Teaching Neural Networks Using Lego-Handyboard Robots in an Artificial Intelligence Course

Susan P. Imberman

College of Staten Island, City University of New York

2800 Victory Blvd.

Staten Island, NY 10314

Imberman@postbox.csi.cuny.edu

Abstract

In this paper we propose a novel method for teaching neural networks with back propagation in an undergraduate Artificial Intelligence course. We use an agent based approach in the course as outlined in the textbook <u>Artificial Intelligence A Modern Approach</u> by Stuart Russell and Peter Norvig [7]. The students build a robot agent whose task is to learn path following behavior using a neural network. Robot agents are constructed from standard lego pieces and use the MIT Handy Board as a controller.

Keywords

robotics, artificial intelligence, handy board, neural networks, back propagation

1. Introduction

The use of lego based robots to augment concepts taught in undergraduate computer science courses has been tried in various ways across the curriculum. A natural place for the use of lego based robots has been in robotics and artificial intelligence courses.[4]. Robots have been incorporated in other courses in the computer science curriculum such as beginning programming [1].

The MIT Handy Board was developed by Randy Sargent and Fred G. Martin. [5]. It is a robotics controller based on the Motorola 68HC11 microprocessor. The controller contains 32K of battery-protected RAM. It has 4 DC motor outputs, 9 digital and 8 analog inputs,

which support a diverse set of sensors. There are several different compilers available for the Handy Board including a version of the C programming language called Interactive C, also developed by Randy Sargent. [3] Using the Handy Board as a micro controller for lego based robots can result in the creation of robots with sophisticated behaviors.

Sondak and Sondak argue [8] the importance of adding the topic of neural networks into the computer science curriculum. They cite the intense media attention and the wide breadth of research interest in the topic as the basis for their conclusion. They also argue that because of the pedagogical and historical relationship between Artificial Intelligence and the subtopic of neural networks, it makes sense to include the study of neural networks in an Artificial Intelligence course.

In this paper, we offer a hands-on methodology for teaching and implementing neural networks. Our agent approach uses robots that learn path following behavior via a neural network. We have used this approach in an undergraduate Artificial Intelligence course with great success.

2. Background

The College of Staten Island (CSI) is a four year liberal arts college. It is part of the City University of New York and the only four year public institution of higher learning on Staten Island. The academic year consists of two 15 week semesters. CSI is a commuter college with approximately 85% of our students working part time or full time while taking classes.

The Computer Science department graduates approximately 80 computer science majors each year. Two other undergraduate majors also include Computer Science courses in their curriculum. These are Bioinformatics and Management Information Systems. The Artificial Intelligence course is an elective course for all three

majors. Artificial Intelligence is given once a year, in the Spring semester and has a prerequisite of a Data Structures. Thus students taking Artificial Intelligence at CSI are proficient in the C++ language. Most students taking the course are in their Junior or Senior year of study.

Artificial Intelligence, meets each week for 4 lecture hours. The course is organized so that the last three weeks of classes, weeks 11-14, are reserved for a hands on robotics project. Final exams are scheduled in the fifteenth week during regularly scheduled class times. Therefore, we were able to use the final exam time during the fifteenth week for the robot project as well. Hence the total time spent on the robot project is $3^1/_2$ weeks. All lecture topics are finished before the start of the robot project. The final exam is also given before the start of the project. Students start their projects without having to worry about "forgetting" topics gone over previously.

3. Methodology

In preparation for the robotics project, during the lecture portion of the course, students are given one lecture on robotics and two lectures on the basics of neural networks using back propagation. Students are taught about perceptrons and the perceptron training rule. They are shown why perceptrons learn concepts that are linearly separable. From here we talk about sigmoid units that are connected in multilayer neural network of two input, two hidden and one output unit. Using this simple neural network we define the back propagation algorithm. A "textbook" C++ implementation of a neural network configured as described is available at the Generation5 website [2]. We use the code provided at this web site in a real time demonstration of neural networks. The code learns the XOR function. Students are shown how accuracy changes with increased training time. By changing the training examples, students see that the same topology can learn an AND function or an OR function. Students are then given the assignment to modify the generation5 code to implement a neural net that has two input nodes, two hidden nodes and two output nodes. It is at this point the students are first exposed to what they will need to do with their robots. The robots are configured with two photosensors, used to navigate the given path. Depending upon the readings from these sensors, the speed of the two motors that drive the robot are adjusted to maintain the robot's position on the line. Students see the direct connection between the topology of the neural net and the architecture of the robot.

