Memory Lens: How Much Memory Does an Agent Use?

Christoph Dann CDANN@CDANN.NET
Machine Learning Department, Carnegie Mellon University
Katja Hofmann KATJA.HOFMANN@MICROSOFT.COM
Sebastian Nowozin SEBASTIAN.NOWOZIN@MICROSOFT.COM
Machine Intelligence and Perception Group, Microsoft Research, Cambridge UK

Abstract

We propose a new method to study the internal memory used by reinforcement learning policies. We estimate the amount of relevant past information by estimating mutual information between behavior histories and the current action of an agent. We perform this estimation in the passive setting, that is, we do not intervene but merely observe the natural behavior of the agent. Moreover, we provide a theoretical justification for our approach by showing that it yields an implementation-independent lower bound on the minimal memory capacity of any agent that implement the observed policy. We demonstrate our approach by estimating the use of memory of DQN policies on concatenated Atari frames, demonstrating sharply different use of memory across 49 games. The study of memory as information that flows from the past to the current action opens avenues to understand and improve successful reinforcement learning algorithms.

1. Introduction

Can you understand the complexity of an agent just by observing its behavior? Herbert Simon provided a vivid example by imagining an ant moving along a beach (Simon, 1996). He observed that the path the ant takes while walking toward a certain destination appears highly irregular and hard to describe. This complexity may indicate a sophisticated decision making in the ant, but Simon postulated that instead the ant follows very simple rules and the observed complexity of the path stems from the complexity in the environment.

In this work we aim to understand the complexity of agents acting according to a fixed policy in an environment. In particular, we are interested in the *memory* of an agent, that is, its ability to use past observations to inform future actions. We do not assume a specific implementation of the agent, but instead observe its behavior to derive statements about its memory.

The study of memory in agents is important for two reasons. *First*, most state-of-the-art reinforcement learning approaches assume that the environment is a Markov decision process (MDP) (Puterman, 1994), but in contrast most real-world tasks have non-Markov structure and are only partially observable by the agent. In such environments, optimal decisions may not only depend on the most recent observation but on the entire history of interactions. That is, to solve a task optimally or even just reasonably well, an agent might need to remember all previous observations and actions taken (Singh et al., 1994; Krishnamurthy et al., 2016). *Second*, in many recent algorithms the policy has access to memory of certain fixed capacity. Popular choices include using a fixed window of history (e.g. the last four observations as in Mnih et al. (2015)) or recurrent neural networks as adaptive memory (Heess et al., 2015; Li et al., 2015). These approaches are practically successful for specific tasks, but it is unclear how much memory capacity they actually use and how much memory capacity to use for different tasks. Our approach allows us to study the use of memory—measured in bits over time—independent of the implementation of the agent.

Our method of estimating memory works as follows. We assume we can observe all interactions of an agent's policy with the environment in the form of sequences of observation-action-reward triples. We then estimate the mutual information between actions and parts of the history. This approach treats the agent and the environment as black boxes which in principle also allows the application of the method to humans or animals as agents.

We provide a theoretical justification of the method in two steps. First, we formally define the minimal *memory capacity* required to reproduce a given policy. Second, we connect our practical estimation method with this formal notion by showing that our method estimates a lower bound to the memory capacity. To demonstrate the usefulness of our approach we analyze the memory capacity of the state-of-the-art Deep Q-Network policies trained on 49 Atari 2600 games (Mnih et al., 2015). In summary our work makes the following contributions:

- A practical method for estimating the memory use of an agent's policy from its behavior;
- A theoretical justification in terms of minimum memory capacity;
- Insight into the memory use of successful DQN policies on Atari games.

