Results analysis：

Introduction: This report introduced the results of the simulation. In the first part, the environment is introduced. Then the performance of traditional handover decision and the model trained with different AI algorithms will be compared.

Method of obtaining data: The simulation is made in a 40x60 grid map. Base stations are fixed. The reward function of ai algorithms is mainly focused on the optimization of the number of handovers.

Each base station has a different parameter.base\_station\_params = {

(3, 3):(20, 2, 4),

(15, 3):(20, 2, 4),

(27, 3):(20, 2, 4),

(3, 15):(10, 3, 3),

(15, 15):(10, 3, 3),

(27, 15):(10, 3, 3),

(3, 27):(15, 1, 5),

(15, 27):(15, 1, 5),

(27, 27):(15, 1, 5),

(3, 39):(15, 1, 5),

(15, 39):(15, 1, 5),

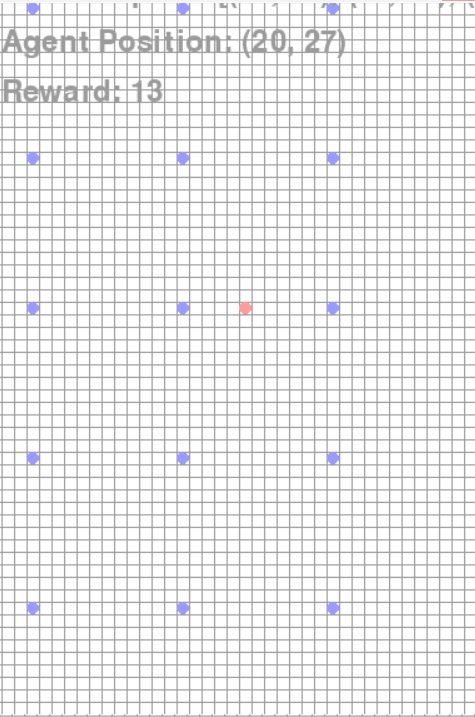
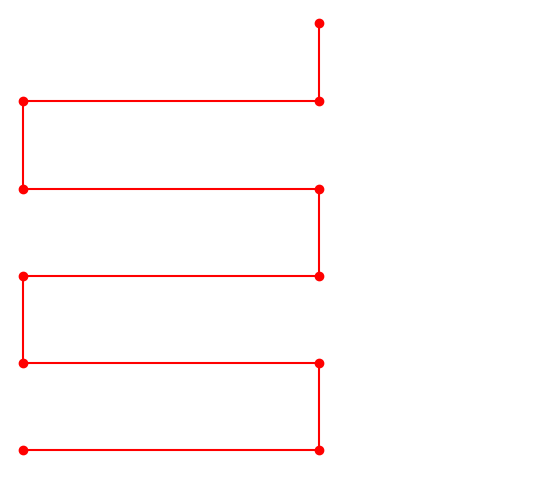
(27, 39):(15, 1, 5),

(3, 51):(15, 1, 5),

(15, 51):(15, 1, 5),

(27, 51):(5, 1, 5),

} the data form is : position: (Pt,Gt,Gr) where Pt is the transmit power, Gt is the transmit antenna gain, Gr is the receive antenna gain.



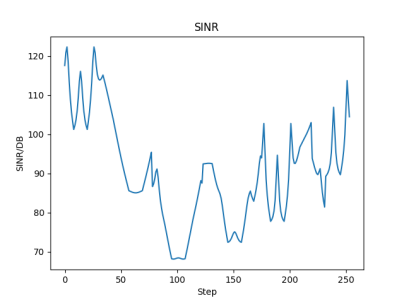
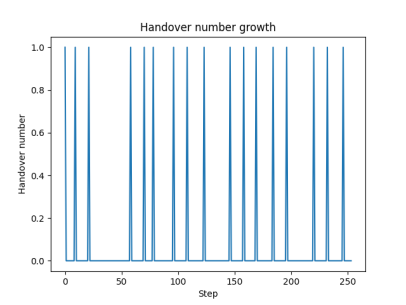
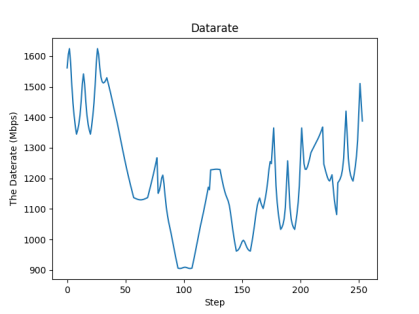
The picture on the left is the fixed route of the agent. the blue points in The picture on the right are the base stations.

