Path-finding in real and simulated rats: On the usefulness of forgetting and frustration for navigation learning

Minija Tamosiunaite[1,2], James Ainge[2], Tomas Kulvicius[1,3], Bernd Porr[4], Paul Dudchenko[2], and Florentin Wörgötter[2,3]

[1]Department of Informatics, Vytautas Magnus University, Vileikos 8, LT-44404 Kaunas, Lithuania, m.tamosiunaite@if.vdu.lt
[2]Department of Psychology, University of Stirling, Stirling FK9 4LA, Scotland
[3]Bernstein Center of Computational Neuroscience, University Göttingen, Germany, worgott@bccn-goettingen.de, tomas@bccn-goettingen.de
[4]Department of Electronics & Electrical Engineering, University of Glasgow, Glasgow, GT12 8LT, Scotland, B.Porr@elec.gla.ac.uk

Abstract

The hippocampal place field system has been considered as one main structure for reward based navigation learning in rodents. Reinforcement learning (RL) mechanisms have been used to model this, associating the *states* in an RL-algorithm to the place-fields in a rat. Conventional RL usually makes a few simplifying assumptions, most notably the state tiling must cover the space densely and in a non-overlapping way, states must respond deterministically, and exploration must be unbiased. Violations of these requirements may impair learning. However, none of the above requirements are valid for exploring rodents. Their place field system covers the space in an overlapping way, neurons fire probabilistically and rats explore with a directional bias favoring straight paths. We analyzed the path characteristics of rats and implemented them in an RL-system based on simulated place fields. We observed that under these conditions convergence strongly deteriorates. To mend this situation two additional bio-psychologically motivated mechanisms - forgetting and frustration - are implemented in our system. Forgetting amounts to a gradual drop of the learned weights and frustration leads to a no-learning situation if the reward is not found after some expected time. We are providing a detailed statistical analysis about the properties of mixed strategies in simulated rodent navigation learning and we show that with these two additional mechanisms convergence strongly improves again.

Keywords: Reinforcement learning, SARSA, place field system, function approximation

Introduction

The learning of goal-directed navigation in rats and other rodents has often been seen as one likely function of their hippocampal place field system. Several publications exist that use variants of reinforcement learning algorithms from the machine learning literature to

show that the place field system could indeed serve as a substrate for navigation learning (Foster et al., 2000; Arleo and Gerstner, 2000; Arleo et al., 2004; Strösslin et al., 2005; Krichmar et al., 2005). In general, reinforcement learning (RL) assumes that the space on which learning takes place is tiled into states where certain actions can be taken from every state to reach a goal (e.g. a reward location). Through exploration of the state space, an agent will try out different actions at different states and in this way it can recursively find the best possible route (the optimal policy) to a goal (for a review see Sutton and Barto (1998)). In the most general case of a reinforcement learning system, all states in the state space will have to be visited "often enough" to try out the different actions necessary for convergence. This can, however, lead to a problem because convergence is very slow if the combined state-action space is large (the "curse of dimensionality" problem). Thus, in big state spaces, value function approximation versions of reinforcement learning algorithms are used (Tesauro, 1995; Sutton and Barto, 1998). These cover the state space with large, possibly overlapping kernels and run RL over this feature space, instead of iterating over every individual state.

A second problem concerns the way RL algorithms usually choose exploration strategies. In order to learn, the agent has to explore the state-action space. Proofs exist that sufficiently dense, unbiased exploration will lead to convergence to the optimal solution in the most common RL-algorithms. To this end, conventional RL methods use random exploration, which in a navigation task leads to random walk patterns that appear incompatible with biological paths. Animals typically produce more ballistic (straight) exploration paths with only a limited degree of randomness, the length of which gradually increases from a home-base into the unknown terrain. Their paths often follow walls and landmarks, especially in daylight, e.g. see (Etienne et al., 1996; Eilam, 2004; Zadicario et al., 2005). This, however, leads to an exploration bias that jeopardizes the convergence of the RL methods.

In this study we will investigate the influence of the exploration pattern (paths) on the place-fields based navigation learning of a simulated rat. The place representations we use are abstract. Thus, our intention is not to produce a model of the hippocampus and its function. Rather, this study will focus on the interaction between biological path generation strategy and the convergence properties of learning.

Path generation is implemented as abstractions from actual rat paths, recorded and analyzed for this study. We observe that convergence of RL is not generally assured when using a realistic exploration pattern by our simulated rats. We will, however, show that our system can be stabilized by two very typical strategies common to humans and animals during learning: *forgetting* and *frustration*. Forgetting can be emulated by a gradual, step-by-step drop of the learned weights and frustration sets when the agent does not find the target in a timely manner, leading to a no-learning effect. This is compatible to the "return to home-base" drive in real rats (Whishaw et al., 2001). Accounting for forgetting and frustration in the model yields slower learning but better convergence. Without these two strategies, the system often yields divergent, irrelevant paths.

Methods

The study uses methods from reinforcement learning with function approximation to achieve fast convergence. The description of these methods is quite technical. Hence we decided to move all this into an appendix as it is not of central interest for the topics of this study. Here it may suffice to explain that in this study we are using on-policy SARSA[1] learning (Sutton and Barto, 1998). This is motivated by recent findings in the midbrain dopaminergic system (Morris et al., 2006). Alternatives would be Q-learning or Actor-Critic Learning and we will in detail discuss the choice of SARSA-learning in the Discussion section.

Model Environment

Our model animal performs a simple navigation towards a goal task in a homogeneous environment similar to a Morris water maze task (Morris, 1984). In Fig. 1a a schematic picture of the model environment is provided.

The modeling scenario includes the following issues:

- The model environment is discretized using a grid 10000×10000 units;
- The model animal at each learning trial is placed at a predefined position (5000,1000);
- A reward of the size 1000×1000 is placed opposite to the start 1000 units away from the upper border of the environment.
- The model animal travels in predefined steps (400 units $\pm$ a random component of up to 100 units).
- After the model animal reaches the reward or does not reach the reward in a predefined number of steps the animal is reset to the start position for the next trial.

The substrate for learning in our system are simulated place fields distributed within an arena. We assume that a place cell i produces spikes with a Gaussian-shaped scaled probability distribution:

$$p(\delta_i) = A \exp(\delta_i^2/\sigma^2) \tag{1}$$

where δ_i is the distance from the i-th place field center to the sample point on the trajectory, σ defines the width of the place field, and A is a scaling factor. In the areas where the values of this scaled distribution are above 1, cells fire with a probability of 1.

Place cell centers are distributed in the model environment randomly, with a uniform distribution. Experiments are performed with 20-2000 cells. Field width is defined by $\sigma = 200, 400, 600$ or 800. A scaling factor of $A = 2.5$ (Eq. 1) has been applied to the probability distributions of place cell firing, to make cell spiking more regular inside a place field.

[1]SARSA stands for "state-action-reward-state-action" referring to the transitions an agent goes through when learning.

