sciendo
Control and Cybernetics
vol. 50 (2021) No. 3
pages: 383-399
DOI: 10.2478/candc-2021-0023

Quantum-inspired method of neural modeling of the day-ahead market of the Polish electricity exchange*

by

Jerzy Tchórzewsk[1] and Dariusz Ruciński[2]

[1]University of Natural Sciences and Humanities, Computer Science Institute, 3-Maja 54, 08-110 Siedlce, Poland
[2]University of Natural Sciences and Humanities, PhD Student at Comupter Science Institute,
3-Maja 54, 08-110 Siedlce, Poland

Abstract: The paper presents selected elements of a modelling methodology involving quantization, quantum calculations and dequantization on the example of the neural model of the Day-Ahead Market of the Polish Electricity Exchange. Based on the fundamental assumptions of quantum computing, a new method has been proposed here of converting the real numbers in decimal notation into quantum mixed numbers using the probability modules of quantum mixed number and the principle of superposition, along with a new method of quantum calculations using linear algebra and vector-matrix calculus, and the Artificial Neural Network was taught accordingly. Dequantization of quantum mixed numbers to real numbers in decimal notation using the new method of dequantization has been proposed as well. The operation of the methods introduced was shown on numerical examples.

Keywords: artificial neural networks, day-ahead market, dequantization with ANN, neural modeling, quantum-inspired method, quantum computing, Polish Electricity Exchange, system quantization

1. Introduction

Neural modeling itself is not a new issue. There are already a significant number of articles on the use of artificial neural networks for modeling systems in various fields and disciplines, including, in particular, the field of testing the quality of

*Submitted: October 2020; Accepted: February 2021.
The paper is an extended and amended version of the presentation, provided during the BOS/SOR2020 conference in December 2020.

electricity load prediction (see Ciechulski and Osowski, 2014), or for forecasting of prices on electricity exchange (see Miller and Bućko, 2014).

On the other hand, quantum computing can be performed, in particular, with the use of linear algebra and vector matrix calculus in terms of control theory and systems (see: Kaczorek, 1998; Kaczorek, et al., 2020; Tchórzewski, 1992, 2013) in the MATLAB and Simulink environment with the use of Matlab language, see the example based on the data quoted on the Day Ahead Market (DAM) of the Polish Power Exchange (PPE) (Mielczarski, 2000; Ruciński, 2018). In this regard, experimental studies were first conducted by designing and implementing an appropriate quantum-inspired Perceptron Artificial Neural Network in the MATLAB and Simulink environment, the parameters of which (weights and biases) were improved using an evolutionary algorithm (see Ruciński, 2018; Tchórzewski and Ruciński, 2020). On the basis of the obtained practical results and conclusions drawn from this implementation, a new method of quantum-inspired neural-evolutionary modeling of the PPE Day-Ahead Market was proposed, as presented here, consisting of:

- method of quantizing the real numbers in decimal notation into quantum mixed numbers,
- method of quantum calculations using linear algebra and vector-matrix calculus,
- method of quantum mixed number dequantization into real numbers in decimal notation,

which can be further used to build a new system modeling methodology using quantum inspired methods of artificial intelligence.

Thus, in particular, after formulating the research problem, the systemic basis for building quantum methodologies, associated with artificial intelligence methods, such as Artificial Neural Networks (ANNs) and Evolutionary Algorithms (AE) is shown here, with the main emphasis in this work being on the quantum inspiration of the Artificial Neural Network. It should be noted, however, that prior to applying quantum inspiration to ANN, weights and biases were corrected using the Evolutionary Algorithm (see Arabas, 2016; Michalewicz, 2003; Obuchowicz, 2013), obtaining an improvement from the mean square error (MSE) level ranging from -0.11 % to 0.12% up to the range of 0.04% ÷0.05%, i.e. by an order of magnitude (see Tchórzewski and Ruciński, 2018; Ruciński, 2018). Then, the basic aspects of the quantum inspired methodology are shown, such as quantization leading to obtaining quantum mixed numbers, performing calculations on quantum numbers, as well as the need to perform dequantization using the Artificial Neural Networks, learned for this purpose. We also show new problems, identified in the course of research, which may constitute new research directions, such as the development of a method of dequantization of quantum mixed numbers into real numbers in decimal notation (Tchórzewski and Ruciński, 2016; Ruciński, 2018).

2. Quantum computing on classical computers

Quantum mathematics and mechanics, and, consequently quantum computing, is based on linear algebra (see Feynman, Leughton and Sands, 2014; Bernhardt, 2020; Adamowski, 2019; Heller, 2016), so adding and subtracting numbers is intuitively simple, with some exceptions, such as finding the square root or calculating the trigonometric function. For data processing, real numbers are used in a simpler form (it is easier to make measurements, then), and, in some more complicated situations, the complex numbers are used (they facilitate the connection of trigonometric and exponential functions). In the vector form of data, the notation of Paul Dirac (one of the founders of quantum mechanics) is used, denoting the column vector with ket |>and the row vector with bra |>(see Le Belac, 2018; Chudy, 2011; Hirvensalo, 2004).