When the agent moves in the fixed route, I collect some parameters (SINR, datarate).

1,Model 1

This strategy for the base station to make handover is based on the signal strength. The agent selects the basestation with the best signal strength. And there is a threshold and a delay time (Time to Trigger) to avoid the Ping-pong effect. I tried serveral different TTT parameters to achieve the best performance.

No TTT:

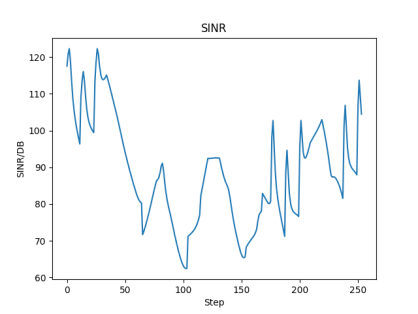
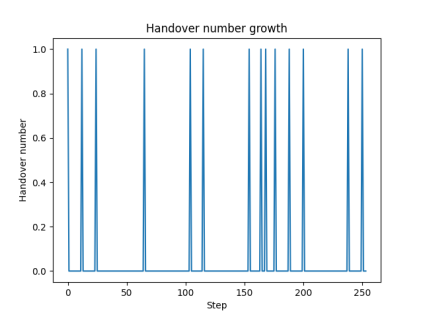
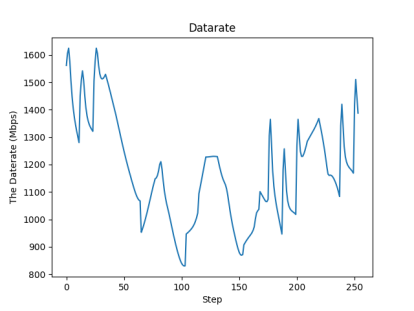


Average handover number: 17

average SINR 90.44292815698905

datarate 1201.7796169946182

TTT=9ms

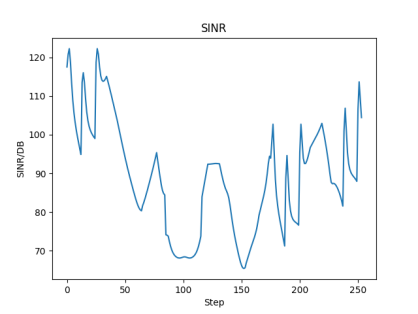
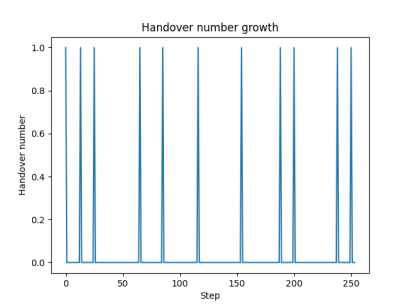
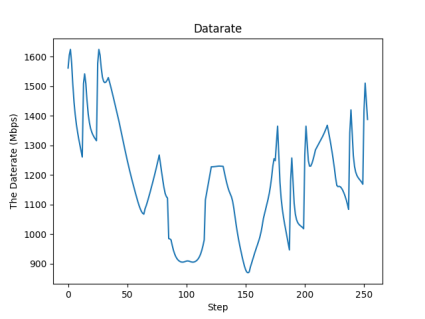


