Berthouze, L., Prince, C. G., Littman, M., Kozima, H., and Balkenius, C. (2007).
Proceedings of the Seventh International Conference on Epigenetic Robotics: Modeling
Cognitive Development in Robotic Systems. *Lund University Cognitive Studies, 135*.

Generalization and Specialization in Reinforcement Learning

Stefan Winberg
stefan.winberg.366@student.lu.se

Christian Balkenius
christian.balkenius@lucs.lu.se

Lund University Cognitive Science
Kungshuset, Lundagård
S-222 22 Lund, Sweden

Abstract

To learn efficiently, it is important that previous experiences can be used to select actions that have been successful in the past when new tasks are learned. We explore how generalization from previous training instances can be used to explore an action space in a more efficient way using ContextQ, which is an algorithm that learn to perform action in a particular context by generalizing maximally from previous learning instances to new situations. When actions are not useful, they are inhibited in the specific context where they do not work. This leads to an interaction between generalization and specialization during learning.

Here we describe our first systematic investigations of the algorithms on typical reinforcement learning problems. It is shown that when the context can be coded in a meaningful way, this algorithm performs better than the standard tabular Q-learning algorithm in many cases although the advantage of the context sensitive generalization depends on the particular environment.

1. Introduction

When learning complex tasks it is essential to be able to generalize from similar instances in the past. Although in theory it would be possible to use blind trial and error, this is not tractable in practice. Even for a very small state-space, the number of possible action sequences are very large and it would be impossible for a robot to test all combinations. It is clear that exploration would be more efficient if the robot could generalize from previous experience and first test behaviors that have been successful in similar situations before.

Looking at developmental psychology, evidence is accumulating that infants and children use similarity based measures to categorize objects (Abecassis et al, 2001, Sloutsky and Fisher, 2004). It is also known that the use of such similarities is context dependent (Jones and Smith, 1993) and that it can be coded at different levels of similarity (Quinn et al, 2006). We want to explore how such similarity based generalization can be exploited during learning.

Context provides a mean to greatly reduce the number of possible inferences, making it easier to separate the relevant facts about a given situation. Context can be thought of as any information that can be used to characterize the situation, such as the task, question, place or even the goal. Studies made on animals suggest that a behavior learned in one context is carried over to other contexts, but learned inhibition of a behavior will be unique to each context where the behavior was extinguished (Bouton, 1991, Hall, 2002). Most reinforcement learning algorithms learn to complete a single task in one context, but animals apply what they learn in one context to other contexts as well.

One way to incorporate this functionality in a reinforcement learning algorithm is to use a function approximator that is able to generalize not only within the context, but between different contexts as well. This role would typically be filled by some sort of artificial neural network. Unfortunately, the property that makes artificial neural networks able to generalize and degrade gracefully is also one of their main weaknesses. Since the information stored in the network share the same set of connection weights, the network often forgets catastrophically if new data is presented without proper rehearsal of the previously learned data. The new data will simply erase the old data (French, 1991, 1999). Since knowledge acquired in different contexts is likely to contradict each other, learning in a new context is likely to remove what was learned in a previous context.

Previous studies have shown that it is possible to construct a context sensitive artificial neural network that fulfill these demands (Balkenius & Winberg, 2004; Winberg, 2004). It has been used to model context sensitive categorization (Balkenius & Winberg,

2004), task-switching (Balkenius & Winberg, 2004) and developmental disorders (Björne and Balkenius, 2005). Here we want to explore how well this algorithm works on the type of mazes often used in reinforcement learning studies.

2. A Reinforcement Learning Framework

The general reinforcement learning framework illustrated in Fig. 1 was used for all the simulations and implemented in the Ikaros system (Balkenius, Morén and Johansson, 2007). This framework is derived from the standard Q-learning model (Watkins and Dayan, 1992) and is divided into three modules, Q, RL-CORE, and SELECT.

