INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 1, PP. 1-6
Manuscript received November 06, 2025; revised January, 2026. doi: 10.24425/ijet.2026.157887

Signal processing using octonionic modules – a path towards a new computational intelligence model

Wieslaw Citko, and Wieslaw Sienko

***Abstract*—This study proposes a modular structure designed for pattern and word sequence recognition. The developed structure is based on an extended Hopfield neural network. The architecture of the word sequence recognition system employs octonionic modules, which are implemented as transversal filter banks. The structure can be used to recognize word sequences containing data represented as both real and complex numbers. The proposed procedure for synthesizing the word sequence recognition system may be useful for the development of computational intelligence systems.**

***Keywords*—computational intelligence; Hopfield neural networks; word sequence recognition**

I. Introduction

HIGH-DIMESIONAL and large-scale datasets have become essential in the application of signal processing. In particular, generative AI-driven models (e.g., high-dimensional imaging) pose technological challenges in the selection of suitable deep-learning architectures. Currently, transformers seemingly outperform other neural architectures, such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CCN) [1, 2]. However, the structure of transformers has been proposed as a language model, while other applications have been suggested in the literature, e.g., vision transformers. It is understood that most deep learning algorithms are implemented within the theory of optimization methods. Nevertheless, the optimal network technology has not yet been determined. In this study, Hopfield Neural Networks are proposed as a neural computing architecture. In a previous study, we proposed an extended Hopfield neural network model defined by the following equation [3, 4, 5]:

$$\dot{\boldsymbol{x}} = (\eta \boldsymbol{W} - w_0 \mathbf{1} + \varepsilon \boldsymbol{W}_s)\boldsymbol{\theta}(\boldsymbol{x}) + \boldsymbol{d} \quad (1)$$

where: $\boldsymbol{W}$ – antisymmetric orthogonal matrix,
$\boldsymbol{W}_s$ – real symmetric matrix,
$\mathbf{1}$ – identity matrix,
$\boldsymbol{\theta}(\boldsymbol{x})$ – activation functions ($\theta(0) = 0$),
$\boldsymbol{d}$ – input vector,
ε, w_0, η – parameters.

The equilibrium state of network (1) takes the following form:

$$(\eta \boldsymbol{W} - w_0 \mathbf{1} + \varepsilon \boldsymbol{W}_s)\boldsymbol{\theta}(\boldsymbol{x}) + \boldsymbol{d} = \mathbf{0} \quad (2)$$

Equation (2) constitutes the basis for universal machine learning models based on biorthogonal transformations, enabling the implementation of common learning systems functions. One of these functions is an associative memories implementation. The use of the system for reconstruction and recognition of distorted or noisy images by employing associative memory was described in previous studies [3, 4, 5]. Of note, (2) enables the processing of complex-valued vectors or images when the $\boldsymbol{W}_s$ matrix becomes complex-valued. Moreover, for $\varepsilon = 0$, the solution of (2) provides the structure of the orthogonal transformations described below for signal processing.

II. Signal Processing Based on Octonionic Modules

One of the forms of the neural network in (1) can be written as follows:

$$\dot{\boldsymbol{x}} = (\boldsymbol{W} - w_0 \mathbf{1})\boldsymbol{\theta}(\boldsymbol{x}) + \boldsymbol{d} \quad (3)$$

where: $\boldsymbol{W}$ – skew-symmetric orthogonal matrix,
$\mathbf{1}$ – identity matrix,
$\boldsymbol{\theta}(\boldsymbol{x})$ – activation functions,
$\boldsymbol{d}$ – input vector,
w_0 – parameter.

The stable equilibrium state of network (3) sets up an orthogonal transformation:

$$\boldsymbol{\theta}(\boldsymbol{x}) = \boldsymbol{y} = \frac{1}{1+w_0^2}(\boldsymbol{W} + w_0 \mathbf{1})\boldsymbol{d} \quad (4)$$

where: $\boldsymbol{W}^2 = -\mathbf{1}$ and $\boldsymbol{y}$ is a Haar spectrum of $\boldsymbol{d}$,
$\boldsymbol{y}$ – output vector.

