ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Unsupervised and reinforcement learning in neural networks

2nd mini project

Lecturer: Marc-Oliver Gewaltig

Assistants: Dane Corneil
Berat Denizdurduran

Reinforcement learning

The goal of the mini-project is to implement a reinforcement learning paradigm using a rate-based neuron model. The setup is as follows: A rat is placed into the corner of an arena and gradually learns the position of a (hidden) goal area where it gets a reward. You will first implement the basic set-up in step 1 and then study the performance as a function of some of the network parameters (step 2). This exercise relates the SARSA algorithm on the neural network level to experiments with behaving animals used in psychology (e.g. the Morris water-maze task).

1 Implement the neural network model.

First, you should implement the environment and the neural network that controls the movement of the rat in the arena. Consider the following specifications:

- The rat is moving around in a rectangular arena with unit area. The position $s(t) = (s_x(t), s_y(t))$ of the (point-like) rat is encoded in the activity of a population of place cells. Let there be 20x20 place cells for which the activity of the jth cell is given by $r_j(s) = \exp(-\frac{(x_j-s_x)^2+(y_j-s_y)^2}{2\sigma^2})$ where the centers of the place cells are arranged on a 20x20 grid (with grid distance 1/19). Set $\sigma = 0.05$.

- The output layer of the neural network consists of eight neurons (*action units*). The activity of the a-th output neuron represents the Q-value of moving in the direction $2\pi a/8$, given the current state. Each output neuron is connected to all neurons in the input layer with connection weights w_{aj}. The activity of the output neuron is $Q(s, a) = \sum_j w_{aj} r_j(s)$.

- When the rat is in state s, its next action is a step of length $l = 0.03$ in the direction of $a^* = \arg\max_a Q(s, a)$ with probability $1 - \epsilon$ or a random movement in one of the eight directions with probability ϵ (this strategy is called *epsilon-greedy*). For the beginning, set $\epsilon = 0.5$.

- When the rat hits the wall (its position after a movement exceeds the unit area), move it back inside the arena and assign a reward of size -2.

- A goal area is defined as a circle of radius 0.1 around the centre with coordinates $(0.8, 0.8)$. When the rat enters the goal area, it receives a reward of +10 and the trial is stopped. At the beginning of a trial, the rat starts at position $(0.1, 0.1)$.

- In each time step, the weights w_{aj} are updated according to the SARSA algorithm with learning rate $\eta = 0.005$, reward discount rate $\gamma = 0.95$ and eligibility trace decay rate $\lambda = 0.95$. For the SARSA algorithm, use the formulas for the synaptic update rule and eligibility trace that were given in the lecture (see your notes or the slides on the moodle).

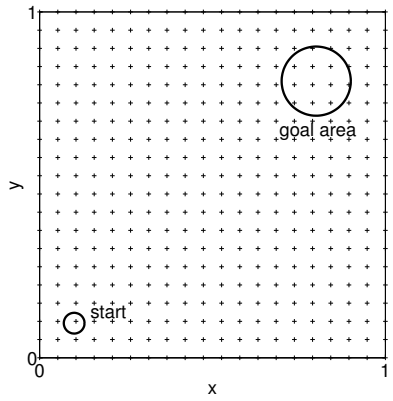

Figure 1: Geometry of the experiment: At the beginning of each trial, the rat starts at position $(0.1, 0.1)$ (small circle). There is one goal area in the opposite corner (big circle). The crosses represent the centers of the place fields.

2 Analyze the neural network model.

- **Learning curve**: Simulate at least 10 independent rats that run 50 trials each. If within a trial, the rat does not find the goal within N_{max} steps, abort the trial. Depending on your implementation, choose e. g. $N_{max} = 10000$. Plot the learning curve, i. e. the number of time steps it takes in each trial until the goal is reached ("latency"), averaged over all rats. If your implementation is correct, the curve should decrease and reach a plateau after a certain number of trials.

- **Integrated reward**: Instead of looking at the time it takes to hit the target area, you can also look at the total reward that was received on each trial. Is the result consistent with the latency curve?

- **Exploration-exploitation**: The parameter ϵ controls the balance between exploration and exploitation (why?). See how different values of $0 \leq \epsilon \leq 1$ change the performance. For the comparison, you could plot several learning curves for different values of ϵ or use e. g. the average latency in the last 10 trials as a performance measure. Let ϵ decrease from a large value at the beginning to a small value at the end of training.

- **Navigation map**: Visualize the $Q(a, s)$ map for different stages of learning (e. g. plot the preferred direction at each place field center using arrows or a color-code).

3 Report

Your report must be handed in by Friday, January 9th 2015, at 23:55. You may work in teams of two persons but you should both upload your report. Submission takes place via the "moodle" web page. You should upload a zip file (named after your last name) containing a PDF of the report and the source code of the programs. Please use either Python or Matlab to generate the source code; the code itself is not part of the written report. The report must not exceed 4 pages.
Feel free to add any comments or additional observations that you had while working on the mini-project.