Marina Djerekarova
DMP Final Report
November 14, 2006

Abstract

My summer research project at the Artificial Intelligence and Robotics Labs at the University of Oklahoma focused on developing a reinforcement learning/evolutionary algorithm to add more capabilities to an existing art/computer science project. The project was a sensor network consisting of 4” x 3” x 2.5” pi-shaped objects. Each object, which from now on will be called bion, is like a mini electronic device, which will be further described in the Introduction section. The network was programmed to play a particular sound sequence as people approached and stayed there. The goal of my project was to make the bions act according to people's wishes. As a person approaches, the bions would start playing a song. If in a short period of time he/she walks away, next time someone approaches, the bions will play a different song. This way the bions will learn the most desired song to play, based on the person's interaction with them. My project was developing and implementing the reinforcement learning algorithm with the guidance of my mentor. This paper will describe the project and the results, which I was able to reach over the course of 8 weeks.

Introduction

The bions are a collaborative project between the computer science and the art departments at the University of Oklahoma. The project was developed by Dr. Andrew Fagg and Dr. Adam Brown. The name of the project was inspired by the following definition: “Bions are transitional forms between non-living and living matter. The bion is the elemental functioning unit of all living matter. At the same time, it is the bearer of a quantum of orgone energy and, as such, functions in a specifically biological way.” [1].

The artwork consists of 1000 bions, around 70 of which are usually hung in the lab. An individual bion is a 4” x 3” x 2.5” synthetic "life form" shaped as orgone, or as some may say, like the greek letter pi. Each consists of 1 k RAM and 8 k instruction space, an audio speaker, 4 blue LEDs, and 4 sensors. The bions exhibition may be thought of as a sensor network. Bions can communicate to each other through infrared. The program, which was installed on the Bions when I started working on the project is as follows: when the bions are connected to electricity, they play a high pitched tone sequence; when people approach the sensors, the bions become quiet, as if they are afraid of the intruder; in 5 seconds, they "adapt" to their new environment and start playing a more lively tone sequence, as if they are excited by their new friend; this happens gradually as the bions closest to the person communicate to their "peers" that it is safe to resume regular activity. So far the bions have been touring the United States at exhibitions in art museums and computer graphics conferences.

Procedures

The first step for me was to become familiar with Reinforcement Learning and the existing fundamental C code operating on the bions.

Reinforcement Learning is learning from interaction. It is a foundational idea underlying nearly all theories of learning and intelligence. Reinforcement Learning is learning how to map situations to actions so as to maximize a numerical reward signal. The four main sub-elements of a reinforcement learning system are a policy, a reward function, a value function, and a model of the environment. A policy defines the learning agent's way of behaving at a given time. It is a mapping from perceived states of the environment to action to be taken when in those states. A reward function defines the goal in a reinforcement learning problem [2].

One challenge in reinforcement learning is the trade-off between exploration and exploitation. An agent must choose actions that it has found effective in the past, but in order to discover them it must try actions that it has never selected before. Reinforcement learning involves interaction between an active decision-making agent and its environment, within which the agent seeks to achieve a goal despite uncertainty about its environment. The agent uses its experience to improve its performance over time.

I started by reading the first few chapters of "Reinforcement Learning" by Sutton and Barto [2]. Then I implemented a Temporal Difference(TD) Learning algorithm on a gridworld in Java. The algorithm I used was the Q-learning off-policy TD control algorithm (Barto, 149).

At that point I had the basic idea of what Reinforcement Learning is and I started working on understanding the bioncore. I familiarized myself as much as possible with the code by reading the documentation. For better understanding my task was to implement a new function, which allowed the bions to play any short song that the user would like to provide the note sequence for.

That being done successfully, I moved to the major part of my project, namely Reinforcement Learning on the Bions. My mentor and I discussed creating a bion simulator, which uses a TD Reinforcement Learning Algorithm. We had a basic outline of the program design.

> If proximity is detected, learn to keep people near the bions.
> Else give penalty, because the goal is to keep people near the bions.
> If proximity has been off for at least three seconds, learn to attract people to the bions.

A linear approximation function was used to calculate the value for a state by summing up the appropriate weights and a bias term. The actions for the learning tasks were chosen from nine possibilities: silence and an octave of notes. Each note would be played for a predefined period of time and every time a new note is played, it would be communicated to all neighbors in proximity. We calculated the TDerror by the following

equation: TD error = r + gamma V(s') – V(s) .On each learning step, each weight and the bias were changed by the equation: weight = weight * alpha * TD error.

Coding and debugging was my main job for the next 2 weeks. Finally, the result was a strange looking RL curve. The graph clearly showed that something did not go as planned. There were too many oscillations, which were partially due to the noise in the environment, and partially to the fact that linear approximation was not the correct choice for this application. Linear feature approximation was unable to approximate a non-linear function without non-linear features. We discussed two possibilities - neural networks and evolutionary algorithms. Neural networks use a lot of memory, which was unavailable on the bions. The evolutionary algorithm was easier to implement and more suitable to our project. I read the chapter on Neural Networks and Evolutionary algorithms in "Machine Learning" by Tom Mitchell [3].

