Forward Models and the Prediction of Undesired Situations

Bruno Lara, Juan M. Rendon, and Marcos Capistrán

Facultad de Ciencias
Universidad Autónoma del Estado de Morelos
Av. Universidad 1001, Col. Chamilpa
Cuernavaca, Mexico
{bruno.lara, rendon, marcos}@uaem.mx
(Paper received on August 10, 2006, accepted on September 25, 2006)

Abstract. The research presented in this paper highlights the importance of prediction and action as part of the perceptual process of a cognitive system. Through interaction with its environment a forward model is trained to predict multi-modal sensory representations formed by visual and tactile stimuli. The forward model is mounted on a simulated artificial agent and implemented as an artificial neural network, the data used for training comes from the simulated environment of the agent. Once trained, the predictions of the network are thoroughly analysed, the forward model is then implemented on the simulated agent in a typical obstacle avoidance task with a phototropic behaviour. Results show that forward models can represent a very important tool for the behaviour of autonomous agents.

1 Introduction

Classical views of cognition explain behaviour as a product of a direct, unidirectional line of information processing. Sensory inputs create a sensory representation and according to this a motor action is performed, actions are regarded as reactions, responses to stimuli. Most of the observed behaviour is considered a consequence of an innate stimulus-response mechanism that is available to the individual [1]. Known as the *information processing metaphor*, this framework thinks of the perception processes as modules that receive, modify and then pass the information available from the environment.

A novel approach to perception considers sensory input and action (motor output) as part of the same cognitive process. Only in recent times the idea that the anticipation of actions and/or sensory states influence behaviour is been appreciated. In the field of cognitive psychology these ideas have recently received much attention. Anticipations are now seen to play an important role on the coordination, planning and realisation of behaviour [2]. The linear information processing approach has given way to new frameworks according to which the direction of information flow is not anymore a one-way path.

At the centre of this view is the importance played by the body of the agent and its dynamic relation with the environment. An interesting proposal is presented by Hommel et al. [3]: The Theory of Event Coding (TEC) which is "*... based on the*

Special Issue in Neural Networks and Associative Memories
Research in Computing Science 21, 2006, pp. 161-170

central notion that perception, attention, intention and action share, or operate on, a common representational domain"[3]. TEC links perception and action functionally. It is this link and its coordination that provides the basis for adaptive behaviour [3].

Within this framework, sensory representations are also considered as consequences of actions. Any action realised by an agent on the environment has effects (action effects) and are the main reasons for behaviour. Representations that code for the environmental and bodily consequences of a movement become associated to motor representations coding for that actual movement [4]. The planning and control of actions becomes anticipatory when it is driven by the desired sensory situations or desired action effects.

The cornerstone of this research is the importance of predictions and actions as part of the perceptual process of the cognitive system. The cognitive model is learned and tested through interaction of the agent with its surroundings. The qualitative testing of the decisions the agent takes is based on the needs of the agent to act within its own world. For these reasons, we argue that the agent is grounded in its environment [5].

The objective of the research reported here is the better understanding of the effects and extent of the abilities the implementation of cognitive models affords an artificial agent.

2 Forward Models

A computational equivalent proposal are forward models. Mainly used in the field of motor control, a forward model is an internal model which incorporates knowledge about sensory changes produced by self-generated actions of an agent ([6],[7]). Given a sensory situation S_t and a motor command M_t(intended or actual action) the forward model predicts the next sensory situation S_{t+1}.

Forward models provide an alternative to the classical information processing views on perception. Möller [8] suggested forward models as a possibility to integrate visual perception and action generation.

In the realm of artificial autonomous agents anticipation and forward models can be used as a base for coherent behaviour. Autonomous agents interact with their environment in a direct way. A basic need for them to deal with their world is to predict the events happening. An anticipatory agent learning and using a forward model should be able then to have sufficient information to form planning strategies avoiding undesired situations and reacting timely to the hazards of its environment. Very interesting results have been presented by Dearden et al. [9], where a robot learns a forward model that successfully imitates actions presented to its visual system. Other implementations of cognitive approaches are discussed in Section 5.

