Lecturer: Henning Sprekeler

Reinforcement learning model of spatial navigation

The report for this project (no more than 4 pages of text with discussion and interpretation as a PDF file) as well as the source code should be handed in no later than **August 7th** through the Moodle interface.

We encourage you to meet with the project instructor as regularly as possible, but at least three times before deadline, to discuss both your advances and hurdles in the project. In addition, if you have any questions you are encouraged at any time to send an email to your project supervisor.

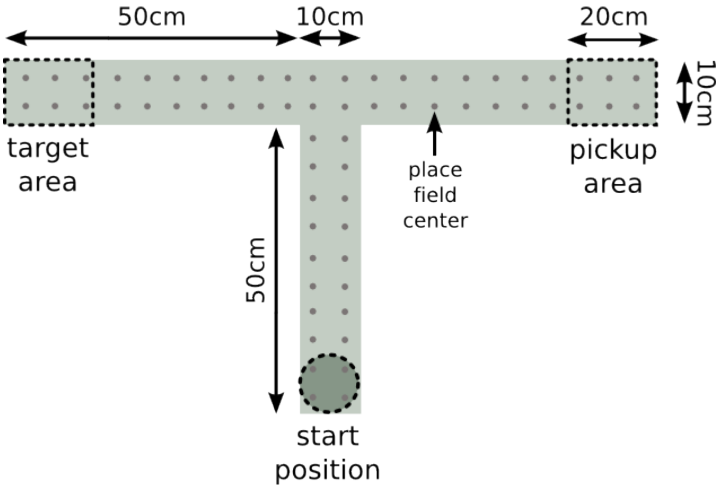

Figure 1: The spatial structure of the T-Maze. In every trial, the rat starts at the start position. Its task is to go to the pickup area before moving on to the target area, where it receives a reward. No reward is given at the pickup area or if the target area is visited without a previous visit of the pickup area. The small grey points indicate example positions for the place field centers.

1 Summary

The goal of the miniproject is to simulate a laboratory version of the stick-throwing game commonly played with dogs. A rat should learn the following task: Starting from the lower end of a T-shaped maze, it should first go to a *pickup area* at the upper right end of the T-maze before running to a *reward area* at the upper left end of the maze, where it receives its reward. There is no reward at the pickup area, so the animal has to find out on its own that it needs to go there first.

2 Detailed instructions

1. Write a neural network computer model with the following specifications:

- Geometry: Each arm of the T-maze is 50cm long and 10cm wide. The pickup and reward areas cover the final 20cm on the 2 ends of the bar of the T-maze, cf. Figure 1.
- Input layer: The input layer consists of 2 populations of neurons (numbered by a population index $\beta \in \{0, 1\}$). One population is only activated when the animal has already visited the pickup area (pickup state variable $\alpha = 1 \rightarrow$ population $\beta = 1$ is active), the other population only if it hasn't ($\alpha = 0 \rightarrow$ population $\beta = 0$ is active). In addition, the activity of the neurons has a Gaussian dependence on position. In total, the activity of the j-th neuron in population β for state $s = (x, y, \alpha)$ is given by

  $$r_{j\beta}(s) = \delta_{\alpha\beta} \exp\left(-\frac{(x_j - x)^2 + (y_j - y)^2}{2\sigma^2}\right) . \tag{1}$$

  Here, $\delta_{\alpha\beta}$ is the Kronecker symbol that is equal to one if $\alpha = \beta$ and 0 otherwise. The Gaussian dependence of activity on the rat's position is a model of so-called place cells that have been found in the hippocampus of rats. The Gaussian activity profile is centered around a position (x_j, y_j) that is characteristic for the neuron. These centers (x_j, y_j) of the Gaussians are arranged on an equidistant grid that covers the T-maze, cf. figure 1. The widths of the Gaussians is $\sigma = 5$cm. The spacing between the place field centers should also be 5cm.
- Output layer: The output layer contains N_a neurons. The a-th output neuron represents the action of moving in the direction $2\pi a/N_a$. Each output neuron is connected to all neurons in the input layer with connection weights $w_{aj\beta}$. The activity of the output neuron a for a given state $s = (x, y, \alpha)$ is $Q(s, a) = \sum_{j,\beta} w_{aj\beta} r_{j\beta}(s)$. Start with a network with $N_a = 4$ different actions.
- Action choice: When the animal is in state s, its next action is determined by the activity of the output cells. With a probability of $1-\epsilon$ it moves in the direction that corresponds to the highest activity in the output layer: $a^* = \text{argmax}_a Q(s, a)$. To make the animal explore the environment, it does not always choose the "best" action a^*, but performs a random action $a = \text{rand}(2\pi/N_a, 2\times 2\pi/N_a, 3\times 2\pi/N_a, ..., 2\pi)$ with probability ϵ. The distance the rat traverses in one time step is drawn randomly from a Gaussian distribution with a mean of 3cm and a standard deviation of 1.5cm (the noise in the distance helps to prevent the rat from running into repetitive motion).
- Rewards: When the animal hits a wall, it gets a negative reward of -1. When it reaches the reward location after having visited the pickup area, it gets a positive reward of +20.
- Learning: The weights $w_{aj\beta}$ are updated on each time step according to the SARSA algorithm with learning rate $\eta \ll 1$, reward discount factor $\gamma = 0.95$ and eligibility trace with a decay rate of $0 < \lambda < 1$. Note that this is the case where the eligibility trace is combined with a function approximation, as covered at the beginning of lecture 7.
- Trial structure: The animal starts a trial from the bottom of the T-maze and moves around until the task is completed. After that a new trial is started.
- Exploration vs. Exploitation: The parameter ϵ should be close to 1 at the beginning of training and progressively decrease to a smaller value ($\approx$0.1) by the end of training.

2. Analyze the computer model

- Simulate at least 10 rats that learn the task and plot the escape latency (time to solve the task), averaged across all animals, as a function of trial number (i.e. the learning curve). How long does it take the rat to learn the task?
- Visualize the behavior of the animal by plotting the navigation map of the animal, i.e., the vector field given by the direction with the highest Q-value as a function of position for each value of the state variable α. Plot the navigation map after different numbers of trials and discuss how it depends on trial number and animal position.
- Compare the learning curves for different values of the decay rate of the eligibility trace, e.g. $\lambda = 0$ and $\lambda = 0.95$. What is the role of eligibility trace? Why?
- Vary the time course of the exploration/exploitation parameter ϵ? Why is it better to have large ϵ in the beginning of training and small ϵ at the end of training?
- Vary the number of actions cells (i.e., the number of different directions in which the rat can run). Does the learning curve depend on the number of actions?

3. Write a report that presents the results of your model analysis. Marks will be based on the clarity of the presentation and on the thoroughness of the analysis.

References

Book: R. Sutton and A. Barto *Reinforcement learning - An Introduction* MIT Press 1998. Chap. 6.4 and 7.5.