ART Networks as Flexible Means to Implement Dependability Properties in Autonomous Systems

K.-E. Großpietsch
EUROMICRO, P.O. Box 2043
Sankt Augustin, Germany
karl-erwin.grosspietsch@online.de

Tanya A. Silayeva
Moscow Aviation Institute, Volokolamsk Highway 4
125871 Moscow, Russia
ta.silaeva@mail.ru

Abstract: In this paper, the potential of adaptive resonance theory (ART) networks for dependability issues is considered. The basic properties of ART architectures are described, and some strategies are discussed to enable a balanced combination of performance and dependability requirements by these networks.

1 Introduction

The development of IT systems in general is mainly realized under observing constraints as costs and resulting perfomance. Dependability issues usually have to be brought into compliance with these two major requirements. In real-time computing, this need is explicitly shining up in the requirement that the system is not only correctly working, but guarantees to fulfil tasks within given upper time bounds: I.e. functioning alone is not enough, sufficiently quick functioning is necessary.

In many applications, the relation between performance orientation and dependability orientation must not be fixed, as e.g dependability-critical and -uncritical situations may quickly follow each other. Consider e.g. a robot moving in unknown territory. The cautious approach, always to move very slowly, always to perfectly check for all possible dangers might imply too much loss of velocity, to reach the goal location in the required time. So, what we would need is a flexible strategy, which also considers performance issues, while not neglecting some background cautiousness.

Here I would like to propose to consider adaptive resonance theory (ART) [CG88], a subfield of the theory of neural networks, as an interesting offer, albeit not yet investigated in more detail with regard to application in the dependability community. In section 2 the main characteristics of ART networks will shortly be exhibited.

Subsequently, in section 3 some ideas about exploiting the ART structure for dependability issues are discussed.

2 Basic Structure and Properties of ART Networks

Standard neural networks, as e.g. backpropagation networks, work in two different modes: the training phase and the recognition phase. In the training phase, given pairs of an input pattern and an output pattern are offered to the – arbitrarily initialized – network. The network tries to systematically minimize the error in reconstructing the output from the input pattern, by corresponding changes of the weight factors of the neurons, until the error reaches 0 or a sufficiently small value. After the end of the training period, these values are frozen so that no change of the learned experience is possible any more.

Then, in the recognition phase the network is to classify unknown input patterns as similar to certain learned inputs, and, thus, to sufficiently exactly reproduce the corresponding output patterns.

In many applications, however, as e.g. for the movement of autonomous robots in unknown territory, it would be desirable to adapt again the experience of the network to the changing environment. However, simply extending the training to the operational phase of the system causes the tradeoff that this treatment would destroy part of the experience learned during the initial training phase.

Here, ART networks have been proposed as a remedy [CG88]. In the basic ART architecture, the entire recognition process mainly proceeds as follows (see also Fig. 1): The input pattern inp is implemented as a vector of Boolean numbers. It activates the neurons i (i=1,...,m) of the so-called comparison layer F1. The output of the layer neuron i is a function $s_i = f(inp_i)$; often simply the identical reproduction of the input vector inp is assumed: $s_i = inp_i$ (i=1,...m). For the subsequent discussion, we shall also adopt this assumption s=inp.

The vector s is multiplied by the so-called bottom-up matrix B_{ij} the elements of which are real numbers. This produces a real number vector t comprising as components the weighted sums

$$t_j = \sum_{i=1}^{m} B_{ij} * s_i = B_i * s \quad (j=1,...,n \; ; B_i \text{ being the row vector i of matrix } B_{ij})$$

The maximum of these sums is determined:

$$t_k = \max_j t_j$$

Neuron k of the recognition layer F2 is then set to 1; all other neurons of this layer are