We can see that for $w_0 = 0$ in (3), the result is the structure of lossless/Hamiltonian neural networks (HNN) [6].

Note 1

A Haar spectrum is the result of a Haar transformation, where the transformation matrix {-1, 0, 1} is orthogonal but not skew-symmetric. By contrast, the main challenge in HNN-based

Wieslaw Citko is with the Department of Marine Telecommunications, Faculty of Electrical Engineering, Gdynia Maritime University, Poland (e-mail: w.citko@we.umg.edu.pl).

Wieslaw Sienko is with the Department of Marine Telecommunications, Faculty of Electrical Engineering, Gdynia Maritime University, Poland (e-mail: w.sienko@we.umg.edu.pl).

orthogonal transformations is to create weight matrices, $\boldsymbol{W}$, that are both skew-symmetric and orthogonal. The most suitable mathematical framework for this task can be an algebraic theory of Hurwitz-Radon matrices [7]. Hence, we show how Hurwitz-Radon matrices can be used in the construction of orthogonal transformations (filters) by defining matrices $\boldsymbol{W}$ as the superposition of Hurwitz-Radon matrices. Moreover, only the matrix $\boldsymbol{W}_8$ has available eight free design parameters, $w_0, w_1, \dots, w_7$, to synthesize any eight-dimensional orthogonal filter and solve the synthesis problem. Thus, an eight-dimensional orthogonal transformation, referred to as an *octonionic module*, can be synthesized by the following formula:

$$\boldsymbol{y} = \boldsymbol{H}_8 \boldsymbol{d} \tag{5}$$

where: $\boldsymbol{H}_8 = \frac{1}{a^2}(\boldsymbol{W}_8 + w_0 \boldsymbol{1})$ - transformation matrix of the octonionic module,
$a = \sqrt{\sum_{i=0}^{7} w_i^2}$ - scaling parameter.

The weight matrix $\boldsymbol{W}_8$ of octionic module is written as:

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

and

$$\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} = \frac{1}{\sum_{i=1}^{8} y_i^2} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \tag{7}$$

We can see that (7) is a solution for the following synthesis problem:
For a given input vector $\boldsymbol{d} = [d_1, \dots, d_8]^T$ and a given output vector $\boldsymbol{y} = [y_1, \dots, y_8]^T$, find the weight matrix $\boldsymbol{H}_8$ of an NN-based orthogonal transformation (octonionic module). Matrix $\boldsymbol{H}_8$, belonging to the family of matrices, can be obtained by the superposition of seven Hurwitz-Radon matrices. Moreover, $\boldsymbol{H}_8$ can be observed as the best-adapted orthogonal basis. The output $\boldsymbol{y}$ in (5) is a Haar spectrum of the input vector $\boldsymbol{d}$. Of note, an octonionic module sets up an elementary memory module as well. For example, designing an orthogonal filter using (6) and (7), performs the following transformation:

$$\boldsymbol{y}_{[1]} = \frac{1}{a^2}(\boldsymbol{W_8} + \boldsymbol{w_0}\boldsymbol{1})\boldsymbol{m} = \boldsymbol{H}_8(\boldsymbol{m})\,\boldsymbol{m} \tag{8}$$

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

$$w_0 = [1, \dots, 1] \cdot \boldsymbol{m} > 0, \text{ i.e. } \sum_{i=1}^{8} m_i > 0 \tag{9}$$

yields an implementation of a real memory module, where $\boldsymbol{H}_8(\boldsymbol{m})$ is the orthogonal matrix.
Thus, when: $\boldsymbol{y} = \boldsymbol{H}_8(\boldsymbol{m})\boldsymbol{x}$, for $\boldsymbol{x} \neq \boldsymbol{m}$ and $\|\boldsymbol{x}\| = \|\boldsymbol{m}\|$,then $\|\boldsymbol{y}\| > \|\boldsymbol{y}_{[1]}\|$. This inequality can be used as a valuable tool for pattern recognition.

