1997

Incorporating a Connectionist Vision Module into a Fuzzy, Behavior-Based Robot Controller

Doug Blank
Bryn Mawr College, dblank@brynmawr.edu

J. Oliver Ross

Citation

Blank, D.S., and Ross, J.O. (1997). Incorporating a Connectionist Vision Module into a Fuzzy, Behavior-Based Robot Controller. Proceedings of the 1997 Midwest Artificial Intelligence and Cognitive Science Society Conference.

Appeared in the *Proceedings of the 1997 Midwest Artificial Intelligence and Cognitive Science Society Conference*.

Incorporating a Connectionist Vision Module into a Fuzzy, Behavior-Based Robot Controller

Douglas S. Blank and J. Oliver Ross
Department of Computer Science, University of Arkansas
232 Science-Engineering Building
Fayetteville, AR 72701, USA
dblank@comp.uark.edu joross@entropy.uark.edu

Abstract

This paper describes the initial steps required to incorporate an artificial neural network vision module into an established fuzzy logic, behavior-based mobile robot and controller. This efficient and robust method is demonstrated to show its effectiveness in simple real-world environments.

Introduction

Although Zadeh defined the basic operations of fuzzy set theory over thirty years ago (Zadeh, 1965), fuzzy logic-based controllers have just recently become the technique of choice for many researchers in robotics. Fuzzy logic controllers allow for the integration of high-level, human-designed plans to operate along side immediate, reactive plans in a robust manner. The key to this successful line of research has been the development of the concept of *behaviors*.

Behaviors

Behaviors as a method of controlling robots were inspired by Brooks' *subsumption architecture* (Brooks, 1986) and later generalized by Maes' descriptions of *behavior-based* designs (Maes, 1993). Generally, a *behavior* is a simple, focused, perceptual trigger and associated action. An example might look like: **IF door-on-left THEN turn-left**. The **door-on-left** is the perceptual trigger that activates the robot to **turn-left**.

The ideas of behaviors and fuzzy logic were a perfect match; behaviors provided an abstract interface for humans to enter plans, and fuzzy logic provided a method for dealing with inexact variable values. A robot under the control of a fuzzy logic, behavior-based rule (such as **IF door-on-left THEN turn-left**) will act in an appropriate manner when **door-on-left** has a value somewhere between absolute true and absolute false. The hope is that general, robust functioning can be attained with just a few simple behaviors mediated by a fuzzy logic controller (Saffiotti, Ruspini, and Konolige, 1993).

Learning

Our goal was to integrate learning into this well-known methodology. Although some success has been reached using fuzzy logic controllers without learning, we believe that learning will be necessary as we require robots to process more complex sensory information, and require them to perform more complex tasks.

One obvious way to incorporate learning into a fuzzy control system is to attempt to learn the fuzzy logic rules. However, this is exactly where human expertise benefits a fuzzy logic controller the most. That is, humans are able to efficiently and effectively write the abstract rules, such as **IF obstacle-on-right THEN turn-left**. We believe that a much better task

Figure 1: The mobile robot with laptop controller and CCD camera.

for a learning module is in the recognition and categorization of the perceptual triggers (i.e., **obstacle-on-right**). Although **obstacle-on-right** may be gleaned directly from low-level sonar sensors, more sophisticated perceptions cannot. For instance, consider the goals of attempting to get close to dogs, but to avoid cats. The human module in the controller can easily create the rules (i.e., **IF cat THEN reverse-motors**). The hard portion of the problem is coming up with the appropriate truth values for the categories **cat** and **dog**.

Our solution to this problem was to insert an artificial neural network between the low-level sensors and the fuzzy logic controller. Artificial neural networks can be trained (via back-propagation (Rumelhart, Hinton, and Williams, 1986) or some other method) to read in low-level sensors, and produce activation values on output nodes. The output values, representing possibly high-level categories (like "cat"), can be used directly (or nearly directly) in a fuzzy logic controller. This methodology is quite general. For example, a network can be trained to read in 2D visual images and sonar readings, and produce output nodes representing the presence or absence of a cat. In addition, such a network's output activation will typically gracefully drop as the input image looks less like a cat. As the output value is treated as a likelihood value in the fuzzy logic controller, this is exactly the desired behavior.

The Robot and Controller

For this research, we chose the low-cost Pioneer 1 mobile robot and Saphira fuzzy logic behavior-based controller from ActivMedia, Inc., in association with Real World Interface, Inc. Our version of the Pioneer 1, called *RazorBot*, has seven sonar sensors and a slightly wide-angle CCD color camera mounted on the front and center of the robot (see Figure 1). RazorBot has an on-board MC68HC11-based micro-controller. RazorBot weighs 17 pounds and measures 45 cm from stern to stern, 36 cm across the wheels, and 23 cm to the top of the console. It has enough battery power to run for a few hours unattached.

