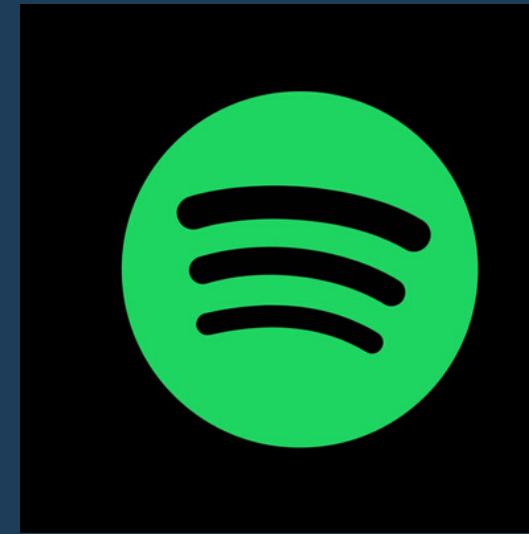


SEMESTER PROJECT : SPRING 2020
EURECOM



REINFORCEMENT LEARNING FOR CACHE-FRIENDLY RECOMMENDATIONS

Clément Bernard & Jade Bonnet



RECOMMENDATION APPLICATIONS

Use state-of-the art recommendation algorithms but ...
don't take into account network latency !



Cache-friendly recommender

Our goal



Recommendation
algorithm



Users' preferences

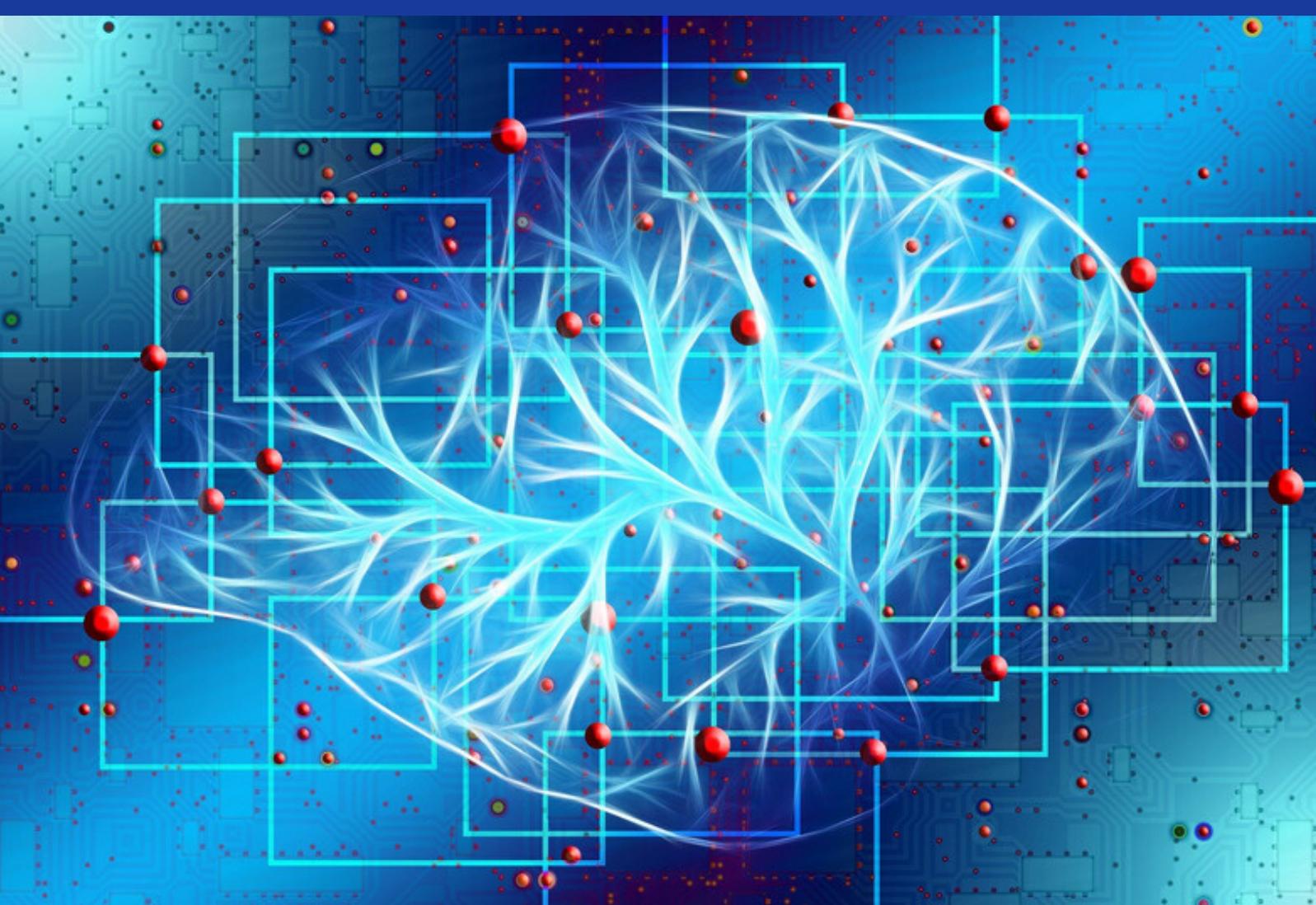


Network latency



Cached-friendly recommender

Algorithm that recommends contents
that can be cached and related



Reinforcement learning

Algorithm that learns by experience

REINFORCEMENT LEARNING

ONE OF THE THREE PARADIGMS OF MACHINE LEARNING

Like supervised and unsupervised learning, reinforcement learning is one of the three main parts of machine learning area

LEARN FROM EXPERIENCE

Algorithm that will interact with the user to learn his preferences and even more ...

Covered Today

Notations

Reinforcement learning concepts

Q-learning

Deep Q-learning

Double Q-learning

Limits & Conclusions

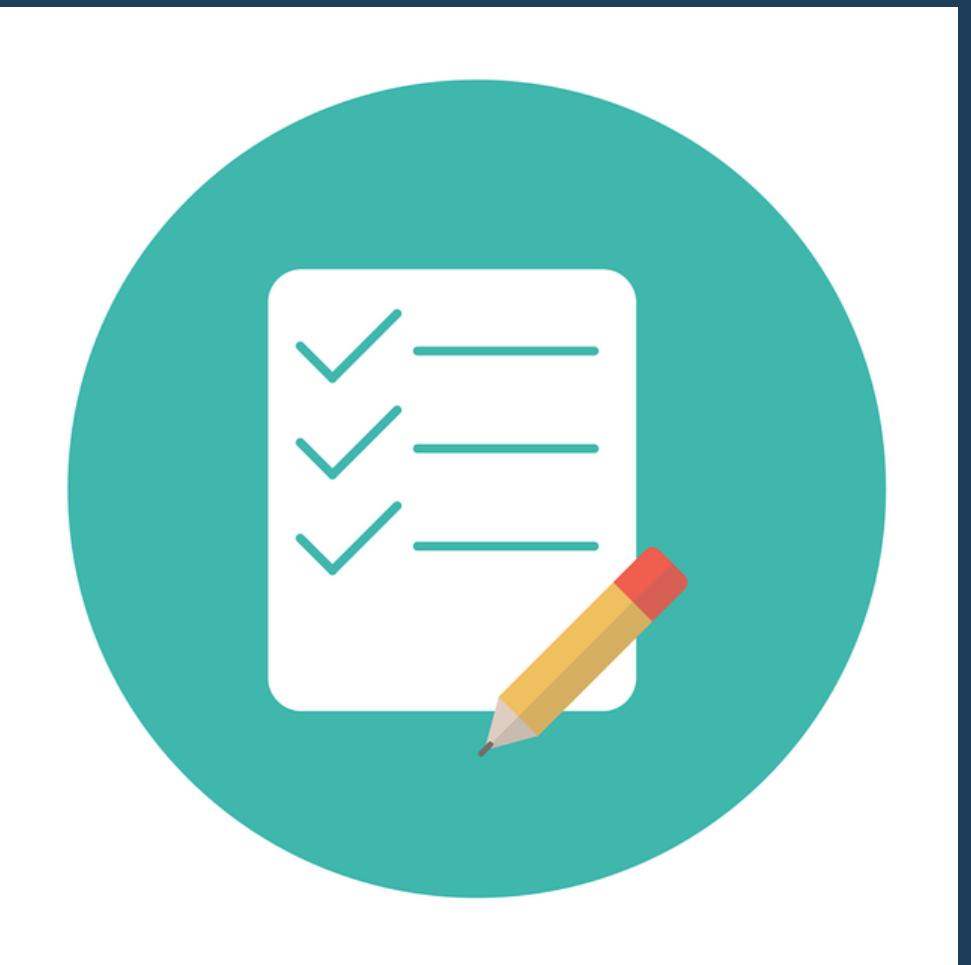
Notations



WHAT IS CACHED, RELATED ?

CATALOGUE

Videos, musics, movies are all coded in the same way as a file/content in the Catalogue



```
array([0.          , 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.          , 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.          , 0.          , 0.91747976, 0.95267253, 0.          , 0.          ,
       0.90697197, 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.97826924, 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.          , 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.91222267, 0.          , 0.          , 0.85051224, 0.          , 0.          ,
       0.84753072, 0.          , 0.          , 0.          , 0.          , 0.          ,
       0.97477846, 0.          , 0.          , 0.88544912, 0.          , 0.          ,
       0.91145699, 0.          , 0.          , 0.          , 0.          , 0.          ]) )
```

U VALUES FOR CONTENT 0, CATALOGUE SIZE OF 50

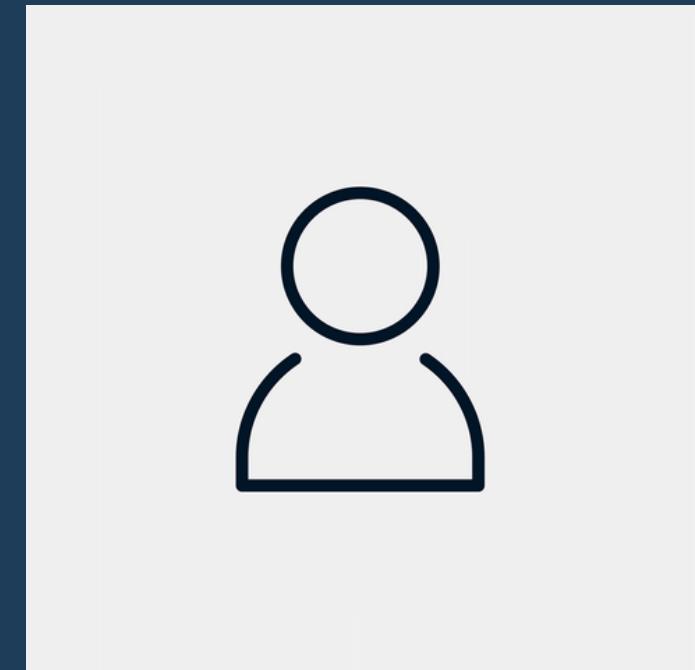
Related

U MATRIX THAT GIVES A SCORE OF SIMILARITY

COST FOR THE CONTENTS

Cached

LIST THAT GIVES THE COST FOR CONSUMING A CONTENT



MARKOVIAN USER

- Leave the simulation with probability alpha
- Listen the suggestion with probability a (otherwise pick randomly a content among the catalogue)

Reinforcement learning concepts



AGENT, STATES, ACTIONS, REWARDS, ...

