

# Zastosowanie modelu uczenia maszynowego do realizacji procesora analogowego

**Streszczenie.** W pracy zaproponowano wykorzystanie, opartego na transformacjach ortogonalnych i biortogonalnych, modelu uczenia maszynowego do syntezy procesora realizującego funkcję dodawania i mnożenia liczb rzeczywistych. Ze względu na cechy bezstratności oraz realizację zasady superpozycji model ten można zakwalifikować jako system kwantowego przetwarzania sygnałów.

**Abstract.** The goal of this paper is to present a universal machine learning model using orthogonal and biorthogonal transformations based on Hurwitz-Radon matrices. This model was used to synthesize a processor that performs the function of adding and multiplying real numbers. Due to the lossless features and implementation of the superposition principle, the model can be qualified as a quantum signal processing system. (Application of the Machine Learning Model to the Implementation of the Analog Processor).

**Słowa kluczowe:** system uczenie maszynowego, inteligencja obliczeniowa, kwantowe przetwarzanie sygnałów, macierze Hurwitza-Radona.

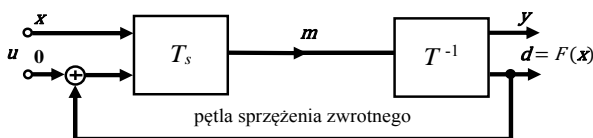
**Keywords:** machine learning system, computational intelligence, quantum signal processing, Hurwitz-Radon matrices.

## Wprowadzenie

W pracach [1, 2, 3] zaprezentowano, bazujący na analizie spektralnej model uczenia maszynowego wykorzystujący transformacje biortogonalne i ortogonalne zrealizowane w oparciu o macierze Hurwitza-Radona [4]. Podstawową funkcją tego modelu jest implementacja pewnego odwzorowania  $d = F(x)$ ,  $d \in R^n$ ,  $x \in R^m$ , generowanego przez zbiór treningowy  $S = \{x_i, d_i\}_{i=1}^N$ . Należy podkreślić, że model ten charakteryzuje się uniwersalnością, pozwalającą na realizację podstawowych funkcji systemów uczących się, takich jak kojarzenie (Pattern Association), rozpoznawanie i klasyfikacja wzorców (Pattern Recognition) oraz modelowanie odwrotne (Inverse Modelling). Ze względu na własności użytych w modelu transformacji wykazuje on cechy pozwalające na zakwalifikowanie go do dziedziny kwantowego przetwarzania sygnałów (Quantum Signal Processing). Wspomniany model został w niniejszej pracy wykorzystany do syntezy analogowego procesora wykonującego operacje dodawania i mnożenia liczb rzeczywistych. We wcześniejszych pracach pokazano także możliwość zastosowania takiego modelu uczenia maszynowego do predykcji ciągów czasowych generowanych przez układy chaotyczne [2] oraz do rekonstrukcji obrazów na podstawie uszkodzonych wzorców [3].

## Struktura modelu uczenia maszynowego

Strukturę rozważanego modelu uczenia maszynowego pokazano na rys.1. Jego podstawową funkcją jest implementacja pewnego odwzorowania  $d = F(x)$ , zadanego w punktach treningowych  $\{x_i, d_i\}_{i=1}^N$ .



Rys.1. Struktura modelu uczenia maszynowego

Zadany zbiór treningowy  $S = \{x_i, d_i\}_{i=1}^N$ , gdzie  $x_i \in R^n$ ,  $d_i \in R^m$  transformuje się do zbioru wektorów  $u_i$  o postaci:

$$(1) \quad u_i = \begin{bmatrix} x_i \\ d_i \end{bmatrix}; i = 1, \dots, N$$

gdzie:  $\dim u_i = n + m$ ,  $n + m = 2^k$ ,  $k = 2, 3, \dots$

(Powyższe założenie wynika z własności macierzy Hurwitza-Radona)