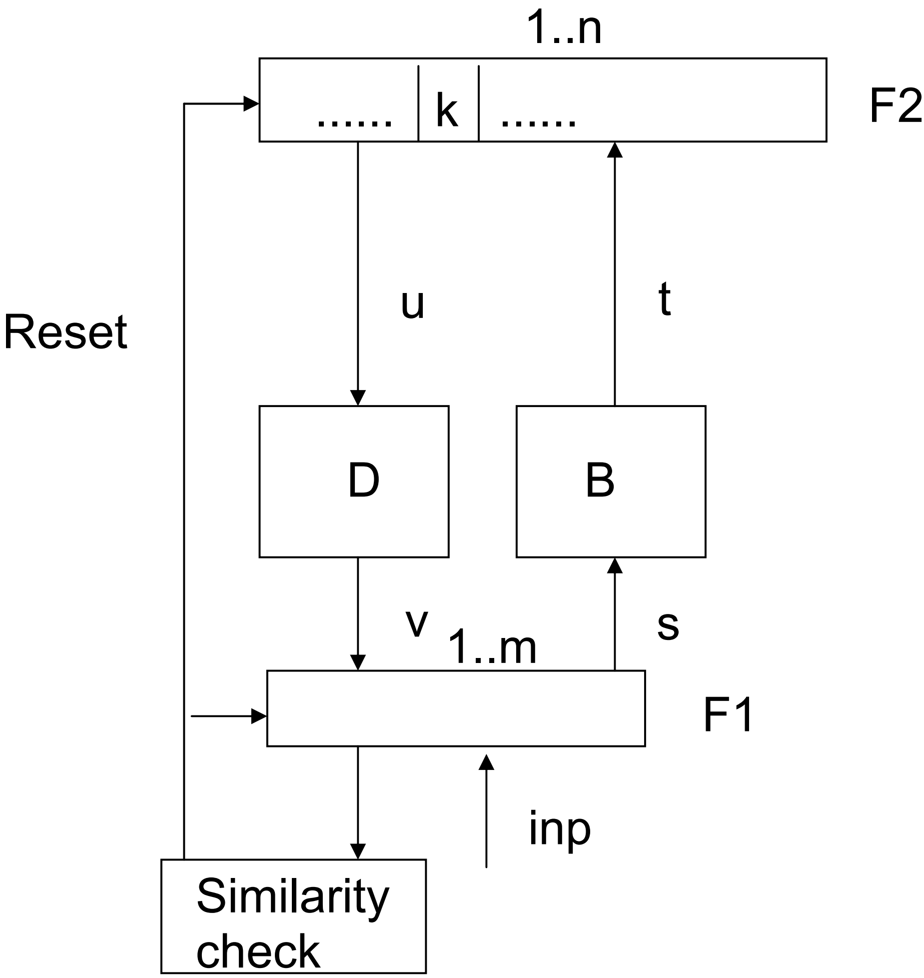

Fig. 1 Basic architecture of ART networks (according to [CG88, Ze 97])
F1 comparison layer, F2 recognition layer, B bottom-up matrix, D top-down matrix, Reset reset line, inp input vector; s, t, u,v generated vectors (see text), k winning neuron of layer F2

set to 0 (neuron k is the „winner neuron“). I.e. the neuron k of F2 represents the class of patterns, to which the input pattern inp is, in this first selection, estimated to belong.

Subsequently, this decision is checked by a control computation. To do so, the vector u of the Boolean values of the recognition layer neurons is multiplied by a second matrix, the so-called top-down matrix D_{ij}, producing a real number vector v:

$$v_i = \sum_{j=1}^{n} D_{ij}\, u_j = d_{ik} * u_k = d_{ik} \quad \text{for } i=1,...,m$$

By AND ing the components of vector v and of the input vector inp of layer F1, a check vector c is formed:

$$c_i = v_i \text{ AND } inp_i \text{ for } i=1,...,m$$

Finally, the similarity of c and input vector inp is compared. This similarity is measured by counting the numbers n_c and n_{inp} , respectively, of 1s in both vectors, and forming their quotient

$$q = n_s/n_e.$$

If q is larger than a previously selected value of a so-called tolerance parameter p, the input inp is assumed to be sufficiently close, „in resonance“, to the column vector D_i of matrix D. If this is not the case, the classification approach is decided to be not fitting. Then the entire recognition process is repeated, with the previous winner neuron k being excluded from the selection process. This causes that the new maximum

$$t_l = \max_{j=1,...,k-1,k+1,...,n)} t_j$$

is etstablished, so that now another neuron l is the winner neuron and the subsequent check for its resonance with the input vector inp is carried out.

This search loop is repeated until a resonant solution is found. If all the actual n class representations of the neural network do not fit to the input vector, an additional neuron n+1 is created in layer F2, and is set to 1. Correspondingly, the number of pattern classes distinguishable by the ART network, increases by one. So, the network is able to respond to the appearance of unidentified patterns, by the creation of new classes.

