ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Active and reactive use of virtual neural sensors

Massimo De Gregorio

Istituto di Cibernetica "Eduardo Caianiello" – CNR
Pozzuoli (NA) – Italy

Abstract. This paper addresses the possible use of virtual neural sensors, implemented by means of weightless systems, as active or reactive sensors. The latter, made possible by the intrinsic characteristic of weightless systems that can be trained on–line. These virtual neural sensors have been adopted for actual applications in different domains.

1 Introduction

A physical sensor, in general, takes a certain form of input (speed, distance, temperature, and so on) and converts it, through read–out circuitry, into readings that can be interpreted. On the contrary, a virtual sensor (also known as software sensor or estimator) is an abstract device supporting a particular sensing functionality. The functionality is realised through sensor value abstraction and sensor control abstraction [1]. The former provides the facilities to measure sensory parameters which are not directly related to physical devices while, the latter, gives helpful sensor information through a suitable interface.

Virtual sensors are frequently used in substitution of physical sensors when physical sensors are not feasible due to the price, size, weight, weakness, the quality of measurement they provide, or when no specific physical sensors are available [2]. Furthermore, a virtual sensor can be dynamically created and destroyed when no longer required: it has an explicit life cycle.

Data–driven virtual sensors [3] are introduced either when the analytical model of the system is unknown or the model development costs must be kept down. They estimate the dependence between known and unknown sensorial magnitudes statistically, from a training set including representative examples of operation. The most common implementation of such sensors is by means of artificial neural networks (ANN) [4]. ANNs can model and control dynamic processes because of their extremely powerful adaptive capabilities in response to non linear behaviours [5][6]. In fact, they can establish highly complex, non–linear, multidimensional associations between input and output [2].

The virtual neural sensors (VNS) can be used in different ways: to filter or transform other sensors data [7][8], to improve physical sensors performance [9], to mitigate the adverse effects of the environment parameters [6], to predict the final product quality on the basis of other sensorial information [10].

In this paper, we show two different ways of use of VNS' implemented by means of WiSARD–like systems [11]. In particular, a modified version of WiSARD [12] is adopted to implement active VNS', while RAM-discriminators are used as reactive VNS'.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

2 WiSARD as VNS

WiSARD is an adaptive pattern recognition machine which is based on neural principles. It is a weightless system whose basic components are RAM–discriminators. A RAM–discriminator consists of a set of N one–bit word RAMs with X inputs and a summing device (Σ). Any RAM–discriminator can receive a binary pattern of $X*N$ bits as input. The RAM input lines are connected to the input pattern by means of a so–called "random mapping". The Σ enables this network of RAMs to exhibit – just like other ANNs that more directly model features of biological neural networks – generalisation and noise tolerance.

Each discriminator is trained on a particular class of patterns, and classification by the overall multidiscriminator system is achieved in the following way. When a pattern is given in input, each discriminator gives a response (r) on that input. The responses are evaluated by an algorithm which compares them and computes the relative confidence c of the highest response (i.e., the difference d between the highest and the second response, divided by the highest response).

RAM–discriminators were selected as neural components to implement VNS' on the basis of the following considerations: RAM–discriminators are tailored for efficient implementation on conventional computers; the use of artificial neurons more closely reflecting biological neurons would not make any difference; they can be trained online. Furthermore, changing the training algorithm, these systems can produce "mental" images [12] to be used as VNS output.

2.1 Weighless systems for active or reactive VNS'

An ANN trained on different classes and then used to recognise objects or part of them is considered an active VNS. This means that, before to install the active VNS we need to train the corresponding ANN. On the other hand, a VNS that can adapt its behaviour in real time (reactive VNS) cannot be trained in advance. In order to obtain such a VNS, we need an ANN that can be trained on line: RAM–discriminators have this characteristic. A single RAM–discriminator is a reactive VNS if: 1) it is trained online with background examples; 2) its output is $1 - r$ instead of r; 3) the training and classification phases are continuously alternate during its use. In this way, the VNS will "react" if everything but the background is given as input.

