Beihang University
School of General Engineering

Beijing, 9th, June
Lin Wei and Yida Ding

Data Management and Artificial Intelligence
Lab Class 15

Task 1 ***Implement the game of Gluttonous Snake and use matplotlib to visualize it****(40 minutes)*

In this lab class, we will implement a snake game based on the given file **snake_game**. You should first glance over the code to understand the structure for recording the information of the snake, food and scores. Then, you should implement two functions named **step** and **render**, which respectively moves the snake and visualizes the snake and food with matplotplib.

The function **step**, as we did in previous lab classes, should use two parameters, current state and action name, and returns a new state of the game. The function **render** should be able to plot one figure for each state of the snake. That is, once the snake is moved, a new figure should be plotted respectively. For instance, with the setting of using red and blue dots to denote the snake and use the green star to denote the food, one should get the figure similar with Figure 27 for one state of the snake game.

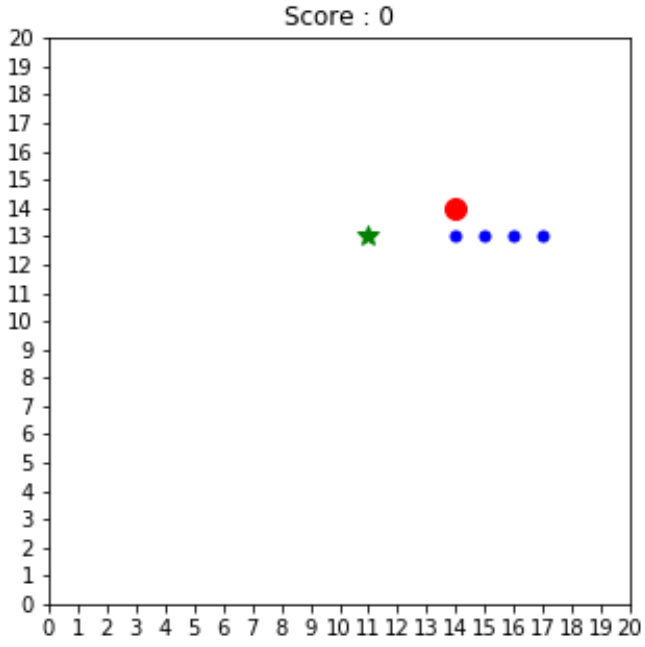

Figure 27: The visualization of one state. The green star is the food and the snake is drawn with five dots.

Task 2 ***Train your computer to be a good player with neural network****(50 minutes)*

As you might have noticed, the random setting of choosing action always meet the collision in a very few steps so the final score is usually low. For this task, you should use the neural network to train your computer to be an experienced gamer. In specific, you should put the actions, states into the network as the input and use the score as the output. The goal is to make the decision of choosing action more sensible.

In the file **neural_networks.py**, several functions are already implemented. You should first understand the structure of the code. What you need to do is to implement the functions named **model** and **test_model**. In this case, no hidden layers is needed. The input layer should preferably contain 25 neurons and the output layer should contain only one neuron because the output is in one-dimension. The activation function prefers to be *relu* and three epoches is enough to build one good model.

After building the model and finishing the learning process, you can load the model for testing and see the score. Then, you should implement the core part of **test_model** function which generates the current action with the prediction from the model. You should pay attention to the type of each variable and make them consistent. By calling the test function, you can visualize the improvement in scores with the training of neural networks.