

Dariusz RUCIŃSKI¹

ORCID: 0000-0001-5458-9170

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

Artificial Neural Network based on mathematical models used in quantum computing

DOI: 10.34739/si.2022.27.02

Abstract. The article is a proposition of a new approach to building a neural model based on the system of Day-Ahead Market operating at TGE S.A. The reason for the proposed method is an attempt to find a better model for the DAM system. The proposed methodology is based on using mathematical models used in quantum computing. All calculations performed on learning the Artificial Neuron Network are based on operations described in Hilbert space. The main idea of calculations is to replace the data from the decimal system into the quantum state in Hilbert space and perform learning operations for a neural model of the DAM system in a special manner which rely on the teaching model for each position of the quantum register for all data. The obtained results were compared to the “classical” neural model with the use of a comparative model.

Keywords. Neural Modeling, Day-Ahead Market, Polish Power Exchange, Hilbert space, Quantum inspired neural network.

1. Introduction

The proposed research is a continuity of previously published articles [18-25, 33-34]. Modeling as a method of describing the phenomena taking place, in reality, is the subject of many studies [4, 13, 26, 29, 31]. One of the systems which is the subject of the research is the Day-Ahead Market (DAM) [38], so far works on DAM use both the classic approach to the neural model with the use of real numerical data [14, 18-22] and proposals for improving the model using various methods [23, 33]. The artificial neural networks themselves as a tool for building models are the subject of numerous publications and research [8, 10-17 25, 28-31, 37]. In this paper, an attempt was made to use the methods used to describe quantum phenomena [2, 3, 5-7, 9, 27, 32] to the method of modeling the neural system of the DAM system operating at TGE S.A. The concept of building an Artificial Neural Network based on quantum inspiration is related to the analysis of the essence of both quantum-inspired computations and the essence of modeling based on ANN. Quantum computing is based on Hilbert space, which is one of the linear vector spaces. In this space, strictly defined actions are possible, in this case, the important permissible operations are the summation of vectors, multiplication of a vector by a scalar (real or complex), an inner product, and an operator's action on a state vector.

The space defined in this way is sufficient for the opium of quantum states. An important property of quantum states is superposition, i.e., the existence of a state vector in space in a transient state, which only after measuring is subject to the so-called collapse, that is, it takes the value of one of the bases of space. Until the moment of measurement, the vector state may occupy any position in space, creating a phase space. In the two-dimensional Hilbert space, the base states are 0 or 1, which is important for this study. An important remark about quantum states understood as the state of a vector in real or imaginary space is the fact that in principle it is difficult to talk about its value understood classically.

The artificial neural network is based on real data, i.e., both input vectors, weight matrices, biases, activation functions, learning algorithms, etc. are based on operations on a set of real numbers. So, the implementation of a quantum-inspired ANN must take this important factor into account. As a result of converting numbers in the decimal system to binary values, we get a set of zeros and ones representing a given decimal value, a 12-element binary set, where the first element represents the largest order of magnitude and the last one is the smallest order of magnitude.

In order to implement the quantum-inspired ANN, it was assumed to build the ANN separately for each order of magnitude of the binary values, so that the set of binary values, corresponding to the real values, is trained by 12 ANNs (the value of 12 was adopted as sufficient to ensure the appropriate accuracy of the calculations). Binary values can be treated

as one-bit pure states in the Hilbert space, which can be quantized, i.e. transformed into mixed states by the developed procedure. In this way, ANN-based models were built for individual quantum states, with a constraint on the operations that can be performed specifically for Hilbert spaces.

2. Hilbert space

Hilbert's space plays a fundamental role in the mathematical description of the structure of laws related to quantum physics. Commonly, quantum mechanics is associated with jumps (collapses) and discontinuities. Nevertheless, one should be aware that one of the two basic mathematical structures describing a Hilbert space is the structure related to the concept of continuity, i.e., topological space. The topological space is a mapping:

$$(T, \text{fam}T), \quad (1)$$

where:

T - any collection in topological space,

famT – a family of its subsets that satisfy the following axioms:

- 1) the empty set and the set T belongs to famT,
- 2) the common part of two sets belonging to types belongs to types,
- 3) any sum of sets belonging to types belongs to types.

The family of T is a topology on T, and sets belonging to T are open sets. Two set points belonging to the set T are considered to be close to each other if they belong to the same open set. The concept of closeness understood in this way is used to define continuity.

