4

Hamiltonian Neural Networks Based Networks for Learning

Wieslaw Sienko and Wieslaw Citko
Gdynia Maritime University
Poland

1. Introduction

The problem of learning represents a gateway to understanding intelligence in brains and machines. Many researchers believe that supervised learning will become a key technology for extracting information from the flood of data around us. The supervised learning techniques, i.e. learning from examples, can be seen as an implementation of the mappings $\mathbf{y} = \mathbf{F}(\mathbf{x})$, relying on the fitting of given data pairs $\{\mathbf{x}_k, \mathbf{y}_k\}$. The key point is that the fitting should be predictive and uncover an underlying physical law, which is then used in a predictive or anticipatory way. A great number of models implementing the supervised learning techniques have been proposed in literature. Artificial Neural Networks (ANN), Radial Basis Functions (RBF), Support Vector Machines (SVM) and Fuzzy Logic based models (ANFIS) should be here mentioned. Support Vector Machines, distinctive tools for data classification, are the product of statistical learning theory. Recently, a new learning algorithm named Regularized Least Squares Classification (RLSC) has been proposed. The RLSC concept relyies on multivariate function approximation with regularization theory as a natural framework for solving ill-posed problems of approximation. It is worth noting that SVM and RBF models can be regarded as special cases in the framework of approximation and regularization theory. On the other hand, the Hamiltonian Neural Networks (HNN) based orthogonal filters can be regarded as a natural implementation of the regularization technique. Using Hamiltonian Neural Networks based spectrum analysis, recognition, and memorization, gives rise to mapping implementations with skew-symmetric and symmetric kernels. The purpose of this chapter is to present how very large scale networks for learning can be designed by using HNN-based orthogonal filters and, specifically, by using 8-dimensional (octonionic) modules. The unique feature of HNN is the fact that they can exist as either algorithms or physically implementable devices. In this chapter we mainly concentrate on algorithmic description of HNN-based networks. Moreover, since the structures of HNN can be based on the family of Hurwitz-Radon matrices, we present here how to design large scale nonlinear mappings by using neural networks with weight matrices determined by Hurwitz-Radon matrices. Hence, this chapter consists of the following issues:

- Fundamentals of HNN
- Family of Hurwitz-Radon matrices
- RLSC basics

Source: Machine Learning, Book edited by: Abdelhamid Mellouk and Abdennacer Chebira,
ISBN 978-3-902613-56-1, pp. 450, February 2009, I-Tech, Vienna, Austria

- Orthogonal filter-based approximation
- Modeling classifiers, pattern recognition and associative memories via nonlinear mappings
- Attractors based very large scale associative memories
- Conclusions

There is a large literature on the subject of networks for learning. Here we only refer to some comprehensive and useful, from the point of view of our presentation, reviews: (Evgeniou et al., 2000), (Poggio & Smale, 2003), (Boucheron et al., 2005), (Predd et al., 2006).

2. Hamiltonian neural networks

It is well known that a general description form of an autonomous Hamiltonian network is given by the following state-space equation:

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

where: $\mathbf{x}$ - state vector, $\mathbf{x} \in R^{2n}$
$\mathbf{v}(\mathbf{x})$ – a nonlinear vector field
and: $-\mathbf{J} = \mathbf{J}^T = \mathbf{J}^{-1}$ i.e. $\mathbf{J}$ is skew-symmetric and orthogonal.

Function $H(\mathbf{x})$ is a Hamiltonian (energy) of the network. Since Hamiltonian networks are lossless (there is no dissipation of energy), their trajectories in the state space can be very complex for $t \to \pm\infty$. It is, however, worth noting that Eq.(1) has constant solutions, i.e. every points $\mathbf{x}_0 \in R^{2n}$ such that $\mathbf{H}'(\mathbf{x}_0) = \mathbf{0}$ is the equilibrium and $\mathbf{x}(t) \equiv \mathbf{x}_0$ is the solution. Equation (1) gives rise to the modeling of Hamiltonian Neural Networks, 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
$\boldsymbol{\Theta}(\mathbf{x})$ – vector of activation functions
$\mathbf{d}$ – input vector (input data)
and: $\boldsymbol{\Theta}(\mathbf{x}) \equiv \mathbf{H}'(\mathbf{x})$
E=H($\mathbf{x}$) - the energy absorbed by HNN

It can be easy seen that HNN, as described by Eq.(2), is a compatible connection of n elementary building blocks – lossless neurons (Fig.1).

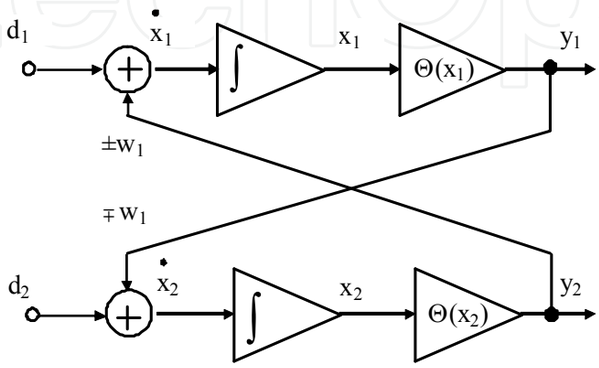


Fig.1. Structure of a lossless neuron.

The state-space description of a lossless neuron is as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \pm w_1 \\ \mp w_1 & 0 \end{bmatrix} \begin{bmatrix} \Theta(x_1) \\ \Theta(x_2) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{3}$$

where the activation function $\Theta(x_i)$, i = 1, 2 has been assumed as a passive nonlinearity, i.e.:

$$\mu_1 \le \frac{\Theta(x_i)}{x_i} \le \mu_2 \; ; \; \mu_1, \mu_2 \in (0, \infty) \tag{4}$$

A lossless neuron is an elementary Hamiltonian network with absorbed energy given by:

$$E = E_1 + E_2 = \int_0^{x_1} \Theta(\varsigma_1) d\varsigma_1 + \int_0^{x_2} \Theta(\varsigma_2) d\varsigma_2 \ge 0 \tag{5}$$

Formula (5) can be directly extended onto n-neuron HNN:

$$E = \sum_{i=1}^{n} E_i$$

where: E_i – energy absorbed by the i-th neuron.
Note, that for weight matrix **W** skew-symmetric but nonorthogonal, Eq.(2) describes a lossless neural network. The Hamiltonian neural network described by Eq.(1) cannot be realized as a macroscopic scale physical object. But HNN determines a type of orthogonal transformation, namely:

$$\mathbf{W} \cdot \mathbf{\Theta}(\mathbf{x}) + \mathbf{d} = \mathbf{0} \tag{6}$$

$$\mathbf{y} = \mathbf{\Theta}(\mathbf{x}) = \mathbf{W}\mathbf{d} \tag{7}$$

$$(\mathbf{y}, \mathbf{d}) = 0; \quad (\cdot\,, \cdot) \text{ – scalar product}$$

Rows (and columns) of **W** constitute an orthogonal Haar basis. The components of output vector are Haar coefficients. Thus, Haar spectrum analysis using HNN is given by:

$$\mathbf{y} = \mathbf{W}\,\mathbf{d} \quad \text{and} \quad \mathbf{d} = -\mathbf{W}\,\mathbf{y} \tag{8}$$

and formula (8) can be used as an algebraic transformation. The problem of physical realizability of HNN can be solved by using HNN-based orthogonal filters. A basic structure of such filters is shown in Fig.2.