Handover number: 14

average SINR 88.64279919083639

datarate 1177.8600220156425

TTT=10ms

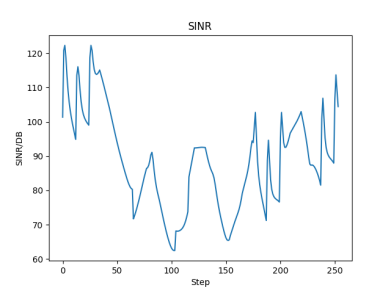
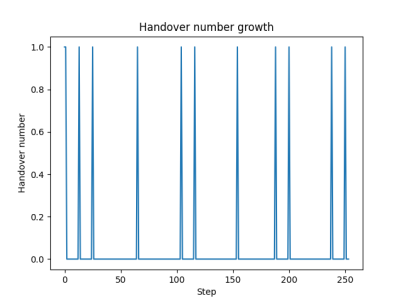
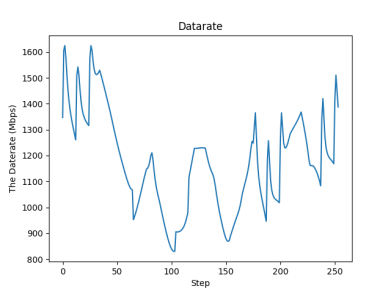


Average handover number: 11

average SINR 89.0295028011665

datarate 1182.9984280988297

TTT= 11ms



Average handover number: 12

average SINR 88.56339370462473

datarate 1176.804904926929

Too long a TTT leads to a decrease in SINR and datarate, and we chose TTT=10ms as the best model for comparison.

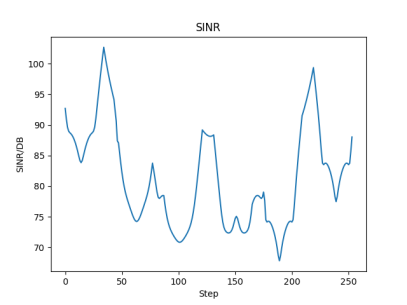
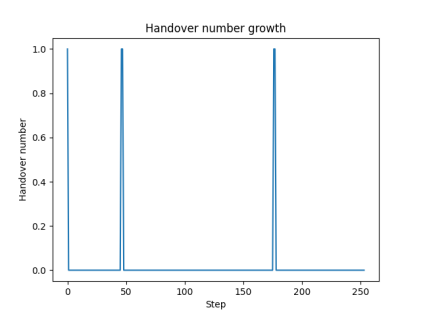
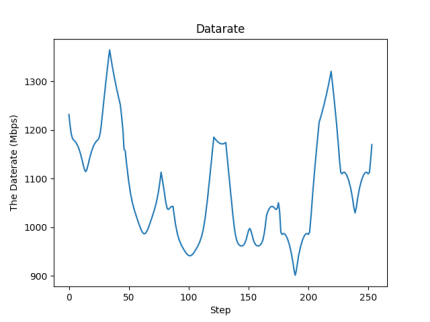
In this part, the agent moves in the same route. I use the trained model to predict the handover and collect the date 6 times. And the three pictures at the end is from a single test.

PPO(Proximal Policy Optimization) algorithm

The performance of AI algorithm in the same environment.

1, the model was trained with PPO algorithm and 50000 training loops.

