ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

New conditioning model for robots

Jean Marc Salotti

IMS laboratory, ENSC-IPB, Bordeaux University
146 Rue Léo Saignat 33076 Bordeaux Cedex France

Abstract. We present a neural network for the prediction of rewards in a conditioning model. It is based on two noisy-or and one noisy-and nodes and update rules inspired from BANNER technique. In specific cases, we show that the computation is similar to Rescorla-Wagner's equation, which inspired many computational models in the domain of conditioning.

1 Conditioning

The objective of this work is to define the behaviors of robots according to the theory of conditioning. Animal conditioning occurs when a stimulus is repeatedly presented before a reward or a punishment. Pavlov' experiments with dogs are well known and classical or operant conditioning have been studied for a long time [3]. However, despite several decades of works on conditioning models, it is still difficult to establish all the rules that govern the properties of conditioning. Most approaches are based on the original model proposed by Rescorla and Wagner [5] in which conditioning is characterized by associative strengths. Modification of the associative strength of a stimulus X after a new trial is given by equation (1). The increase is proportional to the salience of X (parameter α) and the efficiency of conditioning (parameter β). λ is the maximum strength and V_{Total} is the sum of all associative strength of the present stimuli.

$$V_X^{n+1} = V_X^n + \alpha_X \beta(\lambda - V_{Total}^n) \tag{1}$$

The associative strength of a given stimulus can be interpreted as the degree to which a reward is predicted. Another important model has been proposed by Klopf with further considerations by Grossberg [1], [2]. In these models, stimuli were represented by neurons and associative strengths were determined by synaptic weights. Other authors followed the same principles [1], [5], [6]. Sutton and Barto also proposed a model of classical conditioning based on well known reinforcement learning techniques (TD model, 1987 and 1990 [7]). Despite all these works, all models suffer from specific drawbacks. We propose in this paper a new prediction system for conditioning and its application in robotics. It is based on a specific neural network architecture corresponding to the nodes of a Bayesian network. The method is explained in Section 2. Some results are then presented in Section 3. In Section 4, we describe a simple application of our model with robots. We finally conclude with the perspectives of this work.

2 Proposed model

2.1 Description of the network

Stimuli are defined as perceptual events in the representation of robots. In a simplified world, there is only one reward (or only one unconditioned stimulus, which predicts the reward with probability 1) and the objective is to determine if one or several observed stimuli predict the reward event in the next few seconds or eventually if they inhibit it. Our neural network is defined by three levels (see Fig. 1). In the first level, the output of the neurons corresponds to the observation of the different stimuli. In the second level, the output of the neurons correspond to the probability of obtaining a true value for a hidden Boolean variable. There are two hidden variables. Each of them corresponds to a Noisy-Or node of the conditional probabilities of the first level (see next section). The first one can be considered as the probability of triggering the real but hidden cause of the reward event and the second one corresponds to the probability of triggering a hidden inhibitory mechanism that prevents the action of the cause and the observation of the reward. That second variable is necessary to allow inhibitory conditioning [4]. Inhibitory conditioning indeed occurs if the conditioned response is always observed after the detection of a given stimulus X (reward expected) but is never observed if Y is present at the same time. If a single noisy-or node had been used, the reward would have been expected if one of the causal mechanisms had been present. That specific problem has already been identified in previous work [6]. The neuronal output of the third level corresponds to the probability of observing another event. It is defined as the probability of presence of the cause with absence of the inhibitory mechanism. Inhibitory conditioning is thus easily integrated in the model. In general, the last event is the reward, but not necessarily.

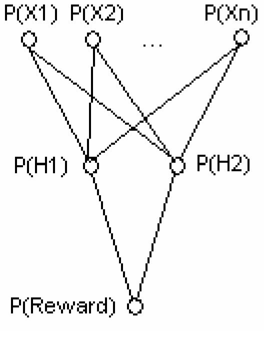

Fig. 1: Neural network model.

2.2 Neural computation

Pearl proposed the Noisy-Or to simplify the problem of updating many conditional probabilities in a Bayesian network [4]. We propose to implement his method as follows. First of all, we define $P(H_1|X)$ as the conditional probability of obtaining H_1=true during a limited period of time (for instance 5 seconds) following the observation of event X. In the neural network, synaptic weights correspond to conditional probabilities. For instance the conditional probability $P(H_1|X)$ is the synaptic weight between node P(X) and $P(H_1)$. If $X_1..X_n$ are possible stimuli events predicting H_1=true, in a Noisy-Or node it is sufficient to have an estimate of all $P(Y|X_i)$ to compute $P(Y|X_1..X_n)$ [4]. For the second level, the output of the neuron is therefore determined by the output of the neurons of the first level and their associative weights according to equation (2).

$$P(H_1) = 1 - \prod_i (1 - P(H_1|X_i)P(X_i)) \qquad (2)$$

A similar equation holds for the computation of $P(H_2)$. The difference only appears during the updates of the synaptic weights (e.g. the conditional probabilities). At the last level of the network, P(Reward) is computed according to equation (3).

