Half Field Offense

In this experiment I wanted to understand how is the agent exploring the different states, so I ended up looking into the features again. I was using 5 discrete features, however, I found out that my agent did not use the direccion, so I only used 4 features. Still used a hand-made goalkeeper (behavior explained bellow).

1. Game parameters:

- Number of **teammates: 0;**
- Number of **opponents: 1;** (Dumb Goalie)
- Number of **episodes: 3000 train;**

2. Q-Learning Agent

2.1. Q learning parameters:

- Learning rate: 0.10;
- Epsilon: 1.0; Epsilon decrescent: 0.999; Epsilon end: 0.01;
- Discount factor: 0.99;
- **Q learning table dim: 48** environment features * **3** number of actions;

2.2. State Features

1. **Position** – int [0, 6], subdivided the field in 6 regions, each integer signalizes one of these regions;
2. **Has Ball** – int (0,1). 1 if agent can kick, else 0.
3. **Should Shoot** – int (0,1). Agent's goal opening angle > 20%;
4. **Opponent is close** – int (0,1). Opponent is close;

2.3. Actions

1. **Move -** Re-positions the agent according to the strategy given by Agent2D. The Move command works only when the agent does not have the ball. If the agent has the ball, another command such as Dribble, Shoot, or Pass should be used.

2. **Dribble -** Advances the ball towards the goal using a combination of short kicks and moves.
3. **Shoot -** Executes the best available shot. This command only works when the agent has the ball.

2.4. Rewards

- - 500 – the game ends without scoring goal;
- + 1000 – score goal;
- - 1 – for each time-step, the idea was to motivate the agent to goal as fastest as possible;

3. Goalkeeper Agent

The agent presents 2 simple behavior:

If ball is on the upper side of the field:

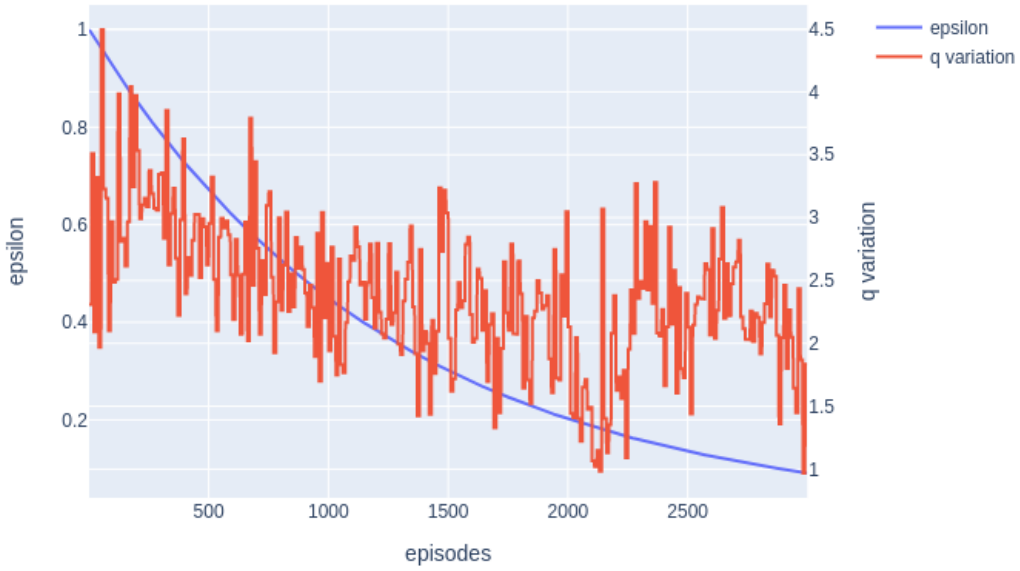
the goalkeeper moves to the goal top corner;

Else:

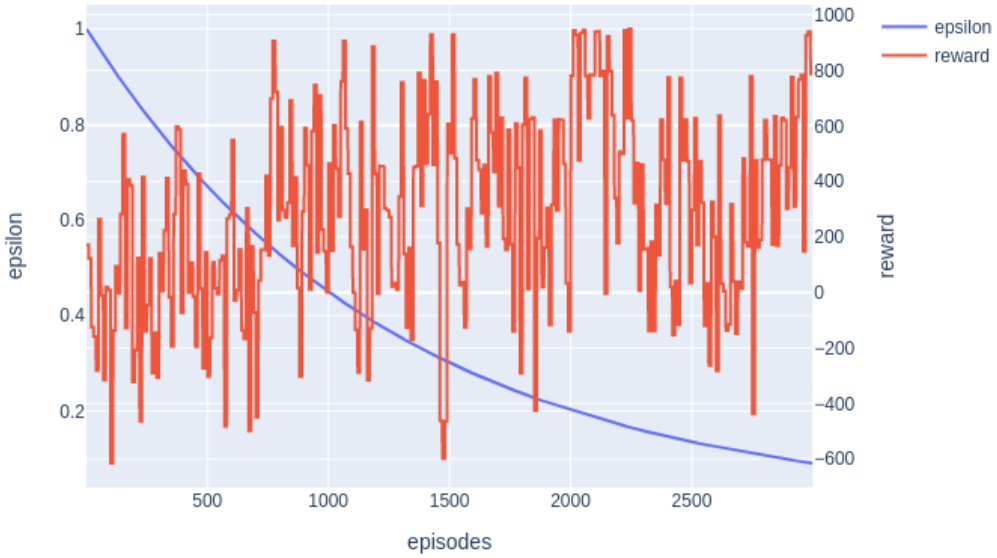
the goalkeeper moves to the goal bottom corner;