To summarize, the octonionic module is a universal building block producing very large-scale orthogonal filters and, specifically, memory blocks. Multidimensional octonionic module-based orthogonal filters can be generated using the family of Hurwitz-Radon matrices. Thus, a 16-dimensional orthogonal filter, for example, can be determined by the following matrix:

$$\boldsymbol{H}_{16} = \begin{bmatrix} \boldsymbol{H}_8 & \begin{bmatrix} -w_8 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -w_8 \end{bmatrix} \\ \begin{bmatrix} -w_8 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -w_8 \end{bmatrix} & \boldsymbol{H}_8^T \end{bmatrix} \tag{10}$$

where: $w_8 \in R$, $\boldsymbol{H}_8$ - weight matrix of an octonionic module.

Similarly, for the dimension $q = 2^k, k = 5, 6, 7, \dots$ all matrices can be determined as:

$$\boldsymbol{H}_{2^k} = \begin{bmatrix} \boldsymbol{H}_{2^{k-1}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_{2^{k-1}}^T \end{bmatrix} + w_K \begin{bmatrix} \boldsymbol{0} & \boldsymbol{1} \\ -\boldsymbol{1} & \boldsymbol{0} \end{bmatrix} \tag{11}$$

where: $w_K \in R$, $\boldsymbol{1}$ - identity matrix.

This analysis supports the following statements:

- A q-dimensional HNN or a q-dimensional orthogonal basis can be created by a compatible connection of octonionic modules.
- The basic function of orthogonal filters is the Haar spectrum analysis of the input data $\boldsymbol{d}$. In particular, an orthogonal filter performs the memory function, as given by (8).

Matrix $\boldsymbol{H}_{\boldsymbol{2^k}}$ can be designed as the best-adapted base by using (6) and (7) (i.e., given $\boldsymbol{d}$ - data, $\boldsymbol{y}$ - demanded spectrum).

Note 2
The synthesis problem formulated in (6) and (7) for real vectors $\boldsymbol{d}$ and $\boldsymbol{y}$ can be used for complex-valued vectors as well, obtaining the complex-valued matrix $\boldsymbol{H}$. This thereby facilitates signal processing with complex-valued wavelets and complex memories $\boldsymbol{H}_8(\boldsymbol{m})$, $\boldsymbol{m}$ – complex-valued vector.

III. Pattern/Sentence Recognition

One of the tasks performed by the Computational Intelligence System, which can be classified as pattern recognition, obtains the following formulation: In the given text, $d(k)$; $k = 0, 1, \dots, N$, find a sentence that is encoded as a sequence of tokens. Without limitation of consideration generality, the searched sentence is assumed to have a length of eight complex numbers:

$$\boldsymbol{d}_p = \left[d_{p1,}, d_{p2,}, \dots, d_{p8,}\right]^T, \boldsymbol{d}_p \in C \quad (12)$$

Synthesizing the matrix $\boldsymbol{H}_8$ of complex memory for given $\boldsymbol{d}_p$, i. e., $\boldsymbol{H}_8(\boldsymbol{m}) = \boldsymbol{H}_8(\boldsymbol{d}_p)$, Equation (8) becomes:

$$\boldsymbol{d}(k_0) = \begin{bmatrix} d(k_0) \\ d(k_0 - 1) \\ \vdots \\ d(k_0 - 7) \end{bmatrix} = \boldsymbol{d}_p \quad (13)$$

If for $k = k_0$ the input sequence fulfills (13), then the output spectrum is $\boldsymbol{y}_{[1]} = [1, 1, \dots, 1]^T$, meaning a searched sequence is recognized. The structure shown in Fig. 1 can be employed as the sequence recognizer.