For these experiments, a remote PC ran the fuzzy logic controller and captured video signals sent from RazorBot via radio transmitters and receivers. The video was captured by a Matrox Meteor frame-grabber on the PC (see Figure 2). The PC controlled the robot by radio modems.

Saphira, the fuzzy logic controller, is a flexible system designed to handle behaviors as described above.[1] In addition to having purely perceptual triggers, Saphira can also post items to a global blackboard, similar to Erman, Hayes-Roth, Lesser, and Raj Reddy's Hearsay II model (1980).

Saphira is capable of handling hierarchies of behaviors like those defined by Brooks (1986). But, more importantly, Saphira is also capable of smooth behavior *blending*. Behavior blending allows for robust robot control in light of competing perceptual triggers. For instance, if two competing rules were both being activated by their perceptual triggers, Saphira can often produce a behavior that is an intelligent combination of the two.

Our goal was to create an artificial neural network which could be trained in a general manner to categorize objects in the visual field. The output of the network could then be used in abstract fuzzy rules in Saphira.

Connectionist Vision Module

Specifically, our task was to recognize a small purple moving ball in the visual scene, and to help steer RazorBot toward the ball.[2] After some experimentation, we decided that we needed to keep

Figure 2: Image captured with the Meteor video frame-grabber as sent from RazorBot.

[1] See (Konolige, Myers, Ruspini, and Saffiotti, 1996) for a detailed description of Saphira.

[2] This ball is often called a Squiggleball because of its apparent random movements.

Figure 3: An image as seen from the robot's reduced color and pixel perspective (the ball is in the lower center of the scene).

the visual resolution as small as possible while still allowing enough information for object recognition. We settled on an image that was 44 x 48 pixels, with three colors per pixels (RGB). That provided an image that has 6336 data points. This was our input into the network (see Figure 3).

The output was decided to be a simple categorization of which quadrant the ball was in. Therefore, four output units were created, one for each quad.

A feed-forward, fully connected network was then created. Back-propagation was chosen as the training method (Rumelhart, Hinton, and Williams, 1986) (see Figure 4).

Before integration into the fuzzy logic controller, we ran many training tests attempting to learn the classification task. We captured 100 images as seen from the robot, and trained various networks on 90 images, keeping 10 to test generalization ability. Initially, we used a real number between 0 and 1 for each of the 6336 RGB data points. However, no configuration of hidden units and parameter settings allowed the network to learn the task. By rounding off each data point to 0 and 1, the network was then able to learn the task.

The final network contained 10 hidden units. Momentum was set to 0.1, and all learning rates (unit and bias) were set to 0.1. The network learned to identify the quadrant that contained the ball in 147 sweeps through the 90 image corpus. To test generalization, the remaining untrained 10 images were tested, and it got 80% correct. Further training produced better performance on the training set, but produced worse performance on the generalization set. [3]

Now that the network was trained to produce appropriate output categorizations, it was ready to be worked into the fuzzy logic controller.

Integration

To integrate Saphira and the connectionist vision module (see Figure 5), we needed a method to create fuzzy variables that could be used to control RazorBot's motors. We decided to create a fuzzy variable for each quadrant. Each variable would hold the value indicating the network's estimation of the likelihood that that quad contained the ball. One measure of that likelihood was simply the output activation of each of the associated output units.

Let `q1` be the activation of the unit representing the presence of a ball in quad 1. Therefore, one can write behaviors of the form:

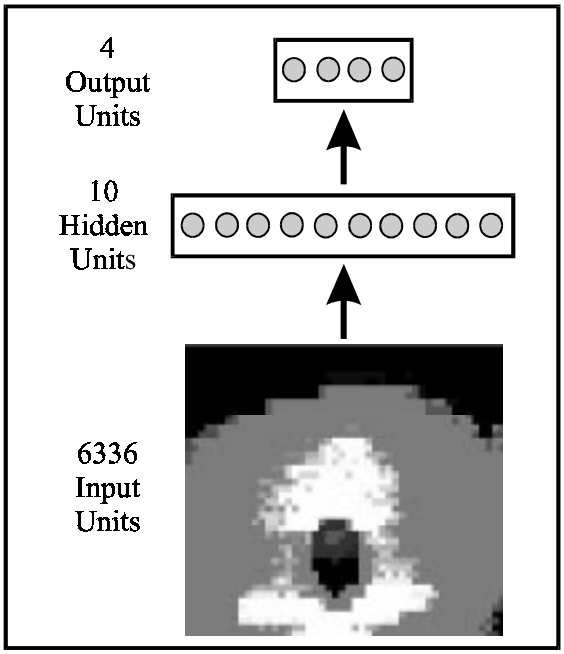

Figure 4: The connectionist network was given images and trained to produce the quadrant containing the ball.

