Robots can Wear Multiple Hats in the Computer Science Curriculum at Liberal Arts Colleges

Christine Shannon

Centre College
600 W. Walnut
Danville, KY 40422
shannon@centre.edu

Abstract

Faculty at liberal arts colleges are often challenged to offer a quality education to their students, complete with opportunities for undergraduate research. To guard against a curriculum that is too theoretical, students want to see applications of their course work and tangible results of their efforts. Like all computer science educators, we want to attract students to our discipline. The use of robotics can often be part of the answer in each of these realms.

Undergraduate Research Projects

Centre College, like many good liberal arts institutions, is anxious to provide its students with opportunities to engage in undergraduate research. Robotics provides a venue where students can make interesting contributions and where an ongoing project can be maintained.

In 2004 I purchased a Sony AIBO from enabling funds associated with a professorship I hold and I was able to secure a grant from the Kentucky Space Grant Consortium to provide stipends for students involved in the project. I was eligible for these resources because the subject of machine learning and robotics was of interest to the funding agency. Three sophomore and junior students, Peter Burns, Jackie Soenneker and Will Larson were involved in the project over the course of two summers and one academic year.

The project was centered around the topic of machine learning, specifically reinforcement learning, and we decided to use the software package *pyro* (http://pyrorobotics.org/) as a means of communication between the computer and the AIBO. Because our students learn Python as their first language and then Java, use of pyro enabled our students to get started on the project more quickly than if they had had to learn C++ first.

While this robot has a large number of sensors our work was almost entirely involved with the vision system which depends on a camera mounted in its nose. Vision processing software allowed us to set filters to detect the presence of the bright pink of a ball in the image produced by the camera. The earliest projects involved the use of reinforcement learning to teach AIBO to move his head so as to locate the pink ball. The results were noticeable. In one experiment, both the ball and the robot always began in the same position. In this case the average number of steps the robot took to locate the ball decreased with training from 29.9 to 5.58. There were similar results for random initial positions of the robot head. With a small data set, initial path averages were as high 33.14 steps but decreased to less than 8.0 with training.

The complexity of the problem increases substantially when we add locomotion to the mix. Now the object was not only for the robot to locate the ball visually but also to move toward it. When the size of the blob exceeded a threshold value, 750 pixels, our experiment declared he was close enough and the number of steps was recorded. AIBO wagged his tail to announce success.

In performing these experiments, one has to make decisions about the sizes of various parameters: how much should the head be moved or tilted, how fast should the robot move forward or turn. Based on ideas contained in the literature (Kohl and Stone 2004) we devised an experiment to find optimal values for six of our parameters. The policy gradient reinforcement learning described there was quite successful in the present setting. Beginning with a base policy which on average required 196 steps to locate the ball, the process of generating base policies, testing, and adjusting the policies in the direction of maximum improvement yielded approximately a fifty per cent improvement in performance. (Soenneker, 2006).

Many of our experiments dealt with temporal difference learning and the problem of estimating the action value function, $Q^{\pi}(s,a)$, which is the value of taking action *a* in state *s* under policy π. At this point we have only worked with action value functions which are implemented as a table. Inspired by a another paper in the literature (Smart and Kaelbling, 1998) we worked on ways to provide initial settings for this array so that the robot did not start with a *tabula rasa* but instead had some guidance in making early decisions. The result was a program which could be run either in a "teacher" mode or a "learning" mode. In the teaching mode, the choice of action was dictated by program but based on the rewards or punishments received, the action value function was updated as though the robot had made the decision. Once the program was switched to learning mode, the robot chose its actions based on the contents of the $Q^{\pi}(s,a)$ table, slowly improving the estimates depending on the rewards and punishments resulting from its choices. Of course, there was some portion of the time when the action was chosen at random to encourage exploration for even better choices than had been tried in the past.

Results from these experiments show that the average number of steps per episode fell from over 350 to less that 250 as training progressed.

