usage: Learning and Simulation

Our project can be run in different modes. These can be configured by changing contents of **configuration.json** file, located in main project directory. This directory will be referred to as **TOP** in this section.

3.2.1 Learning mode

This mode can be run by going into source files directory and launching **main.py** with python3 interpreter.

```
$ cd TOP/src
$ python3 main.py
```

In learning mode, Reinforcement Learning Agent model is learning for a given number of episodes. One episode is typically one day in randomly generated environment. Agent is performing action every few minutes, trying to get perfect balance between required conditions (light, temperature) and energy cost; he is also drawing conslusions from his mistakes. As the time passes, Agent is getting better; we can roughly say that in the long term, a longer trained Agent will perform better than the one after shorter training session, with the same training configuration.

After training session, model is typically saved in

```
$ TOP/src/saved_models/
```

directory, within a separate folder with new index (indexes start at 0). Model is saved along with its configuration file, graph presenting learning curve and log with rewards from every step.

Existing agent model can also be loaded into training session. In this case, you will be prompted to insert existing model's index number.

Basic configuration for this mode can be modified by changing contents of configuration.json, a part called **main**. Precise documentation of the whole configuration file can be found later in documentation.

NOTE: Altough basic configuration for main (learning) mode is fairly limited, succesfull learning is built upon all environment / agent settings in configuration file. Default ones are based on our research in order to provide the best and most meaningful learning results.

3.2.2 Simulation mode

This mode can be run by going into source files directory and launching **simulation.py** with python3 interpreter.

```
$ cd TOP/src
$ python3 simulation.py
```

After launching, you will be prompted to insert existing model's number to use in simulation.

NOTE: Simulation will start in full screen, be sure to have only one monitor connected - otherwise, simulation will span across all monitors.

In simulation mode, user is presented to graphical interface representing Reinforcement Learning Agent in action. There is a continuous environment, inside which Agent is trying to make his best decisions. No learning process included.

Key mapping scheme:

key	action
ESC	exit simulation
SPACEBAR	toggle pause / play
-	slow down simulation
=	speed up simulation
z	toggle charts zoom

Widgets

- Weather widget

  Located on the left side of the screen. This widget presents current time and weather informations outside of the house in form of a timer animation and five weather indicators:

  - temperature
  - sun intensity
  - wind intensity
  - clouds intensity
  - rain intensity

- Devices widget

  Located on the right bottom of the screen. This widget presents current house devices settings. Those include:

  - Energy source (icons) and battery level (percentage)
  - Cooling, heating, light and curtains level (leveled bar chart)

- Charts widget

  Located on the right center of the screen. This widget presents last 100 levels of light / degrees of temperature inside the house, along with users required level at given time.

  These two charts can be zoomed by pressing "z" on the keyboard. Zooming centers chart in a way that required level is placed directly at half of the charts height and the visible range is between $-0.1 * max$ and $+0.1 * max$ for maximum light / temperature levels respectively.

- Gauges widget

  Located on the right top of the screen. This widget presents current and desired level of light / temperature in form of a gauge chart.

Default speed of simulation can be set in **configuration.json** file, in module "simulation", item "fps".

3.2.3 Manual testing mode

This mode can be run by going into source files directory and launching **manual_test.py** with python3 interpreter.

```
$ cd TOP/src
$ python3 manual_test.py
```

This mode is a command-line interface for manual testing of Reinforcement Learning Agent behaviour. User can manually test agent-environment connection by deciding which action to make, or can load existing model to be able to use its choice in particular moment. The whole process is done one step at the time.

All actions are made by entering a number specific for a certain command. Commands and numbers are visible all the time on top of the rendered text.

NOTE: This mode was made mainly for testing purposes, and should be treated as additional, not main functionality.

3.3 Testing

4 Code overview

4.1 HouseEnergyEnvironment module

4.2 Agent module

4.3 Configuration file

5 Accuracy measures and tempo of learning

6 Development Process

6.1 Chronology

6.2 What Failed

7 Conclusions

8 Bibliography and Useful Sources