2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A connectionist actor-critic algorithm for faster learning and biological plausibility

Leonard Johard[1] and Emanuele Ruffaldi[2]

***Abstract*— We propose a novel biologically plausible actor-critic algorithm using policy gradients in order to achieve practical, model-free reinforcement learning. It does not rely on backpropagation and is the first neural actor-critic relying only on locally available information. We show it has an advantage over pure policy gradients methods for motor learning performance in the polecart problem. We are also able to closely simulate the dopaminergic signaling patterns in rats when confronted with a two cue problem, showing that local, connectionist models can effectively model the functioning of the intrinsic reward system.**

I. Introduction

Many psychological and neurophysiological studies imply that motor learning is dictated by adaptation to an ever fluctuating internal reward signal [1], which is is often assumed to be encoded by the dopaminergic system. The same line of thought dictates that the different brain functions are formed in our conscious or unconscious attempt to maximize the received reward from the environment. This has led to the development of a formal machine learning framework known as reinforcement learning. It is the most general form of machine learning, in the sense that the other major machine learning tasks, like unsupervised and supervised learning, can in general be formulated into a equivalent reinforcement learning problem. The loss function to be minimized in these tasks is simply used to define the reward function, with smaller loss giving higher reward.

Reinforcement learning is also a very difficult problem. As we have no error gradient information, we will have to estimate it locally. The question of how the animal brain is able to accomplish this remains unanswered, but is the key to unlocking the motor learning potential it possesses and be able to design self-learning system able to perform effective movements in very complex environments.

The reinforcement learning problem can be described as follows: an agent takes actions based on what it can observe about the world in order to achieve an objective, which is encoded in a reward function. Formally, this is most often defined within the context of Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs). In these frameworks, we define a state s, observation o, action a, transition probabilities to a new state $t(s' \mid s, a)$ and a reward function $r(s, a)$.

In problems of the discounted reward type, we seek to find parameters θ for a probabilistic parameterized policy $\pi(a, s \mid \theta)$ that optimizes:

$$max_\theta E\left[\int_0^\infty r(t)\gamma^{-t}dt\right] \quad (1)$$

where γ is a discount factor used to penalize distant rewards and to limit the convolution in the continuous case.

Unlike supervised learning, which requires the desired output, reinforcement learning requires only a reward signal given at some unknown time after the action in order to effectively learn. This means that we do not have access to the gradient of our reward function, $\frac{\partial r}{\partial a}$, and that we will instead have to estimate the gradient implicitly. This requires a certain degree of exploration, which can be either a search for a global minima, as in dynamic programming-based algorithms with exploration, or a search for a local minima, an approach usually referred to as policy gradient methods.

The structure of this paper is as follows: in the next section we will overview the capacity and limitation of state of the art in reinforcement learning with the aim to provide a good background and motivation for our work. In section III, we will go into details of the learning algorithm. In section IV we will experimentally test the algorithm for reinforcement learning and compare the internal reinforcement signals against neurophysiological experiments.

II. State of the art in Reinforcement Learning

Reinforcement learning is one of the most complex machine learning tasks. Several different approaches have been made and they vary widely in their methodology and assumptions. However, most of these lay in either of two directions: temporal difference (TD)[2] and policy gradient (PG) methods. TD methods, which notably includes Q-learning [3] and SARSA[4], are closely related to dynamic programming and resulted in many early successes in discrete action spaces. They have also resulted in a comprehensive theoretical framework, which is often referenced to in neuroscientific and psychological contexts. Policy gradient methods, on the other hand, are more naturally adapted to continuous action spaces. They optimize their policy by directly estimating the policy gradient on the reward function through small stochastic variations in the action, and use this gradient estimate to perform gradient ascent. This second approach is more closely related to the backpropagation used in supervised learning.

Another related concept is actor-critic (AC) architectures, which involves the creation of an additional, intrinsic reward separate from the received, or extrinsic, reward. The AC

[1]Leonard Johard is with PERCRO, Scuola Superiore S.Anna, Pisa, Italy l.johard@sssup.it
[2]Emanuele Ruffaldi is with PERCRO, Scuola Superiore S.Anna, Pisa, Italy e.ruffaldi@sssup.it

methods were originally derived from the TD error concept, originating from the TD learning framework, but can be readily applied also to policy gradient methods.

