4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, November 17-19, 2005 (pp40-45)

AIBO Goes to School – Basic Math

S. RADHAKRISHNAN and W. D. POTTER
Artificial Intelligence Center
University of Georgia
111 Boyd GRSC, Athens, GA 30602
USA
http://www.ai.uga.edu

Abstract: - . One of the objectives of AI involves creating applications or behaviors on machines that simulate intelligence. This paper presents an application of Neural Networks and Image Processing where AIBO solves mathematical expressions after learning to recognize numbers and operators via a combination of Offline Learning and Online Testing. This application results in a robot that exhibits intelligent behavior (as perceived by the onlooker) by being able to recognize and solve math problems as humans do and can be used as a platform for other recognition tasks.

Key-Words: - Behaviors, Neural Networks, Offline learning, Online testing, Visual Preprocessing, Image Segmentation, Averaging Operators, Uniform Thresholding.

1 Introduction

Neural Networks are well suited for non-linear problems since the network itself is inherently non-linear [7]. When we consider situations that involve classification and recognition based on visual input, Neural Networks are one of the many different methods that can be used, some others being Bayesian Learning, Support Vector Machines and K-Nearest Neighbor algorithms.

AIBO, the Sony Robot dog (although intended for entertainment purposes) is an excellent platform for robotics research because of its design structure. It provides a robust robotic platform to test algorithms and applications. The ERS220A model shown in Figure 1 (used in our experiments) comes equipped with a CCD camera, one infra-red sensor and various pressure sensitive sensors.

This paper describes the development of an application to teach AIBO to recognize numbers and characters and as an extension of the recognition, to be able to solve mathematical expressions. The idea is to have the robot look around a room for a mathematical expression (which is pink in color for easier identification), take a picture of it and process the image to give the result of the expression. If the robot recognizes a pink non-expression it identifies it as such and continues to look around for an expression. This application uses neural networks as the learning algorithm and image processing techniques to focus on necessary segments of the image. As a prelude to evaluating an expression the robot was trained to identify 13 different shapes.

The motivation behind creating this behavior on AIBO is to extend its role as an interactive entertainment robot and add aspects of intelligent behavior to it. The ability of robots to detect specific objects based on features rather than color alone is an important research area. This paper addresses this research interest by training AIBO to learn numbers and operators based on their features rather than color. This behavior can be presented to young children as an education enhancement tool to encourage them to learn math.

The rest of the paper is divided as follows: Section 2 explains the framework the software platform uses to create the behaviors on AIBO. Section 3 justifies the choice of the learning algorithm used. Section 4 describes the Image Processing tasks that were performed on the image taken by AIBO. Section 5 explains the offline learning and online testing performed using neural networks. Section 6 mentions the math solving algorithm designed for this behavior. Section 7 and Section 8 describe the results observed and the future work intended for AIBO using this application as a platform.

2 Background - AIBO and Tekkotsu

Sony AIBO robots were initially marketed as Entertainment Robots but the features that they provide have made them a successful robotic research tool. Table 1 gives the different features that are present in the ERS220A, the robot used in this project.

Figure 1: AIBO ERS220A

Due to an increasing interest in AIBO on the part of AI researchers, Sony started to actively promote a software development environment for the AIBO called OPEN-R. OPEN-R provides the user with modularized hardware and software while supporting wireless LAN and TCP/IP protocol [13]. Tekkotsu is an application development framework for robotic platforms developed by Carnegie Mellon University as part of a grant from Sony. The framework is designed to handle the routine tasks of OPEN-R so the user can focus on higher level programming using C++ [8]. Our application builds on the existing framework of Tekkotsu to develop a new behavior. Behaviors are defined as applications created by users that run on AIBO [8].

Hardware	Details
384 MHz MIPS processor	
32 MB RAM	
802.11b Wireless Ethernet LAN card	
Memory stick reader/ writer	
20 joints	18 PID (proportional integral derivative) joints with force sensing 2 Boolean joints
9 LEDs	
Video camera	Field of view: 47.8° high and 57.6° wide, resolutions: 208x160, 104x80, 52x40 up to 25 frames per second
Stereo microphones	
Infrared distance measure	Range: 100-900mm
X,Y,Z accelerometers	
8 buttons	2 pressure sensitive, 6 boolean
Sensor updates every 32 ms	4 samples per update

Table 1: Features of ERS-220A

The Tekkotsu framework provides access to the camera on AIBO. It also allows the programmer to control the motors and create behaviors using motions on AIBO. It is possible to control AIBO wirelessly using a peer to peer network connection and the telnet console on a host computer or laptop. Tekkotsu has implemented a GUI called Tekkotsu Mon to access the applications that run on AIBO from a laptop. This facilitates an easier debugging process and better PC-Robot communication.