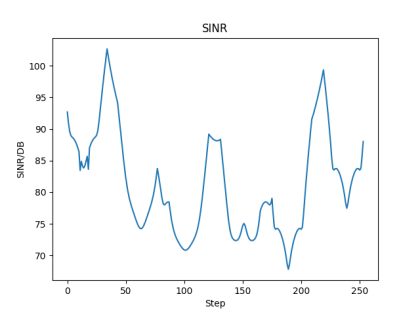
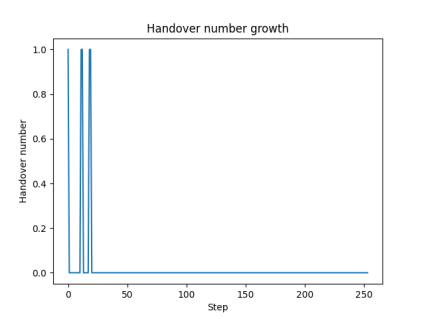
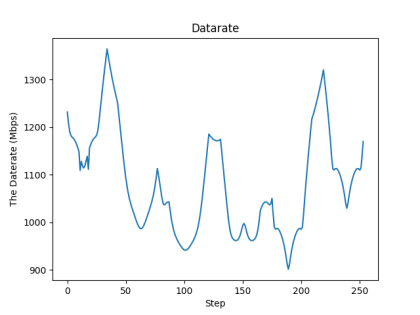
|  |  |  |  |
| --- | --- | --- | --- |
| PPO with 50000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 5 | 82.31 | 1080.47 |
| 2 | 7 | 81.35 | 1080.96 |
| 3 | 5 | 81.4 | 1081.67 |
| 4 | 5 | 81.36 | 1081.11 |
| 5 | 5 | 81.31 | 1080.45 |
| 6 | 12 | 81.3 | 1080.38 |
| average | 6.5 | 81.505 | 1080.84 |



The pictures are from Sample 4

2, the model was trained with PPO and 100000 training loops.

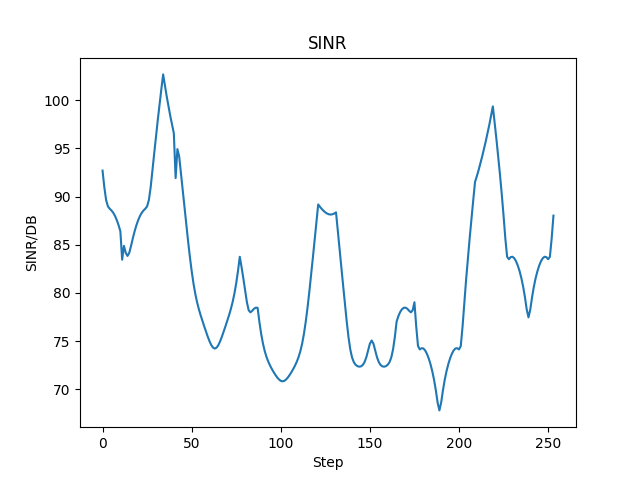
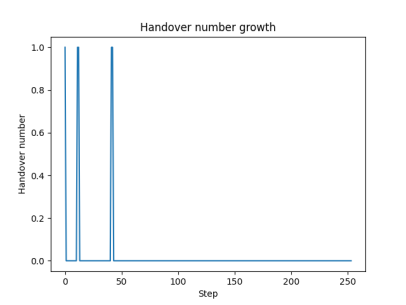
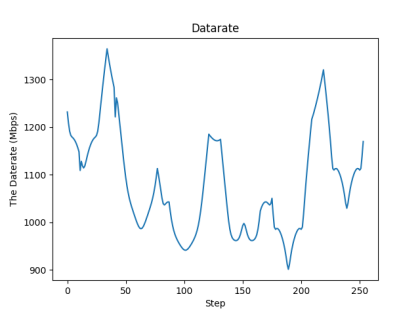
|  |  |  |  |
| --- | --- | --- | --- |
| PPO with 100000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 5 | 81.33 | 1080.7 |
| 2 | 5 | 81.33 | 1080.66 |
| 3 | 3 | 81.34 | 1080.92 |
| 4 | 5 | 81.34 | 1080.86 |
| 5 | 3 | 81.40 | 1081.66 |
| 6 | 9 | 81.37 | 1081.19 |
| average | 5 | 81.35166667 | 1080.998333 |