A. Discrete TD learning

Early reinforcement learning solutions were developed for small, discrete state spaces and inspired by the Bellman equation's ability to find globally optimal solutions using a state-dependent future value estimate. This analogue to the Bellman equation was called temporal difference and introduced the TD error. The small state spaces allow for effective application of dynamic programming in estimating the temporal difference error and the small action spaces allow for an iteration over all possible actions, for a total of $S \times A$ values to be stored. These simple problem types made it possible to use tables, which have long dominated the field for this reason. The two most common TD-learning methods are SARSA and Q-learning. SARSA iteratively develops better policies and discounted reward estimates. Q-learning estimates the best policy while exploring with another, usually predefined, exploration policy.

In the case of a POMDP we would need to replace s with the distribution of states for an exact solution.

B. TD learning in continuous spaces

When tackling MDPs in larger and continuous state and actions spaces, these spaces are often discretized. This can be effective up to medium-sized problems, but for other tasks the complexities quickly get out of hand. Problem size in reinforcement learning long limited the applicability of reinforcement learning in practice. A common idea for solving continuous or large problems has been to use function approximators to estimate the Q-value. In pure Q-learning, we still need to take the maximum Q over all possible actions at each step, which is computationally prohibitive in large action spaces. For SARSA, we can develop an on-policy reward estimate Q, but the problem of actually improving a policy using Q remains unsolved. Thus, once we leave the assumption of small, discrete action spaces the problem of finding a policy that increases $E[Q(s, a)]$ might be no easier than the original RL problem we started with.

C. Policy gradients

A more direct approach for solving reinforcement learning problems in continuous action spaces is known as policy gradients. The REINFORCE algorithm [5] introduced a way to calculate the policy gradient of the reward function using small, stochastic variations in the policy. A small step can be taken in the policy parameter space to ascend the reward function using ths gradient, which is calculated using the correlation between the stochastic variations in action and the reward.

The convergence of policy gradient algorithms is generally slow, but it will converge to a local optimum. A local maximum is indeed the best we can hope for in practical large-scale problems, as finding the global solution of a POMDP is at least PSPACE-complete [6].

An important parameter for the convergence rate is the baseline $b(s)$. Although REINFORCE will converge for any choice of baseline, the rate at which it does so varies substantially. An analysis of the signal-to-noise ratio of the estimate suggest that using the expected future reward as a baseline produces the optimal ratio [7] [8].

D. Actor-critic methods

Actor-critic algorithms form a generalized framework clearly separating the parametrized policy, knows as the actor within the actor-critic framework, from the parametrized discounted reward estimate, known as the critic. Although the actor-critic architecture suggests more complex combinations with value estimation, any reinforcement learning algorithm can be deconstructed into actor-critic pairs for easy comparison

An open question is how to most effectively parametrize the policy and reward estimate. Many different types of actors and critics have been studied, from moving averages and tables to backpropagation networks and spiking neurons.

For a brief overview of current work in actor-critic algorithms we suggest the review by Grondman et al. [8].

An overview of the actor-critic architecture can be seen in figure 1.

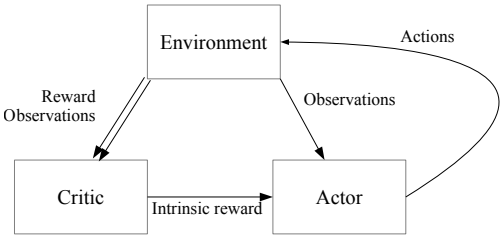


Fig. 1: The actor-critic architecture. The critic receives the extrinsic reward and calculates and produces a separate intrinsic reward, from which the actor updates its parameters.

E. Localized neural models

Neural network models in machine learning have predominantly been relying on backpropagation for over two decades for many good reasons. The existence of a gradient is an intuitive feature with nice convergence properties. The output error can propagate through any depth in order to optimize the network as a whole. Consequently, one would expect to find a similar mechanism in the main source of inspiration to all neural networks: the animal neuron. However, we have yet to identify a biologically plausible such mechanism. This has in fact been one of the most difficult questions for computational neuroscience to answer: how does the brain learn without backpropagation signals? Most realistic models leave the more difficult parts of learning open and focus

on solely unsupervised mechanism, such as spike-timing dependent plasticity, long term potentiation and long-term depression. It is clear that these learning forms are unlikely to produce the full spectrum of complex and goal-oriented animal behaviour that we are familiar with.