After the search process for the given input vector inp has been completed, the elements of the matrices B and D are updated. Updating of bottom-up matrix B is done by changing the row vector which had, in the procedure described previously, produced the maximum sum; this row vector is torn towards the input vector inp. All other rows of matrix elements remain unchanged. Updating of the top-down matrix D is done by component-wise ANDing by the vector s. For the mathematics of the update formulas, and also the influence of some additional so-called gain factors see [CG88, Ze97] which provide a comprehensive discussion of these details.

Also, in this position paper, for ease of understanding we confine the discussion to considering the basic ART architecture ART-1 described in this section; upgrading modifications of this architecture allow e.g. the use of real number input vectors [CG91a, CG91b] or combinations with fuzzy logic [CG91c].

3 Application of ART Networks for the Dependability of Systems

As the main advantage of ART networks their „plasticity“ is claimed, i.e. the ability to integrate new knowledge into the network without destroying old one. Moreover, this is done in a balanced way where still the attempted changes are checked aginst the memorized knowledge, stored in the top-down matrix. So, the network emulates, to a certain degree, the cooperation between short term storage and long term storage known from biological systems: Actual on-the-fly experience or situation estimation might flow into the current input vector inp, and then can be integrated into memory under the surveillance of the long-term knowledge stored in the top-down matrix D.

This scheme might be applied to many applications, especially of autonomous systems as, e.g., robots: In easily-manageable landscapes like plains, the robot without risk could be allowed to move quicker. On the other hand, a situation where none of the stored strategies (each represented by one neuron of the recognition layer) is acceptable, would indicate an extraordinary situation where a high degree of cautiousness has to be paid while exploring this situation and its potential consequences; i.e. where the robot should be in a „high alert“ state.

So, the use of an ART network for control would enable a a three-stage scheme of adaptivity:

- on the one hand, we have individual strategies (each represented by one neuron of the recognition layer) which have learned to adapt to a class of input patterns;

- the network provides switching from one strategy to another one in a quite simple way;

- moreover, for extraordinary situations, considering a new, additional strategy is possible.

We have pointed out that the ART network very flexibly enables the implementation of new distiguishable decision classses. In the basic ART approach, however, it is left open, how, as response to the input, a certain output pattern o, e.g. comprising control signals, is to be generated.

Here we propose to adopt the solution used e.g. in counter-propagation networks [He88], to utilize, after having completed the search for the winner neuron, the recognition layer contents u, also for a matrix vector multiplication with an additional matrix, called here the output pattern matrix OP (see Fig. 2):

$$o_i = \sum_{j=1}^{n} OP_{ij} * u_j = OP_{ik} \ (i=1, ..., n^*),$$

k being the index of the winning neuron in F2, and n* the number of rows in matrix OP (i.e. n* is the width of the control pattern output).

