Deep reinforcement learning in a spatial navigation task: Multiple contexts and their representation

Nicolas Diekmann (nicolas.diekmann@rub.de)
Institute for Neural Computation, Ruhr University Bochum, Universitätsstr. 150
Bochum, 44801, Germany

Thomas Walther (thomas.walther@rub.de)
Institute for Neural Computation, Ruhr University Bochum, Universitätsstr. 150
Bochum, 44801, Germany

Sandhiya Vijayabaskaran (sandhiya.vijayabaskaran@rub.de)
Institute for Neural Computation, Ruhr University Bochum, Universitätsstr. 150
Bochum, 44801, Germany

Sen Cheng (sen.cheng@rub.de)
Institute for Neural Computation, Ruhr University Bochum, Universitätsstr. 150
Bochum, 44801, Germany

Abstract

Deep learning has recently been combined with Q-learning (Mnih et al., 2015) to enable learning difficult tasks such as playing video games based only on visual input. Stable learning in the in the deep Q network (DQN) is facilitated by the use of memory replay, which means that previous experiences are stored and sampled from during an offline learning period. We evaluate the DQN's ability to learn and retain multiple variations of a spatial navigation task in a virtual environment. Task variations are presented in visually distinct contexts by varying light conditions and environmental textures. Replay memory capacity is varied to measure its effect on task retention. The representations of multiple contexts learned by the DQN agents are analyzed and compared. We show that DQN agents learn a preference for common actions early on, irrespective of replay memory capacity. A limited replay memory causes agents to confuse state-values. Furthermore, we find that contexts are quickly forgotten as soon as corresponding experiences are no longer available in the replay memory.

Keywords: Reinforcement learning; Q-learning; Deep learning; Spatial navigation

Introduction

In the real world, humans and other animals learn strategies that help them survive by interacting with their environment. This type of learning is known as reward learning (Pierce & Cheney, 2004) and is used in both classical and operant conditioning (Bouton, 2007; Rescorla, 1988). Behaviors are reinforced according to rewards and punishments an animal receives (White, 2011; Schultz, 2015). Strategies have to be learned sequentially. This biological reinforcement inspired a field of machine learning which is known as reinforcement learning (Sutton & Barto, 2018). In recent years, reinforcement learning was combined with deep learning (Goodfellow, Bengio, & Courville, 2016) in different algorithms (Silver et al., 2016; Mnih et al., 2016). In this work, we use a deep reinforcement learning algorithm known as the deep Q network (DQN) (Mnih et al., 2015). The DQN combines Q-learning (Watkins, 1989) with deep neural networks (DNNs), non-linear function approximators. The stability issues encountered when combining these two methods (Tsitsiklis & Van Roy, 1997) were overcome in DQN by using experience replay (Lin, 1992). Memory replay has been shown to alleviate problems associated with catastrophic forgetting in DQN (Kirkpatrick et al., 2017; Atkinson, McCane, Szymanski, & Robins, 2018), which is similar to neurobiological theories of memory replay driven by the hippocampus (McClelland, McNaughton, & O'Reilly, 1995). However, in the machine learning studies millions of learning steps are required, which is not realistic for biological agents. Therefore, we investigate how a DQN learns multiple variations of a spatial navigation task using a small number

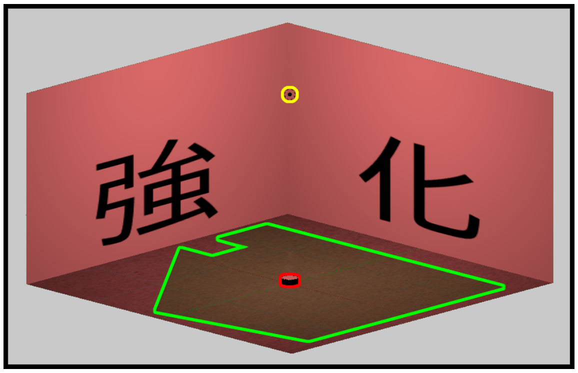

Figure 1: Virtual environment used in the computational study. Room walls are textured with monochrome images. Outlined are the agent (red), the area over which the topology graph is spanned (green) and the light source (yellow).