MARKOVIAN DECISION PROCESSES



AGENT

Recommendation
algorithm

ENVIRONMENT

The user (like markovian
user)

STATES/ACTIONS

State : content that is
consumed by the user

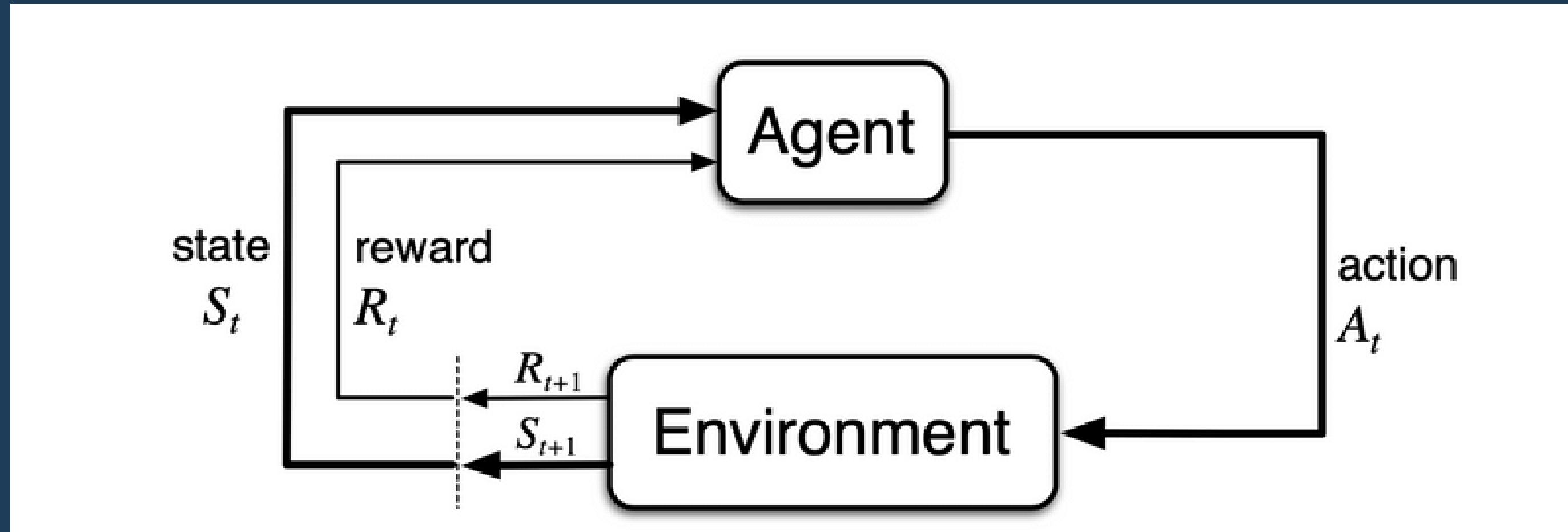
Action : content that is
suggested by the agent

REWARD

Feedback

POLICY

Mapping from states to
probabilities of selecting
an action



Reinforcement learning timeline

EXPECTED RETURN

Expected reward from a given state

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{T-1} R_T = \sum_{k=0}^T \gamma^k R_{t+k+1}$$

$$q_\pi(s, a) = E_\pi(G_t | S_t = s, A_t = a)$$

Q-TABLE

Action-value function that gives the expected return for each couple (state, action)



Q-learning



Q-LEARNING UPDATE RULE

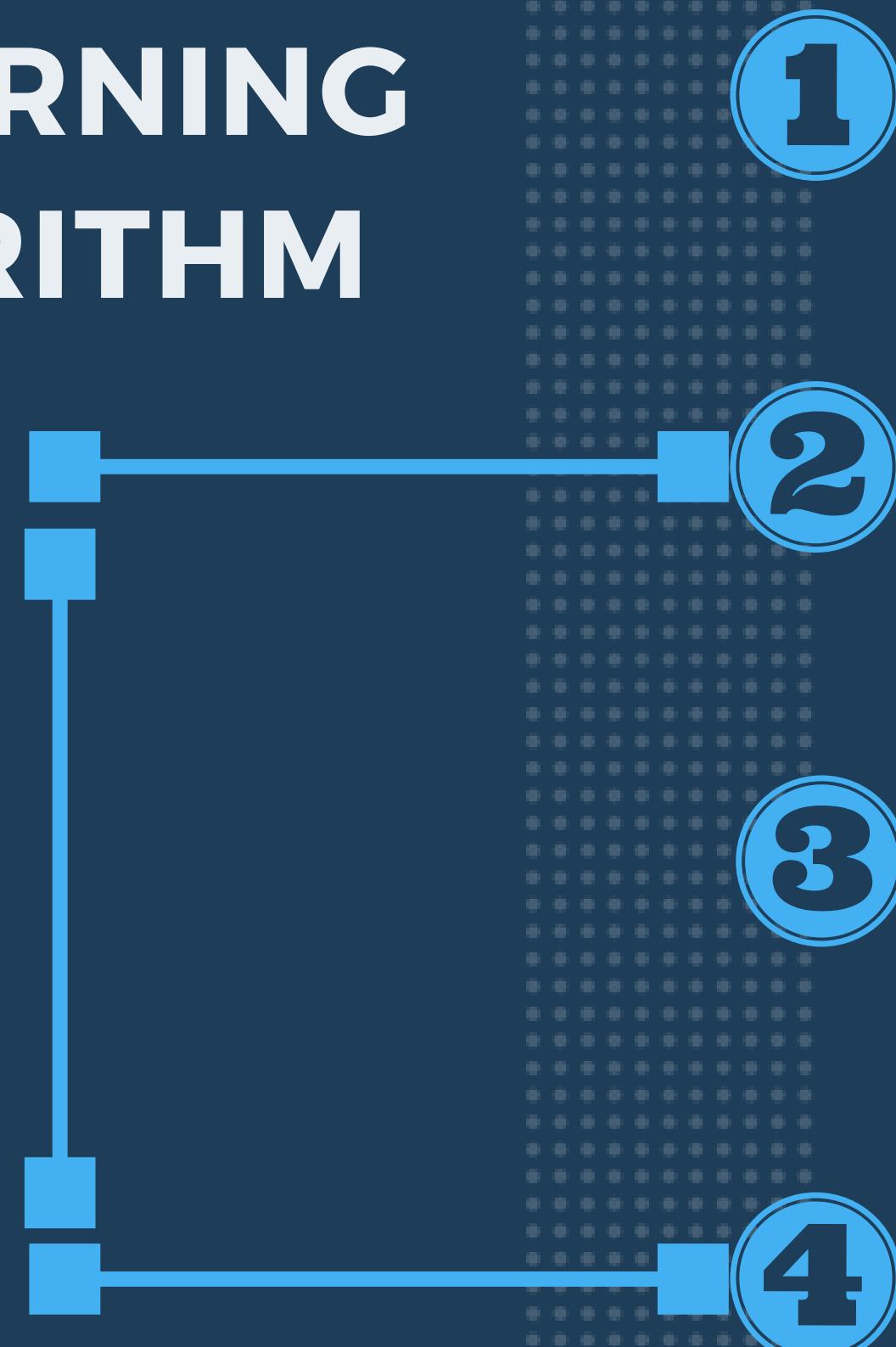
$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q_t(S_t, A_t))$$

current Q-value
learning rate reward discount max of future expected reward

The diagram illustrates the Q-learning update rule with four blue arrows pointing from labels to the equation:

- A double-headed arrow connects "current Q-value" to the term $Q_t(S_t, A_t)$.
- A double-headed arrow connects "learning rate" to the term α .
- A double-headed arrow connects "reward" to the term R_{t+1} .
- A double-headed arrow connects "max of future expected reward" to the term $\max_a Q(S_{t+1}, a)$.
- A single-headed arrow points from "discount" to the term γ .

Q-LEARNING ALGORITHM



Initialise the Q-table

Make zeros everywhere

Choose action from Q (epsilon-greedy)

Choose a content to recommend either by exploration or by exploitation

Perform action and get rewards

Interaction with the user and get the reward for the content that has been recommended