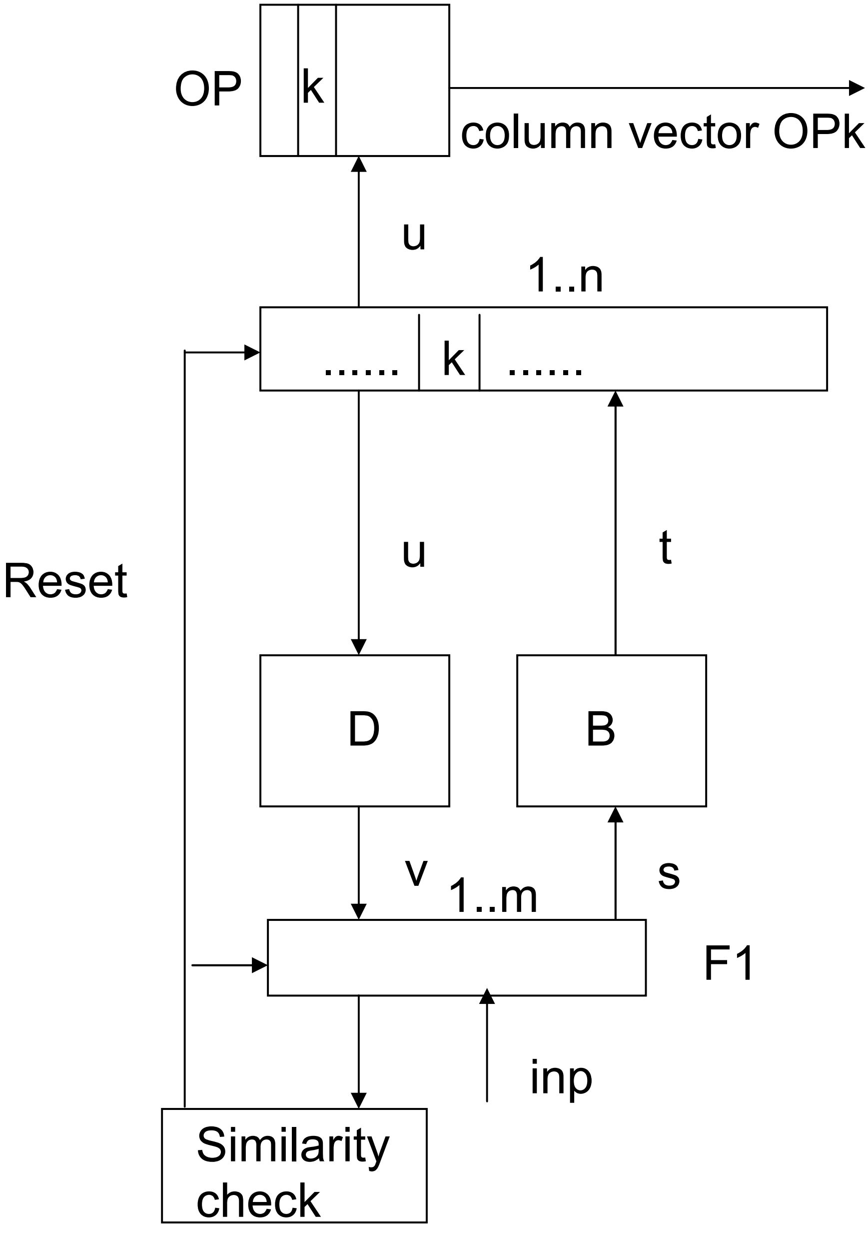

Fig. 2 Extension by an associative memory
F1 comparison layer, F2 recognition layer, B bottom-up matrix, D top-down matrix, Reset reset line, inp input vector; s, t, u,v generated vectors (see text); k winning neuron of layer F2. Additionally the matrix–vector multiplication OP * u produces as result the column vector OP_k of matrix OP, thus providing output of a data pattern associated to the winner neuron k of layer F2. That means the result vector o is just the column vector

OP_k of matrix OP. So, depending on the contents of the input vector inp (not via a memory address !) the output pattern is accessed from the „memory" OP, i.e. OP functions like an associative memory.

To conclude, the ART network approach provides a promising strategy to implement control schemes especially for autonomous systems, where dependability issues as well as performance considerations can flexibly be combined in a balanced way. Of course, beyond the general concept outlined here, several detail problem areas are still open for further investigation, e.g

- efficient definition of the traning procedure for the matrix OP, to produce the contents of the associative memory;

- extending or modifying the strategy sketched here, to cope with the modifications of the basic ART network, mentioned in the literature.

For tackling such problems, we would welcome colleagues interested in this field, for some future cooperation.

4 Conclusion

In this paper, the potential of adaptive resonance theory (ART) networks for dependability issues has been discussed. The basic properties of ART architectures are outlined, and some strategies are shown to enable a balanced combination of performance and dependability requirements by these networks.

5 References

[CG88] Carpenter, G.A.; Grossberg, S.: The Art of Adaptive Pattern Recognition by a Self-Organizing Neural Network, Computer, March 1988, pp. 77-88

[CG91a] Carpenter, G.A.; Grossberg, S.; Rosen, D.B.: ART2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition, Neural Networks, Vol. 4, 1991, pp. 493-504

[CG91b] Carpenter, G.A.; Grossberg, S.; Reynolds, J.H.: ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network, Neural Networks, Vol. 4, 1991, pp. 565-588

[CG91c] Carpenter, G.A.; Grossberg, S.; Rosen, D.B.: Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System, Neural Networks, Vol. 4, 1991, pp. 759-771

[He88] Hecht-Nielsen, R.: Applications of Counterpropagation Networks, Neural Networks, Vol. 1, 1988, February, pp. 131-139

[Ze97] Zell, A.: Simulation neuronaler Netzwerke, R. Oldenbourg Verlag München 1997