ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Model Learning from Weights by Adaptive Enhanced Probabilistic Convergent Network

P. Lorrentz[1], W. G. J. Howells[2], and K. D. McDonald-Maier[3]

[1]Department of Electronics, University of Kent, Canterbury, Kent, CT2 7NT, UK Tel. 0044(0)7813089916, Fax: +441227 456084. e-mail: plorrentz@gmail.com
[2]Department of Electronics, University of Kent, Canterbury, Kent, CT2 7NT, UK .Tel: +44 122723724. Fax: +441227 456084, e-mail: W.G.Howells@kent.ac.uk
[3]Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK. Tel. 0044(0)1206874376, e-mail: kdm@essex.ac.uk

Abstract. Current weightless classifiers require historical data to model a system and make prediction about a system successfully. Historical data either is not always available or does not take a recent system modification into consideration. For this reason an adaptive filter is designed, which when employed with a weightless classifier enables system model, difficult to characterise system model, and system output prediction, successfully.

Results of experiments performed show that the fusion of an adaptive filter and a weightless classifier is more beneficial than the filter or the classifier utilised singly, and that no speed advantage is observed.

1. Introduction

Current research efforts have modelled and characterise unknown or difficult to characterise systems to some success. Researches into system models by employment of weightless neural classifiers are rare, most especially if the structural and internal operations of the system are unknown or difficult to characterise. This motivates the aims and objectives of this paper which are to explore the possibility of modelling an unknown and difficult to characterise systems by combining adaptive filters, the Least-Mean-Square (LSM) algorithm, with an adaptive weightless neural classifier, the Enhanced Probabilistic Convergent Neural (EPCN) network. This research utilises adaptive filter, instead of system history, to model systems. The model coefficients may then be presented to the classifier for learning, making prediction, system identification [1], control, etc. This method, in real-time, requires little or no prior history about the system to model.

This paper is organised as follows. Section 2 explains the proposed LMS-EPCN system. The system is explored in the experiment section 3. Results and analysis are reported in section 4.

2. The EPCN-LMS System

2.1 The Adaptive Filter

An adaptive filter consists of two parts, namely; a digital filter with adjustable coefficient, and an adaptive algorithm which is employed in the adjustment and modification of the coefficient of the filter. In this report, an adaptive filter along with Widro-Hopf Least Mean Square (LMS) algorithm [2] provides a fan-in to a weightless classifier. The adaptive filter is produced from the input data **X**, which output a coefficient weight matrix **W**. Any system may be modelled via autoregressive model by sets of weight coefficients. The mean square error (MSE) e^2, that is the square of error e_k signal (see figure 1), is defined by

$$E = \frac{1}{N}\sum_{n=1}^{N} e^2(n) = \frac{1}{N}\sum_{n=1}^{N} (x(n) + \sum_{k=1}^{p} a(k)x(n-k))^2 \quad \text{----------- (1);}$$

where *e(n)* = error term, *x(n-k)* = input data, and *a(k)* = system parameter. Differentiating equation (1) and setting the differential to zero,

$$R_{xx}(k-j)a(k) = -R_{xx}(k) \quad \text{--------------- (2);}$$

after some calculations. Equation (1) to (2) is schematized in figure 1. In equation (2), $\boldsymbol{R_{xx}}$ = correlation coefficient of input data. The weight matrix **W** is initialized to $\boldsymbol{R_{xx}}$ instead of arbitrary random value as in normal LMS algorithm. This implies:

- It is a valid training set for the classifier, since each such matrix is a class correlation of input data.
- Where a learning algorithm of a classifier makes a one-pass over input data, equation (2) produces an initial better training set rather than arbitrary random values.
- Initialization of weights to arbitrary random values may also access neurons at random, illegal memory access to memory locations, and thus not advisable for weightless classifiers.