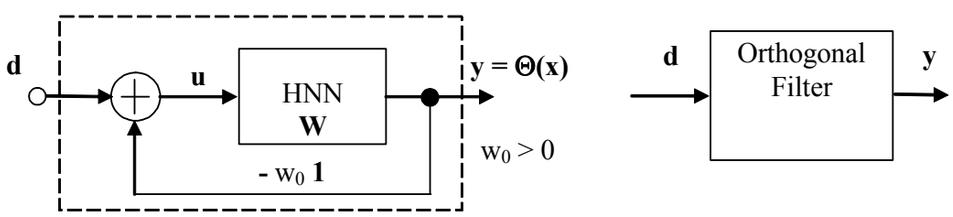


Fig. 2. Structure of an orthogonal filter.

It is worth noting that an orthogonal filter performs the following decomposition:

$$\mathbf{d} = \mathbf{u} + w_0 \mathbf{y} \tag{9}$$

where: $\mathbf{u}$, $\mathbf{y}$ are orthogonal i.e. $(\mathbf{u}, \mathbf{y}) = 0$
Moreover, Eq.(9) sets up the following orthogonal transformation ($\mathbf{W}^2 = -\mathbf{1}$):

$$\mathbf{y} = \frac{1}{1 + w_0}(\mathbf{W} + w_0 \mathbf{1})\mathbf{d} \tag{10}$$

where: $(\mathbf{d}, \mathbf{y}) \neq 0$
The output vector $\mathbf{y} = \Theta(\mathbf{x})$ constitutes the Haar spectrum of input data $\mathbf{d}$. Since, however, $\mathbf{y} = \Theta(\mathbf{x})$ is the output of a nonlinear dynamical system, Eq.(10) is true for bounded input only. It means, for example, that for neuron activation functions of sigmoidal type, the following conditions have to be fulfilled:

$$\left| \Theta(x_i) \right| \leq b_i \, , \; i = 1, 2, \ldots , 2n$$

where: b_i – asymptotical value of a sigmoid
Some orthogonal filter based transformations, given by the following formulae, are useful for further consideration:

$$\mathbf{d} = (\mathbf{W}^T + w_0 \mathbf{1})\, \mathbf{y} \tag{11}$$

product of transformations ($w_0 = 1$):

$$\mathbf{y} = \frac{1}{4}(\mathbf{W} + \mathbf{1}) \cdot (\mathbf{W} + \mathbf{1})\mathbf{d} = \frac{1}{2}\mathbf{W}\mathbf{d} \tag{12}$$

orthogonalization of outputs for given $\mathbf{d}$:

$$\mathbf{y}_1 = \frac{1}{2}(\mathbf{W} + \mathbf{1})\mathbf{d}$$

$$\mathbf{y}_2 = \frac{1}{2}(\mathbf{W}^T + \mathbf{1})\mathbf{d}$$

hence:

$$\mathbf{y}_1^T \cdot \mathbf{y}_2 = (\mathbf{y}_1, \mathbf{y}_2) = 0 \tag{13}$$

Note that the transformations given by formulae (10), (11), (12) and (13) can be regarded either as algebraic algorithms or as physically implementable HNN-based orthogonal filters. Such an implementation is guaranteed by the stabilizing action of negative feedback loops, even if the weight matrix $\mathbf{W}$ is not exactly skew-symmetric.

3. Hurwitz-Radon matrices

As mentioned above, the main issue in HNN-based orthogonal filters is forming the weight matrices $\mathbf{W}$ – skew symmetric and orthogonal. The most adequate mathematical framework

for this task seems to be an algebraic theory of Hurwitz-Radon matrices. Renewed interest in this old algebraic theory of Hurwitz-Radon matrices can be recently observed. Particularly, a link between this important old matrix problem and refined methods of algebraic topology (homology theories) has been established (Eckmann, 1999), (Vakhania, 1993). The purpose of these considerations is to show how Hurwitz-Radon matrices can be used in design of orthogonal filters. Hence, we provide, below, some basic statements from the theory of Hurwitz-Radon matrices. Let us note that a set of real N×N matrices $\mathbf{A}_j$ fulfilling the following equation (so called Hurwitz matrix equation):

$$\mathbf{A}_j^2 = -\mathbf{1}, \quad \mathbf{A}_j\mathbf{A}_k + \mathbf{A}_k\mathbf{A}_j = \mathbf{0} \tag{14}$$

for $k \neq j$, k = 1, ... , s; **1**-unit matrix
is called Hurwitz-Radon family matrices (HR family). The matrices $\mathbf{A}_j$ of family are orthogonal, i.e. $\mathbf{A}_j = -\mathbf{A}_j^T$, $\mathbf{A}_j^{-1} = \mathbf{A}_j^T$, j = 1, … , s. The maximum possible number s of family members for given dimension N is determined by the Radon number ρ(N). It can be found, as follows:
Let $N = 2^a b$, where b is an odd number and $a = 4c + d$; $0 \le d \le 4$; $c \ge 0$. Then the Radon number ρ(N) is given by:

$$\rho(N) = 8c + 2^d \tag{15}$$

and such a family consists of s_{max}(N×N)-matrices, where:

$$s_{max} = \rho(N) - 1 \tag{16}$$

Generally: $\rho(N) \le N$ and for N = 2, 4, 8 only, ρ(N) = N and $s_{max} = N - 1$.
Thus, for example, Hurwitz-Radon family of 8-dim. matrices consists of 7 matrices. The following issues in Hurwitz-Radon theory, useful for further consideration, are worth noting:

