

An Adaptive “Broom Balancer” with Visual Inputs

Viral V. Tolat¹
Bernard Widrow
Stanford University

Abstract

An adaptive network with visual inputs has been trained to balance an inverted pendulum. Simulation results show that the network is capable of extracting the necessary state information from time sequences of crude visual images. A single linear adaptive threshold element (ADALINE) was adequate for this task. When tested by simulation, the performance achieved was sufficient to keep the pendulum from falling. The adaptive network's ability to generalize made this possible since the training set encompassed only a fraction of all possible states.

1 Introduction

Balancing a broom stick in the palm of one's hand is a fairly complex task from a control point of view. However for those of us with a modicum of coordination it is a seemingly simple task. If asked how we do it, most of us would be hard pressed to come up with an accurate account. An analogous problem is that of balancing an inverted pendulum fixed to a nonstationary platform. If the task of balancing such a system were presented to a person in the form of a video game, that person would probably master the game in a short period of time. Of course to master this game we have available to us the most sophisticated adaptive network around, the human brain.

In this paper we analyze the ability of a simple adaptive network to balance an inverted pendulum. There are several reasons why the inverted pendulum was chosen as a basis for adaptive network research. First, the inverted pendulum problem is a classical control problem which has been extensively studied and is well understood. The inverted pendulum problem is representative of many other

¹This work was supported by the NASA Center for Aeronautics and Space Information Systems (CASIS) under grant NAGW 419.

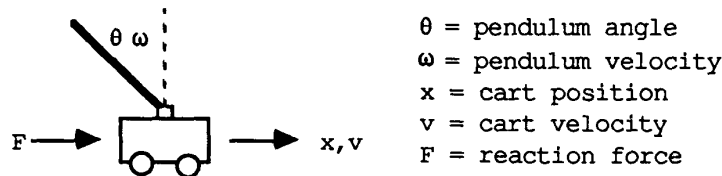


Figure 1: Inverted Pendulum.

control problems and therefore an understanding of how to use adaptive networks to balance the pendulum will allow us to solve other control problems. By using simple visual images of the inverted pendulum, we can examine pattern recognition and vision problems as well. Finally, the complexity of the inverted pendulum is significant enough to make the problem interesting while still being simple enough to make it computationally tractable.

2 Inverted Pendulum

In the inverted pendulum system illustrated in Figure 1, four variables describe the state of the system: the position of the cart (x), the velocity of the cart (v), the angle of the pendulum (θ), and the angular velocity of the pendulum (ω). The force required to stabilize the system at time t is

$$F_t = U \times \text{sgn} [k_1 x_t + k_2 v_t + k_3 \theta_t + k_4 \omega_t], \quad (1)$$

where U is a positive constant representing the magnitude of the force to be applied to the system and the coefficients k_1 , k_2 , k_3 , k_4 , are derived from the physical characteristics of the pendulum system and optimal bang-bang control theory.

If the cart and pendulum velocities cannot be directly measured, they can be estimated by the instantaneous cart and pendulum velocities which can be derived from the current and earlier position measurements. Equation 1 can be rewritten as

$$F_t = U \times \text{sgn} [(k_1 + k_2)x_t - k_2 x_{t-1} + (k_3 + k_4)\theta_t - k_4 \theta_{t-1}], \quad (2)$$

where x_{t-1} and θ_{t-1} represent the cart and pendulum positions at an earlier time, the coefficients k_1 , k_2 , k_3 , k_4 are those of Equation 1, and the time between x_t and x_{t-1} and θ_t and θ_{t-1} is some small fixed interval.

3 Previous Work

The use of an adaptive network to stabilize a mechanical inverted pendulum was first studied by Widrow and Smith[3,7,5]. They demonstrated that a network of one computing element, an ADALINE (Adaptive Linear Element), was capable of balancing an inverted pendulum if the ADALINE input consisted of the four state variables, each encoded with an N -bit linearly independent code, a so called "thermometer code". The force produced by the ADALINE approximated that called for by Equation 1. The network was trained using the Widrow-Hoff LMS algorithm with the output of an optimal controller of the form in Equation 1 as the teacher.

Barto[2] has also studied the problem of balancing an inverted pendulum using an adaptive network. Barto used an input code and computing unit similar to that of Widrow and Smith but for training he used a system based on reward/punishment learning. This learning algorithm was based on one devised by Widrow, Gupta, and Maitra[6] which they had applied to a system that learned to play blackjack.