It is assumed that computing the value of the function $y = f(u)$, which for the elements u of the set of integers $u \in \{0,\ 1,.\ .\ .\ ,\ 2^m\ \text{-}\ 1\}$matches the y elements of another set of integers $y \in \{0,\ 1,.\ .\ .\ ,\ 2^n\ \text{-}\ 1\}$, where m and n are positive integers (see Adamowski, 2019; Susskind and Frideman, 2016) will be performed with a classical computer and with a quantum computer. With the help of a classical computer, the calculation of the function $y = f(u)$ is performed in such a way that each index $u \in \{0,\ 1,\ .\ .\ .\ ,\ 2^m{-}1\}$ on the input is assigned to the correspondingly indexed value of the function on the output, that is, $y \in \{f(0),\ f(1),.\ .\ .\ ,\ f(2^m\ \text{-}\ 1)\}$. In quantum computer calculations, the unitary operation U is used to calculate the value of the function $y = f(u)$, where each input value u is represented by a quantum state vector $|u_i$>of input register and each possible output value $y = f(u)$ is represented by the quantum state vector $|y_i$>of output register. Quantum state vectors corresponding to different input values and different output values are orthonormal, that is:

$$< u|u' >= \delta_{uu'}, < y|y' >= \delta'_{yy}. \tag{1}$$

where δ_{yy} is the Kronecker delta.

The operation of calculating the value of the function $y = f(u)$ in a quantum manner is defined by the unitary operator U, which acts on two registers:

$$U|u > |0 >= |u > |f(u) >, \tag{2}$$

where the first register ($|u>$) stores the input values, and the state of the second register is transformed into the output state ($|0> \rightarrow |y = f(u)>$). It is believed that the state of the input register can be prepared in such a way that it is a superposition of all single-qubit states occurring with the same amplitudes, i.e. (see Adamowski, 2019):

$$|\Psi_{input}\rangle = \frac{1}{2^{\frac{m}{2}}} \sum_{u=0}^{2^m-1} |u\rangle \tag{3}$$

By means of the unitary operator U, the calculation of the function $y = f$ (u) is performed only once, obtaining all 2^m values of the functions $f(0)$, $f(1)$,.

. . , $f(2^m - 1)$, that is:

$$\left|\Psi_{output}\right\rangle = U\left(\frac{1}{2^{\frac{m}{2}}}\sum_{u=0}^{2^m-1}|u\rangle\right)|0\rangle = \frac{1}{2^{\frac{m}{2}}}\sum_{u=0}^{2^m-1}|u\rangle|f(u)\rangle\ . \tag{4}$$

The quantum state, described by the relation (3), contains the superposition of all 2^m states $|f(0)>$, $|f(1)>$,. . . $|f(2^m - 1)>$, however, none of the measurements provides information about all these quantum states. A single measurement in the quantum state (3) allows to obtain the following information (see Adamowski, 2019):

1. Each of the output states $|x>|f(x)>$ can be obtained with an equal probability of $1/2^m$, and therefore each of the values of the functions $f(0)$, $f(1)$,. . . , $f(2^m$ - 1) can occur with the same probability $1/2^m$.

2. If the result of the measurement, for example, is obtained:

$$|\Psi_{output}\rangle = |\tilde{u}\rangle|f(\tilde{u})\rangle, \tag{5}$$

then the result of the next measurement, performed immediately after the first measurement, will be the same state, obtained with the probability equal to 1, which means that the value of the function $f(u)$ will be obtained again, i.e. no additional information about the new value of the function $y = f(u)$ is acquired.

3. Basics of quantum computing

In the quantum calculations conducted in this study, the basic concepts of linear algebra and vector-matrix calculus were used (see Kaczorek, 1998; Sawerwain and Wiśniewska, 2015). This applies, in particular, to such concepts of linear algebra as: vector diagrams, vector lengths, vector multiplication by a scalar, vector addition, bra-ket multiplication (bra-ket product, internal product or an inner product), the concept of vector orthogonality (bra-ket product equal to zero), the notions of an orthonormal basis (a set of n orthonormal unit kets, any two of which are orthogonal to each other), a weighted sum of basis vectors (linear combination), a fixed order (sequence of occurrence) of vectors, the concept of a matrix and the occurrence of matrix rows as bra vectors and of matrix columns as ket vectors, as well as such concepts of vector-matrix calculus as, among others: various matrix forms (including a square matrix), multiplication of two matrices **A** and **B** as a multiplication of matrix **A** consisting of bras and matrix **B** composed of kets (in the order that bra occurs first, then ket, and the dimensions of ket are the same as of bra), etc., see Tchórzewski and Ruciński (2018), Ruciński (2018), Bernhardt (2020).

Computing with matrices is preceded by checking whether the ket vectors form an orthonormal basis, i.e. first of all checking whether they are unit vectors and whether they are vectors orthogonal to each other, i.e. whether the elements on the main diagonal are 1 and 0 outside of it. In the case of an orthogonal matrix that is a square matrix, the product of this matrix transposed by the same matrix ($\mathbf{M}^t\mathbf{M}$) yields an identity matrix. Moreover, if the matrix $\mathbf{M}$ has complex elements, then the result is a unitary matrix (see: Bernhard, 2020; Feynman et al., 2014; Sawerwain and Wiśniewska, 2015). As can be easily seen from the above, the linear algebra tools were used for checking.