The pictures are from Sample 4

|  |  |  |  |
| --- | --- | --- | --- |
| PPO with 300000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 3 | 81.36 | 1081.04 |
| 2 | 7 | 81.31 | 1080.45 |
| 3 | 3 | 81.38 | 1081.31 |
| 4 | 5 | 81.34 | 1080.81 |
| 5 | 3 | 81.38 | 1081.37 |
| 6 | 5 | 81.31 | 1080.52 |
| average | 4.333333333 | 81.34666667 | 1080.916667 |

3, the model was trained with PPO and 300000 training loops:

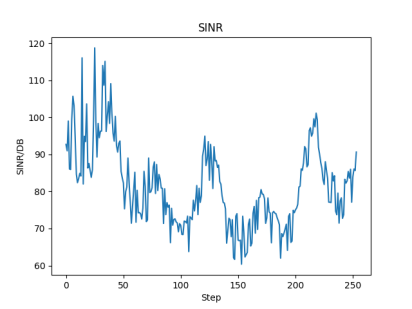
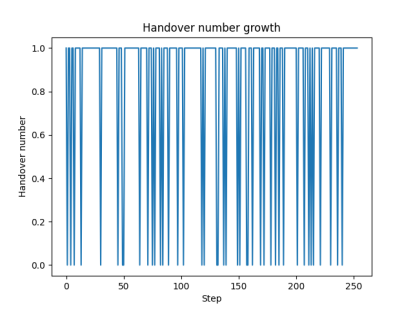
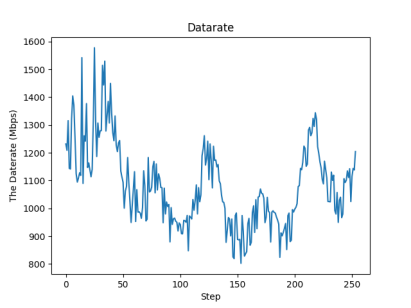


The pictures are from Sample 4

A2C (Advantage actor critic) algorithm

1,the model was trained with A2C and 50000 training loops:

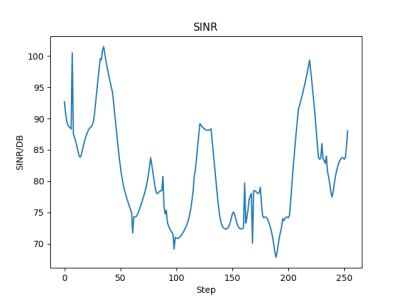
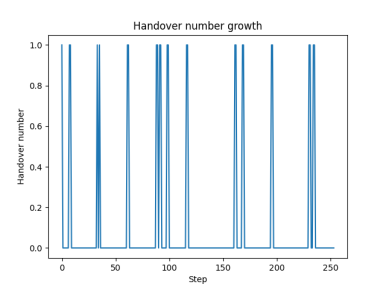
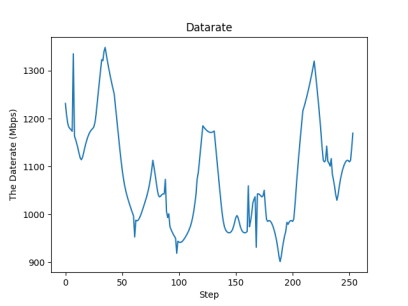
|  |  |  |  |
| --- | --- | --- | --- |
| A2C with 50000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 201 | 81.35 | 1081.01 |
| 2 | 213 | 81.64 | 1084.78 |
| 3 | 209 | 80.72 | 1072.57 |
| 4 | 211 | 81.45 | 1082.3 |
| 5 | 207 | 80.98 | 1075.98 |
| 6 | 204 | 81.73 | 1085.99 |
| average | 207.5 | 81.31166667 | 1080.438333 |



The pictures are from sample 4.

