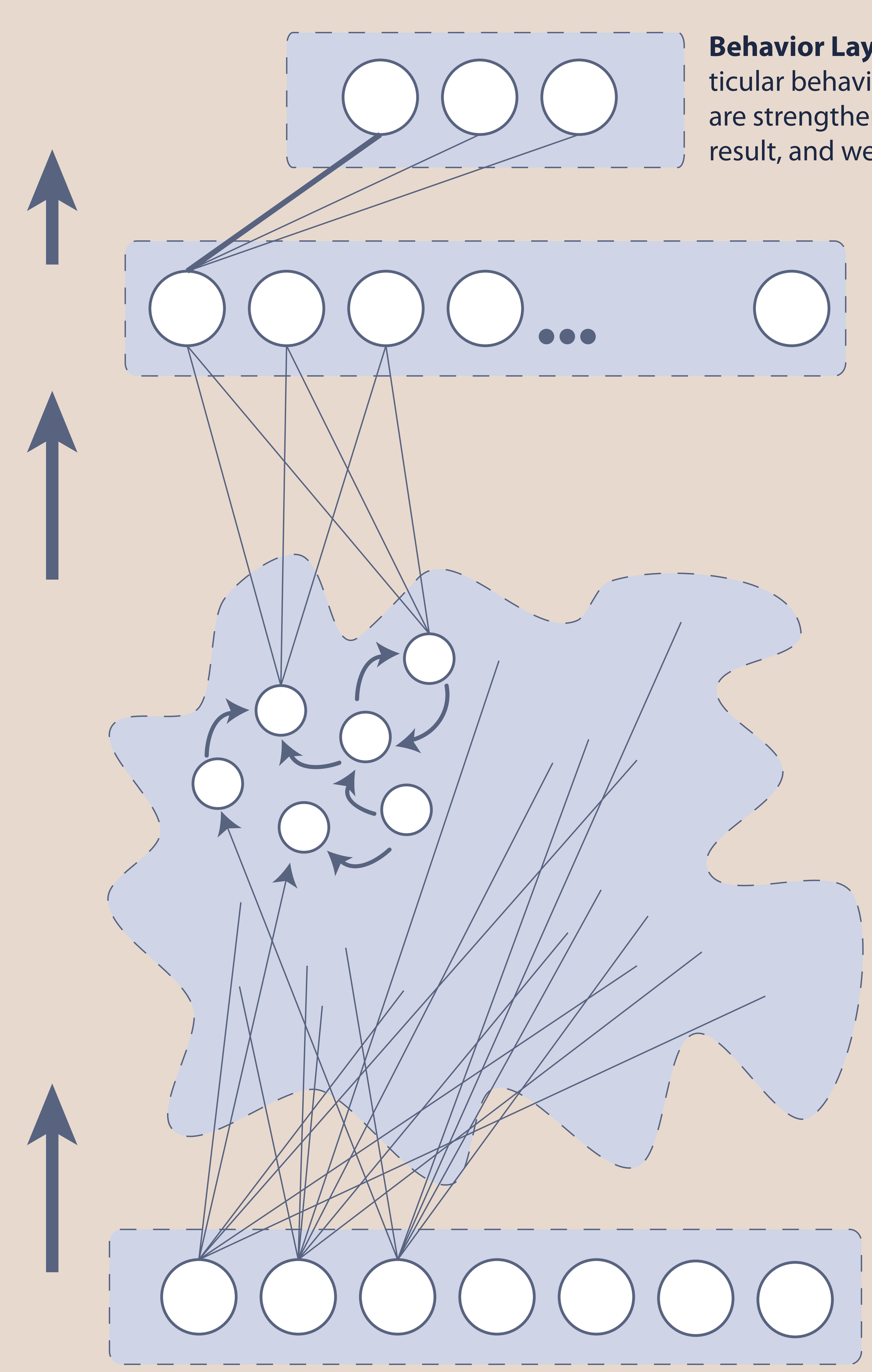
Predictive Category Learning in Mobile Robots

Josh de Leeuw ‘08, Jackie Kory ’11, Ken Livingston

Cognitive Science Program, Interdisciplinary Robotics Research Laboratory, Vassar College

A key problem for artificial intelligence (AI) researchers is how to give AI systems the ability to learn on their own so that programmers don’t have to hard code the knowledge needed to function in a complex environment. Recent theory proposes that the ability to make good predictions based on previous and current information is at the heart of this capacity to learn. We have been adapting a particular class of recurrent neural networks to implement this approach. The Predictive Category Learner (PCL) allows an autonomous robot to learn how to predict the events it will encounter, thus enabling the development of the ability to navigate successfully.

