Efficient concept formation in large state spaces

Efficient Concept Formation in Large State Spaces[⋆]

Fredrik Mäkeläinen, Hampus Torén, and Claes Strannegård

Department of Computer Science and Engineering,
Chalmers University of Technology, Gothenburg, Sweden

Abstract. General autonomous agents must be able to operate in previously unseen worlds with large state spaces. To operate successfully in such worlds, the agents must maintain their own models of the environment, based on concept sets that are several orders of magnitude smaller. For adaptive agents, those concept sets cannot be fixed, but must adapt continuously to new situations. This, in turn, requires mechanisms for forming and preserving those concepts that are critical to successful decision-making, while removing others. In this paper we compare four general algorithms for learning and decision-making: (i) standard Q-learning, (ii) deep Q-learning, (iii) single-agent local Q-learning, and (iv) single-agent local Q-learning with improved concept formation rules. In an experiment with a state space larger than 2^{32}, it was found that a single-agent local Q-learning agent with improved concept formation rules performed substantially better than a similar agent with less sophisticated concept formation rules and slightly better than a deep Q-learning agent.

Keywords: autonomous agents · artificial animals · efficient concept formation · adaptive architectures · local Q-learning.

Neuroplasticity refers to the capacity of animals to alter their nervous systems in response to changes in the environment. The connectivity between neurons may change over time and neurons may be added and removed continuously in a life-long process [1]. Artificial neural network models are frequently based on static architectures that are only plastic in the sense that their connectivity patterns develop over time. Several neural network models also allow nodes to be added and removed, however. For instance, the cascade-correlation architecture adds one hidden neuron at the time [2], and the progressive neural networks grow new columns while retaining previously acquired knowledge [11]. There are also regularization techniques [3] and pruning methods [18] that reduce the size of neural networks, while improving generalization.

Reinforcement learning occurs across the animal kingdom, and its biological basis has been studied extensively [9]. Reinforcement learning algorithms, on the other hand, are powerful tools for learning and decision-making in a general setting [13]. Q-learning is a basic algorithm for learning an optimal policy from

[⋆] Research supported by the Torsten Söderberg Foundation Ö110/17.

experience for any Markov Decision Process [15]. The U-tree model has been used for building decision-trees for state representations [4] and local Q-learning has been used in a multiple agent setting for merging Q-values collected from multiple agents into a global Q-value [10]. Reinforcement learning algorithms have also been applied to homeostatic agents, whose single objective is to regulate their homeostatic variables and thus stay alive as long as possible [5, 19].

Artificial animals have been studied primarily in the context of artificial life [6, 14]. Stewart Wilson defined *animats* as a form of artificial animals, whose sole goal is homeostasis [16]. He also suggested the *animat path to AI* as a way of creating artificial intelligence by modeling animal behavior [17].

In this paper, we consider artificial animals and propose generic mechanisms for perception, learning, and decision-making. For perception we use a graph model that supports sequences and represents sensory concepts as single nodes (*cf.* grandmother nodes). This choice of graph model makes it relatively easy to define efficient rules for adapting the graph topology continuously. The purpose of our dynamic graph model is to support one-shot, life-long, on-line learning while avoiding catastrophic forgetting and the data hunger associated with deep learning.

This paper extends our previous work [12], with its single-agent local Q-learning and basic structural rules for adding new nodes by introducing radically improved rules for node formation. Section 1 presents the improved animat model. Section 2 presents an experiment in which our animat model is compared to four other models. The results of the experiment are presented in Section 3. Section 4 discusses possible directions for future research. Section 5, finally, draws some conclusions.

1 Animats

A schematic description of the *animat* model is given in Figure 1. Time proceeds in discrete ticks in the animat model and the animat is updated at each tick according to Algorithm 1. Code describing the model in full detail is open sourced and available at [7]. Now let us zoom in on the constituents of the animat.

1.1 Body

The *body* is the animat's physical representation. The body has its associated finite sets of variables called *sensors*, *needs*, and *motors*. Sensors and motors take boolean values, whereas needs take values in the real interval $[0, 1]$.

