Towards a Robot Learning Architecture

Joseph O'Sullivan*
School Of Computer Science,
Carnegie Mellon University,
Pittsburgh, PA 15213
email: josullvn@cs.cmu.edu

Abstract

I summarize research toward a robot learning architecture intended to enable a mobile robot to learn a wide range of find-and-fetch tasks. In particular, this paper summarizes recent research in the Learning Robots Laboratory at Carnegie Mellon University on aspects of robot learning, and our current work toward integrating and extending this within a single architecture. In previous work we developed systems that learn action models for robot manipulation, learn cost-effective strategies for using sensors to approach and classify objects, learn models of sonar sensors for map building, learn reactive control strategies via reinforcement learning and compilation of action models, and explore effectively. Our current efforts aim to coalesce these disjoint approaches into a single robot learning agent that learns to construct action models in a real-world environment, learns models of visual and sonar sensors for object recognition and learns efficient reactive control strategies via reinforcement learning techniques utilizing these models.

1 Introduction

The Learning Robots Laboratory at Carnegie Mellon University focuses on combining perceptual, reasoning and learning abilities in autonomous mobile robots. Successful systems have to cope with incorrect state descriptions caused by poor sensors, converge with a limited amount of examples since a robot cannot execute millions of trials, interact with the environment both for efficient exploration towards new knowledge and exploitation of learned facts, while being robust.

The ultimate goal of the research undertaken in the laboratory can be stated as to achieve a robot that continuously improves its performance through learning. Such a robot must be capable of an autonomous existence during which its world knowledge is continuously refined from experience as well as from teachings.

It is all too easy to assume that a learning agent has unrealistic initial capabilities such as a prepared environment map or perfect knowledge of its actions. To ground our research, we use a Heath/Zenith Hero 2000 robot (named simply "Hero"), a commercial wheeled mobile manipulator with a two finger hand, as a testbed on which success or failure is judged.

In this paper, I describe the steps being taken to design and implement a learning robot agent. In **Design Principles**, an outline is stated of the believed requirements for a successful agent. Thereafter, in **Learning Robot Results**, I summarize previous bodies of work in the laboratory, each of which investigate subsets of our convictions. Our current work drawing together these various threads into a single learning robot architecture is presented in **Cohesive Robot Architecture**. In **Approach and Fetch – a Demonstration**, I present a prototype system which demonstrates that performance *can* improve through learning. Finally, I summarize the current research, pointing out limitations and the work that remains.

2 Design Principles

What is required of a learning agent? Mitchell has argued[Mitchell, 1990] that a learning agent can be understood in terms of several types of performance metrics:

- **Correctness** The prediction of the effects of its actions in the world must become increasingly better for completing a task.
- **Perceptiveness** Increasingly relevant features which impact its success must be constructed.
- **Reactivity** The time required to chose "correct" actions must become increasingly quicker.
- **Optimality** The sequence of actions chosen must be increasingly efficient for task completion.

In addition, an agent must be **effective**, that is, have appropriate sensors and actions to be capable of performing tasks.

*This research was sponsored in part by the Avionics Lab, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, Arpa Order No. 7597. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

3 Learning Robot Results

Individual systems have been developed that explore various facets in the creation of a learning robot agent.

Learning, planning, even simply reacting to events, is difficult without having reliable knowledge of the outcomes of actions. The usual human-programming method of defining "what actions do" is both inefficient and often ineffectual. Christiansen analyzed conditions under which robotic systems can learn action models allowing for automated planning and successful execution of strategies[Christiansen, 1992]. He developed systems that generated action models consisting of sets of funnels where each funnel mapped a region of task action space to a reduced region of the state space. He demonstrated that such funnels can be acquired for continuous tasks using negligible prior knowledge and that a simple planner was sufficient for then generating plans when the learning mechanism is robust to noise and non-determinism and the planner is capable of reasoning about the reliabilities associated with each action model.