The Predictive Category Learner Architecture



Behavior Layer: Each node in this layer codes for a particular behavior. Connections from the memory layer are strengthened when the behavior has a positive result, and weakened when the result is negative.

Memory Layer: These nodes are fully connected to the reservoir layer. Each node represents a particular pattern of activity in the reservoir. This layer can adaptively add new nodes to account for new, important patterns in the reservoir.

Reservoir Layer: This layer implements a reservoir network. Reservoir networks are sparsely connected recurrent networks with fixed connection strengths. The reservoir gives the network the ability to discover patterns that occur through time.

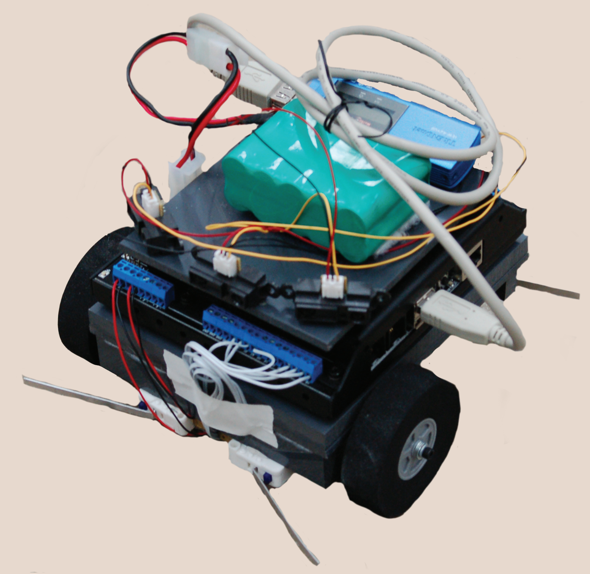
Input Layer: Input to the network is coded by nodes in this layer. These nodes are randomly and sparsely connected to the reservoir layer.

Acknowledgments

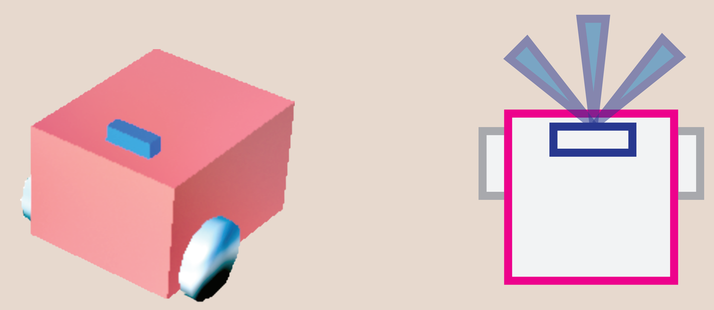
This research was supported in part by grant number BCS-0320764 from the National Science foundation, and by the Vassar College URSI Program.

We also want to thank our colleagues in the Interdisciplinary Robotics Research Lab for stimulating and valuable conversations about all things robotic: John Long, Marianne Porter, Tom Ellman, Hannah Rosenblum, Andres Gutierrez, Sonia Roberts, and Jonathan Hirokawa.