2. Problem Setting and Notation

We consider the following setting: an agent interacts at discrete times $t = 1, 2, \ldots$ with a stochastic environment by (1) making an observation $X_t \in \mathcal{X}$, (2) taking an action $A_t \in \mathcal{A}$ and (3) receiving a scalar reward $R_t \in \mathbb{R}$ at each of these times. For notational convenience, we denote the quantities at time t by $Z_t = (X_t, A_t, R_t)$ and the concatenation of several time steps by $Z_{k:t} = (Z_k, \ldots, Z_{t-1}, Z_t)$. While X_t and R_t are determined by the environment, the action is sampled from the agent's policy and may depend on the entire previous history $Z_1, \ldots Z_{t-1}, X_t$.

We assume that the environment is stochastic but not necessarily Markov. We are interested in agents that have mastered a task, which means that learning has mostly ended and the policy changes slowly if at all. We thus formally assume that the agent's policy is fixed for notational simplicity.[1] A trajectory $\xi = (Z_1, Z_2, \ldots, Z_n)$ consisting of n time-steps is therefore a random vector sampled from a fixed distribution $\xi \sim P$. We further assume that Z_t can take only finitely many values. Given one or more trajectories $\xi_1, \xi_2, \cdots \sim P$ of the agent interacting with the environment, our goal is to estimate the amount of memory required by any agent that implements the observed policy.

3. Method: Memory Lens

Memory allows the action A_t to depend not only on the most recent observation X_t but also on the previous history $Z_1, \ldots Z_{t-1}$. One intuitive notion to describe the amount of memory used for some action A_t is mutual information of action A_t and history $Z_1, \ldots, Z_{t-1}$ given X_t, that is, $I(A_t; Z_{1:t-1}|X_t) = \mathbb{E}\left[D_{KL}(P(A_t|X_t, Z_{1:t-1}) \| P(A_t|X_t))\right]$. This *conditional mutual information* quantifies the information in bits or nats about action A_t one gains by

1. This assumption is not crucial. In the case of changing policies, our results hold for the mixture of policies followed by the agent.

getting to know $Z_{1:t-1}$ when one already knows the value of X_t. If this quantity is zero for all t, then the policy is Markov, that is, the action depends only on the most recent observation. If it is nonzero, every implementation of the agent has to use at least some form of memory. Our approach is to estimate the following mutual information quantities

$$M_0 := I(A_t; X_t); \qquad M_1 := I(A_t; Z_{t-1}|X_t); \qquad M_2 := I(A_t; Z_{t-2}|X_t, Z_{t-1}); \tag{1}$$

$$M_3 := I(A_t; Z_{t-3}|X_t, Z_{t-2:t-1}); \qquad \dots \qquad M_{t-1} := I(A_t; Z_1|X_t, Z_{2:t-1}). \tag{2}$$

Each entry M_i of M quantifies how much additional information about A_t can be gained when considering history of length i instead of only $i-1$. The first entry M_0 is the amount of information X_t shares with A_t.

3.1 Estimating Mutual Information

Mutual information and conditional mutual information can be written as differences of entropy terms, $I(A_t; X_t) = H(A_t) + H(X_t) - H(A_t, X_t)$ and

$$I(A_t; Z_{1:t-k}|X_t, Z_{t-k+1:t-1}) = H(A_t, X_t, Z_{t-k+1:t-1}) + H(X_t, Z_{t-k:t-1}) \tag{3}$$

$$- H(A_t, X_t, Z_{t-k:t-1}) - H(X_t, Z_{t-k+1:t-1}), \tag{4}$$

where the entropy of multiple random variables is defined by the entropy of their joint distribution. We can therefore estimate M_i by estimating the individual entropy terms.

We use the entropy estimator by Grassberger (2003) due to its simplicity and computational efficiency; alternatives (plug-in, Nemenman et al. (2002); Hausser and Strimmer (2009)) yielded very similar results in our experimental evaluations. The Grassberger (2003) estimate of the entropy in nats of a random quantity Y of which we have seen k different values, each $n_1, \dots, n_k$ times, is (Nowozin, 2012) $\hat{H}(Y) = \ln(N) - \frac{1}{N}\sum_{i=1}^{k} G(n_i)$, where $N = \sum_i n_i$ is the total number of samples and $G(n) = \psi(n) + \frac{(-1)^n}{2}\left(\psi\left(\frac{n+1}{2}\right) - \psi\left(\frac{n}{2}\right)\right)$, with ψ being the *digamma function*.

