Presentation

July 9, 2018

```
In [1]: from som import SOM
        import csv
        import networkx as nx
        import numpy as np
        import matplotlib as mpl
        from matplotlib import pyplot
        from mpl_toolkits.mplot3d import Axes3D
        from IPython import display
```

1 GameAI

1.1 Project 03 - Behavior programming

2 ToDo

- 3.1 - Connect4 on a large board (19x19)
- 3.2 - Fuzzy Breakout controller
- 3.3 - self organizing maps to represent player movements
- 3.4 - Bayesian imitation learning

3 3.1 - Connect4 on a large board (19x19)

- Winrate stays the same
- Huge increase in Runtime (3 hrs. vs. 24 min.)

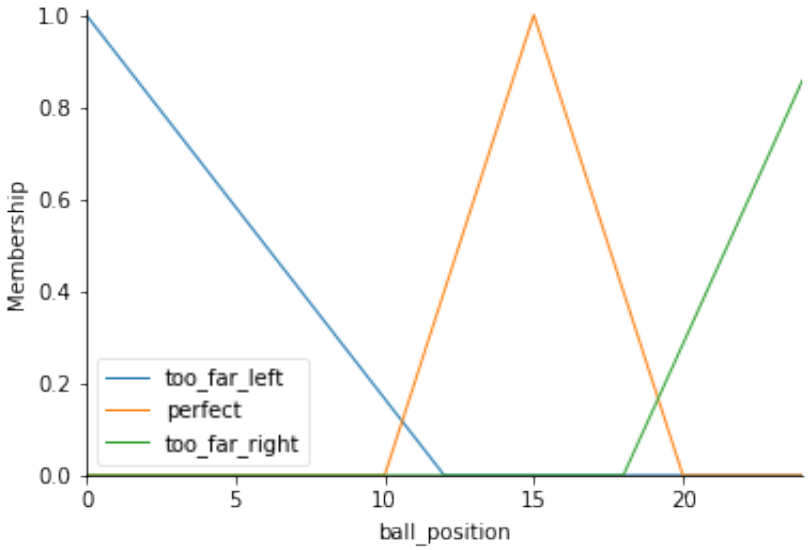
4 3.2 - Fuzzy Breakout controller

```
In [ ]: ball_position = ctrl.Antecedent(np.arange(0, 800, 1), 'ball_position')
        movement = ctrl.Consequent(np.arange(-1, 2, 1), 'movement')
```

```
In [ ]: ball_position['perfect'] = fuzz.trimf(ball_position.universe,
                                              [
                                                  player.rect.x,
                                                  player.rect.center[0],
                                                  player.rect.x + player.width
```

```
                                                    ]
        movement['left'] = fuzz.trimf(movement.universe, [-1, -1, 0])
```

```
In [ ]: paddle_x = 10
        paddle_w = 10
```