Next, it is worth paying special attention to the possibility of using the probability theory in determining quantum mixed numbers, i.e. quantum numbers residing in many different states at the same time (in particular, in two quantum states, i.e. in the pure quantum ket 0 and in the pure quantum ket 1 with the probability amplitudes appropriate for both quantum states). In this situation, the concept of qubit and pure quantum states and the associated quantum measurement operation are important. The measurement, on the other hand, is related to the choice of the direction, which, in turn, is the same as choosing the base corresponding to the selected direction. The basis in R^2 can be the basis of two vectors, that is (|0> and |1>). If this base is rotated by the angle α, then a new base is obtained:

$$\left(\begin{bmatrix} \cos(\alpha) \\ -\sin(\alpha) \end{bmatrix}, \begin{bmatrix} \sin(\beta) \\ \cos(\beta) \end{bmatrix} \right), \tag{6}$$

so that through a 90^o rotation, the quantum state returns to the original basis, but the elements change places. In quantum computing, another concept of linear algebra plays a very important role, that is, the tensor product, which is used to describe the entangled states. It is said that if a quantum complex state is an entangled state, if it is not in a decomposable state, and then it cannot be represented as (see Hirvensalo, 2004; Giaro and Kamiński, 2003):

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_{ij}|u_i\rangle|y_j = \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i\beta_j|u_i\rangle|y_j = \left(\sum_{i=1}^{n} \alpha_i|u_i\rangle\right)\left(\sum_{j=1}^{m} \beta_i|y_j\rangle\right). \tag{7}$$

The concept of the decomposable state is independent of the choice of bases of the vector spaces under consideration. Quantum states in mechanics and in quantum computing are mathematically described as elements of a vector space, and physical observables, i.e. quantities obtained as a result of measurements, are described as linear operators that are linear and Hermitian (see Susskind, 2016; Hirvensalo, 2004). Observables in classical mechanics are presented as linear operators equal to their Hermitian couplings, where the Hermitian operators satisfy the condition:

$$\mathbf{M} = \mathbf{M}^\dagger, \tag{8}$$

which, in terms of vector-matrix calculus, corresponds to the equality of matrix elements:

$$m_{ji} = m_{ji}^*, \tag{9}$$

where the Hermitian operators (and matrices) have the property that all their eigenvalues are real. The possible measurement results are the eigenvalues of the operator, representing a specific observable λ_i, where the measurement results in the corresponding eigenvector $|\lambda_i>$, and more specifically, if the system is in the eigen state $|\lambda_i>$, the measurement results in λ_i, where distinguishable states are perpendicular to each other. If |A> is a vector of the state of the tested system, then by measuring the observable L, one can calculate the probability of observing the value of λ_i from the dependence:

$$P(\lambda_i) = \langle A \mid \lambda_i \rangle \langle \lambda_i \mid A \rangle , \tag{10}$$

where:

λ_i - the eigenvalues of the linear operator L,

$|\lambda_i>$- the eigenvectors corresponding to λ_i.

For the above-mentioned reasons, the linear operator L is used to measure the values of the eigenvectors λ_i, which are related to quantum states. The measurement result will always be one of the eigenvalues λ_i of the operator L, and the measurement result will be the probability P(λ_i) of obtaining the result λ_i.

4. The method of system quantization

A systemic method of converting the real numbers in decimal notation to quantum numbers using the probability modules of a quantum mixed number and the superposition principle was developed, which consists in converting the real numbers in decimal notation to binary numbers, and these to quantum mixed numbers (see Ruciński, 2018, Tchórzewski and Ruciński, 2016, 2018; Wright and Jordanov, 2017). In the case of a single qubit, i.e. a quantum system composed of two pure states ket 0 and ket 1, in order to determine the dominant and recessive ranges, a greater probability of obtaining the pure state |0> takes place when it is the dominant state and the pure |1> state is a recessive state, and, similarly, the pure |1> state is obtained when it is the dominant state and the |0> state is a recessive state. In such a situation, it is convenient to use the principle of superposition. Thus, the probability moduli are assumed to be equal, that is, $\alpha = \beta$, and therefore:

$$2 \cdot \alpha^2 = 1, \tag{11}$$

wherefrom we can further determine the positive value of the solution, which is:

$$\alpha = \frac{\sqrt{2}}{2}, \tag{12}$$

this resulting from the interpretation of the physical values of both the probability modulus α and the probability modulus β concerning the pure state ket

0 and the pure state ket 1, which are always positive. Due to the equality of both probability moduli, this value is approximately:

$$\alpha = \beta \approx 0.71, \tag{13}$$

which is the border value between the dominant and the recessive intervals, i.e. between the interval, from which the values of the dominant states can be drawn, and the interval, from which the values of the recessive states can be drawn. Thus, in the case when the dominant state is the pure state |0>or |1>, then the values of the probability modulus α or β are drawn from the range of the dominant states:

$$0.71 \leq \alpha \leq 1, \tag{14}$$

and the values of the probability modulus β or α are derived from the principle of superposition. The same applies when the recessive state is pure |1> or |0>, then the values of the probability modulus β or α are randomized from the interval of the recessive states:

$$0 \leq \beta \leq 0.71, \tag{15}$$

and the values of the probability modulus β or α are derived from the principle of superposition. Thus, two pairs of probability moduli, α or β, are obtained, which can be used to more accurately define the ranges of the dominant and recessive states, from where the appropriate probability modules of the mixed states will be finally drawn.