Lately, there has been an increase in the amount of work in biological reinforcement learning in particular. Notably Legenstein et al. [9] proposed a neural model using policy gradients to learn using reinforcement signals as a model for human plasticity. They achieved interesting results in simulating several different types of learning using simple neurons with REINFORCE.

Although policy gradients can learn any POMDP, policy gradient updates are likely too slow to organize the whole learning process. Actor-critic system are more practical candidates, but most parameterized critics need to propagate error gradients and this puts the biological plausibility into question.

F. Claims of importance of our work

We introduce eligibility traces for supervised learning and a biologically plausible algorithm for efficient supervised learning. These concepts are able to replicate in-vivo measurements of dopaminergic neurons estimating discounted future reward. We rethink the actor-critic framework by removing the Bellman updates and argue that significant benefits can be derived simply from combining supervised learning with reinforcement learning at different time scales. We strengthen our argument by showing benefits in convergence time and stability in the pole cart problem. This is the first neural actor-critic algorithm using only biologically realistic, local information for both the actor and the critic.

III. The CAC Algorithm

We propose an effective connectionist actor-critic (CAC) algorithm that learns using policy gradients and exclusively local information for solving POMDP problems. This is achieved with a policy gradient neural network and a cascade-correlation type network interlinked in an actor-critic structure. We present each part in the subsections below.

A. Traces for supervised learning

For supervised learning we can estimate the gradient on the discounted reward by keeping two traces, e_1 and e_2:

$$e_1(t+1) = \gamma e_1(t) + y(t)\frac{\partial}{\partial\theta}y(t) \tag{2}$$

$$e_2(t+1) = \gamma e_2(t) + \frac{\partial}{\partial\theta}y(t) \tag{3}$$

where y is the output and θ any given parameter of our estimating model. The resulting weight change with step size ϵ is calculated as:

$$\Delta\theta = \epsilon\left(e_1(t) - re_2(t)\right) \tag{4}$$

This can easily be seen as we try to minimize the error e:

$$e = \sum_i \sum_{\Delta t=0}^{\inf} \left[\left(y(i) - r(t)\right)^2 \gamma^{-\Delta t}\right] \tag{5}$$

where we use the notation $t = i + \Delta t$. We calculate the derivative of the error function:

$$\frac{\partial}{\partial\theta}e = 2\sum_i \sum_{\Delta t=0}^{\inf} (y(i) - r(t))\frac{\partial}{\partial\theta}(y(i) - r(t))\gamma^{-\Delta t} \tag{6}$$

which can finally be rewritten as

$$2\left(\sum_i \sum_{\Delta t=0}^{\inf} \frac{\partial y(i)}{\partial\theta} y(i)\gamma^{-\Delta t}\right) - \left(\sum_i \sum_{\Delta t=0}^{\inf} \frac{\partial y(i)}{\partial\theta} r(t)\gamma^{-\Delta t}\right) \tag{7}$$

The latter is a sum of two terms, which can be calculated separately using eligibility traces and summed using eq. 4.

B. Neural cascades

The cascade-correlation algorithm has been around for many years, but it has seen little practical use despite its biological plausibility and fast convergence [10]. This is mostly due to the success and fame of backpropagation networks and their theoretically attractive promise of exact calculation of the gradient through any network structure. In practice, this propagated error signal is of dubious utility. Deep architectures tend to worsen performance unless either pre-trained in an unsupervised fashion or trained for very long times. In a realistic neural motor learning system, we would also value fast learning over effective use of the network structure.

Backpropagation through time obviously suffers from related problems. In theory, it allows us to eventually propagate information from any moment in the past, but suffers from being attracted to poor local minima and vanishing gradients that prevent practical retrieval of information from just a couple of time steps back.

In contrast to these methods, cascade-correlation is a simple, naturally connectionist algorithm. Since backpropagation in practice is unable to retain information for more than a couple of time steps [11], the unsupevised method would be roughly equivalent.

We combine traces with several modifications to the cascade-correlation algorithm. The changes described below enhance both the plausibility and practicality, while maintaining or improving the generalization ability, of the original model.

