An Artificial Synaptic Plasticity Mechanism for Classical Conditioning with Neural Networks

Caroline Rizzi Raymundo(✉) and Colin Graeme Johnson

School of Computing, University of Kent, Canterbury, Kent CT2 7NF, UK
{cr519,C.G.Johnson}@kent.ac.uk

Abstract. We present an *artificial synaptic plasticity* (ASP) mechanism that allows artificial systems to make associations between environmental stimuli and learn new skills at runtime. ASP builds on the classical neural network for simulating associative learning, which is induced through a conditioning-like procedure. Experiments in a simulated mobile robot demonstrate that ASP has successfully generated conditioned responses. The robot has learned during environmental exploration to use sensors added after training, improving its object-avoidance capabilities.

Keywords: Synaptic plasticity · Classical conditioning · Artificial neural networks

1 Introduction

Natural environments change often, which makes adaptation an essential survival skill for most organisms. Like animals, robotic systems may also find themselves in ever-changing environments. The need for more effective artificial intelligence, capable to overcome environmental changes, has led researchers to find inspiration in nature's solutions for adaptation, such as animals' reflexive behaviors [2,5], and the human brain [1] and its hormonal mechanisms [7,9].

In nature, the ability to learn new behaviors by means of associations between external stimuli is known to be essential for adaptation in a variety of animals, including humans [4]. However, to date, most works on adaptive systems seem to ignore this fact, seeking to adapt only the system's native behaviors and overlooking the need of a system that autonomously learn new ones [1,2,5,7,9].

In this paper, we propose a mechanism for allowing artificial systems to autonomously learn new skills based on environmental feedback and on its pre-existing skills. Our approach consists in an *artificial synaptic plasticity* (ASP) mechanism that builds on the classical artificial neural network (ANN) [3] for simulating associative learning. The system learns at runtime, through a procedure analogous to *classical conditioning* [8], to associate different environmental stimuli and use newly available information to solve problems in ways it was not trained for. We have evaluated ASP in a simulated mobile robot, which has successfully expressed conditioned responses.

Z. Zeng et al. (Eds.): ISNN 2014, LNCS 8866, pp. 213–221, 2014.
DOI: 10.1007/978-3-319-12436-0_24

This paper is organized as follows: Section 2 introduces ASP's biological inspiration, followed by its implementation in Section 3. We present experimental results in Section 4 and conclude in Section 5.

2 Biological Background

2.1 Classical Conditioning

Classical conditioning, first documented by Pavlov [8], is an important form of learning that involves the association of a behavioral response with an event that normally does not trigger that response. In his most famous experiment, Pavlov conditioned a dog to salivate on the ringing of a bell, after repeatedly ringing the bell whenever he presented food to the dog.

Pavlov argued that some reflexes are "hard-wired" and, therefore, do not need to be learned. For example, dogs do not need to learn to salivate when they smell food. This kind of reflex, which is native and automatic, is called *unconditioned response* (UR) and is triggered by an *unconditioned stimulus* (US). In Pavlov's dog example, the smell of food is an US that triggers salivation as an UR.

By contrast, the ringing of a bell is considered a *neutral stimulus* (NS), because it naturally produces no salivation in dogs. After pairing the bell sound with food smell, association occurs and it becomes a *conditioned stimulus* (CS), being able to trigger salivation by itself as a *conditioned response* (CR). Unlike URs, a CR can be extinguished after learned, for example, the dog will diminish its salivation response to the bell if food is repeatedly presented on the absence of the bell sound and vice-versa.

2.2 Synaptic Plasticity

At neural level, associative learning happens when a neuron is simultaneously excited by a strong and a weak electrical stimulus. This process gives rise to a phenomenon known as *long-term potentiation* (LTP), which strengthens the communication between two neurons [4]. LTP takes place at the *synapse*, which is the structure that connects two neurons and allows neural communication. However, LTP is not the only process that affects synapses' efficiency. *Long-term depression* (LTD) is a process similar to LTP, but instead of strengthening, it weakens synapses capability to transmit signals between neurons.

Synapses' ability to change their strength in signal transmission according to neural activity level, called *synaptic plasticity* or *Hebbian plasticity*, is known to play an important role in classical conditioning [4]. As a simplified example of this relation, a weak electrical stimulus could come from a CS, such as the ringing of a bell for Pavlov's dog, whereas a strong electrical stimulation could come from an US, such as the smell of food for Pavlov's dog. The target neuron (i.e., the one receiving these stimulations), in turn, could be a neuron that meaningfully contributes for triggering the dog's salivation response.