[3] This is a familiar occurrence in training back-prop networks called *over-training.*

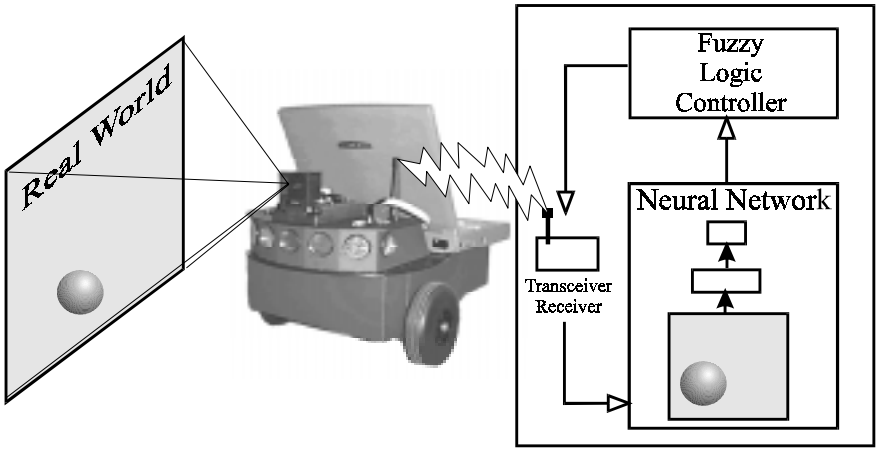

Figure 5: Schematics of processing. The on-board video camera sends an image to the radio receiver. The image is grabbed and sent to the artificial neural network. The ANN propagates the activations through its weights. The outputs are sent to the fuzzy logic controller. Finally, the controller decides the next actions, and sends the commands back to the robot via radio waves.

```
IF (q1 OR q3) THEN Turn Left
IF (q2 OR q4) THEN Turn Right
```

where `q1` and `q3` are the quadrants on the left side of the visual image. Simply using the output values of the network has the advantage of being easy to compute. However, the output activations do not necessarily correspond to likelihood values.

We found it necessary to normalize the output activations prior to using them in Saphira. For example, the output activations for `q1`, `q2`, `q3`, and `q4` might be 0.1, 0.0, 0.1, and 0.4 respectively. Although the network produced only a 0.4 activation on the unit associated with `q4`, this is by far the largest activation of the four output units. However, this activation is not great enough to produce a level of confidence needed to move the robot in Saphira. Scaling the values gives 0.2, 0.0, 0.2, and 0.8, which would be more likely to produce the desired results.

After scaling the output activations, we plugged a routine into Saphira to grab an image and propagate the activations through the network. We then wrote a simple behavior similar to that outlined above so that `q1` through `q4` were set to the scaled versions of the associated output units. The network is small enough and the frame-grabber fast enough to allow for many propagations through the network per second.

The integrated system can successfully follow a ball rolling on the floor, while simultaneously avoiding collisions with walls and other obstacles.

Conclusions

Our initial results are promising, however, critical analysis remains to determine how this methodology compares to other techniques.

In addition, normalization of the output activations is only one method to link network activations to likelihood values. Our simple method does not work in all cases; for instance, if a ball is not present the four quad values are still normalized. Clearly, some type of threshold regulator is needed.

Although the network did correctly categorize 80% of the novel images in the test of generalization, we would, of course, like much better performance. We believe that a much richer training set would help with the generalization, as well as creating output activations that would not need to be normalized. A semi-automatic method of creating a training corpus is being investigated. This method would simply require a human to “drive” RazorBot around, making appropriate moves in response to objects in the visual field. Later, the robot will replay the images and movements, learning to associate one with the other.

Currently, the visual image is probably overly boiled-down. At 44 x 48, the resolution is at the limits for even ball recognition. More complex recognition tasks may require a finer resolution. However, recall that each RGB color value was rounded off to 0 or 1. This in effect gives only 9 different colors (3^2). By allowing three different values for each color data

point (i.e., 0, 0.5, and 1) effectively increases the colors to 27 (3^3). We believe that this should also help recognition and generalization, while keeping the task learnable.

Finally, we would like to add a self-tuning learning component to the fuzzy logic portion of the system. This would allow the system to automatically adapt to neural network-based fuzzy variables.

References

Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, March 1986, 14-23.

Erman, L., Hayes-Roth, F., Lesser, V. and Raj Reddy, D. (1980). The Hearsay II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 28.

Konolige, K., Myers, K., Ruspini, E., Saffiotti, A. (1996). The Saphira Architecture: A Design for Autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*, Special Issue on Architectures for Physical Agents.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems*, Englewood Cliffs, NJ: Prentice Hall.

Maes, P. (1993). Behavior-Based Artificial Intelligence. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 74-83, Boulder, CO.

Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Vol. I.* (J. McClelland and D. Rumelhart, eds.) Cambridge, MA. Bradford Books/MIT Press.

Saffiotti, A., Ruspini, E. and Konolige, K. (1993). Blending Reactivity and Goal-Directedness in a Fuzzy Controller, *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, 134-139, San Francisco, CA.

Zadeh, L. (1965). Fuzzy Sets. *Information and Control*, 8: 338-353.