The essence of converting the decimals into binary numbers is ultimately about obtaining the pure states |0> and |1> of the quantum mixed state. Thus, for example, the values of the weights of the Artificial Neural Network and the values of input signals to the ANN are converted, first, to binary numbers, and then it is assumed that binary 0 is pure |0>, and binary 1 is pure |1>. Then, quantum mixed states are determined by selecting the appropriate probability modules from the determined dominant and recessive intervals using the conditions (14) and (15). Obtaining a quantum mixed state, therefore, comes down to determining the average values of the probability modules on the basis of many randomizations (theoretically infinitely many) of one probability module value and determining the superposition of the value of the second probability module. Thus, a random selection of the instantaneous value of one quantum probability module of the mixed state results from this pure state, which is the dominant state, and the second probability module is determined from the superposition principle (see Ruciński, 2018; Tchórzewski and Ruciński, 2016, 2018).

Hence, if the dominant state is the pure state |0>, then the value of α probability modulus is drawn infinitely many times (the study, reported in this paper assumes the number of draws to be equal 1,000) from the interval (14), on the basis of which the mean value is determined, and the value of the β

probability modulus is determined on the basis of the superposition rule applied to both states. Then, the recessive state is determined in an analogous way, using (15), this being then the pure state 1, and so infinitely many times (it was again assumed in study here reported that the number corresponds to 1,000) the recessive state probability module value is drawn from the range described by the equation (15) and the mean value of the determined 1,000 values is taken as the measured value, and the second probability modulus (in this case the dominant state) is determined from the principle of superposition of both states.

In the above-mentioned manner, the values of both ranges are narrowed down, i.e. the ranges, in which the dominant states and, respectively, the recessive states of quantum mixed numbers can occur. This procedure applies to all bits of a binary number obtained from each real number. It should also be added that when determining the real value, represented by a binary number, the individual bit positions of the quantum number must be taken into account, similarly to binary numbers.

5. Dequantization with ANN

To describe the quantum-inspired Perceptron Artificial Neural Network, the foundations of control and systems theory were used (see Kaczorek, 2020, Tchórzewski, 1992, 2013), and, in particular, the definition of the system state was taken as the basis for determining the mixed states of a quantum number. As a result of the performed calculations, a model of the Quantum Artificial Neural Network (QANN) is obtained, i.e. a neural model, inspired by solutions of quantum computing. It is convenient to interpret quantum calculations in the form of a model of a single QANN neuron, described, for example, by the activation function of the sigmoid tangent of the i-th neuron in the k-th weight layer of the Perceptron Artificial Neural Network:

$$\mathrm{y}(\mathrm{net}_{\mathrm{i}}^{\mathrm{k}}(\mathrm{t})) = [1_{\mathrm{q}} - \mathrm{e}_{\mathrm{q}}^{-\mathrm{net}_{qi}^{k}}][1_{\mathrm{q}} + \mathrm{e}_{\mathrm{q}}^{-\mathrm{net}_{q\mathrm{i}}^{k}}]^{-1}, \tag{16}$$

where:

$$1_{\mathrm{q}} = \begin{bmatrix} 0.0860 & 0.9964 & . .. & 0.0848 & ... & 0.0862 \\ 0.9962 & 0.0845 & . .. & 0.9963 & ... & 0.9962 \end{bmatrix},$$

$$\mathrm{e}_{q} = \begin{bmatrix} 0.9963 & 0.0867 & . ... & 0.9963 & ... & 0.08569 \\ 0.0855 & 0.9962 & ... & 0,0851 & ... & 0.9963 \end{bmatrix},$$

$$\mathrm{net}_{q\mathrm{i}}^{\mathrm{k}} = -\begin{bmatrix} \mathrm{net}_{q\mathrm{i,im}}^{\mathrm{k}} & \mathrm{net}_{q\mathrm{i,lm}}^{\mathrm{k}} \\ \mathrm{net}_{q\mathrm{i,lm}}^{\mathrm{k}} & \mathrm{net}_{q\mathrm{i,lm}}^{\mathrm{k}} \end{bmatrix}$$

and where:

$_q$- index informing about the recording of numbers in the quantum number system, as a consequence of the occurrence of probability modules, associated with the pure state 0 (first rows) and the pure state 1 (second rows),

net_i^k– quantum adder of the i^{th} neuron in the k^{th} layer of neuron weights, determined as the sum of weighted quantum values of the input signals connected to the k^{th} neuron layer,

$net_{i,im}^k$ - adder element net_i^k with the index $_{lm}$, a weighted quantum input signal to the k^{th} neuron layer of an artificial neural network with the pure nature state resulting from two mixed states (the quantum mixed number of the input signal and the quantum mixed number of the weight),

wherein the output signal from a specific layer of neurons, depending on all input signals, is determined from the relationship described by the activation function (here tansig ()):

$$\mathrm{y}(\mathrm{net}_{q\mathrm{i},}^{\mathrm{k}}(\mathrm{t})) = [1_{\mathrm{q}} - \mathrm{e}_{\mathrm{q}}^{-\mathrm{net}_{qi}^{k}}][1_{\mathrm{q}} + \mathrm{e}_{\mathrm{q}}^{-\mathrm{net}_{qi}^{k}}]^{-1}, \tag{17}$$

with, e.g.:

$$\mathrm{net}_{q\mathrm{i}}^{\mathrm{k=1}} = \mathrm{w}_{q11}^{1} \cdot \mathrm{u}_1 + \mathrm{w}_{q12}^{1} \cdot \mathrm{u}_2 + ... + \mathrm{w}_{q1\mathrm{n}}^{1} \cdot \mathrm{u}_{\mathrm{n}}.$$