Needs are denoted by natural numbers i. The status of need i at time t is the real value $\iota_i(t) \in [0, 1]$. Intuitively, 0 means death, while 1 means full need satisfaction. Examples of needs are water, energy, and protein. Now, it is easy to define reward in terms of changes in the status of the needs:

Definition 1 (Rewards). *For each need i and time $t > 0$, the reward signal $r_i(t)$ is defined as follows:*

$$r_i(t) = \iota_i(t) - \iota_i(t-1). \tag{1}$$

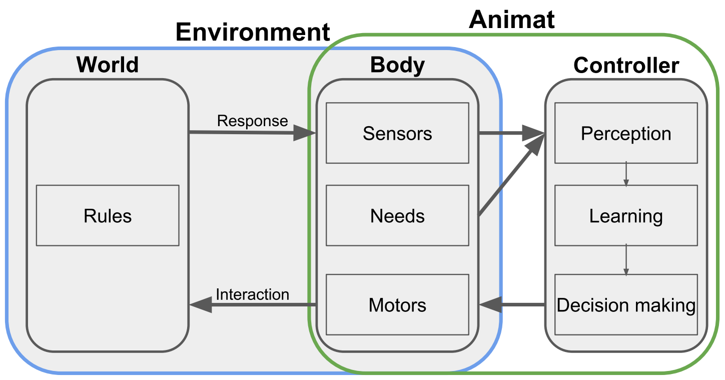

Fig. 1: The main constituents of the animat model in a reinforcement learning setting.

Algorithm 1: The update sequence for the animat.

```
Input: An animat $A$
while $A$ is alive do
    The body receives a response from the environment and updates its sensors
      and needs accordingly
    The controller receives the active sensors and the status of the needs from
      the body
    The top active nodes are determined
    The global Q-values are determined
    The local Q-values are updated
    Formation rules are activated
    The top active nodes are determined again
    The global Q-values are determined again
    The action goodness and utility are determined
    An action is selected and sent to the body
    The action is performed by the body
    The world evaluates the interaction
end
```

1.2 Controller

The controller is responsible for both learning and decision-making. Intuitively, it models the animat's brain. The controller is a function that takes a (physiological) state consisting of sensor values and need values as input and outputs an action, which is immediately executed by the motors. The controller either selects a random action (exploration), or an action that is expected to have the best consequences, based on its experience from previous interactions (exploitation). Next let us describe the controller in more detail.

1.3 Perception

Based on sensory input, a perception graph is used to approximate the state, an example of a perception graph is given in Figure 2. We construct the perception graph as a DAG where the input layer consists of the sensors. Initially, the perception graph consists only of the input layer, but through the use of formation rules, AND nodes can be added over time.

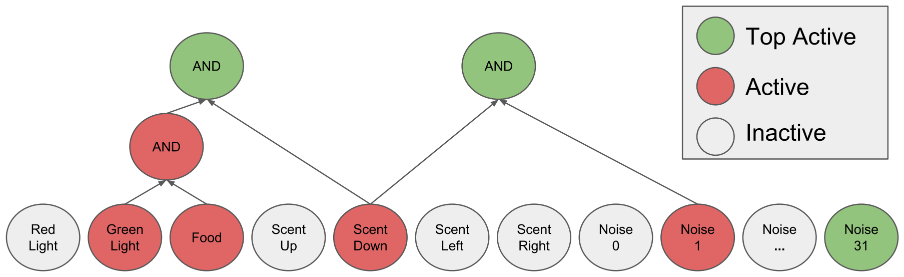

Fig. 2: A perception graph with 5 active and 3 top active nodes. The lowest layer contains the sensors.

Definition 2 (Perception graph). *A perception graph is a graph whose nodes (concepts) are sensors and binary AND-gates.*

Definition 3 (Perception graph activity). *At each time step the perception graph receives boolean values to its sensors. Those that receive the value True are called active. This activity propagates to the AND-nodes within the same tick. An AND-node is active if both its incoming signals are active.*

We use the symbol b for nodes of the perception graph and B_t for the set of all nodes at time t. The set of all active nodes at time t is denoted by B_t^A.

Definition 4 (Top activity). *An active node $b \in B_t^A$ is top active if the set of sensors it represents is not a subset to a set of sensors represented by another active node $b' \in B_t^A$.*

The set of all top active nodes at time t is denoted by B_t^{TA}. The set of top active nodes describes the current state to its maximum level of detail with respect to the structure of the perception graph.

1.4 Learning

In this section we present several experience structures that are updated each time step, and three new formation rules for the perception graph. The rules for expanding the perception graph are the main novelty of the present model compared to our previous work [12].

We will start with how the the quality of each action, with respect to either a single top active node or the set of all top active nodes, are updated and calculated in this model.

