

# DeepedGlove

Deep Learning Based Energy Optimization of Glove Manufacturing Plants ( $Q_2/W_2$ ).

## 1 Introduction

In this project we develop data-driven AI system called DeepedGlove to optimize the energy efficiency of glove manufacturing plant.

The real world manufacturing systems are very complex. It can cause great damage to lives or cost if something goes out. Due to that reason the model is in offline setting. Where the agent isn't allowed have interactions with real system or perfect simulation environment (The SCADA system provides the collected data).

DeepedGlove consists of

- *Glove Manufacturing Plant Simulator*: Facilate RL training.
- *Model Based Offline RL Framework (MORE)*: Learn policy under safety constraints.

## 2 Operation Mechanism of Manufacturing Plant

The manufacturing plant is highly complex system consists of lot of sensors. The general glove manufacturing process has 3 main stages.

1. Glove dipping on polymer tank.
2. Leaching tank treatment.
3. Heating oven process.

## 3 Model

We model the DeepedGlove energy optimization problem as a Constrained Markov Decision Process. It precented by a tuple  $(S, A, T, r, c_{1:m}, \gamma)$ . Here  $S$  and  $A$  denotes state and action respectively.  $T(s_{t+1}|s_t, a_t)$  is transision dynamics.  $r(s_t, a_t) > 0$  is the reward function.  $c_{1:m}(s_t, a_t)$  are  $m$  cost function.  $\gamma \in (0, 1)$  is the discount factor.

- **States** ( $S$ ): We use states as temperature ( $t$ ) of each tank and ovens, rpm values of motors ( $m$ ), level values as ( $l$ ), and flow speed ( $f$ ).
- **Actions** ( $A$ ): Action values are rpm values of motors. All actions are continuous.
- **Reward Function** ( $r$ ): We model the reward as following equation

$$r_d = (w_1 \times Q_d) + (w_2 \times GN_d) - (w_3 \times EC_d) - (w_4 \times HF_d)$$

Here

- $Q_d$ : The quality of the gloves produced in the given day.
- $GN_d$ : Number of gloves manufactured in the given day.
- $EC_d$ : Energy consumption of the plant each day.
- $HF_d$ : Average temperature inside the plant in each day.

The  $w_i \in [0, 1]$  values are set according to priorities.

- **Cost Functions** ( $c_{1:m}$ ): We model safety constraints as costs. Such as the temperature ranges of each section, and motor speed of each sections.

The model is trained using historical operational dataset  $B = (s, a, s', r, c_{1:m})$  generated by the operators. The model learns optimal policy  $\pi^*(s)$  from  $B$  that maximize the expected discounted reward

$$R(\pi) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

While keeping the the expected discounted cost

$$C(\pi) = \mathbb{E}_{\pi} \left[ \sum_{t=0}^{\infty} r^t c(s_t, a_t) \right]$$

below a threshold  $l$ .

### 3.1 MORE

This model uses Model-based Offline RL with Restrictive Exploration (MORE)[1] for the agent. MORE uses two types of Q-functions for reward maximization ( $Q_r$ ) and for cost evaluation ( $Q_c$ ). To handle overestimate problem and panelize the uncertainty of  $Q_r$  MORE uses the clipped double Q method [4] using two  $Q_r$  networks. Since  $Q_c$  normally underestimate the cost, it doesn't uses this method.

The policy optimization is done by Lagrangian relaxation procedure.

$$L(\pi, \lambda) = \mathbb{E}_{s \sim D, a \sim \pi} \left[ \min_{j=1,2} Q_{r_j}(s, a) - \lambda(Q_c(s, a) - l) \right]$$

$$(\pi^*, \lambda^*) = \arg \min_{\lambda \geq 0} \max_{\pi} L(\pi, \lambda)$$

$$\lambda \leftarrow [\lambda + \eta (\mathbb{E}_{a \sim \pi} [Q_c(s, a)] - l)]^+$$

Here  $\lambda$  is the Lagrangian multiplier,  $\eta$  is the step size, and  $[x]^+ = \max\{0, x\}$ .

## 4 Simulator

The simulator is based on recurrent neural network (RNN), with it's internal cel structure is designed according to the actual physical process.

$$Glove \ dipping \ process \Rightarrow Leaching \ tank \ treatment \Rightarrow Heating \ oven \ process$$

Given current state ( $s_t$ ) and action ( $a_t$ ) the simulator model the next state ( $s'_{t+1}$ ), reward ( $r'_{t+1}$ ), and the cost ( $c'_{t+1}$ ).

## 5 References

1. DeepThermal: Combustion Optimization for Thermal Power Generating Units Using Offline Reinforcement Learning
2. LiteLSTM Architecture for Deep Recurrent Neural Networks
3. Constrained Markov Decision Processes
4. Addressing Function Approximation Error in Actor-Critic Methods.