We would like to point out that, in this paper, the outputs of both active and reactive VNS' have to be interpreted and evaluated by another system.

In the next sections, we present the use of VNS' in two different applications: robot global localisation [13] and intelligent video surveillance [14].

3 Active VNS for landmark recognition

In most robotic application the vision system plays a fundamental role in sensory data acquisition and usually it is the slowest and most computationally heavy module of the whole system. It comes as a consequence that having a fast processing vision system could be a crucial point for a generic robotic system. In

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

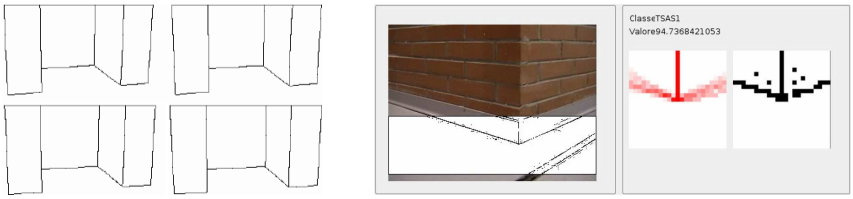

Fig. 1: Left: training set – Right: inward corner detected by the active VNS

order to reduce the elaboration time, it is sometime convenient not to process the whole image. In this case, the vision system has to be designed and implemented taking into account the particular environment in which the robot acts and its main goal (self localisation).

The VNS for landmark recognition is an adapted version of the one used in a previously designed hybrid neurosymbolic system [15]. In this case, corners between walls represent the natural landmarks the VNS has to detect and classify. In order to generate an appropriate training set, it is necessary to create, for each RAM–discriminator, different images representing corners seen from different visual angles. This is due to the fact that it is not predictable, neither decidable, the angle from which the robot will see the corner (see Figure 1 Left).

On this particular scene, a virtual camera, positioned with the same parameters of the real one, analysed the possible cases. The scene has been rotated step by step (5 degrees step) and a frame has been captured for each new angle. From each frame two types of "relevant" corners have been extracted to form the corresponding training set ("inward" and "outward" corners).

The robot camera tilt inclination has been fixed to -15 degrees; this means that the landmarks to be detected are going to be only in the lowest part of the image (see Figure 1 Right). Moreover, to improve the vision system performances, the VNS takes the input by a squared spot (a sort of "attention window") that scans just that part of the image trying to detect and classify the landmarks. The squared spot content is processed by the VNS only if it contains a certain amount of black pixels. So doing, we obtain a vision system capable of quickly detect the landmarks. The VNS actively looks for landmarks and each time it detects a corner, it classifies this feature by showing both the corresponding "mental" image and the spot content (see Figure 1 Right).

The VNS outputs are interpreted by a symbolic system (BDI agent [16]). During the robot navigation, the agent collects information about recognised landmarks and their sequence. The robot keeps on navigating until the BDI agent has gathered enough information to locate the robot in the environment.

4 Reactive VNS' for video surveillance

Reactive VNS' have been adopted for a video surveillance system working 24 hours a day in an outdoor environment (in particular, railway tunnels [14]). The system has to alert the control room in case of "abnormal" situations: a

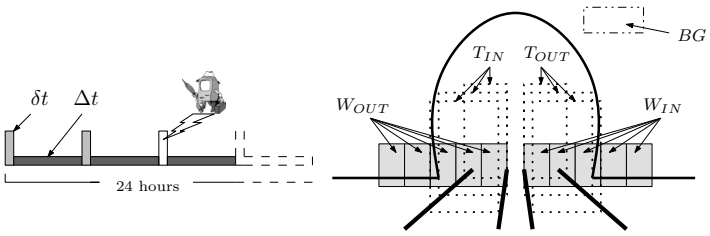

Fig. 2: Left: training and classification phases – Right: VNS setting

tree or a rock on the railway, a man entering the tunnel, and so on. In order to deal with light and weather changes and to distinguish between allowed and not allowed movements, different reactive VNS' have been placed in strategic places on the images coming from the video camera. To each virtual sensor a RAM–discriminator is associated.

