The Introspective Robot: Using Self-Prediction to Improve Robot Learning

James B. Marshall, Neil K. Makhija, Zachary D. Rothman

Department of Computer Science
Sarah Lawrence College
One Mead Way
Bronxville, NY 10708
{jmarshall, nmakhija, zrothman}@slc.edu

Abstract

We investigate the use of self-predicting neural networks for autonomous robot learning within noisy or partially predictable environments. A benchmark experiment is performed in which a network is trained on a task consisting of a mixture of predictable and random patterns. In addition to learning the task patterns, the network is also trained to explicitly predict the internal representations developed for each pattern as well as the resulting output error. Self-prediction is found to speed up learning and may offer an effective framework for distinguishing predictable from unpredictable input data.

Introduction

Inspired by developmental psychology and neuroscience, the newly emerging field of developmental robotics studies how autonomous robots can learn to function completely on their own in rich, dynamic environments, without relying on innate task-specific knowledge being designed into the system in advance [1]. In such an environment, a robot is constantly flooded with multiple streams of raw, uninterpreted sensory information. To use this information effectively, the robot must have the ability to make abstractions in order to focus its attention on the most relevant features of the world. Based on these abstractions, it must be able to predict how the world will change over time, perhaps as a consequence of its own actions. Most importantly, an autonomous system must be able to decide on its own which tasks to learn, rather than relying on people to tell it what to do. The system should be driven by internally generated motivations that push it to learn progressively higher-level abstractions and more complex predictions [2].

However, some aspects of the robot's environment may not be inherently predictable. A self-motivated robot needs to be able to reliably distinguish learnable from unlearnable features of the world, so as not to waste time and effort trying to learn impossible tasks. How can a robot learn to recognize this difference on its own? One promising idea involves the use of "self-predicting" neural networks to control a robot, in which a network is trained to predict its own output error as well as its own internal hidden representations as it learns to solve a task. In Bayesian terms, this can be viewed as the system giving an estimate of the uncertainty of its output. Under certain conditions, a network can learn a deterministic (and hence predictable) task embedded within a larger unpredictable environment more effectively by using self-prediction [3]. This has the potential of enabling the robot to overcome noise in its input and to avoid being distracted by unlearnable features of the environment, on the basis of analyzing the internal representations created by the system from the robot's sensory data.

Self-Predicting Neural Networks

We study the effects of several different network configurations and parameter settings on self-prediction, using a variant of the XOR problem as a benchmark. Our training dataset is intended to model a partially predictable environment, in which some of the input patterns map to well-defined output patterns, while others map to random patterns that change on every training cycle. The portion of patterns with random targets can be varied in order to model different amounts of background noise in the environment. For example, a dataset with 75% noise might contain a total of 64 six-bit binary input patterns, 16 of which map to fixed target patterns (whose bits are a function of applying XOR to the input bits), and 48 of which map to random and dynamically changing target patterns. The network uses a standard backpropagation architecture with input, hidden, and output layers, augmented with additional layers for predicting the amount of error on the output layer as well as the representations developed internally on the hidden layer. During training,

the hidden layer activation pattern is used as a (moving) target pattern for the prediction layer. Self-prediction can be effectively turned off by using constant target values of 0.5 for the prediction layers.

With a dataset containing 0% noise—*i.e.* all input patterns are predictable and thus learnable—there is essentially no difference in behavior when training with self-prediction on or off. Figure 1 (top) shows the sum squared error of the network as a function of training epochs for 10 runs with self-prediction on (grey lines) and 10 runs with it off (black lines). However, as the proportion of unpredictable patterns in the dataset is increased, the effect of self-prediction becomes more noticeable. Figure 1 (bottom) shows the same experiment with a dataset of 75% unpredictable patterns. The network is able to learn the other 16 predictable patterns faster and more effectively with self-prediction on (grey), even when faced with randomly changing targets for the other patterns. Furthermore, analyzing the internal hidden representations created by the network during training with self-prediction on reveals that they are clustered according to whether the patterns represent predictable or unpredictable inputs.

