Half Field Offense

Q-agent with one teammate

During this first experiment with one teammate, the idea was to let the agent to first interact with one teammate.

Challenge: The world is more complex and more dynamic than before. How can our agent learn in such unstable environment?

Solution: Let the agent learn the rules of the game playing solo, then introduce the teammate. So the agent trained solo for 10k episodes, reaching an accuracy of 88%. Then we retrained the agent with a fixed policy teammate for 10k episodes.

1. New Features

Solo Features	
Agent position	0: "TOP LEFT", 1: "TOP RIGHT", 2: "MID LEFT", 3: "MID RIGHT", 4: "BOTTOM LEFT", 5: "BOTTOM RIGHT"
Open angle to goal?	1: "open angle", 0: "closed angle"
Opponent is near?	1: "opponent close" 0: "opponent far"
Ball position	0: "Player Has Ball", 1: "Ball on Up", 2: "Ball on Right", 3: "Ball on Down", 4: "Ball on Left"

→

Team Features	
Agent position	0: "TOP LEFT", 1: "TOP RIGHT", 2: "MID LEFT", 3: "MID RIGHT", 4: "BOTTOM LEFT", 5: "BOTTOM RIGHT"
Teammate further from goal?	0: "teammate near goal", 1: "teammate further goal"
Agent has an open angle to goal?	1: "open angle", 0: "closed angle"
Teammate has an open angle to goal?	1: "open angle", 0: "closed angle"
Ball position	0: "Player Has Ball", 1: "Teammate Has Ball", 2: "Ball on Up", 2: "Ball on Right", 3: "Ball on Down", 4: "Ball on Left"

Fig 1 – The average results of 50 games after x episodes of training. The shaded part is the variance. These were the average results of 6 training processes.

2. New Actions

Solo Actions	
Has Ball	No ball
KICK_TO_GOAL	DO NOTHING
LONG_DRIBBLE_UP	LONG_MOVE_UP
LONG_DRIBBLE_DOWN	LONG_MOVE_DOWN
LONG_DRIBBLE_LEFT	LONG_MOVE_LEFT
LONG_DRIBBLE_RIGHT	LONG_MOVE_RIGHT
SHORT_DRIBBLE_UP	SHORT_MOVE_UP
SHORT_DRIBBLE_DOWN	SHORT_MOVE_DOWN
SHORT_DRIBBLE_LEFT	SHORT_MOVE_LEFT
SHORT_DRIBBLE_RIGHT	SHORT_MOVE_RIGHT

→

Team Features	
Has Ball	No ball
KICK_TO_GOAL	DO NOTHING
PASS BALL TO TEAMMATE	**DO NOTHING**
LONG_DRIBBLE_UP	LONG_MOVE_UP
LONG_DRIBBLE_DOWN	LONG_MOVE_DOWN
LONG_DRIBBLE_LEFT	LONG_MOVE_LEFT
LONG_DRIBBLE_RIGHT	LONG_MOVE_RIGHT
SHORT_DRIBBLE_UP	SHORT_MOVE_UP
SHORT_DRIBBLE_DOWN	SHORT_MOVE_DOWN
SHORT_DRIBBLE_LEFT	SHORT_MOVE_LEFT
SHORT_DRIBBLE_RIGHT	SHORT_MOVE_RIGHT

Fig 1 – The average results of 50 games after x episodes of training. The shaded part is the variance. These were the average results of 6 training processes.

3. Re-train process

In the previous experiments we were able to create a stable q-agent, able to play solo, reaching 80-95% accuracies. So now we wanted to introduce a teammate. However, it wasn't straight forward. So we divided the training into 2 parts:

