

## A combined network architecture using ART2 and back propagation for adaptive estimation of dynamical processes

EINAR SØRHEIM†

*Keywords: System identification, nonlinear systems, adaptive control, artificial neural networks, back propagation, ART2*

A neural network architecture called ART2/BP is proposed. The goal has been to construct an artificial neural network that learns incrementally an unknown mapping, and is motivated by the instability found in back propagation (BP) networks: after first learning pattern A and then pattern B, a BP network often has completely 'forgotten' pattern A. A network using both supervised and unsupervised training is proposed, consisting of a combination of ART2 and BP. ART2 is used to build and focus a supervised backpropagation network consisting of many small subnetworks each specialized on a particular domain of the input space. The ART2/BP network has the advantage of being able to dynamically expand itself in response to input patterns containing new information. Simulation results show that the ART2/BP network outperforms a classical maximum likelihood method for the estimation of a discrete dynamic and nonlinear transfer function.

### 1. Introduction

Most current neural network architectures such as back propagation require a cyclic presentation of the entire training set to converge. They are thus not very well suited for adaptive estimation tasks where the training vectors arrive one by one, and where the network may never see the same training vector twice. The ART2/BP network system is an attempt to construct a network that works well on these problems.

The self-organizing ART2-part of the system is used to control small BP-networks that are trained to specialize on domains of the input space. As such a BP-network covers a small part of the input space, a relatively small network should be sufficient. When a new training vector arrives ART2/BP trains only one of the BP-networks, enabling the training time at each presentation to be small (assuming that small networks with small training sets converge faster than large networks on large training sets).

Main features of our ART2/BP are:

- implements incremental supervised learning
- dynamically self-expanding
- learning of a novel training pattern does not wash away memory of previous training patterns
- short convergence time for learning a new pattern

---

Received 20 February 1991.

† Institute for Energy Technology, Systems Technology Department, Kjeller, Norway.

This paper was presented at the 24th IEEE Conference on Systems Science, Hawaii, 8-11 January 1991, and is reprinted with the permission of IEEE. © IEEE.

## 2. Background

Adaptive estimation of nonlinear functions requires some basic features of the estimation algorithm.

### 2.1. Incremental learning

The input/output pairs arrive to the estimation machine one by one. No representative training set will be available. One does not have any control on the distribution of the arriving input/output pairs. By accumulating the input/output pairs into a training set and rerunning the training procedure at every arrival of a new input/output pair, one could use a conventional method. Obvious disadvantages would be

huge learning required as the size of the training set increases, an upper limit,  $N$ , on the number of elements in the training set will have to be set. The training set will then be a gliding horizon of the  $N$  last input/output pairs, and information prior to the  $N$  last input/output pairs will be lost.

### 2.2. Plasticity

Learning of a new input/output pair should not wash away the memory of previously learned non-conflicting input/output pairs. With most existing feedforward supervised nets this is hard to accomplish, though some efforts have been made (Ottwell 1990). One could implement e.g. a back propagation network with very slow learning on each input/output pair, but this might lead to a never converging network and the number of input/output pairs needed will certainly be huge. It would also adapt very slowly to changes in the input/output pairs.

Some networks, like the ART-family and RCN (Ryan 1988) are plastic but they are self-organizing, not supervised.

To summarize, a supervised network is needed that learns incrementally the mapping of an unknown system and that can be used to predict future outputs. The system in question maps analog vectors to analog vectors.

## 3. Combined architecture

In the proposed network architecture an ART2 network controls a BP network, see Fig. 1.

The BP-network consists of many relatively small subnetworks where the subnets are specialized on one particular domain of the input space. ART2 controls how the input space is divided among the subnets and the total amount of subnets needed.