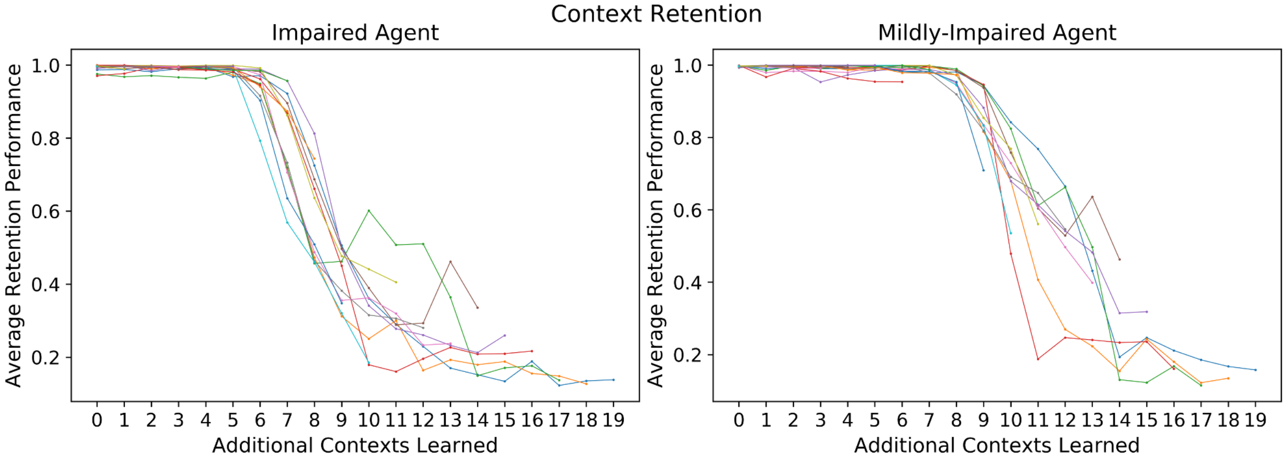

Figure 2: Context retention performance averaged for impaired (left) and mildly-impaired (right) DQN agents. Horizontal axis shows additional contexts learned. Both agents keep retention performance at a high level while corresponding experiences are still in the replay memory. The mildly-impaired agent's larger memory capacity allows it to retain high performance longer than the impaired agent. For the mildly-impaired agent retention performance appears to drop more gradually for most contexts.

of learning trials. We are interested in particular in the role of the replay memory's capacity. Furthermore, we compare representations learned by DQN agents with different memory capacities.

Methods

The following sections provide descriptions for the virtual environment used in the experiments, the network architecture used and the experimental setup. Experiments are performed in the "Hippocampus Project" developed by Walther et al. (2018).

The Virtual Environment

We train DQN agents on multiple variations of a spatial navigation task modeled after the multi-context version of the Morris water maze (Snyder, Clifford, Jeurling, & Cameron, 2012). The virtual environment used in our experiments consists of a quadratic room with a light source in its center (see Fig. 1). Visually distinct contexts are created by changing the room's wall textures and the light source's color. The task is to navigate on a topology graph from random starting nodes to a goal node. The topology graph spans a predefined area within the room (Fig. 3) and is obtained by Delaunay triangulation (Lee & J. Schachter, 1980). Rewards provided to the agents replicate conditions experienced by rats in the Morris water maze (Morris, 1981): As the DQN agents navigate on the graph they receive a negative reward of -1 in each step. In each context, one node in the graph serves as the goal. On arriving at the goal, the DQN agents receive a positive reward of $+1$. Trials end either when the DQN agents reach the goal node or after 100 steps. Because the number of neighbor nodes varies, an action space of 8 is chosen. At every node the action a_i will move the agent to the neighbor node n_i. For actions with no corresponding neighbor node the agents will stay at their current node. The graph used in our experiments has a total of 21 nodes.

At each node the agents make observations. Observations consist of panoramic color images with a FOV of $360°$ by $90°$. Image orientation is fixed and always faces "north". Image dimensions of $120x20x3$ are chosen for the input of the DQN agents.

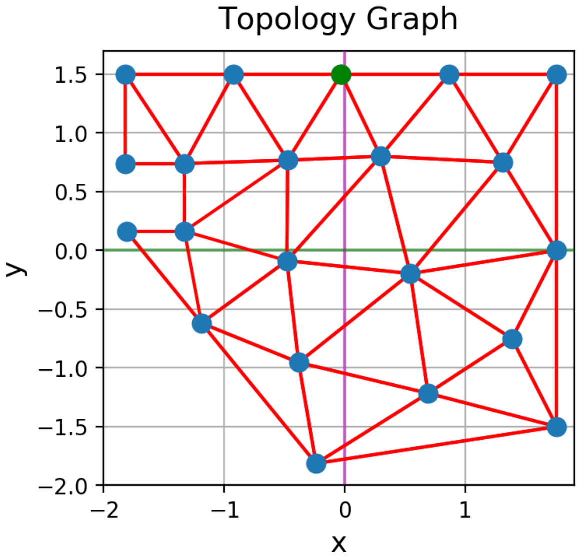

Figure 3: Topology graph with 21 nodes that the DQN agents navigate on in our experiments. Delaunay triangulation (Lee & J. Schachter, 1980) is used to obtain the graph. In each context one node is randomly chosen as the goal (colored in green).