3 Design Choice

Neural Networks have commonly been employed in classification and pattern recognition problems because of their ability to generalize based on fewer training examples and the tolerance exhibited towards error [6]. The most important reason for selecting Neural Networks as the training algorithm for this application is the portability of these systems from training to testing. Since training and testing are to be conducted on different platforms, it is important to be able to transfer the trained results to the testing platform with ease. Since the trained results of a neural network are a set of weights, they can easily be transferred to AIBO.

4 AIBO Looks and Learns – Image Processing

Image Processing on AIBO has been extensively researched for the RoboCup Tournament where the robotic dogs play soccer. In this tournament AIBO uses color segmentation to identify and detect different objects in its environment [9]. Color Segmentation is one of the most commonly used methods of object detection and identification in robots. The Sony AIBO comes equipped with a color CCD camera and is tuned to detect pink objects, such as the pink ball that comes shipped with AIBO. Our application uses the pink tracking capability of AIBO in detecting the expression in the environment. The colored mathematical expression is placed against a black background for ease of segmentation. The maximum resolution of the CCD camera on AIBO is 352x288. This is the size of the image that our application deals with. AIBO saves an image after it is at the right distance from the expression and is properly aligned to it. The right distance from the expression is determined by the distance from which AIBO can see all the characters in it. This image is later accessed by the image processing algorithm to be classified. The saved image is in the RAW format which essentially is untouched by any compression algorithm. The RAW file format is used in the Tekkotsu platform as an alternative to the JPEG format.

The image processing can be classified into three parts. The first portion of image processing extracts the pixel values necessary for the following steps. The second portion recognizes the pink pixels and creates a bounding box around the character that has to be recognized. The third portion converts a bounding box of a random size to a constant 20x20.

The Image Processing algorithm runs individually on every character in the expression. The entire process is carried out for every character in the image separately. So a mathematical expression like 8 * (5 ^ 3) / 2 is processed one character at a time. Every character is extracted, a bounding box created and then fed to the neural network as a 20x20 input.

4.1 Extraction of Pixels

The first step in image processing is to extract the pixel information from the image so that it can be processed. The image saved to the Memory Stick® is in the RAW format which means the pixels are unchanged from the camera. The color format of the image is YUV where Y stands for brightness, U and V stand for chrominance. U relates to the blue-yellow color components while V relates to the red-green color components of the color image [4]. Since the expressions are all pink in color, it is enough to extract only the V component of the pixel. This method leads to fewer computations on the part of the image processing algorithm. Since the V component contains color information, it is to a large extent independent of varying illumination. This method cannot be recommended for other recognition tasks since a large amount of information is lost by ignoring Y and U components. The V component is extracted and sent to the step where a bounding box is created.

4.2 Creating a Bounding Box

The second step of the Image Processing algorithm creates a bounding box around the character. This is done by segmenting the pink pixels from the non pink pixels and extracting the ones that contain information about the character. Bounding boxes are created separately for every character by looking for a break in the pink pixel information between characters. These bounding boxes are then converted into a constant size of 20x20.

4.3 Dimension Reduction

Neural Networks require a consistent number of inputs across all training examples [6]. This feature constrains the Image Processing algorithm to come use a constant number of pixel values when being fed to the neural network. A reliable number of input values to the neural network was found to be 20x20 or 400 by trial and error. Different algorithms were considered to perform the averaging process to convert a random sized bounding box to a constant 20x20. One algorithm that retained the original shape best was the simple averaging process. The bounding boxes formed for the characters were of arbitrary sizes and to make them all a 20x20 image, averaging operators were used. The bounding boxes were all padded to fit one of 20x20, 40x40, 60x60, 80x80, 120x120, 160x160, 180x180 or 240x240 sized boxes. Now 2x2 or 3x3 averaging operators were applied to them when necessary, to reduce them to 20x20. We did not encounter bounding boxes that exceeded 240 pixels in height or width. Hence sizes over 240x240 were not considered. This is the input to the neural network used to classify the images. Figure 2a shows the image as it is taken by AIBO and saved to the memory stick. This image was originally 352x288 in size. Figure 2b shows the 20x20 input to the neural network after being processed.

Notice that the inputs to the neural network retain the same shape as the original image. Uniform Thresholding has been applied to the pixel values making them high (1) or low (0) where high denotes pink values and low denotes non pink values [1,2]. The same process is carried out for every character of the expression and every 20x20 input of a character is fed to the neural network.