The module Q is responsible for calculating the expected value of each action in state s. It has three inputs and one output. One input-output pair is used to calculate the expected value a of each possible action in the current state s. The other two inputs are used to train the module on the mapping from a state delayed by two time steps $s_{2\Delta}$ to a target delayed by one time step T_Δ. Any of a number of algorithms can be used as module Q, ranging from tables in the basic Q-learning model to different types of function approximators and artificial neural networks. Because of the separate input for training and testing, the module Q can simultaneously work in two different time frames without the need to know about the timing of the different signals.

The module RL-CORE is the reinforcement learning specific component of the system. This module receives information about the current state in the world s, the current reinforcement R, the action selected at the previous time step a_Δ and the expected value of all actions in the current state a. This is used to calculate the training target for the module Q.

The module SELECT performs action selection based on its input from RL-CORE. It may also potentially have other inputs that determines what action is selected. This module may for example implement Bolzmann selection or ϵ-greedy selection (Sutton and Barto, 1998). It may also use different forms of heuristics to select an action. It is even possible that the action selected is entirely independent of the inputs from RL-CORE. In this case, RL-CORE will learn the actions performed by some other subsystem.

One advantage of this framework is that the different modules can be exchanged to build different forms of reinforcement learning systems. Another advantage is that all timing is taken care of by the delays on the connections between the different modules. If the world is slower at producing the reward, the only thing that needs to be changed are the different delays in the system.

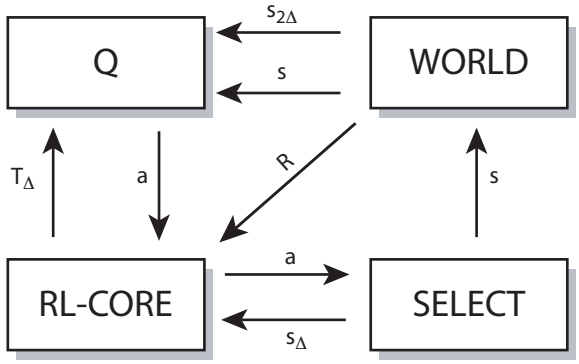

FIGURE 1: *The general reinforcement learning framework used. See text for explanation.*

2.1 Tested Algorithms

We compared the performance of two algorithms. The first is the standard tabular Q-learning algorithm (Watkins and Dayan, 1992, Sutton and Barto, 1998). The other algorithm was ContextQ which is a context sensitive version of Q-learning (Winberg, 2004, Balkenius and Winberg, 2004).

Let s_t be the current state and a_t the selected action at s_t. The result of performing action a_t in state s_t is s_{t+1}. The algorithm attempts to estimate a function $Q(s, a)$ which can be seen as the associative strength between state s and action a. The Q-function is updated as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Delta Q_t,$$

where,

$$\Delta Q_t = r_{t+1},$$

or,

$$\Delta Q_t = \left[\gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]$$

depending on whether a reward is given or not.

In the simplest implementation of the algorithm, s and a are discrete and $Q(s, a)$ is represented by a table with entries $Q(s, a) = q_{s,a}$ that are changed by the update rule. This is tabular Q-learning. The value α is the learning rate and γ is the temporal discount factor (See Sutton & Barto, 1998)

In addition to a state input, ContextQ uses a second input which codes for the current context. Initial learning when the actual reinforcement is larger than the expected reinforcement only influences the mapping from the state to the actions. However, when the actual reinforcement is smaller than expected, the association from the current state to the action is not weakened. Instead, it is inhibited in the current context. This idea was originally derived from studies of animal learning (e.g. Bouton, 1991) and has been used in a number of models of conditioning

(Balkenius and Morén, 2000, Morén, 2002) as well as in models of categorization (Balkenius and Winberg, 2004). The formulation of ContextQ described here is similar to that used before but includes a scaling factor for inhibitory learning that has not been previously described.

