Attention-Gated Reinforcement Learning in Neural Networks—A Unified View

Tobias Brosch, Friedhelm Schwenker, and Heiko Neumann

Institute of Neural Information Processing, University of Ulm,
89069 Ulm, Germany

{tobias.brosch,friedhelm.schwenker,heiko.neumann}@uni-ulm.de

Abstract. Learning in the brain is associated with changes of connection strengths between neurons. Here, we consider neural networks with output units for each possible action. Training is performed by giving rewards for correct actions. A major problem in effective learning is to assign credit to units playing a decisive role in the stimulus-response mapping. Previous work suggested an attentional feedback signal in combination with a global reinforcement signal to determine plasticity at units in earlier processing levels. However, it could not learn from delayed rewards (e.g., a robot could escape from fire but not walk through it to rescue a person). Based on the AGREL framework, we developed a new attention-gated learning scheme that makes use of delayed rewards. Finally, we show a close relation to standard error backpropagation.

Keywords: Reinforcement Learning, Backpropagation, AGREL, Delayed Rewards, Attention, Hebbian Learning.

1 Introduction

When investigating the plasticity of connection weights in a neural network in tasks involving a reward signal, reinforcement learning was shown to be an effective tool. Depending on the underlying methodology its biological plausibility varies [16,14]. The proposed mechanisms range from detailed biological models resembling functionalities of the basal ganglia [14,7,1,5] over models using Hebbian plasticity in neural networks [9,10,11] to machine learning models using error backpropagation (BP) in neural networks [15,13,2,3]. While the latter are very effective, they lack biological plausibility. Here, we focus on the model of [10] utilizing a biologically plausible global learning signal with Hebbian plasticity of connection weights that is gated by an attentional feedback signal to increase plasticity of task relevant-neurons. This attention-gated reinforcement learning (AGREL) [10] is, however, limited to tasks with immediate reward delivery. We propose a novel learning scheme extending AGREL to also incorporate delayed rewards (e.g., a robot walking through fire to rescue a person). Similarly to [11], it also extends AGREL by learning an action-value function in the output layer. We demonstrate that the new model also improves learning in the reduced case of immediate rewards as studied in [10], and show a close relation to BP for learning the connection weights in the employed network.

2 Task and Network Design

We employ a three-layer neural network shown in Fig. 1 to simulate the selection of one of C mutually exclusive actions depending on the presented state s_t (c.f. [10]). The state is represented in the input layer with activities $\{X_i\}_{i=1\dots N}$. Connections weighted by v_{ij} propagate activity to a hidden layer with activities $\{Y_j\}_{j=1\dots M}$. Post-synaptic activity depends on a nonlinear transfer function g (we employ a logistic function like in [10]). Weights w_{ij} propagate activity from

Fig. 1. Three-layer neural network. An input pattern represented by N activities X_i is propagated via weights v_{ij} to M hidden layer neurons of activities Y_j (solid arrows). Weights w_{jk} propagate activity to the output layer with activities Z_k in which neurons engage in a competition (circular arrow heads). Activity of the winning unit a is propagated back to the hidden-layer by feedback weights w'_{aj} (dashed arrows). Learning is initiated by a global learning signal δ .

the hidden to the output layer. An essential part of AGREL is that neurons in the output layer engage in a competition such that only one of C output neurons gets activity $Z_a = 1$, where the probability depends on the presynaptic activity:

$$Y_j = g(Y_j^{pre}), \quad \text{with} \quad Y_j^{pre} = \sum_{i=0}^N v_{ij} X_i, \quad g(x) = \frac{1}{1 + \exp(-x)}, \quad (1)$$

$$\mathbb{P}(Z_k = 1) = \frac{\exp(Z_k^{pre}/\tau)}{\sum_{k'=1}^C \exp(Z_{k'}^{pre}/\tau)}, \quad \text{with} \quad Z_k^{pre} = \sum_{j=0}^M w_{jk} Y_j. \quad (2)$$

Activity of the winning neuron can be propagated backwards by feedback weights w'_{aj} to gate synaptic plasticity of task relevant neurons (attention gating) in earlier layers. In order to incorporate delayed rewards (in contrast to [10]), we utilize the presynaptic activity Z_k^{pre} of the neural network output layer as function approximator for an action-value function $Q_\pi(s, a) = Z_a^{pre}(s)$. This maps a state s and action a to the expected discounted return

$$\mathbb{E}_\pi\{R_t | s_t = s, a_t = a\}, \quad R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (3)$$

where π is the current policy of the agent (with $\pi(s, a)$ denoting the probability of selecting action a in state s), r_t is the reward the agent receives at time t and $0 \leq \gamma \leq 1$ is a discount factor (for γ close to 1, the agent becomes more farsighted). The goal is to find an optimal policy π^* that optimizes the expected discounted return. The corresponding optimal action-value function is denoted $Q^*(s, a)$. To deal with delayed rewards, we will employ a Sarsa-style learning signal from temporal difference (TD) methods for control problems. It encodes the difference between the improved discounted return expectation (based on the experienced reward) and the currently expected discounted return (see e.g. [13] for more details). Because of the combination of a Sarsa-style learning signal and AGREL, we will name our novel extended learning scheme SAGREL.