We designed a new algorithm using the genetic algorithm approach. Following is a description of the key points of its implementation. When the algorithm is implemented on the hardware bion network, the person's precence would be used as an indication that the song sequence was correct. There would be a whole population of bions, each bion evolving its own song. I made a simulation program that had a configurable number of bions and each bion started with a random song of specific length. The initial parameters were a population of 10 bions and a song length of 3.

For the software simulation, each bion was initialized with a random song. The goal of the genetic algorithm was for the bions to play a specific sequence (say 1 3 5). The fitness would be maximum only when that exact sequence was played. The fitness function was a number between 0 and 1 where 0 meant the sequence played did not contain any of the desired notes and 1 meant all notes were played in the correct order. For partially correct songs, the fitness would be a fraction based on the correct number of notes played. To test that in the simulation we used the following parameters: habituation at 3 notes and the fitness function became 0 if none of 1 3 5 is played, 1/3 if only the first note was correct, 2/3 if the first 2 notes were played correctly and 1 if all three notes were correct. After every three-note-sequence, each bion would broadcast its fitness function and song. A crossover was done between a fraction of the bions selected according to a selection algorithm as described in the Machine Learning book.

In the hardware implementation fitness would be measured based on the amount of time the person stayed near the bions. Each bion would be initialized with a random song and it would start playing it when a person approached. If he/she stayed long enough for the bions to habituate, the fitness would become 1. After every three notes, each bion would communicate its fitness function and song to the bions in its proximity cloud. Any bion who heard it and had a lower fitness value, would either fully switch to the new song if the difference in fitness functions was large or crossover between the two songs if the difference was small.

The next week I spent in planning, coding and debugging the Evolutionary simulator. It took less time to finish, since there were some similarities with the RL algorithm and I

had already gained some experience in using linux, the gedit environment, and correcting the compiler and runtime errors. The simulation ran great. The details of the results are in my journal for Week 8. The last week and a half was for me to merge the simulation code with the bioncore.c, which was not an easy task. The simulation took megabytes to run and the available memory on the bions was 1 k on each. I had to change a lot of code and find a way to have the same code on each bion and at the same time they were supposed to be able to communicate and change their state. We realized that we needed to change some of the bioncore, because the data to be communicated had to be encoded in 19 bits and the bioncore was programmed to communicate only 4 bits.

The most dissatisfying part was that I was supposed to work on code in which nothing was certain. Changing the communication from 4 bits to 19 bits meant that the probability of the data not reaching its destination successfully was higher. Communication in the bions happens bit by bit, so the more bits there were, the easier it was to have bits intercepted by external noise. We decided on learning the notes in chunks of 8. And as time was quickly running out, I started implementing the learning of the first 8 notes. Due to runtime errors, I never got the chance to test the communication on more than 2 bions. Debugging hardware was very challenging, since it had to be done using the oscilloscope and the last evening I did not manage to find the error in my function.

Results

The results of the bion simulation with the original reinforcement learning algorithm produced an unexpected learning curve. It did not represent what we were aiming for, which lead us to the conclusion that linear feature approximation was unable to approximate a non-linear function without non-linear features. The results of the evolution simulation were positive in the way that given a short sequence of tones, a simulated population of 10 or 20 bions could learn pretty quickly to play the desired sequence. The results showed that the bions would learn in an average of 400 attempts with a population of 10 and an average of 880 attempts with a population of 20. The trials were based on a song of length 3. The learning simulation for a song composed of more tones, 5 and up, usually never completed in a reasonable amount of time (5 - 10 minutes).

In order to make the communication between the bions for our project, we had to change all integer type fields to long types so that we could encrypt information in 19 bits. The results of the communication test between two bions was successful, but we were unable to obtain results for the entire network due to insufficient time.

Conclusions

This project was a great learning experience and an interesting research project. Due to insufficient time, I did not see the bion network learn to play a song according to a person’s wish. However, I learned many new things in the process of working on this

project. The project was very close to completion and had a good chance of being successful if there were a few more weeks in which I could work on it. Hardware problems and last minute errors arose during the last few days.

The bion computer science/art project has a potential for great success in the new age computer science and one day I would like to see it on an exhibition, with the reinforcement learning program installed on the bions. If the current algorithm is not appropriate for the application, I am sure there are ways that the evolution algorithm could be improved to fit in the capabilities of a network comprised of 1 k bions.

Acknowledgements

I would like to thank the sponsors of this research - the Distributed Mentor Project (DMP), part of the committee on the status of Women in Computing Research (CRA-W). I would also like to thank my mentor Dr. Amy McGovern for her continuous support, as well as Dr. Andrew Fagg and all the students in the Artificial Intelligence labs, whom I worked with.

References

[1] “BION” 9 Sept. 2006 <http://www.isisconceptuallaboratory.com/bion/_Bion.html>
[2] Sutton, Richard S., and Andrew G. Barto. Reinforcement Learning. Cambridge: The MIT Press, 1998.
[3] Mitchell, Tom M. Machine Learning. Cambridge: MIT Press and McGraw-Hill Companies, Inc., (1997).