1. Algebra of complex numbers, quaternions and octonions, is directly related to Hurwitz-Radon families for N = 2, 4, 8, respectively.
2. Maximum number of continuous orthonormal tangent vector fields on sphere $S^{N-1} \in R^N$ is given by $s_{max} = \rho(N) - 1$. Moreover, let $\mathbf{A}_1$, … , $\mathbf{A}_s$ be a family of Hurwitz-Radon integer {-1, 0, 1} matrices. Let $\mathbf{A}_0 = \mathbf{1}$ and a_0, … , a_s be real numbers with $\sum_{i=1}^{s} \alpha_i^2 = 1$. Then N×N matrix:

$$\mathbf{A}(\mathbf{a}) = \sum_{i=1}^{s} a_i \mathbf{A}_i \tag{17}$$

   is orthogonal and $\mathbf{A}(\mathbf{a})$ can be considered as a map of sphere S^s into orthogonal group O(N).
3. All 8-dim. HR matrices have the following form (s_{max}=7)

$$\mathbf{H}_8 = \begin{bmatrix} 0 & h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 \\ -h_1 & 0 & h_3 & -h_2 & h_5 & -h_4 & -h_7 & h_6 \\ -h_2 & -h_3 & 0 & h_1 & h_6 & h_7 & -h_4 & -h_5 \\ -h_3 & h_2 & -h_1 & 0 & h_7 & -h_6 & h_5 & -h_4 \\ -h_4 & -h_5 & -h_6 & -h_7 & 0 & h_1 & h_2 & h_3 \\ -h_5 & h_4 & -h_7 & h_6 & -h_1 & 0 & -h_3 & h_2 \\ -h_6 & h_7 & h_4 & -h_5 & -h_2 & h_3 & 0 & -h_1 \\ -h_7 & -h_6 & h_5 & h_4 & -h_3 & -h_2 & h_1 & 0 \end{bmatrix} \tag{18}$$

where: $h_i \in R$, i =1, …, 7.

Similarly for N =16 HR family consists of s_{max}= 8 matrices and all 16-dim. matrices can be found according to the following structure:

$$\mathbf{H}_{16} = \left[\begin{array}{cc|cc} & & h_8 & \mathbf{0} \\ & \mathbf{H}_8 & & \ddots \\ & & \mathbf{0} & h_8 \\ \hline -h_8 & & & \\ & \ddots & & \mathbf{H}_8^T \\ \mathbf{0} & -h_8 & & \end{array}\right] = \begin{bmatrix} \mathbf{H}_8 & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_8 \end{bmatrix} + h_8 \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix} \tag{19}$$

where: $h_8 \in R$.

For N = 32, ρ(N) = 10 and s_{max}=9. All 32- dim. HR matrices can be found as:

$$\mathbf{H}_{32} = \begin{bmatrix} \mathbf{H}_{16} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_{16} \end{bmatrix} + h_9 \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix} \tag{20}$$

Note that the number of free parameters h_i in $\mathbf{H}_8$, $\mathbf{H}_{16}$ and $\mathbf{H}_{32}$ is equal s_{max}. For dimension $N = 2^k$, k = 6, 7, … all 2^k- dim. HR matrices can be similarly found, i.e.

$$\mathbf{H}_{2^k} = \begin{bmatrix} \mathbf{H}_{2^{k-1}} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}_{2^{k-1}} \end{bmatrix} + h_K \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix} \tag{21}$$

where: $h_K \in R$.

But, then the number of free parameters is smaller than $s_{max} = \rho(N) - 1$ ($K < s_{max}$). HR matrices of dimension $N = 2^k$ are particularly interesting due to their structures- the connections of 8-dim. blocks can be here recognized.

4. Taking into account the definition of HNN given by Eq.(2), weight matrices **W** can be implemented by using HR matrices (e.g. $\mathbf{H}_{2^k}$). Moreover, adding diagonal matrix $h_0\mathbf{1}$, where dim $\mathbf{1} = 2^k$, to skew-symmetric matrix $\mathbf{H}_{2^k}$, we obtain an implementation of the orthogonal transformation from Eq.(10), as follows:

$$\mathbf{y} = \frac{1}{1+h_0^2}(\mathbf{H}_{2^k} + h_0\mathbf{1})\mathbf{d} \tag{22}$$

where: $h_0 > 0$.

It is worth noting that for 8-dim. weight matrix $\mathbf{H}_8$, Eq.(22) describes either the following orthogonal transformation:

$$\mathbf{y} = \frac{1}{1+h_0^2}(\mathbf{H}_8 + h_0\mathbf{1})\mathbf{d} \tag{23}$$

or an equivalently 8-dim. orthogonal filter, as shown in Fig.3.

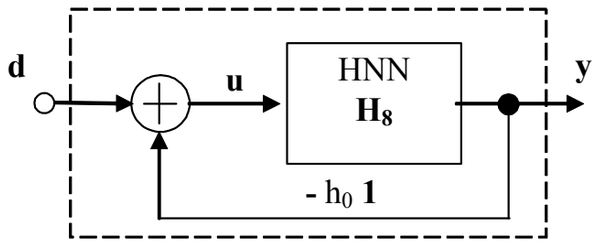


Fig. 3. Structure of 8-dim. orthogonal filter

This type of orthogonal filter will be further called the octonionic module. Because in Eq.(23) we have in disposition eight free design parameters; h_0, h_1 , … , h_7, so this equation allows us to formulate and to solve the following inverse problem: for given input vector $\mathbf{d}_0$ and given output $\mathbf{y}_0$ find parameters h_0, h_1 , … , h_7 such that $\mathbf{d}_0$ is transformed into $\mathbf{y}_0$ ($\mathbf{d}_0 \rightarrow \mathbf{y}_0$). In other words, we set up a best adapted basis for given $\mathbf{d}_0$ and $\mathbf{y}_0$. An adequate solution is given by:

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{bmatrix} = \frac{1}{\sum_{i=1}^{8} y_i^2} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \tag{24}$$

Thus, Eq.(24) can be regarded as a design formula for an octonionic module. It is interesting to note that a classical perceptron performing a scalar product of input data $\mathbf{d}$ and memory vector $\mathbf{m}$ can be implemented by the octanionic module with best adaptive basis ($\mathbf{m} \rightarrow \mathbf{y}_1 = [1, \ldots, 1]^T$), as presented in Fig. 4.

The implementation in Fig. 4 relies on a linear summing of the output flat spectrum of the orthogonal filter.

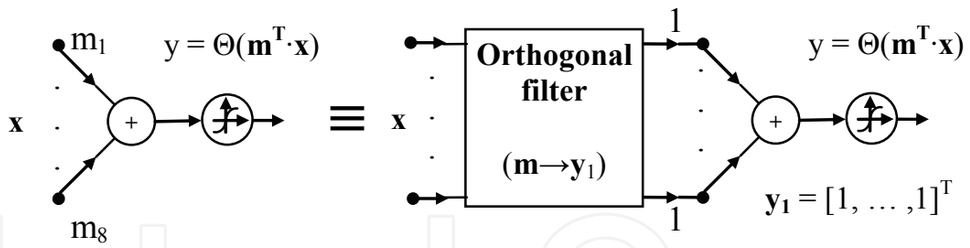


Fig. 4. Implementation of perceptron by octonionic module.