2.1 Learning in SAGREL

Neural plasticity in SAGREL—as well as in AGREL—depends on two signals that jointly determine plasticity: First, the global reinforcement error signal δ and second, an attentional feedback signal originating from the winning neuron in the output layer and delivering a feedback signal along the connection of the reverse hierarchy network to guide attentional resources. To incorporate learning of delayed rewards, we adapt the definition of the global error signal δ . The learning signal δ in AGREL is only defined for immediate rewards as (c.f. [10])

$$\delta_{AGREL} = \begin{cases} 1 - \mathbb{P}(Z_a = 1), & r = 1, \quad \text{in successful trials,} \\ -1, & r = 0, \quad \text{otherwise.} \end{cases} \quad (4)$$

In SAGREL we adopt a TD Sarsa-style learning signal (similarly to [11]) that directly depends on the presynaptic activity of the output layer $Z_a^{pre}(s) = Q(s, a)$

$$\delta_{SAGREL} = \left[r_{t+1} + \gamma Z_{a_{t+1}}^{pre}(s_{t+1}) \right] - Z_{a_t}^{pre}(s_t), \quad (5)$$

where $r_{t+1} \in \mathbb{R}$ is the reward the agent receives after state s_t taking action a_t , experiencing state s_{t+1} and choosing action a_{t+1} to perform in the next step (see e.g. [13] for more details on Sarsa). Learning of delayed rewards is achieved by incorporating the minuend $r_{t+1} + \gamma Z_{a_{t+1}}^{pre}(s_{t+1})$ which provides a more accurate measure of the expected discounted return (because it incorporates the experienced reward r_{t+1}) than the current estimation $Z_{a_t}^{pre}(s_t)$. We adopt the feedback signal and the neural plasticity mechanism of AGREL as described in [10] which we briefly summarize in the following.

Weights are changed according to a variant of an Hebbian rule, where changes depend upon the product of pre- and postsynaptic activity [8,6]. Weight changes between the hidden and output layer are described by

$$\Delta w_{jk} = \beta \cdot f(\delta) \cdot Y_j \cdot Z_k, \quad (6)$$

where β is the learning rate and f is an expansive function increasing the effective learning rate for unexpected rewards. In AGREL f is defined as $f(\delta) = \delta/(1 - \delta)$

for $\delta \geq 0$ and $f(\delta) = -1$ otherwise. In SAGREL f needs to be adjusted to the range of the rewards of a given task but can also be generically set to $f(\delta) = \delta$, the identity function. Note that $Z_a = 1$ and $Z_k = 0$ for $k \neq a$, such that only connections w_{ja} to the winning neuron are adjusted. Weight changes between the input and the hidden layer are also governed by Hebbian plasticity but additionally gated by the factor fb_{Y_j} representing feedback of the winning unit:

$$\Delta v_{ij} = \beta \cdot f(\delta) \cdot fb_{Y_j} \cdot X_i \cdot Y_j, \quad fb_{Y_j} = (1 - Y_j) \cdot \sum_{k=1}^C Z_k \cdot w'_{kj}, \quad (7)$$

where the factor $(1 - Y_j)$ reduces the effect of feedback on the plasticity of the connection weights of highly active units. After the competition $Z_a = 1$ and $Z_k = 0$ for $k \neq a$ and the equation reduces to

$$\Delta v_{ij} = \beta \cdot X_i \cdot Y_j \cdot w'_{aj} \cdot f(\delta) \cdot (1 - Y_j). \quad (8)$$

Cortical anatomy and neurophysiology suggests that feedforward and feedback connections are reciprocal [4,12]. Thus, we set $w_{jk} = w'_{kj}$ (see [10], their sect. 5.4 for a study about deviations from exact reciprocity).