4. Results:

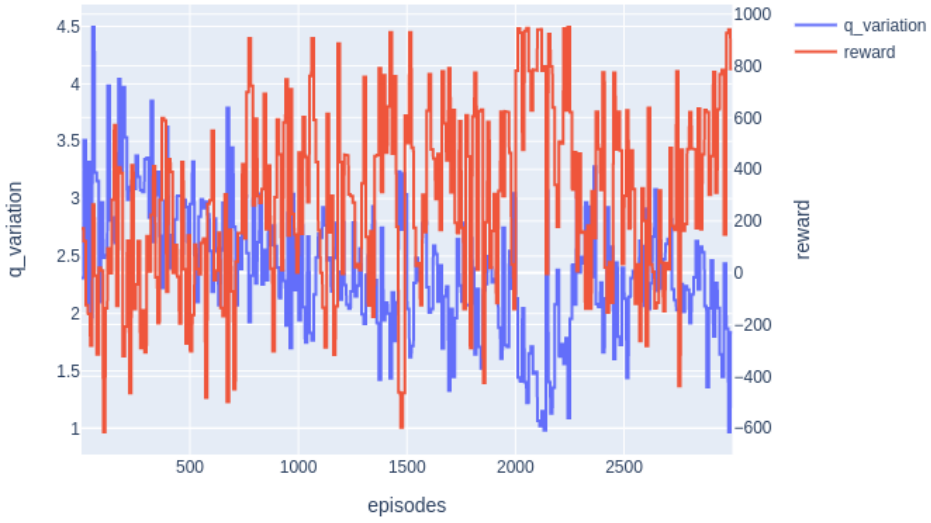
4.1. Relation q-learning variation – epsilon variation:



4.2. Relation epsilon variation – reward variation:

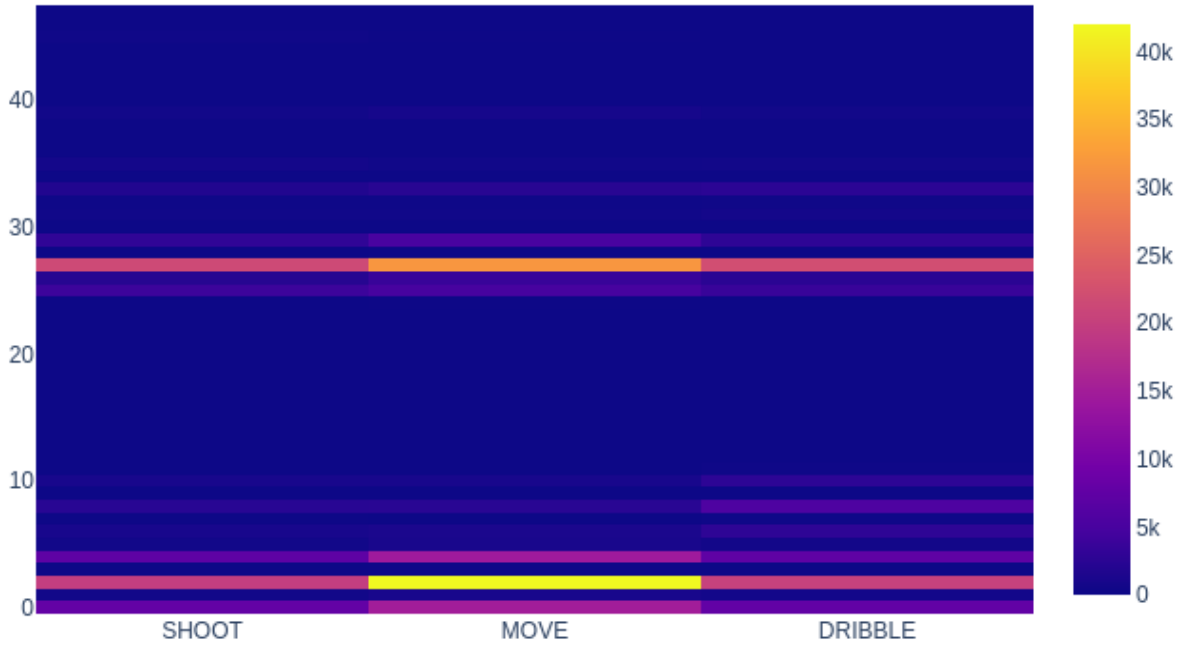


4.3. Relation q-table variation – reward variation:



4.3. Exploration Heat-map:

On the y-axis is the 48 different states, on the x-axis bar are the 3 different actions. This sums the total actions per selected at each state, during all the training.



5. Results Discussion:

This experiment shows that my agent does not explores the spaces and actions correctly. I doubled checked if the features were being correctly extracted. So this leads me to believe that my rewards and my features are not good enough.

I know that I should simplify the problem as much as possible, however I feel that my actions are too high level. What I mean by that, is that the MOVE and DRIBBLE actions have already much intelligence in them, that is part of the reason why the agent does not explore some states. These actions guide the agent straight into the goal. And so the agent does not explores the other states around.