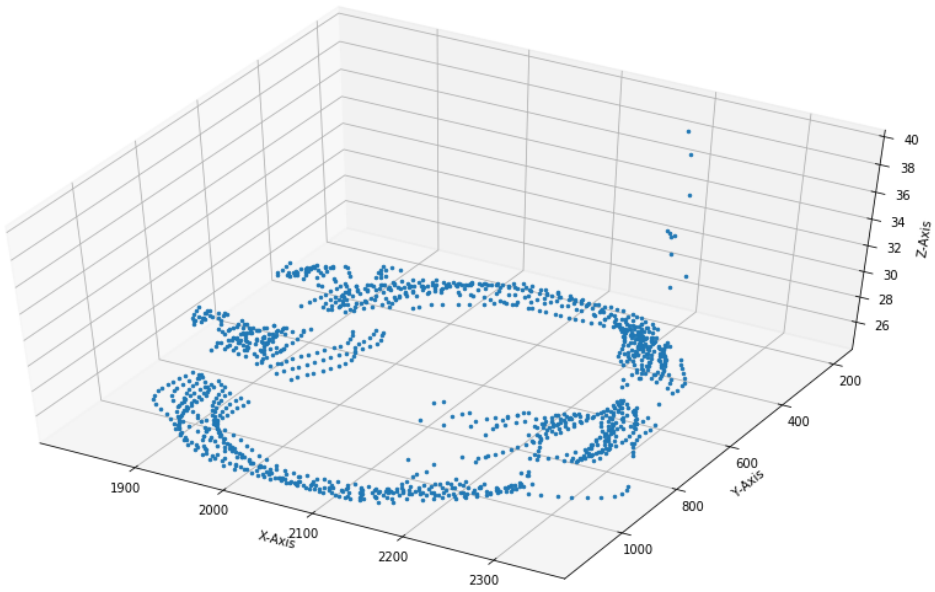
fuzzy

5 Result

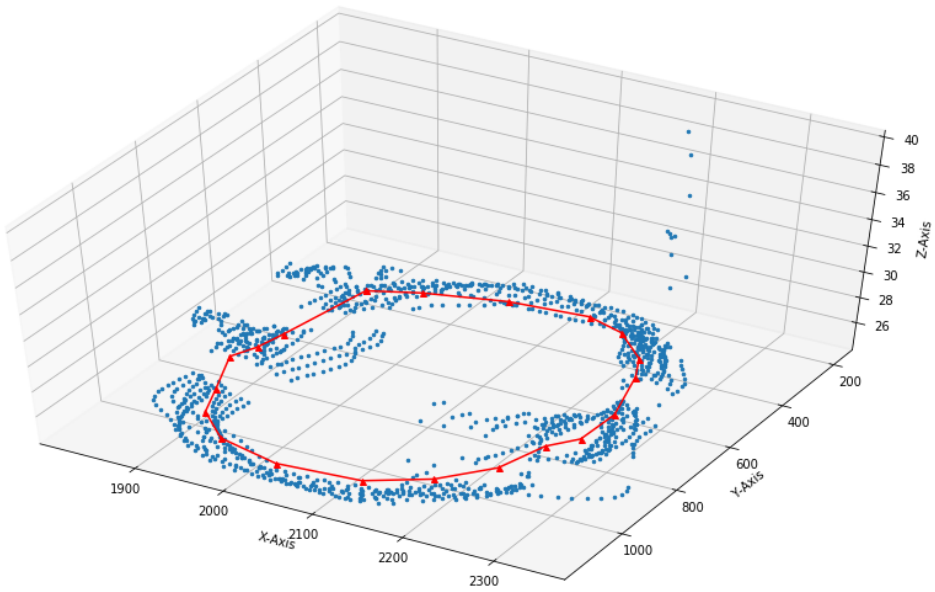
- Paddle moves smooth from left to right.
- Increasing speed gives the same problems as before.

6 3.3 - SOM

```
In [2]: s = SOM('project_03/q3dm1-path1.csv', number_of_nodes=20)
        s.plot_all()
```

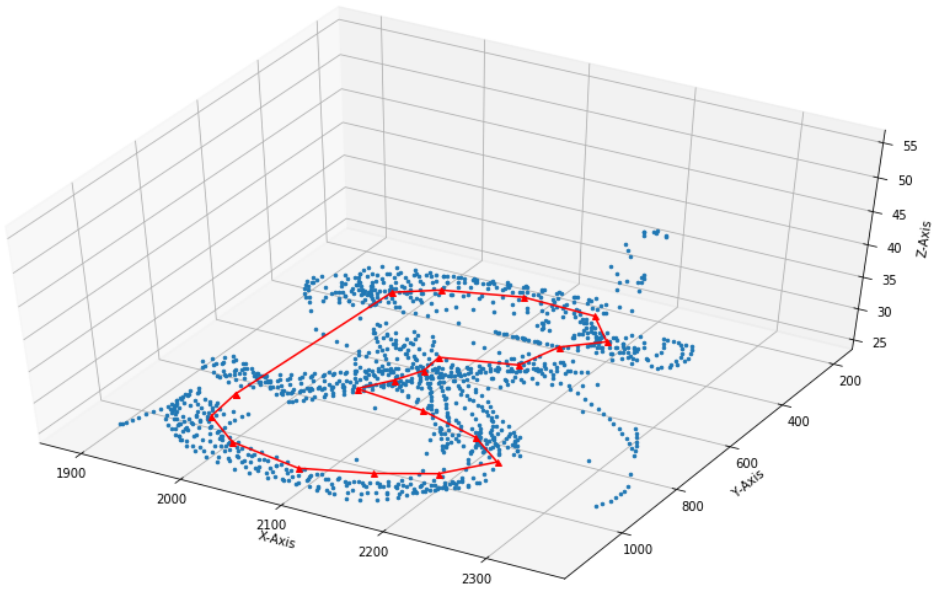


In [3]:
```
s.train(t_max=5000)
```



Out[3]:
```
<networkx.classes.graph.Graph at 0x7fb72fff1a90>
```

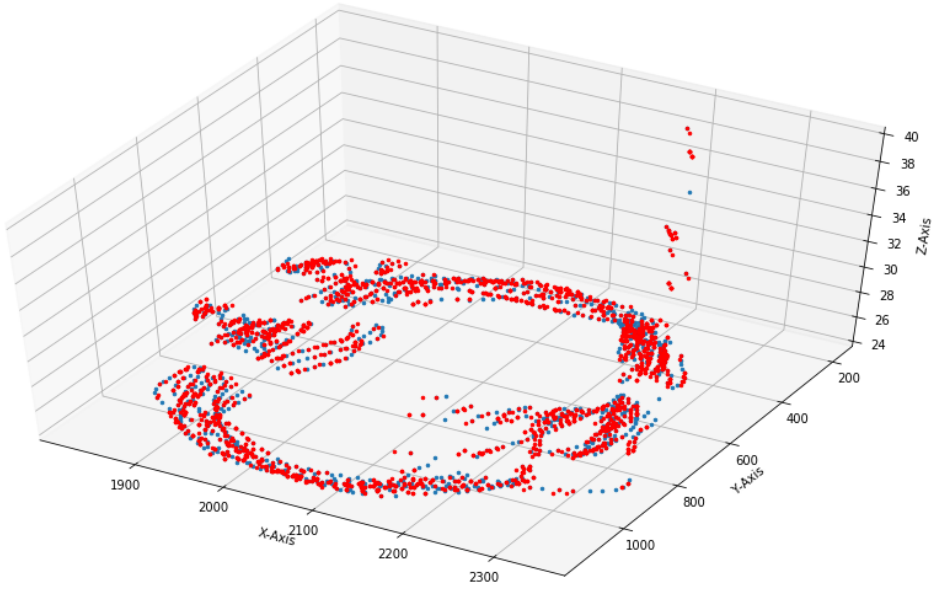
In [4]:
```
s2 = SOM('project_03/q3dm1-path2.csv', number_of_nodes=20)
s2.train(t_max=5000)
```



Out[4]: <networkx.classes.graph.Graph at 0x7fb72feae3c8>

7 3.4 - Bayesian imitation learning

In [6]:
```
t = s.bayesian_imitation()
s.plot_all(overlap=t)
```



In [7]:
```
t2 = s2.bayesian_imitation()
s2.plot_all(overlap=t2)
```