Differentiating e^2 and setting differentials to zero, after some calculations, the weight modification equation, W_{k+1} , is given by;

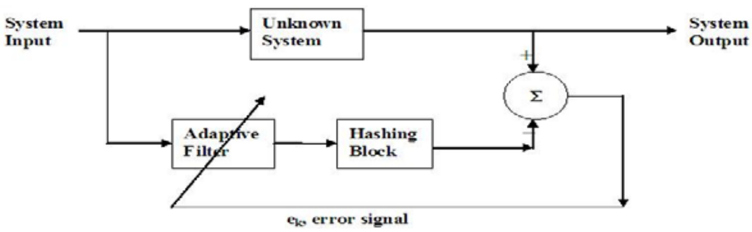

Fig. 1: System modelling by LMS

$$W_{k+1} = W_k + 2\mu e_k X_k \quad \text{----------- (3)}$$

where $e_k = y_k - W_k^T X_k$; $\mu := 0\langle\mu\langle\frac{1}{\lambda_{\max}}$; and $\lambda_{\max}$ = maximum eigenvalue W_k.

Equation (3) is known as Widro-Hopf LMS algorithm [2]. The weight matrix **W** is afterwards modified according to equation (3). To the LMS algorithm output is attached a neural classifier which is explained in the next section.

2.2 An FPGA-based EPCN Network

A *weightless classifier* is a classifier whose learning and recognition algorithm utilises Boolean logic in modifying memories within its layers. The *weightless classifier* employed in this research is known as Enhanced Probabilistic Convergent Network (EPCN) [3]. The hardware architecture of EPCN is outlined in Figure 2. The addresses for RAM locations within a neuron are formed from an input pattern by a hashing function in the Hashing block. The hashing function is derived from XOR and Maximum-length Shift-register code (MLSR) [4]. The MLSR make use of a parity polynomials h(p) given by:

$$h(p) = p^k + h_{k-1}p^{k-1} + \ldots + h_0p^0 \quad \text{-------- (4)}$$

Hashing occurs prior to learning, and subsequent recognition. Learning in the train-block (which consists of pre-group layers which in turn consist of neurons) writes to RAM-locations, independently, the frequency of asses by a class to that location.

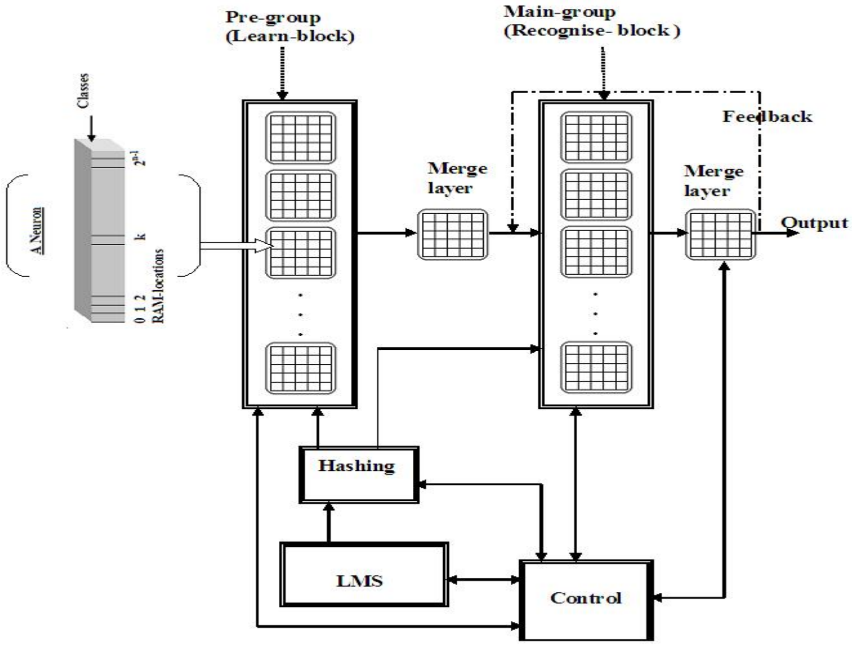

Fig. 2: A simplified schematic of Hardware Architecture of EPCN network.

When the sum of frequency is normalised, the result is a probability of occurrence. Thus the decision (output) of EPCN in a prediction task is a probability of occurrence of classes- the reasons for the term "probabilistic" in the naming of EPCN. The prediction occurs in the main-group which consists of layers which in turn consist of neurons. Reading from and/or writing to the neurons are mediated by connectivity, the size of which is determined by a system parameter known as *n-tuple*. Values in these neurons are averaged and adjusted by *division. Division* is a system parameter, it is a number employed to adjust or/and average the values in neurons. The tuple size (known as *n-tuple*) is required during learning and subsequent recognition to determine the number of address lines. Learning and prediction are both initialised and stopped by the Control block respectively. The control block also checks the output block for stability or oscillation of values. There is an *output feedback* (see figure 2) iteratively to the main-group to obtain convergence – thus the naming "convergence" in EPCN. The procedure is stopped if its value is stable or after a few number of iteration if the value does not stabilise. Since the output of EPCN is expressed in probability values, it is said to *converge probabilistically* to the solution.