The work presented in this paper attempts to provide an artificial agent with the necessary tools to predict undesired situations. The prediction of the agent is based on the inputs to the forward model, this prediction is characterized by the association of visual and tactile stimuli. In the context of TEC, this association can be considered an event composed by the motor command and the sensory situations (actual and desired) [3].

3 Experiments

The forward model is obtained by trainning an artificial neural network (ANN) with data coming from a simulated agent, the network is then tested on trajectories not seen during training. The whole system is implemented on the artificial agent to solve an obstacle avoiding task while seeking for a light source.

3.1 Environment and Data Collection

Using a robot simulator (Fig. 1a) a robot is placed in an arena with obstacles varying in size from 25-50 pixels. In the Figure the robot is moving forward from left to right.

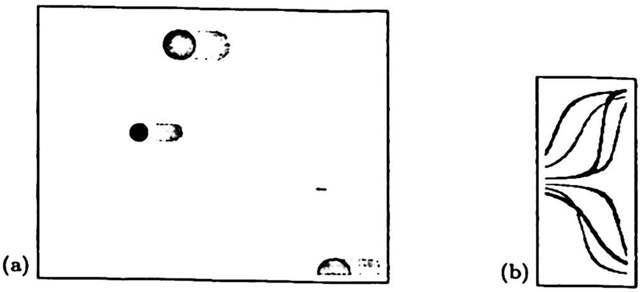

Fig. 1. Virtual World of the robot (a) and the recording of a single trajectory (b).

The robot has a diameter of 30 pixels and is equipped with an omni-directional linear black and white camera and a simulated frontal bumper. The robot moves in a straight line through the arena, taking snapshots of the environment. Every 20 pixels a new image is recorded. Figure 1b shows an 80 steps or 1600 pixels trajectory.

In Figure 1b the y axis represents the spatial dimension of the image, this is, the 360 degrees of the robot view and the x axis represents the time dimension. The front of the robot is located in the middle of the image in the spatial dimension. In the first snapshot, the obstacles can be seen close together and of a relatively small size. As the robot moves forward, the obstacles grow and move away from the centre of the robot until they are at the back of it, (ie. in the far right and far left of the image).

Obstacles are randomly placed to the right, left and front of the robot trajectory. The task of the robot is to predict whether it can perform a collision free trajectory of 1600 pixels. Figure 2 shows two trajectories where a collision occurs forcing it to stop (for convinience the trajectories are rotated 90 degrees). As the robot aproaches obstacles these grow in the image view, the obstaces are located in the central area of the image as this area represents the front of the robot.

3.2 Data Preprocessing and Forward Model Implementation

Before being used in the system the visual information is preprocessed. Originally, the images coming from the robot camera have a size of 1000 x 1 pixels, individual

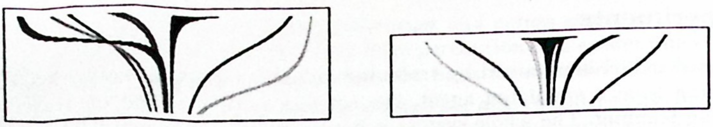

Fig. 2. Trajectory of robot with a crash at the left. Trajectory length: 62 steps. And trajectory of robot with a crash at the front. Trajectory length: 50 steps .

trajectories have different lengths as once the robot encounters an obstacle it stops recording images. The trajectories are preprocessed as follows:

- A section equivalent to 90 degrees (250 pixels) at the front of the robot is extracted from the whole image. As the system requires to predict collision on the front of the robot, this is the visual information relevant to the task.
- A butterworth one dimensional low-pass filter is applied to the spatial dimension of the images. This is done in order to get rid of high frequency redundant information.
- Foveal mapping in the spatial domain. Foveal mapping is basically a weighted subsampling of the image. The farther pixels are from the centre of the image the more they are subsampled, the pixels at the centre remaining nearly unchanged. For the subsampling an averaging mask was used.

The effects of applying the preprocessing algorithms to the image shown in Figure 1b can be seen in Figure 3.

Fig. 3. Frontal 90° after filtering (a) and after fovealisation (b).

A system is needed capable of predicting visual information as well as the simulated bumper states. This system can be implemented as a forward model of the form seen in Figure 4, where the current sensory situation is composed by the visual images V at times t, $t+1$ and $t+2$. This form of the input is expected to provide the model with the necessary information to learn the temporal structure of the data. The output of the forward model is the visual scene and the bumper state for time $t+3$, this is, V_{t+3} and B_{t+3} respectively.