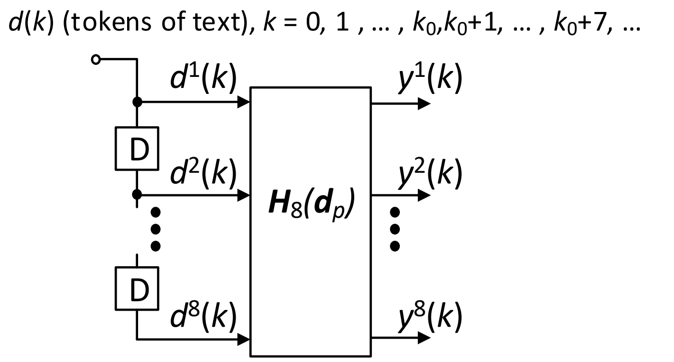

Fig. 1. Structure of the sequence recognizer

A structure such as that shown in Fig. 1 can be parallel-extended to search for multiple sentences in a given text $d(k)$. This structure can be seen as a connection of eight transversal (FIR) filters with the rows of matrix $\boldsymbol{H}_8$ as the impulse responses.

<u>Example 1</u>

The following computation illustrates the robustness of octonionic modules-based memory.

For a given complex memory:

$$\boldsymbol{m} = \left[e^{0.89j}, e^{-1.06j}, e^{-1.20j}, e^{0.84j}, e^{1.13j}, e^{1.67j}, e^{-1.57j}, e^{-1.28j}\ \right]^T$$

The octonionic module is synthesized as shown in Fig. 2a.

Transforming a randomly selected input vector gives:

$$\boldsymbol{x} = [e^{-1.22j}, e^{-0.08j}, e^{0.18j}, e^{-2.91j}, e^{0.54j}, e^{-1.24j}, e^{-0.06j}, e^{0.42j}]^T$$

which obtains a spectrum:

$$\boldsymbol{y} = [1.53e^{1.81j}, 1.74e^{-1.95j}, 2.28e^{-2.79j}, 1.89e^{0.99j}, 0.86e^{1.97\,j}, 2.40e^{-2.16j}, 1.41e^{-1.38j}, 1.34e^{-2.71j}]^T$$

as shown in Fig. 2b.

As mentioned above, the inequality is fulfilled:

$$\left\|\boldsymbol{y}_{[1]}\right\| = 8 < \|\boldsymbol{y}\| = \sum_{i=1}^{8}|y_i| = 13{,}50 \quad (14)$$

The inequality from (14) has been noted in computational experiments with text encoded by hundreds of random tokens. This result, given the sentence $\boldsymbol{d}_p$ recognition, provides justification for the key data for pattern recognition, i.e., $\left\|\boldsymbol{y}_{[1]}\right\| < \|\boldsymbol{y}\|$.

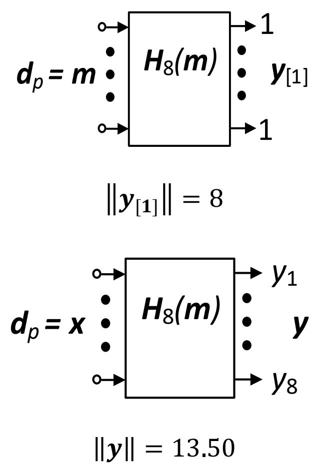

Fig. 2. Octonionic modules
a) memorized vector as input
b) random vector as input

Moreover, it is worth noting how the parts of $\boldsymbol{d}(p)$ sequences are transformed by the system from Fig. 1. This results in the maps for the following inputs:

$$\boldsymbol{x}_1 = [0, e^{0.89j}, e^{-1.06j}, e^{-1.20j}, e^{0.84j}, e^{1.13j}, e^{1.67j}, e^{-1.57j}]^T, \quad \|\boldsymbol{y}_1\| = 11.67;$$

$$\boldsymbol{x}_2 = [e^{-1.06j}, e^{-1.20j}, e^{0.84j}, e^{1.13j}, e^{1.67j}, e^{-1.57j}, e^{-1.28j}, 0]^T, \quad \|\boldsymbol{y}_2\| = 10.30;$$

$$\boldsymbol{x}_3 = [e^{0.89j}, e^{-1.06j}, e^{-1.20j}, 0\ , e^{1.13j}, e^{1.67j}, e^{-1.57j}, e^{-1.28j}]^T, \quad \|\boldsymbol{y}_3\| = 8.76;$$

Thus:

$$\left\|\boldsymbol{y}_{[1]}\right\| < \|\boldsymbol{y}_1\|, \|\boldsymbol{y}_2\|, \|\boldsymbol{y}_3\| \quad (15)$$

The inequality in (15) provides further evidence of the robustness of the sequence recognizer.