The pairing of both weak (from the bell) and strong (from the food) electrical stimulus generates LTP, which makes the target neuron more responsive to the weak stimulus. In the future, the weak stimulus will be able to activate the target neuron by itself, allowing the bell's sound to trigger the salivation response. If CS and US are repeatedly presented in the absence of each other, LTD occurs, leading the dog to no longer respond to the bell sound.

3 Artificial Synaptic Plasticity

We propose to simulate the neural mechanism of classical conditioning in the classical feedforward ANN [3]. As discussed in Section 2, the neural mechanism of classical conditioning consists in strengthening the signaling efficiency of synapses, which are represented as weights in the classical ANN. We argue that it is possible to generate an *artificial synaptic plasticity* (ASP) for artificial systems by gradually changing the ANN's weights according to the activity coincidence of its inputs.

In feedforward ANNs, the input value of a neuron i, known as *net input*, is given by Equation 1, where x_j is the output of neuron j, w_{ij} is the weight that connects neurons j and i, and b_i is the *bias* of neuron i.

$$net_i = b_i + \sum_j x_j w_{ij} \,. \tag{1}$$

The association process induced by ASP takes place after the ANN's training phase, during the system's operational cycle. Each input of the ANN is considered an external stimulus, which may be an *artificial conditioned stimulus* (ACS) or an *artificial unconditioned stimulus* (AUS). Therefore, the ANN's inputs are divided into two groups: the AUS group, depicted by the vector $\boldsymbol{u}$, of size p; and the ACS group, depicted by the vector $\boldsymbol{c}$, of size q. Together, these two stimuli vectors compose the input vector $\boldsymbol{x}$ of the ANN, with size $p + q$. A particular ACS cannot be an AUS at the same time and vice-versa. Therefore

$$\boldsymbol{x} = [u_1, u_2, ..., u_p, c_1, c_2, ..., c_q]^T \,;$$
$$net_i = b_i + \sum_{k=1}^{p} u_k w_{ik} + \sum_{j=1}^{q} c_j w_{ij} \,.$$

Consider, for now on, that the variables $k \in \{1, 2, ..., p\}$ and $j \in \{1, 2, ..., q\}$ are reserved for indexing AUS and ACS elements, respectively. Also, for the following explanation, we assume that the ANN's inputs are normalized in the range [0,1] and that the higher the value of an input, the higher its influence for generating the behavior of interest.

ASP's methodology consists in gradually changing the first-layer weights of ACSs, so that they become able to activate neurons of the second layer by themselves with the same pattern that AUSs would. Thus, each weight w_{ij} connecting ACS inputs to the second layer should be updated by a delta Δw_{ij}, so that $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$, where $w_{ij}(t)$ is the value of w_{ij} at time t. Note that this updating rule excludes weights w_{ik} that are related to AUS inputs.

The value of Δw_{ij} should consider not only the amount by which a given pattern is associated, determined by LTP, but also the amount by which the same pattern is extinguished/dissociated, determined by LTD. The variables Δa (association amount) and Δd (dissociation amount) control the level of association, reinforcing it (if $\Delta a > \Delta d$) or diminishing it (if $\Delta d > \Delta a$), so that

$$\Delta w_{ij} = \alpha_j(\Delta a_{ij} - \Delta d_{ij}) \,, \tag{2}$$

where $\alpha_j \in [0, 1]$ is the constant that determines the rate at which the ANN associates or dissociates stimulus c_j, and is called as the *association rate* (AR) of stimulus c_j. Hence, $\alpha_j = 0$ means that no association will occur, and the closer α_j is to 1 the faster the system associates c_j.

The value of w_{ij} cannot be increased/decreased indefinitely, because the ANN's outcome could be very different from the outcome produced by AUSs, diverging from the concept of classical conditioning. To avoid that, w_{ij} must be kept in a range $[w'_{ij}, w''_{ij}]$, where w'_{ij} is the initial value of w_{ij} and w''_{ij} is the desired conditioned value of w_{ij}. Hence, the closer w_{ij} is from a w''_{ij}, the closer it is from a complete association. Analogously, the closer w_{ij} is from w'_{ij}, the closer it is from a complete dissociation. Therefore

$$\Delta a_{ij} = (w''_{ij} - w_{ij}) \times assoc_j \,, \tag{3}$$

$$\Delta d_{ij} = (w_{ij} - w'_{ij}) \times dissoc_j \,, \tag{4}$$

where the variables $assoc_j$ (association factor) and $dissoc_j$ (dissociation factor), both in the interval [0,1], dictate the degree of synaptic-activity coincidence between c_j and $\boldsymbol{u}$. We will return to these variables later on.