Definition 5 (Local Q-values). *A local Q-value is a real-valued variable $Q_i(b, a)$ that reflects the expected response to the status of need i when performing action a, given that node b is top active.*

Definition 6 (Global Q-values). *A global Q-value is a real-valued variable $Q_i^{global}(B_t^{TA}, a)$ that reflects the expected response to need i when performing action a given the set of top active nodes. It is defined as follows:*

$$Q_i^{global}(B_t^{TA}, a) = \frac{\sum_{b \in B_t^{TA}} Q_i(b, a)}{|B_t^{TA}|}. \tag{2}$$

Definition 7 (Update local Q-values). *The update of the local Q-values is based on Q-learning where the main differences stem from the state representation, which is given by the set of top active nodes. At $t + 1$ the Q-values are updated for all previous top active nodes $b \in B_t^{TA}$, with respect to the selected action a_t, the received rewards $r_i(t + 1)$ and the new top active nodes $b' \in B_{t+1}^{TA}$, as*

$$Q_i(b, a_t) \leftarrow Q_i(b, a_t) + \alpha \left(r_i(t + 1) + \gamma \cdot \max_a \left[Q_i^{global}(B_{t+1}^{TA}, a) \right] - Q_i(b, a_t) \right), \tag{3}$$

where $\alpha \in [0, 1]$ is the learning rate and $\gamma \in [0, 1]$ is the discount rate.

We will now move over to the new formation rules, but before we introduce them, we will present the information that they are based on.

Definition 8 (Pair reward). *$PairReward_i(b, b', a)$ is the probability that the reward for need i will be positive if action a is performed when b and b' are both top active.*

Definition 9 (Reward history). *The reward history $RewardHistory_i(b, a)$ is a pair (pos, neg), where pos (neg) is the number of times a positive (negative) reward for need i has been received when doing action a while node b has been active.*

To increase their chances of surviving, animats must be able to memorize what kind of objects are, e.g. suitable for eating and drinking.

Definition 10 (Positive stable nodes). *Based on the entries in $RewardHistory_i(b, a)$, the positive stable nodes $PositiveStable_i(a)$ is a list of all nodes that have received at least $\phi_{PositiveStable}$ positive rewards and no negative rewards for need i and action a.*

Definition 11 (Relevant nodes). *For each stable node $b \in PositiveStable_i(a)$ all nodes b' that seem to be correlated to b are added to the list of relevant nodes, $Relevant_i(b)$. If at least at least $\phi_{RelevantUpdates}$ updates have been performed for an entry in $RelevantTransition(b, b'|b'', a)$ and $RelevantTransition(b, b'|b'', a) > p_{Relevant}$, then b'' is added to $Relevant_i(b)$.*

Definition 12 (Relevant transition probabilities). *The relevant transition probabilities, $RelevantTransition(b, b'|b'', a)$, contains the conditional probability that b' is active given that b'' was active and action a was performed, where $b \in PositiveStable_i(a')$ and $b' \in Relevant_i(b)$.*

From studying the Definitions 4, 6 and 7, we made the following observation: all sets of top activities a node can be part of needs to have a coherent response from the environment for each action, otherwise conflicting rewards and conflicting global Q-values could prevent the agent from learning a good policy. Our new formation rules were designed with this in mind, and they will now be briefly described.

Definition 13 (Positive reward merge). *At each time step, flip a biased coin. If heads, then select two nodes b and b' with probability proportional to their entry $PairReward_i(b, b', a)$ and so that b and/or b' have received conflicting rewards, i.e. the entry in $RewardHistory_i(b, a)$ is $(> 0, > 0)$. Then if it does not yet exist, add $b'' = b$ AND b' to B_t.*

The positive reward merge creates connections for nodes with conflicting rewards, with the goal that the new node becomes a positive stable node. Entries in *PairReward* with high probability are more likely to be made first.

Definition 14 (Stable node merge). *Suppose a stable node $b \in PositiveStable_i(a)$ is active, $b \in B_t^A$. For $b' \in B_t^{TA}$, if it is not already represented, add $b'' = b$ AND b' to B_t.*

The stable node merge makes sure that all stable nodes receive a coherent response from the environment, i.e. we isolate them by forming new nodes with the top active nodes.

Definition 15 (Relevant node merge). *Suppose a relevant node $b' \in Relevant_i(b)$, is active, $b' \in B_t^A$. For $b' \in B_t^{TA}$, if it is not already represented, add $b'' = b$ AND b' to B_t.*

Similar to stable node merge, we also isolate nodes deemed relevant to a stable node.

1.5 Decision-making

In this section, we present the building blocks that are used by an animat to select an action in a potentially multi-objective setting.