The next experiment involved inserting a test sequence of normalized complex numbers (with a magnitude of one) into a sequence of $d(k)$ random complex numbers. This test sequence was used, according to the procedure described in (5) – (8), for the synthesis of the octonionic module $\boldsymbol{H}_8(\boldsymbol{m})$. The sequence of random numbers containing the test sequence was then used as input into the module shown in Fig. 1. For each step, the norm of the output vector $\|\boldsymbol{y}\|$ was calculated. Fig. 3 shows the change in the value of the norm $\|\boldsymbol{y}_k\|$ in successive steps. When the memorized sequence appears at the module's input, the norm of the output vector should reach its minimum value—in this case, 8.

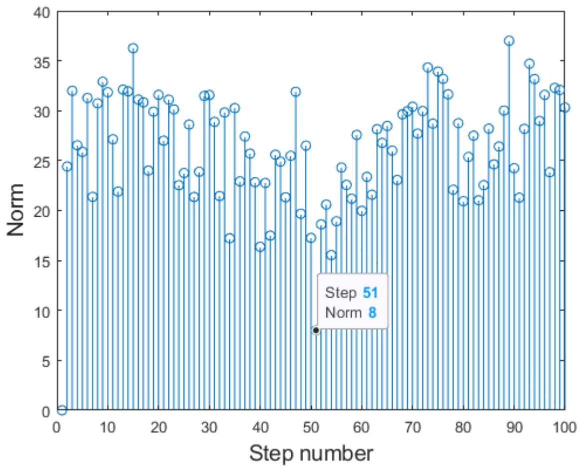

Fig. 3. Relationship between the value of the norm and the step number

Based on the analyzed graph, it is evident that the desired sequence appeared at the input in step 51.

IV. A Path Towards Computational Intelligence Mimicking Darwin Machines

As mentioned in preceding sections, synthesis procedures presented above can be relevant for computational intelligence. Moreover, the use of complex or real valued octonionic modules gives rise to a neural network architecture mimicking the features of so-called Darwin machines, i.e., evolution processes led by recombination and selection forces[. Indeed it is worth noting that forces as antisymmetric (recombination) and symmetric (selection) components determine the structure of the extended Hopfield neural network [8, 9]. A Darwin machine network can be constructed by using the octonionic modules from Fig. 1. and Fig. 2. when the structure of sequence recognizer is augmented to form of selective perceptron, as shown in Fig. 4.

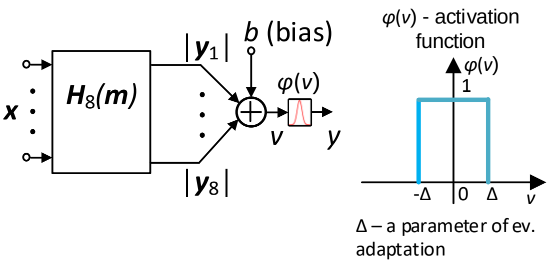

Fig. 4. A selective perceptron (b = - 8)

It is clear that induced local field of the perceptron i.e. number v fulfils:

$$\mathrm{v} = \|\boldsymbol{y}\| + b = \sum_{i=1}^{8} |y_i| + b \tag{16}$$

Thus, assuming an appropriately selective activation function $\varphi(\cdot)$, one obtains $y = 1$ for input vector $\boldsymbol{x} = \boldsymbol{m}$, and $y = 0$ otherwise (for $\boldsymbol{x} \neq \boldsymbol{m}$). The structure of the perceptron shown in Fig. 4 is scalable through the use of multidimensional block-diagonal matrices. For example:

$$\boldsymbol{H}_{16}(\boldsymbol{m}) = \begin{bmatrix} \boldsymbol{H}_8(\boldsymbol{m}_1) & \vdots & \boldsymbol{0} \\ \cdots & \vdots & \cdots \\ \boldsymbol{0} & \vdots & \boldsymbol{H}_8(\boldsymbol{m}_2) \end{bmatrix}, \boldsymbol{m} = \begin{bmatrix} \boldsymbol{m}_2 \\ \cdots \\ \boldsymbol{m}_1 \end{bmatrix} \tag{17}$$