Once action models have been discovered, sensing to decide which action to take can have varying costs. The time it takes a physical sensor to obtain information varies widely from sensor to sensor. Hero's camera, a passive device, is an order of magnitude faster than active sensing using a wrist mounted sonar. Yet, sonar information is more appropriate than vision when ambiguity exists about the distance to an object. This lead Tan to investigate learning cost-effective strategies for using sensors to approach and classify objects[Tan, 1992]. He developed a cost sensitive learning system called CSL for Hero which, given a set of unknown objects and models of both sensors and actions, learns where to sense, which sensor to use, and which action to apply.

Learning to model sensors involves capturing knowledge independent of any particular environment that a robot might face while learning typical environments in which the robot is known to operate. Thrun investigated learning such models by combining artificial neural networks and local, instance-based learning techniques[Thrun, 1993]. He demonstrated that learning these models provides an efficient means of knowledge transfer from previously explored environments to new environments.

A robot acting in a real world situation must respond quickly to changes in its environment. Two disparate approaches have been investigated. Blythe & Mitchell developed an autonomous robot agent that initially constructed explicit plans to solve problems in its domain using prior knowledge of action preconditions and postconditions[Blythe and Mitchell, 1989]. This "Theo-agent" converges to a reactive control strategy by compiling previous plans into stimulus-response rules using explanation based learning[Mitchell, 1990]. The agent can then respond directly to features in the environment with an appropriate action by querying this rule set.

Conversely, Lin applied artificial neural network based reinforcement learning techniques to create reactive control strategies without any prior knowledge of the effects of robot actions[Lin, 1993]. The agent receives from its environment a scalar performance feedback constructed so that maximum reward occurs when the task is completed successfully and typically some form of punishment is presented when then agent fails. The agent must then maximize the cumulative reinforcements, which corresponds to developing successful strategies for success at the task. By using artificial neural networks, Lin demonstrated that the agent was able to generalize to unforeseen events and to survive in moderately complex dynamic environments. However, although reinforcement learning was more successful than action compilation at self-improvement in a real-world domain, convergence of learning was typically longer and the plans produced at early stages of learning were dangerous to the robot.

Another issue in robot learning is the question of when to exploit current plans or to explore further in the hope of discovering hidden shortcuts. Thrun evaluated the impact of exploration knowledge in *tabula rasa* environments, where no *a priori* knowledge, such as action models, is provided, demonstrating the superiority of one particular directed exploration rule, counter-based exploration[Thrun, 1992].

4 A Cohesive Robot Architecture

Following the individual lessons learned from each of these approaches, it remains to scale up and coalesce each of these systems into a single robot architecture. In addition, whereas previously the laboratory robot relied upon sonar for perception, we require an effective robot agent to include a visual sensor, both for speed of sensing (reaction times approaching $\frac{1}{30}$ second are possible) and an increase in discrimination capabilities. An architecture is created based upon the following components (see **figure 1**):

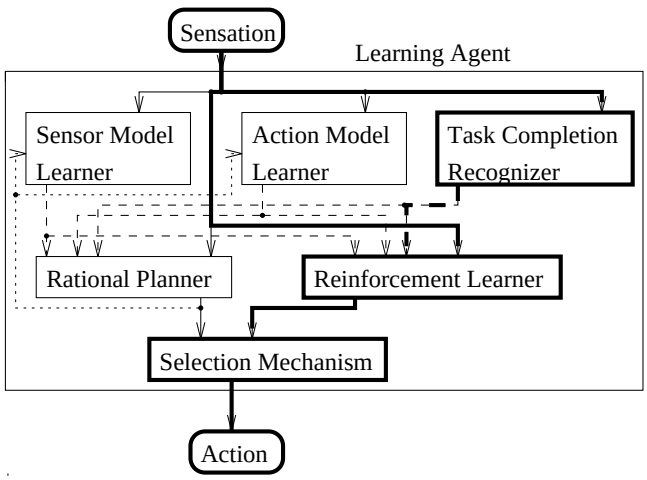

Figure 1: **Schema of Robot Architecture** The bold lines have been implemented in the prototype "approach and fetch" system.

- An action model learning mechanism which compiles domain independent knowledge relating to the expected effect of each action on the sensation of the world.