Definition 16 (Action goodness). *The action goodness is defined as*

$$G_i(a, t) = \iota_i(t) + \omega Q_i^{global}(B_t^{TA}, a), \tag{4}$$

where $\omega \in [0, 1]$ is a constant.

Definition 17 (Utility). *The utility is defined as*

$$utility(a, t) = \min_i \left[G_i(a, t) \right]. \tag{5}$$

Definition 18 (Policy). *Flip a biased coin. If heads then select the action that maximizes $utility(a, t)$, otherwise select a random action.*

2 Experiment

Now let us describe the experiment, whose purpose was to evaluate the three new formation rules (Definitions 13, 14, 15) and make comparisons with some other models. Accompanying code can be found at [7].

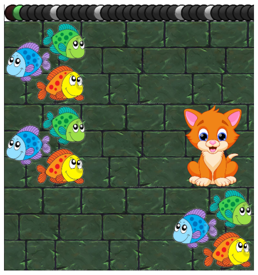

Fig. 3: The 3x3 cat world shown in a state with fish in three locations, red light off, green light on and 4 out of the 32 noise lights active, as shown in the top bar. The cat has to find the fish by using its smell sensors and then determine whether the fish is edible. Because of the noise, the state space of this world is greater than 2^{32}.

2.1 World

The world that is used in the experiment consists of a 3x3 bounded grid populated by a cat and fish, see Figure 3. When exploring the cat discovers that eating the fish sometimes results in (energy) reward, sometimes in punishment. For this environment, a green light indicates that the fish is safe to consume while a red light indicates the opposite. For the agent to find an optimal policy, it must learn to navigate towards fish and then only consume fish when the green light is active. To increase the size of the state space there are also lights that represent noise, these are activated randomly with $p = 0.25$, and they do not affect the received reward. The optimal policy is thus straightforward, but the problem lies in finding this policy with all the noise present.

2.2 Agents

In the experiment we evaluated five agents:

- **New Animat**, which is the animat model described in this paper.
- **Old Animat**, which is an adaptation of the animat model described in [12].
- **DQN**, which uses a two-layer ANN for approximating the Q-value. It also uses a target network and replay-memory similar to [8]. In total there are 200 weights to train.
- **Q-learner**, which is based on ordinary Q-learning, where each unique set of active sensors has an entry in its Q-matrix. Since the state space is vast, it is implemented using lazy initialisation.
- **Random**, which selects actions randomly.

The New Animat, Old Animat and Q-learner share the common Q-learning parameters: $\epsilon_{start} = 1.0$, $\epsilon_{decay} = 0.99$, $\epsilon_{min} = 0.01$, $\alpha = 0.05$, $\gamma = 0.9$, which are the exploration/exploitation parameters, learning rate and the discount rate, respectively.

Sensors The body of the cat has the following sensors: a green light sensor, a red light sensor, 32 noise light sensors, a fish sensor and four remote sensors for fish smell. So in total, the agent's perception of the environment is based on 39 boolean values.

Needs The cat has one need only: energy. The energy is not only affected by the actions, but it also decreases each time step with a constant decay rate of -0.02, an agent that is not following a good policy will usually see its energy reach 0 in about 40 time steps.

Actions The agent can perform five different actions $a \in \{\text{move up, move down, move left, move right, eat}\}$. The actions have the following impact on energy in terms of the received reward: any move, $r = -0.01$; eat fish while green light is active, $r = 0.3$; eat fish while red light is active, $r = -0.3$; eat nothing, $r = -0.015$.

2.3 Evaluation

For each agent, data is collected over 20 independent experiments where one experiment is divided into two parts: training and testing. Learning and exploration are turned on during training and turned off during testing. Each training episode lasts 200 time steps and is followed by a test episode, in which performance data from 20 test runs are collected. Each test run ends after 100 time steps or if the agent's energy level reaches 0.

3 Results

In this section, we present the results of the experiment. Figure 4 shows the performance of the five agents. It is clear that New Animat and DQN has the best performance. DQN is the quickest to improve but is soon overtaken by New Animat. New Animat performs best overall and stabilizes at a level above all other agents. Old Animat performs better than both the Q-learner and the Random agent.

Figure 5 shows that at the end of the experiment, Old Animat's perception graph contained ~ 2000 nodes, while New Animat's perception graph had stabilized around ~ 700 nodes already after 3000 time steps. Although Old Animat used almost three times the number of nodes compared to New Animat, it failed to match its performance level. The main difference between the two lies in the formation rules, and thus they are the explanation of the success of New Animat.