There are three different VNS' placed on the image: green, red and blue. The green VNS' cover those parts of the scene where certain movements are allowed (for instance, train movements); the red ones, those parts where some movements could be very dangerous or risky. In order to allow the system to get accustomed to light and weather changes, the values reported by blue VNS' are used to normalise the other VNS outputs. This values normalisation, in addition to the noise resistant property typical of ANNs, allows the system to have the same performances during the entire day.

In order to make the system less susceptible to light and weather changes, two different time intervals are fixed: δt (training interval) and Δt (classification interval). During δt the VNS' are trained with a set of frames in which the alert level is "normal" (this allows the VNS' to react in case of "abnormal" situations). Δt is the classification time length and, for indoor application, can be as long as the user prefers (depending on the kind of application the system is adopted for); while, for outdoor applications a good length for Δt is about 10 seconds. With these settings, the system can easily face light and weather changes. These two time intervals are repeated during the entire day with the supervision of the symbolic module. In fact, if the symbolic module agrees, the VNS' can be trained otherwise they keep on classifying (see Figure 2 Left).

To each VNS a different interpretation is associated. The green VNS' are devoted to the detection of allowed movements. In fact, in this specific application, they are placed in a way that follows the shape of trains entering and leaving the tunnel. We label them respectively with T_{IN} and T_{OUT} (see Figure 2 right). From the activation sequence of T_{IN} (T_{OUT}) the system "understands" that a train is entering (leaving) the tunnel.

W_{IN} and W_{OUT} are red VNS' and they cover those parts of the scene the user believes to be risky. They are sized such as the shape of a human being nearby the tunnel. W_{IN} (W_{OUT}) are never covered by a train entering (leaving) the tunnel (in Figure 2 right is reported the scene of a tunnel from the best

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

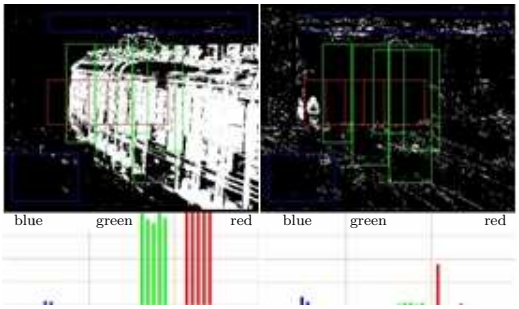

Fig. 3: Two different VNS outputs

angle of view). In case of superimposition, such as T_{IN} on W_{IN}, W_{IN} will be not considered anymore during the train entering. For red VNS' are important both the sequence of activation (a big animal crossing the railway) and the single activation (a rock on the railway or someone stopped before the tunnel entrance).

One or two blue VNS' (BG in Figure 2 right) are used to measure the noise degree of the scene: they are placed where no movement is possible. The values of these VNS' are used by the system just to normalise the other VNS' outputs.

The output of the VNS' is formed by pairs of *response–colour* and it is passed to the symbolic module that already knows the VNS strategic positioning and labelling. In Figure 3, two different patterns of VNS outputs are reported. In the case of a train entering the tunnel both red and green VNS' react and their activation is collected and evaluated by the symbolic module. In the presence of a man crossing the railway, we get only the activation given by red VNS'. A little blue VNS activation is present in both examples.

The VNS activation pattern produced on a single frame is considered but not directly evaluated by the system. In fact, in order to better understand and evaluate what in the scene is happening, the system stores the VNS activation values on the current frame and on a certain number of previous frames. All the pairs *response–colour* of these frames are grouped in one *response–colour* value that will be used to evaluate the scene. This new value (we refer to as *level of activation*) represents a sort of weighted mean: the closer is the frame to the current frame, the higher is considered its contribution to the weighted mean.

The levels of activation of red and green VNS' are the data used by the symbolic module to evaluate the scene and to set the alert level.