2,the model was trained with A2C and 100000 training loops:

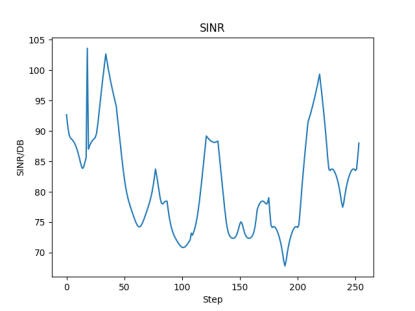
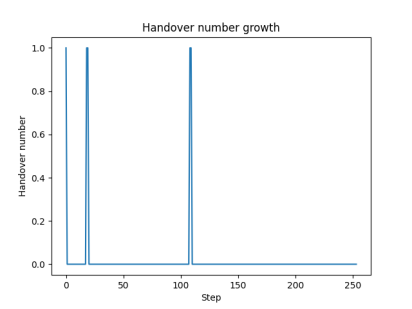
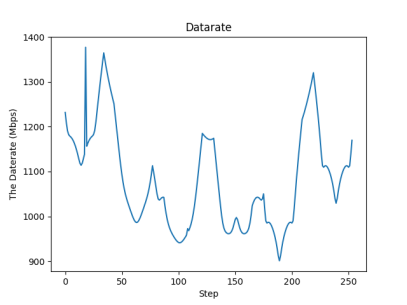
|  |  |  |  |
| --- | --- | --- | --- |
| A2C with 100000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 16 | 81.36 | 1081.15 |
| 2 | 29 | 81.38 | 1081.37 |
| 3 | 19 | 81.23 | 1079.39 |
| 4 | 25 | 81.42 | 1081.82 |
| 5 | 21 | 81.31 | 1080.37 |
| 6 | 28 | 81.21 | 1079.08 |
| average | 23 | 81.31833333 | 1080.53 |



The pictures are from sample 4.

3,the model was trained with A2C and 300000 training loops:

|  |  |  |  |
| --- | --- | --- | --- |
| A2C with 300000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 3 | 81.33 | 1080.71 |
| 2 | 1 | 81.36 | 1081.1 |
| 3 | 3 | 81.46 | 1081.34 |
| 4 | 5 | 81.43 | 1082.06 |
| 5 | 3 | 81.35 | 1081 |
| 6 | 3 | 81.37 | 1081.22 |
| average | 3 | 81.38333333 | 1081.238333 |

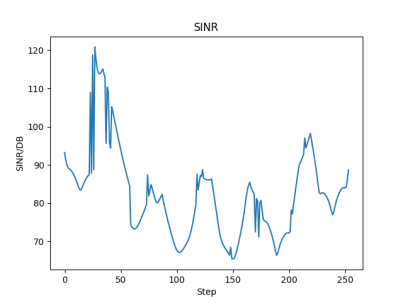
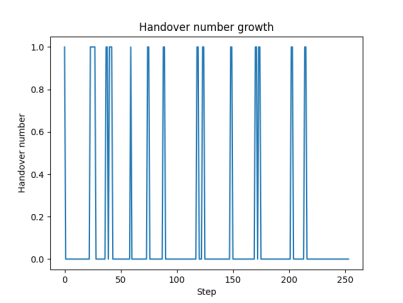
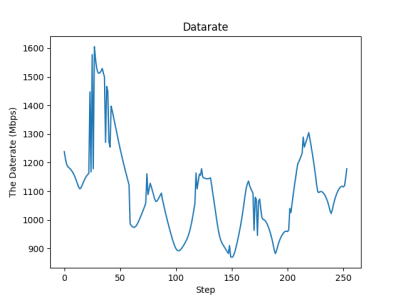


The pictures are from sample 4.

DQN(Deep Q network) algorithm:

1, the model was trained with DQN algorithm and 50000 training loops.

