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. 57-62

Adaptive signal processing with unidirectional Hebbian adaptation laws.*

Jeroen Dehaene[†] Joos Vandewalle

ESAT-SISTA K.U.Leuven,
Kardinaal Mercierlaan 94, B-3001 Leuven.
tel: 32.16.321111 fax:32.16.321986
e-mail:jeroen.dehaene@esat.kuleuven.ac.be

Abstract. We present three continuous time neural systems built with Hebbian connections and a new type of neurons, useful for signal processing applications. Some of the weights obey a classical Hebbian adaptation law, other weights obey a natural generalization of Hebbian adaptation laws, related to laws described by Oja and Sanger, but without bidirectional information flow.

The *first* application is recursive least squares estimation (RLS). The corresponding neural network is a continuous time limit of the well known discrete time Gentleman Kung systolic array. This application was worked out in [3, 4], with emphasis on the mathematical background and the relation with systolic arrays. Here we put more emphasis on the neural interpretation, and give another derivation.

Secondly, an extended system, obtained by putting a non adaptive classical one layer linear neural network in front of the RLS network, can be used for linearly constrained beamforming applications.

Thirdly, if the weights of this extra layer are made adaptive in the classical Hebbian sense and the RLS network is made to work at double speed, one obtains a system that can be used for subspace tracking and principal component analysis. The input output behavior of the system is identical to a continuous time limit of the neural stochastic gradient ascent algorithm of Oja, but it doesn't use bidirectional connections.

All systems are formulated in continuous time, but can be integrated exactly for piecewise constant input, yielding discrete time systolic algorithms.

*The following text presents research results of the Belgian programme on interuniversity attraction poles (IUAP-17 and IUAP-50) initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, and of the Concerted Action Project of the Flemish Community, entitled *Applicable Neural Networks*. The scientific responsibility rests with its authors. In addition, this research was supported by the F.K.F.O. project G.0292.95.

[†]Jeroen Dehaene is a research assistant of the N.F.W.O. (Belgian National Fund for Scientific Research).

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. 57-62

1. Recursive least squares estimation

In this section, we first give a purely descriptif explanation of the system in fig. 1 Then we briefly introduce the recursive least squares problem (RLS), and explain how fig. 1 provides a solution to it.

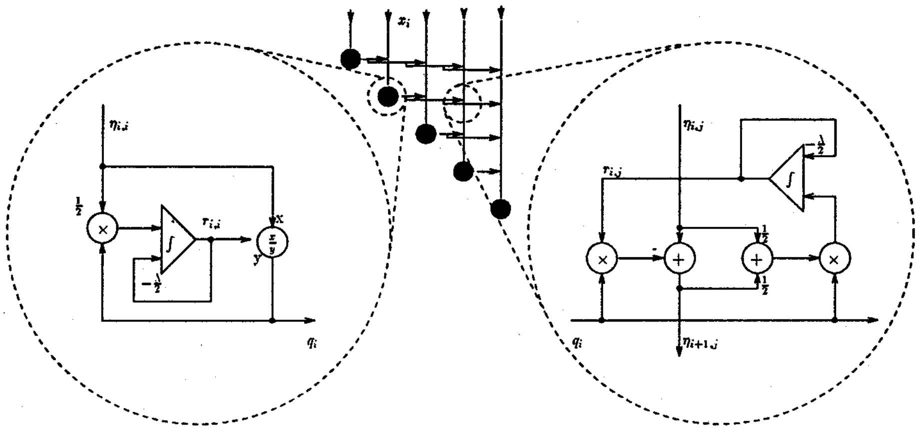

Figure 1: Neural network for recursive least squares estimation and detailed signal flow graph for a neuron i and a synapse (i, j)