The ART2 network analyzes the input part of the input/output pairs as they arrive to the system. For a given input pattern  $i_x$ , ART2 finds the category  $C_x$  which has the closest resemblance to  $i_x$ . If this resemblance is good enough, (as decided by the reset assembly),  $i_x$  is of category  $C_x$  and the LTM-weights of  $C_x$  are updated. The BP-subnetwork  $BP_x$ , connected to  $C_x$ , is as a consequence activated, and relearning of  $BP_x$  is done. The learning set consists of a 'representative' set of the neighbouring subnets

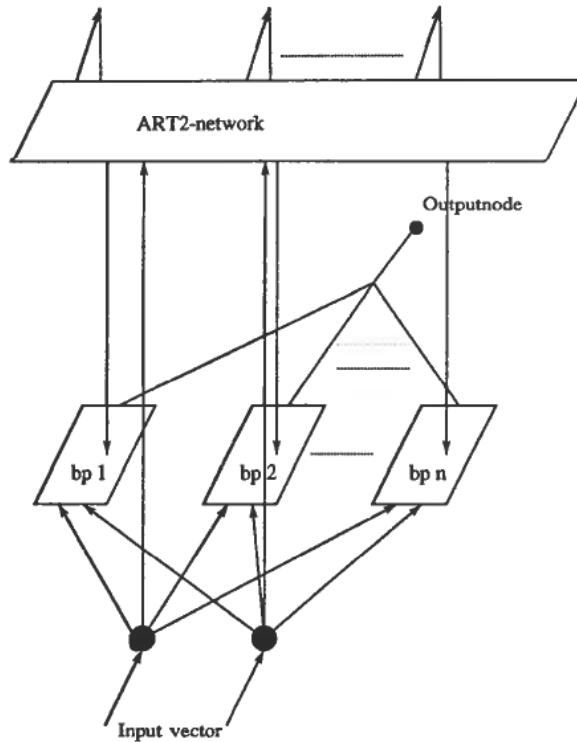


Figure 1. ART/BP network architecture.

patterns and a small number of the previous patterns belonging to category  $C_x$ . To summarize the algorithm goes as follows:

1. Send input vector to ART2 network.
2. ART2 classification.
3. If in learning mode adjust ART2 LTM weights of the winning node.
4. Send input to the back propagation network connected to the winning ART2 node.
5. If in learning mode:
  - find a representative training set.
  - do epoch learning on training set.
- Otherwise
  - compute output of the selected back propagation network.

6. Go to 1. for new input vector.

The ART2/BP neural network can be used for adaptive estimation of nonlinear dynamic processes. The mapping to be estimated then is

$$\begin{aligned}
 y(t + \delta t) &= f(u(t), y(t)) \\
 u(t) &\in \mathfrak{R}^m \\
 y(t) &\in \mathfrak{R}^n
 \end{aligned}
 \tag{1}$$

The input/output pairs will be  $io = [u(t), y(t), y(t + \delta t)]$ , denote the input part of  $io$ :  $i = [u(t), y(t)]$  and the output part of  $io$ :  $o = y(t + \delta t)$ .

#### 4. ART2 modified

##### 4.1. Introduction to ART2

ART2 was developed by Carpenter and Grossberg (1987, 1988). ART2 categorizes arbitrary sequences of analog input patterns [1], and the categories can be of arbitrary coarseness. An ART2 system consists of two interconnected layers or fields called F1 and F2, see Fig. 2.

F1 is a feature representation field while F2 is a category representation field. The number of nodes in the F1 layer is given by the dimension of the input vector. In the F2 layer however a node represents a category given by its top-down and bottom-up weights. These weights are the long term memory states (LTM) of the network, and learning is done by modifying the LTM weights. Nodes in F2 will dynamically be created when patterns of new categories arrive. The coarseness of the categories is given by a vigilance parameter  $\rho$ , and might also be adaptive. In the F1 layer a number of operations on the input pattern are performed, such as normalization, noise suppression and feature extraction. The pattern code in the F1 layer is transmitted to the F2 layer nodes through the bottom-up weights. The F2 node with bottom-up weights that best match the F1 pattern code is declared the winner and its top-down weights are transmitted back to the F1 layer.