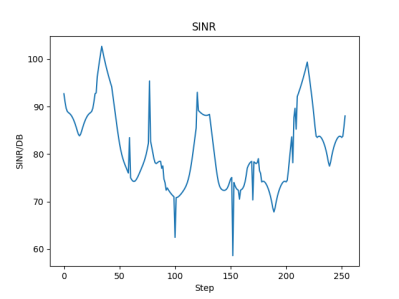
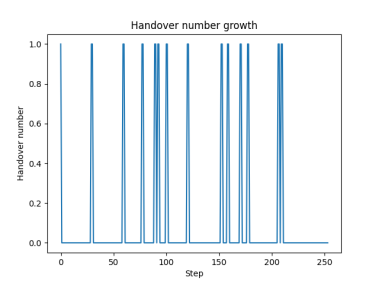
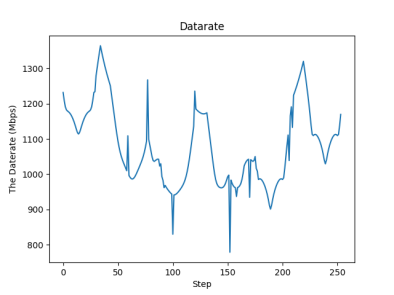
|  |  |  |  |
| --- | --- | --- | --- |
| DQN with 50000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 21 | 82.31 | 1093.67 |
| 2 | 30 | 82.34 | 1094.09 |
| 3 | 33 | 82.54 | 1096.73 |
| 4 | 30 | 82.44 | 1095.37 |
| 5 | 30 | 82.44 | 1095.37 |
| 6 | 25 | 82.51 | 1096.40 |
| average | 28.16666667 | 82.43 | 1095.271667 |



The pictures are from sample 4.

2, the model was trained with DQN algorithm and 100000 training loops.

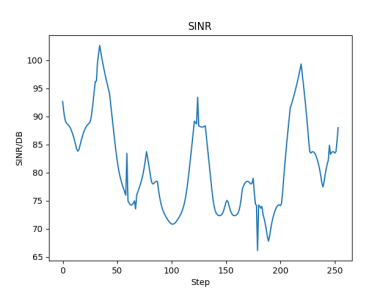
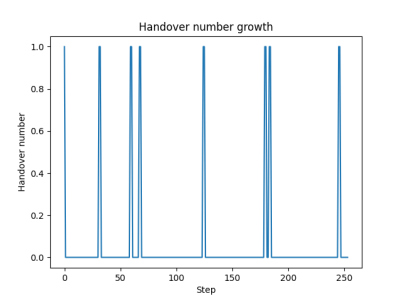
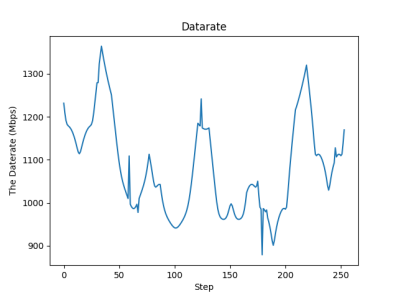
|  |  |  |  |
| --- | --- | --- | --- |
| DQN with 100000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 17 | 81.35 | 1080.92 |
| 2 | 23 | 81.35 | 1080.9 |
| 3 | 27 | 81.3 | 1080.26 |
| 4 | 27 | 81.27 | 1079.95 |
| 5 | 14 | 81.35 | 1080.98 |
| 6 | 17 | 81.43 | 1082.03 |
| average | 20.83333333 | 81.34166667 | 1080.84 |



The pictures are from sample 4.

3, the model was trained with DQN algorithm and 300000 training loops.

|  |  |  |  |
| --- | --- | --- | --- |
| DQN with 300000 training loop | | | |
| sample | Handovers | SINR | Datarate |
| 1 | 9 | 81.38 | 1081.31 |
| 2 | 15 | 81.38 | 1081.29 |
| 3 | 3 | 81.34 | 1080.87 |
| 4 | 15 | 81.38 | 1081.33 |
| 5 | 9 | 81.31 | 1080.44 |
| 6 | 8 | 81.32 | 1080.55 |
| average | 9.833333333 | 81.35166667 | 1080.965 |



