Abstract
Intro
Related work
method
Exp
Citations

How is our work
better than others

Journal ~~B~~ of building
engineering
Building simulation

5 days to open the app

⇒ Train on March
Test on Jan and April

⇒ Train on Dec
Test on Nov and Dec. (slight overlap)

⇒ Exp 13, 14, 15 ⇒ All uses Bestest and sparse reward.
↓
Vanilla policy gradient on sparse reward

Exp 14 ⇒ Transfer learning ~~using~~ at agent level.

Exp 15 ⇒ Trasf lea. at env-model level.

No transfer learning at all here.

Exp 4 ⇒
Exp 7 ⇒ Model is for constant pricing scenario on bop test_hydronic_heat fat.

Model is learnt using Dynamic pricing

Gets from expriment 2 (policy and model based RL with Dynamic pricing scenario)

(⇒ We may want to add an experiment where pretrained agent for constant pricing scenario is used)

⇒ methodology

⇒ Same or model based RL with policy gradient / Reinforce

⇒ Transfer learning

actions are added ⇒ NN ⇒ statespace remains same.

⇒ How sparse reward is created and used in policy gradient algorithm

⇒

ALGORITHM STARTS BELOW

⇓

1 $\theta = \theta_I \Leftarrow$ Initialized

$$\text{policy\_Loss} = -\sum_{t=0}^{T=1344} \log\left(\pi_{\theta_I}(s_t)\right) R_t$$

where, $a_t = \pi_\theta(s_t)$, where, $R_t = 0$ if, $t \% 50 \,!= 0$

$\Rightarrow$ The negetive sign is to indicate we need to carry out gradient ascent instead of descent.

After training, $\theta = \theta_T$

(2) Approach 2: Initialize $\theta^p = \theta^p_I$.

Pretraining:

pump.

$$\text{policy\_loss} = -\sum_{t=1}^{T=672} \log\left(P_{\theta^p}(s_t)\right) \cdot R_t$$

After pretraining: $\theta^p = \theta^p_{PT}$

Fine tuning:

policy_loss

$$= -\sum_{t=1}^{T=1344} \log\left(P_{\theta^p}(s_t)\right) R_t$$

where,

$R_t = 0$ if $t \% 50 != 0$

After Finetuning:

$\theta^p = \theta^p_F$

③ Approach 3:

Here we pretrain using an environment model and finetune the environment model on bestest_hydronic.

We then use the finetuned environment model to train the RL agent

Initialize: $\Theta^E = \Theta^E_I$

Pretraining:

$$\left(E_{\Theta^E}(s_t, a_t)[0] - s_{t+1}\right)^2 + \left(E_{\Theta^E}(s_t, a_t)[1] - R_t\right)^2$$

After pretraining: $\Theta^E = \Theta^E_{P.T}$

Finetuning

$$env\_loss = \sum_{t=0}^{T=1344} \left(E_{\Theta_{P.T}}(s_t, a_t)[0] - s_{t+1}\right)^2 + \left(E_{\Theta_{P.T}}(s_t, a_t)[1] - R_t\right)^2$$

⇒ only calculate loss when t % 50 != 0

After Finetuning: $\Theta^E = \Theta^E_F$

Now training the policy gradient

$$\text{policy\_loss} = -\sum_{t=1}^{T=1344} \log\left(P_{\theta_I^p}(s_t)\right) \cdot R_t$$

where,

If $t \% 50 == 0 \Rightarrow$ Use the actual environment

$t \% 50 \,!= 0 \Rightarrow$ Use the surrogate environment.