Third Problem Set for Machine Learning B

708.068 KU Maschinelles Lernen B, WS 2006/07

Latest Submission Date: Tuesday, January 30th 2007, before 18:00.

Guidelines:

- Write down clearly your name and matriculum number, as well as the names and matriculum numbers of all team members on the first page of your submission.
- Write down your solutions on separate sheets of paper for every exercise.
- Submit your code per email to pfeiffer@igi.tugraz.at for those exercises where it is requested.
- Teams need to submit their code only once, unless different versions were used by different team members.
- Exercises marked with * are bonus exercises.
- No teamwork is allowed for literature survey tasks!

1. [20 points] Literature Survey: Select one topic out of the following catalog, and carry out online search for papers relevant to that subject. Try to get an idea of the current state of knowledge on that question, and describe it on at least 3 and not more than 5 pages. Include an additional section with references to the most relevant (and most recent) papers which relate to that question. Do not summarize papers that were already discussed in the lecture, but reference them if new papers build upon that work. Find out how the summarized papers relate to the research currently pursued by the authors and what their main research goals are.

   (a) Partially-observable Markov Decision Processes (POMDPs)
   (b) Reinforcement learning in game theory and financial applications
   (c) Multi-agent reinforcement learning
   (d) Bayesian methods in reinforcement learning
   (e) Predictive state representations
   (f) Learning with multi-dimensional reward signals
   (g) Combinations of control theory and RL
   (h) Reinforcement learning applications in the RoboCup
   (i) Reinforcement learning for humanoid robots
   (j) Reinforcement learning for dialog systems
   (k) Reinforcement learning for board and computer games
   (l) How can a good online performance for RL (i.e. good performance already during training) be achieved?
   (m) Alternative function approximation techniques for RL
   (n) Reinforcement learning theories for animal behavior
   (o) Reinforcement learning models of the basal ganglia
   (p) Genetic algorithms in computer games
   (q) Genetic algorithms for robotics
   (r) Genetic algorithms for combinatorial optimization (e.g. NP-hard problems)
   (s) Applications of genetic algorithms in finance and economy
   (t) Mathematical models of gene regulatory networks
   (u) Alternative approaches to policy learning (ant-colony algorithms, artificial immune systems, simulated annealing, ...)

1
2. [15 points] **Genetic Algorithms:** Use genetic algorithms to solve the *Truck Backer-Upper* problem. In this control problem, the goal is to guide the midpoint of the rear of the trailer of a truck to the target point \((0,0)\) on the loading dock with the trailer orthogonal to the dock. The control variable is the steering angle \(u\) for the tires of the tractor (cab). The cab is connected to the trailer at a pivot point. Figure 1 illustrates the setup of this task.

![Diagram of Truck Backer-Upper Problem](image)

Figure 1: Truck Backer-Upper Problem

The state space consists of four dimensions: \(x\) and \(y\) are the horizontal and vertical positions of the rear point of the trailer. \(\theta_t\) is the angle between the trailer and the \(x\)-axis, and \(\theta_c\) is the angle of the cab with the \(x\)-axis. The goal is to bring the 3 variables \((x, y, \theta_t)\) close to zero. You can only control the cab by setting the wheel angle \(u\). The truck always drives backwards with constant speed \(R\).

The following default values define the constant properties of the truck:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R)</td>
<td>0.5 [m/s]</td>
<td>speed of truck</td>
</tr>
<tr>
<td>(DC)</td>
<td>6 [m]</td>
<td>length of cab</td>
</tr>
<tr>
<td>(DL)</td>
<td>14 [m]</td>
<td>length of trailer</td>
</tr>
</tbody>
</table>

For this task you should evolve a neural network controller that takes at every time step the four features \((x, y, \theta_t, \theta_c)\) as input and returns the control output \(u\). You can download from the MLB website a Matlab simulation for this problem, which should be used to evaluate your controllers. Your task is to design a fitness function for this problem, such that the controller learns to approach the target point with the correct angle as quickly as possible. You also have to decide upon the architecture of your controller network and the selection, mutation, and crossover parameters for your algorithm.

a) Describe the ideas behind your fitness function and parameter setup verbally, and explain why a high fitness value corresponds to a good control strategy for this problem.
b) Use the *Genetic Algorithm Optimization Toolbox* (GAOT) toolbox to implement your fitness function and genetic parameter setup. Let the truck always start from $x = 4, y = 2, \theta_l = \frac{\pi}{2}, \theta_c = \frac{\pi}{2}$ and evolve a neural network controller to back-up the trailer to the target point with the trailer perpendicular to the loading dock.

c) Show in a plot the evolution of your fitness function over time. Plot the maximum and average fitness per generation as well as the fitness fluctuation within the generations.

d) Describe the policy that your controller finds verbally and graphically (you can use the Matlab-script `plot_TruckBackerUpper.m` to visualize your controller's performance). Also investigate policies at early and intermediate generations and describe how their behavior is different from the final solution.

e)* [5 *-points] Evolve a controller that can solve the problem from multiple starting positions and angles. You can either use a fixed set of min. 8 different starting setups, or you can randomly generate initial conditions. In the latter case make sure to exclude starting positions from which it is impossible to reach the goal in time.

f)* [5 *-points] Instead of using constant backwards speed, add the velocity $r$ of the truck as a second control variable and try if you can find even better solutions. Allow both forwards and backwards movements of the cab.