ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 75-80

Analog Brownian Weight Movement for Learning of Artificial Neural Networks

M.R.Belli, M.Conti, C.Turchetti

Dept. of Electronics, University of Ancona, Italy

Abstract. This paper proposes a stochastic learning approach, called Brownian weight movement, in which weight vector changes accordingly to the well known Brownian motion equation. The main features of such a method are: i) it is suitable for analog implementation; ii) it is able to inspect all the objective function domain so that convergence to global minimum is ensured.

1. Introduction

In the realm of Artificial Neural Networks (ANNs) the controversy between analog and digital implementation is still opened. As a matter of fact, it has been proven that biological neural networks are analog by nature, thus justifying the great attention devoted to this approach in the last years.

However, even though analog neural networks have many advantages over digital counterpart, learning implementation seems to be the main obstacle to this approach due to the complex operations required (such as multiplications, derivatives and so on). Additionally, learning algorithms such as Backpropagation [1-2], where the weight changes depend on gradient of the error, fails when the error objective function has a great number of local minima as in general occur in practical applications.

In this work we propose a learning approach for analog implementation, called Brownian Weight Movement, based on the well known Brownian motion equation, and able to overcome the drawbacks mentioned above. In particular the approach is suitable for analog implementation, giving rise to a dramatic reduction in circuit complexity compared to digital counterpart, and is able to reach global minima.

2. Brownian Weight Movement as stochastic process

Let us given an Artificial Neural Network (no dynamical) described by

$$y = \Gamma(w)\,[x] \qquad (1)$$

where $x \in \mathbb{R}^p$ is the input vector, $y \in \mathbb{R}^q$ the output vector and $\Gamma(w)$ a non-linear operator depending on the weights $w \in \mathbb{R}^r$.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 75-80

Learning is equivalent to the following problem:
for a given desired output $\tilde{y}$, find w such that the error $E = f[d(y,\tilde{y})]$ is minimum, being d(. , .) a suitable distance between functions.*
This is a typical optimisation problem which, in many cases, can be very hard to solve, since the existence of many local minima makes it difficult finding global minimum.
Searching methods based on the gradient of objective function, such as Newton algorithm for instance, are not particularly suited because trial solutions can be trapped into a local minimum. Instead random approaches, such as Simulated Annealing [3], do not suffer for this drawback since they are able to inspect all the objective function domain. This is equivalent to consider learning procedure as a stochastic process which, in the analog world, is described by a random differential equation.
The main properties the procedure must satisfy in this case are:
i) the most probable value of weights w must be close to the minimum of E(w) as $t \to \infty$;
ii) the variance of w must vanish as $t \to \infty$.
Thus with these observations in hand we propose the following random differential equation for weight movement

$$\frac{d^2 w(t)}{d t^2} = \left[-h \frac{d w(t)}{d t} + n(t) \right] u\left(\frac{d E(w(t))}{d t} \right) \qquad (2)$$

where

$$u(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases},$$

$$E = d(y,\tilde{y}) = |y - \tilde{y}|_{L1} = |e|_{L1} = \sup_x |e(x)|$$

and n(t) is a random process vector, whose components are statistically uncorrelated with zero mean and variance vanishing as $t \to \infty$.
This equation has a very simple meaning. Consider a time instant t, when $dE/dt \geq 0$, that is the trajectory of w(t) proceeds in a region where the energy increases, the equation becomes:

$$\frac{d^2 w(t)}{d t^2} = -h \frac{d w(t)}{d t} + n(t) \qquad (3)$$

This is the well known Langevin's equation describing Brownian motion [4], where n(t) represents a forced term. Thus, in this case, the learning scheme determines the subsequent direction of w randomly, according to eq.(3). In this way it allows the trajectory to surmount the energy hills and to search for region where the energy decreases. Conversely, when $dE/dt < 0$ eq. (2) becomes

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 75-80

$$\frac{d^2 w(t)}{d t^2} = 0 \tag{4}$$

corresponding to the motion of a free particle. In this case since the trajectory is moving on the right direction, corresponding to a decreasing energy, the same direction is maintained being the trajectory unaffected by any forced term.

3. Hardware implementation and results

The described approach has been used for the learning of a class of neural networks, named Approximate Identity Neural Networks, recently proposed in [5]. To show the feasibility and performances of the approach suggested, an all-analog breadboard networks has been realised using standard components. The learning circuit can be derived by rewriting eq.(2) as a couple of first order differential equations

$$\frac{d\, w(t)}{d\, t} = h\, v(t) \tag{5a}$$

$$\frac{d\, v(t)}{d\, t} = \begin{cases} -h\,[n'(t) - v(t)] & c(t)<0 \\ 0 & c(t)\geq 0 \end{cases} \tag{5b}$$

$$c(t) = u\left(\frac{d\, E(w(t))}{d\, t}\right) \tag{5c}$$

where $n'(t) = n(t)/h^2$.