Hence the scaled-up structure (Fig. 5)

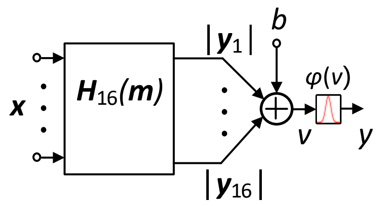

Fig. 5. Scalable perceptron structure (16-dimensional)

Thus, the basic computational intelligence architecture, having the form of Darwin machine networks, is proposed in Fig. 6.

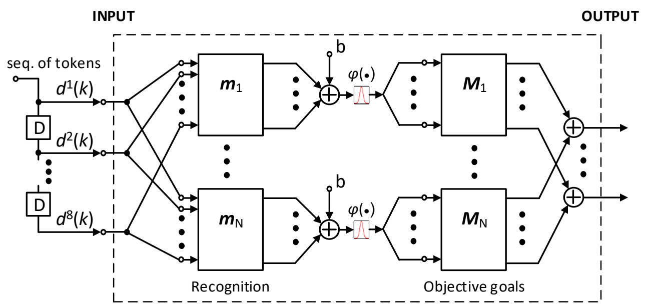

Fig. 6. An architecture of a Darwin machine network

It is worth noting that the architecture of the proposed neural network (Darwin machine network) is based on a scalable connection of selective specialized neurons. It is assumed that the selective activation function can be tuned. In addition to scalability, the network also offers an easy possibility of parallelization.

To sum up, the structure shown in Fig. 6 constitutes an implementation of mappings in vector space with the following features:

- Learning via synthesis of selective neurons.
- Stability due to feed forward architecture without vanishing and exploding gradients.
- Tokenization is performed using a *word2complex.number* model, which enables the construction of both a *sentence2complex.vector* model and an *image.patch2complex.vector* model.
- Performing sequence to sequence tasks (seq2seq).

<u>Tokenization</u>

We propose a tokenization model based on the *word2complex.number* scheme. In this approach, each word in the vocabulary is represented by a complex number of the form $e^{j\varphi_i}$, where $i = 1, \dots, \mathrm{v}$. The token segmenter processes a raw text sentence and converts it into a sequence of tokens contained in the vocabulary. Consequently, the *n*-dimensional (e.g.,8 -,16-dimensional) complex memory $\boldsymbol{m}_i$, illustrated in Fig. 6, encodes sentences consisting of words. This representation can be referred to as the *sentence2complex.vector* model.

Seq2seq tasks

One of the most popular types of seq2seq tasks is Machine Translation (MT). A common training set for MT comes as aligned pairs of sentences. It is easy to see that the network from Fig. 6 can perform the MT task. Moreover, when the input sentence is not recognized, the output vector has a value **0** ("I do not know"). Data for MT has the form of sentence pair set $(x_n, z_n), n = 1, \dots, \mathrm{v}$ where for example $x = a$ sentence in English and $z = a$ sentence in German. Every sentence x_n is assigned 8-dim complex vector $\boldsymbol{m}_i$ and to every z_n is assigned 8-dim complex vector $\widehat{\boldsymbol{m}_i}$.

Hence $\widehat{\boldsymbol{m}_i} = \boldsymbol{M}_i \boldsymbol{y}_{[1]}, i = 1, \dots, \mathrm{v}$

where: $\boldsymbol{y}_{[1]} = [1, \dots, 1]^T$and $\boldsymbol{M}_i$ is a transformation matrix of the octonionic module Eq.(5) ($\widehat{\boldsymbol{m}_i}, \boldsymbol{m}_i$ – aligned pairs).