One can thus think of the bottom-up weights as being a recognition code and the top-down weights as being expectation code. When the resonance between F1 and F2 has stabilized the reset assembly compares the F1 pattern code and the input vector. If

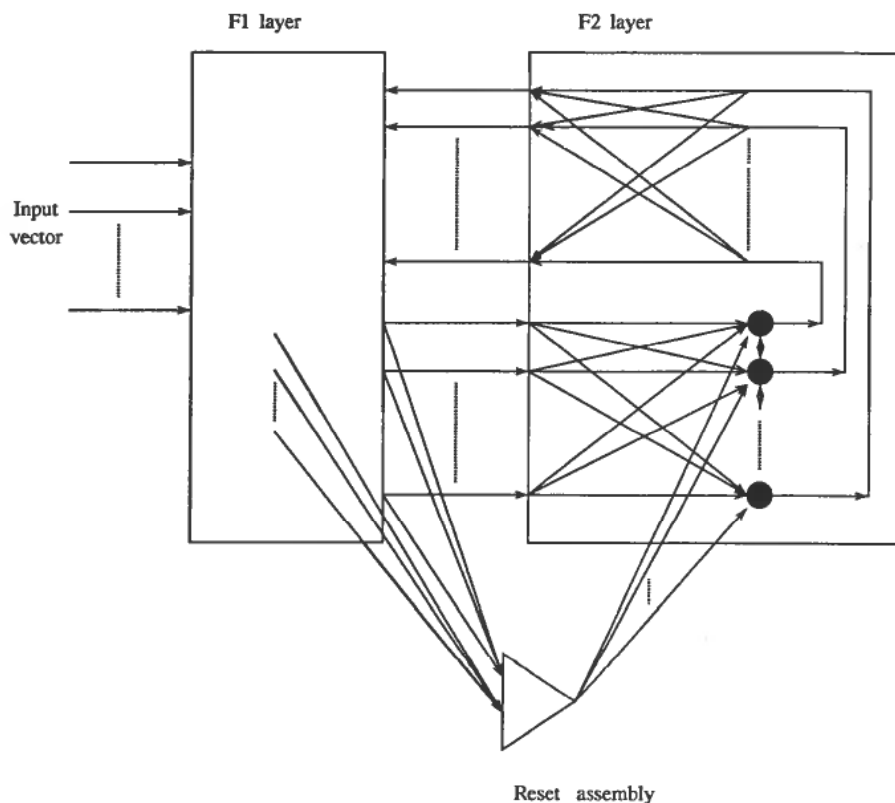


Figure 2. ART2 network architecture.

the resemblance is too low, (as decided by the vigilance parameter  $\rho$ ), the winning F2-node is inhibited and a new F2 node representing a new category is created.

Learning, (that is adaptation of the LTM weights) evolves according to a differential equation and is performed only on the winning node. The interval the input vector is available decides to what degree learning converges to the F1 pattern code.

In standard ART2 the winning node in the F2 layer is the node with the largest inner product of its weights and the F1 pattern code. Reset is done by using a clever angle measure between the input vector and the F1 pattern code. For a more detailed description of ART2, see Carpenter and Grossberg (1987).

#### 4.2. Modification

In the standard ART2-algorithm input vectors (patterns) are normalised. For this application it is not desired to classify parallel vectors of different magnitude as belonging to the same category. By adding an extra element to the input vector where this element is simply

$$i_{n+1} = \|i\|^2 \tag{2}$$

the new input vector becomes

$$i' = [i', \|i\|^2] \tag{3}$$

From a scaled vector of  $i: x = a \cdot i$  the original vector  $i$  could easily be found as:

$$i_i = x_i \cdot \frac{x_{n+1}}{\|x\|^2} \tag{4}$$

$$x = [x_1, x_2, \dots, x_n]$$

and by using the augmented  $i$  as the input to ART2 instead of  $i$  one can at any point in F1 and F2 generate the corresponding non-normalized vector. The F2 node competition is modified so that the node having bottom-up LTM weights with the smallest distance (distance being the euclidean norm) to the F1 layer pattern code wins the competition. The distance  $d_j$  of F2 node  $J$  is given by:

$$d_j = \|p - z_j\| \tag{5}$$

- $\| \|$  being the  $l_2$ -norm
- $p$  F1 pattern code
- $z_j$  bottom-up LTM weights of F2 node  $J$

