

#72



Application of Deep Reinforcement Learning to Decision-Making System based on Consciousness

Kazuteru Miyazaki (teru@niad.ac.jp).



Kazuteru Miyazaki



National Institution
for Academic
Degrees and Quality
Enhancement of
Higher Education

ABSTRACT

We have interested in a **conscious decision-making system** in an environment where multiple types of rewards and penalties exist. We know a method using a basis function and a **penalty avoidance list**.

The **penalty avoidance list** has the problem :
If all actions are registered in the avoidance list
→ the actions are selected at random.

We propose a method for selecting actions using
deep reinforcement learning
in order to avoid such random selection.

The effectiveness is confirmed by Fig.2.

INTRODUCTION

We focus on a **decision-making method based on consciousness** from the engineering point of view.

Previously confirmed in the paper [3]
the effectiveness of adding a system called
a **secondary system** that monitors a **primary system**
that acts in response to external stimuli;

- The **primary system**: an **exploitation-oriented learning (XoL)** [1] or **reinforcement learning (RL)**
- The **secondary system**: evaluating the success or failure of the primary system and encouraging the primary system to re-learn if it was judged that the learning was not going well.

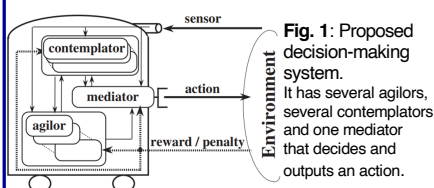
The paper [2] had been certain achievements as a **consciousness decision making system** based on the **penalty avoidance list**.

The method using the **penalty avoidance list** however has the problem that when all actions are registered in the avoidance list, the actions are selected at random.

We propose a method for selecting actions using **deep RL** in order to avoid such random selection.

The effectiveness of the proposed method is confirmed by numerical experiments in Fig.2.

Model Construction Process



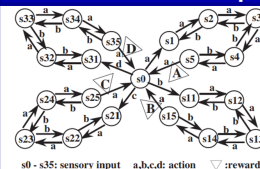
Agilor / Contemplator / Mediator

Agilor: It is defined as a module that learns quickly.
The **primary system** [3] has been used as an **agilor**.
It was defined as a system that makes some output to the environment in response to an input from the environment and defined as a system that learns policy according to the correspondence between inputs and outputs to the environment.
The **agilor** include different kinds of modules such as emotions, reflection, and policy learning.

Contemplator: We define a **contemplator** as a component that controls them as a whole or performs a time-consuming original computation.
The role of it is to realize the appropriate choice of action by advanced computation rather than the **agilor**.
We construct a contemplator using deep RL in addition to the penalty avoidance list proposed in the paper [2].

Mediator: The **agilor** and the **contemplator** each present candidates for action, but in general, only one type of action can be output to the environment.
A **mediator** is defined as a component that determines which module (**agilor** or **contemplator**) actually outputs its actions to the environment.
The effectiveness is verified with a simplified mediator.

Numerical Experiments



The agent exhibits four actions (a-d) for each desire levels (A-D). This decrease is predetermined for each desired level. It can properly observe all desire levels but the function that defines the decrement of each desire level is not known in advance.

The agent receives 21 types of sensory inputs. When it selects action a in s1, it transits to s3 with a probability of 0.5 and then to new state s6 located between s1 and s2 with a probability of 0.5.
The same uncertainty occurs in s11, s21, and s31.

On reaching a goal, represented by triangle in Fig.2, the corresponding desire level is increased.
This increment corresponds to a reward.
If a level decreases to 0, all desired levels are initialized and the agent returns to s0, which represents a penalty.

Consider the problem which action should be selected in s0. The solution depends on the change in desire level resulting from selecting an action and acquiring a reward. The purpose of the agent is to learn a rational penalty avoiding policy.

Specific Model Design

Agilor: We adapt a **minimum priority strategy (MPS)** for **agilor**.
in which the agent always aims to achieve the **minimum desire level**.
When the decrement in **desire level** $A > B$, **action a should be selected over b** even if the **desire level** of $A > B$.

- The threshold changes according to the following factors:
- The decrement of **desire levels** A, B, and difference between these levels,
 - The increment of the **desired level** when the agent reaches the goal.

Contemplator: The **contemplator** is based on a **penalty avoidance list** [2].
The list, corresponding to the set of lower layers that should not be selected in each state, is established for more emphasis on penalty-based learning.
This list is used to select an appropriate lower layer.
When the agent receives a penalty, the following information in the previous decision-making is registered in the avoidance list:

- Type of goal that needs to be achieved
- Desire level value of goal that needs to be achieved
- Desire level value of goal that received the penalty

This registration should take into account the interaction between "the goal that needs to be achieved" and "the goal that received a penalty."
The list is also updated when its range is broadened.
If the desire level changes according to a function other than the monotonically decreasing function, the procedure for registering and updating the "avoidance list" may be modified according to each function.
Only the relationship between the "required goal level of 0" (the minimum requirement), and the "target aimed at that time" was stored in the "avoidance list".