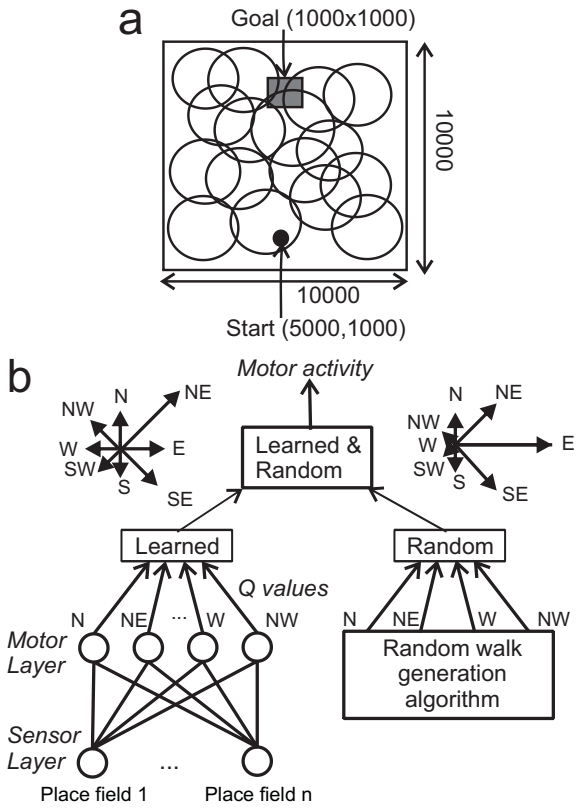

Figure 1: Model environment: the start is shown as a circle, and the goal is a rectangular area of the size 1000×1000*, located opposite to the start 1000 units away from the upper border. Big circles schematically show place fields covering the arena. b) Neural network of a model animal where motor activity is obtained as combination of learned direction values (Q-values, "Learned") and path straightening components ("Random"). The "stars" show examples how these components could look like. The two components are combined summing learned and random components with appropriate coefficients direction-wise ("Learned&Random"), and then choosing the prevailing direction for a motor action.*

General Scheme for Navigation

We are investigating learning in a network composed of two layers of cells (see Fig. 1b. At the lower layer of the network are the place cells. In the upper layer are motor cells, which learn to perform the navigation task. To keep the setup simple, we do not model head direction cells that are often also included in hippocampus-like navigation models (Brown and Sharp, 1995; Arleo and Gerstner, 2000), but allow the motor cells to direct the model animal movement towards eight directions: North, North-East, East, South-East, South, South-West, West and North-West. The actual direction is obtained combining motor cell outputs and path straightening components, used to obtain a realistic path forming strategy (see below and also Fig. 1b).

Path Generation and Exploration Strategies

Two path forming strategies are employed.

E-strategy is a usual RL strategy, with exploration and exploitation, where the path is chosen according to the learned Q-values most times, (probability $1 - p_e$), and a random move is made with probability p_e, where $0.1 \leq p_e \leq 0.2$. For random moves all directions are given equal probability. If not stated otherwise, we have set $p_e = 0.2$. If the model animal was not attaining the goal in 300 steps, it was reset to the start position for a new trial. This also applies for other strategies to be introduced below.

S-strategy performs straightening of the paths, where probabilities $p_1, p_2, ..., p_8$, $\sum_i(p_i) = 1$, are used, depending on the direction of the previous step. We define p_1 as the probability to proceed along the same direction as before, p_2 and p_8 then correspond to 45° to the left and right of this direction, p_3, p_7 reflect 90°, etc. In most of our studies we exclude backwards movement, setting $p_5 = 0$, to prevent an animal from performing small forward-backward cycles. When Q values are present, a weighted mixture of Q-based drives and randomized path-straightening drives are used:

$$\begin{array}{lcl} d_1 & = & wq_1^N + (1-w)p_1 \\ d_2 & = & .... \\ ... & & \\ d_8 & = & wq_8^N + (1-w)p_8 \end{array} \tag{2}$$

where $d_1, ..., d_8$ are the final drives, $q_1^N - q_8^N$, the normalized Q-values of the eight possible directions. Normalization is used to get $\sum_i(q_i^N) = 1$, and to bring Q-values in correspondence with the probabilities of the randomized drives. As default, we have used $p_1 = 0.5$, $p_2 = 0.156$, $p_3 = 0.063$, $p_4 = 0.031$, $p_5 = 0$, $p_6 = 0.031$, $p_7 = 0.063$, $p_8 = 0.156$, $w = 0.5$.

We also investigated a mixture of strategies E and S, where Q-values with straightening *and* some random exploration $0.1 \leq p_e \leq 0.2$ were used.

Forgetting and Frustration

Forgetting (*F-strategy*) is implemented as slow exponential decay of weights:

$$\theta(t+1) = c_f\theta(t) \tag{3}$$

where c_f is in the interval 1.0-0.99, and 1.0 represents no forgetting. When in use, this rule is applied to all weights from sensor to motor layer in each step of a model animal. If not stated otherwise we use $c_f = 0.9995$ for the experiments. If weights fall below a threshold t_f due to decay, they are set to zero. We used $t_f = 0.000001$.

Frustration (*L-strategy*) is implemented as a return to the start position if the reward is not found within an expected number of steps. Initially we limit learning to 200 steps. If the reward is found after $k < 200$ steps, we reset the limit k_l for the expected number of steps for the next trial, otherwise we increase the limit k_l:

$$k_l = \begin{cases} k + \sqrt{k} & \text{if first time exceeded} \\ k_l + jc_l & \text{if j} - \text{th time exceeded.} \end{cases} \tag{4}$$

Hence, for every following trial where the reward is not found, k_l is increased by some constant c_l. During unsuccessful trials nothing is learned. When the reward is again found, we reset k_l using Eq. 4. In this way, a process for expected time to reward is obtained where failure leads to a no-learning situation. We use the abbreviation L for trials with frustration (F is already used for forgetting), where L stands for *limit*.

Forgetting and frustration are two quite realistic bio-psychological mechanisms that can be implemented in the above described way in our model. As will be shown below, these two mechanisms yield a notable improvement in the convergence properties of the system.

Parameters of the model system are provided in a condensed way in Table 1.

Experimental Methods

A total of 5 male Lister hooded rats weighing between 300-400 g were used. In this and the subsequent experiments, compliance was ensured with national (Animals [Scientific Procedures] Act, 1986) and international (European Communities Council Directive of 24 November 1986 [86/609/EEC]) legislation governing the maintenance of laboratory animals and their use in scientific experiments. The rats were equipped with chronic recording electrodes, as described by Ainge et al. (2007), although the primary interest of this experiment was the behavior of the rats in finding a goal location. Rat runs were recorded in a square shaped arena of size 1.5x1.5x0.4 (length, width, height in meters) with blue walls. Each wall was equipped with small (10 x 10 cm) black felt "curtains", spaced equally from one another along the base of the wall. A small piece of food (a chocolate cereal loop) was presented to the rat by the experimenter whenever the rat approached a pre-specified curtain. Rats were initially unfamiliar with this arena, and 10-40 trials were run with the rat being rewarded with food whenever it approached the "correct" curtain. The total number of trials depended on the rat's motivation and learning performance. Our measure of performance was the directness of the rats' paths to the correct location. After obtaining a reward, the rat was put back into a smaller opaque container (50x50cm) for a short inter-trial interval. The position of the rat was monitored during the recording session through a black and white camera mounted on the ceiling above the arena. Two groups of ultra-bright LEDs were attached to the end of the recording cable, which in turn was connected to the chronic electrode. The LEDs were tracked using a recording system (Axona Ltd., St. Albans, UK), which detected the position of the two lights, thus providing information regarding the rat's location and the direction that the rat was facing at a sampling rate of 50 Hz. Data from the LED coordinates were stored on the hard drive of a PC.

