HAMILTONIAN NEURAL NETWORK-BASED ORTHOGONAL FILTERS
A Basis for Artificial Intelligence

Wieslaw Citko and Wieslaw Sienko
Department of Electrical Engineering, Gdynia Maritime University, Morska 81-86, 81-225, Gdynia, Poland

Keywords: Hamiltonian neural network, Machine learning, Artificial intelligence.

Abstract: The purpose of the paper is to present how very large scale networks for learning can be designed by using Hamiltonian Neural Network-based orthogonal filters and in particular by using octonionic modules. We claim here that octonionic modules are basic building blocks to implement AI compatible processors.

1 INTRODUCTION

It is well known that true artificial intelligence cannot be implemented with traditional hardware. However, it should also be clear that in order to build machines that learn, reason and recognize one needs power-efficient processors with computational efficiency unattainable even by supercomputers. Two such processors are theoretically known: quantum computers and neuromophic or brain-like structures. Unfortunately, in recent years, quantum computers have lost much of their luster. Some researchers are sceptical about eventual realization of quantum computers (Gea-Banacloche, 2010). One of the Nobel Prize winners even claims that the ideal quantum computer can never be built: “no quantum computer can ever be built that can outperform a classical computer if the latter would have its components and processing speed scaled to Planck units” (Hooft, 2000). The main premise for the claim above is the essential and unavoidable decoherence in quantum systems. Thus, due to the decoherence, an ideal quantum computer as the state superposition based processor cannot be constructed. It is also worth noting that an ideal quantum computer is an example of a Hamiltonian system. As mentioned above the other way leading to the realization of power-efficient processors involves neuromorphic systems. It is well known that up to date, using different technology, several neuromorphic devices (e.g. oscillatory and static artificial neurons and based on them structures) have been proposed (Basu and Hasler, 2010). The newest project in this field is memristor concept based neuromorphic structure (Versace and Chandler, 2010). The authors of MoNETA, the brain on a chip, claim that memristor based technology, which mimicks biological axon and wetware structure, is a solution leading even to true AI. An interesting question that arises here is whether such structures, classified as bottom-up solutions, can create true AI processors. We claim that a biological brain is an almost lossless dynamic structure and hence neuromorphic systems should be sought in a class of Hamiltonian systems i.e. Hamiltonian neural networks. The main goal of this presentation is to prove the following statement: Let as assume that AI issues can be formulated as implementation of mapping $\mathbf{z} = F(\mathbf{x})$, where F(.) is known by training set $\{ \mathbf{x}_i, \mathbf{z}_i \}$; $i = 1, \dots, m$. Then any such F(.) can be implemented by using Hamiltonian neural networks and in particular by using octonionic modules.

2 HAMILTONIAN NEURAL NETWORKS

It is well known that a general description of Hamiltonian network is given by the following state–space equation:

$$\dot{\mathbf{x}} = \mathbf{J}H'(\mathbf{x}) = \mathbf{v}(\mathbf{x}) \tag{1}$$

where: $\mathbf{x}$ – state vector, $\mathbf{x} \in R^{2n}$
$\mathbf{v}(\mathbf{x})$ – a nonlinear vector field
$\mathbf{J}$ – skew-symmetric, orthogonal matrix.

124 Citko W. and Sienko W..
HAMILTONIAN NEURAL NETWORK-BASED ORTHOGONAL FILTERS - A Basis for Artificial Intelligence.
DOI: 10.5220/0003671501240127
In *Proceedings of the International Conference on Neural Computation Theory and Applications* (NCTA-2011), pages 124-127
ISBN: 978-989-8425-84-3

Function $H(\mathbf{x})$ is an energy absorbed in the network. Since Hamiltonian networks are lossless (dissipationless), their trajectories in the state space can be very complex for $t \in (-\infty, \infty)$. Equation (1) gives rise to the model of Hamiltonian Neural Networks (HNN), as follows:

$$\dot{\mathbf{x}} = \mathbf{W}\boldsymbol{\Theta}(\mathbf{x}) + \mathbf{d} \tag{2}$$

where: $\mathbf{W}$ – (2n×2n) skew-symmetric, orthogonal weight matrix ($\mathbf{W}^2 = -\mathbf{1}$)

$\boldsymbol{\Theta}(\mathbf{x})$ – vector of activation functions (output vector $\mathbf{y} = \boldsymbol{\Theta}(\mathbf{x})$)