- A sensor model learning mechanism which develops awareness of domain independent characteristics of each sensor.
- A reinforcement learning mechanism which performs the basic action compilation and strategy generation utilizing action models, sensor models, raw perception and a notion of history.
- A deliberative planning mechanism which utilizes action models, sensor models, raw perception and a notion of history to plan effective strategies.
- A selection mechanism to arbitrate between the reactive and rational strategies. Since the reactive strategy will respond faster, the function of this mechanism will be to act on the reactive strategy unless uncertainty exists, in which case more costly planning using the deliberative component will be required.
- A task completion recognizer which functions as a reward provider to the strategy mechanisms.

Both sensor and action models rely on the same form of input, namely an internal history of previous sensations and actions over which learning occurs. The rational planner can actively interrogate these models to predict the effects of various action sequences, whereas the reinforcement learner only operates upon the immediately expected sensations. The task completion recognizer can be visualized as a type of action model that predicts when the completion strategy can be executed blindly.

5 Approach and Fetch – a Demonstration

A prototypical system has been developed that encompasses a subset of the architectural requirements.

"Hero" has been enhanced with a fixed head mounted wide angle monochrome camera, while still retaining its head and base sonar as described previously[Lin *et al.*, 1989]. No control exists over the type of sensing action taken and so the cost-sensitive learning approach has been sidestepped at the drawback of more expensive learning strategies. In addition, no deliberative planning component has been included in the system. Since this restriction severely affects initial exploration and thus time to convergence, our promise of autonomous operation has been relaxed slightly, with a teacher being allowed to provide such examples as could be garnered from a deliberative planner.

In order to reduce the dimensionality of the input space, a foveated vision preprocessor filters each picture from the camera through a segmented retina which has high information content near the center of field of view, smoothly decaying to coarse information at the edges of the scene. This has the double advantage of allowing the learning agent to learn in a lower dimensional space (approximately 300 multi-valued inputs – vision and sonar combined – are now used to represent each image point in the perception space) and of speeding up the execution time of the learning algorithm by an order of magnitude, an important factor in a real-world system.

As an initial goal, the robot agent is required to perform a simple find and fetch task. A cup is placed on the floor in the laboratory with the open end facing upwards. The robot then must locate, move towards the cup and execute a blind grasp that is successful at picking up cups placed within a narrow region.

A control policy that performs a similar task has been shown to be learnable using just a sonar sweep for perceptual input and limited discrete actions[Tan, 1992]. The same research also showed this to be a difficult task to learn principally due to the narrow reward region. Exploration costs to find the narrow reward band are prohibitive in our domain and thus a teacher is introduced which provides to the agent a limited number of task completion sequences from several different initial states.

An artificial neural network is used to teach the agent when the goal (task completion) has been reached. Prior to autonomous operation, the object to be approached is placed in front of the robot in a number of varied positions from which grasping would be successful. It is also placed in a number of "near miss" positions and this total set of instances is used to train the task completion detector. Note that since the agent is mobile, a perfect model of the object is not required in all positions. All that is necessary is that a subset of possible task completion states are successfully recognizable. The task of the learning scheme then becomes to develop an active sensing strategy that will correctly enter this subset.

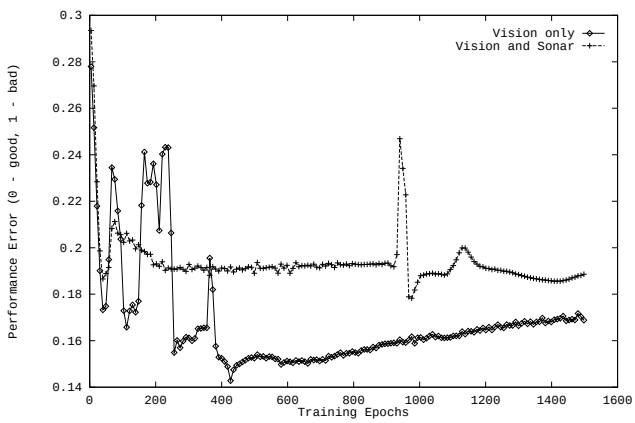

Figure 2: **Object Generalization** Generalization curves averaged from three trainings of the task completion recognizer. At peak generalization the models learned are 85% successful. The graph was produced by evaluating each model on a hold-out set of objects in novel positions. In one series the task completion learner used just visual data as input. In the other series the visual data is combined with sonar data in the hope of synergistic sensor fusion. Note that when the less reliable sonar sensor was added to the vision data, generalization performance decreased. Such results were typical across various network configurations.

Due to the high dimensionality of the world feature space, generality is essential in the reinforcement

learning mechanism. Even though the task has a two-dimensional substructure, 256^{300} possible states exist straining both the representational powers of traditional learning schemas and the practicality of waiting for these states to be explored. An adaptation of Lin's connectionist reinforcement learning strategy[Lin, 1993] is being used to solve this problem. By providing the system with lessons of successful strategies from the start, the agent can be expected to act in a purposeful manner.

The prototype system has been trained in a supervised fashion by placing a "fast-food" cup in various positions in the robot's field of view. Each of these perceptions was then hand classified to specify whether or not that particular position denoted task completion, namely the cup being center in the field of view. In total, 300 different positions were used for training and a further hold-out test set of 50 examples were used for cross-validation (see **figure 2** for results).

This reward provider is currently being utilized in the reinforcement system of our learning agent. The agent is expected to learn, in a realistic time frame, this approach and fetch task. This will demonstrate that autonomous systems may be developed which improve performance by learning and generalization in a real world robotic domain. Current preliminary indications, when the actions of the robot are restricted to just rotations, so as to allow for ease of autonomous operations, indicate that our hopes for the more complex system may be justified.

6 Summary, Limitations and Future Work

A cohesive robot architecture has been presented the goal of which is to illustrate that continuously improving performance through learning is possible. A prototype system is being developed which will:

- create task-dependent feature representations for use in learning and strategy matching. The features produced are represented in the hidden units of the reinforcement learning mechanism and are directly related to the reward as provided by the task description.
- discover robust active strategies for completion of the task despite errors and uncertainty invalidating portions of the policy.
- represent the reinforcement function with a fixed computation cost, leading to a constant response time. Quantitative statements can therefore be made about the reactivity of the system.
- most importantly, learn to produce increasingly correct decisions without assumptions as to the correctness of previous strategies (although having nearly correct strategies speeds up convergence time).

There are a number of extensions to the prototype system that remain to be incorporated.

- The agent suffers from perceptual incompleteness [Chrisman *et al.*, 1991]. Consider a case when two similar objects in the robot's field of view. One object is examined and found not to be the object of the find and fetch task. Then, as attention is switched to the other object, the reactive control strategy will forget that the first object was visited because no state information is store and so will demand that it be re-examined. Lin has addressed these problem using memory-based reactive learners[Lin and Mitchell, 1992], but these have not yet been added to the basic architecture.
- Neither action nor sensor models are currently being utilized in the prototype system, yet they offer assistance with tabula rasa learning. Currently the reinforcement learner encompasses the learning of environment specific action and sensor models along with the learning of control strategies. This approach is initially being taken due to difficulties in scaling up these models to high dimensions. Traditional action models attempt to anticipate the effect of a sensor following an action, for example, to predict the expected sonar readings following a translation. Learning to perform an equivalent prediction with vision could require knowledge about such things as lighting, perspective and lens properties. A proposed solution is to perform predictions only for a reduced set of directly relevant features. For a sonar sensor and our task, suitable features would be the distance and angle to the object. Techniques utilizing explanation based neural networks are being used to address this problem[Mitchell and Thrun, 1993].
- When weak action and sensor models are present, but before a complete reactive strategy has been produced, deliberative planning is appropriate. The current incarnation of the learning agent utilizes a teacher to produce approximate strategies during early stages of learning.

Beyond this future work there are a number of limitations not addressed by the architecture in its present form.