3.2 Test of Significance

In practice the sample size available for estimating the conditional mutual information is limited. Therefore, our estimates will be affected both by bias and statistical variation. To prevent invalid conclusions due to bias and variation we use a simple permutation test as follows. We take the original set of samples $\{(Z_{t-i:t-1}, X_t, A_t)\}$ for estimating the conditional mutual information $\hat{M}_i$ and replace the last action by sampling a new action $\tilde{A}_t \sim \hat{p}(A_t)$ from the empirical marginal of A_t. We then compute the conditional mutual information $\tilde{M}_i$ w.r.t. the modified samples $\{(Z_{t-i:t-1}, X_t, \tilde{A}_t)\}$. This action-resampling process is repeated 100 times to obtain the ordered sequence $\tilde{M}_i^{(1)}, \dots \tilde{M}_i^{(100)}$ with $\tilde{M}_i^{(j)} \leq \tilde{M}_i^{(j+1)}$ for all j. We consider memory use significant if $\hat{M}_i$ is above the 95% percentile of this set, i.e., $\hat{M}_i \geq \tilde{M}_i^{(95)}$.

4. Experimental Results

We trained Deep Q-Network policies for 50 million time steps on 49 Atari games. The network structure as well as all learning parameters have been chosen to match the setting by Mnih et al. (2015). Each policy chooses with probability $\epsilon = 0.05$ an available action uniformly at random and otherwise takes the action that maximizes the learned Q-function.

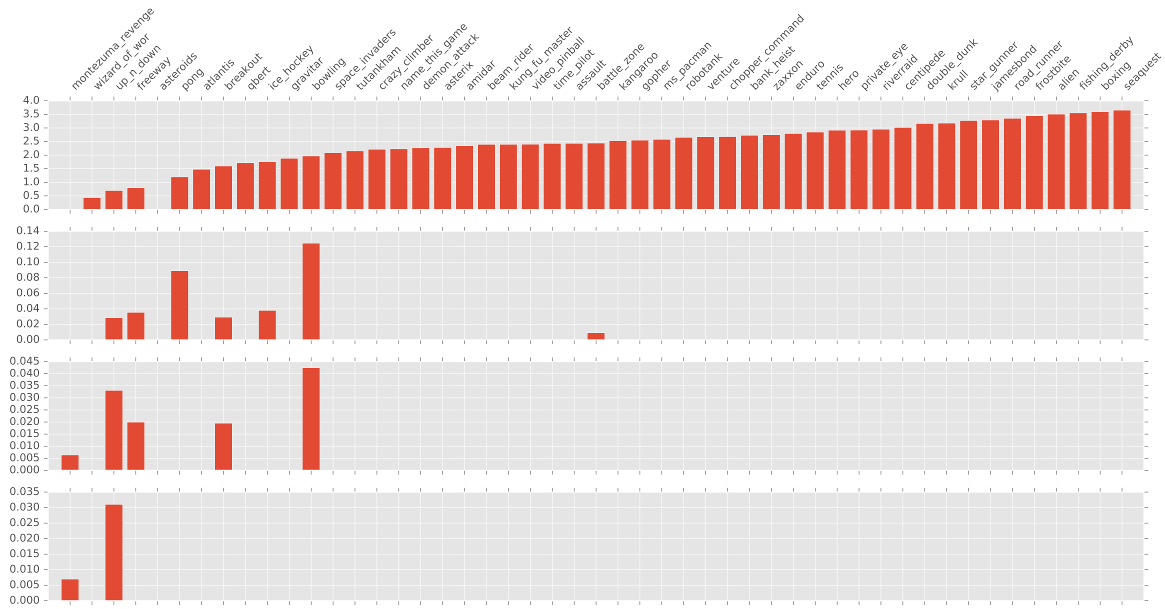

Figure 1: Estimated use of memory by DQN policies on 49 Atari 2600 games. The top bar chart shows the mutual information between the chosen action and the current game screen; The three plots below visualize the additional information used by the policies from previous frames in temporally decreasing order.