According to our definition, the value of w''_{ij} should allow c_j to activate neuron i of the second layer with the same pattern that vector $\boldsymbol{u}$ would, which implies in Equation 5 (remember that $\{x \in \boldsymbol{x} \mid 0 < x < 1\}$). The constant $s_{jk} \in [0, 1]$ is the *sensitivity* of stimulus c_j to stimulus u_k. The matrix that maps the sensitivity between vectors $\boldsymbol{c}$ and $\boldsymbol{u}$ should be calibrated according to the designer's judgment, depending on the purpose of the associative learning and the architecture of the ANN. If correctly calibrated, the SM (sensitivity matrix) can prevent the system from learning "superstitions", i.e., patterns that are no more than random coincidences.

$$w''_{ij} = \sum_k s_{jk} w_{ik} \,. \tag{5}$$

In biological synaptic plasticity, association between a pair of CS and US occur when their values are simultaneously high, and it is analogous for ASP. Therefore, the higher the values of an ACS c_j and an AUS u_k, the higher the association between both (i.e., the higher the association factor $assoc_j$). However, c_j may be associated with more than one AUS, at different sensitivity values. Thereafter, it is more correct to state that $assoc_j$ is proportional to the average signal strength of $\boldsymbol{u}$ weighted by the respective sensitivities. This implies that

$$assoc_j = c_j \sum_k s'_{jk} u_k \,, \tag{6}$$

where $s'_{jk} = {}^{s_{jk}}/\sum_k s_{jk}$. Analogously, the dissociation (i.e., the extinction of an association) of an ACS c_j with an AUS u_k should occur when these stimuli are no longer paired. Therefore, the smaller the value of c_j and the higher the mean of $\boldsymbol{u}$ weighted by the respective sensitivities, the higher the dissociation between both (i.e., the higher the dissociation factor $dissoc_j$):

$$dissoc_j = (1 - c_j) \sum_k s'_{jk} u_k \,. \tag{7}$$

Also, according to classical conditioning, the higher the ACS and the smaller the AUS, the higher the dissociation; however, we have omitted it in the first version of ASP and considered it as future work. By replacing Equations 3, 4, 6 and 7 in Equation 2 and simplifying, we find

$$\mathit{\Delta} w_{ij} = \alpha_j \left[c_j (w''_{ij} - w_{ij}) - (w_{ij} - w'_{ij})(1 - c_j) \right] \sum_k s'_{jk} u_k \,, \tag{8}$$

where w''_{ij} is given by Equation 5. If a particular pair of associated stimuli, say c_j and u_k, have high input values at the same time, a net-input extrapolation may occur. This is because after being associated with u_k, c_j is able to mimic the effect of u_k in the ANN. Therefore, if both inputs are high, the neural network will receive a total input twice as high as it would if association had not occurred. So, the net input value must be restricted according to Equation 9, where $\underline{v}$ and $\overline{v}$ are, respectively, the minimum and maximum values that v can assume.

$$b_i + \sum_j \underline{x_j w'_{ij}} \; < \; net_i \; < \; b_i + \sum_j \overline{x_j w'_{ij}} \,. \tag{9}$$

