Chapter 41
Single SNN Architecture for Classical and Operant Conditioning Using Reinforcement Learning

Etienne Dumesnil
University of Quebec at Montreal, Canada

Philippe-Olivier Beaulieu
University of Quebec at Montreal, Canada

Mounir Boukadoum
University of Quebec at Montreal, Canada

ABSTRACT

A bio-inspired robotic brain is presented where the same spiking neural network (SNN) can implement five variations of learning by conditioning (LC): classical conditioning (CC), and operant conditioning (OC) with positive/negative reinforcement/punishment. In all cases, the links between input stimuli, output actions, reinforcements and punishments are strengthened depending on the stability of the delays between them. To account for the parallel processing nature of neural networks, the SNN is implemented on a field-programmable gate array (FPGA), and the neural delays are extracted via an adaptation of the synapto-dendritic kernel adapting neuron (SKAN) model, for a low resource demanding FPGA implementation of the SNN. A custom robotic platform successfully tested the ability of the proposed architecture to implement the five LC behaviors. Hence, this work contributes to the engineering field by proposing a scalable low resource demanding architecture for adaptive systems, and the cognitive field by suggesting that both CC and OC can be modeled as a single cognitive architecture.

DOI: 10.4018/978-1-7998-1754-3.ch041

INTRODUCTION

Two of the most fundamental learning mechanisms known to exist in nature are classical conditioning (CC) and operant conditioning (OC). CC consists in strengthening the association between an unconditional stimulus (US), which automatically triggers a response, and a conditional stimulus (CS), which does not. When the CS is followed by the US systematically enough, the CS ends up triggering the response even when the US is not presented (Pavlov, 1927). OC consists in strengthening the association between a response and a reinforcement or a punishment (Skinner, 1938). If the association is between the response and the reinforcement, the frequency of the response increases. However, if the association is between the response and the punishment, the frequency of the response decreases. Generally, this sequence of a response followed by a reinforcement or a punishment will only be systematic within a given context. For example, an experimental design could be set within which a rat will only receive food when pressing a lever if a green light was presented first. In this situation, the behavior of the rat at the beginning of the experiment would be exploratory. However, once in a while, the rat will press the lever while the green light is presented and therefore, food will be given to the animal.

Separate spiking neural network (SNN) architectures were recently proposed as very low resource demanding implementations of CC and OC in robotic controllers/brains (Cyr et al., 2015; Dumesnil et al., 2016; Dumesnil et al., 2016, "Robotic"). SNNs use time stamping instead of rate coding to represent individual neural firings (Gerstner & Kistler, 2002), which makes SNNs naturally suited for CC and OC representation. Indeed, in order to implement CC and OC, it is necessary to detect delays between stimuli, responses, reinforcements and punishments. Neuronal spikes thus appear to be a good information transmission method for extracting those delays. The architectures presented in (Dumesnil et al., 2016) were simulated in very large scale hardware description language (VHDL) using an adapted version of the synapto-dendritic kernel adapting neuron (SKAN) model (Afshar et al., 2014). The latter allows implementing the delay extraction process with very few hardware resources (Afshar et al., 2014).

Recent work also allowed to test the architectures proposed in (Dumesnil et al., 2016) within a dynamically changing real-world environment (Dumesnil et al., 2016, "Robotic"). For this purpose, a robotic validation platform was conceived and placed in a maze. It was first configured with a CC architecture and successfully demonstrated its capacity to learn its way through the maze. It was then reconfigured with an OC architecture and once again was successful in learning the correct associations. The separate CC and OC architectures presented in (Cyr et al., 2015; Dumesnil et al., 2016; Dumesnil et al., 2016, "Robotic") followed the generally accepted distinction between CC and OC learning (Weiss, 2014). However, a different perspective is introduced in (Dumesnil et al., 2016, "Robotic"), with the suggestion that they share a common SNN architecture and thereby, a single learning process. This single architecture was then validated through simulation and robotic implementation. However, the validation process described in (Dumesnil et al., 2016, "Robotic") only targeted CC and one form of OC: positive reinforcement (PR). In the present article, this work is extended to all four forms of OC: PR, positive punishment (PP), negative reinforcement (NR) and negative punishment (NP). Reinforcement and punishment can be functionally distinguished as translating respectively into an increase and a decrease in response frequency. The positive cases imply that a stimulus is presented to the system, while the negative cases imply that a stimulus stops being presented to the system. Thus, the four forms of OC are as follows: in PR, a favorable stimulus is presented following the response; in PP, an aversive stimulus is presented following the response; in NR, an aversive stimulus stops being presented following the response; in NP, a favorable stimulus stops being presented following the response (Murphy &