Let each state be represented by a state vector $s = \langle s_0, s_1, \ldots, s_n \rangle$ and let $\{a_0, a_1, \ldots, a_m\}$ be a discrete set of actions. The Q-function is estimated as,

$$Q(s, a_j) = \sum_{i=0}^{n} s_i w_{ij},$$

and the update rule is

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \alpha \frac{s_i a_j}{|s|} \Delta Q_t.$$

where $a_j = 1$ for the selected action j. That is, each weight is updated according to the error in the estimated value multiplied with the value of the state component s_i. This means that only components of the state that contributed to the selected action will be updated.

Let the context be described by a vector $c = \langle c_0, c_1, \ldots, c_p \rangle$. We can reformulate the linear estimator above in the following way by including additional weights u_{ijk} which relate each association w_{ij} to the context c_k:

$$Q(c, s, a_j) = \sum_{i=0}^{n} s_i w_{ij} I_{ij},$$

where,

$$I_{ij} = \prod_{k=0}^{p} (1 - c_k u_{ijk}).$$

In neural network terms, I_{ij} can be seen as shunting inhibition from the context of the association from the state to the action. We now need to consider how the learning rule should be changed to reflect the new context sensitive estimator.

When all $u_{ijk} = 0$, the algorithm work exactly as before which implies that the original equation can still be used for the case when $\Delta Q_t > 0$. This will result in initial learning that is totally independent of the context. On the other hand, when $\Delta Q_t < 0$, instead of changing the weights w_{ij}, we increase the inhibition from the current context according to

$$u_{ijk}^{(t+1)} = u_{ijk}^{(t)} - \beta(1 - u_{ijk}^{(t)}) \frac{s_i a_j c_k}{|s| w_{ij}} \Delta Q_t.$$

In other words, the inhibition from the current context will increase to the association between the current state and the selected action when the actual reinforcement is lower that the expected reinforcement. Once the weights w_{ij} have reached their maximal values, all learning will take place in u_{ijk}. Also, if the appropriate action within a fixed context changes, it may become necessary to decrease the values of u_{ijk}. The solution to these problems is to allow changes in both directions of both w_{ij} and u_{ijk}, but to modulate it with the sign of ΔQ_t.

The simplest scheme is to use two learning rate constants α^+ and β^+, which are used when $\Delta Q_t > 0$, and two constants α^- and β^-, which are used when $\Delta Q_t < 0$, and to update both w_{ij} and u_{ijk} at each time step.

3. Simulations

A typical reinforcement learning problem has a multidimensional state space. This makes it difficult to visualize the problem in a way that is easy to comprehend. Therefore a navigation task through a two-dimensional maze is often chosen as the basic test environment since each state can be represented by a physical location.

When the state space is visualized as a two dimensional surface, the solution can be described as a path from the start state to the goal state. Initially the agent has no knowledge of the state space. Therefore, the first time the agent enters the maze it has to search it through to find the goal. It is important to note that the two-dimensional layout of the state-space is not available to the agent. Our intuitions about the expected behavior can thus be misleading. Nevertheless, a maze is useful visualization of a state-space and we have chosen a set of mazes we call 17T4U that illustrates different strengths and weaknesses of ContextQ (Fig. 2 and Fig. 4). The mazes superficially looks like the letter in the name of the set. In addition, we tested the algorithm on a large maze with a lot of repeated structures where it would be likely that the benefits of generalization would be seen more clearly (Fig. 3).

Parameters The parameters for the two models were set as follows. For the tabular Q-learning implementation, the learning rate was set to 0.2 and the discount was 0.9. The initial weights were set to 0.1. Epsilon-greedy was used for action selection with $\epsilon = 0.05$.

For ContextQ, the learning rates were $\alpha^+ = 0.1$, $\beta^+ = 0.1$ and $\alpha^- = 0.0$, $\beta^+ = 0.7$. The discount was set to 0.9 and the initial weights were all 0.1. Boltzmann selection was used for action selection with a temperature that gradually decreased from 0.05 to 0.005.