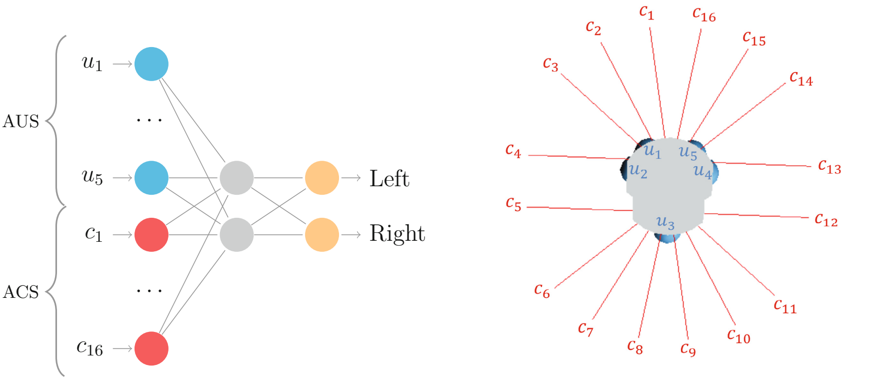
4 Experimental Evaluation

In this section, we evaluate ASP in a multi-stimulus association case. A robot is equipped with distance and touch sensors, but is trained (by means of the *backpropagation algorithm* [3]) to recognize and avoid obstacles using only touch sensors. Environmental exploration provides a natural conditioning, since it is probable that at least one distance sensor will measure high proximity to obstacles whenever a collision occur. By means of ASP, the robot is expected to gradually associate collision with proximity at runtime and eventually start to use information from the distance sensors to avoid obstacles before colliding.

4.1 Experimental Setup

We have used the robot simulator Webots [6] to simulate the Pioneer 2 robot[1], which is equipped with 16 distance sensors plus 5 custom touch-sensors. Figure 1a shows the architecture of the ANN that controls the robot's movements, whose inputs come from the sensors depicted in Figure 1b. The first five inputs,

[1] http://www.mobilerobots.com



(a) Neural network that controls the robot. Inputs are divided into unconditioned (vector $\boldsymbol{u}$) and conditioned (vector $\boldsymbol{c}$) stimuli. Outputs provide the speeds for the left and right wheels in radians per second.

(b) Aerial view of robot's sensors disposition. Red lines represent distance-sensor rays and blue spheres represent the contact area of touch sensors.

Fig. 1. Robot controller setup

representing AUS, come from the touch sensors and assume only binary values, where one means that a collision has been detected and zero means the opposite. The last 16 inputs, representing ACS, come from the distance sensors and assume integer values from 0 to 1024, where the higher the input value the closer the robot is to an obstacle. The robot's maximum detection range is 0.5 meters.

The robot was placed in a $4m^2$ box with a narrow and curved corridor leading to two dead ends, where it was initialized in three different positions and evaluated for five ARs: 0, 0.001, 0.01, 0.1 and 1. AR zero represents the execution of the pure ANN implementation, i.e., when there is no associative learning at all. In order to investigate performance variation deriving from noise error (simulated by Webots), we have executed each setup combination (3 initial positions and 5 ARs) 30 times, each for 5 minutes.

The SM has been configured to prevent the robot from associating random coincidences. For example, if the robot occurs to be near the left wall while touching the wall at its front, it may associate its left distance sensors with its frontal touch sensors, which is a mistake. Therefore, for this particular experiment, the SM should map the disposition of the robot's sensors, so that distance sensors are associated with the nearest touch sensor. Table 1 depicts the SM used in this experiment. Some distance sensors, such as c_3, are close to two touch sensors and, thus, have their sensitivity divided between them. By contrast, c_6 and c_{11} are relatively far from all touch sensors, so they have no sensitivity mapping.

Table 1. Sensitivity matrix (zeroed cells were omitted)

Touch	Distance Sensors															
Sensors	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
u_1	0.2	0.4	0.4	-	-	-	-	-	-	-	-	-	-	-	-	-
u_2	-	-	0.4	0.4	0.2	-	-	-	-	-	-	-	-	-	-	-
u_3	-	-	-	-	-	-	0.1	0.4	0.4	0.1	-	-	-	-	-	-
u_4	-	-	-	-	-	-	-	-	-	-	-	0.2	0.4	0.4	-	-
u_5	-	-	-	-	-	-	-	-	-	-	-	-	-	0.4	0.4	0.2

4.2 Results

Figure 2 counts all detected collisions during a complete run for each initial position of the robot. The number of collisions when ASP is used with AR 0.01 is about 70% smaller than when it is not used. The difference is even greater for AR 1.0, when the number of collisions is about 96% smaller than without ASP.

Outcomes for AR 0.001, however, are not as good as the results observed for the other ARs. This is because, when under AR 0.001, the robot got trapped in a corner from where it could not easily escape due to the disposition of its touch sensors. The trapping was persistent in runs starting from position 3 (causing the high collision count) and more occasional for runs starting from positions 1 and 2 (causing the high standard deviations). Despite trapping, results for the other ARs (0.01, 0.1 and 1.0) present a consistent decrease in number of collisions, with low standard deviation. Also, the lines are very close to each other, suggesting that outcomes are the same regardless the robot's start position.