Deep Reinforcement Learning

The method using the **penalty avoidance list** has the problem that when all actions are registered in the **avoidance list**, the actions are selected at random.
We propose a method for selecting actions using **deep RL** in order to avoid such situations.

The input to the network is obtained by discretizing the values of **desire levels** A, B, C, and D into 64 pieces in Fig.3. If the value of each level $\geq$ the value written in the upper left corner of each box, 1 is given at the corresponding location, otherwise 0 is given as input.
ex) Fig.3b) is obtained when the values of **desire levels** A-D are 54, 99, 77, 16.
Two types of 3×3 filters are applied to such an 8×8 input. Setting the stride to 1 becomes two 6×6 outputs. Average pooling is applied to each 2×2 region to obtain $3 \times 3 \times 2$ output. For these 18 inputs, a fully-coupled network consisting of an intermediate layer 3 and an output layer 4 is constructed, and the coupling weight of the network is learned by the back propagation method.
The four neurons in the output layer correspond to the evaluation values of **desire levels** A-D. When Q-learning (QL) is used, the Q-value is output.
We can also consider using with Profit Sharing (PS).
The connection weight of the network is learned so that the largest value of output value indicates the **desire level** to be selected for the current input pattern.

Mediator: MPS is expected to be effective for many cases in this problem.
The **mediator** selects the **agilor** output as long as there are no problems.
If the agent receives a **penalty** when it follows an action from the **agilor**, the mediator selects the **contemplator** output in the same environment.
Furthermore, there are two types of contemplators:
a **penalty avoidance list** and **deep RL**.
First priority is given to the **penalty avoidance list**, and when random selection is required in that list, the output of **deep RL** is adopted.

Experimental Setup

- The **lower-level learning** that aims to achieve each goal from each state through the **lowest number of actions** is completed by **PS**.
- The number of actions required to reach each goal from state s0 is 6.
- The initial value of each **desire level** =100, the experiment was performed assuming four types of 1, 3, 5, and 7 as the amount of each **desire level** to be reduced when an agent outputs an action.
- The **desire levels** (A, B) = 1, C = {1 to 7}, D = {C to 7}.
For each combination of patterns to reduce each **desired level**, there are five types of rewards when the goal was achieved, that is, the amount of increase in the **desired level** = {20, 40, 60, 80, 100}.
→ 6,250 experiments were executed.
- MPS is effective in many cases. Try MPS that is an **agilor** in the first, if it doesn't work well we use the **contemplator**.
- It is possible to verify how much the change in the **desire level** can be followed by our method when there are four types of rewards.

Results and Discussion

- The number of times MPS failed during all 6,250 experiments is shown in the top row of Table 1.
- The average value (Ave.) and standard deviation (S.D.) of the number of times the method [2] failed are shown at the bottom. We call success not to get a penalty for 1 million actions, and failure when we do not succeed.

Table 1. Experimental Results.

	ST1	ST2	ST3	ST4
MPS fails	4,115	4,170	4,290	4,546
Paper [2] fails	3772.9 (4.5)	3856.0 (3.7)	4029.5 (2.2)	4284.2 (5.4)
Success of Proposed	218.0 (2.8)	187.4 (2.6)	101.4 (3.4)	5.4 (1.4)

- Ave. and S.D. of the number of times that the **contemplator** using **QL** based **deep RL** worked effectively in the combination in which the method [2] failed are shown at the bottom. We call success not to get a penalty for 1 million actions, and failure when we do not succeed.
- ST1 means the result of experiments in an environment that introduces uncertainty such that the shortest number of steps from state s1 to s3 changes to 1 or 3 with a probability of 50% for the environment in Fig. 2. ST2, ST3 and ST4 are the same as ST1.
- The experiment was performed 5 times, changing the random seed.
- From Table 1, it can be seen that both methods can cope with a wider variation than MPS because a constant number is obtained with a small S.D. in both cases. It can be seen that our method has a wider range of functions than the method in the paper [2].
- It was confirmed that the range of learning can be expanded by using **deep RL with the contemplator**.

CONCLUSIONS

- We proposed a method that incorporates **deep RL** into a **contemplator of a conscious decision-making system** consisting of **agilor**, **contemplator**, and **mediator**.
- The effectiveness was confirmed by numerical experiments.
- In the future, in addition to using **PS**, the effectiveness of the method in a multi-agent environment and the theorem for a mediator with learning will be planned.

ACKNOWLEDGMENTS & REFERENCES

I wish to thank Prof. Junichi Takeno for his helpful comments.

- [1] Exploitation-oriented learning PS-#
- [2] Proposal of an exploitation-oriented learning method on multiple rewards and penalties environments and the design guideline
- [3] The necessity of a secondary system in machine consciousness
- [4] Proposed decision-making system based on consciousness in multiple rewards and penalties environments