Update the Q table

Update the Q-table with the formula and start again from 2) with the new state

CHOOSE AN ACTION : EPSILON- GREEDY

EXPLORATION

- Search among all the catalogue to not miss good contents

EXPLOITATION

Exploit the learning process by taking the action that maximises the Q-table

Hyperparameters

LEARNING RATE

DISCOUNTED GAMMA

How far does the agent take into account
the actions

RELATED CONTENTS

Number of related contents for each item

CACHED CONTENT

Number of cached content in the
catalogue

ENVIRONMENT

HYPERPARAMETERS

Results

- CATALOGUE OF SIZE 50
- 10 RELATED CONTENTS BY STATE
 - 5 CACHED CONTENTS
 - LEARNING RATE : 0.2

REWARDS

NEITHER
CACHED NOR
RELATED

0

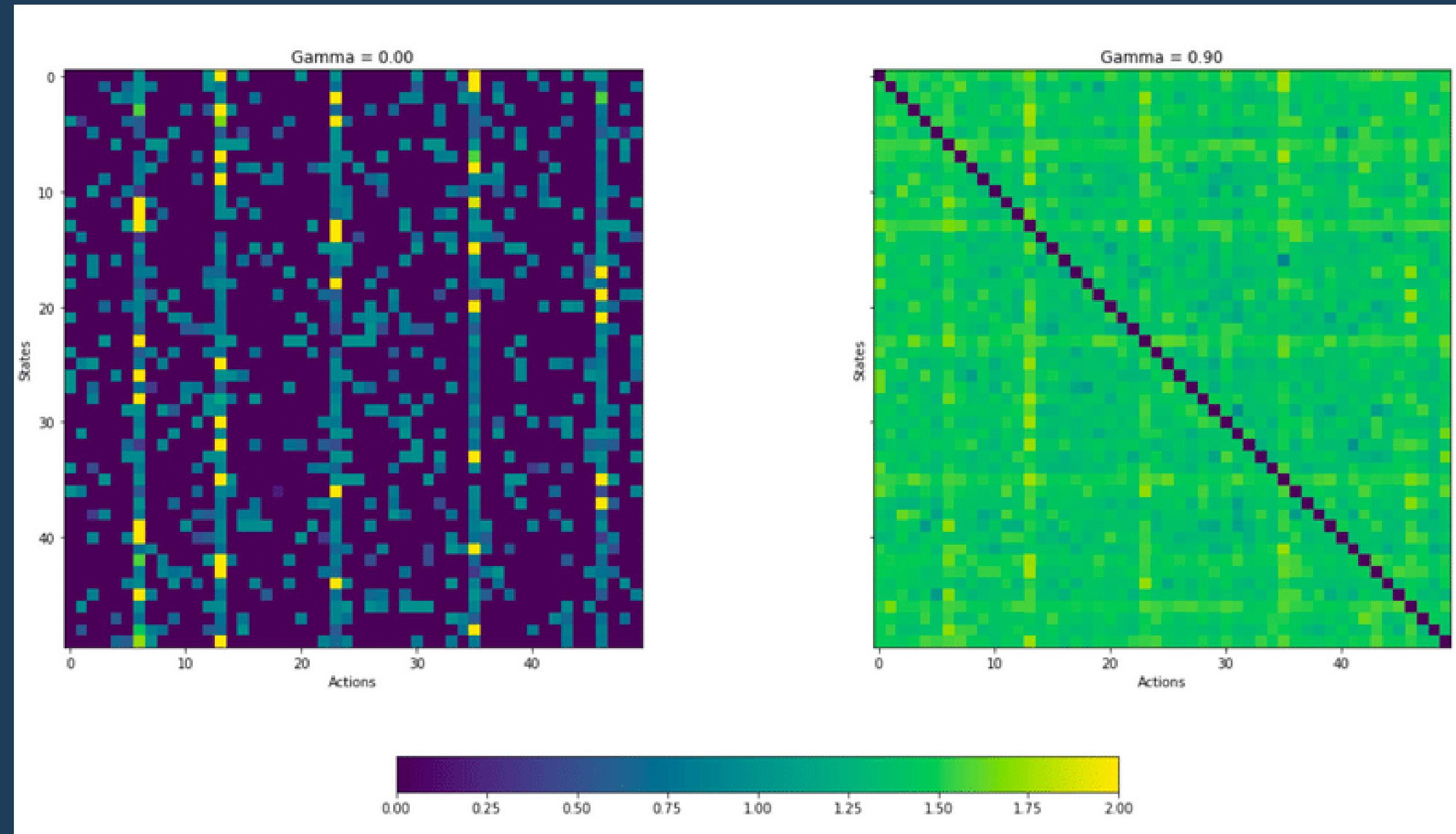
CACHED OR
RELATED

+1

CACHED AND
RELATED

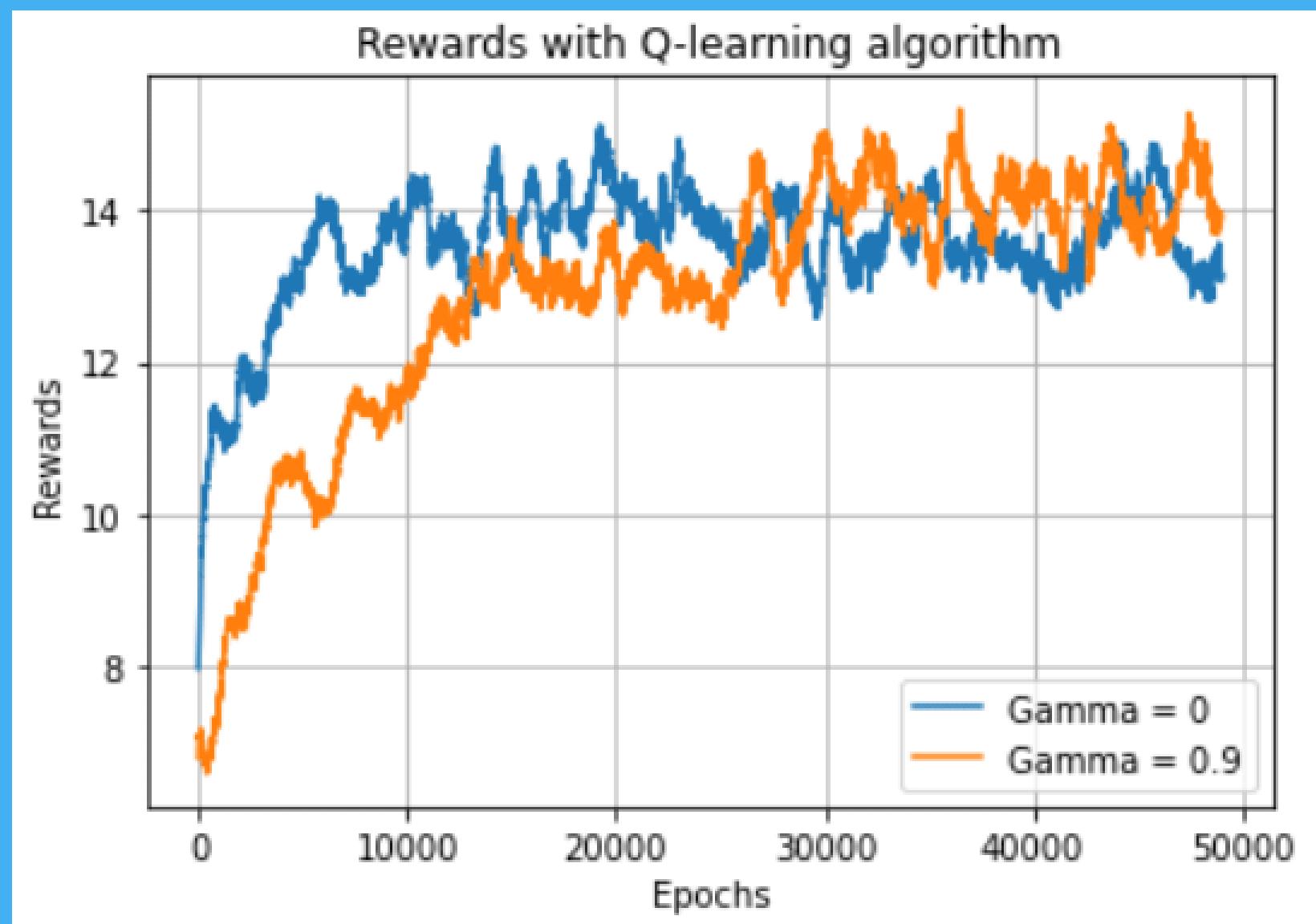
+2

Q-table

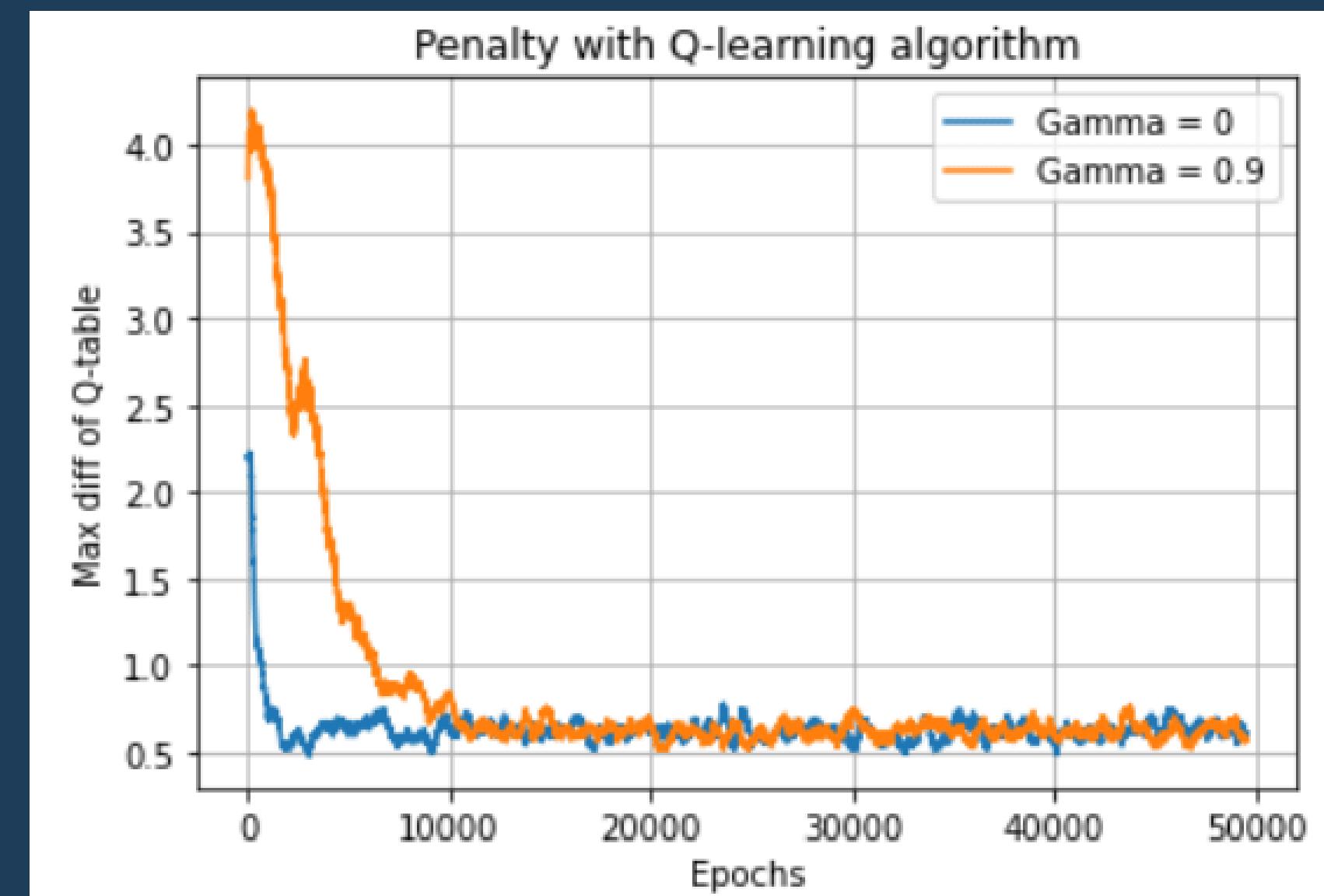


Gamma = 0 (left) and Gamma = 0.9 (right)