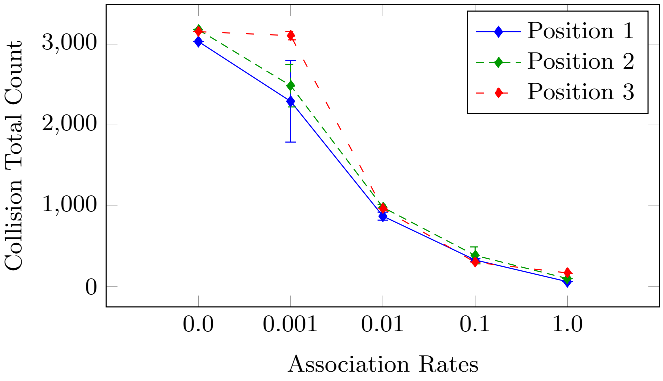


Fig. 2. Count of all detected collisions during an complete run

The robot's behavioral changes were beyond our expectations. Because of locomotion difficulty in the dead ends of the narrow corridor, the robot had to move more "carefully" in order to make a turn without touching the walls. As consequence, the more we increased the AR, the more time the robot spent making the turn in the dead ends, and the less it collided. This "cautiousness" is a positive collateral-effect that was neither deliberately designed, nor predicted.

The explored area is another unexpected and positive collateral-effect. When running without ASP, the robot's vision-range was limited to touch, so it could not perceive alternative (and perhaps better) paths to avoid an obstacle. As consequence, the robot kept doing laps in a small space. By contrast, when using ASP, the increased vision-range of the distance sensors improved the robot's space-awareness, which doubled the robot's explored area.

In addition to this experiment, we have also performed experiments for evaluating ASP regarding the dissociation process, i.e., when the robot forgets the association learned. Videos of all experiments are available on-line[2].

5 Conclusion and Future Work

We have presented an artificial mechanism of synaptic plasticity for generating associative learning in artificial systems through a conditioning-like process. Experiments with ASP successfully generated a conditioned response, allowing the robot to learn a new skill at runtime: use its distance sensors to avoid obstacles before bumping into them. This improved the robot's locomotion efficiency, allowing it to explore a bigger area and preventing physical damage.

For future work, we plan to use ASP for triggering and memorizing artificial emotions in computer systems. Neuroscientific findings indicate that emotions are essential for intelligent behavior and fast decision making in humans [4]. As for animals, emotions may be also valuable for artificial intelligence [5,7,9]. We hypothesize that ASP could be used for generating emotive conditioned responses, so that the system could adapt its behavior according to its past "emotional experiences" in relation to a particular place, entity or object.

In addition, we plan to test ASP with a real mobile robot in a physical environment in order to provide more realistic evaluation on its efficacy.

Acknowledgments. The first author is financially supported by CAPES, a Brazilian research-support agency (process number 0648/13-2).

References

1. Alnajjar, F., Murase, K.: A Simple Aplysia-Like Spiking Neural Network to Generate Adaptive Behavior in Autonomous Robots. Adaptive Behavior **16**(5), 306–324 (2008)
2. Arkin, R.C., Ali, K., Weitzenfeld, A., Cervantes-Pérez, F.: Behavioral models of the praying mantis as a basis for robotic behavior. Robotics and Autonomous Systems **32**(1), 39–60 (2000)
3. Callan, R.: The Essence of Neural Networks. The Essence of Computing Series. Prentice Hall Europe, London (1999)

[2] http://carolrizzi.github.io/ASP

4. Kolb, B., Whishaw, I.Q.: An Introduction to Brain and Behavior, 2nd edn. Worth Publishers, New York (2004)
5. Likhachev, M., Arkin, R.C.: Robotic Comfort Zones. In: Proceedings of SPIE on Sensor Fusion and Decentralized Control in Robotic Systems III, Boston, vol. 4196, pp. 27–41 (2000)
6. Michel, O.: Webots: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems **1**(1), 39–42 (2004)
7. Neal, M., Timmis, J.: Timidity: A Useful Emotional Mechanism for Robot Control? Informatica **27**(2), 197–204 (2003)
8. Pavlov, I.P.: Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex. Oxford University Press (1927)
9. Timmis, J., Neal, M., Thorniley, J.: An Adaptive Neuro-Endocrine System for Robotic Systems. In: IEEE Workshop on Robotic Intelligence in Informationally Structured Space, pp. 129–136. IEEE Press, Nashville (2009)