The balls in the fig. 1 are neurons. They receive a current total input $\eta_{i,i}$ and keep track of $r_{i,i}$ which is an exponentially weighted quadratical average of past input $\eta_{i,i}$ (see below). The output q_i of a neuron i equals the current total input $\eta_{i,i}$ divided by $r_{i,i}$. The horizontal lines are axons sending the output q_i of a neuron i to other neurons to its right. The vertical lines carrying signals $\eta_{i,j}$ are dendrites, accumulating products of signals q_i coming from neurons to the left of the line and weights $-r_{i,j}$ of the synapses (i, j) between neuron i and j. The signal $\eta_{i,j}$ carries the accumulated input to neuron j from the first $i-1$ neurons. The adaptation of the synapses $r_{i,j}$ is Hebbian in the sense that it is proportional to presynaptic information q_i and postsynaptic information $\frac{1}{2}(\eta_{i,j} + \eta_{i,j+1})$. That is a classical Hebbian synapse, except for the fact that normally the total accumulated input $\eta_{j,j}$ to neuron j or even the output q_j of neuron j would be used as postsynaptic information. However, in view of the configuration, where input to a neuron is accumulated on one vertical line, the present law is not far fetched, as it uses even more local information.

The continuous time recursive least squares problem (RLS) can be stated as the following *time dependent optimization problem*

$$\min_{w_1(t),\ldots,w_{n-1}(t)} \int_{-\infty}^{t} (x_n(\tau) - \sum_{i=1}^{n-1} x_i(\tau)w_i(t))^2 e^{\lambda(\tau-t)} d\tau \tag{1}$$

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. 57-62

That is, at any time instant t the signals $x_1, \ldots, x_n$ are considered to be functions on the interval $(-\infty, t]$. And $\sum_{i=1}^{n-1} x_i w_i^*(t)$, where $w_i^*(t)$ denotes the optimal $w_i(t)$, is the best approximation of x_n as a linear combination of the signals x_i, $i = 1, \ldots, n-1$, in terms of the norm $\|f\|^2 = \int_{-\infty}^{t} f(\tau)^2 e^{\lambda(\tau - t)} d\tau$. The quantity $\lambda > 0$ is called the *forgetting rate* and determines the relative importance of past information. In some applications, one is interested in the approximation $\sum_{i=1}^{n-1} x_i w_i^*(t)$ itself, in other applications in the residual $x_n(t) - \sum_{i=1}^{n-1} x_i(t) w_i^*(t)$. Below, we will assume the latter. In all applications one is not interested in the whole function over $(-\infty, t]$ (note that the solution $\sum_{i=1}^{n-1} x_i w_i(t)$ yields a whole new function over $(-\infty, t]$ for any new t) but only in the "last" value of this function at time t. We will use the following notations for the resulting function of t taking all these "last" values

$$\begin{aligned} x_n \parallel \{x_1, \ldots, x_{n-1}\}(t) &= x_n(t) - \textstyle\sum_{i=1}^{n-1} x_i(t) w_i^*(t) \\ x_n \perp \{x_1, \ldots, x_{n-1}\}(t) &= x_n(t) - x_n \parallel \{x_1, \ldots, x_{n-1}\}(t) \end{aligned} \tag{2}$$

We will show that if the signals $x_1, \ldots, x_n$ are supplied as inputs to the system of fig. 1, that the signal $\eta_{n,n}$ gives the wanted residual $x_n \perp \{x_1, \ldots, x_{n-1}\}$.

This problem and its discrete time counter part have a broad spectrum of applications, such as adaptive noise cancellation, adaptive equalization, beam forming. We refer to [5, 6].

In [3, 4] we have given a matrix derivation of the algorithm implemented by the neural network of figure 1. The algorithm is given by

$$\dot{R} = \text{upph}(R^{-T} x x^T R^{-1}) R - \frac{\lambda}{2} R$$

where R is an upper triangular matrix storing the weights and where the operator upph takes the upper triangular part and halves the diagonal elements. This adaption law fits in a class of adaptation laws, with a simple parallel realization, introduced in [4].

Another derivation (less constructive, but giving more insight in the meaning of the different signals) starts from an analysis of the case where $n = 2$. In that case the solution is

$$x_2 \perp x_1 = x_2 - x_1(t) \frac{\int_{-\infty}^{t} x_1(\tau) x_2(\tau) e^{-\lambda(t-\tau)} d\tau}{\int_{-\infty}^{t} x_1(\tau)^2 e^{-\lambda(t-\tau)} d\tau} \tag{3}$$

This can be read as a projection of x_2 on x_1 (a new projection for any new t) in a function space over $(-\infty, t]$, with inner product $\langle f, g \rangle = \int_{-\infty}^{t} f(\tau) g(\tau) e^{-\lambda(t-\tau)} d\tau$.