Figure 2a: Original RAW Image taken by AIBO

```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,0,0,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 2b: 20x20 Neural Network input

5 The Learning Process

The process of learning is divided into two parts, Offline Learning and Online Testing. The reason for the two different parts is mainly due to time constraints. It takes a prohibitively long time for the learning process to be performed onboard AIBO's processor. The advantage of having the robot perform real time learning is that it can better react to new circumstances or environments than offline learning allows. But real time learning also increases the interference effect associated with Neural Networks where learning in one zone causes loss of learning in other zones [12]. Since the training examples have been taken under various lighting conditions it well represents the test samples that AIBO might encounter. Hence the disadvanatages that one might face by training offline are offset by the varied training examples. The neural network is not trained to detect rotational variance in images since the numbers and operators are viewed upright only.

5.1 AIBO gets Homework – Offline Learning

Offline Learning and Online Testing are carried out using Neural Networks. The reason for choosing Neural Networks is the ease of portability these systems possess. A learned system can be transferred to any robotic platform by merely transferring a set of weighted numbers. Another reason is that the Sony AIBO runs all the behaviors or applications stored on a Memory Stick® which is 16 MB in size. It is important that the Memory Stick® is not used up entirely for the learning process.

The back propagation neural network has been used extensively for classification systems and in pattern recognition [6, 10]. Hence this neural network was considered with 400 inputs which corresponds to a 20x20 image as shown in Figure 2b. The number of hidden nodes was selected to be 40 based on trial and error. The system has 18 output nodes, one for each character of the mathematical expression. The 18 characters considered are, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, ^, %, (,).

Two sets of training samples were used for the training process. One set of training samples consisted of one character per image as shown in Figure 2a. Another set of training examples consisted of groups of 6 characters each as shown in Figure 3. Three groups of 6 characters each were used as training samples. The network produced high accuracy levels when trained and tested on individual characters. But training on single character images alone and testing on expression string images was not successful and resulted in low accuracy. Since the test images consisted of a group of characters, it seemed rational to make the training examples a group of characters as well. Each group of characters and each individual character had 11 images each as training samples.

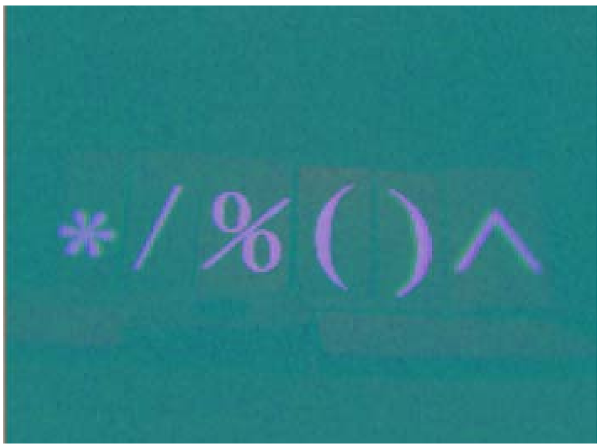

Figure 3: One group of 6 characters used as training sample.

5.2 AIBO takes the Test – Online Testing

Online Testing is a process of using the learned weights and classifying new instances based on these weights. In this testing process, AIBO runs a feed forward neural network which takes an image taken by the camera, processes the image using the onboard image processor and feeds the reduced image to the neural network. AIBO serves as a platform for testing the results obtained from offline learning. Figure 4 shows a typical expression that AIBO can parse. This expression is segmented into characters, the neural network is run on the individual characters and classified numbers and operators are fed to the algorithm that solves math expressions.

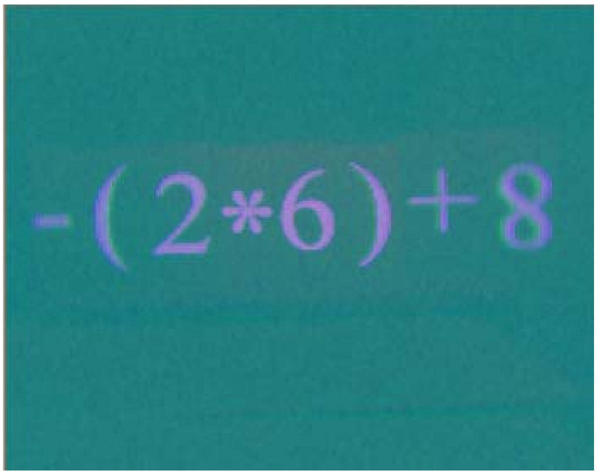

Figure 4: An expression to be solved by AIBO

6 Doing the Math

As a means of testing the entire process, an algorithm was developed to parse and solve the mathematical expression observed by AIBO. After all the characters in the mathematical expression have been identified, they are analyzed by the Expression Solver which determines if the equation is valid. Non-valid expressions include an expression divisible by 0, parentheses mismatch or just improper expressions with two operators next to each other. If the expression is valid, AIBO proceeds to calculate the result which is printed out on the telnet console on the laptop over the wireless network. The accuracy of the results of the Expression Solver depends entirely on the accuracy of the classification process. When AIBO encounters an invalid expression it shakes its head, says the expression is invalid and waits for the user to give it a new expression to calculate.