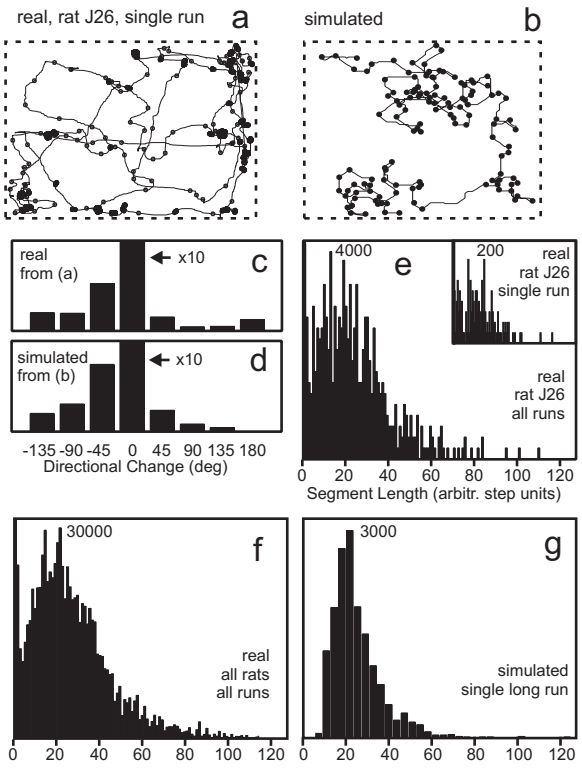

Figure 2: Statistical analysis of real and simulated rat runs. a) Example of a real rat run in a rectangular arena. Walls of the arena are shown by the dashed box. Dots mark break points between straight stretches. b) Simulated run. c, d) Distribution of turning angles for a real (c) and a simulated (d) run. e-g) Distribution of straight stretches for real (e, f) and simulated (g) runs scaled to their individual peak height as given in the diagrams. Panel (e) shows the distribution of straight stretches for a single rat (all runs), inset in (e) shows a single rat run as given in panel (a). Panel (f) contains all five experimental sessions with real rats and (g) simulated run. Units of segment length are arbitrary and scaled relative to the size of the arena. The simulated path was generated using 'SE' strategy (see subsection 'Path Generation and Exploration Strategies').

Results

Analysis of real rat paths

Fig. 2 shows examples of real (a) and simulated (b) rat runs and their statistical properties analyzing how long rats run straight (Segment Length, e-g) and how often they turn (Directional Change, c,d). Analysis has been scaled by us to unit step size relative to the size of the real as well as simulated arena. Hence step lengths can be compared. The simulated runs in (b) and (g) have been generated without learning. Dots (break-

points) in (a) and (b) indicate division into straight segments. Straight segments have been determined by standard linear regression moving along the path on a sliding window. This window was extended along the signal until a threshold for the average residual of $r \approx 1.25\%$ was reached; corresponding to 2 *cm* in (a) or 120 place units in (b). For real rats we were in addition resetting the analysis window with every stop of a rat. Choice of the threshold will clearly influence the placing of the break-points, and local analysis with a sliding window will not give optimal division of a path into longest possible straight segments, but we are not concerned with an exhaustive analysis; we only have to make sure that the generative algorithm used to create paths for the learning in the following sections will produce statistical properties which are 'within reason' similar enough to the real rat runs. Hence using the same threshold relative to the environment size for real- and simulated runs will allow for this comparison.

Already the example in (a) shows that rats have the tendency to continue on their path at least for some time without much turning. In addition, they tend to keep along the walls exploring from there on inwards. Smooth curves are rare; instead the animals turn rather sharply. Accumulations of dots occur at locations where the rat had stopped (sniffing, resting). This creates almost all contributions for the leftmost bins in the distributions (e, f), where we have clipped the bin for segment length 1.0 because a huge number of such mini-segments occur when a rat stops and just moves its head. Thus, in simulations, very small segments occur very rarely (g). In general, distributions for single runs (inset in e), all runs of one rat (e), and all runs of all rats (f) are similar getting smoother with more data. The distribution of one simulated long run (g) is similar to (f). Averaging over more simulated runs will lead to a smoother distribution (not shown), but does not otherwise alter its shape as the same generative algorithm is always used.

Distributions (c) and (d) represent the number of turns and their degree for a given path. We move along the path (real and artificial) in predefined steps, such that the step length takes the same proportion of the arena both in real and artificial example, and evaluate the angle between each two successive steps. We then bin angles into 8 categories: zero degrees turn, ± 45 degrees turn, ± 90 degrees turn, ± 135 degrees turn, and 180 degrees turn. Note, the zero-bin is 10 times larger than shown in the histograms. Both distributions are similar, and somewhat skewed to the left as the actual rat runs used were dominated by a leftward running tendency (see a,b). The 180 deg bin is empty in the simulated runs, because we did not allow the rat to directly run back. For the real rat some entries in this bin are probably due to switchbacks that occurred while stopping.

Artificially generated path shapes with and without learning

The previous section gives confidence that the path generation algorithm produces realistic paths, at least before learning. Fig. 3 displays several more artificially generated paths to the reward. Examples are shown from early (panels a-d) as well as late (panels e-h) learning. If paths are generated in the 'traditional' way by mixing exploration and exploitation (strategy E), as necessary to assure convergence of Q- or SARSA-learning, (panels a and b) they are 'wiggly' and do not resemble those of real animals. Statistical

analysis of these paths has not been performed, because their difference to real rat paths is obvious. Similar to Fig. 2, the straightened mode of exploration (strategy S) provides more realistic paths (Fig. 3 c,d) when learning starts.

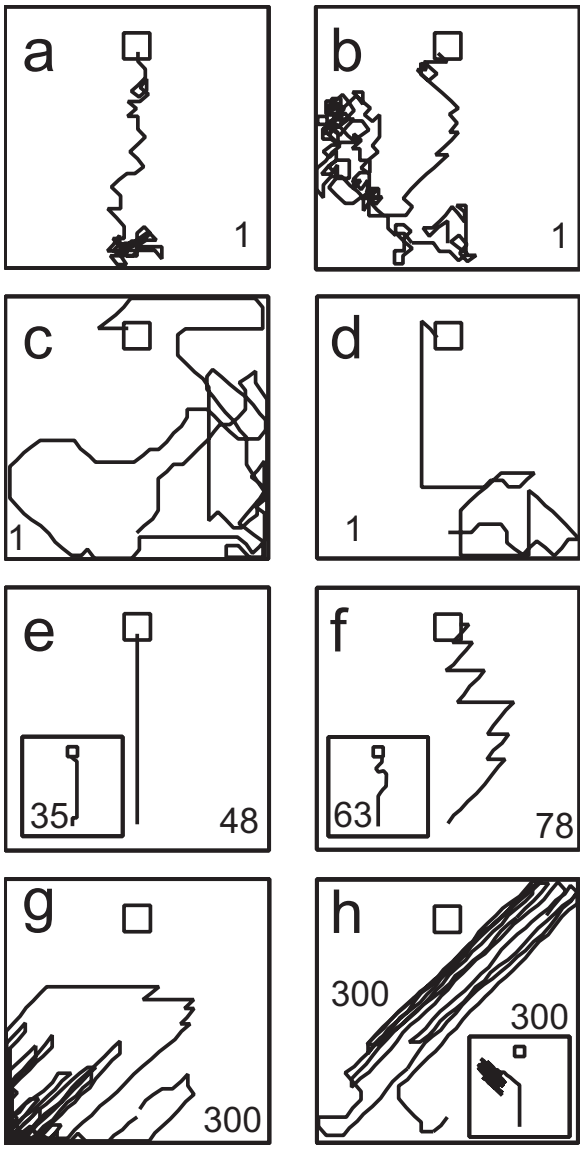

Figure 3: Examples of paths obtained with SARSA learning under different exploration-exploitation strategies: a, b - traditional exploration-exploitation, initial learning stages, c, d - straightened exploration-exploitation, initial learning stages, e - learned path in traditional exploration-exploitation case, f- zigzagging learned path in straightened exploration-exploitation case, g, h - divergent paths for straightened exploration-exploitation case, inset in h - divergence pattern when direction "back" is not forbidden. Default parameters (Table 1) were used. Small numbers at the bottom refer to the trial number from which the examples were taken.