2.3 The LMS-EPCN System

The LMS-EPCN consists of LMS in parallel, and the output coefficients produced are combined by the hashing function of the EPCN. The LMS employed for system model is required to model an unknown system. The implementation of LMS-EPCN in FPGA removes the requirement of explicit external binary input data to the EPCN weightless classifier (this is advantageous to weightless classifiers in general). The EPCN will normally access all data supplied for learning which may be small or large [3]. In the proposed system, the coefficients are first formed by LMS according to equation (2), and subsequent training data for a class modifies these coefficients according to Widro-Hopf algorithm (equation (3)). The filter coefficients are system models which are learned by the EPCN part of the LMS-EPCN system. The LMS-EPCN fusion is able to models and learns about a system in real-time and thus suited to portable system, embedded systems, and mobile/robotic systems.

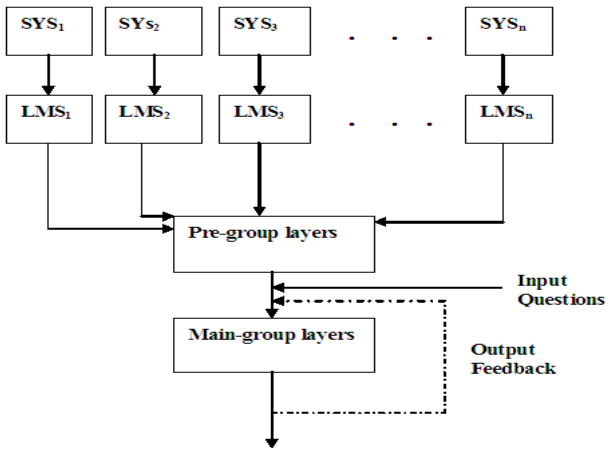

Fig. 3: Parallel LMS_i, each attached to an external system, SYS_i, to model. All LMS_i (i = 1, 2, 3,...,n) outputs are used in training EPCN.

The proposed system is easily implemented wholly on-chip, thereby using the parallel architectural possibility inherent in hardware design to advantage. A schematic of the LMS-EPCN system is shown in figure 3. Specifically, the proposed system is implemented on Virtex II pro, and the reconfigurable potential of the Virtex II pro permits variation in the number of LMS to use. In the experimentation section (next section), SYS_1 represents O-rings (section 3.1) of experiment section, while SYS_2 represents the landing scenario (see section 3.2) of the space shuttle.

3. Experiments

Prediction problems could be grouped into two categories: those which the results are known a priori, and those of unknown result. Prediction problems addressed in this section concerns the taking off and landing of space vehicles. Specifically, the space shuttle challenger is considered in this regard, and both types of prediction problems are examined relative to the space shuttle challenger.

3.1 Space Shuttle Take-off (O-rings)

The first experiment is a prediction of how many O-rings may fail on shuttle lift-offs at temperatures outside the range of that given in the database. The database source is: http://archive.ics.uci.edu/ml [5]. The existing data concerns USA space shuttle challenger which explodes due to failure of O-rings. The temperature and pressure of previous flights were input for LMS to produce model weights from which EPCN learns. During the prediction stage,

	No. Of O-rings to fail	Temperature/F	Pressure/psi
1	?	31	50
2	?	31	100
3	?	31	200

Table 1: Questions for LMS-EPCN fusion to answer

the following question in Table 1 is presented to the LMS-EPCN system. That is, given certain temperature and pressure, how many O-rings would fail?

3.2 Space Shuttle Landing

The aim of this experiment is to predict whether manual landing of a space shuttle is better than automatic landing due to weather conditions and the state of the space shuttle. The database from [5]: http://archive.ics.uci.edu/ml. The database is presented to LMS part of the system during which a model (weights) is formed. The EPCN is trained on this model. During prediction stage, the questions posed to the system are:
1) Auto-landing? yes/no; 2) Manual-landing? yes/no. Prediction of a known result (example of which is the landing prediction of space shuttle) is significant and applies to a communication channel whereby a result could not be transmitted from one end to another. The result could then be predicted at the other end given relevant information.