First, we use only sibling units in order to limit network depth, although any combination of inputs to the neurons could be used as an alternative using the same learning rules. A detail overview of the difference between sibling units and the descendant units of the original cascade correlation algorithm can be found in Baluja et al. [12]. Further, we have a preset size of the network and do not incrementally add neurons. Overfitting can instead be more elegantly prevented through ridge regression. We also do not freeze neurons, but let the whole neuron chain adapt continuously. This has been shown to result in roughly equivalent overall training time

and brings advantages in minimizing the error on the training set[13].

All experimental results in this paper make use of this new algorithm, which we call the neural cascade algorithm.

1) Proposed biological mechanism: These traces can be calculated locally. Cascade-correlation requires two operations in a biological model: regression of the error signal and updating of the residual error. This could be implemented by two neuron types: Neuron type 1 updates the residual error by estimating the reward through linear regression using inputs from surrounding neurons. Neuron type 2 is nonlinear and uses observations from input neurons to estimate the difference between the best local reward approximation from neuron type 1 and the actual reward. We use eligibility traces, but do not rely on Bellman-like updating of the error estimation against its own estimate, which means we are in agreement with the experimental results in Pan et al. [14]. Note that relying on the eligibility traces alone allows us to learn an unbiased discounted reward estimate from each observation in the POMDP setting without maintaining an explicit belief state.

C. Reinforcement learning neuron

The role of our actor is to learn exact solutions on small time scales that can be assumed to be sufficiently short to guarantee enough samples for an accurate estimation of the policy gradient. Our actor is a neural policy gradient model that is similar to the model proposed by Legenstein et al. [9], but it implements eligibility traces and uses the noise to directly estimate the gradient. Using an eligibility trace allows us to learn with delays in the feedback, i.e. when both the action itself and the reward of an action might not be visible to the critic until a few time steps later.

1) Algorithm: We use a standard artificial neuron with a tangent sigmoid activation function. Exponentially decaying eligibility traces have been widely used in the policy gradient context and is equivalent to ascent on the discounted future reward for REINFORCE [15]. We get the following weight changes:

$$\theta_{elig}(t+1) = \gamma e + \sigma \frac{\partial}{\partial \theta} y(t) \tag{8}$$

$$\Delta\theta(t) = \epsilon \left[e(t) + \gamma r_{internal}(t) \right] \tag{9}$$

where $e(t)$ is the eligibility trace.

The policy gradient be estimated in any output neuron and backpropagated through hidden neurons. Alternatively, it can be re-estimated locally in the hidden neurons [5]. We choose the latter approach, as it is more realistic in avoiding both backpropagation through time and the need to identify output neurons. The expectation of this gradient is identical to the backpropagated signal, as the hidden neurons essentially performs hierarchical reinforcement learning with local estimation of the gradient. We will be able to estimate the gradient effectively even in recurrent networks without using backpropagation.

D. Actor-critic structure

In temporal difference learning we replace the extrinsic reward signal r_{ext} given to the actor with the following intrinsic reward $r(t)$:

$$r(t) = r_{ext}(t) + \gamma r_{est}(s(t)) - r_{est}(s(t-1)) \tag{10}$$

where r is the extrinsic reward and r_{est} the state-dependent discounted future reward estimate from the critic. In an POMDP, this estimate is simply calculated as above, but with the observation as an intermediate step between state and reward estimate.

The TD-error term (eq. 10) does not bias the actor if the critic and actor use identical eligibility traces, as noted by Kimura and Kabayashi [15]. However, we would indeed like to bias the actor towards taking actions that provide a higher average reward in longer time scales than those its short eligibility trace takes into consideration. By changing the critics γ value the, the actor will instead start ascend these long-term reward gradients.

The role of our critic is to move as much as possible of the learning task into faster supervised learning tasks. This is especially important for longer reward delays, as time scales involved makes each sample much more precious. Reinforcement learning will perform the learning in the shorter term, covering primarily the delay from action to observable change in state. Our algorithm is valid even if the state is not observable on the short term, while it works under an MDP approximation in the longer time scales. This relaxation of the problem formulation greatly reduces the credit assignment problem and allows us to perform effective motor learning practically impossible with policy gradients alone. The drawback is that direct action-reward pairs over long time scales cannot be learnt if the actions do not result in an observable change within the actor's shorter learning scale. This is a very reasonable limitation in the motor learning context and is also not a very strict limitation in general, given the amble possibilities to extend the observation space through recurrent connections and long-term memory.