The system performs a local symmetrical prediction. This is, every predictor takes its input from a section of the sensory input and predicts only the central pixel of that section. On the edges of the image this is not possible, so the input window is shifted in order to use the available information.

Given that the final size of the images is 50 pixels, the system consists of 50 feed-forward, fully connected, multi-layer perceptron networks, using sigmoid activation functions. The networks are trained using Resilient Back-propagation (RPROP) [10].

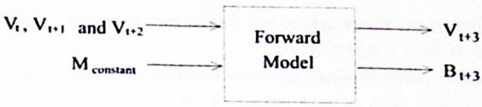

Fig. 4. Particular Forward Model

RPROP is a method to train MLP networks with a weight specific adaptation rule depending on the current and past values of each weight partial derivatives. This adaptive method has the effect of accelerating the learning process.

Each network has the following structure:

- Intput layer: 45 input units that code the visual data of three time steps (15 pixels for each time step).
- Hidden layer: 10 units, which at the time were a compromise between performance, network complexity and training time. Currently we are working on a *EEG* type of analysis of the dynamics of the hidden units activation. With this analysis, we hope to learn among other things whether the size of the hidden layer is the optimal one for the task.
- Output layer: 2 units, one corresponding to the predicted visual value and one for the predicted bumper value (0 for no collision and 1 for collision).

Training patterns for each network are prepared consisting of several images (46000 patterns) with a mixture of different collisions and collision-free trajectories. It is important to note that each one of the networks gets the same bumper value assigned, so that when there is a collision *all* networks should have an activation value of 1.

3.3 Testing the Forward Model

After training the networks for 6000 cycles of batch training, the testing is done with trajectories that were unseen by the network during training. The networks are expected to perform two kinds of prediction. First a one step prediction (OSP), this is, given the values of V_t, V_{t+1} and V_{t+2} the values of V_{t+3} and B_{t+3} are predicted. This is the standard network output. Second a *long term prediction (LTP)* which consists of using the predicted visual data back as input to the system. This prediction compares to an internal simulation of the events.

The OSP bumper values are not binary, instead the activation increases as the robot approaches an obstacle. More importantly, the networks that increase activity are those on the side of the robot where the obstacle is approaching or where significant changes on the visual field occur. A threshold is implemented during OSP to indicate a collision, if 5 or more neurons show an activation higher than 0.5 it means there is a collision.

The system should have a necessity to trigger LTP and this is also defined as a threshold. When 3 or more bumper output neurons present an activation greater than 0.3 during OSP, an internal simulation of the rest of the trajectory starts. Although the threshold might seem low it guarantees that there is no false predictions. The system was tested on 30 different trajectories. It never failed to trigger LTP when there was in reality a (future) collision on the trajectory. Likewise, it never triggered the LTP when presented with collision-free trajectories.

The visual output a typical testing run can be seen in Fig 5 which shows a trajectory with a collision on the left side of the robot. The system triggers LTP 2 steps before it actually happens, in the robot *world* this means 40 pixels or more than once its own size of 30 pixels.

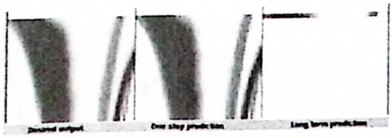

Fig. 5. System evaluation during a collision on the left side of the robot

Similar behaviours are observed during different trajectories. The system triggers LTP when a collision is likely to occur, most importantly during LPT the activation of the bumper units signals the presence of a crash.

The activation of the whole array of bumper output neurons for the last step of the LTP for three different collisions is presented in Fig. 6. It can be seen that the output neurons presenting higher activation are those located on the side of the robot where the collision is going to occur. In the case of the frontal collision, the activation is higher on the networks where changes in the visual field occur.