Fig. 3e shows two optimal paths after learning. Using path-straightening strategy (S) only (or in conjunction with forgetting and frustration), straight paths were often learned. If an exploration component was added (mixing strategies E and S) a few off-path moves

occur (inset in panel e). The learned components, however, assure that in these cases the rat gets back on track immediately. When using the strategy S, we find that SARSA learning frequently produces zigzagging paths at the end of learning[2] (Fig. 3f). Those paths are normally not fully constant but zigzags will change often to a small degree trial by trial due to the random component in place cell firing which, as a consequence, also leads to some oscillations of the Q-values. When adding random exploration, mixing strategies S and E, zigzagging gets much reduced and - if convergent - paths are essentially stable in the end (inset in panel f). Panels (g) and (h) show divergent paths, which frequently develop when using strategy S. The inset in Fig. 3 h shows a divergent pattern when the direction "back" was not forbidden, which for divergence cases often leads to very fast switchbacks.

When looking qualitatively at many runs at this stage, the following early summary observations could be made (and will be supported by statistics later): with random exploration E paths are unrealistic early during learning, but often optimally convergent[3]. Path straightening S, on the other hand, leads to realistic paths when learning starts, but many times strange, zigzagging or even divergent paths develop. In the next section we substantiate these statements and try to provide a solution to these conflicting demands.

[2]Note: When using Q-learning, such zigzagging does only occur in very rare cases, and curved paths occur instead. However, the off-policy strategy performed during Q-learning does not seem to be compatible with reinforcement learning in animals and Q-learning will therefore not be discussed any further in the context of this study.

[3]Note: For a plain Q- or SARSA-learner, hence without function approximation, paths would *always* be converging to optimal ones in these cases.

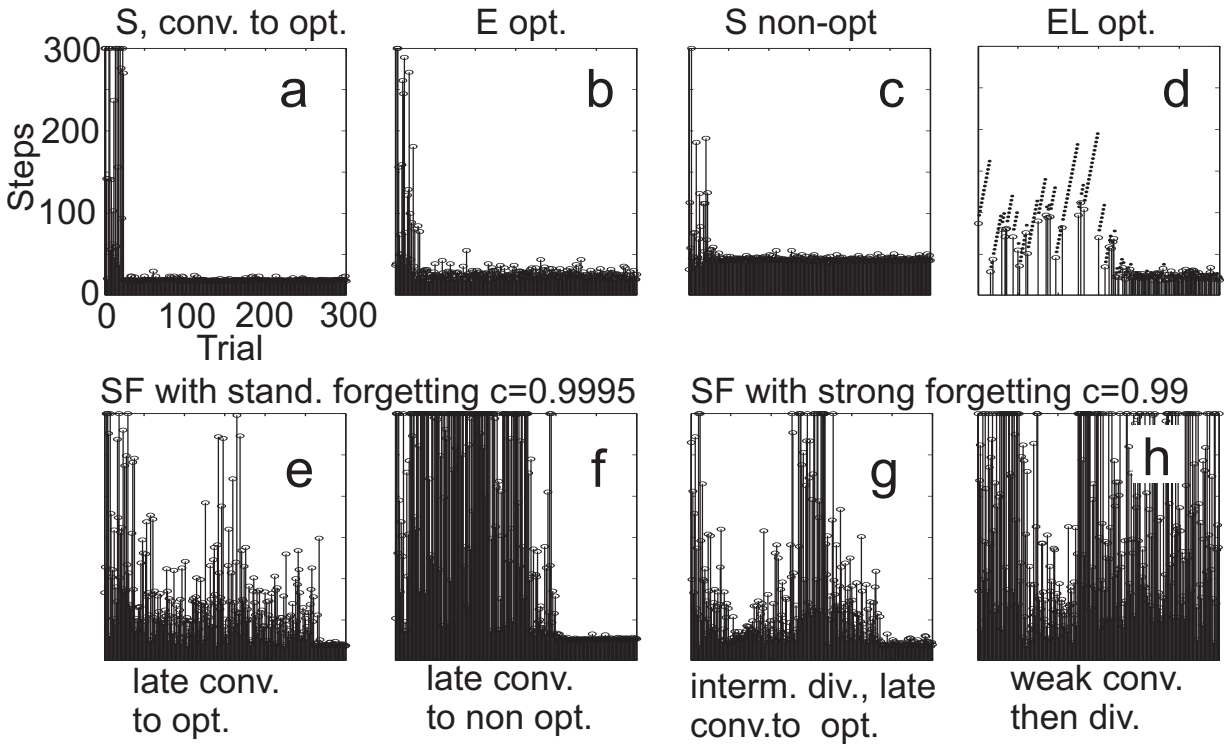

Figure 4: Patterns of convergence for different path generation and learning strategies as given above each panel (S, E, EL, SF). "opt" denotes that an optimal (straight) path had been learned, "non-opt" that the finally learned path was non-optimal (e.g. zigzagging).The small dots in panel (d) indicate trials where the rat had not found the reward within its limited learning horizon (strategy EL). Abbreviations: conv.=convergence, div.=divergence, interm.=intermediate. Default parameters (Table 1) were used.

Convergence Patterns

First we consider the convergence patterns for different combinations of path generation strategies (Fig. 4, number of steps to goal is shown). In the top row convergent cases for different path strategies are shown. In the bottom row cases where convergence was less clear are displayed. In panel (a) quick convergence to an optimal path is shown for a case of path straightening S. Note, the randomness in the spiking of the place fields leads to path lengths which vary minimally. In (b) quick convergence using traditional exploration-exploitation E is presented. One can see that here the path length fluctuates more due to occasional off-path steps (compare with inset in panel f of Fig. 3). In panel (c) a case is shown where the path straightening strategy S converges to a non-optimal (zigzagging) path-type and in panel (d) we give an example for convergence with limited path length L combined with exploration E. The dots indicate trials where the reward has not been found, which leads to a gradually growing path length limit. At the end of these learning

trials convergence to an optimal path is reached.

In cases (e-h) we show examples of convergence patterns for cases where we used straightened paths S with forgetting F. A longer time to convergence is a typical feature here and cases exist (e, g) where the behavior intermittently diverges. If forgetting is too strong, convergence is bad and even cases that begin to converge will at the end not find a good path (h).

Note, time to convergence (hence, the number of trials to find an optimal path to the reward) cannot easily be compared to real animals, because this depends again on the relative size of reward and arena. When being trained to find food, real rats found good paths in about 10 to 30 trials, which roughly compares to the simulation results shown here. Many times, however, after having learned, real animals would "get distracted" and kept on exploring, eventually going towards the reward much later. Furthermore, real animals are also using other cues (like odor) for navigation (Save et al., 2000), which have not been modeled. We have also observed that real animals sometimes show very strong inter-individual differences probably driven by more general states of, e.g., motivation, intention, fright, etc. Trying to model of this would go beyond the scope of this study.

