A Laboratory Exercise Using LEGO Handy Board Robots to Demonstrate Neural Networks in an Artificial Intelligence Class

Susan P. Imberman Ph.D.

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. The students build a robot whose task is to learn path-following behavior with a neural network. Robots are constructed from standard LEGO® pieces and use the MIT Handy Board as a controller.

Introduction

LEGO® based robots have been used across the curriculum to augment concepts taught in undergraduate computer science courses. A natural place for the use of LEGO® based robots has been in robotics and artificial intelligence courses.[3]. Our approach has been to specifically use these robots to help students learn the concepts involved in neural network back propagation.

Many students have difficulty in understanding the concepts behind machine learning. Machine learning algorithms such as decision trees and neural networks are usually taught using hand traced examples. Freeware or commercial versions of the algorithms are often times demonstrated as well. We have found that when using these methods, students lose sight of the real problems associated with the use of machine learning algorithms, such as finding good training examples. Specifically, with regards to neural networks, we have used a different learning tool. Our approach uses LEGO® handy board robots. Students use neural networks to train these robots to implement path-following behavior.

Our educational objectives for using this approach are as follows, students should:

- understand the basic back propagation algorithm.
- understand the difference between neural network training and the trained neural network
- realize the problems inherent in working "in the real world" such as
    - my program works but my robot doesn't move
    - my sensor readings don't show any difference between the path and the background

The laboratory exercise described in this paper is the result of four years of trial and error. We hope that by sharing our experience with other artificial intelligence educators, they will be able to implement a similar exercise sans some of the learning curve difficulties we experienced.

The remainder of this paper is organized as follows. First we describe some of the background lectures and assignments students needed prior to the start of the robot lab. We then describe the basic architecture of the robots. Next we give a list of materials used in implementing this architecture. Last, we describe the lab project in detail including how students were evaluated.

Background requirements

Prior to the robot project, the class is taught in a lecture based manner. Students are given two lectures on the basics of neural networks using back propagation. [5] [6] Using a simple neural network we define the back propagation algorithm. A "textbook" C++ implementation of a neural network configured to implement the XOR function is available at the Generation5 website [1]. We use this code in a real time demonstration of neural networks. The neural network has a topology of two input nodes, two hidden nodes, and one output node. Using this code we are able to show 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 given the assignment to modify the Generation5 code to implement a neural network 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

---

Instructor Initials	Robot Project
	Task 1 - Build a robot. Use *fehmbot* as your model. Remember the body of the robot has to be strong enough and big enough to hold the handy board. Demonstrate that you have a working robot. (5 points) Read pages 773 - 785 in your text Answer the following questions. (Only hand in one set of answers per robot team) 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. (1 point)
	Task 2 - - Have the robot use its photo sensors to follow a path. The path is defined by silver tape on a black background. Use the neural network software you modified to teach your robot how to follow a path. (3 points)
	Extra Credit - Hard code your robot to follow the silver tape path. (No neural net, lots of if-else statements!!!) 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? (1 percentage point added to your overall grade)
	Extra Credit - Have robot move forward until it detects that it bumped into an obstruction.(Use touch sensors). When this happens the robot will turn right and then move backward. When it bumps into an obstruction the robot will turn right and then move forward again. (1 percentage point added to your overall grade)
	Extra Credit - Use decision tree software to teach your robot how to follow a path. Compare the decision tree solution to the neural net solution. Which performed better? What reasoning can you give for any differences? If they performed equivalently, what reasoning can you give for this?(2 percentage points added to your overall grade)

figure 1 Robot Evaluation Sheet

configured with two photo sensors and two motors which it uses to navigate the given path. Depending upon the readings from these sensors, the speed of the motors are adjusted to maintain the robot's position on the line. Students see the direct connection between the topology of the neural network and the architecture of the robot they are to build.

In addition to preparing the neural network code, students are required to read the robotics chapter in Artificial Intelligence A Modern Approach by Stuart Russell and Peter Norvig [6], and answer several questions based on their reading and their robotic experience. The answers are collected with the evaluation sheet, *figure 1*, at the end of the project.