For the project phase of the course, students are grouped into teams of three. Students construct their robots based on a model found to be suitable for the path finding task. Our goal is not to spend too much time constructing the robot. Students build their robots based on the architecture exhibited by two working models. A web page detailing the robot model is available to students for reference purposes. Student robots must contain a set of

standard features but need not mimic the construction of the basic model. Figure 1 shows a picture of the basic model and Figure 2, two student built variations.

Our basic robot architecture uses two photosensors, obtainable from any electrical supply company. The body is "tank like" with wheels connected using gears and Lego chain link. The gear train is an essential part of our basic architecture. Improper gear reduction can result in a robot that does not have the necessary power to move. Web based instruction illustrating each step in the creation of a gear box is provided for reference.
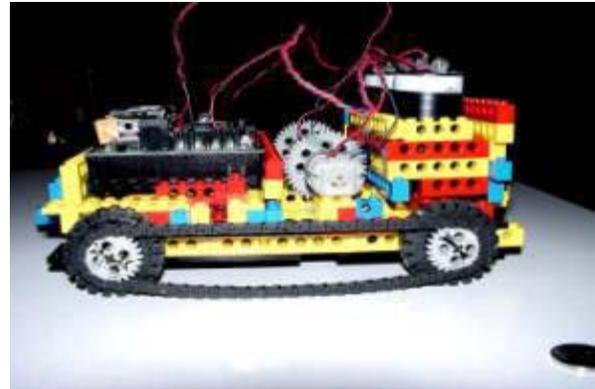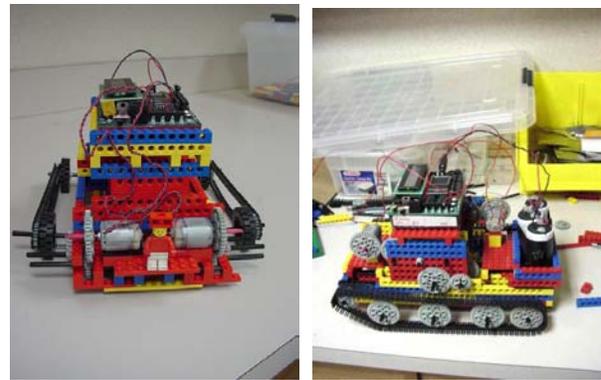

Figure 1 - *The Basic Model*

Figure 2 - *Student Robots*

Students became very enthusiastic when constructing their robots. Many bring Lego pieces from home in order to "customize" their robots. Most student teams finish robot construction in about 4 class hours or one week.

In our first attempts at path following we used a "road" constructed from white poster board and black electrical tape. It was found that sensor readings from this combination were not accurate enough to differentiate between the road and the background. Our solution was to place the black electrical tape directly on the floor. This proved more successful. Problems still occurred since the floor tile had a mottled rather than solid pattern. It

appeared that color contrast was not significant for accurate photo sensor readings, but the reflectivity of the surface was. For the last two years we constructed roads from black poster board with a silver tape. With experience we learned that the robots performance was enhanced when we had "gentle" rather than sharp curves in the road. The tank design we used couldn't handle sharp turns. Figure 3 shows a picture of the road we used.

Figure 3 - *Sample Robo-Road*

Robots were programmed using Interactive C 4.0 which can be downloaded form the Kiss Institute Web Site [3]. Interactive C is an interpreter that has special libraries for controlling sensor and motor ports. Web based instruction on the use of this compiler with the robots was provided [9].