Note that the argument presented here is superficially similar to but fundamentally different from the variance explanation in Konda et al.[16]. We are not dealing with MDPs, but with POMDPs that can be approximated by MDPs over larger time scales and this can be exploited by limiting strict POMDP assumptions to the time scales when they are needed and by adding a critic to the larger time scales. In a diametrically opposite approach, we also assume the actor to be the faster of the two updates.

E. Biological motivation

Our critic model supports both the reward modulation hypothesis of dopamine and several superficially contradictory results mentioned by Berridge [17]. Assuming dopaminergic neurons are only transmitters of the intrinsic reward, lack of dopamine would prevent the actor from pursuing long term rewards while still learning immediate rewards through direct

reinforcement learning. As a consequence, learning of long-term predictions still take place in the brain in the absence of dopamine, but information about the benefits of future rewards do not reach the actor neurons.

IV. Experimental evaluation

We provide two experments to validate our actor-critic model from different perspectives. We would like to validate both the practical value over simple policy gradients and the agreement of the TD-error with neurophysiological results on dopaminergic neurons in animals. Both experiments were performed with an implementation of our CAC algorithm written in C++ and made available on `https://github.com/eruffaldi/neuralcascade/`.

A. Pole cart balancing

Our first experiment seeks to evaluate the learning rate of the actor-critic algorithm compared to the regular policy gradient updates. Supervised learning tries to estimate the observation-reward mapping given the policy and relies on the actor for sampling the best policy assuming the critics estimation is accurate. Supervised learning can estimate this correlation mapping quickly given a specific trajectory, as it can calculate the derivative directly instead of approximating it through correlations with a noisy output.

On the other hand, using a critic suffers from having a two-step process first estimate the value function of the policy and later perform gradient ascent instead of just performing direct ascent on the reward estimate. Our hypothesis is therefore that supervised learning with large traces in relation to the actors traces, where the learning time of the actor is small or negligible, will lead to faster learning measured in the number of trials for an actor-critic structure. If they instead learn on similar time scales, we hypothesize that this will lead to slower learning due to the two different learning phases needed. This mechanism counteracts the advantages of the MDP relaxation if the difference in time scales is not large enough.

We will test this on the classic pole cart balancing problem. We will repeat the experiment with different time steps, where we define step as a simulation step in time, while an epoch is the period from the start of the cart until the pole balancing fails as the pole falls below our threshold angle. According to our hypothesis,the actor will have more samples per epoch to use for adaptation, an easier credit assignment problem as we decrease the time step. On the other hand, the complexity of the control problem as such can be reasonably assumed to be largely independent of the time resolution used, given that the time step is small enough to allow efficient control. We therefore hypothesize that a policy gradient algorithm will remain largely unaffected as step size is decreased, while the actor-critic method described will improve as we move a greater part of the problem into its MDP-relaxed equivalent.

1) Experimental setup: In our pole cart problem, the pole is randomly initiated for each epoch at an angle with an even distribution in the interval [-1 1] degrees and the cart in the interval [-0.8 0.8] meters. Initial velocity for the cart and pole is intiated in the interval [-0.01 0.01] m/s. Inputs are the pole position and velocity, while output is the acceleration of the cart. A small amount of noise is added to the inputs to ensure POMDP conditions on the short scale.

The epoch is a failure if the pole angle exceeds 36 degrees or if the cart moves more than 2.4 meters from its original position. An epoch is a success if it manages to balance the pole for more than 10 seconds. A trail is a series of epochs training the same CAC network. A trial is considered successful when algorithm succeeds in three consecutive epochs. If more than 40.000 epochs passed without convergence, the trial is assumed to be stuck in a local minimum and rejected.

We keep the critic's trace constant in time through all experiments, while we optimize the actor's trace in early testing. In practice this results in traces that are close to constant in time steps across the experiments.

B. Reinforcement signal

In this experiment we test the critic's response to training with cues and compare with the firing rates of dopaminergic neurons as reported by Pan et al.[14]. Similarly to their setup,we present two cues followed by a reward with probability 0.6. We give probability 0.2 for each of two alternative scenarios: 1) omitting cue 2 but still giving reward and 2) presenting both cues but omitting the reward. We compare patterns early and late in the training. Early training and late training are the same 100 and 400 trials, respectively, that were used in their simulations.