Stimulus Coding To make ContextQ useful, it is necessary to code the input and context in a suitable way. Since we want the algorithm to generalize between similar states it is necessary that the state

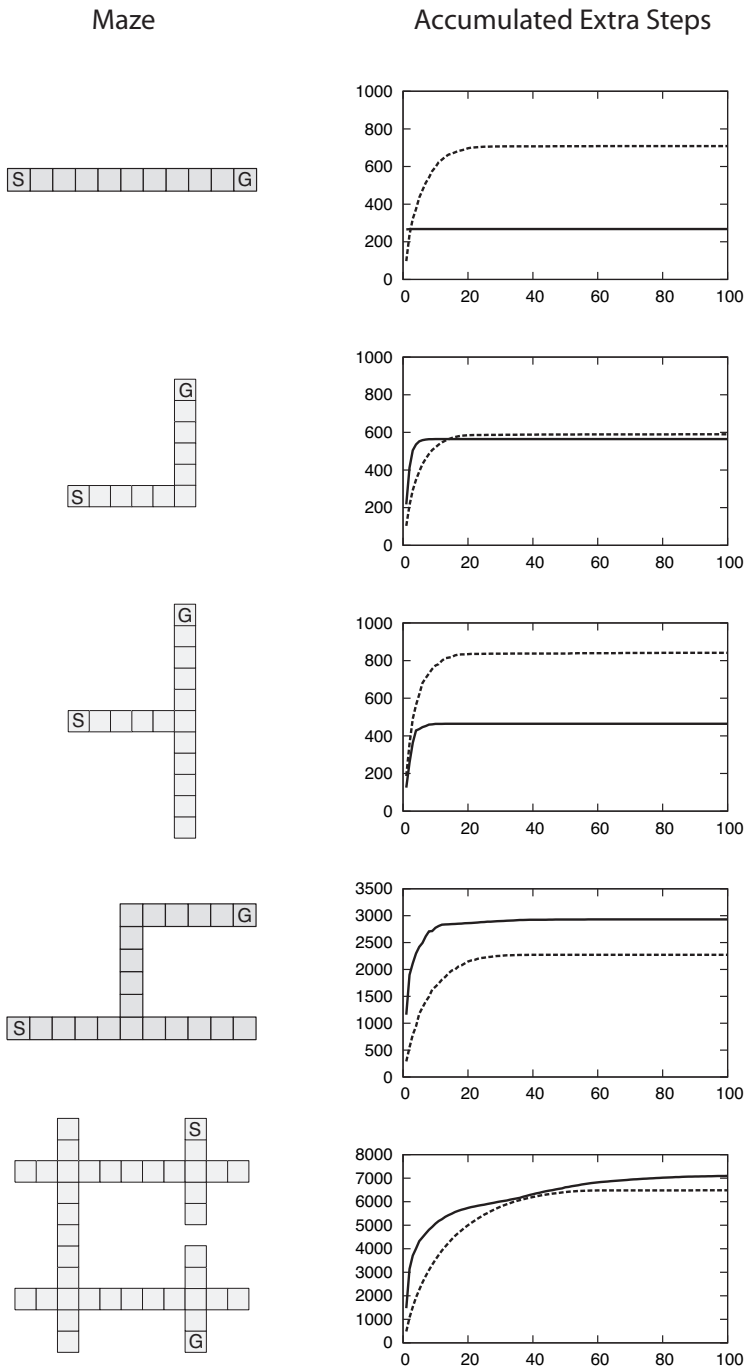

FIGURE 2: *Results for the narrow 17T4U dataset. The different mazes are shown together with the learning progression for standard Q-learning (dashed line) and ContextQ (solid line). The plots show the accumulated number of unnecessary steps used to go to the goal from the start for each consecutive trial. The slope of the curve thus indicates the number of steps needed to reach the goal and a flat line means that the optimal solution has been found since no extra steps are needed. The curves represent averages of 60 simulations. See text for further explanation.*