It can be further noted that, as in the case of the Perceptron ANN, also in the case of the Quantum ANN, the overall model will consist of interconnected models of single neurons according to the connections resulting from the ANN architecture, in this case the Perceptron ANN architecture (see Mulawka, 1996; Osowski, 2020; Tadeusiewicz and Szaleniec, 2015). Dequantization is the conversion of quantum mixed numbers into real numbers in decimal notation. And since the nature of the ANN quantum model results from the matrix power of the number e – dependencies (16) and (17), hence the dequantization of a quantum mixed number into a real (or complex) number comes down to solving the problem consisting in the ability to raise the matrix to the matrix power or to the ability to bring the obtained result to a decomposable state, described by equation (7).

According to the available literature on the subject of quantum mathematics, quantum mechanics and quantum computing (see Le Belac, 2018; Chudy, 2011; Heller, 2016, Wright and Jordanov, 2017; and others) the above problem has not been fully resolved so far, due to the fact that there are entangled states of two quantum numbers, in this case a quantum weight and a quantum input signal, and in addition, the problems of this type, resulting from the entangled states, not from the states of decomposable quantum problems, described by equation (7), are now considered to be irreducible, and therefore unsolvable problems. In order to solve this problem, this work proposes the use of the method of quantum mixed numbers dequantization using ANN learned dequantization, based on the input quantities that are the elements of the net_i quantum matrix, and the output quantities y_i. The aforementioned ANN can be further used, e.g. in the simulation model, built in Simulink for the quantum quantization of the net_i matrix (see Ruciński, 2018).

6. Quantum computing

In quantum inspired calculations, the matrix form of operators and vectors is being used, the so-called finite-dimensional Hilbert space, hence each state of a quantum mixed number in a Hilbert space H_2 corresponds unambiguously to the matrix of the form:

$$l_m = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{18}$$

hence a quantum mixed number corresponding to a binary number will be expressed as:

$$l_m = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \beta_1 & \beta_2 & \ldots & \beta_n \end{bmatrix}. \tag{19}$$

When calculating matrices, a vector-matrix calculus can be used, which is associated with appropriate addition, multiplication, transposition of matrices, etc. (see Kaczorek, 1998). At this point, it is also worth supplementing the Dirac notation convention, which simplifies notations and facilitates calculations. Well, in the case of a quantum register, the value of the expression $|\alpha>$, where α is a variable, whose values are natural numbers, should be read like a notation in the binary system, supplemented with zeros on the left hand side to the length of n characters, which leads to the notation (see Sussking and Friedeman, 2016; Sawerwain and Wisniewska, 2015):

$$|\Phi\rangle = \sum_{k=0}^{2^n-1} \alpha_k \,|\rangle, \tag{20}$$

where the α_k coefficients meet the normalization conditions (the sum of squared modules equals 1).

The quantum computation process, using the input data, subjected to the quantization process and using the weights of the neural model, also subjected to the quantization process and appropriately selected activation functions, such as tansig() type functions, was carried out on the basis of an algorithm including the following basic steps (Tchórzewski and Ruciński, 2016; Ruciński, 2018):

Step 1. Converting real numbers in decimal notation to quantum mixed numbers using the system quantization method.

Step 2. Determining the weighted adders for individual outputs from the first layer of neurons, e.g. multiplication of the first mixed quantum number w_{11} and the mixed quantum number of the input quantities u_1 for neuron 1 in layer 1 for the first training pair, i.e. two quantum mixed numbers:

$$net_1^1 = w_{11}^1 \cdot u_1, \tag{21}$$

where:

$$w_{11}^{1} = \begin{bmatrix} 0.8256 & 0.8285 & 0.8294 & 0.8251 & 0.4162 & 0.8292 & 0.8305 & 0.8249 \\ 0.4175 & 0.4140 & 0.4141 & 0.4181 & 0.8267 & 0.4132 & 0.4116 & 0.4184 \end{bmatrix}$$

$$\begin{bmatrix} 0.8294 & 0.4196 & 0.8294 & 0.8312 & 0.8311 & 0.8277 & 0.8295 & 0.8297 & 0.8333 \\ 0.4129 & 0.8265 & 0.4129 & 0.4108 & 0.4109 & 0.4149 & 0.4128 & 0.4125 & 0.4081 \end{bmatrix}$$

$$u_{1} = \begin{bmatrix} 0.8256 & 0.8285 & 0.8294 & 0.8251 & 0.4162 & 0.8292 & 0.8305 & 0.8249 \\ 0.4175 & 0.4140 & 0.4141 & 0.4181 & 0.8267 & 0.4132 & 0.4116 & 0.4184 \end{bmatrix}$$

$$\begin{bmatrix} 0.8294 & 0.4196 & 0.8294 & 0.8312 & 0.8311 & 0.8277 & 0.8295 & 0.8297 & 0.8333 \\ 0.4129 & 0.8265 & 0.4129 & 0.4108 & 0.4109 & 0.4149 & 0.4128 & 0.4125 & 0.4081 \end{bmatrix}$$

