I ran the following 9 CQL policy:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q-function | batch-normalization | dropout | alpha |
| mean | mean | 🗶 | 🗶 | 1 |
| qr | qr | 🗶 | 🗶 | 1 |
| iqn | iqn | 🗶 | 🗶 | 1 |
| qr\_do | qr | 🗶 | 0.5 | 1 |
| qr\_bndo | qr | ✓ | 0.5 | 1 |
| qr\_bndo\_alph0 | qr | ✓ | 0.5 | 0 |
| qr\_bndo\_alph0.5 | qr | ✓ | 0.5 | 0.5 |
| qr\_bndo\_alph5 | qr | ✓ | 0.5 | 5 |
| bcq\_qr\_bndo | qr | ✓ | 0.5 | 1 |

Note that the last one ‘bcq\_qr\_bndo’ uses bcq instead of CQL. Here is the initial\_state\_value and TD\_error for each of the policies, with names for each policy referred to as follows:

|  |  |
| --- | --- |
| Name | Policy |
| CQL\_0 | lim\_q\_mean |
| CQL\_1 | lim\_q\_qr |
| CQL\_2 | lim\_q\_iqn |
| CQL\_3 | lim\_q\_do |
| CQL\_4 | lim\_q\_qr\_bndo |
| CQL\_5 | lim\_q\_qr\_bndo\_alph0.5 |
| CQL\_6 | lim\_q\_qr\_bndo\_alph0 |
| CQL\_7 | lim\_q\_qr\_bndo\_alph5 |
| BCQ\_0 | bcq\_q\_qr\_bndo |

图表, 折线图

描述已自动生成

As can be seen from the graphs, although using the mean Q-function results in higher ECR, it also leads to higher TD\_error, which suggests that it is prone to overfitting, hence not a suitable policy for future prediction. In contrast, the policy ‘CQL\_7’ that uses quantile regression with dropout rate 0.5 and batch-normalization has a relatively high ECR and low TD-error.

After that I ran FQE for mean, qr, iqn, qr\_bndo\_alph0, qr\_bndo\_alph5, BP. The result is as follows, with the name each model is referring to:

|  |  |
| --- | --- |
| Name | Policy |
| CQL\_0 | fqe\_mean |
| CQL\_1 | fqe\_qr |
| CQL\_2 | fqe\_q\_iqn |
| CQL\_6 | fqe\_qr\_bndo\_alph0 |
| CQL\_7 | fqe\_qr\_bndo\_alph5 |
| BP | fqe\_bp\_deterministic |
|  |  |

Note that the parameters shown corresponds to the CQL policy it evaluates, not the parameters used for evaluation.

图表, 折线图

描述已自动生成

As can be seen from the graphs, the policy ‘fqe\_qr\_bndo\_alph0’ produced the highest ECR, but also highest loss, and it is more unstable across epochs. Like earlier results, the policy ‘fqe\_qr\_bndo\_alph5’ has a relatively high ECR yet low loss. The BP policy was significantly outperformed by the various CQL policies.

The cumulative reward is given as follows:

图表, 图示

中度可信度描述已自动生成

As can be seen from the graph, the reward for most of the policies start to diminish after 200 steps. All the policies outperform the behaviour\_policy. The policies MDP\_aug46, MDP\_aug468, MDP\_aug4687, yields the highest rewards with MDP\_aug4687 being slightly better than the other two. Behaviour policy still acts as a baseline for comparison.