Observations

The kinds of experiments that were carried out by these students gave them a lot of freedom to pose and test hypotheses. In the course of our work, we tried a variety of state descriptions, different types of reward and punishment schemes, and variations in the types of questions asked. The work was accessible even though the mathematics was quite advanced to justify everything that was done. One student has submitted an abstract on the results of this work to an undergraduate research contest at a regional meeting.

There are a great many more things that can be explored – including the possibility of implementing the entire thing without the use of the pyro software which does add a layer between the programmer and the hardware. The introduction of eligibility traces will probably be the next stage in this process.

It is my hope to keep this as an ongoing project that students can join primarily in the summer but also during the academic year as time permits. Centre maintains an honors program named in honor of John C. Young in which students can propose research projects to be conducted during their senior year. The research papers are presented in a Spring symposium open to the entire community and published by the college. The existence of this continuing project makes it possible for students to propose an honors thesis which is substantive while at the same time having a reasonable horizon. Robotics offers a very suitable platform for interesting and useful experimentation.

Robots as Projects in CS Courses

Before the inclusion of robotics projects in my artificial intelligence course, I regularly struggled to find assignments that enabled students to implement many of the ideas that were presented. Most projects were simply too big to implement in a semester. However, whether it is using Lejos to program Lego robots, or the use of pyro with simulators, there are now ways to make interesting but reasonable assignments in small worlds where students can exercise the techniques that are introduced in lecture. I now regularly have them use Lego robots to implement a simple reactive agent early in the course and later an agent which makes use of a more complex decision structure and state to make its decisions.

I also teach about a two week segment on robotics at the very end of my CS I course. Having learned Python in the course, the students have not had any experience with complex syntax or the need to declare variables and types. The use of NQC to program Lego robots gives me an opportunity to introduce a second language with the more complicated syntax as well as to introduce some other ideas like asynchronous execution.

Observations

Students like the idea of building robots but they also experience a significant level of frustration when they realize that effectors are not as exact as they would like them to be. They find that building and testing a robot is more complicated than trying a line of code to see if it will work. In spite of this, there is usually a fair amount of cheering when we demonstrate the projects.

Robots and Outreach

All of us are concerned about attracting more students to our discipline. In recent years, robots have been an attraction to some students. I have given presentations with the AIBO at the local middle schools and they have been very warmly received. Actually, they would have been quite satisfied just to watch the robot. I am asked to give talks about robots and artificial intelligence to prospective students and while the numbers are not large, they do garner an interested audience.

In the past year we have attracted some students who have been involved in some sort of robotics competition during their high school years and some want to continue that involvement during college. We have only begun the process of developing something in this area. We have acquired a VEX radio controlled robot but significant progress awaits more time to devote to this activity.

Observation

Time is probably the resource in shortest supply around a liberal arts college. Developing significant outreach can be quite prohibitive in view of the time commitments involved. When the content of research projects can be used to effectively garner student interest, the extra time involved can be marginal. Projects in robotics can often be used to demonstrate the kind of excitement that often arises in the study of computer science.

Conclusion

Robotics can be used as a source of undergraduate research projects and to enhance the learning of topics in computer science as well as to attract additional students to disciplines like computer science and engineering. In a liberal arts college it offers the opportunity to provide students with experiences which enrich their education and provide applications of theory. They can be a good investment of faculty and student time.

Acknowledgements

Research described in this paper was done with students Peter Burns, Jackie Soenneker and Will Larson. It was supported in part by a grant from the Kentucky Space Grant Consortium and by funds in the Margaret V. Haggin professorship.

References

Kohl, Nate and Peter Stone. “Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion.” *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA 2004). Pp. 2619-2624, New Orleans, LA, May 2004.

Soenneker, Jackie “Machine Learning on an AIBO Robot.” CCSC:SE , Lipscomb University, Nashville, TN. Nov. 10-11, 2006, submitted.

Smart, William D., and Leslie Pack Kaelbling. “A Framework for Reinforcement Learning on Real Robots.” <http://www.cse.wustl.edu/~wds/content/papers/aaai1998/aaai1998.pdf>.

Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction*. London: The MIT Press, 1998.