Classification

Another important task for computational intelligence is the classification of sentences. It is easy to see that the network from Fig. 6 can be extended to a parallel structure, thus performing the function of classification (Fig. 7).

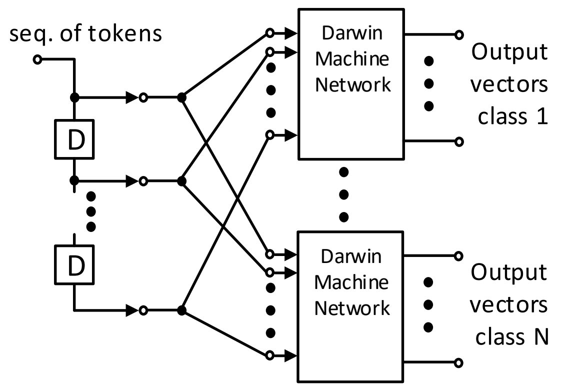

Fig. 7. The architecture of a classifier

Reconstruction of incomplete sentences

By a 'damaged' sentence we mean deletion of one word in a given input sentence. Such a numerical experiment is presented in (15) and named as robustness of the recognizer. To point out such a feature for selective perceptron from Fig. 4, the sequence of random complex vectors encoding incomplete sentences was used as input to this perceptron.

Example 2

To assess the robustness and detection capability of the selective perceptron model, a numerical experiment was performed. The experiment aimed to evaluate how the model responds to input disturbances and incomplete data patterns. During the tokenization process, a test vector $\boldsymbol{x}_t$ was generated and used to learn the selective perceptron model:

$$\boldsymbol{x}_t = \left[e^{2.731\mathrm{j}}, e^{0.524\mathrm{j}}, e^{-2.27\mathrm{j}}, e^{2.04\mathrm{j}}, e^{1.23\mathrm{j}}, e^{0.59\mathrm{j}}, e^{0.27\mathrm{j}}, e^{-1.45\mathrm{j}}\right]^T.$$

To investigate the model's resistance to disturbances, modified vectors $\widetilde{\boldsymbol{x}}_t$ were created by removing one component from $\boldsymbol{x}_t$:

$$\widetilde{\boldsymbol{x}}_t = [0, e^{0.524\mathrm{j}}, e^{-2.27\mathrm{j}}, e^{2.04\mathrm{j}}, e^{1.23\mathrm{j}}, e^{0.59\mathrm{j}}, e^{0.27\mathrm{j}}, e^{-1.45\mathrm{j}}]^T,$$

$$\widetilde{\boldsymbol{x}}_t = [e^{2.731\mathrm{j}}, 0, e^{-2.27\mathrm{j}}, e^{2.04\mathrm{j}}, e^{1.23\mathrm{j}}, e^{0.59\mathrm{j}}, e^{0.27\mathrm{j}}, e^{-1.45\mathrm{j}}]^T,$$

$$\widetilde{\boldsymbol{x}}_t = [e^{2.731\mathrm{j}}, e^{0.524\mathrm{j}}, 0, e^{2.04\mathrm{j}}, e^{1.23\mathrm{j}}, e^{0.59\mathrm{j}}, e^{0.27\mathrm{j}}, e^{-1.45\mathrm{j}}]^T,$$

$$\widetilde{\boldsymbol{x}}_t = [e^{2.731\mathrm{j}}, e^{0.524\mathrm{j}}, e^{-2.27\mathrm{j}}, e^{2.04\mathrm{j}}, e^{1.23\mathrm{j}}, e^{0.59\mathrm{j}}, e^{0.27\mathrm{j}}, 0]^T.$$

These corrupted vectors represent an incomplete or noisy version of the original input pattern. The corrupted vectors were embedded within a random sequence of complex numbers applied to the input of the selective perceptron model. The sequence of interest began at step 20.