4. RLSC basics

The problem of learning from examples is about predicting the unknown class of observations generated by the underlying physical system. In the last decade the learning problems have been formalized by probabilistic setting, giving rise to statistical learning theory. As products of learning theory, some new and effective techniques, like boosting and support vector machines have been developed. On the other hand, approximation theory, supported by regularization theory, provides a new perspective on learning theory. Regularization theory has been introduced as a natural framework for solving ill-posed problems of approximation (Evgeniou et al., 2000). The purpose of this section is to provide some basic knowledge of Regularization Networks (RN) and, particularly, of RLSC, useful for further consideration. We limit ourselves to briefly describing the main ideas in a simple way.

As mentioned above, learning issues can be formulated as a problem of approximating a multivariate function from sparse data. Starting with training pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{m}$, where input vectors $\mathbf{x}_i \in X \subset R^n$ and $\mathbf{y}_i \in Y \subset R$, one can synthesize a function which represents the relation between the input **x** and **y**, in the best way. In the language of statistics this means that the probability of error $f(\mathbf{x}) \neq y$ should be minimal. According to (Evgeniou et al., 2000) the most general framework, unifying several learning techniques can be formulated by considering functionals of the form:

$$H(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_K^2 \tag{25}$$

where: $f : X \rightarrow Y$

$V(\cdot, \cdot)$- loss function

λ - regularization parameter

$\|f\|_K^2$ - norm in a Reproducing Kernel Hilbert Space (RKHS)

K - kernel (positive definite function)

The synthesized function $f(\mathbf{x})$ corresponds to the minimum of functional H for different loss functions V. Choosing square loss V (L_2 loss function):

$$V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2 \tag{26}$$

the approximation scheme arises from the minimization of quadratic functional:

$$\min_{f\in H} H(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i)) + \lambda \left\|f\right\|_K^2 \tag{27}$$

where: $\lambda > 0$;

H - Reproducing Kernel Hilbert Space (RKHS) defined by symmetric, positive definite function $K(\mathbf{x}, \mathbf{y})$

$\left\|f\right\|_K^2$ - norm in this RKHS

Thus, Eq.(27) presents the classical Tikhonow minimization problem formulated and solved in his regularization theory. It can be shown that the function that solves Eq.(27), i.e. that minimizes the regularized quadratic functional, has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{m} c_i K(\mathbf{x}, \mathbf{x}_i) \tag{28}$$

where: $\mathbf{c} = [c_1, \ldots, c_m]^T$

and kernels $K(\mathbf{x}, \mathbf{x}_i)$ are symmetric, i.e. $K(\mathbf{x}, \mathbf{x}_i) = K(\mathbf{x}_i, \mathbf{x})$, positive definite functions continuous on $X \times X$. The coefficients c_i are such as to minimize the error on the training set, i.e. they satisfy the following linear system of the equations:

$$(\mathbf{K} + \lambda \mathbf{1})\mathbf{c} = \mathbf{y} \tag{29}$$

where: $\mathbf{K}$ is square positive-definite matrix with elements $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{y}$ is the vector with coordinates y_i. The equation (29) is well-posed, hence a numerical stable solution exists:

$$\mathbf{c} = (\mathbf{K} + \lambda \mathbf{1})^{-1} \mathbf{y} \tag{30}$$

and, moreover, the regularization parameter $\lambda > 0$ determines the approximation errors. It is worth noting that:

1. an approximation is stable if small perturbations in the input data $\mathbf{x}_i$ do not substantially change the performance of the approximator. Hence, the regularization parameter λ can be regarded as the stability control factor.
2. a construction of effective kernels is a challenging task. One of the most distinctive kernels is the Gaussian:

$$K(\mathbf{x}, \mathbf{x}_i) = e^{-\left\|\mathbf{x}_i - \mathbf{x}\right\|^2 / 2\sigma^2} \tag{31}$$

leading to RBF networks.

4. Orthogonal filter-based approximation

The purpose of our considerations is to show how a function, given at limited number of training data $\mathbf{x}_i$, can be implemented in the form of composition of HNN based orthogonal filters.

Define f: X→Y by:

$$f(\mathbf{x}) = \sum_{i=1}^{m} c_i K(\mathbf{x}, \mathbf{x}_i) \tag{32}$$

where kernels $K(\mathbf{x_i}, \mathbf{x})$ are defined by the following function (induced by the activation function of neuron, Eq.(4)):

$$K(\mathbf{x}, \mathbf{x}_i) = \Theta(\mathbf{x}_i^T \mathbf{H}_n \mathbf{x}) \tag{33}$$

where: $\mathbf{x}_i^T = [x_1, \ldots, x_n], \mathbf{x}_i \in R^n$ is i-th training vector

$\mathbf{H}_n$ is skew-symmetric matrix

$\Theta(\cdot)$ is an odd function

Hence:

$$\mathbf{x}_i^T \mathbf{H}_n \mathbf{x}_i = 0 \tag{34}$$

and

$$\mathbf{x}_i^T \mathbf{H}_n \mathbf{x}_j = -\mathbf{x}_j^T \mathbf{H}_n \mathbf{x}_i \tag{35}$$

Thus, the matrix

$$\mathbf{K} = \{K_{ij}\} = \{K(\mathbf{x}_i, \mathbf{x}_j)\} \tag{36}$$

is skew-symmetric

Notice that in the case of kernels given by Eq.(33), regulizer $\|f\|_K^2$ in Eq.(26) is seminorm i.e.:

$$\begin{aligned} \|f\|_K^2 &= \left(\sum_{i=1}^m c_i K(\mathbf{x}_i, \bullet), \sum_{j=1}^m c_j K(\mathbf{x}_j, \bullet) \right) = \sum_{i=1}^m \sum_{j=1}^m c_i c_j (K(\mathbf{x}_i, \bullet), K(\mathbf{x}_j, \bullet)) = \\ &= \sum_{i=1}^m \sum_{j=1}^m c_i c_j (K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{c}^T \mathbf{K} \mathbf{c} = 0 \end{aligned} \tag{37}$$

Despite the property given by Eq.(37), we use the key approximation algorithm as formulated by Eq. (29) and (30), i.e. the regularized kernel matrix takes the form:

$$\mathbf{K}_R = (\gamma \mathbf{1} + \mathbf{K}) \tag{38}$$

where: $\gamma > 0$

$\mathbf{K}$ –skew-symmetric kernel matrix

Thus, the key design equation is well-posed:

$$\mathbf{c} = \mathbf{K}_R^{-1} \mathbf{y} = (\gamma \mathbf{1} + \mathbf{K})^{-1} \mathbf{y} \tag{39}$$

It is easy to see that the type of regularization proposed by Eq.(38) means that one changes the type of $\Theta(\cdot)$ function, in kernel definition, as follows:

$$\Theta(\bullet) \rightarrow \Theta(\bullet) + \gamma \cdot \gamma_0(\bullet) = \Theta_r(\bullet) \tag{40}$$

where: $\gamma > 0$

$\gamma_0(\cdot)$ – distribution, e.g. $\gamma_0(p) = \lim_{\delta \to 0} e^{-p^2/\delta^2}, p \in R$

In other words, the activation function $\Theta(\cdot)$ should be endowed with "a superconducting impulse γ" as shown in Fig.5.