It is easy to show that eqs.(5) together with eq.(1) map into the schematic of Fig.1. Here the AINN is simply depicted as a block corresponding to the operator $\Gamma(w)$. Learning is achieved by two other blocks: i) a block generating the signal c(t) defined by (5c); ii) a dynamic circuit whose trajectory w(t), representing the weight vector at time instant t, follows eqs.(5a)-(5b). The source of noise n(t) is applied at the input of the circuit through a switch acting under the control of the signal c(t).

Figs.2 and 3 show the schematic circuits adopted for the two blocks of Fig.1. As you can see only standard components are used. Additionally, the circuit suggested in [6] to generate uncorrelated noise has been used for n(t).

As an example of capability of such circuit, the learning of a triangle to sine wave conversion has been performed. In this case the input signal x(t) is a symmetric triangular waveform and the desired output y(t) is a sinusoidal waveform with the same frequency. The AINN used has 2 neurons with 6 parameters to be learned. Fig.4 shows the desired output y(t) and the output of neural network $\tilde{y}$ after learning process is terminated. The error function E(t) (lower line) and the control signal c(t) (upper line) during learning process are shown in Fig.5. As you can see when c(t) is low the error goes in the right direction, while when c(t) is high a random search starts.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 75-80

In Fig.6 a component of w(t) (lower line) and the control signal c(t) (upper line) during the learning process are reported: as w(t) tends to become constant, at the end of time interval, convergence to the global minimum is reached. Finally Fig.7 shows the error function E(t) (upper line) during Brownian weight movement (lower line).

References

[1] T.Morie,Y.Amemiya,"An all-analog expandable Neural Network LSI with on-chip Backpropagation Learning", IEEE J. on Solid-State Circuits, Vol.29, n.9, pp.1086-1093, Sept. 1994.

[2] Y.K.Choi,S.Y.Lee,"Subthreshold MOS implementation of neural networks with on-chip error back-propagation learning", Proc. of I.J.C.N.N., 1993, pp.849-852.

[3] J.Kirkpatrick et al.,"Optimization by simulated annealing", SCIENCE n.4598, Vol.220, May 1983, pp.671-680.

[4] A.Papoulis,"Probability, random variables, and stochastic processes", McGraw Hill, New York, 1965.

[5] M.Conti, S.Orcioni, C.Turchetti, "A class of neural networks based on approximate identity for analog IC's hardware implementation", IEICE Trans. on Fundamentals, Japan, Vol.E77-A, n.6, pp. 1069-1079, June 1994.

[6] J.A.Alspector et al., "A VLSI-efficient technique for generating multiple uncorrelated noise sources and its application to stochastic neural networks", IEEE Trans. on CAS, Vol. 38, n.1, pp.109-123, Jan 1991

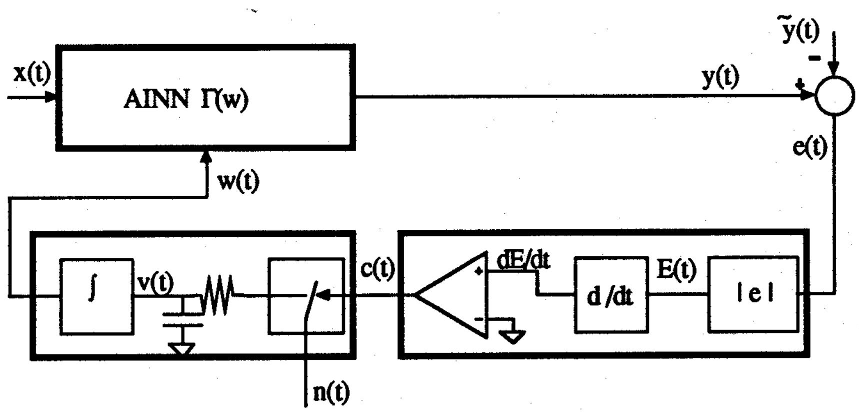

Fig. 1 Schematic of the neural network with learning.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 75-80

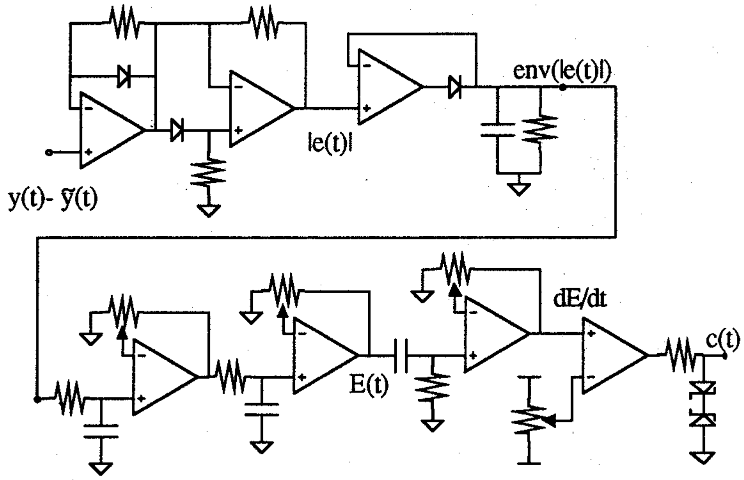

Fig.2 Detailed circuital schematic of the control block.