Rewards per epoch

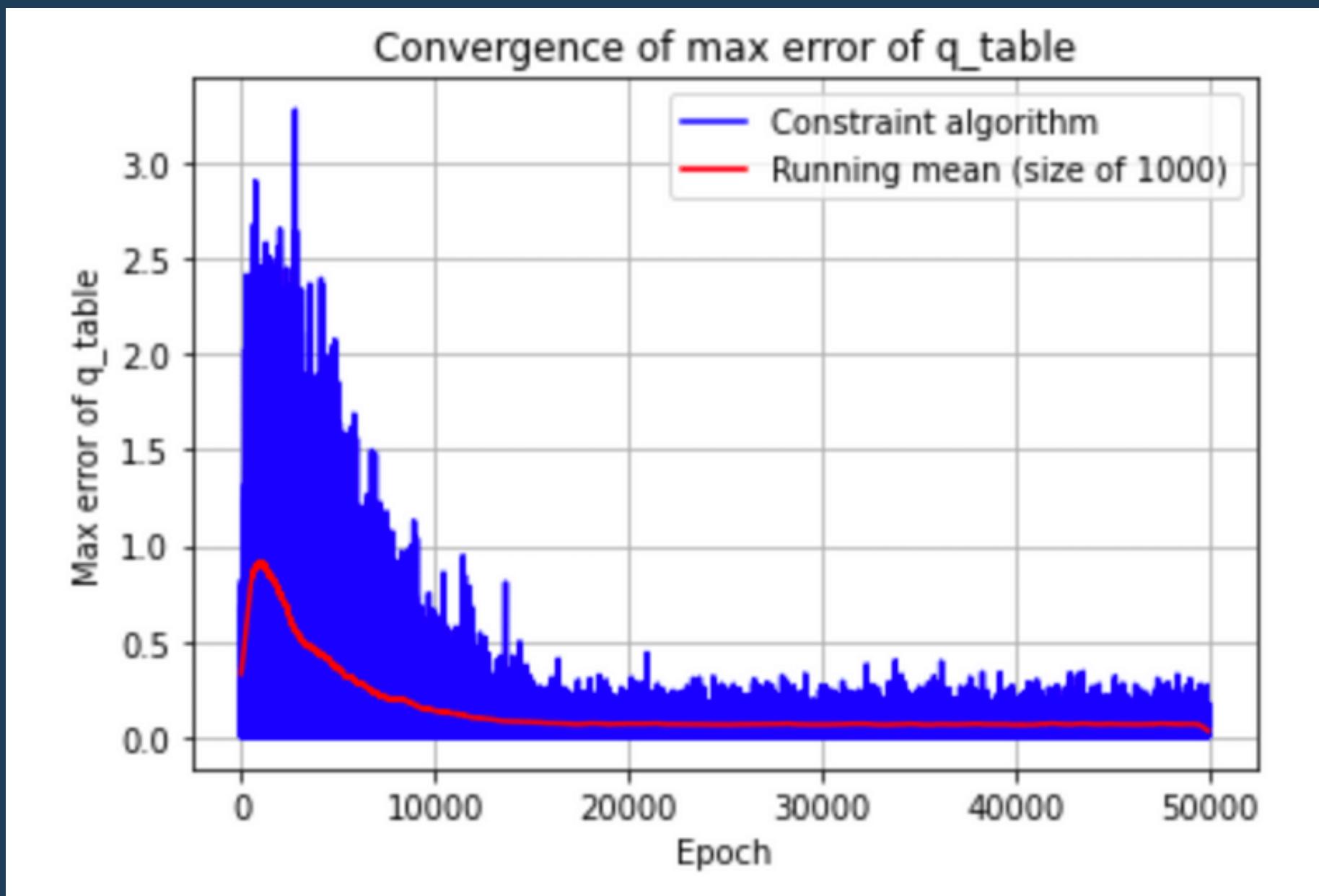


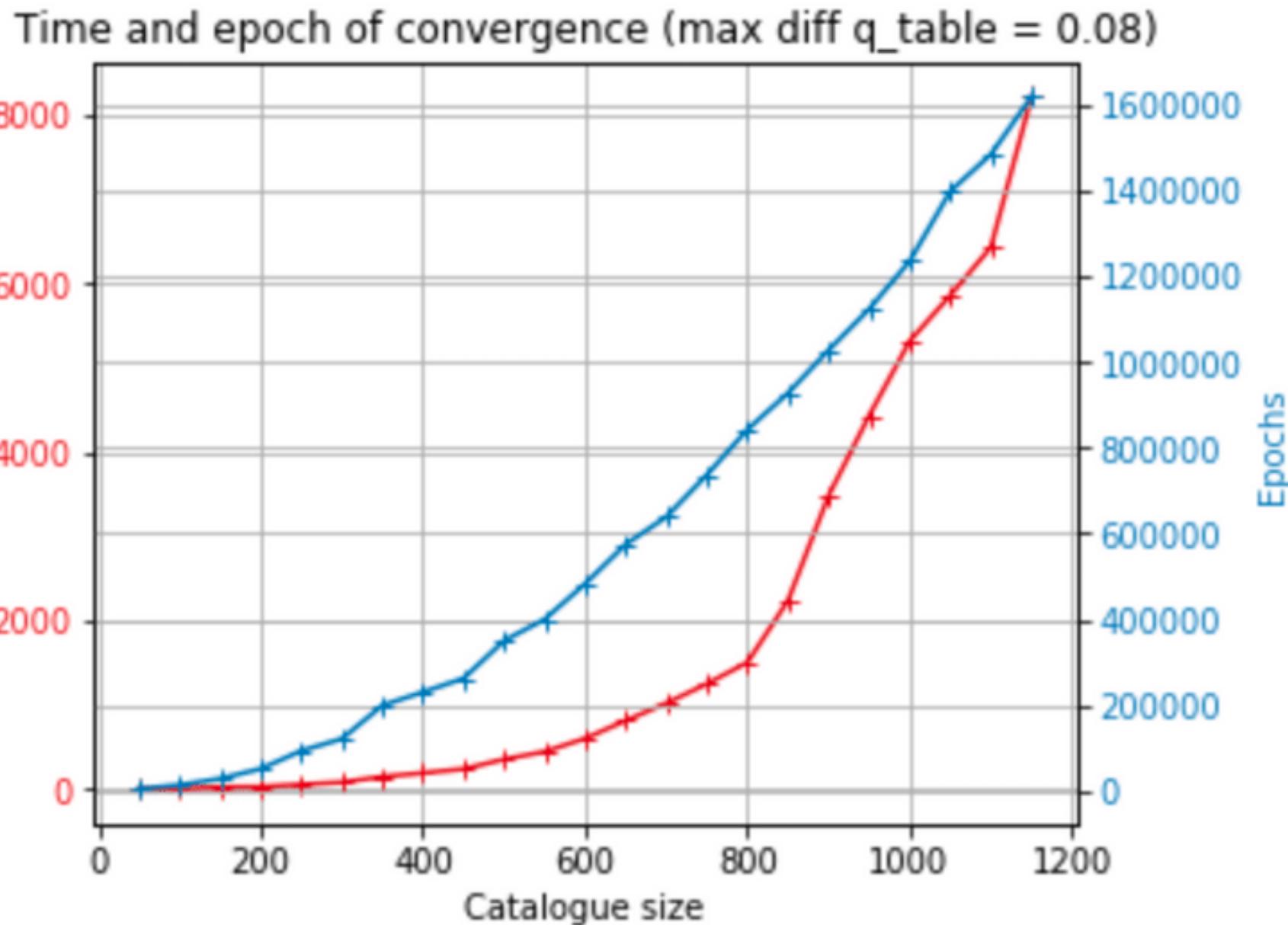
Penalty per epoch



Convergence criteria

Maximum difference of Q-table for one epoch





Time of convergence

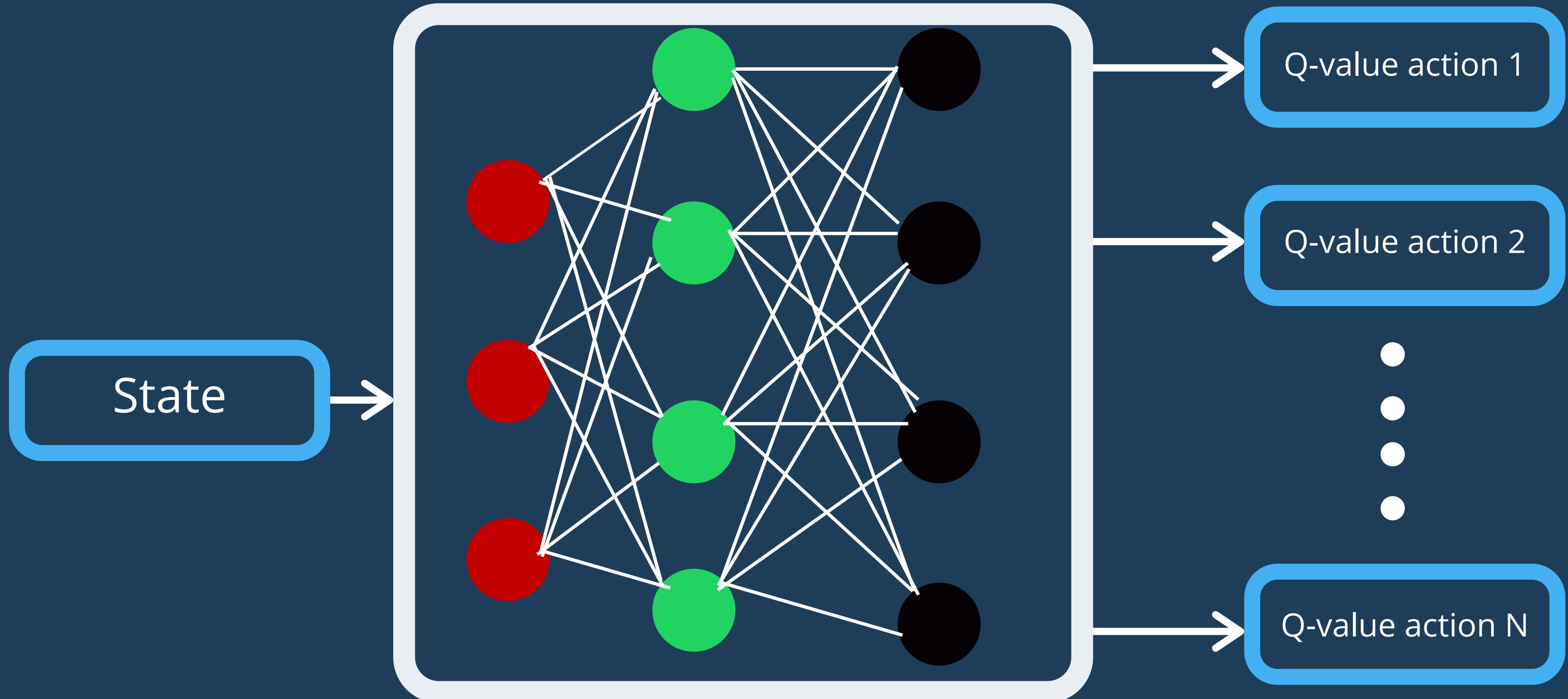
Loop over the catalogue size

2 hours for catalogue size of 1200

Deep Q-learning



FUNCTION APPROXIMATION



Deep Q-learning

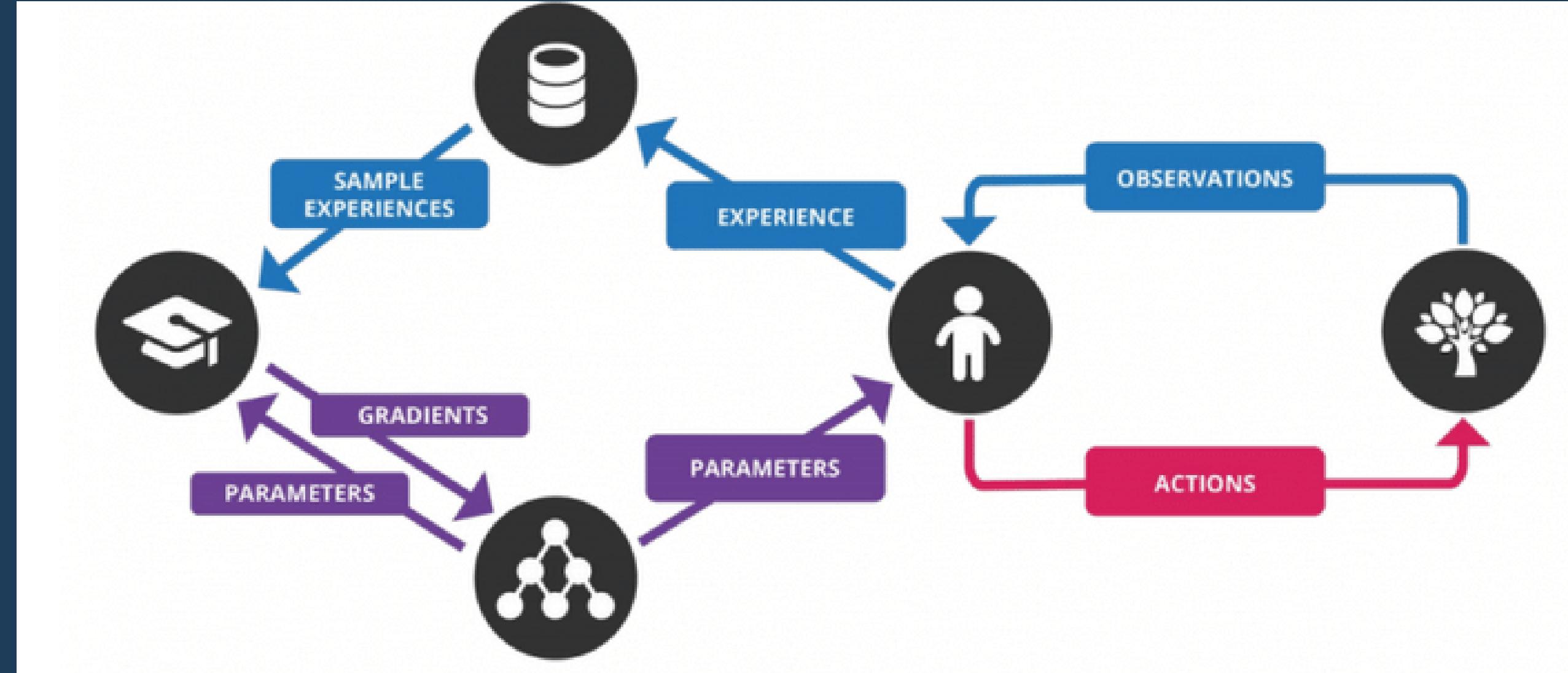
Target

$$R(S_t, A_t) + \gamma \max_a Q(S_{t+1}, a, w)$$

Loss

$$loss = (R(S_t, A_t) + \gamma \max_a Q(S_{t+1}, a, w) - Q(S_t, A_t, w))^2$$

DEEP Q-LEARNING



MEMORY REPLAY

Store the experiences to make the training process more stable

STATE REPRESENTATION



HOT ENCODING

1 for the given state, 0 otherwise

REWARDS

Replace the state by :
+2 if the content is related
and cached
+1 if the content is either
related or cached
0 otherwise

U HOT ENCODED

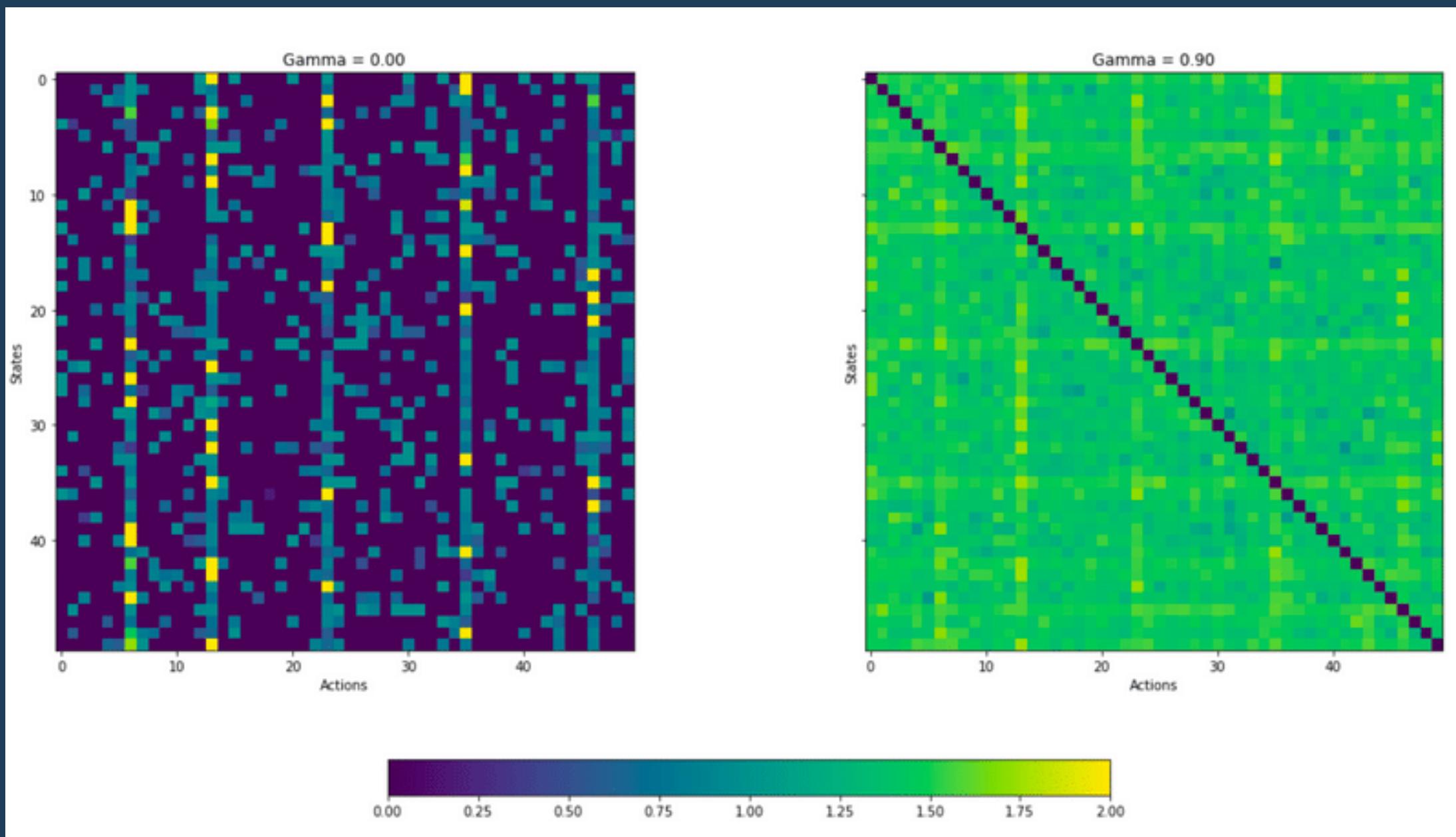
Replace the state by the 1
for every related contents