2.2 Comparison between AGREL and SAGREL

In the reduced case of single step tasks and immediate reward delivery (as studied in [10]) δ_{SAGREL} reduces to $r - Z_a^{pre}$ because the expected reward for the next state $Z_{a_{t+1}}^{pre} = 0$ (episode ends after the first state). In rewarded trials ($r = 1$) this reduces to $\delta_{SAGREL} = 1 - Z_a^{pre}$ which is qualitatively the same as $\delta_{AGREL} = 1 - \mathbb{P}(Z_a = 1)$ because $\mathbb{P}(Z_a = 1)$ is a monotonically increasing function of Z_a^{pre} . In unrewarded trials ($r = 0$), $\delta_{AGREL} = -1$ causes weights to potentially diverge to $-\infty$ because exploration keeps the agent encountering each state in an infinite amount of trials. In SAGREL $\delta_{SAGREL} = -Z_a^{pre}$ which for $Z_a^{pre} \geq 0$ is qualitatively identical to AGREL but prevents weights from approaching $-\infty$ because for $Z_a^{pre} < 0$, δ_{SAGREL} becomes positive. To summarize, the novel SAGREL can deal with delayed rewards tasks while maintaining the computational complexity and allowing an unlimited range of rewards, i.e. $r \in \mathbb{R}$.

2.3 Comparison to Standard Backpropagation in Function Approximation Using Layered Neural Networks

We will now compare the attention-gated reinforcement mechanisms (see previous sections) against standard BP. We utilize the same network as sketched in Fig. 1 but in standard BP, the competition is not modelled in the final layer but carried out separately. Similarly to SAGREL, we interpret the presynaptic activity Z_k^{pre} as function approximation of an action-value function $Q(s, a) = Z_a^{pre}$. For now we assume knowledge of the optimal action-value function Q^* . A commonly used L^2 error function is (c.f. [13])

$$E = \frac{1}{2} \delta^2 = \frac{1}{2} (Q^*(s_t, a_t) - Q^{\pi_t}(s_t, a_t))^2 = \frac{1}{2} (Q^*(s_t, a_t) - Z_{a_t}^{pre}(s_t))^2. \quad (9)$$

The derivative with respect to a parameter vector θ (here $\theta = (\mathbf{v}, \mathbf{w})$ the concatenation of the weights v_{ij} and w_{jk}) is then given by

$$\nabla_{\theta} E = (Q^*(s_t, a_t) - Z_{a_t}^{pre}(s_t)) \nabla_{\theta} Z_{a_t}^{pre}(s_t) = \delta \cdot \nabla_{\theta} Z_{a_t}^{pre}(s_t). \quad (10)$$

Approximating Q^* by the one-step Sarsa-style return, $r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) = r_{t+1} + \gamma Z_{a_{t+1}}^{pre}(s_{t+1})$, results in the following gradient descent update in the three-layer network in Fig. 1:

$$\Delta w_{jk} = -\beta \cdot \frac{\partial E}{\partial w_{jk}} = \begin{cases} \beta \cdot \delta \cdot Y_j(s), & k = a \\ 0, & k \neq a \end{cases} \quad (11)$$

$$\Delta v_{ij} = -\beta \cdot \frac{\partial E}{\partial v_{ij}} = \beta \cdot \delta \cdot w_{ja} \cdot g'(Y_j^{pre}(s)) \cdot X_i(s), \quad (12)$$

where $\delta = r_{t+1} + \gamma Z_{a_{t+1}}^{pre}(s_{t+1}) - Z_{a_t}^{pre}(s_t)$ and g' denotes the derivative of the transfer function g . We can now compare the weight updates of SAGREL and BP. For weight updates between hidden and output layer, Δw_{jk} , we compare eqns. (11) and (6) which are identical if and only if $f(\delta) = \delta$, the identity function. For weight updates for connections between input and hidden layer, Δv_{ij} , we compare eqns. (12) and (8). When g is the logistic function, eqn. (1), $w_{ja} = w'_{aj}$ and $f(\delta) = \delta$, the identity function, we note that both equations are identical. This is a remarkable result with respect to two properties. *First*, the result implies that BP results in biologically plausible Hebbian plasticity in a paradigm where the teacher signal is replaced by a reward deployment mechanism. *Second*, attention-gated reinforcement learning employs the same efficient updates as the mathematically motivated gradient descent BP. This also explains the empirical observations of [10] that the average weight changes of AGREL are identical to those of BP for instantaneous learning situations. Additionally, this result also extends to [11] in the special case, when the tags (which mark previously active and task relevant neurons) immediately decay (i.e. their $\lambda\gamma = 1$).