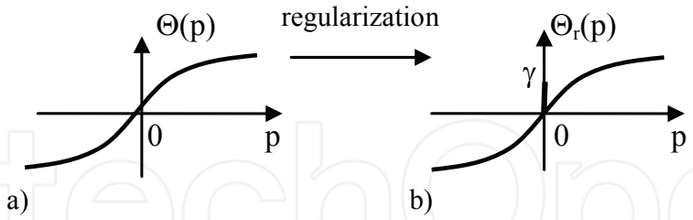


Fig. 5. Regularization by adding γ impuls.

The mechanism of stabilization by means of $\Theta_r(\cdot)$ can be easy explained when one considers the solution of Eq.(39) in a dynamical manner. Such an orthogonal filter-based structure, solving Eq.(39), is shown in Fig.6.

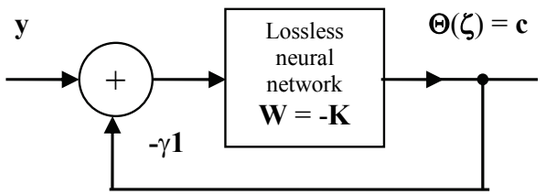


Fig. 6. Structure of orthogonal filter for solution of Eq.(39).

The state-space description of the filter from Fig.(6). is given by:

$$\dot{\boldsymbol{\varsigma}} = -(\gamma \mathbf{1} + \mathbf{K})\boldsymbol{\Theta}(\boldsymbol{\varsigma}) + \mathbf{y} \tag{41}$$

and the output in steady state as:

$$\mathbf{c} = \boldsymbol{\Theta}(\boldsymbol{\varsigma}) = (\gamma \mathbf{1} + \mathbf{K})^{-1}\mathbf{y} \tag{42}$$

The stability of approximation in the sense mentioned above can be achieved by damping influence of parameter γ. One of the possible architectures implementing approximation equation (32) is schematically shown in Fig.7 (Sienko & Zamojski, 2006).

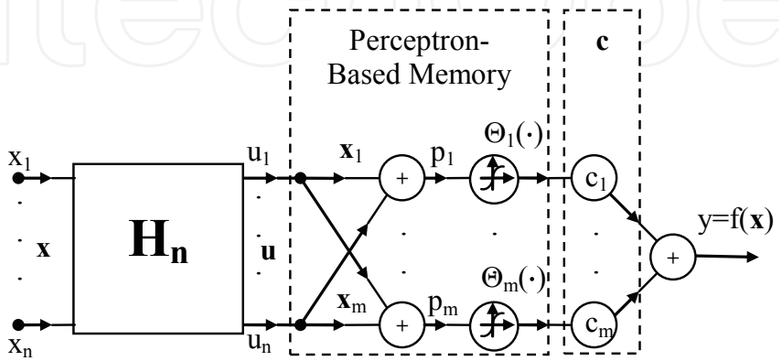


Fig. 7. Basic structure of function approximator.

This structure consists of three basic blocks:

1. Block $\mathbf{H}_n$, where matrix $\mathbf{H}_n$ is randomly skew-symmetric or $\mathbf{H}_n$ belongs to Hurwitz-Radon family, e.g. $\mathbf{H}_n = \mathbf{H}_{2^k}$ (Eq.(21)).
2. Perceptron-Based Memory consists of m perceptrons, each designed at points $\mathbf{x}_i$ of one of the m training points, for any $m < \infty$. Note that activation functions $\Theta_i(\cdot)$, i =1, …, m are odd functions (e.g. sigmoidal) allowing for error approximation at training points $\mathbf{x}_i$. Modeling a nonsmooth function only, they have to be extended by γ impulses.
3. Block of parameters c_i. Note that an implementation of a mapping $\mathbf{y} = F(\mathbf{x})$ needs l such blocks, where $l = \dim \mathbf{y}$.
   The approximation scheme, illustrated in Fig.7., can be described by:

$$\mathbf{p} = \mathbf{S}_{m,n} \cdot \mathbf{H}_n \cdot \mathbf{x} \tag{43}$$

and

$$f(\mathbf{x}) = \mathbf{\Theta}^T(\mathbf{p}) \cdot \mathbf{c} \tag{44}$$

where: $\mathbf{H}_n$- (nxn) skew-symmetric matrix
$\mathbf{S}_{m,n}$ – (m×n) memory matrix, $\mathbf{S}_{m,n} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]^T$
$\mathbf{\Theta}(\mathbf{p}) = [\Theta(p_1), \Theta(p_2), \ldots, \Theta(p_m)]^T$
$\mathbf{c} = [c_1, c_2, \ldots, c_m]^T$

Another orthogonal filter-based structure of function approximator, is shown in Fig.8 (Sienko & Citko 2007).

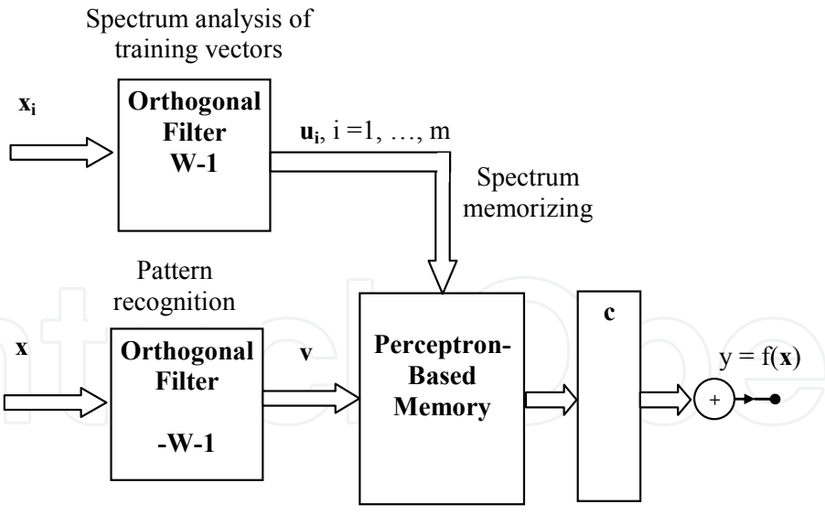


Fig. 8. Orthogonal Filter-Based structure of an approximator.