Applying Self-Prediction to Robot Control

We also investigate the effects of several types of self-predicting neural networks within the context of a simple robot prediction task, in which a simulated "watcher" robot, equipped with a camera, learns to anticipate the movements of a "decoy" robot exhibiting various levels of behavioral complexity and predictability. The decoy's behavior can range from trivially predictable (*e.g.* do nothing) to simple periodic motion (*e.g.* pacing back and forth, moving in a circle) to more complicated types of behavior (*e.g.* avoiding obstacles, random wandering). The watcher robot is controlled by a Simple Recurrent Network [4], which is trained to predict the movements of the decoy robot as well as the discrepancy between the observed movements and the network's own prediction of those movements.

We compare the internal hidden representations created by the watcher network as a result of observing predictable versus unpredictable behavior in the decoy robot, and examine the degree to which such behaviors can be distinguished on the basis of these representations. Several different sensory input representation schemes for the watcher robot are considered, including a discrete linear spatial representation and a continuous 2-D coordinate representation.

References

[1] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, 2001. Autonomous mental development by robots and animals. *Science*, 291, 599-600.

[2] D. Blank, D. Kumar, L. Meeden, and J. Marshall, 2005. Bringing up robot: fundamental mechanisms for creating a self-motivated, self-organizing architecture. *Cybernetics and Systems*, 36(2), 125-150.

[3] D. Blank, J. Lewis, and J. Marshall, 2005. The multiple roles of anticipation in developmental robotics. *AAAI 2005 Fall Symposium: From Reactive to Anticipatory Cognitive Embodied Systems*, pp. 8-14. Menlo Park, CA: AAAI Press.

[4] J. Elman, 1990. Finding structure in time. *Cognitive Science, 14*, 179-211.

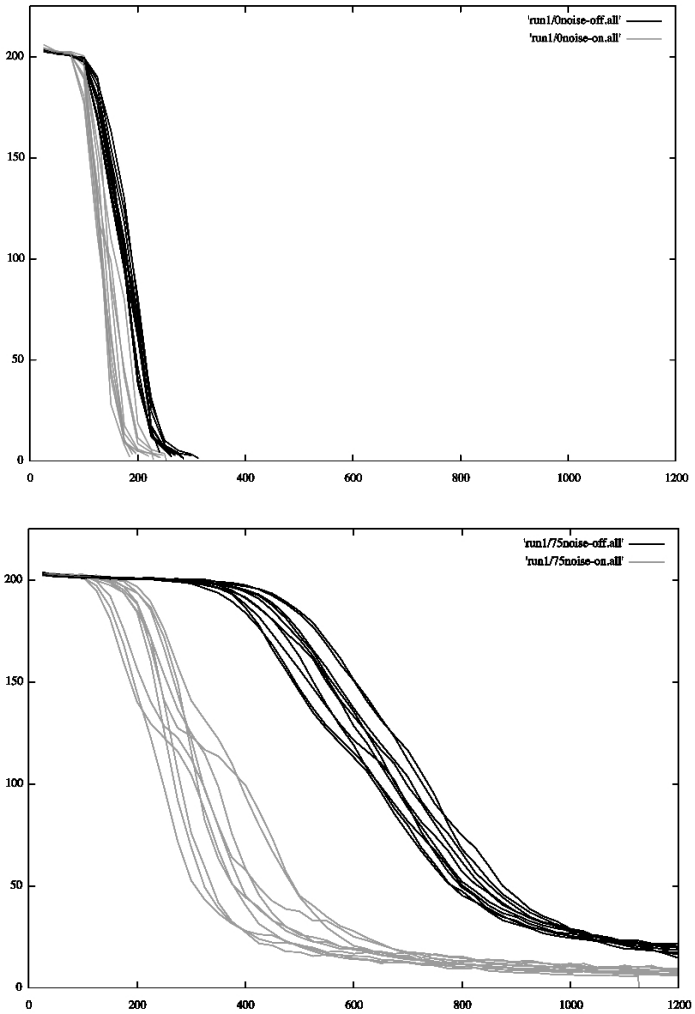

Figure 1. Ten training runs with self-prediction on (grey) and ten runs with self-prediction off (black). X-axis shows training epochs; y-axis shows sum squared error (for predictable patterns only). Top: dataset contains 64 predictable, 0 random patterns (0% noise). Bottom: dataset contains 16 predictable, 48 random patterns (75% noise).