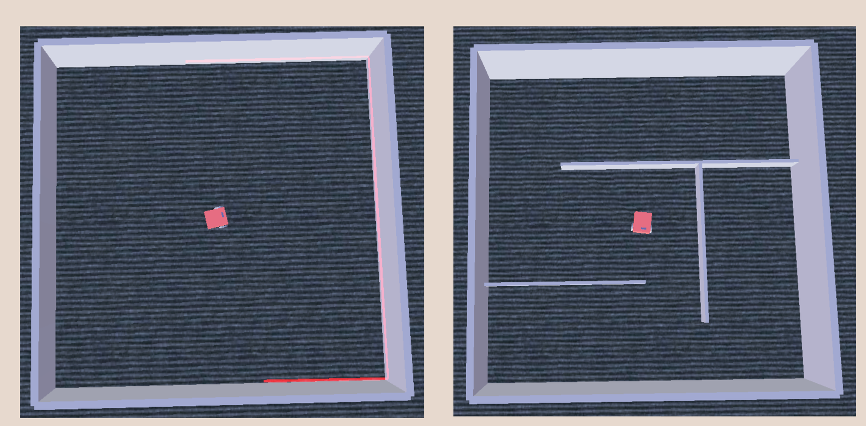
Experiment #1 - Learning to Navigate



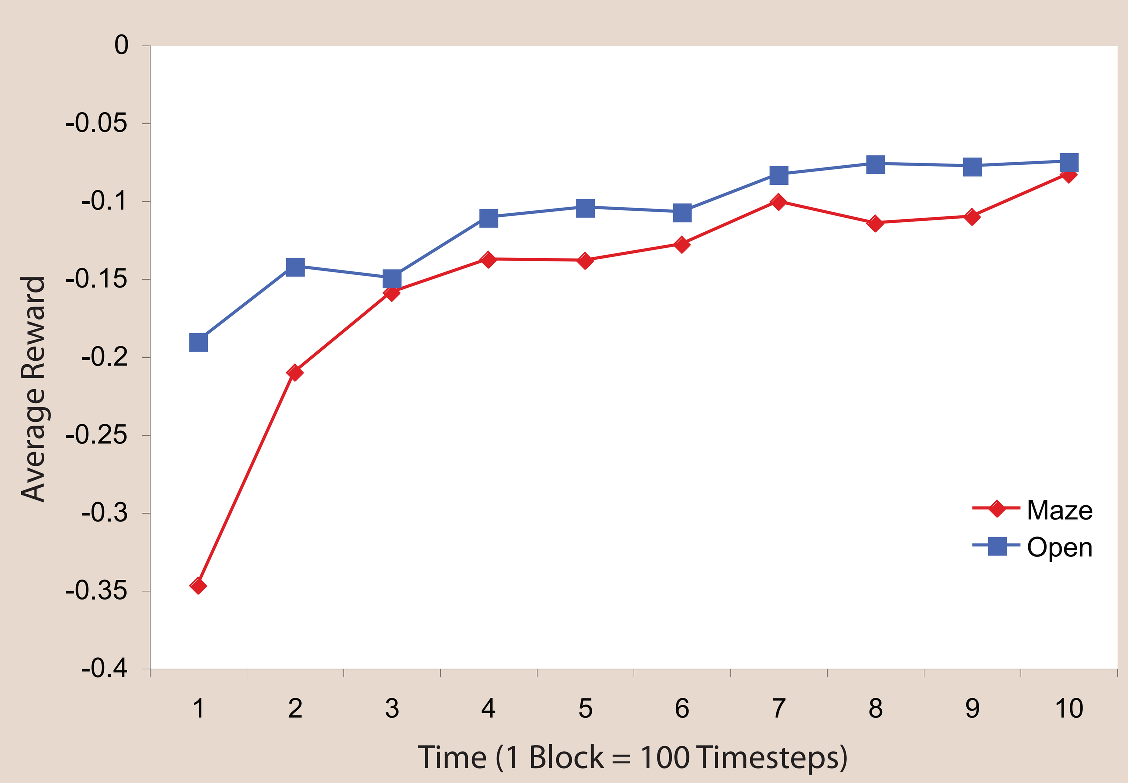
The Real Robot: Though this experiment was conducted in simulation, we modeled a robot in our lab so that we could compare our results with future work using this robot. The robot, Qwerty, consists of a Qwerk microcontroller, front and rear bump sensors, and three IR sensors that can detect objects between 8-80cm from the robot. The robot is WiFi capable and can be controlled from another computer.