VALUABLE

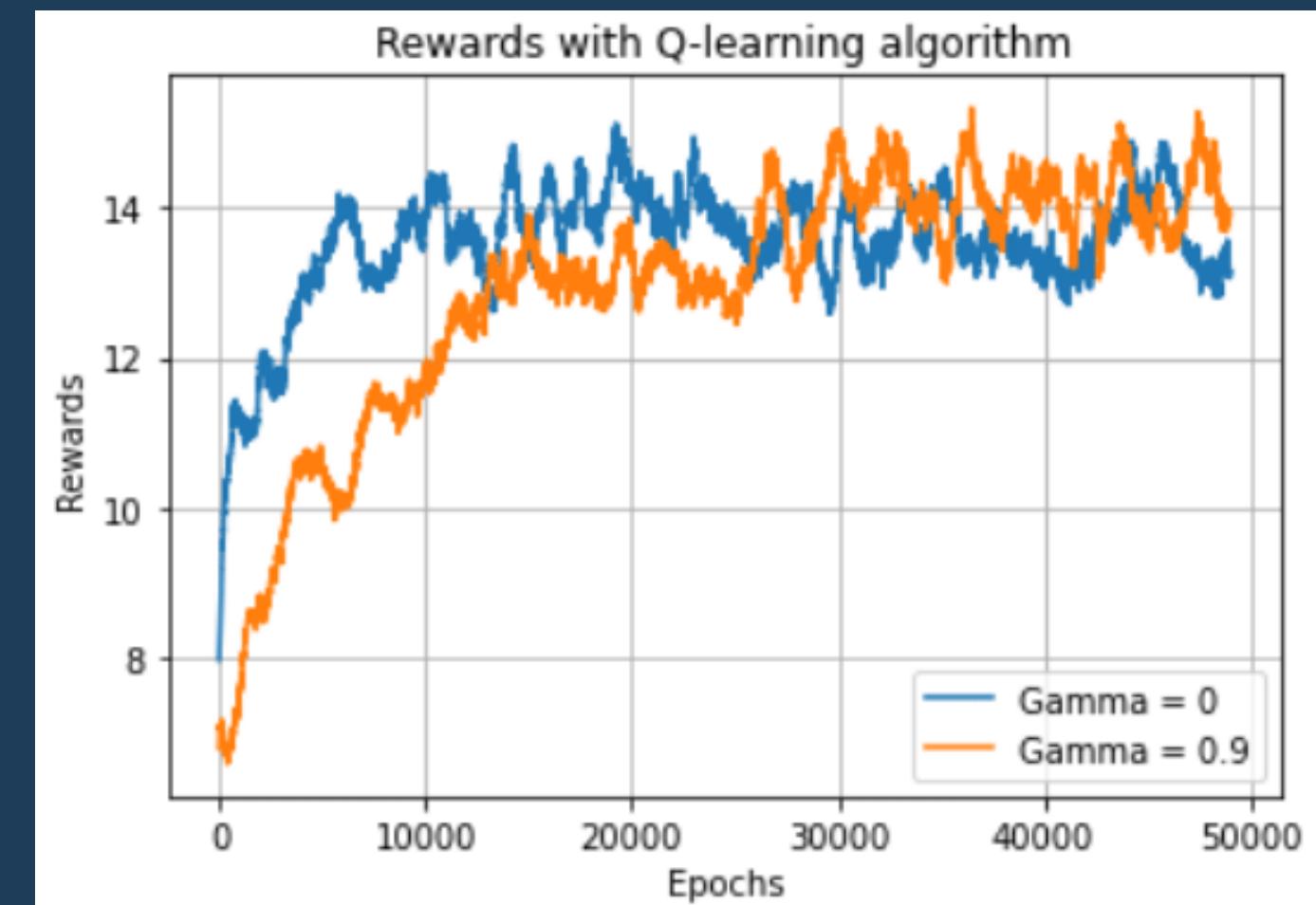
Rewards representation
with :
+1 if the next state has
contents related that are
cached

Q-learning results

To compare to



Q-tables



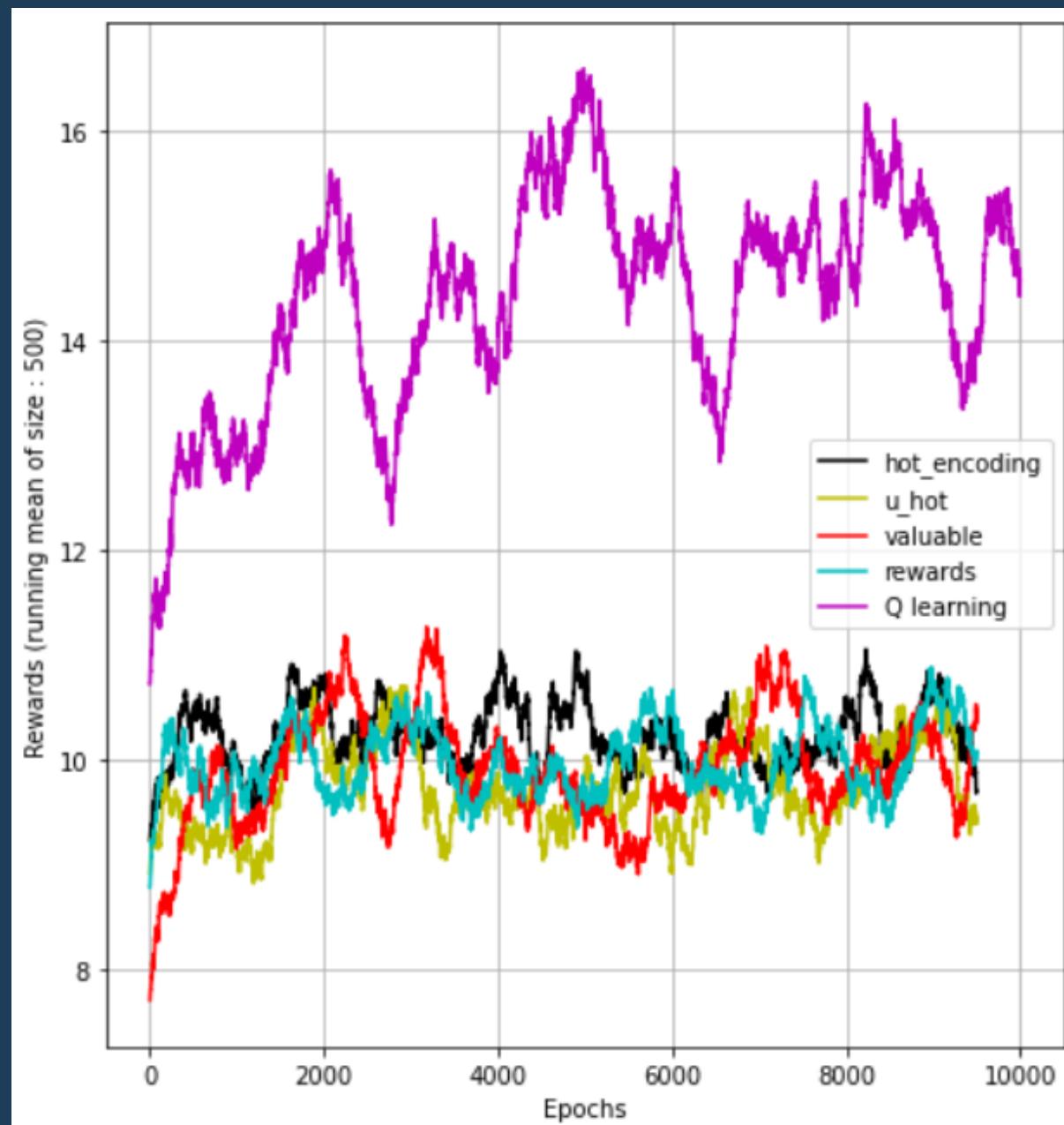
Rewards

Results

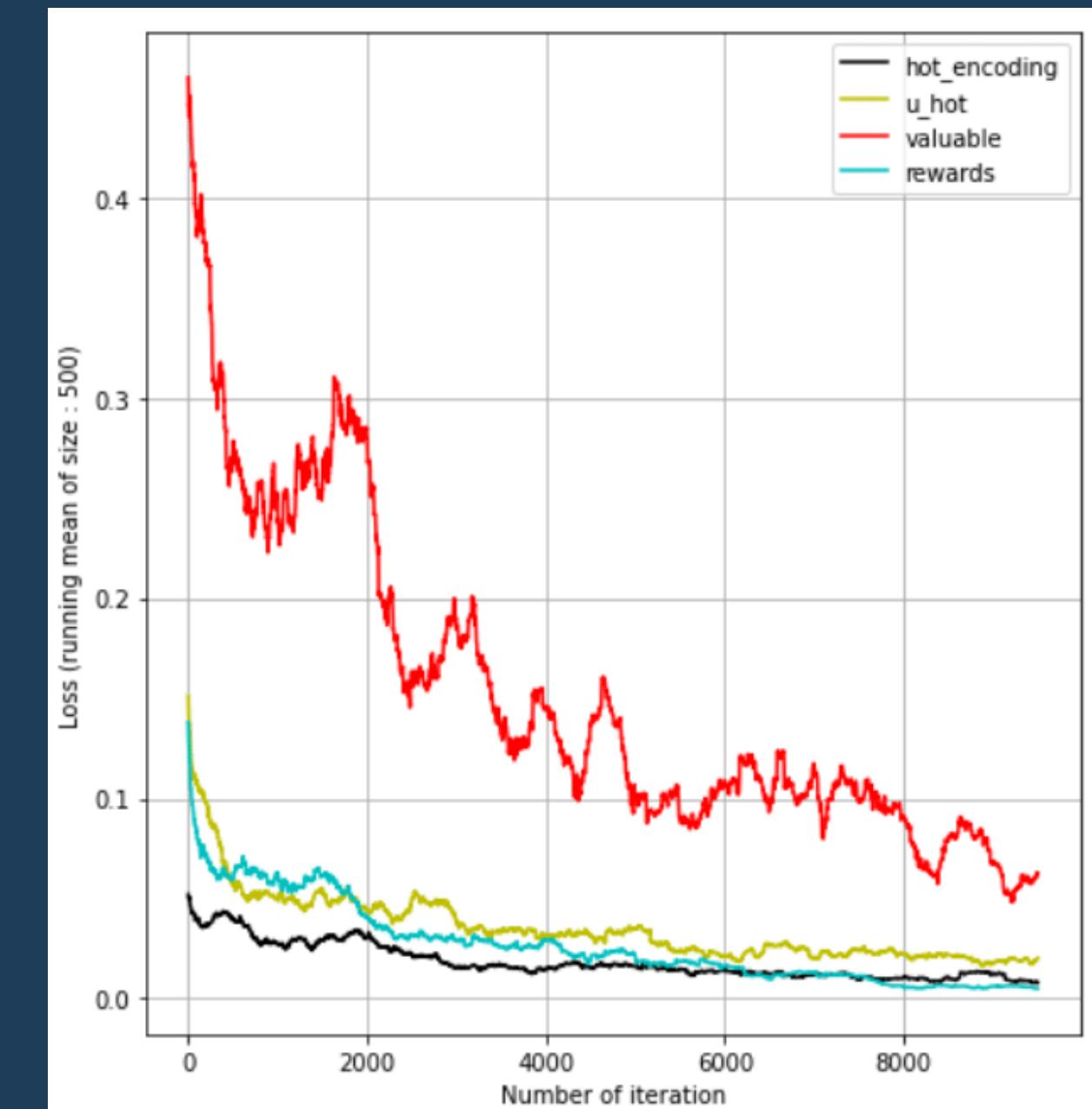
- CATALOGUE OF SIZE 50
- 10 RELATED CONTENTS BY STATE
- 5 CACHED CONTENTS
- LEARNING RATE : 1E-4
- MEMORY SIZE : 50
- BATCH-SIZE : 10
- NUMBER OF EPOCHS : 10 000