5 Conclusions

In this paper, we have shown the use of weightless systems as active or reactive VNS' in two different applications. The VNS' are just used to gather information about the environment and their outputs have to be necessarily interpreted and evaluated by another system (in our case, a system implemented by means of BDI agents). The coupling of VNS' with symbolic reasoning for interpreting their

outputs, makes the presented systems both very light from the computational and hardware point of view and quite robust in performances. Furthermore, these kind of VNS' can be readily put on hardware (micro–controller) further improving the computational performances. In fact, to implement these VNS' one needs only a certain amount of RAM and some simple control.

References

[1] Nigel Hardy and Aftab Ahmad. De-coupling for re-use in design and implementation using virtual sensors. *Auton. Robots*, 6(3):265–280, 1999.

[2] J. Estremera, P. Gonzalez de Santos, and J. A. López-Orozco. Neural virtual sensors for terrain adaptation of walking machines. *J. Robot. Syst.*, 22(6):299–311, 2005.

[3] M. Masson, S. Canu, Y. Grandvalet, and A. Lyngaard-Jensen. Software sensor design based on empirical data. *Ecological Modelling*, 120:131–139, 1999.

[4] S. Ablameyko. *Neural Networks for Instrumentation, Measurement and Related Industrial Applications (Nato-Computer and Systems Sciences, 185)*. IOS Press, Inc., 2003.

[5] Kenan Danisman, Ilker Dalkiran, and Fatih V. Celebi. Design of a high precision temperature measurement system. In Jun Wang, Zhang Yi, Jacek M. Zurada, Bao-Liang Lu, and Hujun Yin, editors, *ISNN (3)*, volume 3973 of *Lecture Notes in Computer Science*, pages 997–1004. Springer, 2006.

[6] J.C. Patra, E.L. Ang, N.S. Chaudhari, and A. Das. Neural network based smart sensor framework operating in a harsh environment. *Journal of Applied Signal Processing*, 2005(4):558–574, 2005.

[7] Xing Chen, Mingfu Cao, Yi Li, Weijun Hu, Ping Wang, Kejing Ying, and Hongming Pan. A study of an electronic nose for detection of lung cancer based on a virtual saw gas sensors array and imaging recognition method. *Measurement Science and Technology*, 16:1535–1546(12), August 2005.

[8] Michele Folgheraiter, Giuseppina Gini, Alessandro Nava, and Nicola Mottola. A bioinspired neural controller for a mobile robot. In *IEEE International Conference on Robotics and Biomimetics*, pages 1646–1651, Kunming, China, 17-20 December 2006.

[9] Duro R.J., Becerra J.A., and Santos J. Behavior reuse and virtual sensors in the evolution of complex behavior architectures. *Theory in Biosciences*, 120:188–206(19), 2001.

[10] R. Rallo, J. Ferre-Gine, A. Arenas, and F. Giralt. Neural virtual sensor for the inferential prediction of product quality from process variables. *Computers & Chemical Engineering*, 26(12):1735–1754, December 15 2002.

[11] I. Aleksander, W. Thomas, and P. Bowden. WISARD, a radical new step forward in image recognition. *Sensor Rev.*, 4(3), 1984, 120-124.

[12] Ernesto Burattini, Massimo De Gregorio, and Guglielmo Tamburrini. Generation and classification of recall images by neurosymbolic computation. In *Second European Conference on Cognitive Modelling - ECCM98*, pages 127–134, 1998.

[13] Paolo Coraggio and Massimo De Gregorio. WiSARD and NSP for robot global localization. In José Mira and José R. Álvarez, editors, *IWINAC (2)*, volume 4528 of *Lecture Notes in Computer Science*, pages 449–458. Springer, 2007.

[14] Massimo De Gregorio. A hybrid intelligent system for active video surveillance. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, pages 21–26, Rio de Janeiro, Brazil, 22-24 October 2007. IEEE Computer Society.

[15] Ernesto Burattini, Paolo Coraggio, Massimo De Gregorio, and Mariacarla Staffa. Agent WiSARD in a 3D world. In José Mira and José R. Álvarez, editors, *IWINAC (2)*, LNCS 3562. Springer, 2005, 272-280.

[16] Michael Wooldridge and Nicholas R. Jennings. "Intelligent agents: Theory and practice". *Knowledge Engineering Review*, 10(2), 1995, 115-152.