Reset is done by calculating the distance  $d$  between F1 layer pattern code  $p$  and  $i$ :

$$d = \|p - i\| \tag{6}$$

and comparing it to a largest acceptable bound  $\rho$ . If  $d > \rho$  the winning node is inhibited and a new node will be created. If  $d \leq \rho$  LTM-patterns of the winning node  $J$  are modified (learning).

#### 5. Backpropagation network

The backpropagation network used in this work is of the standard feedforward type, see Rumelhart, Hinton and Williams (1986). The number of hidden layers and nodes should be kept low in the subnetworks, for the problems in our simulations we used 1 hidden layer with 2 nodes. The learning algorithm used is a modified error propagation where the learning rate and momentum is adjusted according to the sign of the inner product between the present and immediate past gradient vectors.

## 6. Learning

Learning in ART2/BP is a two stage process. First the input pattern is sent to the ART2 network for categorizing and learning. ART2 will then activate the BP subnetwork that is a local expert on patterns of the same category as the input pattern, and learning of this subnetwork will occur.

The BP-network connected to the winning recognition node is then activated. If the F2 node is new, a subnetwork is created and connected. A training set that is representative for the domain of the input space has to be found. A small number of the last categorized input/output pairs will be allocated to its corresponding subnet to provide a part of the training set. Denote such a set as  $L_{IO_C}$ , ( $C$  being the category). Define the location of F2 node  $J$  to be its bottom-up weights  $z_j$ . Let the current input  $i_x$  define an origin, then find the F2 nodes closest to origin in each  $n$ -ant of the input space. Call this set of nodes  $N_x$  and the set of last input/output pairs stored in these nodes  $N_{IO_x}$ . The training set is then chosen to be:

$$T_x = N_{IO_x} \cup L_{IO_x}$$

Before training, the elements in  $T_x$  are scaled to increase the accuracy and to accelerate learning. BP-learning is then performed, the stopping criteria being a fixed error term or a maximum number of iterations.

## 7. Estimation

In estimation mode learning in the network is turned off. Given an input the network will produce an output that hopefully will be close to the output of the real system.

The ART2-network selects a winning node in the same way as described before but now the reset assembly is not activated. Then the input is fed to the corresponding BP subnetwork and its output is used as an estimate of the original functions output.

Because each subnetwork is scaled to cover the domain of the input space made up by the complex hull  $Co(T_x)$  of its training set  $T_x$ , the entire ART2/BP network will cover the complex hull  $Co(T) \subset \mathfrak{R}^{n+m}$  where:

$$T = \{\text{set of all previous } i \text{ is used to train the network}\}$$

Good estimation/prediction can thus be expected if  $i \in Co(T)$ .

This means that if the input vector  $i$  lies in a domain of the input space that has not been previously explored by the elements in the training set, the network will generalize poorly.

## 8. Example

The ART2/BP network has been used to estimate a dynamic model of a tank filled with liquid. The liquid level is sampled every  $\delta t$  time interval and the ART2/BP network is used to estimate the discrete dynamic nonlinear transfer function of the liquid level as a function of inlet liquid flow and previous liquid level. That is we want to find a good estimate  $f(.,.)$  of:

$$\begin{aligned} y(t + \delta t) &= f(u(t), y(t)) \\ u(t) &= \text{inlet liquid flow at time } t \\ y(t) &= \text{liquid level at time } t \end{aligned} \quad (7)$$