The Simulation: We modeled the robot using Microsoft Robotics Developer Studio. Our model has the same dimensions, wheel placement, and sensor placement as the real robot. We carefully modeled the IR sensors of QWERTY to ensure that the field of view and distance of detection were closely approximated.



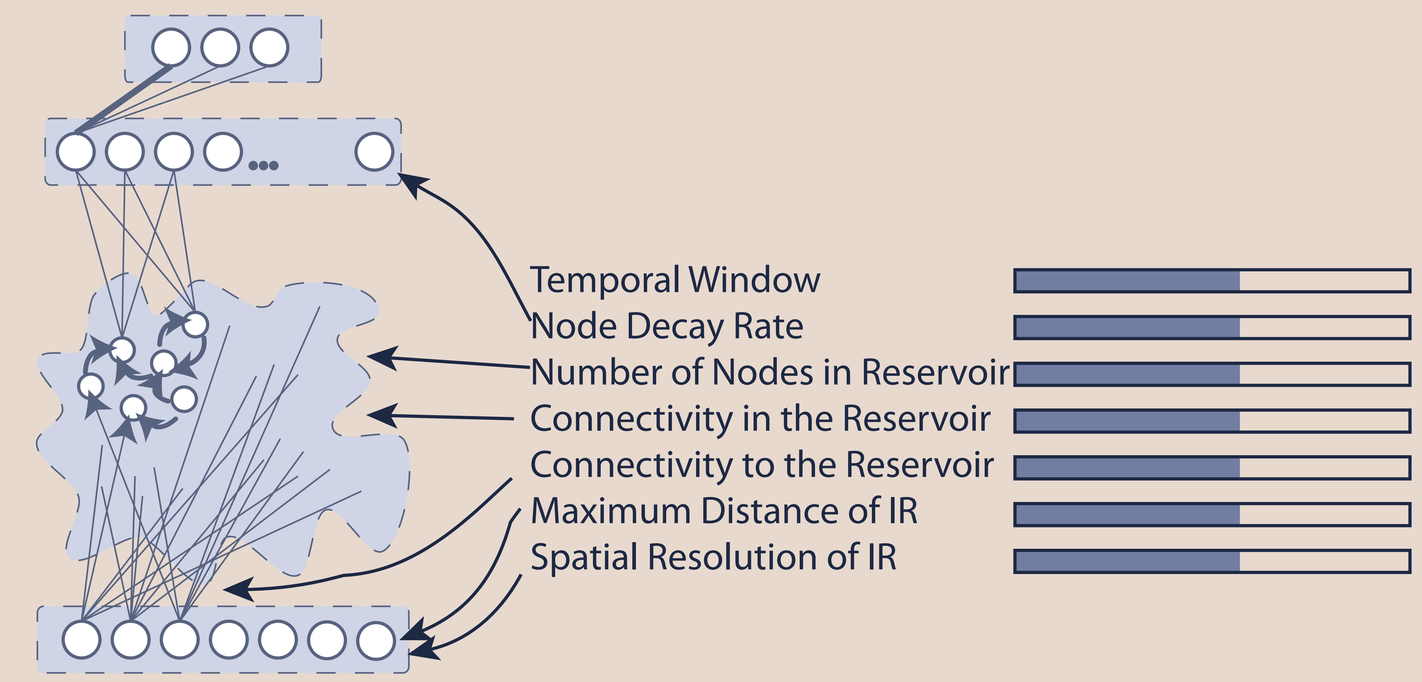
The Experiment: We implemented the Predictive Category Learner (PCL) architecture on our simulated robot and exposed it to two different environments. The first 20 simulated robots ran in the open environment (left); the next 20 learned to navigate a more complex maze (right). The connection pattern in the reservoir layer of the PCL network was randomly generated for each trial. Robots could move forward and backward, turn in place or stop. Moving forward and backward was rewarded slightly more than turning in place. Stopping and colliding with the walls of the environment were punished.



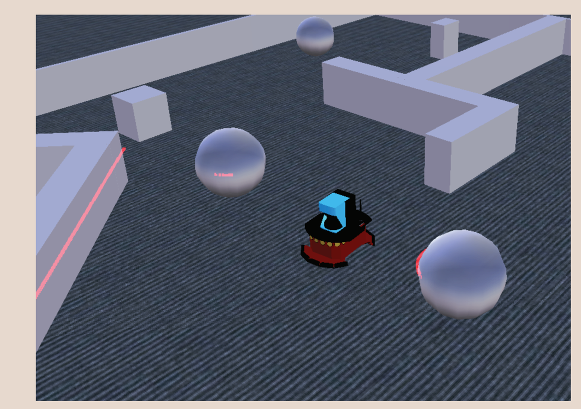
The Result: We calculated the average reward in 10 blocks of 100 timesteps for each trial. We plotted the average reward over time and found that in each trial the robot learned to move in a pattern that returned the greatest reward. Learning was effective in both the simple and maze environments. These results compare favorably with our initial work on the PCL architecture.

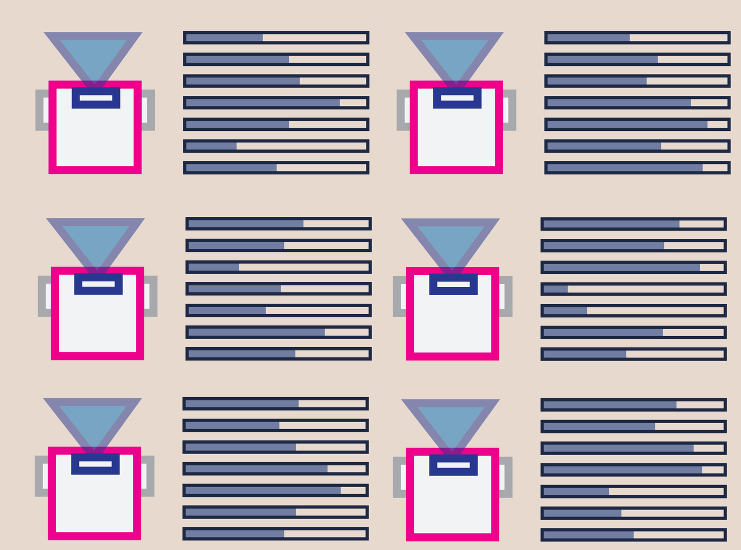
Experiment #2 - Genetic Algorithm

Genome Coding: The Predictive Category Learner (PCL) architecture has a number of properties that can be varied. Our second experiment attempted to identify the optimal values for these properties through the use of a Genetic Algorithm. Each robot individual was coded with a genome with seven traits (below). In each generation, we simulated 20 individual robots; the genomes of the starting population were determined by taking the median value of each of the traits and randomly mutating that value within 25% of the median.



Testing: Each individual in the population was then run for a fixed number of time steps in a complex maze environment that included moveable objects (right). At each timestep the PCL network made a prediction about whether or not the robot would collide with an object in the next timestep. We recorded the percentage of correct predictions and the percentage of false alarms (predicting a collision when there was none).





Mating: The six individuals with the greatest ratio of correct predictions to false alarms were each allowed to contribute “gametes” to a mutual gene pool which contain exactly half of the genetic code. The gametes were then randomly mutated and combined to form a new generation of individuals. We repeated this process for 15 generations.

Results: Performance improved across the 15 generations of evolution, from an average fitness (hit rate / false alarm rate) of 1.09 to 1.71. Furthermore, there were many interesting trends in the evolution of the genome (see table, right) that may offer clues as to what properties are most important in the PCL architecture.

Property	Generation 1	Generation 15
Temporal Window	1130 steps	2500 steps
Node Decay Rate	484 steps	516 steps
Number of Nodes in Reservoir	95 nodes	85 nodes
Connectivity in the Reservoir	15 percent	12 percent
Connectivity to the Reservoir	14 percent	10 percent
Maximum Distance of IR	400 cm	190 cm
Spatial Resolution of IR	179 degrees	131 degrees
Fitness (Hit Rate / False Alarm Rate)	1.09	1.71