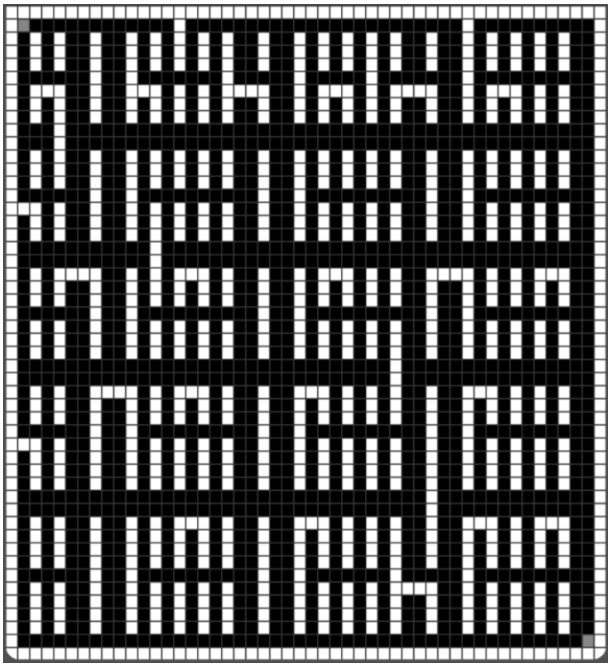

FIGURE 3: *A large maze used to illustrate the ability of ContextQ to generalize behavior in an environment with a lot of repeated structure. The start location is in the upper left corner and the goal is in the bottom right corner.*

coding reflects such a similarity. We chose to code each state in a vector of 18 elements, where each element codes for the presence or absence of free space at each of nine locations around the current position of the agent in the maze. A 1 was used to indicate free space and 0 was used to indicate a wall for the first nine elements. The following elements contained the same information inverted. The current location was always coded as a 0.

The current location in the maze was used as contextual input. This is consistent with the idea that exceptions should be learned about particular locations in the mazes.

Simulations Each maze was tested 60 times for each of the models. The average number of extra steps needed to reach the goal for each trial were recorded. For example, if the shortest path from the start to the goal is ten steps and the model needed twelve, this would constitute two extra steps. For the large maze. The average of 30 runs was used instead.

4. Results

4.1 The Narrow Mazes

The 1-Maze The 1-Maze is a straight corridor from the start to the goal (Fig. 2). This maze demonstrates clearly the power of generalization in the ContextQ algorithm. On the first trial, the random walk is used to move from the start to the goal. Once the agent has been rewarded the action of moving to the right is directly generalized to all location in the corridor. As a result, the agent will perform perfectly after a single trial. Tabular Q-learning, on the other hand, will not be able to generalize and will slowly learn to use the same action at all locations.

The 7-Maze The 7-Maze is a simple maze with one corner (Fig. 2). Here, two different actions are need. First, the agent needs to move to the left and after the corner it needs to move upwards until it reaches the goal. To Q-learning, there is little difference between this maze and Maze 1 since in both cases ten individual actions need to be learned to go from the start to the goal. To ContextQ, on the other hand, the situation is entirely different. The first trial is again random walk until the agent learns to move upwards after it has been rewarded. During the second trial, this action will be incorrectly generalized to the horizontal corridor since this is the only action that has been rewarded so far. This will lead to an extinction phase where this action will become inhibited within the horizontal arm of the maze. Once this incorrect generalization is completely inhibited, the agent will move toward the vertical arm where the generalization is still valid. It will subsequently move directly to the goal through the vertical part of the maze. At the same time, the action of moving to the right in the horizontal part of the maze will be reinforced and the agent will behave perfectly in the maze. The action of moving to the right will have been generalized to all location in the horizontal part and the action of moving upward has been generalized to all locations in the vertical part.

The T-Maze The T-Maze was included in the set since it is common maze in many studies of reinforcement learning. ContextQ is very quick to learn and only requires a few trials to learn the maze perfectly. The explanation is the same as in the 7-Maze.