Materials needed

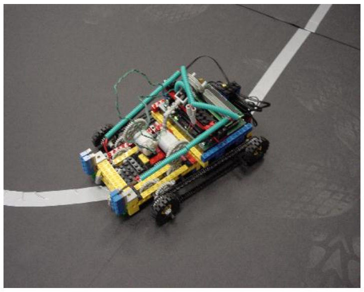
In the past we have used a basic tank architecture for our robots. This past year we experimented on an architecture that replaced the front wheels with angled LEGO pieces. These "skidders" allowed the robot more maneuverability than the tank design. Students are encouraged to build their robots based on these two basic models. The goal is not to spend too much time in robot construction. All student robots must have two photo sensors and two motors. Beyond that, students can be as creative as they wish. Given the two basic models, students have come up with some interesting variations on the basic designs. *figure 2* shows the basic tank model and two student interpretations.

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. Unless otherwise indicated, assume each item will accommodate one robot.

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
8 tooth gear wheel (for ~5 robots)	10.00
24 tooth gear (for ~5 robots)	10.00
LEGO special elements for team challenge (for 4 robots)	16.99
Handy Board	300.00
Photo Sensitive Detectors (2)	3.00
9 volt DC motors (2)	2.00
28 gauge wire (10 feet ~ 4 robots)	5.00
heat shrink tubing (10 feet ~ 10 robots)	1.50
0.1-inch male headers	1.60

The basic Model

Two Student Variations



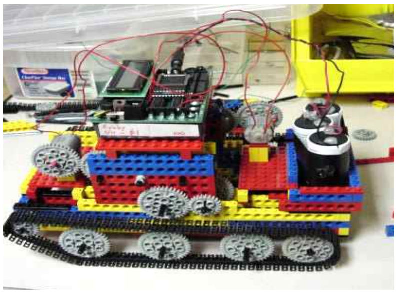


figure 2 - Student and Model Robots

Based on the robot materials list, we estimate the per-robot cost to be approximately $380. The Handy Board was purchased from Gleason Research, http://www.gleasonresearch.com, but they can also be purchased through the KISS Institute, http://www.kipr.org/products/orderform.html. We used Handy Boards without an expansion board. Purchasing boards with an expansion board will add $50 to the price of the robot. Photo sensors and motors were purchased from Jameco Electrical Supply, http://www.jameco.com. LEGO components were from Pitsco LEGO Dacta, http://www.pldstore.com/catalog.cfm. Other supplies used included:

- black presentation board
- silver duct tape
- hot glue and hot glue gun
- solder and soldering iron
- plastic storage boxes for each student robot kit and spare parts

These supplies came to approximately $300. It is important to note that most of this is a one time expense since robots are dismantled and recycled in subsequent semesters.

Sensors and motors are wired as described in [4]. It is important to order motors that have shafts of about 1/2 inch. Motor shafts, generally, are not wide enough to hold gears securely. To widen the shaft, two layers of heat shrink tubing are placed over the shaft. While the second layer is still hot, the gear is placed onto the shaft [4]. We use a 24 tooth gear. This works well with our gear boxes.

Black presentation board and silver duct tape were used in road construction. From experience we found that color contrast was not as significant for accurate photo sensor readings as was the reflectivity of the surface material. We also learned that the robot was able to negotiate turns better when we had "gentle" rather than sharp curves in the road. *Figure 3* shows a picture of a completed road.

figure 3 - Follow the silver tape road

Software used in this project was the Interactive C compiler, available as a free download from the KISS Institute's web page, http://www.kipr.org/ic/ . Interactive C runs on Windows, Linux, MacOSX, and Mac OS7-9 platforms. Our robot lab is currently equipped with IBM compatible, Pentium IV computers using the Windows version.

Since memory on the Handy Board is only 32K, students quickly realized that back propagation wouldn't run efficiently on the robot boards. Therefore, training took place on a desktop computer using a standard C++ compiler. Students quickly saw that the linear equations from neural network training were are all that was able to fit within the board's memory. Hence there was a physical separation between training a neural network using back propagation, and the actual trained neural network. This separation, dictated by the hardware, allowed students to better understand the difference between the two.

In The Lab

The following is the set of instructions given to the students.

Overall Problem Statement: Design and build a robot that will use a neural network to successfully navigate a circular path.

