AI & Automation Exam 1

04/04/2022

Mads Hegewald (EAAMHEG)

# Exercise 1

When exercise1 is run it will automatically start with motion detection. To change to full body detection press ‘p’, for face recognition press ‘f’ and to go back to motion detection press ‘m’. The detection mode can also be seen in the window name at the top left.

Motion detection

Pedestrian detection

cascades

Face detection

cascades

# Exercise 2

## a)

Write as separate parts?

Chart, line chart

Description automatically generated

The above experiment was run with the settings of 2 hidden layers with 20 and 10 neurons. The number of points used was 2500 and was run for 50 epochs. I used sigmoid as activation function in the hidden layers because it resembles the mathematical function which it is trying to learn.

To get the different points for our x is have used the following line:



Gives 5000 random points between -2 and 6.

Chart, line chart

Description automatically generated

For the experiment above I changed the neural net to the following:

Graphical user interface

Description automatically generated with medium confidence

I also decided to increase the number of points to 5000 and make it run for 150 epochs.

Sometimes it would learn the function in less than 100 epochs and other times it wouldn’t learn the function at all. But I have found most success with above given parameters.

## b)

## c)

## Chart, line chart Description automatically generated

## d)

The neural network to predict the sin(x) function consist of 2 hidden layers with 100 neurons each. For the number of points used I chose 2500, which I felt was standard for this type of problem. It goes through 150 epochs and gets better at predicting once it nears the end. The result is not perfect, but I would say that it is not far away. Increasing the number of epochs to 200 makes it even more precise in its predictions and the 50 extra epochs doesn’t take very long to go through.

Run more experiments with different parameters

Chart, line chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

(activation=sigmoid, epochs=150, data points=2500)

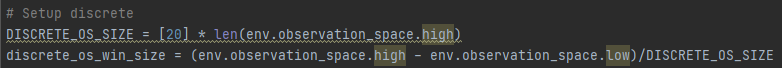
# Exercise 3.a

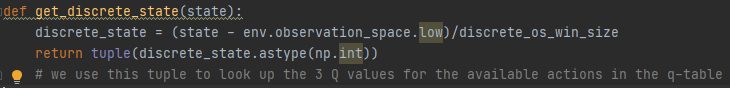
# Exercise 3.b

## a)

The “infinite number of states” problem is caused by the number of possible positions existing between -1.2 and 0.06 and the same with velocity our between -0.07 and 0.07. When the state space is infinite we can’t use q-learning to solve the problem. To make it so that the state space is not infinite we can discretize the state space. This means we are basically making categories for the space, where we can choose the best action to take based on the category. When making these categories we don’t want too many or too few.

Basically the following lines:





Returns a tuple with the categories for positioning and velocity. Maybe add snippet about different states that are returned or all the possible states. Based on those it becomes easier to determine the next action to take.

Another way to solve this could be by using a DQN (Deep Q Learning), where we can use a neural network to approximate the q-function. Necessary?

## b)

Default:

Chart, line chart, histogram

Description automatically generated



If we try and change the number of states that are used, which as mentioned earlier is discretized, we will get…

Learning rate… missing in 3.a

To make it learn more effectively the epsilon can be implemented dynamically, where at the beginning is a high value making it explore and learn the different actions. Once it reaches higher episodes it can begin lowering the epsilon and taking more greedy actions. maybe

Discount rate…

<https://ikvibhav.medium.com/open-aigym-simple-sarsa-and-q-learning-reinforcement-learning-implementations-7d5ea6f1ff9>

# Exercise 4

## a)

If the rounds are set to 30 or 50 rounds the agent won’t be able to explore enough of the board to find the optimal route. We will see the agent with and exploration rate of 0.1

# Exercise 5