The 4-Maze Since we wanted to test what would happen in a situation that appeared to be optimally bad for ContextQ, we studied the behavior of the algorithm in a 4-Maze. Each time ContextQ reaches the goal, the action of moving to the right will be reinforced. This will lead to an ever increasing tendency for the agent to not turn upward at the choice point. Instead it will continue right into the dead end. Since there is no reward at the end of the lower arm of the maze, the move right action will be gradually extinguished until the agent takes the upper path again. However, once the agent reaches the goal and gets rewarded again, it will again choose to

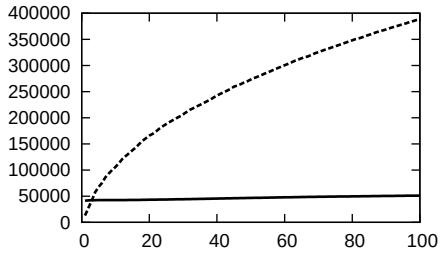

FIGURE 5: *In the large maze, the ability of ContextQ to generalize from one part of the maze to another is very evident..*

move to the right at the choice point since the reward has counteracted the previous inhibition. This is thus a very hard maze for ContextQ to solve. Despite this, ContextQ is slightly faster than Q-learning at this maze. The disadvantage of the continuously repeated incorrect generalization is smaller than the gain from the correct generalizations along the corridors.

The U-Maze The U-Maze was designed as a more realistic example with a number of dead ends where ContextQ could get stuck. Yet, ContextQ learns the maze at approximately the same time as tabular Q-learning.

4.2 The Wide Mazes

The mazes describe above are built from narrow corridors where it is obvious that there is a single good generalization. To test if the advantages of ContextQ would carry over to less obvious situations, we designed wide version of the mazes where the corridors had a width of two squares instead of one (Fig. 4). In all cases, the goals were placed as to reward the least useful action for ContextQ. ContextQ was able to learn all mazes correctly in less time steps than tabular Q-learning for all mazes but the last(Fig. 4). The wide U-maze is hard for ContextQ to learn since there is very little sensory information to use during learning. Despite this, the performance is not much worse than tabular Q-learning.

4.3 The Large Maze

The difference between ContextQ and tabular Q-learning is most clearly seen in the large maze (Fig. 5). The behavior of ContexQ converges to an optimal behavior in very few trials while tabular Q-learning shows no tendency to converge even after 400000 steps. Unlike the previous mazes, the large maze contains plenty of opportunity for successful generalization and this is what gives ContextQ a great advantage in this maze.

5. Discussion

We have run simulations of the ContextQ algorithm, which uses generalization and specialization to learn a behavior, on a number of maze problems to compare it with standard Q-learning.

The ContextQ algorithm performed better than tabular Q-learning in some simulations with narrow corridors which shows the ability to generalize can be used to great advantage. Even though many of the mazes tested were selected to be to the disadvantage of ContextQ, the algorithm is still on par with standard Q-learning. The advantages of correct generalizations overshadow the disadvantages of the incorrect ones. This gives support for the view that it is efficient to generalize maximally from previous experiences and then gradually specialize in specific contexts. For the wide mazes, ContextQ was as fast as or faster than tabular Q-learning.

It can be argued that there exist more efficient forms of Q-learning and that the comparison is not fair. For example, by using an eligibility trace, Q-learning will learn much faster. This is most likely also the case for ContextQ and here we only wanted to compare the minimal implementations of the two algorithms. In the future, we want to add further mechanisms to both algorithms to get more efficient learning systems.

Another difference between the two models is that tabular Q-learning can make use of initially positive weights to explore the environment efficiently (Koenig and Simmons, 1996). Such a strategy is not available to ContextQ as there is no weight specific to each state initially. We are currently investigating how context dependence can be included also in an actor-critic architecture, where such a mechanism is easier to implement since the generalization can take place in the actor while state-spcific values can be learned by the critic.

In particular, we want to test the effect of a hierarchical state-space coding for the spatial context (Balkenius, 1996). Presumably, this could lead to much faster specialization in the 4-maze, since the whole corridor could be treated as a single context. We also want to add eligibility traces to the algorithm to see if this is as efficient for ContextQ as for the standard case.