$\mathbf{d}$ – input data

and $\boldsymbol{\Theta}(\mathbf{x}) \equiv H'(\mathbf{x})$

One assumes here that activation functions are passive i.e.:

$$\mu_1 \le \frac{\Theta(x)}{x} \le \mu_2 \ ; \ \mu_1, \mu_2 \in (0, \infty)$$

The HNN described by Eq.(1) cannot be realized as a macroscopic scale physical object. Introducing the negative-feedback loops, the Eq.(2) can be reformulated as follows:

$$\dot{\mathbf{x}} = \left(\mathbf{W} - w_0\mathbf{1}\right)\boldsymbol{\Theta}(\mathbf{x}) + \mathbf{d} \tag{3}$$

where: $w_0 > 0$

and Eq.(3) sets up an orthogonal transformation (HNN-based orthogonal filter):

$$\mathbf{y} = \frac{1}{1 + w_0^2}(\mathbf{W} + w_0\mathbf{1})\mathbf{d} \tag{4}$$

where: $\mathbf{W}^2 = -\mathbf{1}$

8-dim. orthogonal filter, referred to as <u>octonionic module</u>, can be synthesized by the formula:

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} = \frac{1}{\sum_{i=1}^{8} y_i^2} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \tag{5}$$

i.e. $\mathbf{w} = \mathbf{Y}\,\mathbf{d}$

It can be seen that Eq.(5) is a solution the following design problem: for a given input vector $\mathbf{d} = [d_1, \dots, d_8]^T$ and a given output vector $\mathbf{y} = [y_1, \dots, y_8]^T$ find weight matrix $\mathbf{W}$ of HNN based orthogonal filter (octonionic module). Thus:

$$\mathbf{W}_8 = \begin{bmatrix} 0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ -w_1 & 0 & w_3 & -w_2 & w_5 & -w_4 & -w_7 & w_6 \\ -w_2 & -w_3 & 0 & w_1 & w_6 & w_7 & -w_4 & -w_5 \\ -w_3 & w_2 & -w_1 & 0 & w_7 & -w_6 & w_5 & -w_4 \\ -w_4 & -w_5 & -w_6 & -w_7 & 0 & w_1 & w_2 & w_3 \\ -w_5 & w_4 & -w_7 & w_6 & -w_1 & 0 & -w_3 & w_2 \\ -w_6 & w_7 & w_4 & -w_5 & -w_2 & w_3 & 0 & -w_1 \\ -w_7 & -w_6 & w_5 & w_4 & -w_3 & -w_2 & w_1 & 0 \end{bmatrix} \tag{6}$$

$\mathbf{W}_8$- matrix belongs to the family of Hurwitz-Radon matrices.

Octonionic module can be seen as a basic building block for the construction of AI processors. Moreover, the output $\mathbf{y}$ of filter in Eq.(4) is a Haar spectrum of input vector $\mathbf{d}$. It is worth noting that an octonionic module sets up an elementary memory module as well. Designing, for example, an orthogonal filter, using Eq. (4) and (5), which performs the following transformation:

$$\mathbf{y}_{[1]} = \frac{1}{1 + w_0^2}(\mathbf{W} + w_0\mathbf{1})\mathbf{m} \tag{7}$$

where: $\mathbf{y}_{[1]} = [1, 1, \dots, 1]^T$ i.e. synthesizing by Eq.(5) a flat Haar spectrum for given input vector $\mathbf{m}$, such that

$$\sum_{i=1}^{8} m_i > 0$$

one gets an implementation of linear perceptron, as shown in Fig.1.

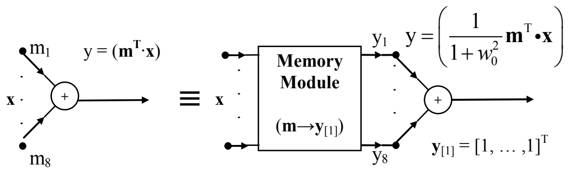

Figure 1: Implementation of elementary memory module by octonionic module.

Moreover, according to Eq.(5) and (7) the matrix $\mathbf{Y}$ with $y_1 = y_2 = \dots = y_8 = 1$ generates the structures of all memory modules. It is worth noting that transformation in Eq.(5) can be also realized by the octonionic modules, as shown in Fig.2.

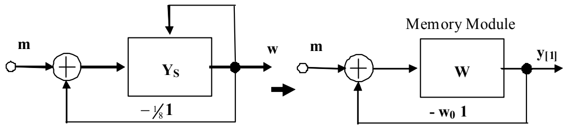

Figure 2: Self-creation of memory module.

where: $\mathbf{Y}_s$–skew-symmetric part of matrix $\mathbf{Y}$ (Eq.(5))

W - weight matrix of memory modules (Eq.(6) and Eq.(7)).

Such transformation can be seen as a process of self creation of memory modules.