The pictures are from sample 4.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PPO | | | A2C | | | DQN | | |
| training step | Handover | SINR | Datarate | Handover | SINR | Datarate | Handover | SINR | Datarate |
| 50000 | 6.5 | 81.50 | 1080.84 | 207.5 | 81.32 | 1080.44 | 28.16 | 82.43 | 1095.27 |
| 100000 | 5 | 81.35 | 1081.0 | 23 | 81.32 | 1080.53 | 20.83 | 81.34 | 1080.84 |
| 300000 | 4.3 | 81.35 | 1080.92 | 3 | 81.38 | 1081.24 | 9.83 | 81.35 | 1080.97 |

The reward function of these algorithms are the same, the main task is to reduce the number of handovers and try to keep the SINR in a acceptable range. The approach of PPO algorithm is the most quickly. The PPO algorithm with more training steps doesn’t improve significantly.

Overall, the PPO algorithm is the most efficient in terms of training. With 50000 training steps. The performance of PPO algorithm is the best. With 100000 training steps. The improvement of PPO algorithm is not very significant. But with more training steps the performance of A2C is better and finally after 300000 steps it can have a best performance among these three algorithms

Some important parameters in the training algorithms

1,neural network architecture. In this case the DQN algorithm used a deep neural network(DNN) as the base function. The default architecture of the neural network is multi-layer perceptron (MLP) which has two hidden layers of 64 units each and the activation function is Rectified linear Unit(ReLU)

2,start learning steps. When the agent is trained with DQN algorithm. There is a learning\_start parameter. It means that the agent wont start learning until enough steps have been taken. In this way, the initial behavior of the agent my be only suboptimal. Because it just follow a random policy during the learning\_starts steps.

3, exploration vs exploitation, in the three algorithms A2C DQN PPO

They all need to balance the exploration with exploitation. Exploration in reinforcement learning means that the agent tries to find the new actions and the exploitation is using the bast-known action. In order to find the optimal policy, both of them are very important. If the exploration is to low, the agent will be easily drop into the suboptimal policy and the learning process will be slow.

4,Learning rate: If the learning rate is too high, the agent might not converge. If it is too low, the agent might converge too slowly.

Why the approach speed of DQN is slow and why the performance of DQN with 300000 training steps is the worst.

1, for DQN there is a learning start steps. That means the agent won’t learn anything during the start steps. In this case the approach speed will be slow down.

2, When the learning start steps are not large enough, the agent may get into a suboptimal area. That can be a reason for the bad performance with 300000 training steps.

3, DQN algorithm is more sensitive to hyperparameters than A2C and PPO, it is possible that the hyperparameters used in the traing is not well-suited for the specific task.

4, The exploration rate might be not enough, it may get stuck in a suboptimal policy. On the other hand, if it explores too much, it may take longer to converge to an optimal policy.

Why the approach speed of PPO is faster than A2C :

1,A2C is a algorithm that combines the policy-based and value based method. It uses a neural network to represent the policy and value function, and updates the model based on the advantage estimate. PPO used a clipped surrogate objective function. It helps to stabilize the learning process and prevent large updates to the policy parameters. This can lead to more consistant and faster learning compared to A2C.

2, PPO was updated with minibatch. It means that the update of policy parameters is based on the small batches of experience. However, the updates of A2C network is made after every timestep.

3, There is a value function baseline in PPO algorithm, which is used to reduce the variance of the policy gradient estimates. It can help to improve the accuracy of the gradient estimate and converge faster.

1,DQN is known to have a problem with instability, particularly when the environment has a large state space. In this environment, the state space is relatively large, which could have contributed to the instability of the DQN algorithm. PPO, on the other hand, is designed to be more stable than DQN, particularly when dealing with large state spaces.

2,PPO is a more advanced algorithm than DQN, and it incorporates several improvements over the DQN algorithm. For example, PPO uses an adaptive clipping parameter to prevent large policy updates, which can help prevent the policy from collapsing. PPO also uses a value function network to estimate the state value, which can help improve the stability of the algorithm.