What are the implications for epigenetic robotics for the algorithm described above? For a robot that needs to develop autonomously, it will be necessary to learn a large number of behaviors in different situations. Such learning can be made much faster if the robot generalizes from previous instances during learning. The interplay between generalization based on the current sensory information coded in the state and the specialization based on contextual inhibition has a number of advantages.

First, the generalization is maximal. This means

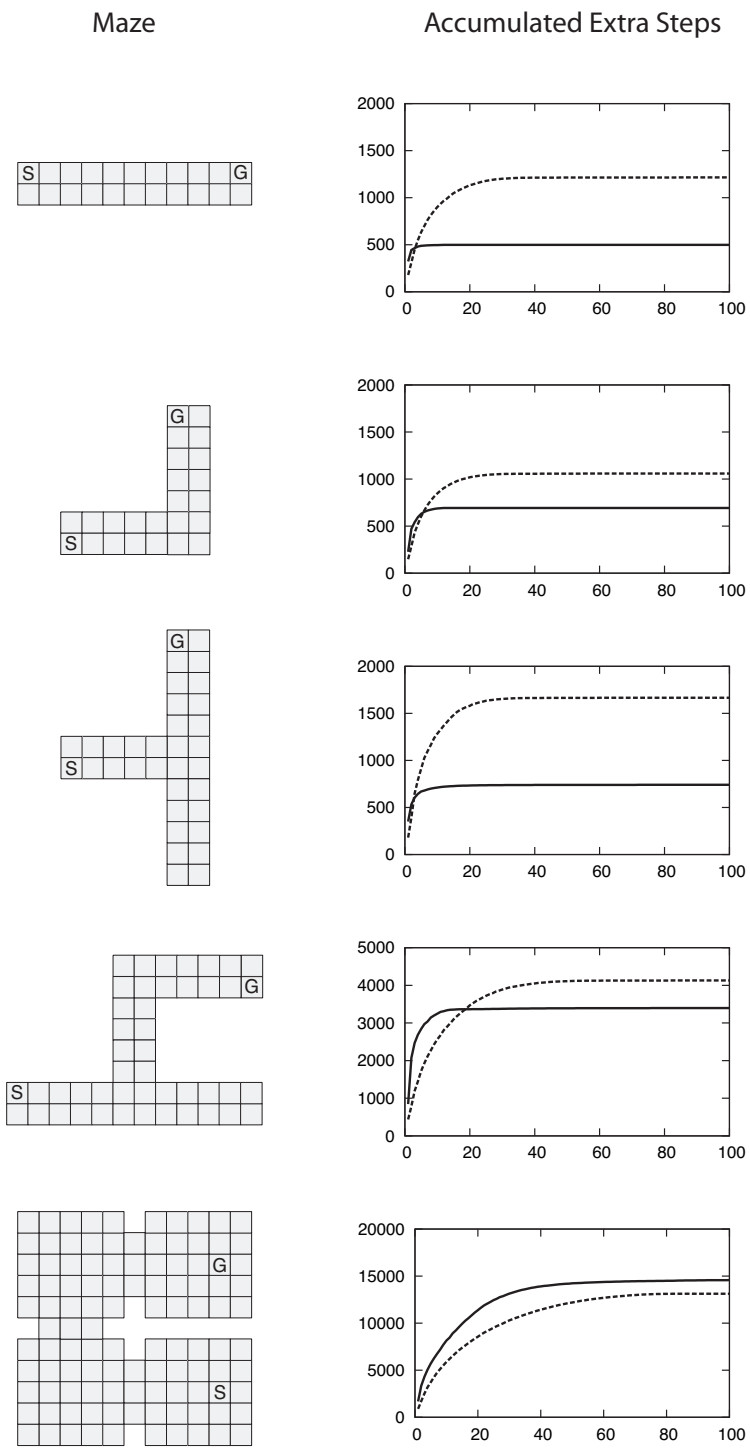

FIGURE 4: *Results for the wide 17T4U dataset. Solid lines: ContextQ. Dashed lines: Tabular Q-learning. See text for explanation.*

that any similarity between the current state and a previously learned state will be exploited when actions are tested in a new context. If there are regularities in the world where this is useful, it will lead to faster learning.