Network Architecture and Parameters

We use a convolutional neural network (CNN) (Lecun, Bengio, & Hinton, 2015) to process the visual input. The network consists of two 2-dimensional convolutional layers, each of which is followed by a max pooling layer for dimensional reduction. The first and second convolutional layers implement 32 4x4x3 filters and 64 3x3x32 filters respectively. The last max pooling layer is flattened and fed into a fully connected layer with 512 units. In order to boost sample efficiency we employ Wang et al.'s (2015) dueling architecture, which splits the network into state-value and action-advantage streams. Q-values are obtained by aggregating state-value and action-advantage streams. Up until the network split, ReLU activations are used for all layers. State-value and action-advantage streams use linear activations.

The DQNs are trained with mini-batches of size 64. For exploration, an ε-greedy strategy with $\varepsilon = 0.2$ is chosen. As an optimizer Adam with a learning rate $\eta = 0.001$ is used. Future rewards are discounted with a factor of $\gamma = 0.9$.

Experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ are stored in a dataset $\mathcal{D}_t = \{e_1, ..., e_t\}$ called the replay memory. During learning, small batches are drawn uniformly at random from the replay memory to update the DQN's weights.

Experimental Setup

We train DQN agents with three different replay memory capacities:

1. 'Intact' agent with a replay memory that can store all experiences.
2. 'Impaired' agent with a replay memory that can store the most recent 15,000 experiences.
3. 'Mildly-impaired' agent with a replay memory that can store the most recent 21,000 experiences.

We generate 20 random contexts with random goal nodes to be presented to the agent. Contexts are presented sequentially for 500 trials each.

Results

Context Retention

We measure context retention by comparing the length of the selected path to an optimal path. If the agent selects an optimal path it receives a performance rating of 1 and receives a performance rating of 0 if the agent has not reached the goal within 21 steps (meaning it cannot find the goal). Performance ratings are averaged for all starting nodes. We obtain optimal path lengths using scikit-learn's *graph_shortest_path_module*.

Figure 2 shows retention performance over additional contexts learned. For the intact agent performance for all contexts is kept close to 1 after they are learned. The impaired agent retains a context for roughly 6-7 additionals contexts before performance starts to degrade considerably. Retention performance falls quickly but not instantly. Contexts are retained for roughly 2 more additional contexts by the mildly-impaired agent.

Network Representations

Network layer activations of action-advantage and state-value streams are recorded at each node of the topology graph in each context. For dimensionality reduction we use principal component analysis (PCA) (Abdi & Williams, 2010) to project the activations of the action-advantage stream to the first two principal components. Projections are computed every 500 trials. We analyze representations learned by the intact and impaired DQN agents.

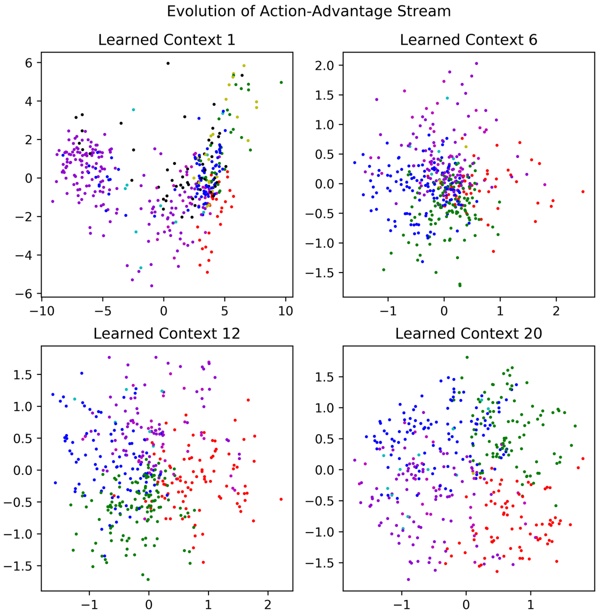

Figure 4: PCA projections of the intact DQN agent's action-advantage stream to the first two principal components (for dimensionality reduction). Shown are different stages during the experiment: After learning context 1 (top left), after learning context 6 (top right), after learning context 12 (bottom left) and after learning all contexts (bottom right). Action a_0 to a_3 (indicated by different colors) start forming a distinct clustering after context 6 is learned.