To summarize the considerations above, one can state that the octonionic module is an universal building block to realize very large scale orthogonal filters and in particular memory blocks. Multidimensional, octonionic modules based orthogonal filters can be realized by using family of Hurwitz-Radon matrices. Thus, 16-dim orthogonal filter can be, for example, determined by the following matrix:

$$\mathbf{W}_{16} = \left[\begin{array}{c|c} \mathbf{W}_8 & \begin{matrix} w_8 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & w_8 \end{matrix} \\ \hline \begin{matrix} -w_8 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -w_8 \end{matrix} & \mathbf{W}_8^T \end{array}\right] = \begin{bmatrix} \mathbf{W}_8 & \mathbf{0} \\ \mathbf{0} & -\mathbf{W}_8 \end{bmatrix} + w_8 \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix} \quad (8)$$

where: $w_8 \in R$

Similarly, for dimension $N = 2^k$, $k = 5, 6, 7, \ldots$ all Hurwitz-Radon matrices can be found, as:

$$\mathbf{W}_{2^k} = \begin{bmatrix} \mathbf{W}_{2^{k-1}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{W}_{2^{k-1}} \end{bmatrix} + w_K \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix} \quad (9)$$

where: $w_K \in R$.

3 ORTHOGONAL FILTER BASED APPROXIMATION OF FUNCTIONS

The purpose of these considerations is to show how a function $f(\mathbf{x})$, given at limited number of trainings data $\mathbf{x}_i$, can be implemented by a composition of HNN based orthogonal filters in particular by using octonionic modules. Such an implementation can be regarded as a problem of approximation of multivariate function from sparse data i.e. training pairs $\{\mathbf{x}_i, \mathbf{z}_i\}$, $i = 1, 2, \ldots, m$ (the problem known from learning theory). Let us define $f: x \to z$ by:

$$f(\mathbf{x}) = \sum_{i=1}^{m} c_i K(\mathbf{x}, \mathbf{x}_i) \quad (10)$$

where coefficients c_i are such as to minimize the errors on the training set, i.e. they satisfy the following system of the linear equations:

$$\mathbf{K}\ \mathbf{c} = \mathbf{z} \quad (11)$$

where: $\mathbf{c} = [\ c_1, \ldots, c_m]^T$ and $\mathbf{K}$ is kernel matrix:

$$\mathbf{K} = \{K_{ij}\} = \{K(\mathbf{x}_i, \mathbf{x}_j)\},\ i, j = 1, 2, \ldots, m$$

The solvability and quality of approximation depends on the properties of the kernel matrix. Orthogonal filter based structure of function approximator is shown in Fig.3. To simplify the presentation, we assume that the structure in Fig.3 is 8-dimensional, i.e. dim $\mathbf{x} = 8$.

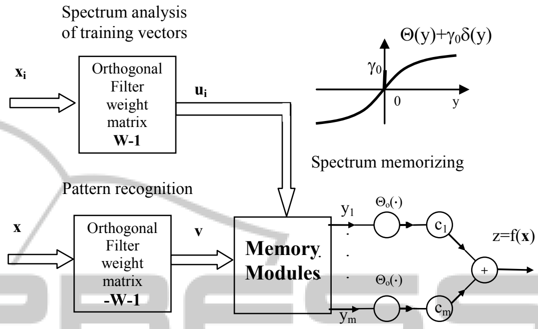

Figure 3: Orthogonal filter-based structure of a function approximator.

This structure relies on using the following kernels (Sienko and Citko, 2009):

$$K(\mathbf{u}_i, \mathbf{v}) = \Theta_0\left(\frac{1}{1+w_{0i}^2}\mathbf{u}_i^T\mathbf{v}\right) + \gamma_o\delta(\mathbf{u}_i^T\mathbf{v}) \quad (12)$$

where: $\Theta_0(\ .\ )$ – a nonlinear odd function

$\delta(\ .\ )$ – Kronecker's delta

$\gamma_0 \geq 0$

$$w_{oi} = \frac{1}{8}\sum_{k=1}^{8} u_{ki}\ ,\ \mathbf{u}_i = [\ u_{1i}, u_{2i}, \ldots, u_{8i}]^T$$

Since:

$$\mathbf{u}_i = \frac{1}{2}(\mathbf{W} + \mathbf{1})\mathbf{x}_i \quad (13)$$

$$\mathbf{v} = \frac{1}{2}(-\mathbf{W} + \mathbf{1})\mathbf{x} \quad (14)$$

then

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Theta_0\left(\frac{1}{2(1+w_{0i}^2)}\mathbf{x}_i^T\mathbf{W}^T\mathbf{x}_j\right) + \gamma_0\delta\left(\frac{1}{2}\mathbf{x}_i^T\mathbf{W}^T\mathbf{x}_j\right) \quad (15)$$

and kernel matrix has a form:

$$\mathbf{K} = \mathbf{K}_s + \gamma_0\ \mathbf{1} \quad (16)$$

where: $\mathbf{K}_s$- skew-symmetric matrix.

dim $\mathbf{K}_s = m$ (number of training points)

It is clear that the design equation (11), with the kernel matrix (16), is for $\gamma_0 > 0$ well-posed. Hence, a numerical stable solution exists:

$$\mathbf{c} = \mathbf{K}^{-1}\mathbf{z} \quad (17)$$

Moreover, Eq.(11) can be embedded into the following differential equation:

$$\dot{\varsigma} = (-\mathbf{K}_s - \gamma_0 \mathbf{1})\mathbf{\Theta}(\varsigma) + \mathbf{z} \qquad (18)$$

and hence, for number m even (i.e. even number of training points), it can be implemented by a lossless neural network, as shown schematically in Fig.4.