4 Future work

The animat model was extended with three new formation rules. They all have the goal of creating a coherent response for all top active nodes, by identifying

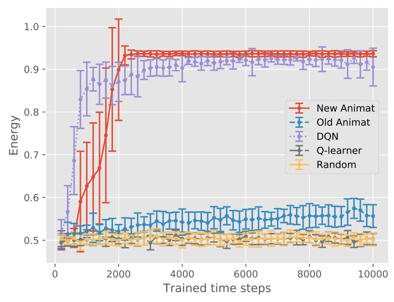

Fig. 4: For each agent, the mean and standard deviation of the energy with respect to the number of trained steps can be seen.

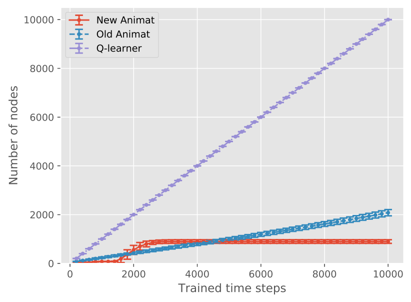

Fig. 5: The mean and standard deviation for the total number of nodes for New Animat and Old Animat, and the total number of unique states for Q-learner, with respect to the number of trained steps.

and isolating important nodes. Instead of isolating the important nodes by creating many nodes in the perception graph, we suggest that it might be possible to simply filter the top active nodes so that only the important nodes are taken into account. We believe this approach is well worth exploring further, since it has the potential to significantly reduce the number of nodes in the perception graph, while possibly maintaining the same level of performance.

5 Conclusion

Improvements to a computational model for artificial animals were proposed. The model combines generic mechanisms for homeostatic decision-making, local reinforcement learning, and dynamic concept formation. An experiment was conducted in which this model was compared to four other models: a previous version of the animat model, a deep Q-learning model, a basic Q-learning model, and a randomizer. The experiment was conducted in an environment with a state space of size exceeding 2^{32}, which rendered basic Q-learning infeasible. It was found that the improved animat model performed substantially better than the previous animat model and somewhat better than an optimized deep Q-learning model, despite starting with a blank slate architecture.

References

1. Draganski, B., May, A.: Training-induced structural changes in the adult human brain. Behavioural brain research **192**(1), 137–142 (2008)
2. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Advances in neural information processing systems. pp. 524–532 (1990)
3. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`

4. Jonsson, A., Barto, A.G.: Automated state abstraction for options using the U-tree algorithm. In: Advances in neural information processing systems. pp. 1054–1060 (2001)
5. Keramati, M., Gutkin, B.S.: A reinforcement learning theory for homeostatic regulation. In: Advances in neural information processing systems. pp. 82–90 (2011)
6. Langton, C.G.: Artificial life: An overview. MIT Press (1997)
7. Mäkeläinen, F., Torén, H., Strannegård, C.: Dynamic state representation for homeostatic agents (2018), `https://gitlab.com/fredrikma/aaa_survivability`
8. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
9. Niv, Y.: Reinforcement learning in the brain. Journal of Mathematical Psychology **53**(3), 139–154 (2009)
10. Russell, S.J., Zimdars, A.: Q-decomposition for reinforcement learning agents. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03). pp. 656–663 (2003)
11. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
12. Strannegård, C., Svangård, N., Lindström, D., Bach, J., Steunebrink, B.: The animat path to artificial general intelligence. In: Workshop on Architectures for Generality and Autonomy, IJCAI-17 (2017), `http://cadia.ru.is/workshops/aga2017/proceedings/AGA_2017_Strannegard_et_al.pdf`
13. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (1998)
14. Tuci, E., Giagkos, A., Wilson, M., Hallam, J. (eds.): From Animals to Animats. 1st International Conference on the Simulation of Adaptive Behavior. Springer (2016)
15. Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, King's College, Cambridge (1989)
16. Wilson, S.W.: Knowledge growth in an artificial animal. In: Adaptive and Learning Systems, pp. 255–264. Springer (1986)
17. Wilson, S.W.: The animat path to AI. In: Meyer, J.A., Wilson, S.W. (eds.) From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior. pp. 15–21. MIT Press (1991)
18. Wolfe, N., Sharma, A., Drude, L., Raj, B.: The incredible shrinking neural network: New perspectives on learning representations through the lens of pruning. arXiv preprint arXiv:1701.04465 (2017)
19. Yoshida, N.: Homeostatic agent for general environment. Journal of Artificial General Intelligence (2017). https://doi.org/https://doi.org/10.1515/jagi-2017-0001, dOI: https://doi.org/10.1515/jagi-2017-0001