$$P(\mathrm{Re}\,ward) = P(H_1)(1 - P(H_2)) \qquad (3)$$

2.3 Update rules

As it is suggested in the BANNER method proposed by Ramachandran, the update of the conditional probabilities can be performed according to a rule that minimizes the mean square error [8]. However, since we do not want to stop the robot during learning, we propose to take into account each error and to update the probabilities in an incremental way. Let us consider the error for a given trial. There are two cases. If the reward is observed after a given set of stimuli, the global error is 1-P(Reward) and if no reward is observed the global error is P(Reward). Since the computation of P(Reward) involves a multiplication between $P(H_1)$ and $P(H_2)$, the propagation of the error to the second level is trivial. If X_k is an observed stimulus, the update of the conditional probabilities $P(H_1|X_k)$ and $P(H_2|X_k)$ can be easily performed using a gradient descent technique, which is very similar to the one proposed in BANNER, see equations (4) and (5).

$$Err(P(H_1)) = P_{corr}(H_1) - P(H_1) \qquad (4)$$

$P_{corr}(H_1)$ is the correct value of $P(H_1)$ considering only the current observations. It is equal to 1 if the reward has been observed and 0 otherwise.

$$P_{t+1}(H_1|X_k) = P_t(H_1|X_k) + -\alpha\left(\frac{\partial Err(P_t(H_1)}{\partial P_t(H_1)}\right)$$

$$P_{t+1}(H_1|X_k) = P_t(H_1|X_k) + -\alpha \prod_{i \neq k}(1 - P_t(H_1|X_i)\,Vu(X_i)) \qquad (5)$$

where t is the time according to a given discretization, α is a learning rate smaller than 1, $Vu(X_i)$ is a logical variable equal to 1 if X_i has been observed and 0 otherwise. There is an increment or a decrement of the probability according to the sign of the error. A similar equation holds for $P(H_2|X_k)$ with a symmetric error. The problem is that we are dealing with probabilities and even though the update rule takes into account the partial derivative of the error and evoluates in the appropriate slope there is no guarantee that the probability would remain in the range [0..1] depending on the value of the learning parameter. We therefore propose taking into account the maximum modification of the probability. If the update rule increases the probability, the maximum local error is 1-P(H_1|X_k) and if it is decreasing, it is equal to P(H_1|X_k). We propose to multiply the right term by this maximum local error so that the update is always a fraction of the maximum allowed modification. The new equations are therefore (6)(7) and (8)(9), respectively, when the reward is observed and when it is not.

$$P_{t+1}(H_1|X_k) = P_t(H_1|X_k) + \alpha_1 \prod_i (1 - P_t(H_1|X_i)\, Vu(X_i)) \qquad (6)$$

$$P_{t+1}(H_2|X_k) = P_t(H_2|X_k) - \alpha_2 P_t(H_2|X_k) \prod_{i \neq k} (1 - P_t(H_2|X_i)\, Vu(X_i))$$

(7)

$$P_{t+1}(H_1|X_k) = P_t(H_1|X_k) - \alpha_3 P_t(H_1|X_k) \prod_{i \neq k} (1 - P_t(H_1|X_i)\, Vu(X_i)) \qquad (8)$$

$$P_{t+1}(H_2|X_k) = P_t(H_2|X_k) + \alpha_4 \prod_i (1 - P_t(H_2|X_i)\, Vu(X_i)) \qquad (9)$$

Note that in equations (6) and (9) the new term does not explicitly appear because it is now included in the product. Remark: We suggested that a predicted reward event would be expected during a fixed period of time after a stimulus event. We therefore have to memorize all events during that period. However, if a reward is observed at a given time t, the traces of the stimulus and the reward events might still be present at time t+1, t+2 and so on depending on time discretization. If this is the case, we have to apply equations (6) and (7) at every step. This is in fact an interesting property because the increase of the conditional probability is inversely proportional to the interval between the stimulus and the reward and that property has been observed in the domain of animal conditioning. Conversely, if the reward event is absent during all the fixed period, equations (8) and (9) are applied only once and the trace of the stimulus event is forgotten.

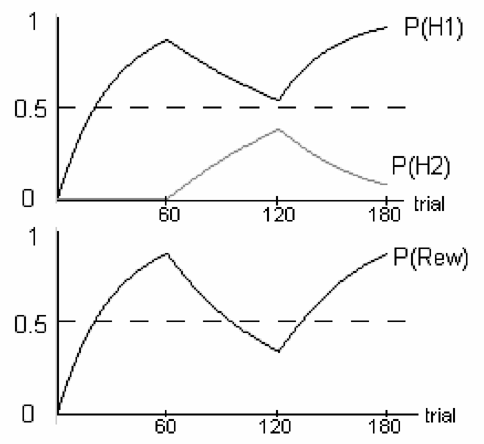

Fig. 2: Conditioning experiment: during the first 60 trials a stimulus event occurs and after 3 seconds a reward is presented. The probability of reward after that stimulus is equal to P(H1). Then during the next 60 trials, the stimulus is still presented but not the reward. The probability of reward after that stimulus

decreases below 0.5. There is extinction of conditioning. Finally, during the last 60 trials, the stimulus and the reward are presented (reacquisition of conditioning).

