Contextual Affordances for Action-Effect Prediction in a Robotic-Cleaning Task

Francisco Cruz, German I. Parisi, and Stefan Wermter

***Abstract*— Affordances are a useful method to anticipate the effect of an action performed by an agent. In this work, we present a robotic-cleaning task using contextual affordances implemented through a self-organizing neural network to predict the effect of the performed actions and avoid failed states. Current results on a simulated robot environment show that our architecture is able to predict future states with high accuracy.**

I. INTRODUCTION

Affordances are available action possibilities for an agent in its environment [1]. They represent characteristics of the relation between an agent and an object in terms of opportunities the object offers to the agent. In robotics, they have been used as a triplet $affordance :=< object, action, effect >$ which encodes relationships between its components [2]. Hence, it is possible to predict the effect using objects and actions as domain variables, i.e. $effect = f(object, action)$.

Nevertheless, although this model has been shown to be suitable for many scenarios, it does not include context information which allows to anticipate effects in all situations properly. Naturally the fact of being able to utilize or not one affordance in certain states does not determine the existence of the affordance itself. On the contrary, the affordance is still present but cannot be applied in some states or can imply a different effect using a certain action with a certain object.

To overcome this issue, it is possible to use contextual affordances where an additional variable is considered to introduce information about the state [3]. In this case, the previous triplet is now extended to $contextualAffordance :=< state, object, action, effect >$ and to predict the effect we consider the function $effect = f(state, object, action)$. In some cases the object can also be a location, e.g., we can state that a hill affords climbing where the action is to climb and the object or rather the location is the hill. In general, we use the term object to refer to both objects and locations.

II. ROBOTIC SCENARIO

In this work we extend a reinforcement learning scenario previously presented in [4]. The task consists of a robot standing up in front of a table to clean it. The robot can make use of one arm and its gripper to manipulate objects in order to complete the cleaning task. In this regard, we define *objects*, *locations*, and *actions*. The scene includes two objects: a *sponge* and a *cup*. The table is divided in three zones, the *left* and *right* table sides and an additional position called *home* where we place the sponge during the execution of the task. We allow the robot to perform four actions: *get* <*object*>, *drop* <*object*>, *go* <*location*>, and *clean* the table section where the robot arm is placed at that moment. All actions are performed using the v-rep simulator.[1]

TABLE I

REPRESENTATION OF DATA USED AS INPUT AND OUTPUT FOR NEURAL CLASSIFICATION.

Data Representation								
Side conditions					**Locations**			
d-d	1	0	0	0	home	1	0	0
d-c	0	1	0	0	left	0	1	0
c-d	0	0	1	0	right	0	0	1
c-c	0	0	0	1	none	0	0	0
Actions					**Objects**			
get	1	0	0	0	sponge	1	0	
drop	0	1	0	0	cup	0	1	
go	0	0	1	0	free	0	0	
clean	0	0	0	1				

Each robot state in the scenario is represented as $s_t =< handPos, handObj, cupPos, sideCond >$, which takes into account four variables: (i) the robot's hand position, (ii) the object held in the hand, if any, (iii) the position of the cup, and (iv) the condition of each side of the table, i.e. whether the surface is still dirty or already clean. Nevertheless, from certain states the agent could perform actions which lead it to a failed state from where it is not possible to complete the task. Therefore, we use contextual affordances to avoid such failed states. For instance, let us assume the state is $s_t =< right, sponge, right, (dirty, dirty) >$. If the robot then cleans the *right* section of the table, it may shatter the cup, hence, it is not feasible to finish the cleaning task from the next state s_{t+1}.

III. OUR APPROACH

Our approach uses contextual affordances to predict the effect after an action has been performed by the robot in the cleaning scenario. We encode all the variables as presented in Table I where we show the data representation for side conditions, locations, actions, and objects. In side conditions,

Francisco Cruz, German I. Parisi, and Stefan Wermter are with the Knowledge Technology Group, Department of Informatics, University of Hamburg, Germany. {cruz, parisi, wermter}@informatik.uni-hamburg.de - http://www.informatik.uni-hamburg.de/wtm/.

[1]v-rep virtual robot experimentation platform - http://www.coppeliarobotics.com/

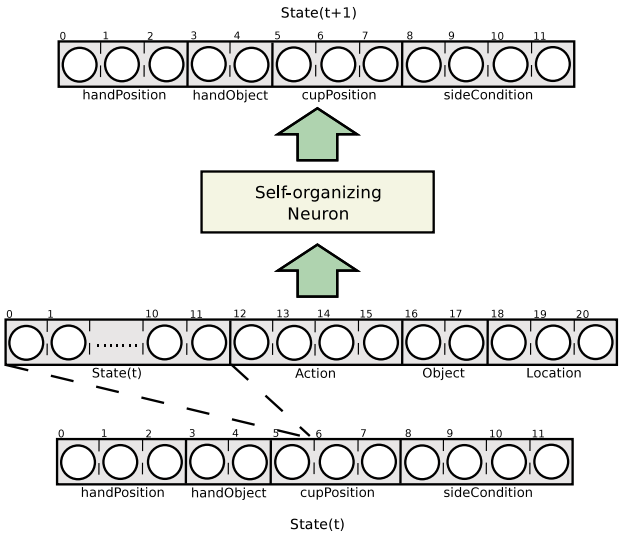

Fig. 1. Our architecture with a self-organizing complex neuron for future state prediction. In our scenario, the next state reached by the robotic agent represents the affordance effect.