Transformacja ortogonalna  $T(\cdot)$  z wykorzystaniem macierzy Hurwitza-Radona pozwala na wyznaczenie widma Haar'a wektorów  $u_i$ ,  $i = 1, \dots, N$ :

$$(2) \quad m_i = \frac{1}{2}(W_{2^k} + I)u_i = T(u_i)$$

gdzie:  $W_{2^k}$  – macierz Hurwitza-Radona,  $W_{2^k} = -I$ ;  $m_i$  – widma Haar'a wektorów  $u_i$ ,  $i = 1, \dots, N$ ;  $M = \{m_1, m_2, \dots, m_N\}$  – macierz wektorów widma;  $I$  - macierz jednostkowa.

Zatem blok  $T^{-1}$  systemu z rys.1. realizuje operację:

$$(3) \quad u_i = (-W_{2^k} + I)m_i = T^{-1}(m_i)$$

Natomiast blok  $T_s$  transformację biortogonalną:

$$(4) \quad m_i = (2 \cdot I - W_s - W_{2^k})^{-1}u_i$$

Stąd

$$(5) \quad T_s(\cdot) = (2 \cdot I - W_s - W_{2^k})^{-1}$$

gdzie:  $W_s = M(M^T M)^{-1}M^T$ ;  $(M^T M)^{-1}M^T$  – macierz pseudoodwrotna macierzy  $M$ .

Warto zauważyć, że struktura z rys. 1 pełni funkcję estymatora widma  $\{\hat{m}_i\}$ :

$$(6) \quad \hat{m}_i = T_s \begin{bmatrix} x_i \\ \dots \\ 0 \end{bmatrix}, i = 1, \dots, N$$

gdzie:  $0$  – wektor zerowy.

Ze względu na działanie pętli sprzężenia zwrotnego uzyskuje się konwergencję wektorów:

$$(7) \quad \hat{m}_i \rightarrow m_i, \hat{y}_i \rightarrow x_i$$

oraz sygnału wyjściowego:

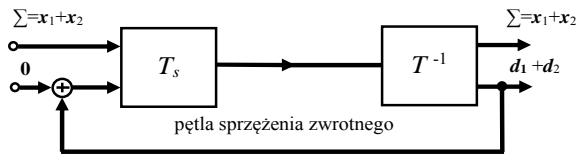
$$(8) \quad \hat{d}_i \rightarrow d_i = F(x_i), i = 1, \dots, N$$

Można także zauważyć, że każda superpozycja  $x_s = \sum_{k=1}^N \alpha_k x_k$  wektorów przynależnych do zbioru treningowego implikuje superpozycję wektorów wyjściowych  $\sum_{k=1}^N \alpha_k d_k$ ,  $\alpha_k \in R$ . Tę własność wykorzystuje się w projekcie procesora analogowego.

### Struktura procesora analogowego

W systemach przetwarzania informacji inspirowanych zasadami mechaniki kwantowej zwykle rozważa się dwa zagadnienia: kwantowe przetwarzanie informacji (QIP-Quantum Information Processing) oraz kwantowe przetwarzanie sygnałów (QSP-Quantum Signal Processing) [5, 6]. O ile kwantowe przetwarzanie informacji wymaga użycia komputerów kwantowych to kwantowe przetwarzanie sygnałów jest rozumiane jako algorytmiczna realizacja zasad mechaniki kwantowej. Model uczenia maszynowego rozpatrywany w niniejszym artykule, ze względu na cechy bezstratności i realizację zasady superpozycji można zatem zakwalifikować jako system kwantowego przetwarzania sygnału realizujący funkcje dodawania i mnożenia liczb rzeczywistych.

W celu wykonania operacji dodawania (dodawanie komponentów) dwóch wektorów  $d_1$  oraz  $d_2$  należy zdefiniować wektory systemowe  $x_1$  i  $x_2$ , których suma  $\Sigma = x_1 + x_2$ , ( $x_1 \neq x_2$ ) znana jest *a priori*. Aby otrzymać wynik sumowania ( $d_1 + d_2$ ), korzysta się z systemu jak na rys. 2.



Rys.2. Realizacja sumatora w oparciu o system uczenia maszynowego

Ponieważ, zgodnie z zależnością (8) zachodzi:

$$(9) \quad F \begin{bmatrix} x_1 \\ \dots \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ \dots \\ d_1 \end{bmatrix}, F \begin{bmatrix} x_2 \\ \dots \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_2 \\ \dots \\ d_2 \end{bmatrix}$$

więc:

$$(10) \quad F \begin{bmatrix} \Sigma \\ \dots \\ 0 \end{bmatrix} = F \begin{bmatrix} x_1 + x_2 \\ \dots \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 + x_2 \\ \dots \\ d_1 + d_2 \end{bmatrix}$$

Działanie sumatora oparte jest na podstawowej własności systemu a mianowicie na zdolności do rekonstrukcji danych przy czym  $\dim x_i > 0.1 \dim d_i$ ,  $i = 1, 2$ . Można wykazać możliwość zastosowania procesora do dodawania większej liczby wektorów.

Realizację mnożenia przez analizowany procesor można sformułować w sposób następujący: Mnożenie danego wektora  $d$  przez liczbę  $A < 1$ , tzn.  $A \cdot d$  można zdefiniować poprzez operację superpozycji. Ponieważ reprezentacja liczby  $A$  w notacji dziesiętnej przyjmuje standardową postać:

$$(11) \quad A = a_1 \cdot 10^{-1} + a_2 \cdot 10^{-2} + \dots + a_k \cdot 10^{-k}$$

Stąd

$$(12) \quad A \cdot d = (a_1 \cdot 10^{-1} + a_2 \cdot 10^{-2} + \dots + a_k \cdot 10^{-k}) \cdot d$$

gdzie:  $a_1, \dots, a_k \in [0, 1, \dots, 9]$

Tak więc otrzymuje się superpozycję następujących wektorów:

$$(13) \quad \left( \begin{bmatrix} x_1^{(1)} \\ \dots \\ 10^{-1} d \end{bmatrix} + \dots + \begin{bmatrix} x_{a_1}^{(1)} \\ \dots \\ 10^{-1} d \end{bmatrix} \right) + \dots + \left( \begin{bmatrix} x_1^{(k)} \\ \dots \\ 10^{-k} d \end{bmatrix} + \dots + \begin{bmatrix} x_{a_k}^{(k)} \\ \dots \\ 10^{-k} d \end{bmatrix} \right)$$

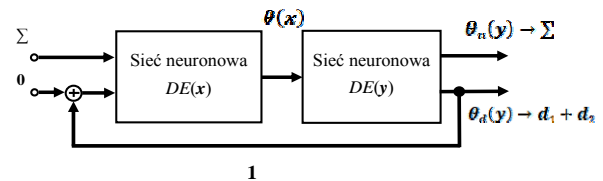
gdzie:  $x_1^{(1)}, \dots, x_{a_1}^{(1)}, \dots, x_1^{(k)}, \dots, x_{a_k}^{(k)}$  są różnymi wektorami systemowymi, których suma  $\Sigma$  znana jest *a priori*.

Przykładowo, mnożenie  $A \cdot d$  dla  $A = 0.23$  formułuje się w sposób następujący:

$$(14) \quad 0.23 \cdot d = \begin{bmatrix} x_1^{(1)} \\ \dots \\ 10^{-1} d \end{bmatrix} + \begin{bmatrix} x_2^{(1)} \\ \dots \\ 10^{-1} d \end{bmatrix} + \begin{bmatrix} x_1^{(2)} \\ \dots \\ 10^{-2} d \end{bmatrix} + \begin{bmatrix} x_2^{(2)} \\ \dots \\ 10^{-2} d \end{bmatrix} + \begin{bmatrix} x_3^{(2)} \\ \dots \\ 10^{-2} d \end{bmatrix}$$

gdzie:  $\Sigma = x_1^{(1)} + x_2^{(1)} + x_1^{(2)} + x_2^{(2)} + x_3^{(2)}$  jest znana *a priori* oraz komponenty sumy  $x_k^l$  są różne.

Warto zauważyć, że struktura aproksymatora, a zatem i procesora analogowego, może być zaimplementowana w postaci pierścieniowego połączenia sieci neuronowych jak na rys. 3.



Rys.3. Implementacja procesora w oparciu o sztuczną sieć neuronową

Struktura procesora z rys. 3 opisana jest układem równań różniczkowych zwyczajnych o następującej postaci:

$$(15) \quad DE(x) : \dot{x} = (W - 2 \cdot I + W_s) \Theta(x) + \begin{bmatrix} \Sigma \\ \dots \\ 0 + \Theta_d(y) \end{bmatrix}$$

$$DE(y) : \dot{y} = \frac{1}{2} (-W - I) \begin{bmatrix} \Theta_u(y) \\ \Theta_d(y) \end{bmatrix} + \Theta(x)$$

gdzie:  $\Theta(\cdot)$  - funkcja aktywacji neuronów:

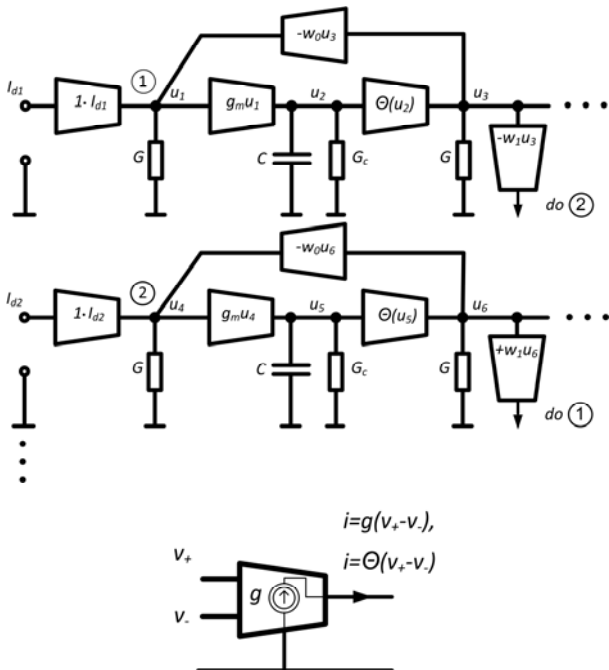
$$\mu_1 \leq \frac{\Theta(z)}{z} \leq \mu_2; \mu_1, \mu_2 \in (0, \infty)$$

Rozwiązanie powyższych równań w stanie ustalonym dane jest wyrażeniem:

$$\Theta_d(y) \rightarrow d_1 + d_2$$

$$\Theta_u(y) \rightarrow \Sigma$$

Aby nie ograniczać zakresu liczbowego procesora można przyjąć liniową funkcję aktywacji neuronów:  $\Theta(x) = x$ .



Rys.4. Realizacja neuronu z wykorzystaniem OTA

#### Realizowalność układowa procesora

Sieci neuronowe stanowiące struktury procesora są pewnym rozszerzeniem sieci typu Hopfield'a [7]. Warto zauważyć, że sieci zaprezentowane na rys. 3 opisane są równaniem różniczkowym:

$$\dot{x} = (W - w_0 I + \varepsilon W_s) \Theta(x) + I_d$$

gdzie:  $W_s$ - macierz symetryczna,  $W$  - macierz ortogonalna skośnie symetryczna,  $\varepsilon$ ,  $w_0 \geq 0$ ,  $I_d$  - wektor danych wejściowych,  $\Theta(x)$  - funkcja aktywacji neuronów

Dla wartości parametrów  $\varepsilon$ ,  $w_0 = 0$  sieć (17) stanowi dynamiczny system hamiltonowski (HNN-Hamiltonian Neural Network) [8, 9]. Strukturę sieci opisanej równaniem (17) można uzyskać poprzez kompatybilne połączenie elementarnych neuronów o postaci jak na rys. 4 gdzie przykładowo wykorzystano realizację układową za pomocą makromodeli OTA (Operational Transconductance Amplifier) lub RZPSN. Równania różniczkowe opisujące model neuronu dane są następująco:

$$\mu \frac{du_2}{dt} = -w_0 \Theta(u_2) + w_1 \Theta(u_5) - \frac{\mu G_c}{C} u_2 + G I_{d1}$$

$$\mu \frac{du_5}{dt} = -w_1 \Theta(u_2) - w_0 \Theta(u_5) - \frac{\mu G_c}{C} u_5 + G I_{d2}$$

gdzie:  $\mu = \frac{G^2 C}{g_m}$

Nietrudno zauważyć, że równania (15) i (17) można otrzymać przy warunku bezstratności neuronów tzn.  $G_c \rightarrow 0$  w strukturach z rys. 4. Badanie realizowalności technologicznej procesora o strukturze jak na rys. 3 i rys. 4 wykraczają poza zakres rozważań niniejszego artykułu.

#### Podsumowanie

Istotnymi cechami rozważanego w niniejszej pracy modelu uczenia maszynowego są jego uniwersalność, potencjalna realizowalność technologiczna oraz skalowalność, pozwalająca na proste tworzenie wielkich systemów uczenia maszynowego. W pracy, korzystając z ogólnych własności systemu uczenia maszynowego, sformułowano model analogowego procesora o postaci sprzężonych sieci neuronowych. Zaproponowano także jego układowe rozwinięcie wykorzystujące makromodeli wzmacniaczy OTA. Wydaje się, że tego typu struktury ze względu na cechy bezstratności i realizację zasady superpozycji można zakwalifikować do systemów kwantowego przetwarzania sygnału. Przewiduje się dalsze badania nad ich technologiczną realizowalnością.

**Autorzy:** dr inż. Wiesław Citko, Akademia Morska w Gdyni, Wydział Elektryczny, ul. Morska 81-87, 81-225 Gdynia; dr hab. inż. Wiesław Sieńko, Akademia Morska w Gdyni, Wydział Elektryczny, ul. Morska 81-87, 81-225 Gdynia.

#### LITERATURA

- [1] W. Citko, W. Sieńko, Modele optymalnego przetwarzania sygnałów losowych, *Zeszyty Naukowe Akademii Morskiej w Gdyni*, nr 90/2015, (2015)
- [2] W. Citko, W. Sieńko, Zastosowanie macierzy Hurwitza-Radona w uczeniu maszynowym, *Elektronika: konstrukcje, technologie, zastosowania*, (10), 28-30, (2016)
- [3] W. Citko, W. Sieńko, Realizacja pamięci skojarzeniowej z zastosowaniem uczenia maszynowego, *Przegląd Elektrotechniczny*, (8), 77-80, (2017)
- [4] B. Eckmann, Topology, Algebra, Analysis-Relations and Missing Links, *Notices of the AMS*, vol. 46, No 5, (1999)
- [5] Y. C. Eldar, A.V. Oppenheim, Quantum Signal Processing, *IEEE Signal Processing Magazine*, vol. 19, pp. 12-32, Nov (2002)
- [6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum Machine Learning, *Nature*, vol. 549, pp. 195-202, Sep (2017)
- [7] J.J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. of Nat. Acad. Sci. USA, Biophysics*, Vol. 79, p. 2554, April (1982)
- [8] W. Sienko, W. Citko, Hamiltonian Neural Networks Based Networks for Learning, in *Machine Learning*, ed. A. Mellouk and A. Chebira, Austria, Vienna: I-Tech, pp. 75-92, (2009)
- [9] W. Citko, W. Sienko, Hamiltonian Neural Networks-Based Orthogonal Filters a Basis for Artificial Intelligence, in *Proc. of the International Conference on Neural Computation Theory and Applications (NCTA)*, pp. 124-127, France, Paris, Oct 24-26, (2011)