The respective cues are given at 5 and 15 steps before the reward, which is given after step 29. This roughly equates a step with 1 second. We used a longer 100 step separation between trials, compared to a pseudo-random 10 - 20 s in Pan et al., in order to avoid contamination and produce clearer results. Inputs to critic is a vector of length 4. We set variables 1 and 2 to 1 when cue 1 and 2 is given, respectively. Variable 3 is set to 1 immediately preceding and and variable 4 is set to 1 immediately after the delivery of reward. Each input is then exposed to exponential decay of 0.02 per time step.

We initialize our critic with random weights close to 0, in order to remove the impact of the initialization noise in early training results.

C. Results

The results in the pole cart balancing test can be seen in figure 2. The CAC algorithm requires more epochs to learn in trials with large time steps, which is a reasonable result given the overhead of using two learning phases instead of one. We can see rapid reduction in the number of epochs required as we decrease the time step , which agrees with our stated hypothesis. We also found an unexpected benefit in stability; the CAC algorithm converged to good solutions in all experiments, while we were unable to find a parameters setting for the policy gradient that resulted in a convergence

probability greater than about 0.5. This is possibly due to the policy hitting a local minima or a plateau.

The imitation of the neural dopamine signal is found in figure 3. Our simulation is in good agreement with the results in rat neurons by Pan et al. [14]. A notable difference is that neural cascades have a stronger response to cue 1 than cue 2 in early training, while the $TD(\lambda)$ simulation of Pan et al. achieved the opposite result. Unfortunately, the population recordings in-vivo are too noisy to establish a clear relative size of the cues. Moreover, one should be careful in interpreting early cue responses as they are also dependent on the gradient descent strategy used, e.g. the use of adaptive learning rates and information from the Hessian. In contrast, late cue responses should be closer to a minima and largely independent of the gradient descent method used.

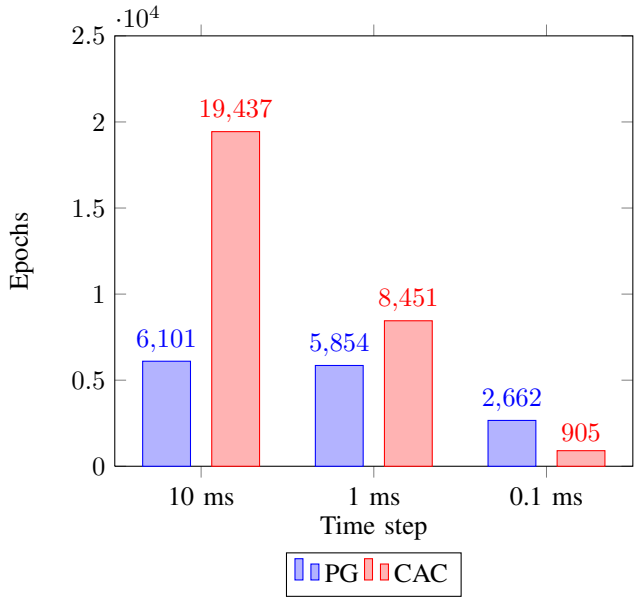
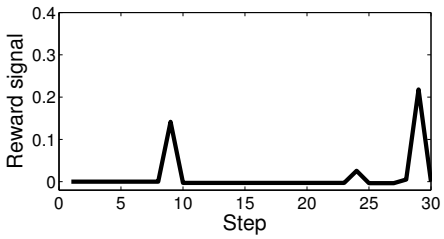
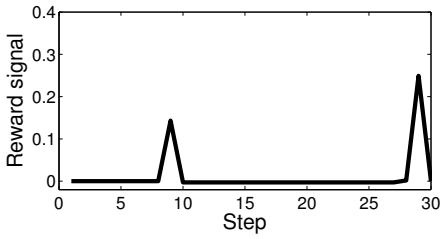


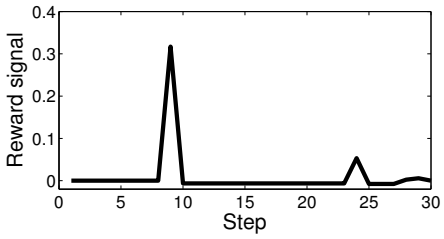
Fig. 2: Average iterations over 20 successful trials. Failure to reach the target 10 s in three consecutive epochs within 40.000 epochs led to rejection of the trial. The CAC algorithm succeeded in all trials, while the direct policy gradient failed in approximately half of the trials.