To increase the nonlinearities function, the area of the tank varies with a step function of the liquid level. The BP subnetworks have 2 input nodes, 1 hidden layer with 2 neurons and a single neuron output layer. In the simulations  $\rho = 0.04$  and the last three categorized input/output pairs are stored at every subnetwork. As the input space is 2-dimensional giving 4 neighbouring nodes the maximum size of the training set 7 input/output pairs. After a learning period of 1000 samples with random inlet flow, three test cases are run with the network in estimation mode. The network had then formed about 140 categories. The same set of simulation data is also run through an offline maximum likelihood method to estimate a linear ARMA model of the plant, see Ljung and Söderström (1983). Figures 3, 4, 5 show the simulation results of the three test cases where:

- samples 1–100: random input flow.
- samples 101–200: constant input flow at a low level.
- samples 201–300: constant input flow at a high level.

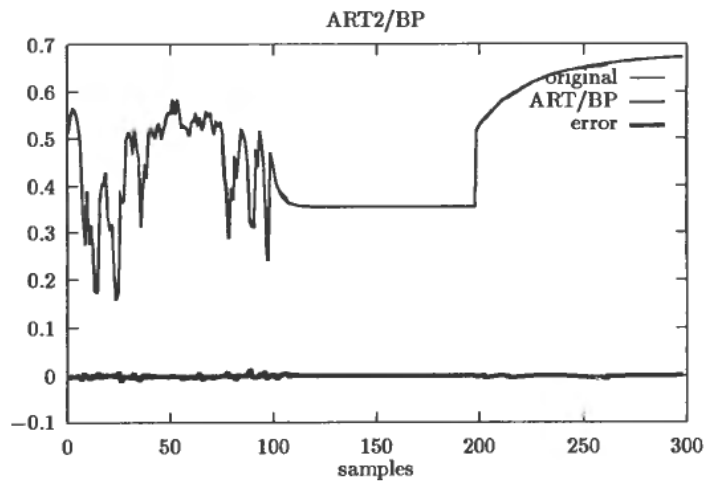


Figure 3. ART2/BP one step ahead prediction compared with original model.

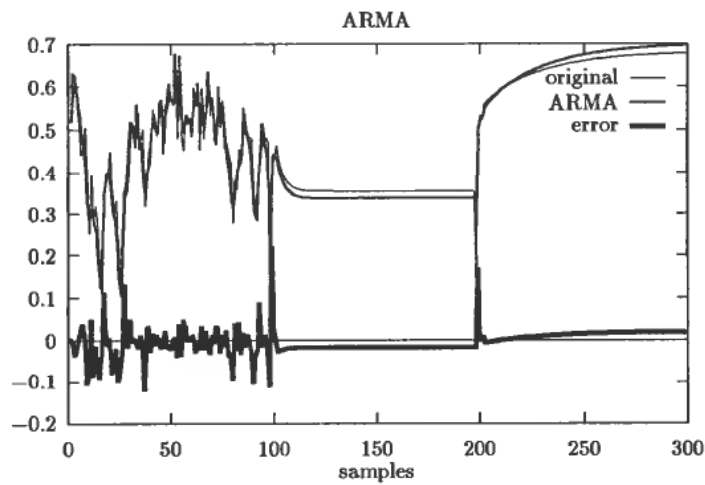


Figure 4. ARMA 2nd order one step ahead prediction compared with original model.