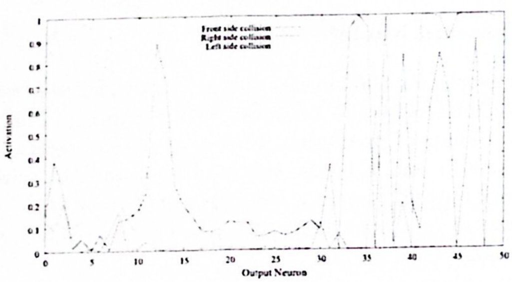

Fig. 6. Activation of the output neurons for bumper states at the last step of LTP

It is worth noting that Fig. 2 show trajectories and collisions before pre-processing. Fig 5 shows a trajectory with a collision after preprocesing, the fovealization process has the effect of *stretching* the central region of the real image. It is for this reason that in Fig. 5 the obstacle seems to be very far from the center of the image when in fact it is not. The existance of a crash is coded in the data, the system however, detects the crash correctly just due to the activation of the bumber output neurons.

3.4 Error Measure

A Sobolev norm E_S is applied to the visual data to find an objective measure of the error, such that:

$$E_S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 + \frac{1}{n-1}\sum_{i=1}^{n-1}(y_i - \hat{y}_i)^2 \tag{1}$$

where $y_i = x_i - x_{i+1}$ and $\hat{y}_i = \hat{x}_i - \hat{x_{i+1}}$

The first term of Eq. 1 is the *Sum Squared Error (SSE)* between the real data x and the predicted data $\hat{x}$. The second term provides a measure not only of the difference between the two sets of data but of the posible existence of oscillation between the data and the prediction.

Fig. 7a shows the E_S for several trajectories during 40 time steps of Long Term Prediction (LTP). Fig. 7b shows the second term of Eq. 1 for the same period of time and the same trajectories, showing that the oscillations between real data and prediction are minimal. The curves on Fig. 7b stick to the general shape of their counterparts on Fig. 7a.

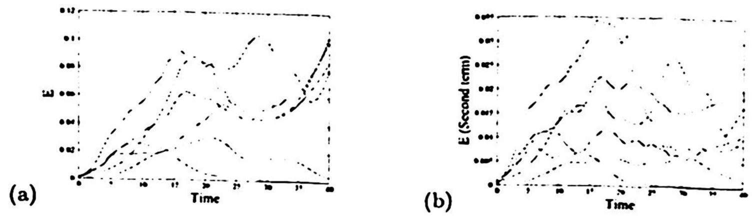

Fig. 7. E_S and the second term of Eq. 1 during 40 steps of LTP

The different shapes of the curves are due to the different starting points of the LTP as well as to the characteristics of the data. The two curves that tend to zero are predictions where no crash occurs, therefore the obstacles in the visual data dissappear.

4 Implementation

The trained forward model was implemented on the simulated agent used for collecting the data. The data coming from the camera was preprocessed and fed to the network as the agent was moving. The robot has the task of seeking a light source, avoiding collisions with nearby obstacles. The light sensing ability is completely independent of the vision or tactile senses of the robot. The robot, the light source and a number of objects are set on an initial random position.

Three behaviours are defined: a) light seeking. making the robot head directly into the light, b)prediction, if detecting high activation of the bumper units, the robot stops moving and performs an internal prediction and c) obstacle avoidance. The same thresholds are used; when there is activation on the bumper units long term prediction (LTP) is triggered. In case LTP finds the possibility of a future crash the obstacle avoidance behaviour takes over the light seeking behaviour.

As shown in Fig. 6 the activation of the bumper output units conveys information about the location of obstacles. This information is used to take avoiding action. Depending on the units that are registering the highest activation the robot decides where to turn. Fig. 8 shows the robot making its way through a set of obstacles on its way towards the light source. The robot performs a straight trajectory until the bumper units present high activation. The robot then performs an internal prediction which prompts it to change course.

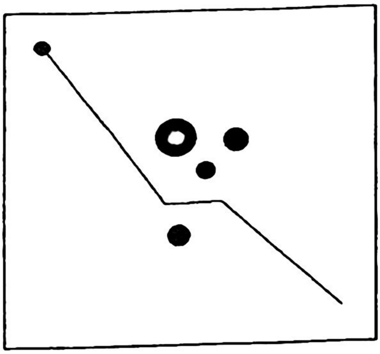

Fig. 8. Robot seeking a light source and avoiding a set of obstacles.

It is important to note that the robot can only predict sensory situations when the motor command has been constant for at least 3 steps. This means that once the robot turns, either by changing course towards the light or by avoiding an obstacle, it needs to move 3 steps with a constant motor command before being able to perform a prediction.