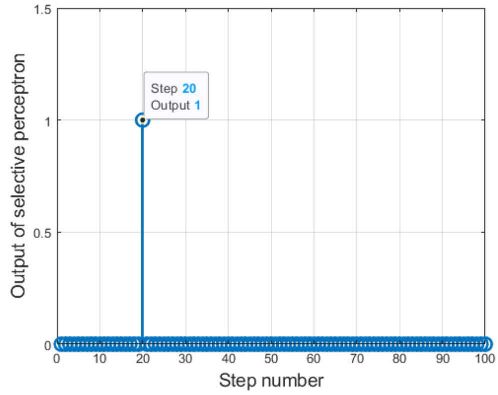

Fig. 8. Relationship between perceptron output and step number

Figure 8 presents the relationship between the model output and the step number. In every instance where the distorted sequence appeared at the model input, the output value reached 1. Experiments were also conducted in which two or three components were removed from the test vector. In most cases, the selective perceptron correctly recognized the target symbol sequence; for instance, vector $\widetilde{\boldsymbol{x}}_t$ was accurately identified:

$$\widetilde{\boldsymbol{x}}_t = [e^{2.731\mathrm{j}}, 0, e^{-2.27\mathrm{j}}, e^{2.04\mathrm{j}}, e^{1.23\mathrm{j}}, 0, e^{0.27\mathrm{j}}, 0]^T$$

for Δ = 5.5 (Fig. 4).

When more components of the test vector are removed, the selection of the Δ parameter in the perceptron becomes crucial. If the value of this parameter is too high, incorrectly recognized sequences may occur, whereas if it is too low, the learned sequence may fail to be recognized. In summary, the selective perceptron model effectively detects the damaged vector on which it was trained, demonstrating a certain degree of robustness to input disturbances.

V. Conclusion

In this study, we briefly show the design/synthesis of real and complex wavelet bases for discrete finite and infinite-dimensional vector spaces. The decomposition and reconstruction of an input signal in terms of wavelet basis is implemented via a transversal filter-bank architecture that employs octonionic modules as universal blocks. The primary objective of this study is to present the structure of a pattern/sentence recognizer. The proposed synthesis procedure can be valuable for computational intelligence. Moreover, the complex-valued octonionic modules give rise to neural networks architecture mimicking the features of Darwin machines. We claim that Darwin machine networks can be used as a framework for (non LLM) AI systems. But unlike standard neural network learning, we propose learning by synthesis.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017. [Online]. Available: arXiv:1706.03762.

[2] M. Phuong and M. Hutter, "Formal algorithms for transformers," *arXiv preprint*, 2022. [Online] Available: https://arxiv.org/abs/2207.09238

[3] W. Citko and W. Sienko, "Hamiltonian and Q-inspired neural network-based machine learning," *IEEE Access*, vol. 8, pp. 220437–220449, 2020, https://doi.org/10.1109/ACCESS.2020.3043035

[4] W. Citko and W. Sienko, "Inpainted image reconstruction using an extended Hopfield neural network based machine learning system," *Sensors*, vol. 22, no. 3, p. 813, 2022, https://doi.org/10.3390/s22030813

[5] W. Citko and W. Sienko, "Image recognition and reconstruction with machine learning: An inverse problem approach," *IEEE Access*, vol. 11, pp. 107463–107471, 2023, https://doi.org/10.1109/ACCESS.2023.3315831

[6] W. Sienko and W. Citko, "Hamiltonian neural networks based networks for learning," in *Machine Learning*, A. Mellouk and A. Chebira, Eds. Vienna, Austria: I-Tech, 2009, pp. 75–92.

[7] W. Sienko, W. Citko, and D. Jakóbczak, "Learning and system modeling via Hamiltonian neural networks," in *Artificial Intell. Soft Comput., 7th Int. Conf.*, Zakopane, Poland, 2004, *Lecture Notes in Artificial Intelligence*, vol. 3070, pp. 266–271. Heidelberg, Germany: Springer-Verlag.

[8] E. Akin, "Hopf bifurcation in the two locus genetic model", in *Memories of the American Mathematical Society*, vol. 44, no. 284, 1983.

[9] W. H. Calvin, *The Cerebral Symphony: Seashore Reflections on the Structure of Consciousness*. New York, NY: Bantam Books, 1990.