As a result of the multiplication of the above two quantum mixed numbers in a matrix form, a square matrix with the dimension of 2 x 2 for each neuron, appearing in the first layer (and then in a similar way in the second layer), is obtained:

$$net_i^k(t) = \sum_{i,lm,k} \begin{bmatrix} net_{i\,\mathrm{lim}}^k & net_{i,lm}^k \\ net_{i,lm}^k & net_{i,lm}^k \end{bmatrix}. \tag{22}$$

So, e.g., for the first neuron of the first layer the following is obtained:

$$net_1^1(t) = \begin{bmatrix} 11.6712 & 5.9219 \\ 5.4275 & 3.2112 \end{bmatrix} + ... + \begin{bmatrix} 11.1596 & 5.3631 \\ 5.9722 & 2.2569 \end{bmatrix} = \begin{bmatrix} 46.4663 & 21.7725 \\ 21.9636 & 12.5232 \end{bmatrix} \tag{23}$$

Step 3. In this step, the value of the activation function, called tansig (), is determined for the first layer of ANN neurons, according to the relationship (17), and the activation function, called purlin () – for the second layer of neurons.

For example, when determining the output value from the hidden layer of the Artificial Neural Network, one can also use the expanding matrix exponents:

$$e^x = \sum_{k=0}^{\infty} \frac{1}{k!} x^k, \tag{24}$$

where:

x = -2*net*,

k = 4.

And then it is possible to obtain, for example, the following approximate possibility of determining the value of *net* in quantum form for individual 24 hours:

pot_e = 1 + mac_tymcz^1 + 1/2*mac_tymcz^2 + 1/6*mac_tymcz^3+
1/24 *mac_tymcz^4; (25)

and thus, the values of outputs y in the form of quantum states for individual 24 hours of the day:

$$y = f(net) = (1\text{- pot\_e})^*(1 + \text{pot\_e})\hat{}\text{-1}. \tag{26}$$

Step 4. Due to the existence of the matrix as an exponent of the number e (written in the activation function (24) as a quantum mixed number), the exponent e was dequantized by means of the learned ANN dequantization. The value of expression (24) becomes the input value for the layer 2 of the ANN neurons, which can also be determined analogously to the first layer. The outputs from the remaining neurons of the first layer (hidden layer) and the second layer (output layer) are determined in a similar way. The vector of quantum outputs from the output layer is also used in the clotting function of the evolutionary algorithm for each individual from the parental population, e.g. in relation to the average value of all individuals.

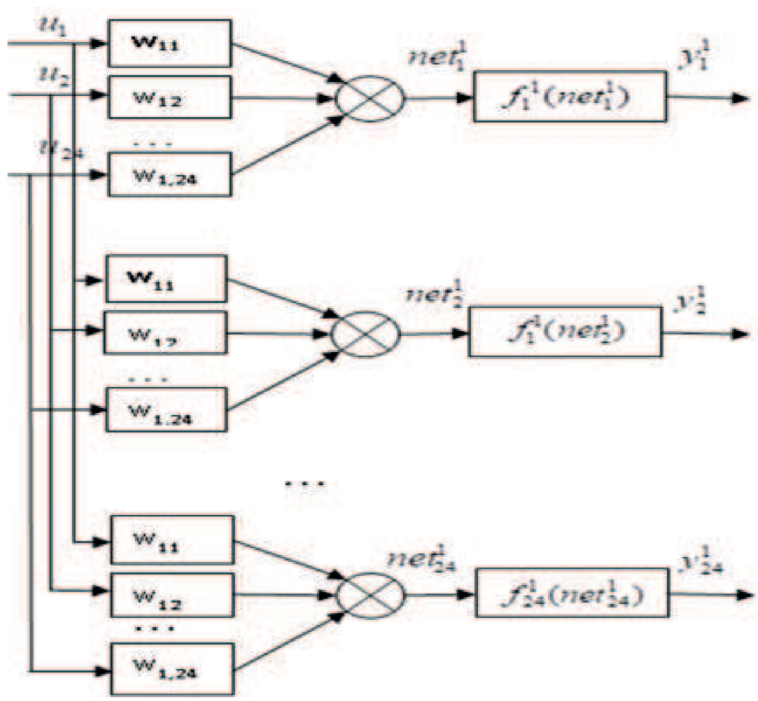

Figure 1. Block diagram of the net_i^k adder for the first layer of the PPE neural model. Symbols: u_1 ... u_{24} - ANN input signal (here the value of the volume of electricity supplied and sold in each of the 24 hours of a day) in [MWh], y_1 - ANN output signal (here volume-weighted average unit price for electricity supplied and sold in at a given hour of the day, in (PLN/MWh), w_{11}, $w_{12}\ldots w_{1,24}$ - ANN weights

$$y_1^1\left(\mathrm{net}_1^1\right)=\frac{1-e^{-2*\begin{bmatrix}46,4663 & 21,7735\\ 21,8636 & 13,5332\end{bmatrix}}}{1+e^{-2*\begin{bmatrix}46,4663 & 21,7735\\ 21,8636 & 13,5332\end{bmatrix}}}.$$

Hence, the following specific values of the individual 24 outputs from the ANN hidden layer are obtained:

val(:,:,1) =

Columns 1 through 5

0.773451649996588 0.761968556091428 0.807633442276137 0.819077107781521 0.758114626076228