Rewards and loss for linear model

Gamma = 0



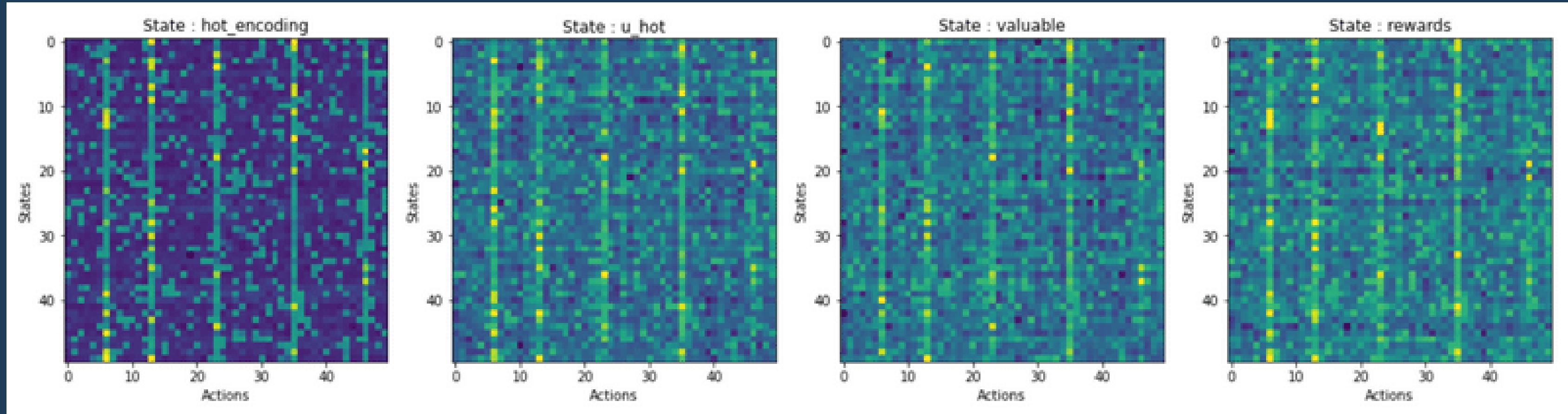
Rewards



Loss

Q-tables for linear model

Gamma = 0



Hot encoding

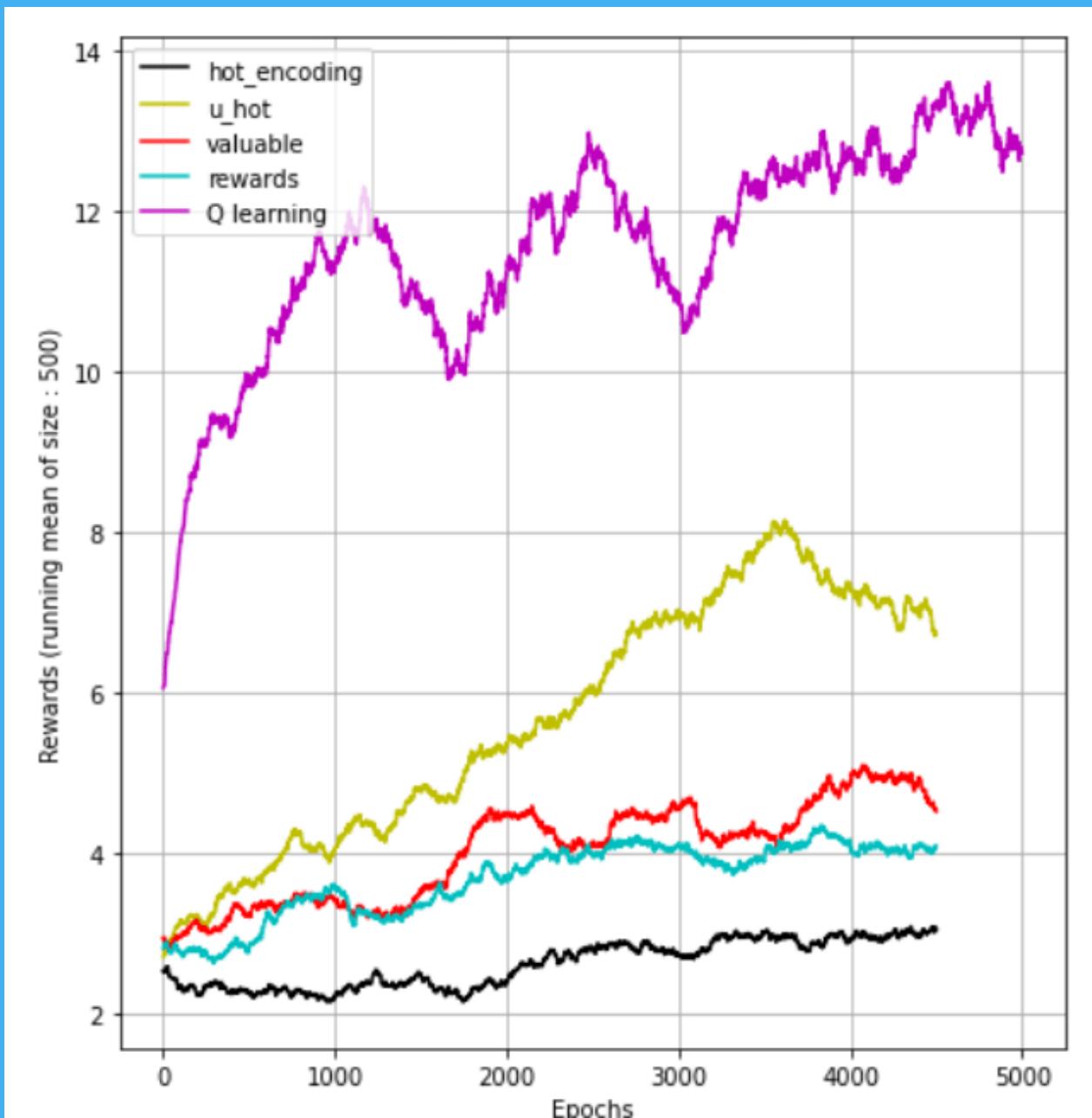
U hot encoding

Valuable

Rewards

Linear model

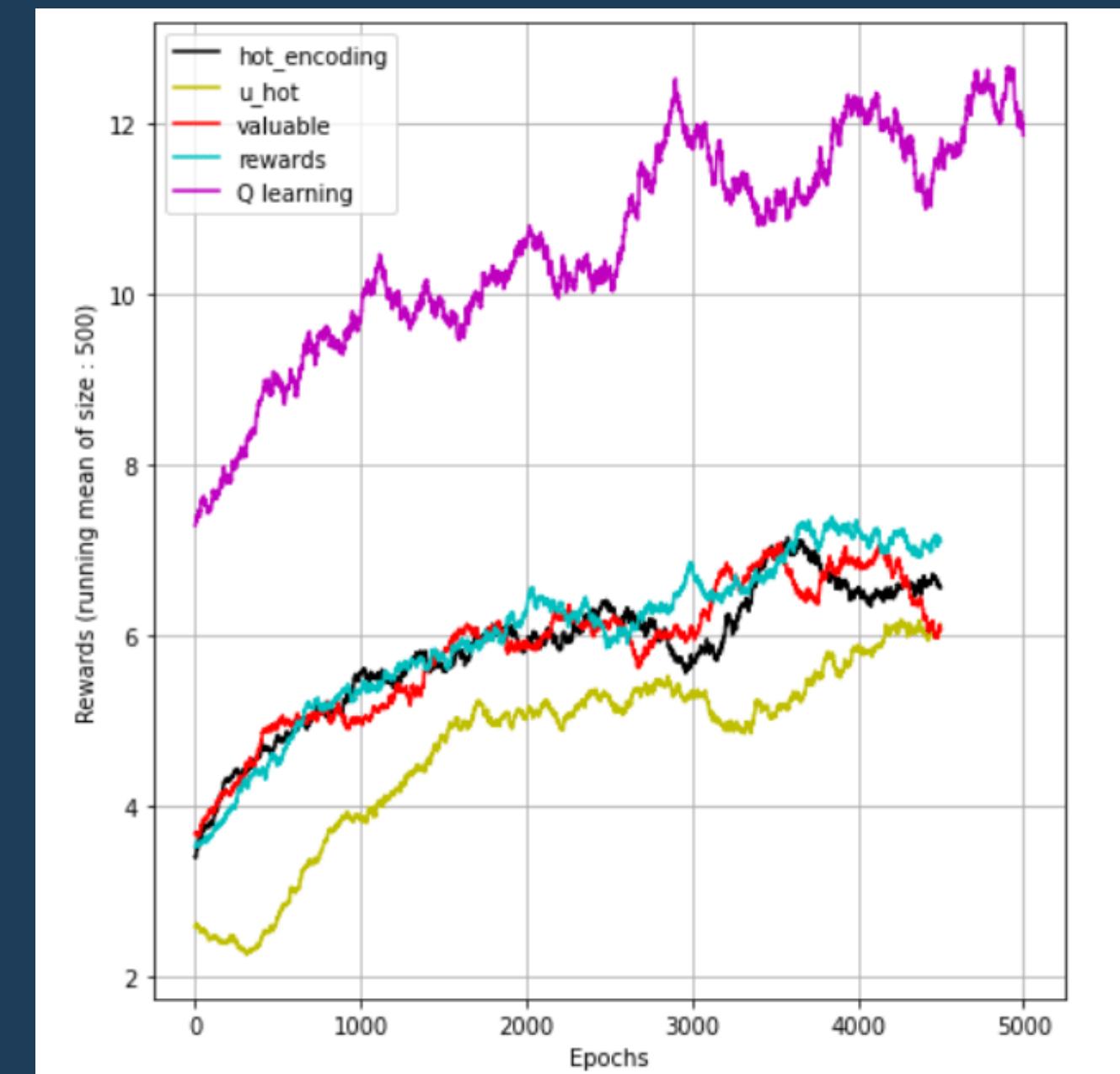
REWARDS



Gamma = 0.9

Fully connected model

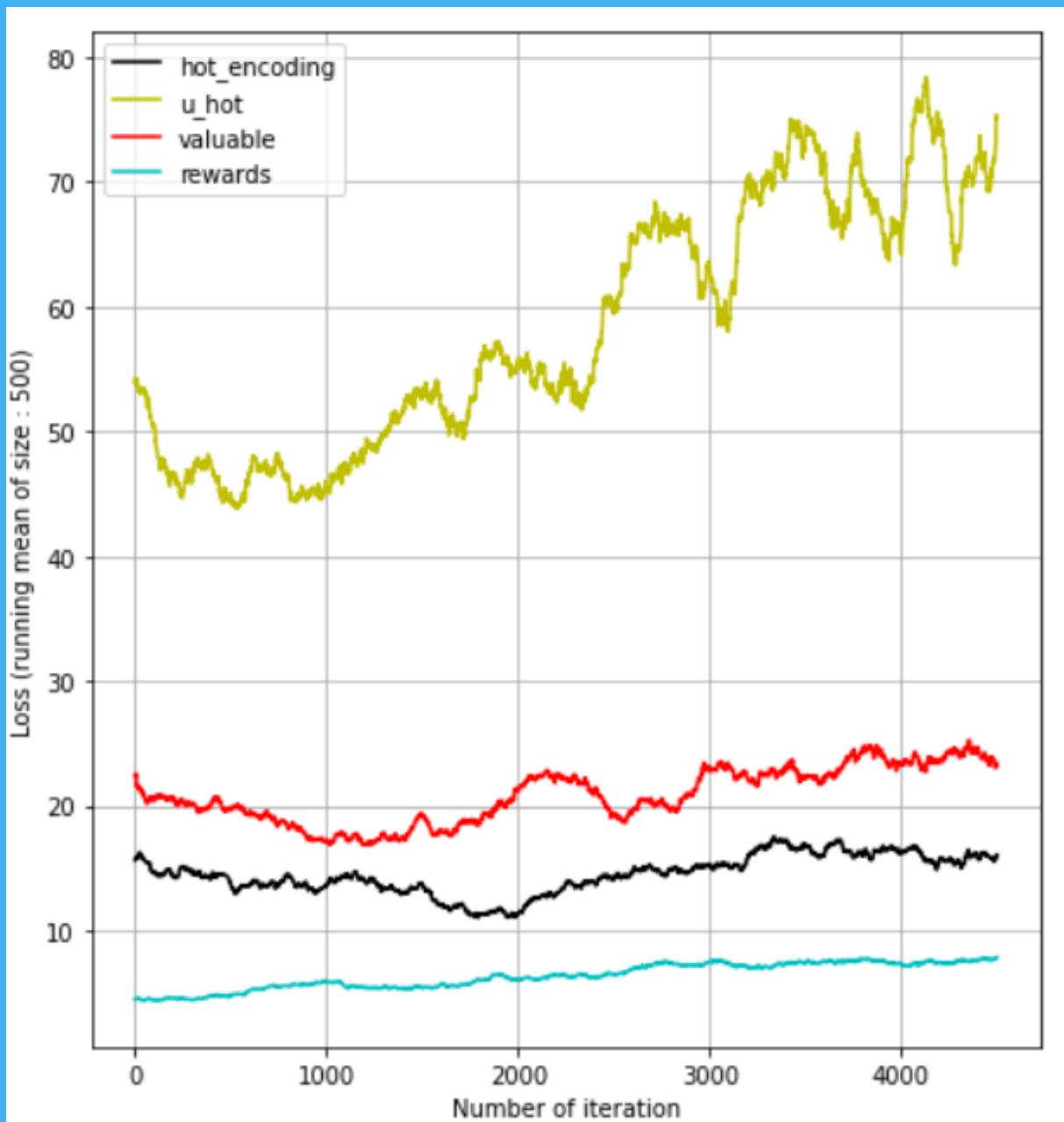
REWARDS



Gamma = 0.9

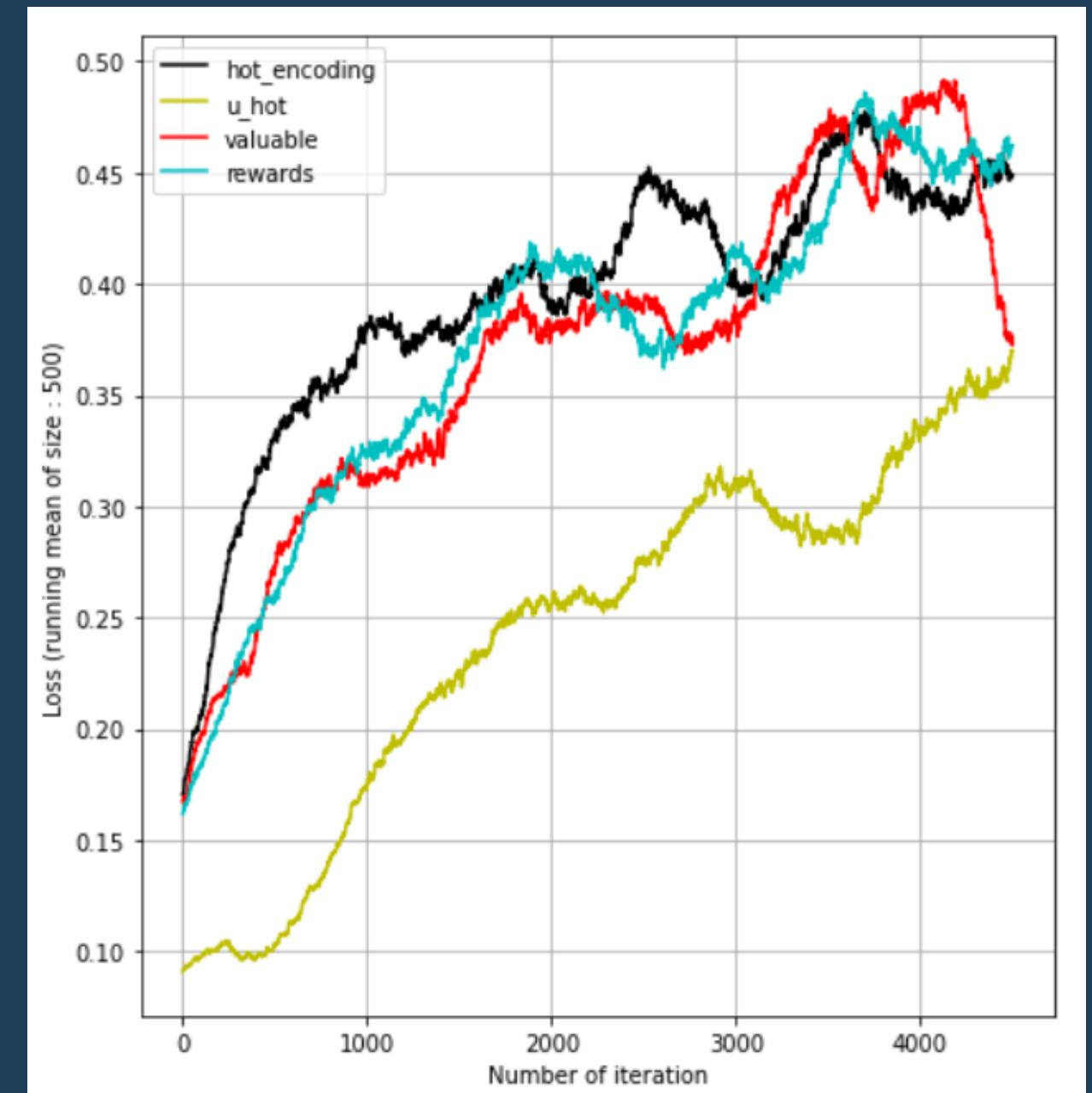
Linear model

LOSS



Fully connected model

LOSS

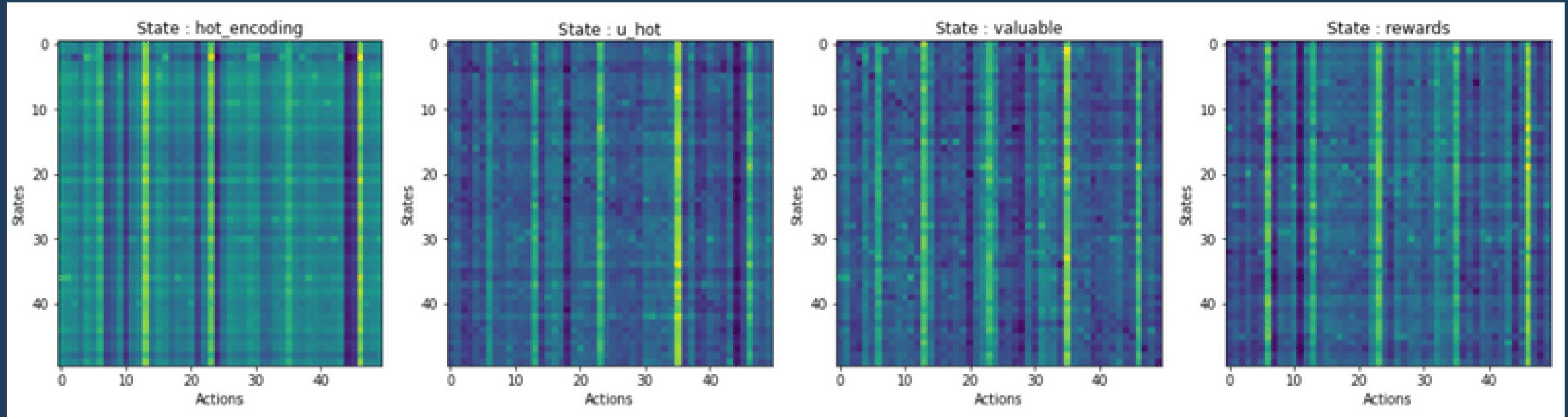


Gamma = 0.9

Gamma = 0.9

Q-tables for fully connected model

Gamma = 0 .9



Hot encoding

U hot encoding

Valuable

Rewards



Results summary

FINAL REWARDS STILL BELOW THE Q-LEARNING

But it seems the algorithm has converged

LOSS IS INCREASING FOR GAMMA = 0.9

It means it doesn't learn correctly and few errors tend to increase the loss

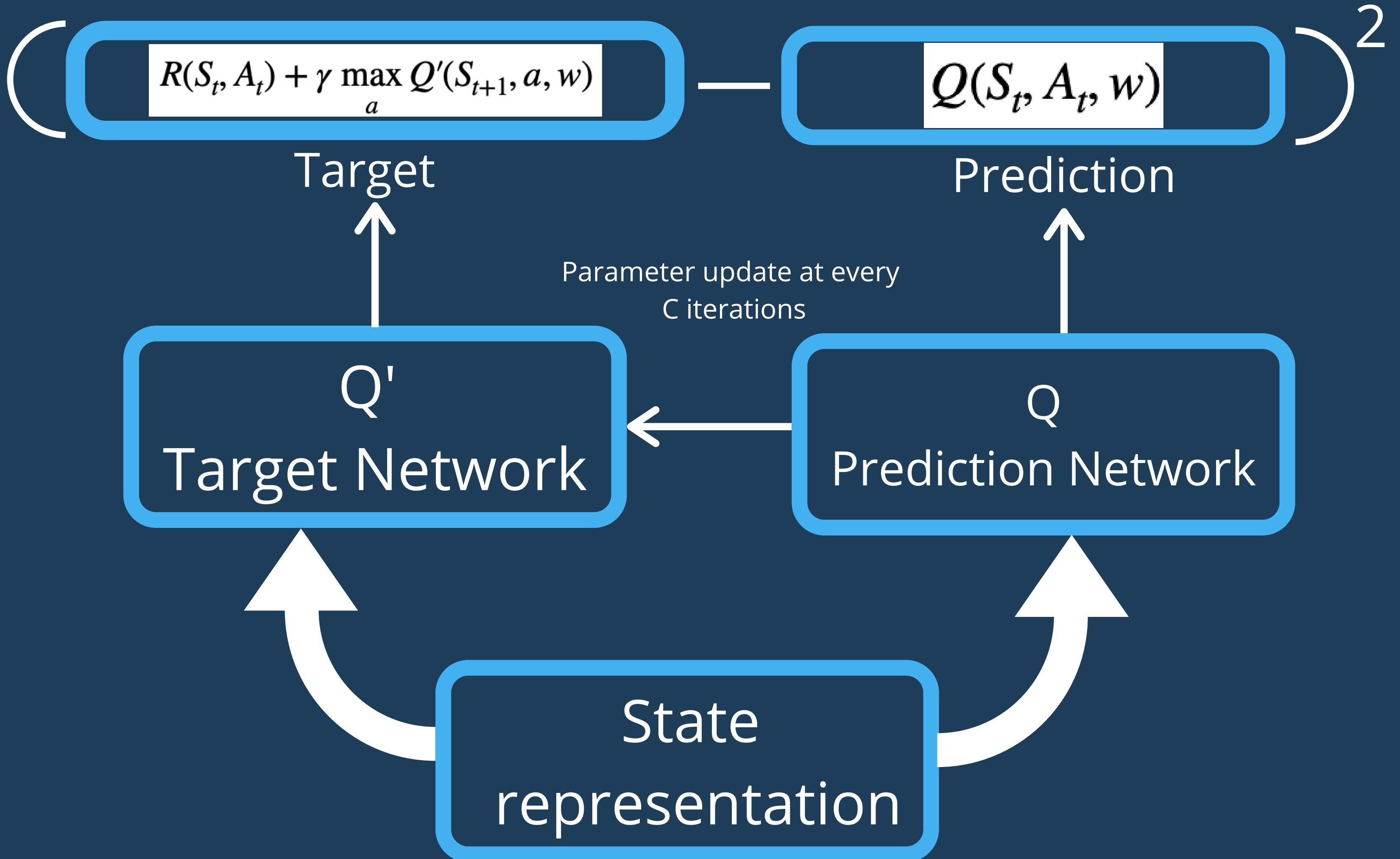
MORE A TABULAR METHOD

It doesn't seem to approximate something

Double Q-learning



IMPROVEMENTS ?



NEW STATE REPRESENTATIONS



MULTIPLE

State i is replaced by :
(a,b,c) for each action with
:
- a = 0 if action neither
cached nor related
- b = 1 if the action is
related
- c = 1 if the action is
cached

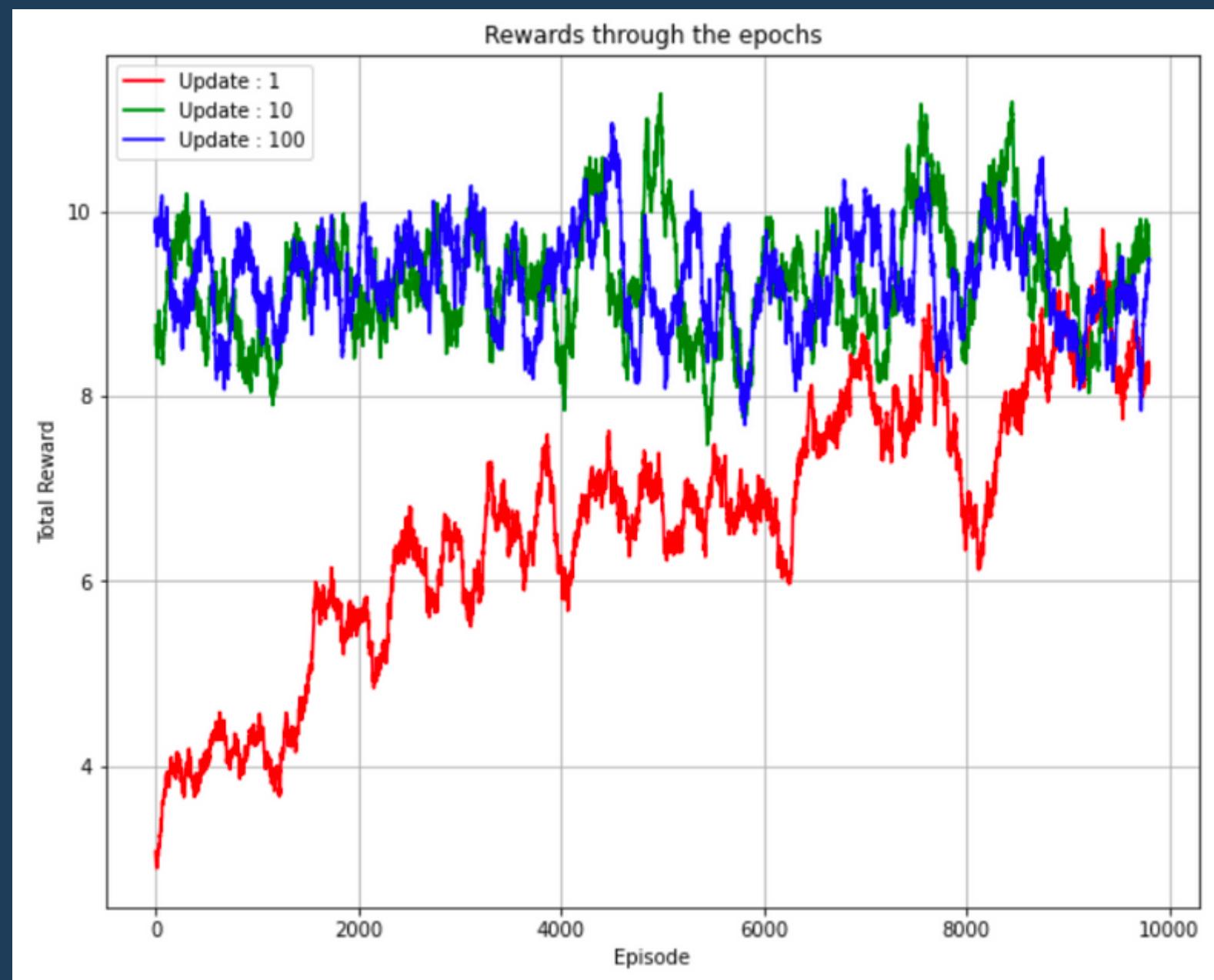
MULTIPLE VALUABLE

State i is replaced by :
(a,b,c,d) for each action
with :
-(a,b,c) like multiple
representation
-d += 1 if the next action
has contents related
cached