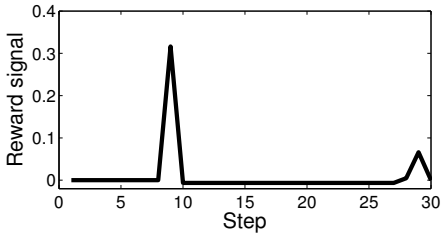
(a) Early training, second cue given



(b) Early training, second cue omitted



(c) Late training, second cue given



(d) Late training, second cue omitted

Fig. 3: Simulated conditioned responses to cues using our critic network. Top two images is early training (100 trials) with and without the seconds cue respectively. Bottom two images is late training (400 trials). For details of this particular cue problem see Pan et al.

V. Conclusions

Our results prove that actor-critics can greatly reduce overall policy gradient learning time. Using the MDP assumptions seems to be the key in improving the learning rate, while preserving the short-term POMDP assumptions allows us to use outputs whose effects are not immediately observable. This actor-critic relaxation of the POMDP assumption could be the key to reach efficient reinforcement learning for motor control tasks.

It is also likely that higher cognitive function can result from similar learning mechanisms, as all hidden neurons in the network estimate their reward gradient locally and will perform hierarchical reinforcement learning. Thus, in solving the motor learning problem we might also be able to automatically tackle a range of increasingly abstract learning tasks using the very same building blocks.

We can also conclude that it is the use of traces that is is the essential factor in reaching agreement with extracellular recordings, not the dynamic programming nature of TD-learning. We have showed that the results of Pan et al. [14] can be replicated in simulations using such connectionist networks. We note that the local learning rules proposed in this paper presents a plausible explanation of the role dopamine in its mechanism, its purpose and is in agreement with neurophysiological measurements of the resulting signal.

In summary, the CAC algorithm provides an effective working hypothesis on mechanism of animal motor learning

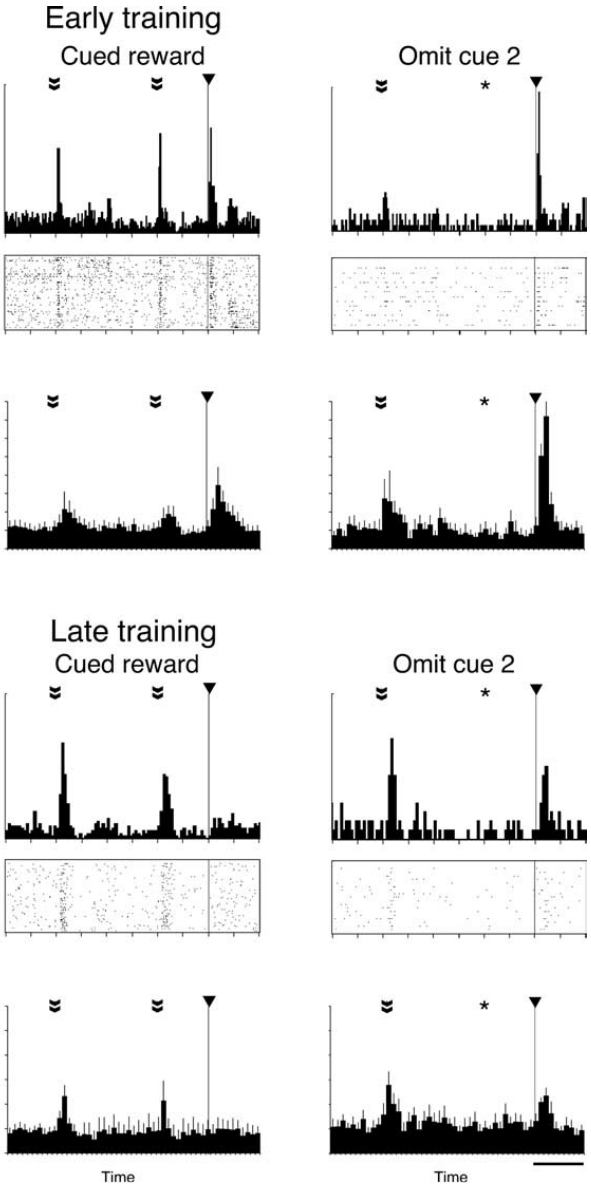


Fig. 4: Single neuron recordings in rats of the effect of training on responses of dopamine neurons to reward delivery in a two cue problem, as originally presented by Pan et al. Top rows of each training stage is single neuron recordings, while bottom row is population recordings. [14]