The Robot Project was worth 10% of the student's grade. The project was broken into several subtasks, each worth some fraction of the 10%. As robot teams completed a task, the appropriate entry in a robot evaluation form was noted. These forms were submitted to the instructor at the end of the project. Figure 4 shows the evaluation form.

Materials

Most of the material used to construct robots had been purchased prior to being used in the Artificial Intelligence course. Therefore we can only estimate the per robot costs. Below is a list of all Lego parts used for our tank architecture.

Axels and Extenders (for 2 robots)	$6.75
Lego Beams (2 pkgs)	19.00
Connectors and Bushings (for ~10 robots)	7.50
Lego Plates (1Pkg)	8.50
Wheels and Hubs	6.50
Small Chain Links	7.50

Instructor Initials	Robot Project
	Task 1 - Build a robot. Use *fembot* as your model. Remember the body of the robot has to be strong enough and big enough to support the battery pack and the handy board. Have Robot 1. move forward for 1 second, turn right about 90 degrees 2. move forward 1 second 3. turn left about 90 degrees move forward 4. Stop beep three times 5. Do the above in reverse Read pages 773 - 785 in your text Answer the following questions. 1. Describe your robot's environment. 2. What type of "real world" tasks might a robot like the one you've built be good for? 3. What types of effectors does your robot have and what are they used for? 4. How many actuators does your robot have? Describe your robot's actuators. 5. What is a nonholonomic robot? What is a holonomic robot? Which one of these describes your robot? 6. Is your robot statically stable or dynamically stable? Why? 7. Our robots have very simplistic sensors. Choose one sensor type, as described in your text, and describe how it could be used to help solve the your robot 's problem task. (6 points)
	Task 2 - Have the robot use its photo sensors to follow a path. The path is defined by silver tape on a black background. (2 points)
	Task 3 - Use the neural network software you created to teach your robot how to follow a path. Evaluating the performance of an intelligent agent allows one to improve the agent. Did your robot follow the track better when trained with a neural or when it was hard programmed? (2 points)
Total Score:	

Figure 4 - *Robot Evaluation Form*

8 tooth gear wheel (for ~5 robots)	10.00
24 tooth gear (for ~5 robots)	10.00
Handy Board	300.00
Photo Sensitive Detectors (2)	3.00
DC motors (2)	2.00

Based on the robot materials list, we estimate the per robot cost to be approximately $365. The Handy Board was purchased from Gleason Research, http://www.gleasonresearch.com , photo sensors and motors were from Jameco electrical supply, http://www.jameco.com , and Lego components were from Pitsco Lego Dacta, http://www.pldstore.com/catalog.cfm . Other supplies used, such as poster board, tape, hot glue, solder, soldering irons, etc., came to approximately $150. It is important to note that most of this is a one time expense since robots are dismantled and recycled in subsequent semesters.

The lecture portion of the course is given in a standard classroom. During the period allotted for the robot project, the class is held in our robotics lab. Our robotics lab is composed of an oversized classroom space with sufficient desk space to accommodate 35 students. There is a substantial amount of "unused" space allocated for "robot roaming". The robotics lab is attached to a separate room configured with 20 IBM compatible desktop PC's running Windows 98.

4. Discussion

Memory on the Handy Board is only 32K. Because of this, students quickly realized that back propagation would not run on the robot boards. This automatically differentiated between the process of training a neural network and a trained neural network. Students quickly saw that the linear equations from neural net training are all that was able to fit with the board's memory. Training took place on a desktop computer using a standard C++ compiler. The conceptual difference between training and the "final product" is a difficult one for undergraduate students to grasp. The separation of the two, dictated by the hardware, made this concept very clear.

Because many of our students work while attending school, their course selection tends to slant towards what would give them an advantage in the marketplace. In the past, students tended to favor electives such as database and networking rather than courses in artificial intelligence and graphics. After the addition of robots into the Artificial Intelligence course, large numbers of students opted for this elective.