The structure of the approximator shown in Fig.8. relies on using the skew-symmetric kernels, as given by:

$$K(\mathbf{u}_i, \mathbf{v}) = \Theta(\mathbf{u}_i^T \mathbf{v}) \tag{45}$$

where: $\mathbf{u}_i = (\mathbf{W\text{-}1})\, \mathbf{x}_i$

$\mathbf{v} = -(\mathbf{W}+\mathbf{1})\,\mathbf{x}$
$\Theta(\bullet)$ is an odd function

Assuming: $\mathbf{W}^2 = -\mathbf{1}$, $\mathbf{W}^T\mathbf{W} = \mathbf{1}$ i.e. $\mathbf{W}$- skew-symmetric, orthogonal e.g. $\mathbf{W} = \mathbf{H}_{2^k}$ Hurwitz-Radon matrix, Eq.(21).
Then: $\mathbf{u}_i$, $\mathbf{v}$ – Haar spectrum of input $\mathbf{x}_i$ and $\mathbf{x}$, respectively.
thus, elements of kernel matrix fulfill:

$$K(\mathbf{u}_i, \mathbf{v}_j) = \Theta(\mathbf{u}_i^T\mathbf{v}_j) = \Theta(2\mathbf{x}_i^T\mathbf{W}\mathbf{x}_j)$$

and

$$K(\mathbf{u}_i,\mathbf{v}_j) = -K(\mathbf{v}_j, \mathbf{u}_i) \tag{46}$$

Hence, matrix
$\mathbf{K} = \{K_{ij}\} = \{K(\mathbf{u}_i, \mathbf{v}_j)\}$ is skew-symmetric.

Note that for the structure from Fig.8., the same key design equation (39) is relevant. However, the structure from Fig.8. can be seen as HNN-based dynamically implemented system, as well. Moreover, taking into account the implementation presented in Fig.4. one can formulate the following statement:

Statement 1

Orthogonal filter-based structures of function approximator can be implemented by compatible connections of octonionic modules.

Other important remarks concluding the above described approximation scheme can be formulated as follows:

Statement 2

Due to the skew-symmetry of kernel matrix, the orthogonal filter based approximation scheme can be regarded as a global method. It means that the neighborhood of the training point $\mathbf{x}_i$ is reconstructed by all the other training points. Exceptionally, this global method is completed by a pointwise local one, if the activation function of used perceptrons has a form $\Theta_r(\cdot)$ (Fig.5.).

Statement 3

Orthogonal filter-based approximation scheme can be easy reformulated as a local technique. Indeed, taking into considerations the kernel defined by Eq.(33), where activation function is an even function e.g. Gaussian function:

$$\Theta(p) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\frac{p^2}{\sigma^2}} \tag{47}$$

where: $p \in R$
then the kernel matrix

$$\mathbf{K}_S = \{K_{ij}\} = \{K(\mathbf{x}_i, \mathbf{x}_j)\} = \{\Theta(\mathbf{x}_i^T\mathbf{H}_n\mathbf{x}_j)\} \tag{48}$$

is a symmetric, positive matrix.

For $\Theta(p)$ given by Eq.(47) matrix $\{K_{ij}\}$ fulfils:

$K_{ii} > 0$ for all i,
$K_{ii} > K_{ij}$ for $i \neq j$
and there is such a $\sigma > 0$ that $\det \mathbf{K_s} > 0$.
Thus, matrix $\mathbf{K_s}$ is positive definite.
Hence, it is clear that the key design equation is well-posed:

$$\mathbf{c} = \mathbf{K}_s^{-1}\mathbf{y} \tag{49}$$

and the local properties of this approximation scheme can be controlled by parameter σ. It should be however noted that positive definiteness is not necessary for $\det \mathbf{K_s} > 0$ and for existing an inverse $\mathbf{K_s}^{-1}$. To summarize this section, let us note that by choosing a different type of activation functions, one generates a family of functions or mappings fulfilling:

$$\mathbf{y}_i = F_q(\mathbf{x}_i),\ i = 1, \ldots, m;\ q = 1, 2, \ldots$$

To minimize the approximation errors, one should select a function or mapping which, in terms of learning, optimally transforms a neighborhood $S(\mathbf{x}_i)$ of $\mathbf{x}_i$ onto $\mathbf{y}_i$.

5. Modeling classifiers and associative memories

As mentioned in the previous section, an approximation of a mapping can be obtained as an extended structure of a multivariate function approximation. Hence, for the sake of generalization, we below use a notation of mapping approximation.

Define mapping $F: X \rightarrow Y$

where X, Y are input and output training vector spaces, respectively. The values of mapping are known at training points $\{\mathbf{x}_i, \mathbf{y}_j\}_{i=1}^{m}$ where, dim $\mathbf{x}_i$ = n and dim $\mathbf{y}_i$= l:
Thus:

$$\mathbf{y}_i = F(\mathbf{x}_i) \quad i = 1, \ldots, m \tag{50}$$

where: $\mathbf{x}_i \in X, \mathbf{y}_i \in Y$
Classification issues can be seen as a special problem in mapping approximations. If output vectors **y** of mapping $F(\cdot)$ take values from an unordered finite set, then $F(\cdot)$ performs the function of a classifier. In a two-class classification, one class is labeled by y = 1 and the other class by y = -1. The general functionality of classifiers can be then determined by the following equation:

$$F(\tilde{\mathbf{x}}_i) = \mathbf{y}_i,\ i = 1, \ldots, m \tag{51}$$

where: $\tilde{\mathbf{x}}_i$ denotes a neighborhood of "center" $\mathbf{x}_i$
$\mathbf{y}_i$ – class label
The determination of neighborhoods $\tilde{\mathbf{x}}_i$ depends on the application of a classifier, but generally, to minimize the erroneous classifications, $\tilde{\mathbf{x}}_i$ have to be densely covered by spheres belonging to $\tilde{\mathbf{x}}_i$, i = 1, … , m. Thus, the problem of classifier design can be formulated as follows:

1. generate a family of mappings $F_q(\cdot)$, q =1, 2, … fulfilling:

$$X \xrightarrow{F_q} Y, \quad F(\tilde{\mathbf{x}}_i) = \mathbf{y}_i \tag{52}$$

where X, Y are input and output training vector spaces, respectively.
Members of this family are created by choosing different type of kernels (antisymmetric or symmetric) and different values of regularization parameters γ or σ

2. select the mapping that transforms input points onto output vectors in an optimal way (minimizing approximation errors):

$$\mathbf{x}(\in X) \xrightarrow{F_{opt}} \mathbf{y}(\in Y)$$

The problem of optimal mapping selection has been recently formulated in the framework of statistical operators on family (52) (e.g. bagging and boosting techniques). We propose here to consider an optimal solution as a superposition of global and local schemes. In the simplest case, we have the following equation:

$$F_{opt}(\bullet) = (1-\alpha)F_G(\bullet) + \alpha F_L(\bullet) \tag{53}$$

where: weight parameter α; $0 \le \alpha \le 1$.
and

$F_G(\bullet)$ - a global model of mapping obtained by using antisymmetric kernels Eq.(33) and Eq.(45)

$F_L(\bullet)$ - a local model of mapping obtained by using symmetric kernels, Eq.(48).