Intact Agent We find that the state-value stream learns goal node distances. State-values are ordered and form 5 clusters. Action-advantages show a distinct clustering of actions. Actions a_0 to a_3 form clustered quadrants. Other actions form loose clusters but require projection to three dimensions (not shown). When looking at the evolution of the action-advantage stream, we find that said clustering emerges as early as after learning the 6th context (Fig. 4). For the state-value stream we find that goal distances are learned gradually over the course of learning. This is expected, because the agent cannot know the position of a goal it has never found before.

Impaired Agent The impaired DQN agent is not able to retain goal node distances in its state-value stream for all contexts. Only when corresponding experiences are in the replay memory can the DQN retain correct distances. As is the case for the intact agent, the impaired agent's action-advantage stream forms distinct clusters for actions a_0 to a_3.

Discussion

We find that context retention performance strongly depends on the replay memory's capacity. This shows that memory replay also benefits smaller learning problems. The representations learned by the DQN agents in our study reveal that the state-value stream learns goal node distances, but can only retain them when corresponding experiences are still in the replay memory. Furthermore, the action-advantage stream forms clusters of actions which appear in PCA projections as quadrants. The DQN agents learn, irrespective of memory capacity, a preference for actions a_0 to a_3 and they do so early in the experiments. Thus, the agents learn to exploit the structure of the topology graph since nodes have roughly 4 neigbor nodes on average.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project number project number 316803389 SFB 1280, project A14.

References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433-459. doi: 10.1002/wics.101

Atkinson, C., McCane, B., Szymanski, L., & Robins, A. V. (2018). Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting. *CoRR*, *abs/1812.02464*.

Bouton, M. E. (2007, 01). Learning and behavior: A contemporary synthesis. Sunderland, MA, US: Sinauer Associates.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (http://www.deeplearningbook.org)

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526. doi: 10.1073/pnas.1611835114

Lecun, Y., Bengio, Y., & Hinton, G. (2015, May). Deep learning. *Nature*, *521*, 436-444. doi: 10.1038/nature14539

Lee, D., & J. Schachter, B. (1980, 06). Two algorithms for constructing a delaunay triangulation. *International Journal of Parallel Programming*, *9*, 219-242. doi: 10.1007/BF00977785

Lin, L.-J. (1992, May 01). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, *8*(3), 293–321. doi: 10.1007/BF00992699

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychologial Review*, *102*, 419-457.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., ... Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *CoRR*, *abs/1602.01783*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... Hassabis, D. (2015, Feb 25). Human-level control through deep reinforcement learning. *Nature*, *518*, 529 EP -.

Morris, R. G. (1981). Spatial localization does not require the presence of local cues. *Learning and Motivation*, *12*(2), 239-260. (Exported from https://app.dimensions.ai on 2019/02/24) doi: 10.1016/0023-9690(81)90020-5

Pierce, W. D., & Cheney, C. D. (2004). Behavior analysis and learning. In (3rd ed.). Lawrence Erlbaum Associates.

Rescorla, R. (1988). Pavlovian conditioning. it's not what you think it is. *The American psychologist*, *43 3*, 151-60.

Schultz, W. (2015). Neuronal reward and decision signals: From theories to data. *Physiological reviews*, *95*, 853-951.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*, 484-489. doi: 10.1038/nature16961

Snyder, J. S., Clifford, M. A., Jeurling, S. I., & Cameron, H. A. (2012). Complementary activation of hippocampalcortical subregions and immature neurons following chronic training in single and multiple context versions of the water maze. *Behavioural Brain Research*, *227*(2), 330 - 339. (Special Issue of Behavioural Brain Research on Neurogenesis and Behavior) doi: https://doi.org/10.1016/j.bbr.2011.06.025

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. In (2nd ed.). The MIT Press.

Tsitsiklis, J. N., & Van Roy, B. (1997, May). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, *42*(5), 674-690. doi: 10.1109/9.580874

Walther, T., Vijayabaskaran, S., Diekmann, N., Schüler, M., Wiskott, L., & Cheng, S. (2018). A model of context-dependent spatial learning in rodents. In *Bernstein conference 2018.* doi: 10.12751/nncn.bc2018.0234

Wang, Z., de Freitas, N., & Lanctot, M. (2015). Dueling network architectures for deep reinforcement learning. *CoRR*, *abs/1511.06581*.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Unpublished doctoral dissertation, King's College, Cambridge, UK.

White, N. M. (2011). Reward: What is it? how can it be inferred from behavior? In J. A. Gottfried (Ed.), *Neurobiology of sensation and reward.* CRC Press.