5 Conclusions

The presented experiments show that anticipation and forward models can provide agents with useful strategies. The agent presented here is capable of predicting the consequences of its own acts and take provisions for future actions. The system does this by learning an association between visual and tactile stimuli.

Our work differs from other research using forward models in several aspects. Hoffman et. al [11] presented a chain of forward models that provides an agent with the capability to select different actions to achieve a goal situation and perform mental transformations. The main difference with this system is that the forward model presented here performs the prediction of the same event (collision) using two sensor modalities (tactile and visual). The visual input to our forward model is completely grounded in that it does not have any sort of assumptions about distance. The system learns to estimate distance through interaction with its environment, associating visual input and tactile experience.

A form of anticipation is presented by Ziemke et al. [12], however their results for long term prediction are very constrained by the environment in which the robots evolved. In our case the model is capable of solving diferent scenarios, performing the necessary predictions and reacting timely to obstacles on its path.

Although the task presented here can be solved by different and simpler systems, this model presents the advantage of anticipation. The implementation of more complex tasks in the robot environment should prove that an anticipatory agent is capable of avoiding typical problems in which a reactive agent fails such as corners and dead-end situations.

6 Future Work

Further use of the system here presented can include behaviour in which the understanding of ego motion and its consequences is necessary. Current work, not reported here, includes analysis of the system predictions to distinguish between, changes brought up in the environment because of its own movement and those changes in the environment brought up by other agents and their respective actions.

The multi sensory representation of stimuli (visual and tactile) should also be present in the internal dynamics of the network. These dynamics should become more interesting with non-constant motor commands. As proposed by TEC [3], hidden units of the system would share resources when coding of actions and multi-sensory perception.

References

1. Witkowski, M.: Anticipatory learning: the animat as a discovery engine. In V., B.M., Gerard, P., O., S., eds.: Adaptive Behaviour in anticipatory learning systems (ABiALS'02), Edinburgh, Scotland (2002)
2. Butz, M.V., Sigaud, O., Gerard, P.: Internal models and anticipations in adaptive learning systems. In Butz M. V., Gerard, P., O., S., eds.: Adaptive Behaviour in anticipatory learning systems (ABiALS'02), Edinburgh, Scotland (2002)
3. Hommel, B., Müsseler, J., Aschersleben, G., Prinz, W.: The theory of event coding (TEC): A framework for perception and action planning. Behavioral and Brain Sciences 24 (2001) 849-937
4. Knoblich, G., Prinz, W.: Linking perception and action: An ideomotor approach. In Freund, H.J., Jeannerod, M., Hallett, M., Leiguarda, R.C., eds.: Higher-order motor disorders. Oxford University Press, Oxford, UK (2005)
5. Ziemke, T.: Rethinking grounding. In Riegler, A., Peschl, M., von Stein, A., eds.: Understanding Representation in the Cognitive Sciences. Kluwer Academic Publishers, New York (1999) 177-190
6. Kawato, M.: Bidirectional theory approach to integration. In Inui, T., McClelland, J., eds.: Attention and Performance XVI. MIT Press, Cambridge, MA (1996) 335 367
7. Blakemore, S.J., Frith, C.D., Wolpert, D.M.: Spatio-temporal prediction modulates the perception of self-produced stimuli. Journal of Cognitive Neuroscience 11(5) (1999) 551 559
8. Möller, R.: Perception through anticipation. A behavior-based approach to visual perception. In Riegler, A., von Stein, A., Peschl, M., eds.: Understanding Representation in the Cognitive Sciences. Kluwer Academic Publishers, New York (1999) 169 176

9. Dearden, A., Demiris., Y.: Learning forward models for robots. In: Int. Joint Conferences on Artificial Intelligence, Edimburgh (2005) 1440
10. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. (1993)
11. Hoffmann, H., R.Moeller: Action selection and mental transformation based on a chain of forward models. In: Proc. of the 8th Int. Conference on the Simulation of Adaptive Behavior, Cambridge, MA, MIT Press (2004) 213–222
12. Ziemke, T., Jirenhed, D.A., Hesslow, G.: Internal simulation of perception: a minimal neuro-robotic model. Neurocomputing 68 (2005) 85–104