0.633855302981331 0.647614020484381 0.589684850506775 0.573683441898281 0.652121318260107

Columns 6 through 10

0.810160735718816 0.820458711707111 0.799853290237652 0.807837618293933 0.839427001904776

0.586207797883606 0.571705783059702 0.600195563209195 0.589405108960879 0.543472454199069

Columns 11 through 15

0.817627582232084 0.815436489157857 0.791934624986455 0.841694276396647 0.800672417836415

0.575747459198316 0.578846553198607 0.610605887416395 0.539954391667597 0.599102394683905

Columns 16 through 20

0.808597147787839 0.822774070562434 0.814487063918959 0.812605562620117 0.831883090071349

0.588362687964976 0.568368567753463 0.580181715248484 0.582814035176610 0.554950920760875

Columns 21 through 24

0.791699485258787 0.824446541022465 0.841465998871876 0.814958854135236

0.610910734101940 0.565939838672003 0.540310070924609 0.579518822875136

Then, the runs of the Net Quantum-Inspired ANN and Perceptron ANN adders were compared, obtaining absolute errors close to zero. An example of the net value runs for 181 days for 0-1 is shown in Fig. 2.

In this way, the determined quantum states of 24 outputs from the ANN hidden layer are simultaneously 24 inputs to the ANN output layer. And since

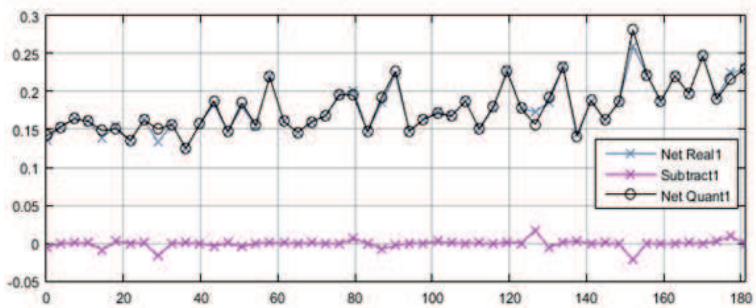

Figure 2. Discrepancy between the NetReal1 adder (nei_1) and the NetQuant1 adder (net_i) for the hidden layer of both models. Marking: crosses (x) denote the NetReal1 model (nei_1), circles (o) – NetQuant1 (net_i)

the linear function was assumed as the activation function in the output layer, the determined weighted net adders are also outputs from the ANN.

7. Conclusions and directions for further research

This paper proposes some new elements of the Quantum-Inspired Perceptron Artificial Neural Network method, which was verified on the example of the Day-Ahead Market System. The quantum inspired neural model was designed and implemented in the MATLAB and Simulink environment with the use of numerical data, listed on the Day-Ahead Market for the set of Matlab language (DAM).

The research performed required the use and implementation of appropriate new methods: quantization of real numbers in decimal notation, quantum computations using matrix-vector calculus, and building, implementing and teaching ANN dequantization.

It turned out that it is possible to develop an artificial intelligence method, such as an artificial neural network, inspired by quantum computing solutions to improve the parameters of the neural model, which was verified on the example of the Polish Electricity Exchange system.

The constructed models satisfactorily illustrate the behavior of the PPE system for the DAM. The attempts to improve the classic ANN model have shown that the improvement of the quality of the model with the methods appropriate for the classical Evolutionary Algorithms gave positive results. In the case of methods inspired by quantum computing, attempts were made to propose respective own solutions, based on linear algebra and vector-matrix calculus in Hilbert space.

The designed and implemented hybrid model of the DAM system consists of a neural model (Perceptron ANN), with weights modified by AE (neural-evolutionary model), which improved the parameters of the DAM PPE system model from the range between -0.17% and 0.18% to the range between -0.11% and 0.12%, and the designed and implemented AE-assisted and quantum-inspired neural model (neural-evolution-quantum model) improved the parameters of the model from the range of -0.11% to 0.12 % to the range of -0.04% to 0.05%, i.e. by an order of magnitude, notwithstanding the fact that the MSE error for the Perceptron ANN was already relatively low, as this is shown in Fig. 2.

The here reported research will be continued in the direction of testing the applied quantum inspiration method on other examples of neural modeling in order to check its effectiveness and efficiency, including such examples as, e.g., quantum-inspired neural modeling of the development of the National Power System, carried out with the use of relatively large training and testing files.

References

Adamowski, J. (2019) *Podstawy obliczeń kwantowych* [*Foundations of quantum calculations*; in Polish]. Wykłady na Wydziale Fizyki i Informatyki Stosowanej AGH, Kraków, http://www.ftj.agh.edu.pl [accessed: 14 September 2019].

Arabas, J. (2016) *Wykłady z algorytmów ewolucyjnych* [*Lectures in evolutionary algorithms*; in Polish]. WNT, Warszawa.

Bernhardt, C. (2020) *Obliczenia kwantowe dla każdego* [*Quantum calculations for everyone*; in Polish]. PWN, Warszawa.

Chudy, M. (2011) *Wprowadzenie do informatyki kwantowej* [*Introduction into quantum informatics*; in Polish]. OW EXIT, Warszawa.

Ciechulski, T. and Osowski, S. (2014) Badanie jakości predykcji obciążeń elektroenergetycznych za pomocą sieci neuronowych SVM, RBF i MLP [Analysis of quality of prediction of electric power loads with neural networks; in Polish]. *Przegląd Elektrotechniczny*, **90**, 8, 148-151.