3 Results

Some results are presented Fig. 2. The following parameters have been used: α1=α2=0.002, α3=α4=0.008, time discretization: 0.2s, fixed period: 5s. It is important to note that similar curves would be obtained with different parameters. If we consider that conditioning is acquired if the probability of the reward exceeds 0.5, our experiment illustrates a conditioning followed by its extinction and finally its reacquisition at a faster rate. Other conditioning experiments are correctly described with the proposed model. Let us consider the blocking effect. Conditioning with a given stimulus Y is blocked or takes a long time if Y always follows a stimulus X and X is already a stimulus that predicts the reward. In equation (6), $P(H_1/Y)$ does not increase much because $P(H_1/X)$ is close to 1 and the product is proportional to 1-$P(H_1/X)$. Latent inhibition is characterized by an inhibition of conditioning if the stimulus has already been observed without the reward. In our model latent inhibition is observed because $P(H_2)$ increases before $P(H_1)$. Our model can account for many conditioning properties. More importantly, the main advantage of our model is that it is not necessary to reset the parameters before a new experiment. The conditioning with a given stimulus can be extinguished, reacquired, combined with another stimulus at all time. The model can therefore be used in robotics in real time.

4 Application in robotics

We implemented our model in Java for Lego Mindstorms robots. The robot is a simple rover equipped with ultrasonic sensors for measuring distance to an obstacle and a RFID sensor. There are three action modes. In the normal mode, the robot randomly chooses an action among several ones, such as "go forward 20 cm", "turn 45° east", "turn 45° west". If an obstacle is detected at a distance less than 20 cm, the robot switch on a red light and enters a reactive mode. It waits 2 seconds, then makes a 180° turn and return to the normal mode. The obstacle plays the role of an unconditioned stimulus. In the real world, stimuli are typically sounds or visual features and the problem of stimulus recognition is known to be hard. A RFID sensor has been used to simplify interactions with the robot. If an emitter is presented in front of the sensor, its radio frequency is immediately identified with 100% certainty. In our experiments stimuli are different emitters. Each time a stimulus is detected (and identified), its trace is memorized during 5 seconds. Meanwhile, if an obstacle is detected, the conditional probability of observing an obstacle after the detection of that stimulus is updated according to our model. We adapted the learning rates such that conditioning is functional after 3 similar situations. As a consequence, the fourth time the same stimulus is detected, the robot enters a conditioning mode, switch on a red light and goes foward during 5 seconds waiting for the detection of the obstacle. If it is detected before the end of the 5 seconds, it enters a reactive mode. If the obstacle is not detected it returns to the normal mode. Experiments have also been conducted with several stimuli. Extinction, reacquisition, latent inhibition, blocking and inhibitory conditioning (obstacle never expected after a given stimulus) have been successfully observed.

5 Conclusion

Our model has been briefly described. There are many other interesting results that could be discussed. The application of our model to the real world of robots is currently investigated. A promising perspective is the use of the model for the training of robots like the training of animals to do simple tasks such as sit down, jump, go and take an objet and so on. This is possible if the model is used for operant conditioning. Since conditional probabilities can take into account any event, it is easy to consider the actions of the robots as possible events that would predict rewards or punishments.

References

[1] Balkenius C. and Morén J., Computational models of classical conditioning: a comparative study, in Mayer, J.-A. , Roitblat, H. L., Wilson, S. W., and Blumberg, B. (Eds.), From Animals to Animats 5. Cambridge, MA: MIT Press (1998).

[2] Commons, M. L., Grossberg, S., and Staddon J.E.R. (editors), Neural Network Models of Conditioning and Action. Hillsdale, NJ: Lawrence Erlbaum Associates (1991).11. Klopf A., A neuronal model of classical conditioning, Psychobiology, 16, 2, 85--125 (1988).

[3] Pavlov, I.P., Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex (translated by G. V. Anrep). London: Oxford University Press (1927).

[4] Pearl, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA: Morgan Kaufmann Publishers, Inc. (1988).

[5] Rescorla R.A. and Wagner A.R., A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement, In Black, A. H., & Prokasy, W. F. (Eds.), Classical conditioning II: Current research and theory, 64-99, New York: Appleton-Century-Crofts (1972).

[6] Salotti J.M., "Noisy-Or nodes for conditioning models", Lecture Notes in Artificial Intelligence, Springer, from the conference "Simulation of Adaptive Behaviors", (SAB 2010), Paris, 24-28 August, 2010.

[7] Sutton R.S. and Barto A.G., A temporal-difference model of classical conditioning, Proceedings of the 9th Annual Conference of the Cognitive Science Society, 355--378 (1987).

[8] Sowmya Ramachandran, "Theory Renement of Bayesian Networks with Hidden Variables, PhD dissertation, The University of Texas at Austin, 1998