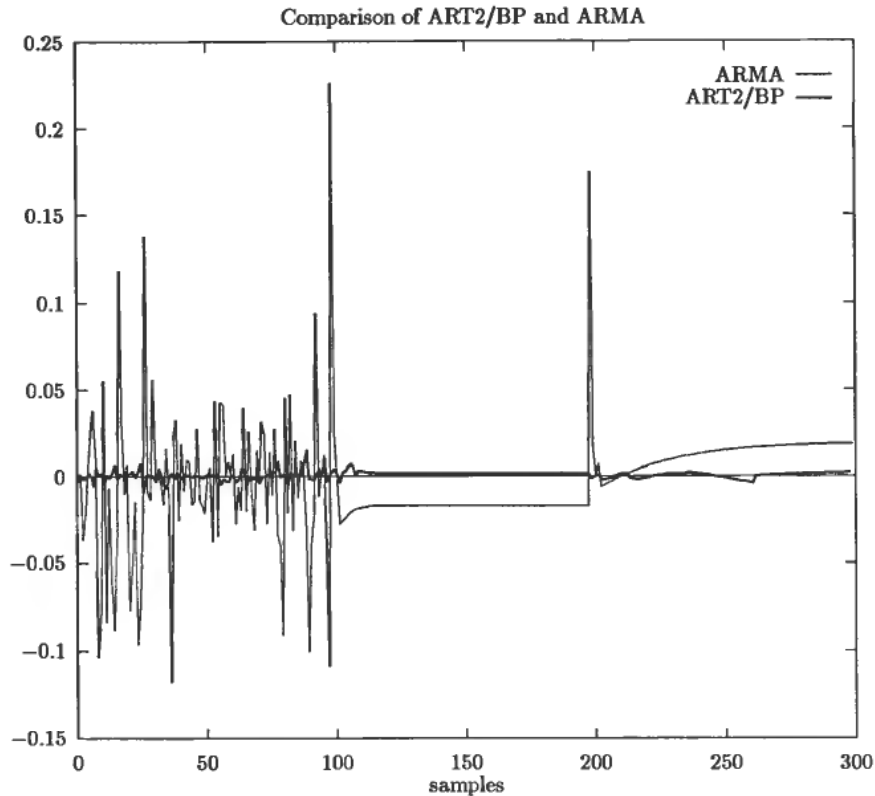


Figure 5. One step ahead prediction errors for ARMA and ART/BP.

Figure 3 shows that the ART2/BP networks estimate has no significant deviation from that of the original model. In Fig. 4 one can see that the 2nd order ARMA model does not follow the reference model nearly as good as ART2/BP. This clearly seen in Fig. 5, where the estimation errors of the two methods are compared. In the first 100 samples with stochastic input flow, the estimation error variance of the ART/2BP network is roughly a factor 10 less than that of the ARMA-model. The performance of ART2/BP is also significantly better for the constant input flow cases, here the ARMA model has an error of  $\sim 0.02$  while the ART2/BP-error is  $\sim 0.002$ .

The overall improvement in estimation error is a reduction of roughly 0.1. Also keep in mind that ART2/BP is compared to an offline maximum likelihood method while ART2/BP clearly is an online method. The online version of the maximum likelihood would most probably have given a worse performance than the offline version.

### 9. Conclusions/comments

The proposed ART2/BP neural network architecture offers some unique features compared to backpropagation. It provides incremental learning and can be applied to truly adaptive estimation tasks.

In our example it also outperforms a classical maximum likelihood for the estimation of a discrete dynamic nonlinear transfer function.



Future work will be the investigation of ART2/BP's properties for multistep-ahead prediction of dynamic nonlinear functions, and embedding ART2/BP in a neural adaptive controller.

#### ACKNOWLEDGMENT

Special thanks to Steve Lehar at Boston University for providing me with his ART2 simulation program. It proved to be crucial for getting a quick start on ART2 and understanding the concept.

#### REFERENCES

- CARPENTER, G. A. and GROSSBERG, S. (1987). ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, **26**, 4919–4930.
- CARPENTER, G. A. and GROSSBERG, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, **21**, 77–88.
- LJUNG, L. and SØDERSTRØM (1983). *Theory and Practice of Recursive Identification* (The MIT press, Cambridge, MA).
- RUMELHART, D. E., HINTON, G. E., and WILLIAMS, R. J. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1 (The MIT Press, Cambridge, MA).
- RYAN, T. W. (1988). The resonance correlation network. Proceedings IJNN88.