3 Experimental Comparison of Models

First, we compare AGREL with SAGREL in two different experiments presented in [10]. Second, we demonstrate that SAGREL is capable to learn a delayed rewards task. Similarly to [10], we equipped the network with additional degrees of freedom by adding bias neurons with activities $X_0 = Y_0 = 1$, initialized connection weights with values drawn uniformly from the interval $[-0.25, 0.25]$ and used the same performance criterion. The criterion for a successful learning episode was that the probability of correct classification was at least 75% for each of the input patterns.

3.1 Immediate Reward Tasks

Two tasks were used to compare the performance levels to the results in [10].

Exclusive-Or (XOR) Task. The exclusive-or task presents binary input patterns to two input units, namely 01, 10, 11 and 00. The agent must activate one

of two output units with the first one being active for 01, 10 and the second one being active for 11, 00. The number of hidden neurons was 2 or 3.

Counting Task. In this task, the network must determine how many of N input neurons are switched on. Consequently, the number of output units is $N + 1$ (also the number of hidden units). In one iteration of a learning trial all of the $N + 1$ different stimulus classes are presented. The specific units that are switched on in each presentation are determined randomly (identical as in [10]).

3.2 Comparison of AGREL and SAGREL

Table 1 displays the minimal median number of iterations (one iteration equals the presentation of all input patterns) required to reach the criterion when optimizing with respect to learning rate β (data for AGREL from [10]) for both models. To compare our method to the results presented in [10], τ was set to one. The data show that SAGREL indeed outperforms AGREL in both tasks (though we used the simplest choice for the function f , the identity function). Based on the investigation outlined in sect. 2.2, this is expected since SAGREL utilizes a more distinct error signal. This demonstrates that our extended learning scheme not only incorporates learning of delayed rewards but also improves the performance in the reduced case studied in AGREL.

Table 1. Comparison of AGREL and SAGREL in the XOR and Counting task (smaller number of iterations is better). Values for AGREL are from [10]. Presented is the median number of iterations (from 500 separate trials) required to reach the 75% performance criterion optimized with respect to β . In SAGREL we determined the minimal number of iterations by performing a grid-search for values of β in $[0.3, 1]$ in steps of 0.025. The data show that the new learning scheme (SAGREL) constantly outperforms the original implementation (AGREL).

Task	AGREL	[10]	SAGREL (new)	
	Iterations	β	Iterations	β
XOR (2 hidden units)	535	0.35	422	0.475
XOR (3 hidden units)	474	0.45	297	0.6
Counting 2 inputs	157	0.4	30	0.875
Counting 3 inputs	494	0.25	148	0.325
Counting 4 inputs	1316	0.1	485	0.2

3.3 Delayed Rewards Task

To confirm that SAGREL can indeed learn tasks incorporating delayed rewards, we tested it in a simple task. In this task, N input and output units are used. It always starts with the first unit being active. In a correct trial, the agent selects the same output neuron a as the active input neuron and experiences the next state in which neuron $a + 1$ is active (only one input neuron is active at a time). The agent always receives a negative reward, $r = -0.1$, except for the last state, when neuron $a = N$ is active and the agent selects neuron a .

In this case the agent is rewarded with $r = 1$. Whenever a wrong action is selected, the task is reset to the beginning. An episode ends, when the agent gains a positive reward or exceeds a maximum of 30 trials in trying so. Learning was performed for $\tau = 2$ that was multiplied with $\tau_D \leq 1$ after each episode to make the policy a bit more greedy. The employed performance criterion was that the greedy policy with respect to the current action-value function represented by the network (i.e. $\tau = 0$) gained the maximum reward for 50 iterations in a row. To identify the parameters that gain a minimal number of iterations to reach the performance criterion (we measure the median number of iterations in 500 learning task repetitions), we performed a grid search for the parameters $\beta \in [0.6, 1]$ in steps of 0.025 and $\tau_D \in [0.92, 1]$ in steps of 0.005. For $N = 3$, the SAGREL algorithm needed a minimum median number of 190 iterations for $\beta = 0.7$ and $\tau_D = 0.94$ (note that AGREL is not able to solve this task because it cannot deal with delayed rewards). In comparison, a Q-table Sarsa implementation needed only 55 iterations for $\beta = 0.625$ and $\tau_D = 0.93$. This disadvantage can be accounted to the fact that the task is much easier for a Q-table implementation, because there are only N different input states (in contrast to an infinite number of possible input states in SAGREL).

To summarize, SAGREL can indeed learn tasks of delayed rewards and, as such, extends the functionality of the AGREL framework.