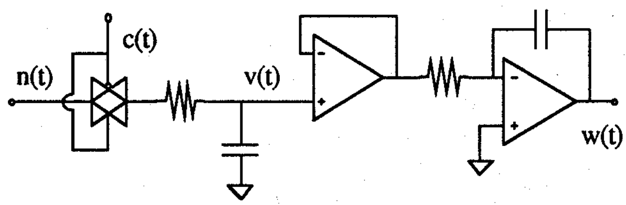

Fig.3 Detailed circuital schematic of the learning block.

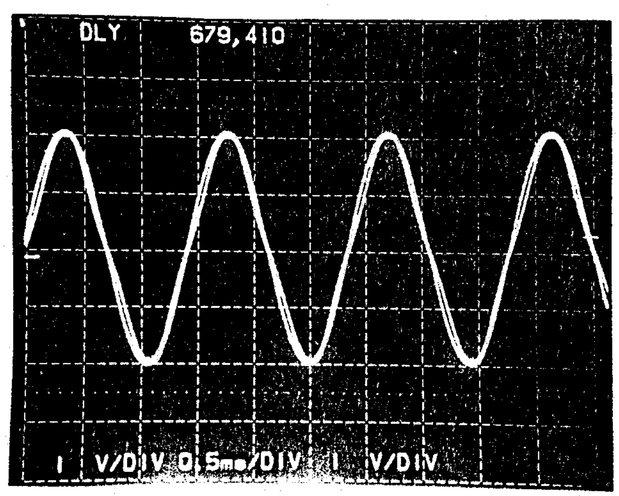

Fig.4 Desired output y(t) and output of the neural network $\tilde{y}$ when the learning process is terminated.(Vertical axis 1V:div, horizontal axis 0.5msec/div).

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 75-80

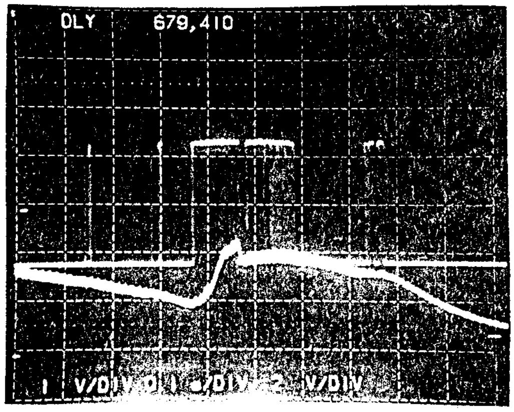

Fig.5 Error function E(t) (line below, 1V/div) and control function c(t) (upper line 2V/div) during the learning process in a time interval of 1 sec.

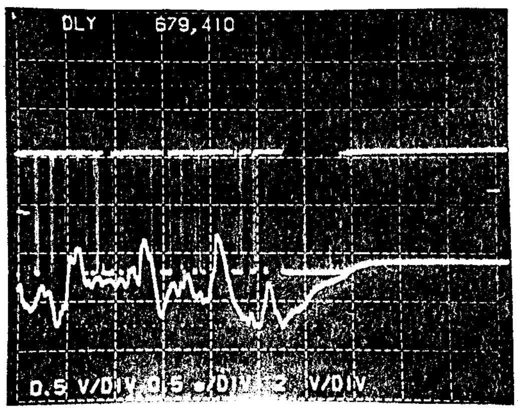

Fig.6 The value of one weight w(t) (line below, 0.5V/div) and control function c(t) (upper line 2V/div) during the learning process in a time interval of 5 sec.

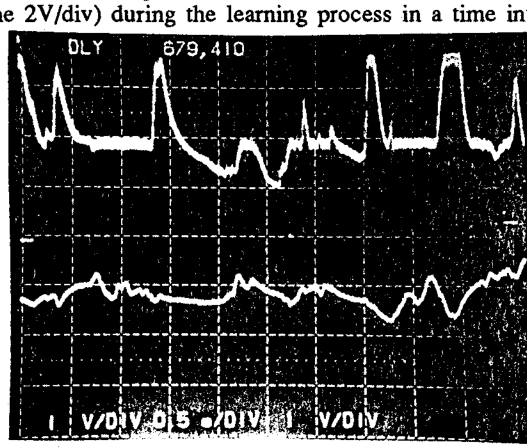

Fig.7 The value of one weight w(t) (line below, 1V/div) and error function E(t) (upper line 1V/div) during the learning process in a time interval of 5 sec.