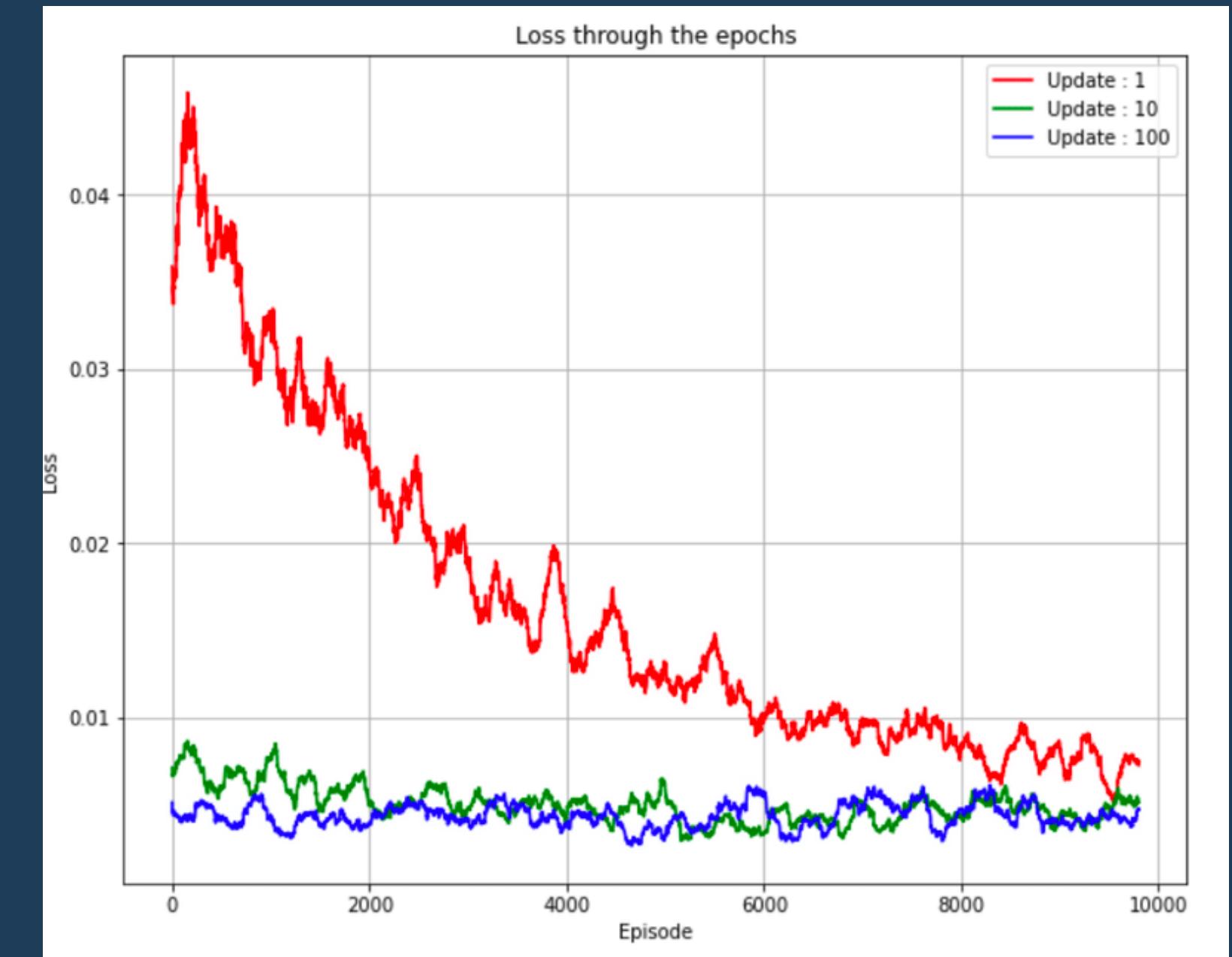
Update hyperparameter

Parameter to update the target and the prediction

Rewards



Loss

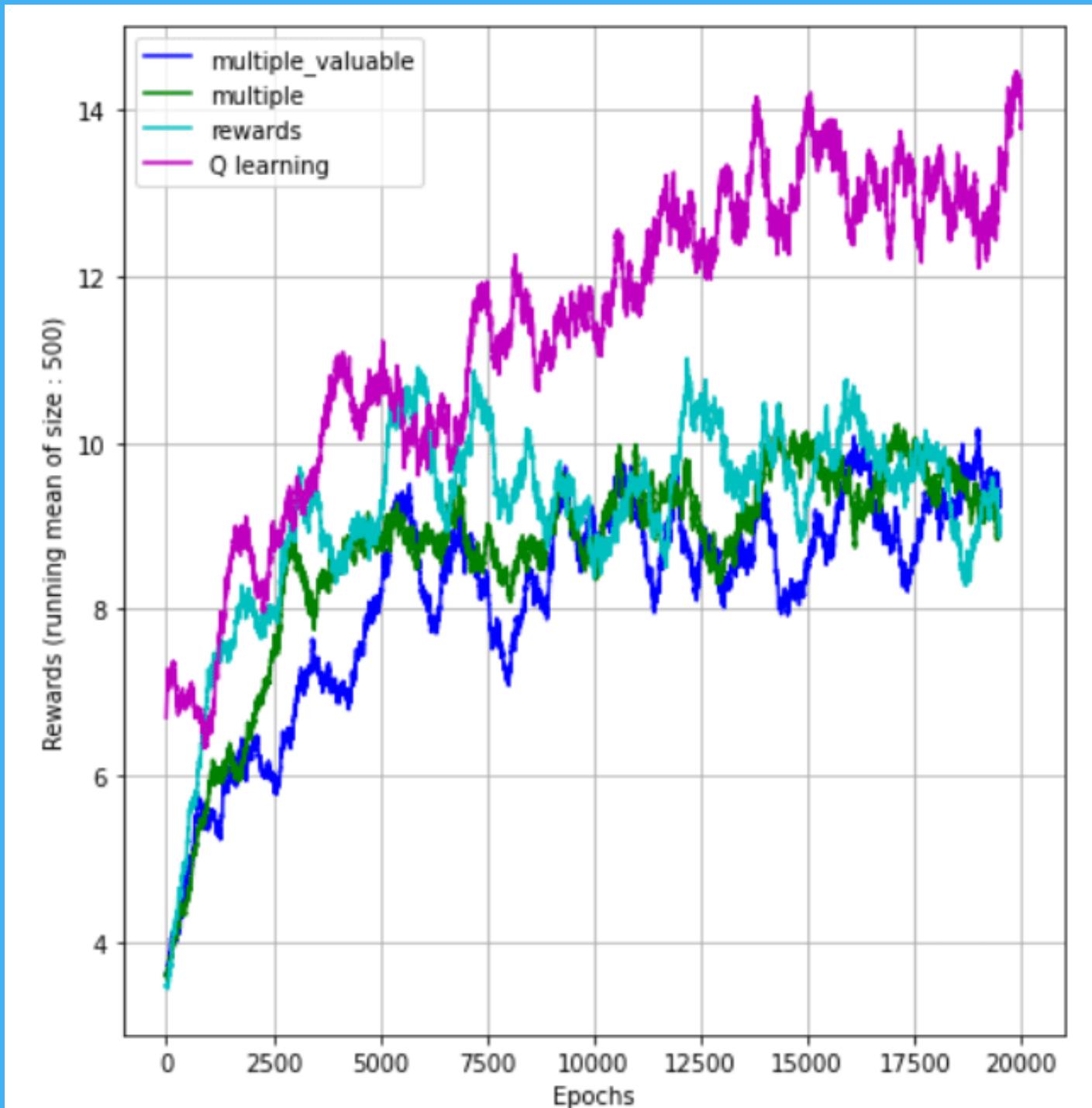


Results

- CATALOGUE OF SIZE 50
- 10 RELATED CONTENTS BY STATE
 - 5 CACHED CONTENTS
 - LEARNING RATE : 1E-4
 - MEMORY SIZE : 200
 - BATCH-SIZE : 20
- NUMBER OF EPOCHS : 20 000
- TARGET UPDATE : 50

Rewards

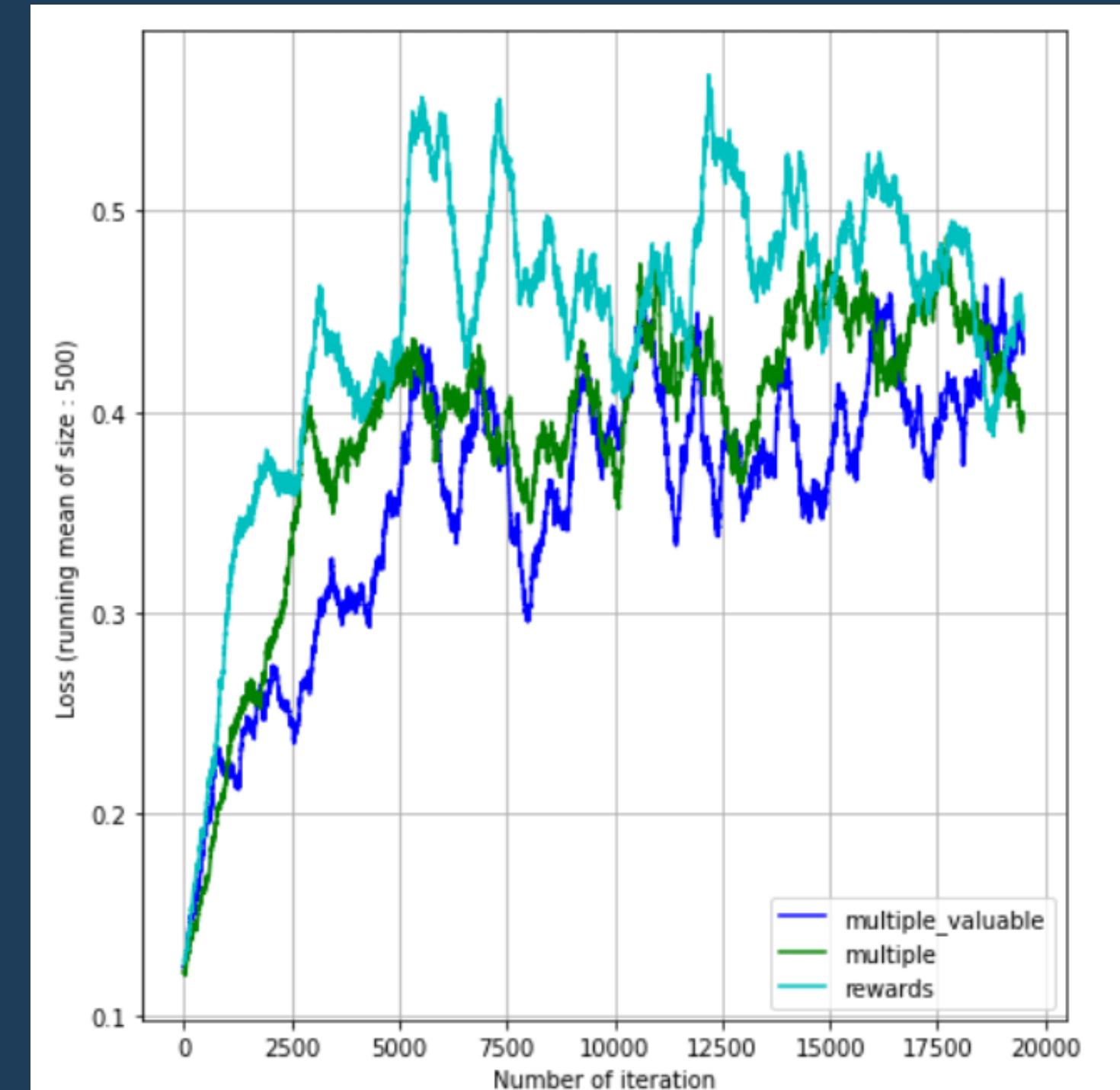
DOUBLE Q-LEARNING



Gamma = 0.9

LOSS

DOUBLE Q-LEARNING



Gamma = 0.9

Limits & Conclusions



Limits & Conclusions

- **STILL TABULAR METHODS**
It doesn't approximate enough, due to state representation

- **DOUBLE Q-LEARNING BETTER**

- The fact that the target is different from the prediction makes things better

- **REWARDS PER EPISODE**

- Still less than the Q-learning algorithm

- **CONVERGES FASTER**

- Deep Q-learning and Double Q-learning converges much faster

ADDITIONAL READING



COMPLETE REPORT

https://github.com/clementbernardd/network_friendly/blob/master/FinalReport/ReinforcementLearningFinalReport.pdf



OUR CODE IN GITHUB

https://github.com/clementbernardd/network_friendly

Group Members



CLEMENT BERNARD
Student in Data Science



JADE BONNET
Post master degree in
communication for ITS

Responsibles

**SPYROPOULOS
THRASYVOULOS**

GIANNAKAS THEODOROS