4 Discussion

We presented a novel reinforcement learning algorithm for a three-layer neural network to choose one of a finite number of actions based on an infinite number of possible input states. The novel model we propose is based upon the work by Roelfsema and colleagues in [10] and extends it to deal with the important group of delayed rewards tasks. Similarly to [11], it learns Q-values in the output layer. Unlike [11], this paper focuses on the Hebbian nature of the learning dynamics and the connection to standard backpropagation (BP) in reinforcement learning. To do so, we did not explicitly model the transient neuron type used in [11] (which does not change the underlying theoretic comparison to BP but makes it harder to identify). As shown in [10], the model can easily be extended to actor-critic architectures and more than three layers. Our theoretical results lead to the prediction, that the novel learning scheme (and thus also [11]) is superior to the previous framework presented in [10] which we confirmed in two different experiments. Finally, we demonstrated that attention-gated reinforcement learning is closely related and in some cases formally identical to standard error backpropagation in the same neural network architecture. This is a remarkable result since this proves that the biologically plausible mechanisms employed here result in the same effective neural plasticity rule that aims at optimizing an error energy function as in the mathematically motivated BP learning scheme. Additionally, this connection proves that whenever the network (seen as a function) can describe the mapping of states to action values, SAGREL will succeed in finding at least a local solution to the error minimization problem given small enough learning rates.

Acknowledgments. We thank P.R. Roelfsema for his helpful comments. TB is supported by the Graduate School MAEIC at Ulm University and FS and HN are supported in part by the SFB/TRR 62 funded by the DFG. HN was further supported by a grant from the German BMBF, project 01GW0763.

References

1. Brown, J., Bullock, D., Grossberg, S.: How the Basal Ganglia Use Parallel Excitatory and Inhibitory Learning Pathways to Selectively Respond to Unexpected Rewarding Cues. *Journal of Neuroscience* 19(22), 10502–10511 (1999)
2. Faußer, S., Schwenker, F.: Learning a Strategy with Neural Approximated Temporal-Difference Methods in English Draughts. In: ICPR, pp. 2925–2928. IEEE (2010)
3. Faußer, S., Schwenker, F.: Ensemble Methods for Reinforcement Learning with Function Approximation. In: Sansone, C., Kittler, J., Roli, F. (eds.) MCS 2011. LNCS, vol. 6713, pp. 56–65. Springer, Heidelberg (2011)
4. Felleman, D.J., Van Essen, D.C.: Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cerebral Cortex* 1(1), 1–47 (1991)
5. Frank, M.J., Badre, D.: Mechanisms of Hierarchical Reinforcement Learning in Corticostriatal Circuits 1: Computational Analysis. *Cerebral Cortex* 22, 509–526 (2011)
6. Gustafsson, B., Wigström, H.: Physiological mechanisms underlying long-term potentiation. *Trends in Neurosciences* 11(4), 156–162 (1988)
7. Joel, D., Niv, Y., Ruppín, E.: Actor-Critic Models of the Basal Ganglia: New Anatomical and Computational Perspectives. *Neural Networks* 15(4-6), 535–547 (2002)
8. Malinow, R., Miller, J.P.: Postsynaptic Hyperpolarization During Conditioning Reversibly Blocks Induction of Long-Term Potentiation. *Nature* 320, 529–530 (1986)
9. Pennartz, C.M.A.: Reinforcement Learning by Hebbian Synapses with Adaptive Thresholds. *Neuroscience* 81(2), 303–319 (1997)
10. Roelfsema, P.R., van Ooyen, A.: Attention-Gated Reinforcement Learning of Internal Representations for Classification. *Neural Computation* 17, 2176–2214 (2005)
11. Rombouts, J.O., Bohte, S.M., Roelfsema, P.R.: Neurally Plausible Reinforcement Learning of Working Memory Tasks. In: NIPS, pp. 1880–1888 (2012)
12. Salin, P.A., Bullier, J.: Corticocortical Connections in the Visual System: Structure and Function. *Physiological Reviews* 75(1), 107–154 (1995)
13. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, London (1998)
14. Vitay, J., Hamker, F.H.: A Computational Model of Basal Ganglia and its Role in Memory Retrieval in Rewarded Visual Memory Tasks. *Frontiers in Computational Neuroscience* 4(13), 1–18 (2010)
15. Williams, R.J.: On the Use of Backpropagation in Associative Reinforcement Learning. In: ICNN, vol. 1, pp. 263–270 (1988)
16. Wörgötter, F., Porr, B.: Temporal Sequence Learning, Prediction, and Control: A Review of Different Models and Their Relation to Biological Mechanisms. *Neural Computation* 17(2), 245–319 (2005)