4. Results and Analysis

Table 2 is obtained as answer to the questions of table 1. Each row of the table should be read in probability terms since the classifier is a probabilistic network.

	No. Of O-rings to fail	Temperature/F	Pressure/psi
1	2	31	50
2	2	31	100
3	3	31	200

Table 2: Prediction of the numbers of O-rings due to fail

For example at row 1, two rings are most likely to fail at lift-off temperature of 31^0F and pressure 50psi. The LMS-EPCN fusion system also predicts 4 O-rings failure (most likely) at absolute zero at pressure 200psi. Predictions concerning O-rings (table 2 inclusive) are also supported by material science and qualitative analysis. Table 3 is obtained as a result of performing the landing experiment of the space shuttle. In column 1 of table 3, it states that it is better to land manually than to land automatically.

Landing Instances	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Answer: auto=1; manual=2	2	1	1	1	1	1	2	1	2	2	2	2	2	2	2

Table 3: Landing prediction of Space Shuttle.

Neural Networks	Platform/Circuit size	Time/Cycles	
EPCN	Virtex-II Pro.; 2492 SLICEs used	7.5e-8 sec	[3]
LMS-EPCN	Virtex-II Pro.; 2602 SLICEs used	7.5e-8 sec	
Back-propagation	Xilinx XC4005XL-PQ; 196 CLB used.	1879 ns	[7]
Space Vector Modulation (on Neural network)	Xilinx XCV50hq240 FPGA, max. 1096 SLICEs	max. 149 cycles	[8]
Support Vector Machine	Spartan3 XC3s50pq208-5; max 1244SLICEs	-	[9]
Radial Basis Function (on pressure sensor)	Virtex-II FPGA; 14334 SLICEs used	206 ms	[10]

Table 4: *Resource* comparison of LMS-EPCN with other Neural networks.

Table 4 shows resource utilisation and comparison of LMS-EPCN with other neural networks. Though the LMS-EPCN consumes more SLICEs (at 2602) than a single EPCN (at 2492), speed advantage of a single EPCN (at 7.5e-8 s) is not observed. The LMS-EPCN fusion exceeds Radial Basis Function in terms of speed performance.

References

[1] Raja M.A.Z. and Ijaz M.Q., A modified Least Mean Square algorithm Using Fractional Derivative and its Application to System Identification, European Journal of Scientific Research ISSN 1450-216 Vol. 35 No. 1pp. 14-21, 2009.

[2] S. Haykins and B. Widrow, Least-Mean-Square adaptive Filters, A John Wiley & Sons, Inc. Publication, 2003.

[3] Lorrentz P., Howells W.G.J, and McDonald-Maier K.D: An FPGA based adaptive weightless Neural Network Hardware, IEEE, NASA/ESA 2008, AHS-2008, 2008.

[4] John G. Proakis: Digital Communication, 4th Edition, McGraw Hill Int. Edition, 2001.

[5] Databases available at http://archive.ics.uci.edu/ml/datasets.html, Accessed date 13/01/2009.

[6] Ouslim M., Curtis M.: Pattern recognition based on a probabilistic RAM net using n-tuple input mapping, IEE Proc.-Vis. Image Signal Process., Vol. 145, No. 6, Dec. 1998.

[7] Coric S., Latinovic I., Pavasovic A.: A Neural Network FPGA Implementation: Issues & Applications, NEUREL-2000, Yugoslavia, sept. 25-27, 2000

[8] Muthuramalingam A., Himavathi S., Srinivasan E.: Neural Network Implementation using FPGA, Int. Journal of Information technology Voln. 4, Nr. 2, 2007.

[9] Fasial M. Khan, Mark G. Arnold, and William M. Pottenger: Hardware-based Support Vector Machine Classification in Logarithmic Number System, IEEE Symposium on Circuit and Systems (ISCAS), May 2005.

[10] Jagdish C. Patra, Theodora A. Devi, and Pramod K. Meher: radial Basis Function Implementation of Intelligent Pressure Sensor on Field Programmable Gate Arrays, ICICS, 2007.