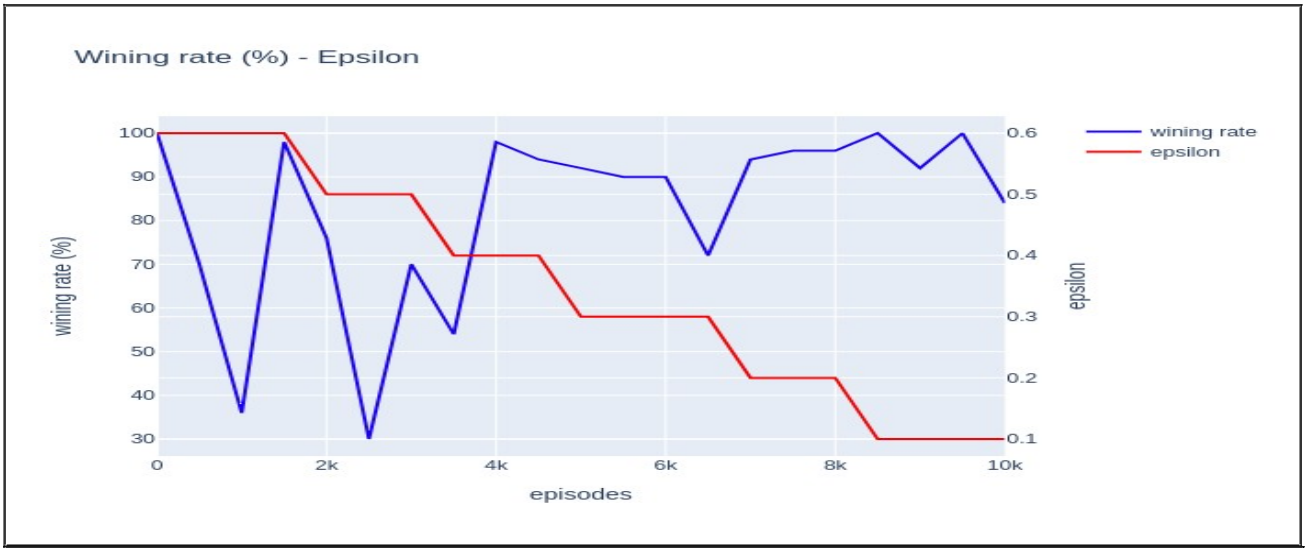
3.1 Train with a Static Teammate

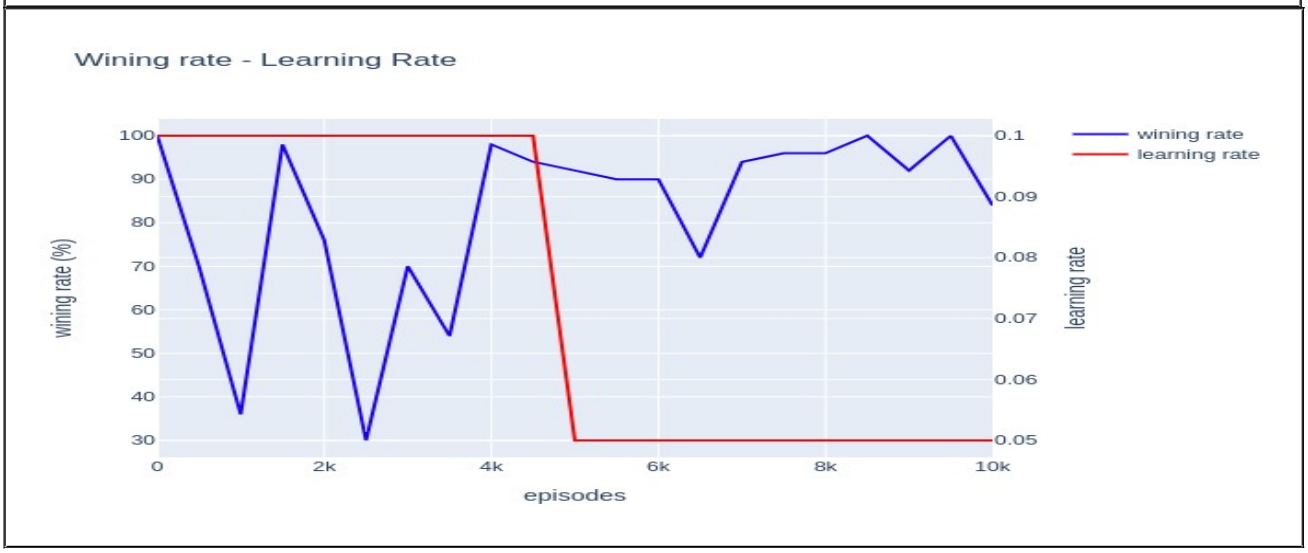
In the first part, the agent played with a static teammate, using the new features and the new actions. The idea was to simulate a solo game, in order to the agent to learn how to navigate and score in the environment. The agent reached an accuracy of 88% after 10k episodes of training, which I considered as good basis for the second part of the train.

3.1 Train with a Fixed Policy Teammate

In the second part, we used the previous model and introduced a fixed policy teammate, created by me. The agent trained for 10k episodes, reaching an accuracy of 84%. However, the agent did not learnt any kind of cooperation. The plots bellow show how the agent evolved during the re-train process:


Wining rate (%) - Q-Table Variation (%)
wining rate
q-table variation
wining rate (%)
q-table variation (%)
episodes
100
90
80
70
60
50
40
30
4
3
2
1
0
0
2k
4k
6k
8k
10k


Wining rate (%) - Epsilon
wining rate
epsilon
wining rate (%)
epsilon
episodes
100
90
80
70
60
50
40
30
0.6
0.5
0.4
0.3
0.2
0.1
0
2k
4k
6k
8k
10k


Wining rate - Learning Rate
wining rate
learning rate
wining rate (%)
learning rate
episodes
100
90
80
70
60
50
40
30
0.1
0.09
0.08
0.07
0.06
0.05
0
2k
4k
6k
8k
10k

Game / Environment Setup

1. Game parameters:

- Number of **teammates: 1; (**Static Teammate or Fixed Policy Teammate**)**
- Number of **opponents: 1;** (Dumb Goalie)
- Number of **episodes: 10k train;**

2. Q-Learning Agent

2.1. Q learning parameters:

- Learning rate: [0.10, 0.05];
- Epsilon values: [0.6, 0.5, 0.4, 0.3, 0.2, 0.1];
- Discount factor: 0.9;
- **Q learning table dim: 240** environment states * **10** number of actions

2.2. Rewards

- - 1 – the game ends without scoring goal;
- + 1 – score goal;

2.3. Behavior

In the beginning of the game one of the **six starting positions is randomly choosen**. So in the beginning of the game, the a**gent go gets the ball** (even take it from the teammate), and goes to the starting position. Then he c**ommunicates "READY"**, which means that it started exploring-exploiting;

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. Static Teammate

If has ball:

Pass the ball to teammate;

Else:

Do Nothing;

5. Fixed Policy Teammate

While agent is not in the start position:

Wait;

Else:

If **has ball**:

If **teammate further from goal**:

If **in goal region:**:

→ *SHOOT*;

Else:

→ *DRIBBLES* to goal region;

Else:

→ *PASS* the ball to teammate;

Elif **teammate has ball**:

→ *MOVES* to a corner of the goal region;

Else: (**no one has the ball**)

If **teammate further from goal**:

→ *MOVES* to the ball - go get the ball;

Else:

→ *MOVES* to a corner of the goal region;