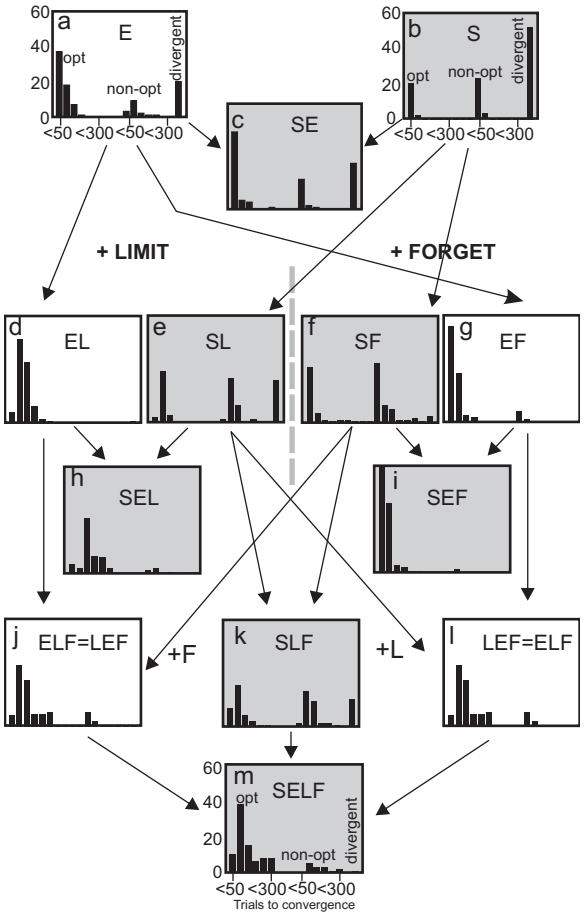

Figure 5: Statistical analysis of mixed strategies for path generation and learning as given inside each panel for 100 experiments each. Arrows indicate how properties get inherited from each other. Gray panels are the ones containing an S component. Three histograms are shown in each panel: convergence to an optimal path (left histogram), to a non-optimal path (middle histogram) and divergence cases (right single-bin). Note, panels (j) and (l) are identical. Default parameters (Table 1) were used.

Statistical analysis of mixed strategies

In Fig. 5 a summary of the influence of the different mechanism is presented. For each diagram 100 experiments were analyzed. Arrows indicate how histograms inherit properties from top to bottom. The two top diagrams (a) and (b) show the basic cases of pure exploration E and pure path straightening S respectively. Below, cases with frustration L are shown (mostly) in the left and middle part of the diagram, cases with forgetting F are found right and middle. Towards the bottom more and more strategies are mixed and the left-right separation in the diagram vanishes.

Histograms in the figure show three groups of bins. In the leftmost group the optimal path has been found, the middle group shows cases were a sub-optimal path was found (e.g. zigzag) and the rightmost group, which consists of a single bin, shows the number of divergent cases. Bins in the groups are ordered to show cases of convergence in less than 50 steps (leftmost bin) up to less than 300 steps (rightmost bin).

Gray shading indicates those cases where the paths are realistic, because they contain an S component. Cases without shading produce unrealistic paths.

As expected, pure exploration (a) often leads to convergence, but a few cases are observed where the system diverges. While biologically more realistic, the type of place field-like function approximation used here does not belong to the few known classes of function approximation algorithms for which convergence has been proved. Path straightening (b) makes the situation better and worse at the same time: The convergent cases converge faster, but there are many more divergent runs now. Mixing cases S and E (c) produces a result "in between" the pure S and E cases.

Limiting the learning horizon (frustration, L, panel d) or adding forgetting (F, panel g) efficiently eliminates divergence from the pure exploration case (compare to a). As expected learning is now a fraction slower though. If not concerned with realistic paths, strategies EL and EF would be the best choice for fast and robust convergence. Doing the same with straight paths (panels e, f) also leads to substantial improvement as compared to (b).

Panels (j) and (l) represent the case (note, j and l are identical!) where "the other" (F or L) component has been added to cases EL and EF, respectively. Now learning becomes again slower because L and F, *both* limit the memory of the learning system.

Mixing path straightening with a bit of exploration in general seems to be a good strategy (panels c, h, i, k, m), by which convergence is most of the time assured together with realistic looking paths. Case SEF (panel i) leads to fast *and* robust convergence. Many cases were found where convergence happened within 10 to 30 trials, similar to real rats. For case SEL (panel h) convergence was slower and this also holds true when mixing both limiting strategies L *and* F in cases SLF (k) and SELF (m). In our simulation, convergence times now became unrealistically long.

In summary, straight paths together with forgetting and/or frustration will not lead to good performance (panels: e,f,k; representing cases SL, SF, SLF). Adding exploration (SE, c) will immediately improve on this, while still leaving the path shape realistic, but many divergent cases remain. This can be mended by adding forgetting (SEF, i), which represents the most realistic case concerning path shapes *and* convergence times. Frustration is also a powerful mechanism to eliminate divergence, but in these simulations convergence times became now rather long (h). Mixing too many strategies will also lead to performance deterioration, because they all work in the same way, reducing the memory of the system.

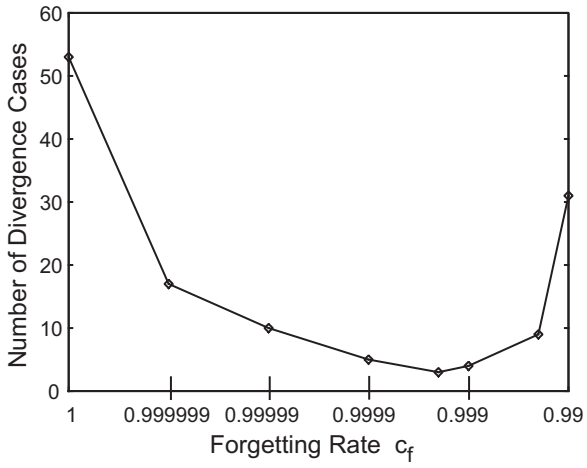

Figure 6: Number of diverging cases for different degrees of forgetting using strategy SF.

Furthermore, we investigated how sensitive the system reacts to parameters L and F, because, clearly, too much forgetting or frustration is also harmful. In Fig. 6 we show how the number of divergent cases depends on the rate of forgetting for the SF case. It shows a rather wide flat minimum, and any forgetting rate between approximately 0.999 and 0.99999 may be advantageously used. We also used frustration regimes for SL with constants $c = 2$, $c = 3$, $c = 5$, and $= 7$, and found that with $c = 2$ the process converges slower, whereas with $c = 7$ more divergent cases remain, similar to the pure S case. Cases between $c = 3$ and $c = 5$ performed similarly, successfully diminishing the number of divergent cases. Exploration has similar effects for a wide range of values, from $p_e = 0.05$ to around 0.5. With increasing exploration learning is mostly successful, but the paths get more disordered because of frequent, off-path exploratory steps.

Place Fields Size and Density

Finally we will address the question of how the obtained results depend on place cell radius and density. Hippocampus place fields are rather wide, with single cells spiking in a substantial region of the arena (Wilson and McNaughton, 1993). Little is known about place field density, because from standard recordings in a single animal, density is not straightforward to evaluate. Some authors find over-representations of places that are more important for an animal or are more densely explored (Hollup et al., 2001).

To investigate the influence of place field size and density we performed an exhaustive analysis over size/density pairings adjusted to lead to a similar coverage for the whole arena. Coverage values of 0.8, 2.4 and 4.5 were investigated. Depending on their size, different numbers of cells were required for this, as given in the central part of Table 2.

Note, as coverage is not uniform, a certain part of the surface will always on average remain uncovered. For an average coverage of 4.5 we have about 1%, for 2.4 about 6% and for 0.8 about 45% of the surface area of the arena uncovered.

Convergence of paths was evaluated through path-length histograms for 100 trials each (Fig. 7). A mixed, generic strategy for path generation and learning (SELF) has been employed with default parameters. The shortest path found in these experiments contained 16 steps, and path length was limited to 300 steps maximally; a new trial was started if this number was exceeded (see x-axis labeling in panel d). Hence, in a given bin, we plot how many times the rat had found the reward within the number of steps with which that bin is labeled from 100 experiments performed. Clearly, early in learning, paths are longer and later they are, if convergent, shorter. To show this, histograms are color coded. Starting phase of learning is shown in blue (leftmost color in each bin), and ending phase in red (rightmost color in each bin). Middle stages are shown by the other colors. For each field size/density pair a separate histogram is provided in the figure. Histograms are given in the same order as in Table 2; they are normalized to 100.