Course offerings alternate between day and the evening class sessions. Elective courses given in the evening have a higher enrollment since most upper level students work during the day. The Artificial Intelligence course has been using the format described in this paper for the last three years. Enrollment for these years has been 36 (evening class), 24 (day class), and 27 students (evening class). Computer classes are capped at 30 students. In year one, the number of electives for the evening session was not enough to accommodate student need. More students were allowed into the Artificial Intelligence course than usual. It is interesting to note that class enrollment has

remained significant in spite of student perceptions about the "usefulness" of the course material. The course's reputation is that it's "fun". We feel the fun factor associated with the robotics component has encouraged many students to experience an Artificial Intelligence course who might not ordinarily have done so. Many students have returned to admit that the topics covered in the course were indeed relevant to their work experience..

Although we used the MIT Handy Board, a less expensive alternative is the Lego Mindstorms® RCX programmable Brick controller. The Lego Mindstorm Brick was developed by Lego and MIT's Media Laboratory. It can also runs interactive C and supports the two photosensors and two motors needed for this project. Photosensors and motors may have to be modified in order to be used with the Brick. Lego infrared sensors and Lego DC motors can be substituted for the photosensor and DC motors used with the Handy Board. No other changes are necessary to implement this project.

One of the subtasks in the robot project had students hard coding the path following behavior. The idea behind this was to compare the performance of the neural net implementation with the hard coded one. Some robot teams realized better performance with the hard coded implementation and others had better performance with the neural net. Since no significant differences could be found, in the future, we intend to remove this task from the required set.

An optional extra credit assignment requires students to repeat their experiments using decision tree software. For future we would like to add this subtask to the required set of subtasks, replacing the hard coded subtask. A comparison in performance between the neural networks and a decision tree might lead to more meaningful conclusions.

Using photosensors for robot vision presented the problems outlined previously. Lego infra red sensors might yield better results and we intend to try them in Spring '03.

Although we have a dedicated space for robotics, we have done this project in a standard computer laboratory. Robot equipment was stored in a locked storage area. At the start of class, students would retrieve their robots and bring them to the lab. Spare Legos, sensors, and other materials were placed on mobile carts and moved into the laboratory. This is not an ideal setup, but it does work.

5. Conclusions

In this paper we described a method for teaching neural networks and back propagation using Handy Board controlled Lego robots. We have successfully implemented this method into our Artificial Intelligence course. The experience has been a positive one for both students and instructor. Because of the architecture of the robots, students must train their neural networks on a separate desktop computer. The resulting equations are

then programmed separately into the robots. This illustrated the difference between training a neural network and implementing one. All web pages referred to in this paper along with pictures and an MPEG showing a trained path following robot, can be reached via the world wide web from:

http://domanski.cs.csi.cuny.edu/imberman/ai/robotresources.htm

References

[1] Fagin, B.S., Merkle, L.D., and Eggers, T., Teaching Computer Science With Robotics Using Ada/Mindsotrms 2.0.

[2] Generation5.org, *Back-Propagation: CBPNet,* http://www.generation5.org/cbpnet.shtml

[3] Interactive C 4.0, Kiss Institute, Available WWW. http://www.kipr.org/ic/download/

[4] Kumar, D. and Meeden, L., A robot Laboratory for Teaching Artificial Intelligence, In *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education (1998).*

[5] Martin, F., The Handy Board. Available WWW. http://www.handyboard.com/

[6] Mitchell, T., *Machine Learning*, The McGraw Hill Companies Inc., 1997 ISBN 0-07-042807-7

[7] Russell, S., and Norvig, P., *Artificial Intelligence A Modern Approach*, Prentice-Hall Inc. 1995 ISBN 0-13-103805-2

[8] Sondak, N.E., Sondak, V. K., Neural Networks and Artificial Intelligence, In *Proceedings of the 20th SIGCSE Technical Symposium on Computer Science Education February (1989).*

[9] Using Interactive C 4.0, Available WWW. http://domanski.cs.csi.cuny.edu/imberman/csc126/startinginteractiveC40.htm