.

and higher mental functioning, as well as a practical and effective framework for reinforcement learning in motor control tasks.

VI. Future directions

Many improvement are already planned on these local learning algorithms. We estimate that further development the critic will likely have the most profound effect on increases of the learning rate. Stacked critics with different traces could allow us to learn more effectively at several different time scales at once. We will also attempt to utilize unsupervised pre-training with eligibility traces, possibly simultaneously with supervised and reinforcement training.

In terms of biological realism, we are working on equivalent spiking models, but expect them to be too computationally expensive for practical evaluation in realistic robot tasks. Instead, we hope that experience from practical reinforcement learning will continue give clues to and take inspiration from biological neurons indirectly.

We are also looking at implementing well-known rehearsal methods in order to stabilize learnt experience and simulate dreaming, which could be a requirement for avoiding catastrophic forgetting in more complex learning tasks.

References

[1] Y. Niv, "Reinforcement learning in the brain," *J. Math. Psychology*, vol. 53, no. 3, pp. 139 – 154, 2009.

[2] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[3] C. Watkins, "Learning from Delayed Rewards," Ph.D. dissertation, University of Cambridge, England, 1989.

[4] G. A. Rummery and M. Niranjan, "On-line q-learning using connectionist systems," Univ. Cambridge, Depart. Engineering, Tech. Rep., 1994.

[5] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learning*, vol. 8, pp. 229–256, 1992.

[6] C. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, Aug. 1987.

[7] J. W. Roberts and R. Tedrake, "Signal-to-noise ratio analysis of policy gradient algorithms," in *Advances of Neural Inform. Process. Syst. (NIPS)*, 2008, pp. 1361–1368.

[8] I. Grondman, L. Busoniu, G. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C*, vol. 42, no. 6, pp. 1291 – 1307, Nov. 2012.

[9] R. Legenstein, S. Chase, A. Schwartz, and W. Maass, "Functional network reorganization in motor cortex can be explained by reward-modulated Hebbian learning," in *Advances in Neural Inform. Process. Syst. 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 1105–1113.

[10] G. Drago and S. Ridella, "Convergence properties of cascade correlation in function approximation," *Neural Computing & Applicat.*, vol. 2, no. 3, pp. 142–147, 1994.

[11] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber, "Solving deep memory pomdps with recurrent policy gradients," in *Artificial Neural Networks–ICANN 2007*. Springer, 2007, pp. 697–706.

[12] S. Baluja and S. E. Fahlman, "Reducing network depth in the cascade-correlation learning architecture," DTIC Document, Tech. Rep., 1994.

[13] C. S. Squires, Jr., and J. W. Shavlik, "Experimental analysis of aspects of the cascade-correlation learning architecture," Comput. Sci. Dept., Univ. Wisconsin,Madison," Working Paper, 1991.

[14] W.-X. Pan, R. Schmidt, J. R. Wickens, and B. I. Hyland, "Dopamine Cells Respond to Predicted Events during Classical Conditioning: Evidence for Eligibility Traces in the Reward-Learning Network," *J. Neuroscience*, vol. 25, no. 26, pp. 6235–6242, 2005.

[15] H. Kimura, K. Miyazaki, and S. Kobayashi, "An analysis of actor/critic algorithms using eligibility traces: Reinforcement learning with imperfect value function," in *Proceedings of the 15th International Conference on Machine Learning*, 1998, pp. 278–286.

[16] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms." in *NIPS*. Citeseer, 1999, pp. 1008–1014.

[17] K. C. Berridge, "The debate over dopamine's role in reward: the case for incentive salience." *Psychopharmacology*, vol. 191, no. 3, pp. 391–431, Apr. 2007.