Each trial was started with a path length of 200 steps, and correspondingly there is a large thin blue column in all histograms for the 200-bin showing that early during learning all 200 steps usually are needed to search for a reward. We allowed the path to grow beyond the starting length (due to the L-component), up to 300 steps in badly convergent cases. Hence, there are some instances of paths longer than 200 in the histograms, and those allow well convergent cases to be distinguished from badly convergent cases.

At first we note that the peak of all blue contributions (early during learning) is, as expected, in all cases shifted to the right with respect to the red contributions (late during learning). Trivially, learning makes the paths shorter.

Looking in more detail at the red contributions (after learning), one can see that the shortest paths were found on average for panel (b), where the red peak falls in the 10-30 bin. Panel (b) corresponds to $\sigma = 400$, $n = 500$ and this combination has, therefore, been used for all experiments reported above. With smaller place fields $\sigma = 200$ (column one, a,e,i) convergence is much slower. The red peak is here found in the 30-50 or 50-70 bin and the finally learned paths, even with high coverage, are frequently twice as long as those in panel (b). Bigger place fields (right two columns) produce poorer convergence, introducing many red contributions into the higher order bins, pointing to long, final path lengths. Small coverage (last row), independent of the field width, produced poor convergence.

Discussion

In this study we analyzed how path formation strategies interact with reinforcement learning in a place-field like system for action value function approximation. We have shown that our model-based paths corresponded well to actual rat paths and we have tried to provide evidence that biologically plausible mechanisms (forgetting, frustration) improve the learning in such a system.

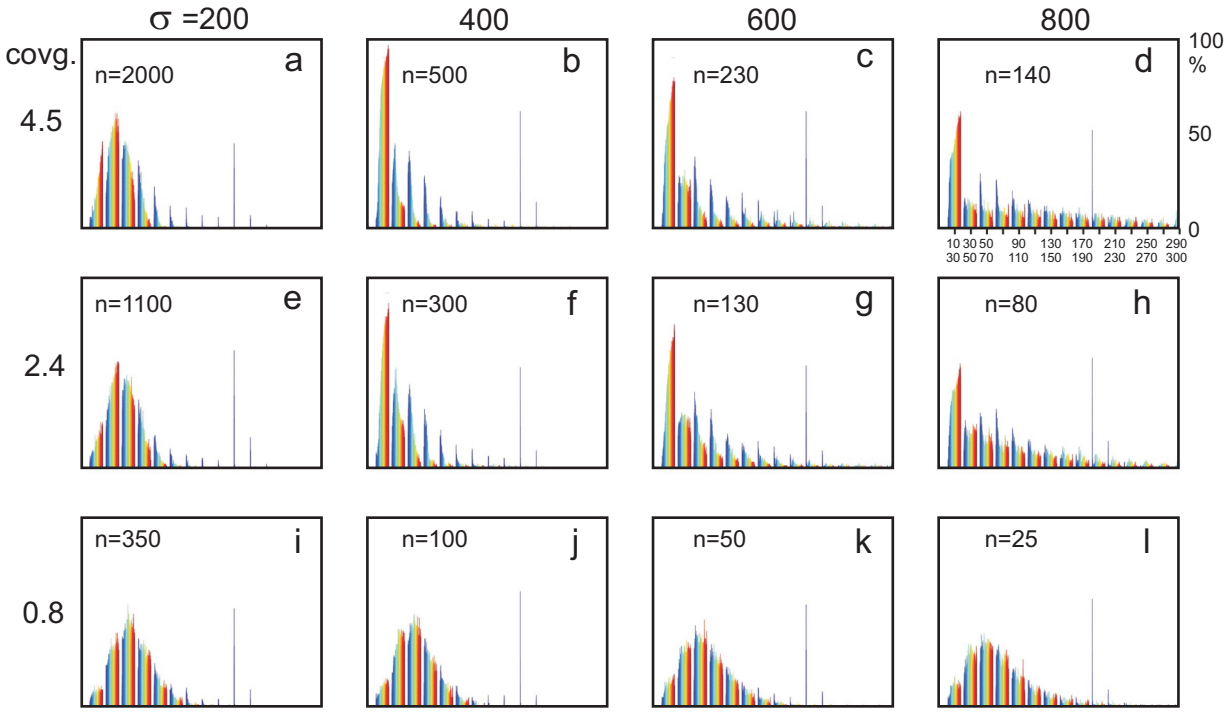

Figure 7: Path length distributions for different combinations of place field size σ and number of cells n, leading to different degrees of coverage as shown on the left side. Colors encode the stage of learning (blue=beginning of learning, red=end). For further explanation see text.

Relevance and Influence of the Chosen Learning Algorithm

In this study we have chosen SARSA learning for our experiments. This choice is motivated by several reasons. In the first place, SARSA is on-policy learning. Hence an agent updates (synaptic) weights by the outcome of the actually performed action, different from the (more commonly used) Q-learning algorithm, where weights are being updated by the best *possible* action outcome, even if the agent has actually chosen a different *performed* action. Q-learning would thus require some kind or reasoning along the lines: "I know the best possible outcome and update my learning with this even though I am trying out something else (exploration)." It has been shown that such human-like "reasoning" does not seem to be represented in the midbrain dopaminergic system, which is the structure mostly held responsible for the implementation of reward based learning in the brain (Schultz, 2002, 2007). At the level of monkeys it seems that SARSA learning prevails (Morris et al., 2006) as discussed by Niv et al. (2006). We would, thus, believe that it is unlikely that the complexity of a Q-learning off-policy update would be represented in rats. In spite of this we have also performed quite an exhaustive analysis of Q-learning (not shown) and reach the same basic conclusions. Paths will differ though. For example zigzagging in badly convergent SARSA is much less pronounced and will now be replaced by long curved paths

in badly convergent Q-learning cases[4]. Another alternative would be to employ Actor-Critic modules (Barto et al., 1983; Barto, 1995; Sutton and Barto, 1998), for example using a temporal-difference (TD, Sutton (1988)) based critic to decided about possible actions. Such a choice would be justifiable by the suggested relation of Actor-Critic architectures to the interfacing between basal ganglia and the motor system (Barto, 1995; Houk et al., 1995). It is, however, known that convergence of Actor-Critics is often quite difficult to achieve and there a many ways to construct such an architecture (Sutton and Barto, 1998), which leaves the choice too unconstrained in conjunction with this investigation. Furthermore, little is known about the behavior of Actor-Critics together with function approximation. As the possible relations of this different RL-algorithms to brain function is still a matter of debate (Wörgötter and Porr, 2005, 2007), SARSA seems to be a justifiable choice.

Relation to Biological Behavioral Mechanisms

Path straightening (S) makes not only the paths more realistic, but also makes finding of the reward easier during the initial stages of an experiment. However, we have found final convergence to be more problematic with path straightening. The number of divergent cases with path straightening increased two-fold compared to traditional exploration-exploitation strategies.

Paths with long straight stretches have been observed for rodents in open field experiments (Etienne et al., 1996; Eilam, 2004; Zadicario et al., 2005). There are differences between daylight and night behavior, where in daylight animals tend to run along walls or hide in the corner, while at night some more exploration in the center of the field happens (Eilam, 2004; Zadicario et al., 2005). Our path generation algorithms did not include all the complexity (loops, stopping) observed in actual rodent paths, but nevertheless our path geometry, and path statistics resemble real rat paths.