4 Current Work

To model the inverted pendulum system, we replaced the the mechanical cart and pendulum used by Widrow and Smith with a Macintosh computer simulation and display. The adaptive circuits were replaced by software simulation and the encoded state variable inputs were replaced with spatially quantized binary images of an inverted pendulum. Examples of the quantized images are shown in Figure 2. The images were 5 by 11 binary pixel representations of the cart and pendulum; 65 different images were possible with 5 different cart positions and 13 different pendulum angles. Pictures larger than the 5 by 11 pixel image were not used because they provided no additional information and added computational overhead. Smaller pictures did not provide the resolution needed.

The actual input to the network consisted of two 5 by 11 pixel images, one representing the present visual image and the other representing the visual image at a slightly earlier time. Both images were necessary because the network had to derive the cart and pendulum velocities as well as their positions². With these two images there were 4225 different possible input images (65×65) with 110 bits ($5 \times 11 \times 2$) each.

²The system could be designed to work with only one image by internally reusing the current image with the next image with the aid of delay circuits. The end result would be the same.

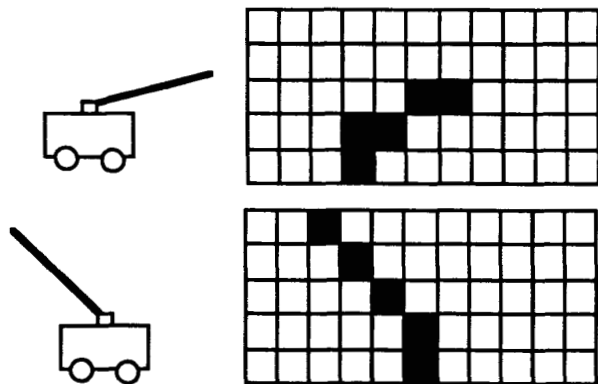


Figure 2: Two examples of Macintosh generated pendulum and cart images. The pictures on the right are spatially quantized images of the pendulum and cart figures on the left.

For the network to balance the pendulum, in some sense it needed to implement Equation 2, estimating the pendulum and cart positions multiplied by the coefficients k_1 , k_2 , k_3 , and k_4 . Finding a set of weights to do this requires solving a system of M linear equations with N unknowns where M is the number of images and N is the number of weights or equivalently the number of bits in the input image. A solution that minimizes the mean-squared-error can be found iteratively by using the Widrow-Hoff LMS algorithm. If an exact solution exists, the LMS algorithm will eventually converge to that solution. There are various methods of calculating the least squares solution directly[4,1].

The LMS algorithm was first simulated using all the possible images for training. The learning curve shown in Figure 3 is a plot of the number of presentations of the entire training set versus the percentage of classification errors. The error drops dramatically after just a few passes through the training sequence, then flattens out. After 1000 passes through the training sequence, the error was 3.4%. If the simulation was allowed to continue, the error would have eventually reached 1.78% which is the error obtained from a directly calculated least squares solution. The weights for the solution after 1000 passes are illustrated in Figure 4.

Next, simulations were done to determine the network's performance when trained on a fraction of the total number of possible images. Training sets of various sizes were picked at random from the set of possible training images. Results were computed for four different training sets of each size and averaged. The final error after 300 passes, computed over all possible images, is plotted as a function of the training set size in Figure 5.

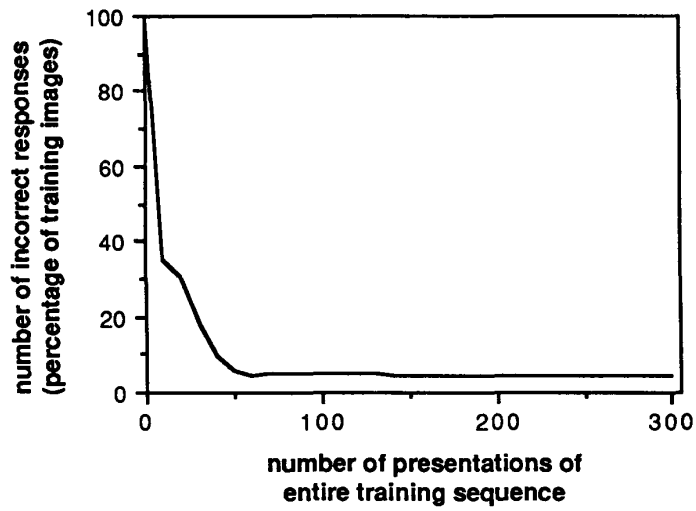


Figure 3: Learning curve.