7 AIBO gets a Grade – Results

The experimental verification of the behavior was performed by letting AIBO explore its surroundings looking for pink equations. This part of the experiment has been adapted from ChaseBallBehavior (going towards pink objects) and CameraBehavior (for taking pictures) of the Tekkotsu framework. If AIBO finds a pink object, it takes a picture of it and determines if it is an equation. If it is not an equation, AIBO moves on and continues searching for other pink objects. Once a pink equation is found, it recognizes the characters and prints the result of the equation on the telnet port.

The minimum average error over 100 epochs for the training samples was observed to be 0.0001. The minimum average error for test samples over 200 events was observed to be 0.0167. These low error values increased the accuracy of the classification process and if positioned with a full view of the expression AIBO always gave the right answers. One hardware constraint observed was the camera on AIBO not being able to handle more than 8 characters in an equation at one time. Panning the head to look for different parts of images has been considered and will be included in future research. When an equation consisted of more than 8 characters, AIBO was not able to correctly identify the characters at the edges of the image.

Alignment and Positioning of AIBO in front of the expression in order for it to see the entire expression are difficult challenges. At times AIBO would need us to intervene and help out with the alignment. The new panning scheme is expected to eliminate the need for any external help.

8 AIBO Graduates – Future Work

The process of object identification has immense potential in the field of Robotics. The ability of robots to identify particular objects based on feature recognition and pattern recognition has applications in fields such as Search and Rescue Missions and Security Monitoring Systems. This application of AIBO can be extended to reading. AIBO can be made to recognize the alphabet and an algorithm can be developed to make AIBO read words and reply to them. Using this application as a base, AIBO has the ability to learn and simulate intelligence in a variety of situations. Also as an extension of this application, AIBO can be taught gestures and can generate responses to them. Our goal is to embed a number of these behaviors on AIBO to reflect a broad spectrum of elementary educational tasks.

These applications can be considered as providing a means of using AIBO as an educational tool exhibiting intelligent behavior to encourage learning among youngsters. Watching AIBO solve mathematical expressions, or reading sentences and replying to them or reacting to gestures can be used to make learning fun for kids. And AIBO is an ideal platform to showcase these tasks because of its status as an entertainment dog.

References:

[1] Jahne Bernd, *Digital Image Processing,* Springer-Verlag, 2002.

[2] Nixon, Mark S., and Aguado, Alberto S., *Feature Extraction and Image Processing*, Newnes Publications, 2002.

[3] A. Grossmann and R. Poli, Continual Robot Learning with Constructive Neural Networks, *Proceedings of the 6th European Workshop EWLR-6 Brighton, England, August 1997*, pp. 95-109.

[4] Miano John, *Compressed Image File Formats*, Addison-Wesley Publications, 1999.

[5] N. Chagas and J. Hallam, A Learning Mobile Robot: Theory, Simulation and Practice, *Proceedings of the 6th European Workshop EWLR-6 Brighton, England, August 1997*, pp. 142-155.

[6] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford UK, 1995

[7] J. Ramesh, K. Rangachar, S. Brian G. *Machine Vision*, New York: McGraw-Hill, p.474-479.

[8] Tekkotsu, http://www.tekkotsu.org

[9] RoboCup, http://www.robocup.org

[10] Shalkoff, R.J., *Pattern Recognition- Statistical, Structural and Neural Approaches*, Wiley and Sons Inc., NY USA, 1992

[11] Trier, O. D., Jain, A. K. and Taxt, T., Feature Extraction Methods for Character Recognition – a Survey, *Pattern Recognition*, 29(4), pp. 641-662, 1996

[12] S. Weaver, L. Baird, and M. Polycarpou. An Analytical Framework for Local Feedforward Networks. *IEEE Transactions on Neural Networks*, 9(3), 1998.

[13] OPEN-R SDK http://www.openr.aibo.com