Introducing $\eta_{1,1} = x_1$, $\eta_{1,2} = x_2$, $r_{1,1} = \sqrt{\int_{-\infty}^{t} x_1^2(\tau) e^{-\lambda(t-\tau)} d\tau}$, $r_{1,2} = \int_{-\infty}^{t} x_1(\tau) x_2(\tau) e^{-\lambda(t-\tau)} d\tau / r_{1,1}$, and $q_1 = \frac{x_1}{r_{1,1}}$, (3) becomes

$$\eta_{2,2} = x_2 \perp x_1 = \eta_{1,2} \perp \eta_{1,1} = \eta_{1,2} - q_1 r_{1,2}$$

and $r_{1,1}$ and $r_{1,2}$ can be updated by

$$\begin{aligned} \dot{r}_{1,1} &= \tfrac{1}{2} q_1^2 r_{1,1} - \tfrac{\lambda}{2} r_{1,1} \\ \dot{r}_{1,2} &= q_1 \tfrac{1}{2} (\eta_{1,2} + \eta_{2,2}) - \tfrac{\lambda}{2} r_{1,2} \end{aligned} \tag{4}$$

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. 57-62

This is exactly what is realized by neuron 1 (storing $r_{1,1}$) and synapse $(1, 2)$ (storing $r_{1,2}$) in fig. 1.

The solution of fig. 1 recursively calculates all $\eta_{i+1,j} = x_j \perp \{x_1, \ldots, x_i\}$, $(i < j)$, according to

$$x_j \perp \{x_1, \ldots, x_i\} = (x_j \perp \{x_1, \ldots, x_{i-1}\}) \perp (x_i \perp \{x_1, \ldots, x_{i-1}\})$$

that is

$$\eta_{i+1,j} = \eta_{i,j} \perp \eta_{i,i}$$

This is realized by neuron i and synapse (i, j).

2. LCMV filtering and LCMV beamforming

In this section we consider the application of linearly constrained minimum variance (LCMV) beamforming [6], which can be realized by the neural system introduced in the previous section, with an extra fixed linear neural layer up front. We first give a brief sketch of the application.

LCMV filtering and LCMV beamforming are other applications for which RLS can be used, but before the vector x of input signals is supplied to, for instance, a Gentleman Kung array, it is multiplied by a fixed matrix [6]. For a beamforming application, the inputs come from a linear array of n antennas, onto which a planar wave impinges. The beamformer linearly combines these inputs in an adaptive way, minimizing the output power subject to one or more constraints. A typical constraint requires that signals from a given direction are passed unchanged. The constraints determine the fixed preprocessing matrix.

For the neural system, the multiplication with the fixed matrix, is realized by adding a linear one layer neural network in front of the network of fig. 1 with non adaptive weights and non dynamic linear neurons. This is shown in fig. 2.

3. Principal component estimation and subspace tracking

In this section we discuss a system similar to the one of the previous section, but with Hebbian weights in the first layer.

In [7] we have given a derivation of a continuous version of the neural stochastic gradient ascent algorithm of [1], which is closely related to the algorithm given by [2]. The derivation given in [7] derives the algorithm as a continuous time spherical subspace tracker, by analogy with the discrete time spherical subspace tracking algorithm of [8]. Below we first sketch the problem of subspace tracking. Than we derive a new algorithm which can be realized by the system of fig. 2 with Hebbian weights in the first layer.

Subspace tracking consists in the adaptive estimation of the column space of a slowly time-varying $n \times \kappa$ matrix M, given a signal x, generated by $x(t) = Ms(t) + n(t)$, where $s(t)$ is a source signal with non singular correlation matrix

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. 57-62

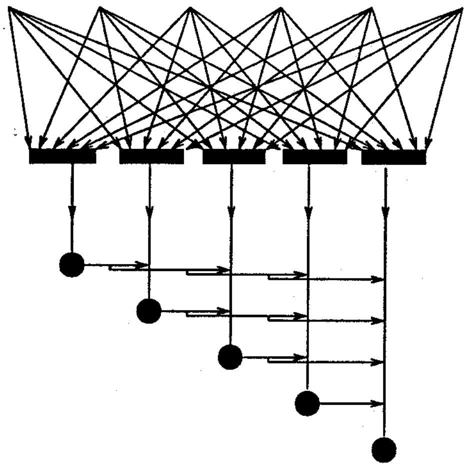

Figure 2: Neural network for LCMV beamforming (with fixed weights in the first layer) or for subspace tracking (with Hebbian weights in the first layer)

$E\{s(t)s(t)^T\}$, and $n(t)$ is additive white noise. That is, we know that $x(t)$ is generated by some κ order linear model, and we want to track the (slowly evolving) κ-dimensional subspace in which $x(t)$ would lie in the absence of noise. Typical applications are frequency estimation, direction of arrival estimation, and beam forming [5].