The Q-function takes as input the last four frames of the games as 84×84 pixel grayscale images. We can interpret this as the agent having memory to perfectly store the last 4 observations. We applied our memory lens method and estimated to what extent each of these observations are actually used when making decisions in each game.

We recorded 10000 games played by each of the fully trained 49 policies. For each policy we used these 10000 trajectories of length up to at most 10000 time steps to estimate the memory use. Since we know that the policies are stationary, we expect their use of memory to be fairly stationary too. We therefore did not estimate M for the actions at each time t individually but aggregated samples for all actions.

The results are shown in Figure 1. The top bar plot shows the mutual information estimate $\hat{M}_0$ of action and most recent frame (requires no memory) and the plots below show $\hat{M}_1$, $\hat{M}_2$ and $\hat{M}_3$ for the policies of each game. Only bars are displayed that indicate statistically significant dependencies (see Section 3.2).

5. Formalization of Memory

In this section we provide a more formal definition of amount of memory required to implement an agent's behavior and relate the quantities M estimated by our memory lens approach to it. For the sake of conciseness, we focus on finite-horizon episodic decision problems with a fixed horizon of H. A single episode ξ is then an element of $\mathcal{Z}^H$. We use the short-hand notation $[H] = \{1, 2, 3, \dots, H\}$. Assume an abstract model of memory where the state of memory can take $K \in \mathbb{N}$ different values. Following the formalization by Chatterjee et al. (2010), we define:

Definition 1 *A* memory function $g : [H] \times \mathcal{Z} \times [K] \to [K]$ *maps for each time step the previous memory and current observations to a new memory configuration. The policy* π *of an agent can be implemented with* K *memory if and only if there is a memory function* g *so that* $P(A_t|X_t, Y_{t-1}) = P(A_t|X_t, Z_{1:t-1})$ *for all* $t \in [H]$ *where* Y_t *denotes the memory configuration after time* t*. That is* $Y_t = g(t, Z_t, Y_{t-1})$ *for* $t \in [H]$ *and* $Y_0 = 1$*. We say that* g *is a memory function for* π *in this case and denote the set of memory functions with capacity* K *for* π *by* $\mathcal{M}_{M,\pi}$*.*

Note that this abstract model of memory is very general and, for example, recurrent neural networks can be considered a direct implementation of it. We can formally define the minimum amount of memory required by a policy as

Definition 2 *The memory capacity* $\mathcal{C}(\pi)$ *of a policy* π *is the smallest amount of memory capacity required to reproduce this policy. Formally, it is* $\mathcal{C}(\pi) = \min\{K \in \mathbb{N} : \mathcal{M}_{K,\pi} \neq \emptyset\}$*.*

If the distribution P from which the episodes are sampled from is known, $\mathcal{C}(\pi)$ can be computed in finite time (there are only finitely many memory functions). This implies that in principle $\mathcal{C}(\pi)$ can be estimated just from observations by estimating P and then computing $\mathcal{C}(\pi)$. However, to determine $\mathcal{C}(\pi)$ one has to perform tests on equality of conditional probabilities (condition in Def. 1). Each conditional probability has to be estimated accurately which requires infeasibly many samples for any problem beyond simple toy settings. Instead, the mutual information quantities M used in our method are much easier to estimate.

5.1 $\sum_{i>0} M_i$ is a lower bound on $\log \mathcal{C}(\pi)$

The next corollary states that for any t the sum of all conditional mutual quantities $M_1, \ldots, M_{t-1}$ estimated by our memory lens approach (excluding M_0) is a lower bound on the log-memory capacity.

