Neural Networks in Robotic Programming

Fred Kang
NASA SHARP Apprentice
Ames Research Center

Samson Cheung
Mentor
Table of Contents

Intro

Process
1 Introduction

Advanced robots have served many important uses to our society, particularly in the field of exploration. They are able to travel farther, dive deeper and endure harsher conditions than humans can; however, they lack the ability to think. Current advanced exploration robots such as the recently launched Mars rovers are all directed by user input or hard coded commands. Human manned missions have the distinct advantage of the human’s ability to learn from the environment and make decisions based on that learning.

Neural networking is a programming technique that allows a computer to learn to solve problems, instead of being programmed to do so. Neural networking is designed to model the biological neural system. It is composed of very simple neurodes, which model neurons, and are connected by weighted links over which signals can pass.

A neural network learns through training, or exposure to controlled input. The training process consists of the network’s response being compared to the desired response to a controlled input pattern, and the learning process consists of adjusting of the weights to improve the network’s response performance. When training and learning are completed and uncontrolled input patterns are exposed to the network, the network will respond according to its learning with the controlled input patterns. This allows the computer to learn from its surroundings and base decisions on that learning, similar to human behavior.

I am programming an extremely simple robot in the PBASIC computer language with an extremely simple neural network to demonstrate the practicality of neural networking as applied to robotic programming. The definition of practicality is two fold: the computational power of
the neural network, and the economic benefits of the neural network. Through this project I will assess problems of the neural network with robotic programming.

2 Process

2.1 Learning Adaline

The adaline is one of the simplest neural networks. It can only categorize input patterns into two categories that are linearly separable. Its weighted input is calculated like most neural networks with the inclusion of a bias: \( I = X \times W + b \). The adaline has a bipolar signal, which limits it to only be able to separate data into two categories; when weighted input is greater than a given threshold, output is +1, and when weighted input is less than a given threshold, output is −1. For this project I am using a single neuron node adaline network. This makes the neural network extremely simple, yet producing a powerful program from this simple neural network will only make this project more successful.

2.2 Learning PBASIC and Boe-Bot

The Boe-Bot is a complete kit distributed by Parallax Inc, that includes a robot programmed in PBASIC, various input components, software, and a small textbook to teach novices the basics of robotics.
I went through the textbook to learn how to program in PBASIC and get a good feeling for the input components I would be using; this took a very long time as I learned a language from scratch. The Boe-Bot’s main components include servos that run the wheels, touch-sensitive whiskers, photoresistors, and infrared emitters and detectors. Each chapter in the textbook focuses on a specific component the Boe-Bot, and includes pre-written programs to be understood and other small projects to write.

2.3 Neural network on the Boe-Bot v1.0

The object of the neural network will be for the Boe-Bot to distinguish being “interesting” objects and “uninteresting” objects. The objects will be Styrofoam panels with a variable number of lights on them. The chief component for the Boe-Bot to decide between interesting and uninteresting will be the intensity of the brightness of the light as reported to it though the photoresistors, and the distance of the object as reported to it by the infrared emitters and detectors. Note: The simplicity of the adaline network does not limit me to the simplicity of this program; the power of the robot processor and its memory space limits the length of my program and thus its complexity.

Through a neural network, the Boe-Bot will learn to differentiate between interesting and uninteresting objects, and when objects are detected it will act accordingly. This has a distinct advantage over hard coded sequential commands to define “interesting”, because in a place like Mars or 5 miles under the Pacific Ocean, there are so many unknowns that defining “interesting” with a hard coded rule would exclude many objects that did not fit the rule only by a small variation. The adaline solves this problem by generalizing the definition of “interesting” based on its past exposure to “interesting” and “uninteresting” things. This program is analogous to
advanced exploration robots such as the Mars rovers, where a much more powerful processor and a much more complex program will be used to handle many more input components and variables to consider.

This version served as a stepping-stone towards the final program. In writing this program, I had to take measurements and plot the photoresistor readings with a different number of lights and at different distances. I then had to write an interface program that would convert the photoresistor readings (in RCTIME, readings between 0 and 500) to a much smaller scale that would fit an adaline program that I had previously written. This program has no applicable capabilities though, because the interface program is a separate program from the learning program, and the two must be run and used together by a programmer. I also had to decide which input patterns would be “interesting” and which would be “uninteresting”.

2.4 **Neural Network on the Boe-Bot v1.1**