We have shown that convergence can be improved if forgetting (F), frustration (L), or a little bit of exploration (E) are added to the path straightening setup. In real rats the mixing of different strategies is a common observation. Different forms of forgetting are common in animals and humans. The "frustration" mechanism used here can be linked to the return-to-home-base drive in real rats. It is known that rats return to their home location in an open arena exploration process and investigate the environment in loops of increasing length (Zadicario et al., 2005). Furthermore there are behavioral differences during daylight as compared to the night, where homing is less prevalent because animals are less frightened (Eilam, 2004; Zadicario et al., 2005).

Our model makes relatively few assumptions about the determinants of behavior. We have not accounted for motivational state, thigmotaxic tendencies, arousal, and fatigue, to name but a few factors that likely influence path learning. To capture the full complexity of real animals, additional mechanisms would have to be considered. However, a strength of

[4]Note, to be more specific, we have used SARSA($\lambda = 0$). There would also be the choice of using SARSA or Q with $\lambda \neq 0$. Speed of convergence in conventional RL can change as a consequence of λ. There are, however, in general no predictions possible for which value the fastest convergence is obtained.

the current model is that with just a few assumptions we can produce a good approximation of actual rat navigation.

Relations to Hippocampus Modeling

Concerning the links of our study to hippocampus investigations, it was not our objective to create a general hippocampus model (Kali and Dayan, 2000; Becker, 2005). We do not distinguish between regions of hippocampus (dentate gyrus, CA1, CA3), nor do we model its inputs or the process of place cell development (Samsonovich and McNaughton, 1997; Tsodyks, 1999; Hartley et al., 2000). Instead we focused on the interaction between behavioral constituents (paths) with reinforcement learning in a place-field like function approximation system.

In the field of hippocampus-based navigation our model has similar properties to models by Arleo and Gerstner (2000); Krichmar et al. (2005); Strösslin et al. (2005). Those models try to incorporate many known details about the included brain structures and types of cells present, thus attempting to study hippocampal function. No special attention has been devoted in these studies to path formation and its influence on the learning. The question, however, arises to what degree the conclusions in the older studies might be influenced by our findings.

Relations to Machine Learning

We have emulated navigational learning by ways of reinforcement learning with function approximation, based on hippocampus-like place field representation. We used the SARSA algorithm, to stay closer to biological learning mechanisms, though Q-learning can be implemented in the same framework, as discussed above. The algorithm is similar to the one proposed by Reynolds (2002). However, we do not normalize the learning equation and our learning rates are, thus, independent of the number of activated features (activated place fields). In spite of this, the current algorithm produces convergent weights as well as convergent behavior in a conventional exploration-exploitation setup. On many occasions it produces optimal paths to reward. As our place field system spikes probabilistically, it is difficult, if not impossible, to provide a rigorous convergence proof. In general such proofs are notoriously hard to obtain for any function approximation system even under more relaxed conditions (Szepesvari and Smart, 2004). While of possible theoretical interest, these machine learning related issues may not relate directly to our more biologically-inspired model.

Concerning forgetting we observe that in machine learning, weight decay is known from the general purpose neural network learning literature as means to prevent weights from saturation (Werbos, 1988). In a reinforcement learning framework, weight decay has been used in several isolated studies, usually to produce agents who can adapt to changing environments (Yen and Hickey, 2004). Here we show how learning in a static environment can also benefit from forgetting mechanisms in preventing divergence.

Frustration in our setup is implemented as return to the home base if the reward was not reached in predefined number of simulation steps. Path-to-goal limiting is a natural option for any simulation of reinforcement learning; it is reasonable to stop a trial after some steps if the reward is not found (e.g. Glaubius and Smart (2004)). Here we used an even more complex path limiting process, where the allowed number of steps depends of the path-to-goal length in the previous epoch. In robotics applications with reinforcement learning, frustration is often included to switch the pattern of behavior when the goal is not reached (e.g. Okhawa et al. (1998)), which is not of relevance, though, for our system.

It is worth noting that pure exploration-exploitation strategies might in some cases be incompatible with the compliance requirements of machines. Especially in multi-joint robot arms the very jerky, wiggly movements obtained by random exploration might not be permissive as they might even damage such a machine (T. Asfour, personal communication). Hence, straighter exploration "paths" should be employed for reinforcement learning problems in these domains. The improved convergence found in our study could therefore help to better adapt RL-methods to such problems.

Acknowledgements

FW and PD acknowledge funding by Grant: BB/C516079/1 from the Biotechnology and Bioligical Sciences Research Council (U.K.); FW furthermore acknowledges funding by the European Commisson "PACO-PLUS".

We thank Dr. D. Sheynikhovich and Prof. W. Gerstner for drawing our attention to the path finding problem.

References

Ainge, J. A., Tamosiunaite, M., Wörgötter, F., and Dudchenko, P. A. (2007). Hippocampal CA1 place cells encode intended destination on a concatenated Y-maze. *J Neurosci*, page in revision.

Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biol Cybern*, 83(3):287–299.

Arleo, A., Smeraldi, F., and Gerstner, W. (2004). Cognitive navigation based on nonuniform Gabor space sampling, unsupervised growing networks, and reinforcement learning. *IEEE Trans Neural Netw*, 15(3):639–652.

Barto, A. (1995). Adaptive critics and the basal ganglia. In Houk, J. C., Davis, J. L., and Beiser, D. G., editors, *Models of Information processing in the basal ganglia*, pages 215–232. MIT Press, Cambridge, MA.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 13, pages 835–846.

Becker, S. (2005). A computational principle for hippocampal learning and neurogenesis. *Hippocampus*, 15(6):722–738.

Brown, M. A. and Sharp, P. E. (1995). Simulation of spatial learning in the Morris water maze by a neural network model of the hippocampal formation and nucleus accumbens. *Hippocampus*, 5(3):171–188.

Eilam, D. (2004). Locomotor activity in common spiny mice (Acomys cahirinuse): The effect of light and environmental complexity. *BMC Ecology*, 4(16):4–16.

Etienne, A. S., Maurer, R., and Seguinot, V. (1996). Path integration in mammals and its interaction with visual landmarks. *J Exp Biol*, 199(Pt 1):201–209.

Foster, D. J., Morris, R. G., and Dayan, P. (2000). A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus*, 10(1):1–16.

Glaubius, R. and Smart, W. D. (2004). Manifold representations for value function approximation. In *Proceedings of the the AAAI-04 Workshop on Learning and Planning in Markov Processes*, pages 13–189.

Hartley, T., Burgess, N., Lever, C., Cacucci, F., and O'Keefe, J. (2000). Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus*, 10(4):369–379.

Hollup, S. A., Molden, S., Donnett, J. G., Moser, M. B., and Moser, E. I. (2001). Accumulation of hippocampal place fields at the goal location in an annular watermaze task. *J Neurosci*, 21(5):1635–1644.

Houk, J. C., Adams, J. L., and Barto, A. G. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. In Houk, J. C., Davis, J. L., and Beiser, D. G., editors, *Models of information processing in the basal ganglia*, pages 249–270. MIT Press, Cambridge, MA.

Kaelbling, L. P., Littman, M., and Moore, A. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Kali, S. and Dayan, P. (2000). The involvement of recurrent connections in area CA3 in establishing the properties of place fields: A model. *Journal of Neuroscience*, 20:7463–7477.

Krichmar, J. L., Seth, A. K., Nitz, D. A., Fleischer, J. G., and Edelman, G. M. (2005). Spatial navigation and causal analysis in a brain-based device modeling cortical-hippocampal interactions. *Neuroinformatics*, 3(3):197–221. Comparative Study.

Morris, G., Nevet, A., Arkadir, D., Vaadia, E., and Bergman, H. (2006). Midbrain dopamine neurons encode decisions for future action. *Nat Neurosci*, 9(8):1057–1063.

Morris, R. (1984). Developments of a water-maze procedure for studying spatial learning in the rat. *J Neurosci Methods*, 11(1):47–60.