Feynman, R. P., Leughton, R. B. and Sands M. (2014) *Feynmana wykłady z fizyki* [*Feynman's lectures in physics*; in Polish]. *Tom 3. Mechanika kwantowa* [*Volume 3: Quantum mechanics*]. PWN, Warszawa.

Giaro, K. and Kamiński, M. (2003) Wprowadzenie do algorytmów kwantowych [*Introduction into quantum algorithms*; in Polish]. AOW EXIT, Warszawa.

Heller, M. (2016) *Elementy mechaniki kwantowej dla filozofów* [*Elements of quantum mechanics for philosophers*; in Polish]. Copernicus Center Press, Kraków.

Hirvensalo, M. (2004) *Algorytmy kwantowe* [*Quantum algorithms*; in Polish]. WSiP, Warszawa.

Le Belac, M. (2018) *Wstęp do informatyki kwantowej* [*Introduction into quantum informatics* in Polish]. WN PWN, Warszawa.

Kaczorek T. (1998) *Wektory i macierze w automatyce i elektronice* [*Vectors and matrices in control and electronics*; in Polish]. WNT, Warszawa.

Kaczorek, T., Dzieliński, A., Dąbrowski, W. and Łopatka R. (2020) *Podstawy teorii sterowania* [*Foundations of control theory*; in Polish]. WNT, Warszawa.

Michalewicz, Z. (2003) *Algorytmy genetyczne + struktury danych = programy ewolucyjne* [*Genetic algorithms + data structures = evolutionary programs* in Polish]. WNT. Warszawa.

Mielczarski, W. (2000) *Rynki energii elektrycznej. Wybrane aspekty techniczne i ekonomiczne* [*Electric power markets. Selected technical and economic aspects*; in Polish]. ARE S.A. and Energoprojekt-Consulting S.A., Warszawa.

Miller, A. and Bućko, P. (2014) *Zastosowanie Sztucznych Sieci Neuronowych do Prognozowania Cen na Giełdzie Energii* [*Applying Artificial Neural Networks in Forecasting of Prices on Power Exchange*; in Polish]. *Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki Politechniki Gdańskiej*. 40.

Mulawka J. J. (1996). *Systemy ekspertowe* [*Expert systems*; in Polish]. WNT, Warszawa.

Obuchowicz, A. (2013) *Algorytmy ewolucyjne z mutacją stabilną* [*Evolutionary algorithms with stable mutation*; in Polish]. OW EXIT, Warszawa.

Osowski, S. (2020) *Sztuczne sieci neuronowe do przetwarzania informacji* [*Artificial neural networks for information processing*; in Polish]. OW PW, Warszawa.

Ruciński, D. (2018) *Modelowanie neuronalne cen na Towarowej Giełdzie Energii Elektrycznej wspomagane algorytmem ewolucyjnym oraz inspirowane obliczeniami kwantowymi* [*Neural modeling of prices on Electric Power Exchange assisted by an evolutionary algorithm and inspired by quantum calculations*; in Polish]. Doctoral dissertation under supervision of Prof. Jerzy Tchórzewski. IBS PAN, Warszawa.

Sussking, L and Friedeman, A. (2016) *Mechanika kwantowa. Teoretyczne minimum* [*Quantum mechanics. Theoretical minimum*; in Polish]. Prószyński i S-ka, Warszawa.

Sawerwain M. and Wiśniewska J. (2015) *Informatyka kwantowa. Wybrane obwody i algorytmy* [in Polish]. PWN, Warszawa.

Tadeusiewicz, R. and Szaleniec M. (2015) *Leksykon sieci neuronowych* [*Lexicon of neural networks*; in Polish]. Wydawnictwo Fundacji Projekt-Nauka, Wrocław.

Tchórzewski, J. (1992) *Cybernetyka życia i rozwoju systemów* [*Cybernetics of life and development of systems*; in Polish]. *Monografie* 22. Wydawnictwo WSRP, Siedlce.

Tchórzewski, J. (2013) *Rozwój systemu elektroenergetycznego w ujęciu teorii* *sterowania i systemów* [*Development of the electric power system in the perspective of control and systems theory*; in Polish]. OW PWr., Wrocław.

Tchórzewski, J. and Ruciński D. (2020) Design Methodology for the Quantum-Inspired Artificial Intelligence Methods on the Example of Price Determination at the Polish Electric Power Exchange. *Konferencja PAEE*, Warszawa.

Tchórzewski, J. and Ruciński, D. (2018) Quantum-inspired Artificial Neural Networks and Evolutionary Algorithms Methods Applied to Modeling of the Polish Electric Power Exchange Using the Day-ahead Market Data. *Information Systems in Management*, **7**, 3, 201–212.

Tchórzewski, J. and Ruciński, D. (2016) Quantum inspired evolutionary algorithm to improve parameters of neural models on example of Polish electricity power exchange. *IEEE Digital Library, Proceedings of Electric Power Networks (EPNet)*, 1-8.

Wright, J. and Jordanov I. (2017) Quantum inspired evolutionary algorithms with improved rotation gates for real-coded synthetic and real world optimization problems. *Integrated Computer-Aided Engineering*, **24**, 3, 203-223.