Corollary 1 *For all* $t \in [H]$, $\sum_{i=1}^{t-1} M_i = \sum_{i=1}^{t-1} I(A_t; Z_i|X_t, Z_{i+1:t-1}) \leq \log \mathcal{C}(\pi)$*.*

Instead of proving this corollary directly, we show a stronger version in the theorem below. This theorem allows to restrict the measure P to an event that can be decided based on the history and shows that this still results in a valid lower bound on $\log \mathcal{C}(\pi)$. In some scenarios one might have prior intuition when an agent uses memory to make a decision. One can then restrict the measure to the event E where one expects the memory use will be higher than in $\Sigma \setminus E$ and obtain a possibly tighter lower bound on $\log \mathcal{C}(\pi)$.

Theorem 1 *Let* $k < t$ *and* $E \in \sigma(Z_{1:t-1}, X_t)$ *be an event in the sigma-field generated by the history up to time* $t-1$ *and the observation at time* t*. Denote by* P_E *the probability measure that restricts* P *to* E*. Then* $I_E(A_t; Z_{1:k}|X_t, Z_{k+1:t-1}) \leq \min_{g\in\mathcal{M}_{\infty,\pi}} \log |g(k, \mathcal{Z}, [H])| \leq \log C(\pi)$*, where* I_E *denotes the (conditional) mutual information with respect to* P_E*.*

Proof Since Y_t is a function of Y_k and $Z_{k+1:t}$ for any $k < t$, the generated sigma-fields satisfy $\sigma(Z_{1:t-1}) \supseteq \sigma(Y_k, Z_{k+1:t-1}) \supseteq \sigma(Y_{t-1})$. From $P(A_t|X_t, Y_{t-1}) = P(A_t|X_t, Z_{1:t-1})$, it follows that $P_E(A_t|X_t, Y_{t-1}) = P_E(A_t|X_t, Z_{1:t-1})$ and hence $P_E(A_t|X_t, Y_k, Z_{k+1:t-1}) = P_E(A_t|X_t, Z_{1:t-1})$. We can then equivalently write

$$\frac{P_E(A_t, X_t, Y_k, Z_{k+1:t-1}) P_E(X_t, Z_{k+1:t-1})}{P_E(X_t, Z_{k+1:t-1}, Y_k) P_E(X_t, Z_{k+1:t-1}, A_t)} = \frac{P_E(A_t, X_t, Z_{1:t-1}) P_E(X_t, Z_{k+1:t-1})}{P_E(X_t, Z_{1:t-1}) P_E(X_t, Z_{k+1:t-1}, A_t)}$$

which implies that $I_E(A_t; Z_{1:k}|X_t, Z_{k+1:t-1}) = I_E(A_t; Y_k|X_t, Z_{k+1:t-1})$. We then can bound the conditional mutual information using basic properties of entropies as

$$\begin{aligned} &I_E(A_t; Z_{1:k}|X_t, Z_{k+1:t-1}) = I_E(A_t; Y_k|X_t, Z_{k+1:t-1}) \\ &= H_E(Y_k|X_t, Z_{k+1:t-1}) - H_E(Y_k|A_t, X_t, Z_{k+1:t-1}) \leq H_E(Y_k|X_t, Z_{k+1:t-1}) \\ &\leq H_E(Y_k) \leq \log|Y_k(E)| \leq \log|Y_k(\Omega)|. \end{aligned}$$

■

6. Related Work

Papapetrou and Kugiumtzis (2016) perform statistical tests based on conditional mutual information to identify the order of Markov chains. This is similar to our method but we are only interested on parts of the stochastic process, namely the agent's actions. In the work of Tishby and Polani (2011) mutual information is used as part of an optimization objective for policies. Instead of just for maximum cumulative reward, they optimize for the best trade-off between information processing cost and cumulative reward.