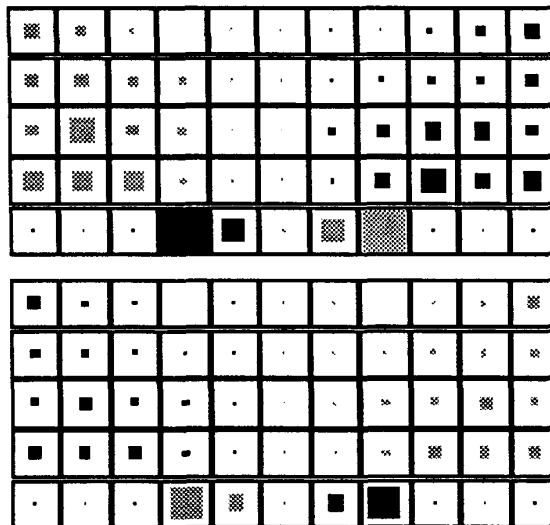


Figure 4: Weights for a network solution using the LMS algorithm. The top 5 rows correspond to the current visual image of the cart and pendulum and the bottom 5 rows correspond to the image at an earlier time. The size of the squares denote the magnitude of the weights. Grey represents negative weights and black represents positive weights.

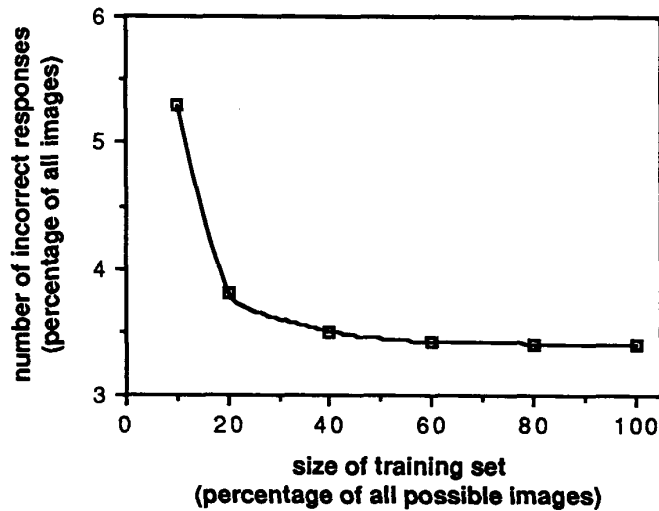


Figure 5: "Generalization" curve.

When the adaptive network solutions were tested with the Macintosh computer simulation, the pendulum was successfully kept from falling in spite of the fact that there were a small number of incorrect responses.

5 Future Work

Although control of the inverted pendulum was achieved by a single ADALINE with visual inputs, more complicated control problems will require larger adaptive networks with appropriate sensory inputs.

The long term objective of this research is to demonstrate that a human expert can impart learned skills to an adaptive network in complex problem environments for which it may not be possible to develop explicit decision rules. This objective leads toward a new form of man-machine interaction in which a person trains a machine to perform a task by having the machine "look over the person's shoulder", observing the environment and the person's responses to the environment. After limited training, the machine should be able to perform the task independently of the person, responding correctly to almost all situations, even those not specifically trained on.

References

- [1] *PRO-MATLAB User's Guide*. The Mathworks, Inc.
- [2] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, smc-13(5), 1983.
- [3] Fred W. Smith. *Contact Control by Adaptive Pattern-Recognition Techniques*. Technical Report 6762-1, Stanford Electronics Lab, Stanford University, April 1964.
- [4] Gilbert Strang. *Linear Algebra and Its Applications*. Harcourt Brace Javanovich, Inc., second edition, 1980.
- [5] B. Widrow. The original adaptive neural net broom-balancer. In *International Symposium on Circuits and Systems*, pages 351–357, IEEE, May 1987.
- [6] B. Widrow, N. K. Gupta, and S. Maitra. Punish/reward: learning with a critic in adaptive threshold systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3:455–465, 1973.
- [7] B. Widrow and F.W. Smith. Pattern recognizing control systems. In *Computer Information Sciences (COINS) Symposium*, 1963.