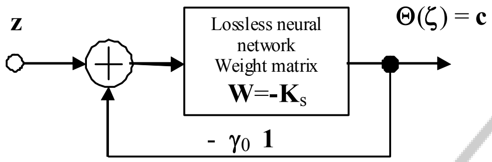

Figure 4: Lossless neural network-based structure for solution of Eq.(17).

The output of this neural network is:

$$\mathbf{c} = \mathbf{\Theta}(\zeta) = (\mathbf{K}_s + \gamma_0 \mathbf{1})^{-1}\mathbf{z} \qquad (19)$$

The stability of solution (19) can be achieved by damping action of parameter $\gamma_0 > 0$ (it can be regarded as a regularization mechanism (Evgeniou et. al., 2000). It is easy to see that the lossless neural network shown in Fig.4. can be realized by using octonionic modules, similarly as Hamiltonian neural network given by Eq.(3). Thus, one gets the following statement: Octonionic module is a fundamental building block for the realization of AI compatible processors. The 8-dimensional structure from Fig.3 can be directly scaled up to dimension $N = 2^k$, k = 5, 6, 7, … using octonionic modules.

4 ON IMPLEMENTATION OF OCTONIONIC MODULES

It can be seen that HNN as described by Eq.(2) is a compatible connection of n elementary building blocks-lossless neurons. A lossless neuron is described by the differential equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \pm w_1 \\ \mp w_1 & 0 \end{bmatrix} \begin{bmatrix} \Theta(x_1) \\ \Theta(x_2) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \qquad (20)$$

Hence, the octonionic module, with weight matrix $\mathbf{W}_8$ consists of four lossless neurons, according to Eq.(6). A practical circuit solution of near lossless neurons can be realized using nonlinear voltage controlled current sources (VCCS), which are compatible with VLSI technology. A concrete circuit however, is beyond the scope of this presentation.

5 CONCLUSIONS

The main goal of this paper was to prove the following statement:

AI compatible processor should be formulated in the form of top-down structure via the following hierarchy: Hamiltonian neural network (composed of lossless neurons) – octonionic module (a basic building block) – nonlinear voltage controlled current source (device compatible with VLSI technology).

Hence, it has been confirmed in this paper that by using octonionic module based structures, one obtains regularized and stable networks for learning. Thus, typical for AI tasks, such as realization of classifiers, pattern recognizers and memories, could be physically implemented for any number $N=2^k$ (dimension of input vectors) and any even $m < \infty$ (number of training patterns).

It is clear that octonionic module cannot be ideally realized as an orthogonal filter (decoherence-like phenomena).

Hence, the problem under consideration now is as follows: how exactly an octonionic module be realized by using cheap VLSI technology to preserve the main property-orthogonality, power efficiency and scaleability.

The possibility to directly transform the static structure to the phase-locked loop (PLL)-based oscillatory structure of octonionic modules is noteworthy.

REFERENCES

Gea-Banacloche, J., (2010). Quantum Computers: A status update, *Proc. IEEE*, vol. 98, No 12.

Hooft, G., (2000). Determinism and Dissipation in Quantum Gravity. *arXiv*: hep-th/0003005v2, 16.

Basu, A., Hasler, P. A, (2010). Nullcline-based Design of a Silicon Neuron. *Proc. IEEE Tr. on CAS*, vol 57, No 11.

Versace, M., Chandler, B., (2010). The Brain of a New Machine. *IEEE Spectrum*, vol.47, No 12.

Sienko, W., Citko, W., (2009). Hamiltonian Neural Networks Based Networks for Learning, In Mellouk, A. and Chebira, A. (Eds.), *Machine Learning*, ISBN 978-953-7619-56-1, J, pp. 75-92, Publisher: I-Tech, Vienna, Austria.

Evgeniou, T., Pontil, M., Poggio, T., (2000). Regularization Networks and Support Vector Machines, In Smola, A., Bartlett, P., Schoelkopf, G., Schuurmans, D., (Eds), *Advances in Large Margin Classifiers*, pp. 171-203, Cambridge, MA, MIT Press.