Step 1 - Construction of the Gear Box. Proper gear box construction is the foundation for a properly working robot. If the gear box is too tight, once the weight of the handy board is added to the robot, movement may appear very slow or nonexistent. Detailed step by step instruction on building a gear box can be found at:

http://www.cs.csi.cuny.edu/~imberman/ai/Build Gear Box.htm

Step 2 - Finish robot construction by connecting the two gear boxes and threading the motors onto the gear indicated in the diagram. This is the gear that moves fastest when turning the wheel. Use the model robots as the basis for your robot design. Note, all robots have two motors and two photo sensors located at the "front" of the robot. Make sure your sensors are placed so that they are slightly wider than the duct tape path.

In addition to the working lab models, the basic robot architecture can be viewed at:

http://163.238.35.144/~grant/imberman/robosp01/fehmbot.htm

Make sure you constantly check to see that your wheels move freely as you build your robot.

Step 3 - Programming your robot: Write a simple program, using Interactive C, that has the robot move forward for 1/2 second, turn right and then move backward 1/2 second. Use the Interactive C Manual, http://handyboard.com/software/icmanual/icmain.html, as a reference for the different C commands you will need. Instructions on how to use Interactive C with your robots can be found at:

http://www.cs.csi.cuny.edu/~imberman/ai/startinginteractiveC40.htm

Demonstrate your robot to the instructor once it is finished and moving!

Step 4 - Write an Interactive C program that will display readings from both photo sensors. You will use these readings to create your training examples. Given the way you position the robot on the road, and the sensor readings from the robot, estimate the parameters needed for the left and right motor functions to control the rear wheels. Therefore, each training example will consist of two inputs (the sensor readings), and two outputs (the values passed to the right wheel's motor function and the left wheel's motor function). Please try not to step on the road. Dirt from your shoes may interfere with future sensor readings. You may have to write some small programs and experiment with the robots in order to get a good set of training examples.

Step 5 - Use the modified Generation 5 code created earlier this semester to program your robot with a neural network. Take the trained neural network function, and modify it for use in Interactive C. The training portion of the program is executed in the desktop environment, with the Microsoft Visual C++ compiler. Once you have the output values for the neural network equation, incorporate them into the Interactive C neural network program and try your robot on the road. You may have to repeat steps 4 and 5 several times before your robot reasonably follows the road. One important thing to remember is to make your robot move slow enough so that it has time to take readings from the road and act upon these readings.

Student Evaluation

The robotics project is worth 10% of the student's final grade. Students maintain an evaluation sheet, see figure 1, and after each task is completed, the instructor checks as to its satisfactory completion and grades the task accordingly.

In addition to the required set of tasks, students are encouraged to do extra credit tasks as well.

Discussion

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.

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. 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.

Summary

In this paper we have described a robotics project using LEGO® handy board controlled robots to implement a neural network. We have successfully integrated this project into our artificial intelligence curriculum. All of the web pages referenced in this paper can be accessed from a central source: http://www.cs.csi.cuny.edu/~imberman/ROBOTRESOURCES.htm .

We hope that our description of this project will influence other Artificial Intelligence educators to attempt this with their own students.

Acknowledgments

I would like to thank Miriam Tausner and Christopher Rigby for their inspiration during the development of this project. I would also like to thank Miroslav Krajcow, Orit Gruber, and Chang Guo for their technical help. Thanks to Roberta Klibaner for her suggestions during the writing of this paper.

References

[1] Generation5.org, *Back-Propagation: CBPNet,* http://www.generation5.org/cbpnet.shtml
[2] Interactive C, Kiss Institute, Available WWW. http://www.kipr.org/ic
[3] 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).*
[4] Martin, F., The Handy Board. Available WWW. http://www.handyboard.com
[5] Mitchell, T., *Machine Learning*, The McGraw Hill Companies Inc., 1997 ISBN 0-07-042807-7
[6] Russell, S., and Norvig, P., *Artificial Intelligence A Modern Approach*, Prentice-Hall Inc. 1995 ISBN 0-13-103805-2
[7] Using Interactive C 4.0, Available WWW. http://www.cs.csi.cuny.edu/~imberman/ai/startinginteractiveC40.htm