Our definition of a *memory function* matches the one by Chatterjee et al. (2010). While we are concerned with the analysis of our method, they use memory functions for asymptotic theoretical analysis of memory required to solve POMDPs with parity objectives. In the abstract model of memory in Section 5, the memory state is essentially a sufficient statistic summarizing all information from the past relevant for any action in the future. Sufficient statistics for general stochastic processes are discussed by Shalizi and Crutchfield (2001) introducing the concept of ϵ-machines. Unlike ϵ-machines, we require memory to be updated recursively and we are only concerned with the predictive power regarding future actions.

7. Conclusion

In this paper, we have proposed an approach for analyzing memory use of an agent that interacts with an environment. We have provided both a theoretical foundation of our method and demonstrated its effectiveness in an analysis of state-of-the-art DQN policies playing Atari games. Our treatment of memory usage in agents opens up a wide range of directions for follow-up work. First, our method assumes discrete observation and action spaces. The key challenge in extending to continuous space is the need to efficiently compute mutual information of high-dimensional, continuous observations. A promising avenue is to explore approximations that have been developed in domains such as neural coding, such as variational information maximization (Agakov and Barber, 2004).

Another interesting question to explore is whether the estimate of memory use by a policy can be improved when the environment can be controlled actively. That is, the behavior of an agent can actively be explored by manipulating the observations and rewards the agent receives. The task of identifying the events in which the agents requires the maximum amount of memory by manipulating its observations could possibly be set up as a reinforcement learning task itself. Further, estimating the amount of memory necessary to solve a task could potentially be used as an empirical measure for difficulty of sequential decision making tasks. Many real-world tasks require high-level reasoning with longer-term memory. While current reinforcement learning algorithms still mostly fail to achieve reasonable performance on such tasks, often experts, e.g. humans, can be observed when solving the task. Analyzing their memory use could not only give an indication of how difficult a task is but also possibly inform the design of successful reinforcement learning architectures.

References

F. Agakov and D. Barber. Variational information maximization for neural coding. In *International Conference on Neural Information Processing*, pages 543–548. Springer, 2004.

K. Chatterjee, L. Doyen, and T. A. Henzinger. Qualitative Analysis of Partially-Observable Markov Decision Processes. *Mathematical Foundations of Computer Science*, 6281:258–269, 2010.

P. Grassberger. Entropy Estimates from Insufficient Samplings. *arXiv preprint physics/0307138*, (7):5, 2003.

J. Hausser and K. Strimmer. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 10 (June):1469–1484, 2009.

N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver. Memory-based control with recurrent neural networks. *arXiv*, pages 1–11, 2015.

A. Krishnamurthy, A. Agarwal, and J. Langford. Contextual-MDPs for PAC-Reinforcement Learning with Rich Observations. *arXiv*, pages 1–30, 2016.

X. Li, L. Li, J. Gao, X. He, J. Chen, L. Deng, and J. He. Recurrent Reinforcement Learning: A Hybrid Approach. *arXiv*, pages 1–11, 2015.

V. Mnih, K. Kavukcuoglu, D. Silver, A. a. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

I. Nemenman, F. Shafee, and W. Bialek. Entropy and inference, revisited. In *Advances in Neural Information Processing*, 2002.

S. Nowozin. Improved information gain estimates for decision tree induction. In *ICML*, 2012.

M. Papapetrou and D. Kugiumtzis. Markov chain order estimation with conditional mutual information. *Simulation Modelling Practice and Theory*, 61, 2016.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, volume 10. John Wiley & Sons, 1 edition, 1994. ISBN 0471619779.

C. R. Shalizi and J. P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104(3-4):817–879, 2001.

H. Simon. *The Sciences of The Artifical*. MIT Press, 3rd edition, 1996. ISBN 0262193744.

S. Singh, T. Jaakkola, and M. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the eleventh international conference on machine learning*, volume 31, page 37, 1994.

N. Tishby and D. Polani. Information Theory of Decisions and Actions. In *Perception-Action Cycle*, pages 601–636. 2011. ISBN 978-1-4419-1451-4.