The relation (53) is motivated by the general properties of dynamical systems: a vector field $F(\cdot)$ underlying a physical law, object or process generally consists of two components-global and local (recombination and selection in biological systems, respectively).

To illustrate the considerations above, let us consider the following example:

Example1

Let us design a classification of 8-dim. vector input space X, where $\mathbf{x} = [x_1, x_2, \dots, x_8]^T$, $x_k \in [-1, 1]$, k = 1, … , 8. into 2^5 classes centered in randomly chosen points: $\mathbf{x}_i$, i =1, … , 32. This classification has to be error free, with probability 1, for solid spheres $\mathbf{x} \in S_\rho(\mathbf{x}_i)$, where ρ(radius) = 0.2. It has been experimentally found (i.e. by simulation) that covering randomly every sphere $S_\rho(\mathbf{x}_i)$ with 10 balls, such a classifier design can be reformulated as the following mapping approximation (n = 8, m = 320-number of inputs points):

$$F(\mathbf{x}_{ij}) = \mathbf{y}_i, \; i = 1, \dots, 32; \; j = 1, \dots, 10$$

where: $\mathbf{y}_i = [\pm 1, \pm 1, \pm 1, \pm 1, \pm 1]^T$ (binary label of classes)
The set of input points is given by:

$$\{\mathbf{x}_{ij}\} \in S_\rho(\mathbf{x}_i), \; i = 1, \dots, 32, \; j = 1, \dots, 10$$

where: ρ = 0.2
To implement the above defined mapping $F(\mathbf{x}_{ij})$, let us choose the antisymmetric kernels Eq.(33), where:

$$\Theta(p) = 5\left(\frac{2}{1+e^{-3p}} - 1\right), \quad p \in R$$

and

$$\mathbf{H}_8 = \frac{1}{\sqrt{7}}\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 0 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 0 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 0 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 0 & 1 \\ -1 & -1 & 1 & -1 & 1 & 1 & -1 & 0 \end{bmatrix}$$

Some simulation experiments showed that the mapping $F(\mathbf{x}_{ij})$ fulfils formulated constraints on classification for the case: min $d(\mathbf{x}_i, \mathbf{x}_j) \geq 0.7$ (distance between sphere centers) and under condition that regularization parameters $\gamma \geq 0.75$ (Eq.(38)).
Equation (51) can be seen as a definition of associative memory as well, under the assumption that dim $\mathbf{x}_i$ = dim $\mathbf{y}_i$, where $\mathbf{x}_i$ is a memorized pattern. For $\mathbf{y}_i \equiv \mathbf{x}_i$, one gets a feedforward structure of an autoassociative memory, i.e.:

$$F(\tilde{\mathbf{x}}_i) = \mathbf{x}_i, \; i = 1, \ldots, m \tag{54}$$

Hence, the problem of a nonlinear mapping-based design of the associative memory can be regarded as a covering problem of input space X by spheres $S_\rho(\mathbf{x}_i)$.
Moreover, Eq.(54) determines an identity map i.e. :

$$F(\mathbf{x}_i) = \mathbf{x}_i, \; i = 1, \ldots, m \tag{55}$$

and $F(\cdot)$ is an expansion.
Hence, the mapping $F(\cdot)$ possesses at least one fixed point, i.e. :

$$F(\mathbf{e}) = \mathbf{e} \tag{56}$$

where: **e**- a fixed point of $F(\cdot)$
Specifically, let us construct the family of identity maps for orthogonal vectors $\mathbf{h}_i$, i =1, …, 8, constituting eight columns of matrix $\mathbf{H}_8$ in Eq.(18), i.e.:

$$F_q(\mathbf{h}_i) = \mathbf{h}_i, \; i = 1, \ldots, m; \; q = 1, 2, \ldots \tag{57}$$

using antisymmetric kernels Eq.(33), $\mathbf{h}_i \in R^8$.
It can be shown that in family (57) there are mappings $F_q(\cdot)$ with the number of fixed points $n_e \leq 256$ (e.g. n_e =144), giving rise to a feedback structure of associative memories. Indeed, let us embed such a $F_q(\cdot)$ into a dynamical system, as shown in Fig. 9.
The state-space equation of structure from Fig.9. is given by:

$$\dot{\varsigma} = -\beta\varsigma + F_q(\varsigma) \tag{58}$$

where: ς - 8-dim. state vector, $0 < \beta \leq 1$.

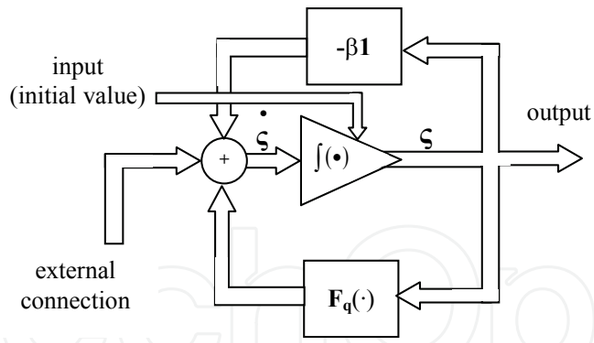


Fig. 9. Dynamical structure of an attractor type associative memory.

Thus, one obtains a feedback type structure of an associative memory with e.g. over 144 asymptotic stable equilibria, but generally with different diameters of attraction basins. Unfortunately the set $\{\mathbf{e}_k\}$of fixed points of a map $F(\cdot)$, can not be found analitically but rather by a method of asymptotic sequences. This can be done relatively simply for 8-dim. identity map presented by Eq.(57) and (58). Thus, due to the exceptional topological properties of a 8-dim vector space, very large scale associative memories could be implemented by a compatible connection of the 8-dim. blocks from Fig.9. An example of such a connection is presented in Fig.10, where two 8-dim. blocks from Fig.9., weakly coupled by parameters $\varepsilon_i > 0$, create a space with a set of equilibria given by:

$$\left\{\mathbf{e}_c\right\} = \left\{\mathbf{e}_k^{(1)}\right\} \times \left\{\mathbf{e}_j^{(2)}\right\} \text{ where: } k = 1, 2, \ldots, 144, \ldots; j = 1, 2, \ldots, 144 \ldots$$

Finally, it is worth noting that the structure from Fig.10 can be scaled up to very large scale memory (by combinatorial diversity), due to its stabilizing type of connections (parameters ε_i). More detailed analysis of the above presented feedback structures is beyond the scope of this chapter.