- When weak action models are present, the architecture must explicitly learn these models. The embedding of knowledge in the system *without* learning has not been explored. It would require a mechanism for arbitrating between a weak model and a "more accurate" learned model. Since such learning may be performed quite well using supervised techniques, as with the task completion recognizer, it was felt that maintaining such separate competing model representations was unnecessary.
- Extending the framework to maintaining a number of concurrent goals may be approximated either by having multiple competing learning agents or extending the task completion recognizer to control the sequence in which the goals may be satisfied. Both these extensions requiring capabilities to analyze strategies so that conflicting policies may be correctly blended.
- Collaboration with a human operator for purposes of task specification and demonstration has been poorly defined. The prototype system allows this

to occur by using "Hero" in a tele-operated fashion. Alternative more natural methods would include demonstrating the task to the agent[Ikeuchi and Suehiro, 1991], or verbally dictating the task requirements.

- Explanation of reactive strategies which the agent has learned is not a simple task. It is often crucial for plans to be accountable, that is, for a reason to be present for each control decision. Since the representation of the control strategy is internal to the reinforcement learning algorithm and in terms of task dependent features, relating these features back to the real world and the actions taken is a real issue. This is especially important when it is wished to apply the compiled reactive strategies to novel task domains.

Acknowledgements

I thank the various members of the Learning Robots Lab, both past and present, whose ground work has allowed the ideas presented here to have been developed . I'm specifically grateful to Tom Mitchell for his advice on this work, to Rich Caruana and Justin Boyan for both being part of the development of the prototype system and, along with Sven Koenig, for commenting on an earlier draft.

References

[Blythe and Mitchell, 1989] Jim Blythe and Tom M. Mitchell. On becoming reactive. In *Proceedings of the Sixth International Machine Learning Workshop*, pages 255–259. Morgan Kaufmann, June 1989.

[Chrisman *et al.*, 1991] Lonnie Chrisman, Rich Caruana, and Wayne Carriker. Intelligent agent design issues: internal agent state and incomplete perception. In *Proceedings of the AAAI Fall Symposium on Sensory Aspects of Robotic Intelligence*. AAAI Press/MIT Press, 1991.

[Christiansen, 1992] Alan D. Christiansen. *Automatic Acquisition of Task Theories for Robotic Manipulation*. PhD thesis, Carnegie Mellon University, March 1992.

[Ikeuchi and Suehiro, 1991] Katsushi Ikeuchi and Takashi Suehiro. Towards an assembly plan from observation. Technical Report CMU-CS-91-167, School of Computer Science, Carnegie Mellon University, 1991.

[Lin and Mitchell, 1992] Long-Ji Lin and Tom Mitchell. Memory approaches to reinforcement learning in non-markovian domains. Technical Report CMU-CS-92-138, School of Computer Science, Carnegie Mellon University, 1992.

[Lin *et al.*, 1989] Long-Ji Lin, Tom M. Mitchell, Andrew Philips, and Reid Simmons. A case study in autonomous robot behavior. Technical Report CMU-RI-89-1, Robotics Institute, Carnegie Mellon University, 1989.

[Lin, 1993] Long-Ji Lin. *Reinforcement Learning for Robots Using Neural Networks*. PhD thesis, Carnegie Mellon University, January 1993.

[Mitchell and Thrun, 1993] Tom M. Mitchell and Sebastian B. Thrun. Explanation-based neural network learning for robot control. In J. E. Moody, S. J. Hanson, and R. P. Lipmann, editors, *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, December 1993.

[Mitchell, 1990] Tom M. Mitchell. Becoming increasingly reactive. In *Proceedings of the Eight National Conference on Artificial Intelligence*, pages 1051–1059. AAAI Press/MIT Press, 1990.

[Tan, 1992] Ming Tan. *Cost-Sensitive Robot Learning*. PhD thesis, Carnegie Mellon University, 1992.

[Thrun, 1992] Sebastian B. Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, School of Computer Science, Carnegie Mellon University, 1992.

[Thrun, 1993] Sebastian B. Thrun. Exploration and model building in mobile robot domains. In *Proceedings of the IEEE International Conference on Neural Networks*. IEEE Press, March 1993.