The continuous time algorithm, discussed and analyzed in [7] is given by

$$\dot{A} = \gamma \cdot \{xx^T A - 2A\,\mathrm{upph}(A^T xx^T A)\} \tag{5}$$

where A is an orthogonal matrix, whose columns span an estimate of the column space of M. In most cases $y = A^T x$ is considered as output. Here, we derive another algorithm with the same input output behavior but storing a rectangular matrix B and a triangular matrix R, such that $A = BR^{-1}$. If $B(0) = A(0)$ and $R(0) = I$, and

$$\begin{aligned} \dot{B} &= \gamma \cdot xx^T B \\ \dot{R} &= 2\gamma \cdot \mathrm{upph}(R^{-T} B^T xx^T BR^{-1})R \end{aligned} \tag{6}$$

then, with $S = R^{-1}$ one finds (from $\frac{d}{dt}(RS) = 0$) that $\dot{S} = -S\dot{R}S = -2\gamma \cdot S\,\mathrm{upph}(S^T B^T xx^T BS)$ and

$$\dot{A} = \dot{B}S + B\dot{S} = \gamma \cdot \{xx^T A - 2A\,\mathrm{upph}(A^T xx^T A)\}$$

which corresponds to the algorithm (5).

Clearly, algorithm (6) can be implemented by a linear one layer network for B with classical Hebbian synapses (now using the total accumulated input to a neuron as postsynaptic information), followed by the RLS network of section 1. for R, working at double speed. In fig. 2 the fact that the weights in

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. 57-62

the first layer adapt proportionally to the total input to a neuron instead of the partially accumulated input along a long dendrite, is represented by drawing other neurons than in figure 1, getting their input in parallel (one could think of a dendrite with n short branches) instead of along one long dendrite.

4. Conclusion

We have shown how continuous time algorithms for some classical signal processing operations, can be performed by simple neural networks, with unidirectional Hebbian connections, and a new type of neurons.

References

[1] E. Oja, "Principal Components, Minor Components, and Linear Neural Networks", Neural Networks 5 (1992) 927-935.

[2] T.D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network", Neural Netowrks, 2 (1989) 459-473.

[3] J. Dehaene, M. Moonen, J. Vandewalle, "Continuous time recursive least squares estimation, adaptive neural networks and systolic arrays.", to appear in IEEE trans. on Circuits and Systems I.

[4] J. Dehaene, M. Moonen, J. Vandewalle, "Uniform local implementation on networks and arrays of continuous time matrix algorithms for adaptive signal processing", report 93-34I, ESAT-SISTA, K.U.Leuven, available by anonymous ftp to gate@esat.kuleuven.ac.be as /pub/SISTA/dehaene/reports/93-34I.ps.Z and shorter version in "Systems and Networks: Mathematical Theory and Applications, Proceedings of the International Symposium MTNS '93, Regensburg, Germany, August 2-6, Volume II, Invited and Contributed Papers", ed. U. Helmke, R. Mennicken, J. Saurer, Akademie Verlag, 1994, pp. 651-654.

[5] S. Haykin, "Adaptive filter theory", Prentice Hall, Englewood Cliffs, 1991.

[6] B. Van Veen, K.M. Buckley, "Beamforming: a versatile approach to spatial filtering", IEEE ASSP Magazine, pp.4-23, April 1988.

[7] J. Dehaene, M. Moonen, J. Vandewalle, "A continuous time approach to the analysis and design of parallel algorithms for subspace tracking" to appear in proceedings of the third international workshop on SVD and signal processing, August 22-25, 1994, Leuven, Belgium.

[8] R.D. DeGroat, "Non-iterative subspace tracking", IEEE Transaction on Signal Processing 40 no.3, pp. 571-577, 1992.