To summarize, this section points out the main features of orthogonal filter-based mapping approximators:

1. Due to regularization and stability, orthogonal filter-based classifiers can be implemented for any even n (dimension of input vector space) and any $m < \infty$ (number of training vectors). Particularly for $n = 2^k$, $k \geq 3$ such classifiers can be realized by using octonionic modules.
2. As mentioned above, the problem of a nonlinear mapping-based design of classifiers and associative memories can be regarded as a covering problem of input space X by spheres with centers $\mathbf{x}_i$. The radius of the spheres needed to cover X depends on the topology of X and can be changed by a suitably chosen nonlinearity of function $\Theta(\cdot)$. Using, for example, a sigmoidal function for the implementation of $\Theta(\cdot)$., this radius depends on the slope of $\Theta(\cdot)$ at zero. Hence, note that antisymmetric kernels allow us to classify very closely placed input patterns in terms of $\Theta(\cdot) \rightarrow \mathrm{sgn}(\cdot)$.

6. Conclusions

The main issue considered in this chapter is the deterministic learning of mappings. The learning method analysed here relies on multivariate function approximations using mainly skew-symmetric kernels, thus giving rise to very large scale classifiers and associative memories. By using HNN-based orthogonal filters, one obtains regularized and stable structures of networks for learning. Hence, classifiers and memories can be implemented for

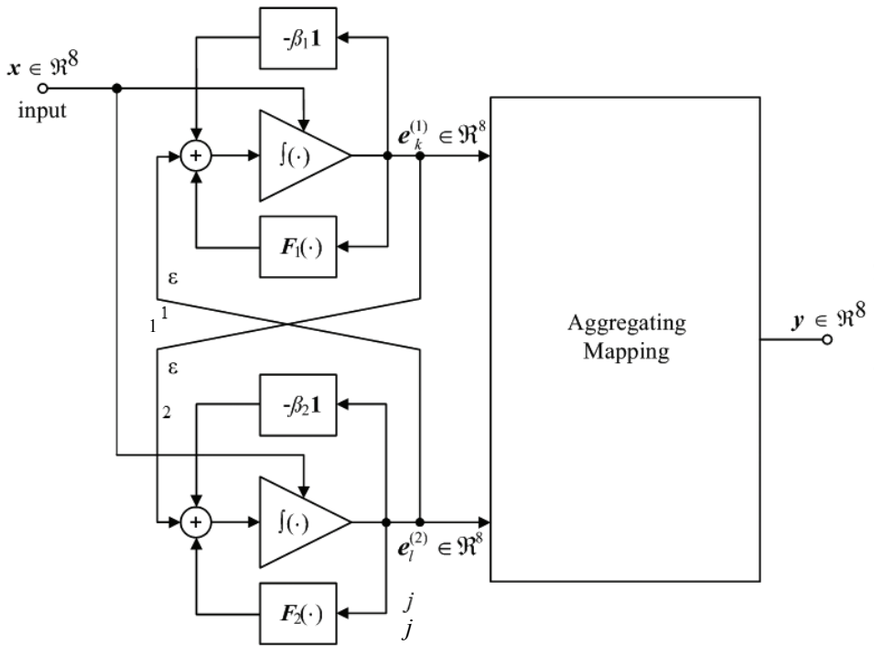


Fig. 10. Very Large Scale Structure of associative memory.

any even n (dimension of input vectors) and any m < ∝ (number of training patterns). Moreover, they can be regarded as numerically well-posed algorithms or physically implementable devices able to perform their functions in real-time. We believe that orthogonal filter-based data processing can be considered as motivated by structures encountered in biological systems.

6. References

Boucheron, S.; Bousquet, O. & Lugosi, G. (2005). Theory of Classification: A survey of some Resent Advances, ESAIM: Probability and Statistics, pp. 323-375.

Eckmann, B. (1999). Topology, Algebra, Analysis-Relations and Missing Links, *Notices of the AMS*, vol. 46, No 5, pp. 520-527.

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

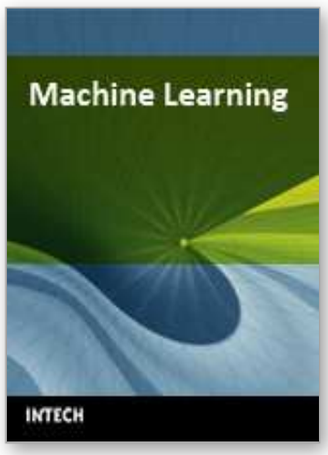
Poggio, T. & Smale, S. (2003). The Mathematics of Learning. Dealing with Data, *Notices of the AMS*, vol. 50, No 5, pp. 537-544.

Predd, J.; Kulkarni, S. & Poor, H. (2006). Distributed Learning in Wireless Sensor Networks, *IEEE Signal Processing Magazine*, vol. 23, No 4, pp.56-69.

Sienko, W. & Citko, W. (2007). Orthogonal Filter-Based Classifiers and Associative Memories, *Proceedings of International Joint Conference on Neural Networks*, Orlando, USA, pp. 1739-1744.

Sienko, W. & Zamojski, D. (2006). Hamiltonian Neural Networks Based Classifiers and Mappings, *Proceedings of IEEE World Congress on Computational Intelligence*, Vancouver, Canada, pp. 1773-1777.

Vakhania, N. (1993). Orthogonal Random Vectors and the Hurwitz-Radon-Eckmann Theorem, *Proceedings of the Georgian Academy of Sciences, Mathematics*, 1(1), pp. 109-125.



Machine Learning

Edited by Abdelhamid Mellouk and Abdennacer Chebira

ISBN 978-953-7619-56-1
Hard cover, 450 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Machine Learning can be defined in various ways related to a scientific domain concerned with the design and development of theoretical and implementation tools that allow building systems with some Human Like intelligent behavior. Machine learning addresses more specifically the ability to improve automatically through experience.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wieslaw Sienko and Wieslaw Citko (2009). Hamiltonian Neural Networks Based Networks for Learning, Machine Learning, Abdelhamid Mellouk and Abdennacer Chebira (Ed.), ISBN: 978-953-7619-56-1, InTech, Available from: http://www.intechopen.com/books/machine_learning/hamiltonian_neural_networks_based_networks_for_learning

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen