

Dariusz RUCIŃSKI¹

¹ PhD Student at Institute of Computer Science
Siedlce University of Natural Sciences and Humanities
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

Quantum inspiration to build a neural model based on the Day-Ahead Market of the Polish Power Exchange

DOI: 10.34739/si.2021.25.02

Abstract. The article is an attempt of the methodological approach to the proposed quantum-inspired method of neural modeling of prices quoted on the Day-Ahead Market operating at TGE S.A. In the proposed quantum-inspired neural model it was assumed, inter alia, that it is composed of 12 parallel Perceptron ANNs with one hidden layer. Moreover, it was assumed that weights and biases as processing elements are described by density matrices, and the values flowing through the Artificial Neural Network of Signals are represented by qubits. Calculations checking the correctness of the adopted method and model were carried out with the use of linear algebra and vector-matrix calculus in MATLAB and Simulink environments. The obtained research results were compared to the results obtained from the neural model with the use of a comparative model.

Keywords. Neural Modeling, Day-Ahead Market, Polish Power Exchange, Mean Square Error, Determination index, Quantum Inspired Neural Model.

1. Introduction

Modeling the Day-Ahead (DAM) system with an Artificial Neural Network (ANN) is the subject of much research and endless discussion. In the works so far, the classical approach to the neural model of the DAM system with the use of real numerical data has usually been used [13, 17-23, 27, 37].

In this study, an attempt was made to approach the quantum-inspired method of neural modeling of the DAM system operating at TGE S.A., i.e., by using cybernetics and systems engineering [29-30, 34-35]. The first work on the systemic approach to the Artificial Neural Network was published in 2001 [25]. It was assumed that the systemic approach would be particularly important in the case of the construction of a quantum-inspired Artificial Neural Network.

Thus, it was assumed that two analogous models would be ultimately built, i.e., the 'classical' neural model of the DAM system [17-22] and the quantum-inspired neural model of the DAM system [19, 32], and then the obtained results should be compared, for the purpose to build the comparative model. In the proposed quantum-inspired neural model it is assumed that the model will be composed of 12 parallel elementary Perceptron ANNs based on qubit values representing by 12-bit values (12 results from the adopted measurement accuracy) of parameters and variables used in ANN training and then in its functioning.

The processing is therefore based on quantum inspiration and boils down to operations performed in the vector space for states described by the space of imaginary numbers. The vector space in which the computational operations are performed is defined as a two-dimensional Hilbert space. Such a space is defined as a linear space over the field of real or complex numbers, on which it is possible to perform the scalar product operation, is complete, i.e., each set of states described in this space, is convergent, i.e., it has a limit in this space.

It is an extension of the Banach space involving the introduction of the concept of the internal product. The set of actions and their properties that are possible to be performed on a Hilbert space are defined and constitute the basis for the operations that can be performed in calculations inspired by quantum computing [2-3, 5-6, 8, 26, 31, 33].

When constructing a neural model with quantum inspiration, it is crucial to follow the postulates of quantum mechanics adapted by quantum computing, because they describe quantum phenomena on the basis of known analytical tools.

The first postulate is that any quantum computing system can be described by a vector space, in this case a Hilbert space.

The next one relates to the possibility of a deterministic description of the evolution of quantum states based on the Schrödinger equations, which allow to describe the evolution of the state of a quantum system in time.

The third of the postulates allows to define a system that consists of several subsystems of quantum computing as their tensor products.

The fourth postulate relates to the study of the degree of entanglement of quantum states using the Schmidt distribution.

The last postulate, very important from the point of view of this work, describes the nondeterministic operation of quantum state measurement. It says that after performing the measurement operation on the state of the quantum system, it will go to one of the eigenstates of the linear operator representing the measurement operation [2-3, 5-6, 8, 26, 31, 33].

The algorithm of the quantum-inspired neural method will therefore involve three basic methods, namely: the method of quantizing real numbers into quantum mixed states, the method of quantum computing using the basis of quantum computing and the method of transforming quantum mixed states into real numbers, which is related to the application of the measurement operator [17-23, 31-33].

Such assumptions bring with them many research issues that must be solved, among others:

- the problem of real numbers representation by quantum mixed states or by density matrices in a two-dimensional Hilbert space,
- the construction of the Artificial Neural Network architecture consisting of 12 serial-parallel Artificial Neural Networks arranged hierarchically from the youngest qubit to the oldest one,
- the problem of the method of learning the Artificial Neural Network consisting of 12 neural networks being its specific architecture, adapted to the specifics of quantum computing,
- the problem of performing quantum calculations using data represented by quantum states specific to ANN,
- the problem of reading quantum mixed states and to obtain decimal values.

This work is an attempt at a systemic approach to this research issue, together with a summary of the research conducted by the author so far, including research for the future of quantum computations and algorithms conducted on classical computers.

Therefore, the subject of the research was to determine whether, based on the set of operations described in this way in the quantum space, it is possible to implement the Perceptron Artificial Neural Network, the classic form of which is shown in Fig. 1 [1, 32].

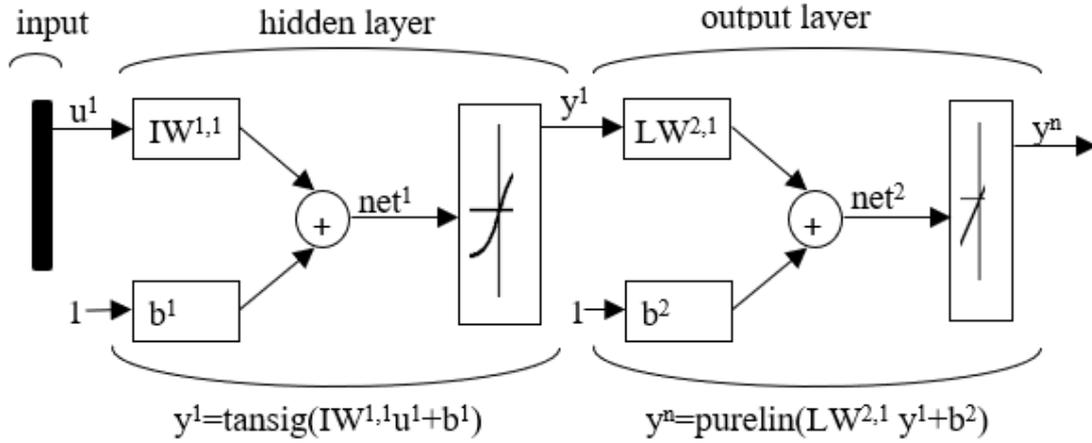


Figure 1. Diagram of the Artificial Neural Network. Notations: \mathbf{u}^1 - vector of input signals to the first layer of neurons, \mathbf{net}^1 - vector of net adders at the output of the first layer of neurons, $IW^{1,1}$ - weight matrix between input neurons and hidden layer neurons, $LW^{2,1}$ - weight matrix between the hidden layer of neurons and the layer output, b^1 , b^2 - biases, y^1 , y^n - values of signals from the hidden and output layers. Source: [1, 32].

The operation of ANN can be seen as processing the n -dimensional vector space representing the input data into the m -dimensional vector space representing the output data of the model. The processing method consists in projecting the input vectors onto successive vectors, which are ANN layers by appropriately scaling their values with proportional members, the set of which is the weight matrix for a given transformation. The mathematical record of these operations is represented by the cardinal two equations, that is [4, 7, 9-12, 14-16, 24-25, 27-28,]:

weighted adder equation:

$$\mathbf{net}_i = \sum_{i=1, j=1}^{i=n, j=m} w_{ij} u_i, \quad (1)$$

where:

\mathbf{net}_i – weighted adder of input streams on the i -th neuron,

w_{ij} – weight values between the i -th neuron of one layer and the j -th neuron of the next layer,

u_i – value of the i -th input signal;

and the equation of the activation function:

$$y_i(t) = f_i(\text{net}_i) \quad (2)$$

where the argument of the activation function is a weighted adder net_i.

2. The method of determining quantum mixed states, density matrix, activation function, etc.

In the case of the classical ANN, the real numbers in the decimal number system are processed. However, in the case of quantum computations carried out on classical computers, as shown in chapter 1, there are two kinds of situations. Firstly, they are signals flowing between neurons, and secondly, they are processing elements (specific transducers) occurring in ANNs. The flowing signals were described using quantum mixed states and the processing elements using a density matrix. The method of obtaining quantum mixed states and the density matrix was formulated on the basis of concepts introduced in quantum computing, such as:

- the concept of a qubit in a two-dimensional Hilbert space,
- Dirac notation introducing bra and ket vectors,
- the concept of a density matrix,
- the concept of an inner product,
- the concept of a tensor product, etc.

which have been discussed in detail in works on quantum computing [2-3, 5-6, 8, 26, 31] and in earlier authors or co-authors [17-23, 32-33].

It was assumed that the values of the signals flowing through the Artificial Neural Network are described by qubits, and the weights and biases as elements processing signals flowing through the Artificial Neural Network are described by the density matrices. Obtaining quantum mixed states is associated with the conversion of real numbers written in the decimal number system into numbers written in the binary number system, and these into quantum mixed states in the Hilbert space, which can be reduced to the following algorithm:

Step 1. Converting values from the space of real numbers to binary numbers.

Step 2. Convert binary values to quantum mixed states representing them - where it is assumed that binary 0 is treated as pure ket 0 and binary 1 is treated as pure ket 1.

Quantum mixed states are made up of pure states ket 0 and ket 1 for the individual bits of a binary number. The idea of a quantum mixed state comes down to the determination (measurement) of a value written in a quantum system, which is represented for a given value by the ket 1 pure state probability module or ket 0 pure state probability module, pure state ket 1 or ket 0 within the probability interval $0.71 \div 1$ [31-33].

The principle of superposition allows the designation of a second value.

Step 3. Converting selected quantum states into density matrices representing them.

The density matrix ρ , is a representation of the quantum state φ , which for an incompletely known quantum state can be described as follows [26, 31]:

$$\rho = \sum_i \rho_i |\varphi\rangle\langle\varphi|, \quad (3)$$

where the density operator is characterized by the following theorem [2-3, 5-6, 8, 26]:

The operator ρ is a density operator if and only if it satisfies the trace condition (its trace is equal to 1) and the positivity condition (it is a positive operator).

Regarding the density matrix, an important problem is whether a given operator ρ represents a pure state or a mixed state. In the case of the pure state:

$$\text{Tr}(\rho^2) = 1,$$

and when $\text{Tr}(\rho^2) < 1$ then the state of ρ is a mixed state.

3. The method of obtaining the quantum architecture of the neural model and the quantum learning method

The method of obtaining the quantum architecture of the Artificial Neural Network results from the specificity of the structure of the matrix describing the mixed states of the signals flowing through the Artificial Neural Network, such as e.g., input signals u , in the analyzed example being the volume flows of electricity supplied and sold on the Day-Ahead Market (ee) [37].

Thus, the following basic steps of the algorithm can be distinguished (Table 1) [17-23, 31-33]:

1) Converting input data represented by real numbers in the decimal number system into binary values;

2) Converting the output data, representing the goals in learning ANN with the teacher, represented by real numbers in the decimal number system into binary values;

3) Converting the weight values of the hidden layer and the output layer which were previously randomly generated represented by the real numbers written in the decimal number system into binary values;

4) Converting the bias values of the hidden layer and the output layer which were previously randomly generated represented by the real numbers written in the decimal number system into binary values;

5) Determination of the quantum state for the data obtained in points 1 ÷ 4 using the algorithm described in section 2 (step 2);

6) Calculation of the density matrix for the weights and biases obtained in point 5 using the algorithm described in section 2 (step 3).

Table 1. Summary of quantum representations of input values, weights, biases and output values.
Source: Own study.

Quantum input values	Quantum weights (and likewise biases) of the hidden layer	Quantum weights (and similarly biases) of the output layer	Quantum output values
$u_{1,12}$ $u_{1,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$	$w_{1,1,1,12}$ $w_{1,1,1,1}$ $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$ [...] $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$w_{2,1,1,12}$ $w_{2,1,1,1}$ $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$ [...] $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$y_{1,12}$ $y_{1,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$
$u_{1,12}$ $u_{2,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$	$w_{1,2,1,12}$ $w_{1,2,1,1}$ $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$ [...] $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$w_{2,2,1,12}$ $w_{2,2,1,1}$ $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$ [...] $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$y_{1,12}$ $y_{2,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$
$u_{3,12}$ $u_{3,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$	$w_{1,3,1,12}$ $w_{1,3,1,1}$ $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$ [...] $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$w_{2,3,1,12}$ $w_{2,3,1,1}$ $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$ [...] $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$y_{3,12}$ $y_{3,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$

4. Quantum method of teaching the Artificial Neural Network with disturbances

It was assumed that the quantum method of teaching the Artificial Neural Network with 12 neural networks embedded in it, adapted to the appropriate qubits, will be based on the Perceptron Artificial Neural Network and the ANN method by means of back propagation errors, which were described in detail, among others in works [4, 7, 9-12, 14-16, 27-31, 34], which in the case under consideration can be reduced to the following basic steps:

Step 1. Determination of the mean square error. The mean square error is computed as follows:

$$e_j(n) = d_j(n) - y_j(n). \quad (4)$$

where:

$d_j(n)$ - quantum representation of the target value,

$y_j(n)$ – representation of quantum states obtained from ANN.

For all outputs in the n-th iteration, this error is respectively:

$$E(n) = \frac{1}{2} \sum_1^n e_j^2(n), \quad (5)$$

of which the average error for the n-th iteration is:

$$E_{avg}(n) = \frac{1}{n} \sum_1^n E(n). \quad (6)$$

Step 2. Determination of the input value of signals to neuron j from the previous layer

$$v_j(n) = \sum_1^m w_{ji}(n) y_j(n), \quad (7)$$

where:

v_j - input value of signals to neuron j from the previous layer

m – number of input neurons d from the previous layer to neuron j,

y_j - value of output signals from the previous layer,

w_{ji} - weights connecting neurons and the previous layer with neuron j.

The output value from neuron j is:

$$y_j(n) = f(v_j(n)), \quad (8)$$

where:

$f()$ - activation function.

Step 3. Determining the change in weight value

Due to the fact that the change in the weight value $\Delta w_{ji}(n)$ can be presented as $\frac{\partial E(n)}{\partial w_{ji}(n)}$ using the Leibniz notation one can write¹:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} * \frac{\partial e_j(n)}{\partial y_j(n)} * \frac{\partial y_j(n)}{\partial v_j(n)} * \frac{\partial v_j(n)}{\partial w_{ji}(n)}, \quad (9)$$

¹ Leibniz's notation is as follows. If the variable z depends on the variable y, which itself depends on the variable x (i.e., y and z are dependent variables), then z, through the intermediate variable y, also depends on x. In this case, the string rule is: $\partial z \partial x = \partial z \partial y * \partial y \partial x$.

of which we can write further on the basis of (9):

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n), \quad \frac{\partial e_j(n)}{\partial y_j(n)} = -1, \quad \frac{\partial y_j(n)}{\partial v_j(n)} = f'(v_j(n)), \quad \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n). \quad (10)$$

From the dependence (10) it follows that:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)f'(v_j(n))y_j(n). \quad (11)$$

In this case, the change in weight taking into account the factor η can be represented:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} = \eta e_j(n)f'(v_j(n))y_j(n). \quad (12)$$

So finally, the new value of weight in next iteration is:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (13)$$

Learning the Artificial Neural Network by means of backward error propagation can be presented in a systemic way as in Fig. 2.

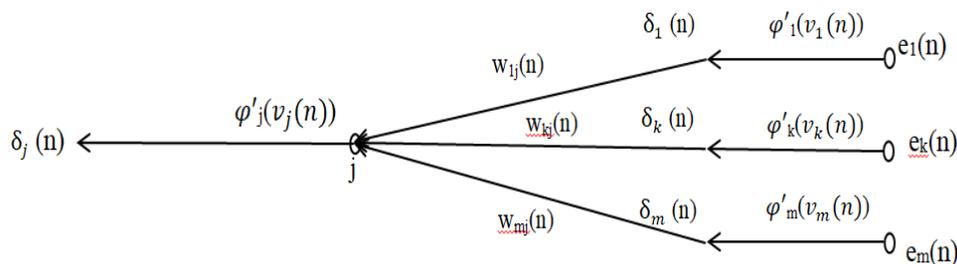


Figure 2. ANN connection method with the backward error propagation method. Markings: n - error on the j -neuron, $\delta_j = f'_j(v_j(n))$ - partial derivative of the function $f()$ on the weights. Source: Own elaboration based on the work [27].

5. Quantum computations specific to ANN

After the Perceptron Artificial Neural Network has been taught the neural model of the Day-Ahead system, that is, 12 parallel Artificial Neural Networks, two procedures are possible. It is possible to build a simulation model of a neural model consisting of 12 generated Qubit Artificial Neural Networks, between which there should be appropriate couplings resulting from the method of quantum calculations carried out on qubits. But it is also possible to build a mathematical model including individual quantum computations describing the processes taking place in 12 interconnected artificial neural networks.

As part of the currently performed research, the ANN Perceptron model was developed according to the method shown in the form of an algorithm in Fig. 3.

6. Measurement of quantum mixed states and their dequantization

In the course of the conducted research, there were some difficulties related to, inter alia, with accuracy in measuring the obtained quantum mixed states in the signals flowing between neurons, as well as difficulties related to the accuracy of real numbers obtained as a result of the dequantization process, described in detail in [17-23, 31-33], which had an impact on the obtained results.

The measurement consisted of:

1. Reducing the superposed mixed quantum state to one base value 0 or 1 with a specified probability modulus;
2. Converting the obtained value, written in the form of a binary 12-element number into a real number written in the decimal number system.

7. Conclusions and directions for further research

The conducted research has shown that it is possible to implement a quantum-inspired Perceptron Artificial Neural Network. In the considered computational example, such a Perceptron Artificial Neural Network was composed of 12 artificial neural networks quasi-parallel processing information related to individual qubits of a mixed quantum state obtained as a result of quantization of real numbers.

The quantum streams flowing through the ANN were recorded in the form of quantum mixed states. On the other hand, the processing elements, i.e., weights and biases, respectively, were represented by density matrices. In the case of the activation function, its arguments as input elements of these functions were introduced in the form of quantum mixed states. Then they were further used, like scales and biases, as processing elements, and thus also in the form of a density matrix. The Perceptron ANN built in this way could be further learned and then used as a quantum-inspired model of the TGE S.A. DAM system. The work will be continued.

References

1. Beale M. H. [at all]: Neural Network Toolbox™ User's Guide, by The MathWorks, Inc, 1992-2020, pp. 846.
2. Bernhardt Ch.: Obliczenia kwantowe dla każdego (English: Quantum computing for everyone). WN PWN, Warszawa 2020, stron 202.
3. Chudy. M.: Wprowadzenie do informatyki kwantowej (English: Introduction to quantum computing). AOW EXIT, Warszawa 2012, stron 84.
4. Faliński M.: Wstęp do sztucznej inteligencji (English: Introduction to artificial intelligence). WN PWN, Warszawa 2020, stron 332.
5. Giaro K., Kamiński M.: Wprowadzenie do algorytmów kwantowych (English: Introduction to quantum algorithms). AOW EXIT, Warszawa 2003, stron 165.
6. Hirvensalo M.: Algorytmy kwantowe (English: Quantum algorithms). WSiP, Warszawa 2004, stron 244.
7. Jankowski N.: Ontogeniczne sieci neuronowe. O sieciach zmieniających swoją strukturę (English: Ontogenic neural networks. With networks changing their structure). AOW EXIT, Warszawa 2003, stron 312.
8. Johnston E. R., Harrigan N., Gimeno-Segovia M.: Komputer kwantowy. Programowanie, algorytmy, kod (English: Quantum computer. Programming, algorithms, code). Hellion, Warszawa 2020, stron 278.
9. Kosiński R. A.: Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos (English: Artificial neural networks. Nonlinear dynamics and chaos). WNT, Warszawa 2002, stron 195.
10. Korbicz J., Obuchowicz A., Uciński D.: Sztuczne sieci neuronowe. Podstawy i zastosowania (English: Artificial neural networks. Basics and applications), Problemy Współczesnej Nauki. Teoria i zastosowania, Informatyka, AOW PLJ, Warszawa 1994, stron 251.
11. Mańdziuk J.: Sieci neuronowe typu Hopfielda (English: Hopfield neural networks). Teoria i przykłady zastosowań. AOW EXIT, Warszawa 2000, stron 262.

12. Marecki J.: Metody sztucznej inteligencji (English: Artificial intelligence methods), WSiP, Bielsko-Biała 2001, stron 115.
13. Monteiro, [at all]: Short-Term Price Forecasting Models Based on Artificial Neural Networks for Intraday Sessions in the Iberian Electricity Market. *Energies*, No. 9, 721/2016, stron 24.
14. Obuchowicz A., Patan K.: Sieć neuronów dynamicznych jako generator residuów. Optymalizacja architektury (English: Dynamic neuron network as a residual generator. Architecture optimization). *Materiały III Konferencji nt. "Diagnostyka procesów przemysłowych"*, Gdańsk-Jurata 1998, s. 101-106.
15. Osowski S.: Sieci neuronowe do przetwarzania informacji (English: Neural networks for information processing). OW PW, Warszawa 2013, stron 422.
16. Patterson J., Gibson A.: Deep Learning. Praktyczne wprowadzenie (English: Deep Learning. A practical introduction), Helion, Warszawa 2017, stron 451.
17. Ruciński D.: Neural modelling of electricity prices quoted on the Day-Ahead Market of TGE SA shaped by environmental and economic factors. *Studia Informatica. System and Information Technology*, Vol. 1-2(24)2020, pp 25-35.
18. Ruciński D.: The Influence of the Artificial Neural Network Type on the Quality of Learning on the Day-Ahead Market Model at Polish Electricity Exchange Join-Stock Company, *Studia Informatica. System and Information Technology*, Vol. 1-2(23)2019, pp. 77-93.
19. Ruciński D.: Modelowanie neuronalne cen na Towarowej Giełdzie Energii Elektrycznej wspomagane algorytmem ewolucyjnym oraz inspirowane obliczeniami kwantowymi (English: Price neural modeling on the Polish Power Exchange supported by an evolutionary algorithm and inspired by quantum calculations). *Rozprawa doktorska pod kierunkiem dr hab. inż. Jerzego Tchórzewskiego, prof. UPH w Siedlcach, IBS PAN*, Warszawa 2018, str. 137.
20. Ruciński D.: The neural modelling in chosen task of Electric Power Stock Market. *Studia Informatyka. System and Information Technology*. Vol. 1-2(21)2017. Publishing House of University of Natural Science, pp. 1-22.
21. Ruciński D., Tchórzewski J.: Neural modeling of the electric power stock market in usage of MATLAB and Simulink tools for the day ahead market data. *Journal of Information System in Management*, Vol. 5 (2)2016, pp. 215-226.

22. Ruciński D.: Neural-evolutionary Modeling of Polish Electricity Power Exchange. Electrical Power Networks, EPNet 2016, Katedra Energoelektryki PWr., Institute of Power Engineering and Control Systems of Lviv Polytechnic National University, Ukraine and O/Wrocławski SEP, XPlore Digital Library, Szklarska Poręba 2016 r., pp. 1-6.
23. Ruciński D., Kłopotek M., Tchórzewski J.: Self-Organizing Wireless Ad-hock Sensor Networks. *Studia Informatica. Systems and Information Technology*, Volume: 1(5)2005, Publishing House of University of Podlasie, pp. 69-80.
24. Rutkowska D., Piliński M., Rutkowski L.: *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte* (English: *Neural networks, genetic algorithms and fuzzy systems*), WN PWN, Warszawa – Łódź 1997, stron 411.
25. Rutkowski L.: *Metody i techniki sztucznej inteligencji* (English: *Methods and techniques of artificial intelligence*). WN PWN, Warszawa 2020, stron 435.
26. Sawerwain M., Wiśniewska J.: *Informatyka kwantowa. Wybrane obwody i algorytmy* (English: *Quantum informatics. Selected circuits and algorithms*). WN PWN, Warszawa 2015, stron 371.
27. Sengupta, S. (2014-02): *Neural Networks and Applications*. <http://nptel.ac.in/courses/117105084> [dostęp: 15-08-2017, 04-02-2021].
28. Tadeusiewicz R.: *Archipelag Sztucznej Inteligencji* (English: *The Artificial Intelligence Archipelago*). AOW EXIT, Warszawa 2021, stron 125.
29. Tadeusiewicz R., Korbicz J., Rutkowski L., Duch W.: *Sieci neuronowe w inżynierii biomedycznej* (English: *Neural networks in biomedical engineering*). Seria: *Inżynieria biomedyczna. Podstawy i zastosowania*, tom 9, Akademicka OW EXIT, Warszawa 2013, str. 746.
30. Tadeusiewicz R.: *Biocybernetyka. Metodologiczne podstawy dla inżynierii biomedycznej* (English: *Biocybernetics. Methodological foundations for biomedical engineering*). WN PWN, Warszawa 2013, str. 236.
31. Tchórzewski J.: *Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterownia i systemów* (English: *Methods of artificial intelligence and quantum computing in terms of the theory of control and systems*). WN UPH, Siedlce 2021, stron 343.

32. Tchórzewski J., Ruciński D.: Evolutionally-Supported and Quantum-Inspired Neural Modeling Applied to the Polish Electric Power Exchange. 2019 Progress in Applied Electrical Engineering (PAEE), IEEE Digital Library, Kościelisko 2019, pp. 1-8.
33. Tchórzewski J., Ruciński D., Domański P.: Artificial neural network inspired by quantum computing solutions using the movement model of the PR-02 robot. ITM Web of Conferences 19/2018, pp. 1-2.
34. Tchórzewski J.; Cybernetyka życia i rozwoju systemów (English: Cybernetics of life and systems development). Wyd. WSRP, Siedlce 1992, stron 408.
35. Tchórzewski J., Kłopotek M.: A case study in neural network evolution. Prace Instytutu Podstaw Informatyki Polskiej Akademii Nauk, Nr 943/2001, pp. 1-12.
36. Żurada J., Barski M., Jędruch W.: Sztuczne sieci neuronowe. Podstawy teorii i zastosowania (English: Artificial neural networks. Fundamentals of theory and application), WN PWN, Warszawa 1996, stron 375.
37. www.tge.pl – the website of Polish Power Exchange [access: 15-08-2017, 04-02-2021].