Second, the contextual inhibition makes it possible to unlearn an inappropriate behavior in a particular situation, but it does not extinguish was was already learned. When the initial learning context or another new context is reestablished, what was already learned is still there. It thus avoids catastrophic forgetting which is a large problem in many learning systems.

Third, although ContextQ is at heart a reinforcement learning algorithm, it generates efficient search paths through the state space in the case of the mazes tested here. This is what makes it possible for the algorithm to learn correct behavior on very few trials in simple mazes.

Acknowledgements

We would like to thank Alexander Kolodziej for help with running Ikaros on the Linux cluster used for the simulations. This work was supported by the EU project MindRaces, FP6-511931.

References

Jones, S. S., and Smith, L. B. (1993). The place of perception in children's concepts. Cognitive Development, 8, 113-139.

Abecassis, M., Sera, M., Yonas, A., and Schwade, J. (2001). Whats in a shape? children represent shape variability differently than adults when naming objects. *Journal of Experimental Child Psychology, 78, 3*, 213239.

Balkenius, C. (1996). Generalization in instrumental learning. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S. W. (Eds.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.

Balkenius, C., and Morén, J. (2000). A computational model of context processing. In Meyer, J-A., Berthoz, A., Floreano, D., Roitblat, H. L., Wilson, S. W. (Eds.), *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behaviour*. Cambridge, MA: MIT Press

Balkenius, C., Morén, J. and Johansson, B. (2007). *System-level cognitive modeling with Ikaros*. Lund University Cognitive Studies, 133.

Balkenius, C. and Winberg, S. (2004). Cognitive Modeling with Context Sensitive Reinforcement Learning, *Proceedings of the AILS-04 Workshop*, 10-19.

Björne, P., and Balkenius, C. (2005). A model of attentional impairments in autism: First steps toward a computational theory. *Cognitive Systems Research, 6, 3*, 193-204.

Bouton, M. E. (1991). Context and retrieval in extinction and in other examples of interference in simple associative learning. In Dachowski, L. W. and Flaherty, C. F. (Eds.), Current topics in animal learning: Brain, emotion, and cognition (pp. 2553). Hillsdale, NJ: Erlbaum.

French, R. M. (1991). Using Semi-Distributed Representations to Overcome Catastrophic Forgetting in Connectionist Networks. In *Proceedings of the 13th Annual Cognitive Science Society Conference*, 173-178.

French, R. M. (1999). Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences*, 3, 128-135.

Hall, G. (2002) Associative Structures in Pavlovian and Instrumental Conditioning". In Pashler, H. and Gallistel, R. (eds.), *Stevens Handbook of Experimental Psychology. Volume 3: Learning, Motivation, and Emotion*. John Wiley & Sons.

Koenig, S. and Simmons, R.G. (1996).The Effect of Representation and Knowledge on Goal-Directed Exploration with Reinforcement-Learning Algorithms. *Machine Learning*, 22, (1-3), 227-250.

Morén, J. (2002). *Emotion and Learning - A Computational Model of the Amygdala*, Lund University Cognitive Studies, 93.

Quinn, P.C., Westerlund, A., and Nelson, C.A. (2006). Neural markers of categorization in 6-month-old infants. *Psychological Science, 17, 1*, 5967.

Sloutsky, V.M. and Fisher, A.V. (2004). Induction and categorization in young children: A similarity-based model. *Journal of experimental psychology. General, 133, 2*, 166188.

Sutton, R., and Barto, A., (1998). Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, A Bradford Book.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. Machine Learning, Vol. 9, 279-292. Williams, R. J. and David, Z. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation, Vol. 1, 270-280.

Winberg, S. (2004). Contextual Inhibition in Reinforcement Learning, MSc Thesis in Cognitive Science. Lund University.