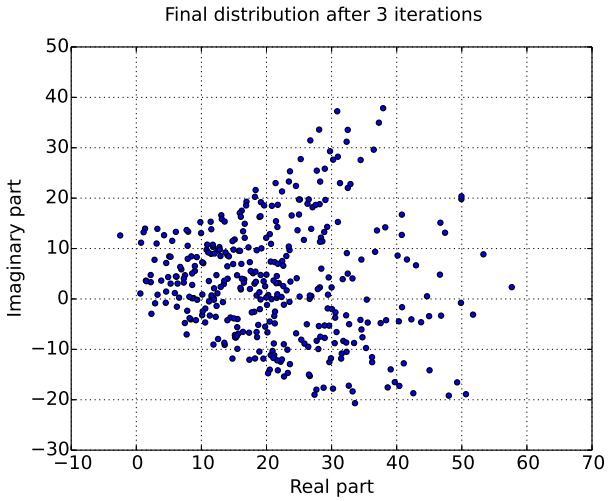

Fig. 2. Final distribution of the output projected into the complex domain after 3 iterations for future state classification.

letters *d* and *c* represent the fact of being dirty or clean for each part of the table.

Afterwards, we use this representation to create the training data. As input we use vectors with 21 variables containing information about the current state, the action, the object and/or the location, whereas each state is contained in the first 12 components of the vector considering the four variables that define a state (see Fig. 1). The output corresponds to the effect from contextual affordances encoded as 12 variables representing the next state. When the performed action leads to a failed state all components of the output vector are equal to zero. The data were created considering all possible states together with actions and objects (or locations). The total number of data samples is 368 instances for the training of the complex neuron.

We use a clustering technique with a complex-valued quadratic neuron [5] to define a new two-dimensional grid on the output space of the neuron as presented in [6]. The output of the neuron is $y = W^T X$, where the superscript $(\cdot)^T$ is the matrix transpose and $y \in \mathbb{C}$ is a complex scalar. For a given input vector X, the desired output y to be used in the learning algorithm is defined as the nearest intersection point of the grid lines of the complex plane. In practice, a rounding function Ψ is defined that rounds to the nearest integer for grid lines spaced at a fixed distance δ in both directions:

$$\Psi(y) = \frac{round(\delta Re(y))}{\delta} + i\frac{round(\delta Im(y))}{\delta}. \qquad (1)$$

In our implementation, we set $\delta = 0.001$ with a decaying learning rate. With each output value $\Psi(y)$ we associate the desired output state label for classification purposes.

After training, the self-organizing complex neuron is capable to associate high-dimensional input to low-dimensional output with 100% accuracy. The final distribution of the output after 3 iterations is shown in Fig. 2 in the complex plane, where the x and y axes are the real and imaginary parts respectively.

In this regard, our single complex neuron is able to predict the caused effect of performing an action in this robotic scenario with very few training iterations.

IV. DISCUSSION AND FUTURE WORK

The proposed architecture was able to successfully predict the effect of performing an action in the robotic-cleaning scenario in order to avoid failed states and effectively finish the cleaning task.

The self-organizing neuron allows to map the input vectors into valid states with few training iterations, which represents an advantage for online learning applications where the response time plays a crucial role.

In the near future, we are interested in extending the simulated scenario to a real robot platform obtaining the input vector using vision and replacing the output vector with the real state of the robot after performing the action.

REFERENCES

[1] J. J. Gibson, *The Ecological Approach to the Visual Perception of Pictures*, Boston: Houghton Mifflin, 1979.

[2] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, *Learning Object Affordances: From Sensory-Motor Coordination to Imitation*, in IEEE Transactions on Robotics, Vol. 24, No. 1, pp. 15–26, 2008.

[3] M. Kammer, T. Schack, M. Tscherepanow, and N. Yukie, *From Affordances to Situated Affordances in Robotics - Why Context is Important*, in Frontiers in Computational Neuroscience (Conference Abstract: IEEE ICDL-EpiRob 2011), Vol. 5(30), 2011.

[4] F. Cruz, J. Twiefel, S. Magg, C. Weber, and S. Wermter, *Interactive reinforcement learning through speech guidance in a domestic scenario*, in The International Joint Conference on Neural Networks (IJCNN), pp. 1341–1348, 2015.

[5] G. Georgiou, *Exact interpolation and learning in quadratic neural networks*, in The International Joint Conference on Neural Networks (IJCNN), pp. 230–234, 2006.

[6] G. Georgiou and K. Voigt, *Self-organizing maps with a single neuron*, in The International Joint Conference on Neural Networks (IJCNN), pp. 1–6, 2013.