Niv, Y., Daw, N. D., and Dayan, P. (2006). Choice values. *Nat Neurosci*, 9(8):987–988. Comment.

Okhawa, K., Shibata, T., and Tanie, K. (1998). Method for generating of global cooperation based on local communication. In *Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 108–113, Victoria, B.C., Canada.

Reynolds, S. I. (2002). The stability of general discounted reinforcement learning with linear function approximation. In *UK Workshop on Computational Intelligence (UKCI-02)*, pages 139–146, Birmingham, UK.

Samsonovich, A. and McNaughton, B. L. (1997). Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model. *The Journal of Neuroscience*, 17(15):5900–5920.

Save, E., Nerad, L., and Poucet, B. (2000). Contribution of multiple sensory information to place field stability in hippocampal place cells. *Hippocampus*, 10(1):64–76.

Schultz, W. (2002). Getting formal with dopamine and reward. *Neuron*, 36:241–263.

Schultz, W. (2007). Reward signals. Scholarpedia, *http://www.scholarpedia.org/article/Reward_Signals*.

Strösslin, T., Sheynikhovich, D., Chavarriaga, R., and Gerstner, W. (2005). Robust self-localisation and navigation based on hippocampal place cells. *Neural Netw*, 18(9):1125–1140.

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

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3:9–44.

Szepesvari, C. and Smart, W. D. (2004). Interpolation-based Q-learning. *Twenty-First International Conference on Machine Learning (ICML04)*, 21:791–798.

Tesauro, G. (1995). Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–67.

Tsodyks, M. (1999). Attractor neural network models of spatial maps in hippocampus. *Hippocampus*, 9(4):481–489.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8:279–292.

Werbos, P. J. (1988). Backpropagation: Past and future. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 343–353. IEEE Press.

Whishaw, I. Q., Hines, D. J., and Wallace, D. G. (2001). Dead reckoning (path integration) requires the hippocampal formation: evidence from spontaneous exploration and spatial learning tasks in light (allothetic) and dark (idiothetic) tests. *Behav Brain Res*, 127(1-2):49–69.

Wilson, M. A. and McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space. *Science*, 261(5124):1055–1058.

Wörgötter, F. and Porr, B. (2005). Temporal sequence learning, prediction, and control: a review of different models and their relation to biological mechanisms. *Neural Comput*, 17(2):245–319.

Wörgötter, F. and Porr, B. (2007). Reinforcement learning. Scholarpedia, *http://www.scholarpedia.org/article/Reinforcement_Learning*.

Yen, G. G. and Hickey, T. (2004). Reinforcement learning algorithms for robotic navigation in dynamic environments. *ISA Transactions*, 43(2):217–230.

Zadicario, P., Avni, R., Zadicario, E., and Eilam, D. (2005). 'Looping': an exploration and navigation mechanism in a dark open field. *Behavioural Brain Research*, 159:27–36.

Tables

*Table 1: Default parameters used for modeling experiments in a standard setup. *Note, additional justification for these default parameters is given in section 'Place field size and density'.*

Parameter type	**Parameter name**	**Value**
SARSA-learning (see Appendix)	learning rate α	0.7
	discount factor γ	0.7
Environment/steps	size	10000 × 10000
	step size	400
	noise on the step size	± 100
	reward size	1000 × 1000
Place fields*	number	500
	width, through σ	400
	scaling factor A	2.5
Learning strategies	exploration probability p_e in E	0.2
	probabilities for S	
	p_1	0.5
	p_2	0.156
	p_3	0.063
	p_4	0.031
	p_5	0
	p_6	0.031
	p_7	0.063
	p_8	0.156
	weighting factor w in S	0.5
	forgetting factor c_f in F	0.9995
	zero weight threshold t_f in F	10^{-6}
	starting path length in steps in L	200
	path increase step in L, c_l	5
	path limit in steps for any strategy	300

Table 2: Number of cells required to achieve a certain coverage given the field width σ. Value of $\sigma = 400$, $n = 500$ have been used for most other experiments. Labels (a-l) refer to the panels in Fig. 7.

	Field Width (σ)			
Coverage	**200**	**400**	**600**	**800**
4.5	2000 (a)	500 (b)	230 (c)	140 (d)
2.4	1100 (e)	300 (f)	130 (g)	80 (h)
0.8	350 (i)	100 (j)	80 (k)	25 (l)

Appendix: Reinforcement learning with function approximation

Reinforcement learning is a procedure where a value function $V(s)$ over states s develops as an agent acts in its environment and attains goals. In RL with delayed reward, the function shows a gradient towards a goal. In the Q and SARSA learning approaches (Watkins and Dayan, 1992; Kaelbling et al., 1996), instead of the state value function, the state-action value function $Q(s, a)$ (short: *action value function*) is developed, where s denotes a state and a an action. Action-value functions describe values of concurrent actions in every state, and can be directly used for making a decision on which action to perform.

Q-learning is described by the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \tag{5}$$

Where $Q(s_t, a_t)$ is the action value function at time step t, r_{t+1} is a reward obtained with action a_t, α is the learning rate and γ a discount factor. SARSA learning differs by a single aspect that the current action value is updated according to the value of the next *actual* action, but not by the *best possible* next action, as in Q learning:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \tag{6}$$

Hence, SARSA is designed to work on-policy, which means that learning takes place as an agent moves in the state space according to the path that was actually performed. Evidence exists that animals follow an on-policy, rather than an off-policy, learning strategy (Morris et al. (2006), see also commentary by Niv et al. (2006)). Hence in this study we are investigating the SARSA algorithm and only sometimes comment on Q-learning.

For big and/or continuous state spaces, function approximation methods need to be used, where the action value function is a function of more abstract and wider-embracing entities commonly called *features* in the RL-literature. We define normalized Q-values by:

$$Q(s_t, a_t) = \sum_i \theta_{i,a_t} \Phi_i(s_t) / \sum_i \Phi_i(s_t) \tag{7}$$

where $\Phi_i(s_t)$ are the features over the state space, and θ_{i,a_t} are the adaptable weights binding features to actions (Reynolds, 2002).

We assume that a place cell i produces spikes with a scaled Gaussian-shaped probability distribution:

$$p(\delta_i) = A \exp(\delta_i^2/\sigma^2) \tag{8}$$

where δ_i is the distance from the i-th place field center to the sample point (x, y) on the trajectory, σ defines the width of the place field, and A is a scaling factor. In the areas where the values of this scaled distribution are above 1, cells fire with a probability of 1.

We then use the actual place field spiking to determine the values for features Φ_i, $i = 1, .., n$, which take the value of 1, if place cell i spikes at the given moment on the given point of the trajectory of the model animal, otherwise it is zero:

$$\Phi_i(s_t) = \begin{cases} 1 & \text{if place cell i spikes at } \mathrm{s_t} \\ 0 & \text{else.} \end{cases} \tag{9}$$

SARSA learning then can be described by:

$$\theta_{i,a_t} \leftarrow \theta_{i,a_t} + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - \theta_{i,a_t})\Phi_i(s_t) \tag{10}$$

where $\theta_{i,a}$ is the weight from the i-th place cell to action(-cell) a, and state s is defined by (x, y), which are the actual coordinates of the model animal in the field.

We sum over all features, but in each place only a specific subset of cells will fire rendering their corresponding features non-zero. Note that function $\Phi_i(s)$ has a probabilistic nature in our approach, differently from usual features used for function approximation in RL. The update rule (Eq. 10) we use has a straightforward biological interpretation: the weight of a particular action is increased at the given place if this weight leads either to a reward, or if it leads on to pieces of an already known rewarding path. In the latter case this results from the non-zero $Q(s_{t+1}, a_{t+1})$-values in the next state.