Mapping a topological space $(T, \text{fam}T)$ into a topological space $(R, \text{fam}R)$:

$$f: (T, \text{fam}T) \rightarrow (R, \text{fam}R), \quad (2)$$

is called continuous if its inverse

$$f^{-1}: (R, \text{fam}R) \rightarrow (T, \text{fam}T), \quad (3)$$

assigns an open set in type, i.e., if:

$$(U) \in \text{fam}R, \text{ to } f: (U) \in \text{fam}T, \quad (4)$$

Another important issue in determining a Hilbert space is vector space properties. The vector (linear) space is defined as follows:

consists of a set \mathbf{V} , the elements of which are called vectors, operations on this space are possible, which are assigned to two vectors \mathbf{V} and \mathbf{W} belonging to the set \mathbf{V} described in this space by the vector $\mathbf{G} = \mathbf{V} + \mathbf{W}$, which also belongs to the same vector space,

from the operation which assigns the vector $a\mathbf{V}$ to the real or complex number a , called a scalar, and the vector \mathbf{V} - this operation is called the multiplication of the vector by the scalar.

The following conditions must be met:

commutability of the sum of vectors, i.e.,

$$\mathbf{V} + \mathbf{W} = \mathbf{W} + \mathbf{V}, \quad (5)$$

communication

$$\mathbf{V} + (\mathbf{W} + \mathbf{G}) = (\mathbf{V} + \mathbf{W}) + \mathbf{G}, \quad (6)$$

the existence of a null vector such that:

$$\mathbf{0} + \mathbf{V} = \mathbf{V} + \mathbf{0} = \mathbf{0}, \quad (7)$$

where:

\mathbf{V} - is any vector,

existence of the opposite vector $-\mathbf{V}$ for each vector \mathbf{V} ,

$$\mathbf{V} + (-\mathbf{V}) = \mathbf{0}, \quad (8)$$

the condition that multiplication of vectors by scalars is joint, i.e.:

$$\mathbf{V}(\mathbf{W}\mathbf{G}) = (\mathbf{V}\mathbf{W})\mathbf{G}, \quad (9)$$

the existence of a unit vector \mathbf{J} such that:

$$\mathbf{J}\mathbf{V} = \mathbf{V}. \quad (10)$$

For any vector \mathbf{V} ,

the ability to divide the multiplication with respect to addition, i.e. :

$$a(\mathbf{W} + \mathbf{G}) = a\mathbf{W} + a\mathbf{G}, \quad (11)$$

$$(a + b)\mathbf{V} = a\mathbf{V} + b\mathbf{V}. \quad (12)$$

The first four axioms define the properties of an Abelian (commutative) group, i.e., they state that each linear space is an Abelian group due to the addition (sum) of vectors. Axioms 5-6 define the properties of the multiplication of vectors by scalars. Axioms 7 guarantees proper cooperation of both activities.

It is also important that there are scalars in the vector space (a, b) . If they are real, the vector space is called the real \mathbf{R} , and if the complex vector space is called the complex \mathbf{Z} space. In models describing quantum mechanics, calculations are usually based on the space of complex numbers although operations based on real numbers are also possible.

The selected properties of the topological and vector spaces are the starting points in the construction of Hilbert spaces by combining their properties into one mathematical structure (model). It is possible to describe the vector space \mathbf{V} in the $\text{fam}\mathbf{V}$ topology, but in such a way that the vector structure and the topological structure are consistent with each other, that is:

- that for any pair of vectors $\mathbf{v}, \mathbf{w} \in \mathbf{V}$ and a given open set $\mathbf{U} \in \text{fam}\mathbf{V}$ such that $\mathbf{v} + \mathbf{w} \in \mathbf{U}$, there exist open sets $\mathbf{V}, \mathbf{W} \in \text{fam}\mathbf{V}$ containing vectors \mathbf{V} and \mathbf{W} , respectively, such that for any vectors $\mathbf{v}' \in \mathbf{V}$ and $\mathbf{w}' \in \mathbf{W}$ there is $\mathbf{v}' + \mathbf{w}' \in \mathbf{U}$.
- so that for any $\mathbf{v} \in \mathbf{V}$, any $a \in \mathbf{Z}$ and any open set $\mathbf{U} \in \text{fam}\mathbf{V}$ containing $a\mathbf{V}$ there exists an open set $\mathbf{V} \in \text{fam}\mathbf{V}$ containing \mathbf{v} and an open set \mathbf{S} containing a (in the natural topology of the space \mathbf{Z}), such that for any vectors $\mathbf{v}' \in \mathbf{V}$ and $a \in \mathbf{S}$ is $a\mathbf{v}' \in \mathbf{U}$.

A space with these properties is called a vector topological space. Such a space has two important properties, i.e., linearity and continuity.

- Other properties important for Hilbert spaces are:
 - the concept of vector length (in this case the vector norm),
 - the existence of boundaries (and thus the completeness of space),
 - orthogonality of vectors (more generally – the angle between vectors).

Another essential feature of space is its completeness, i.e., a metric space is complete if every sequence of convergence in the understanding of Cauchy's convergence reaches its limit in it. In a space with such properties, it is possible to describe the equations of quantum mechanics.

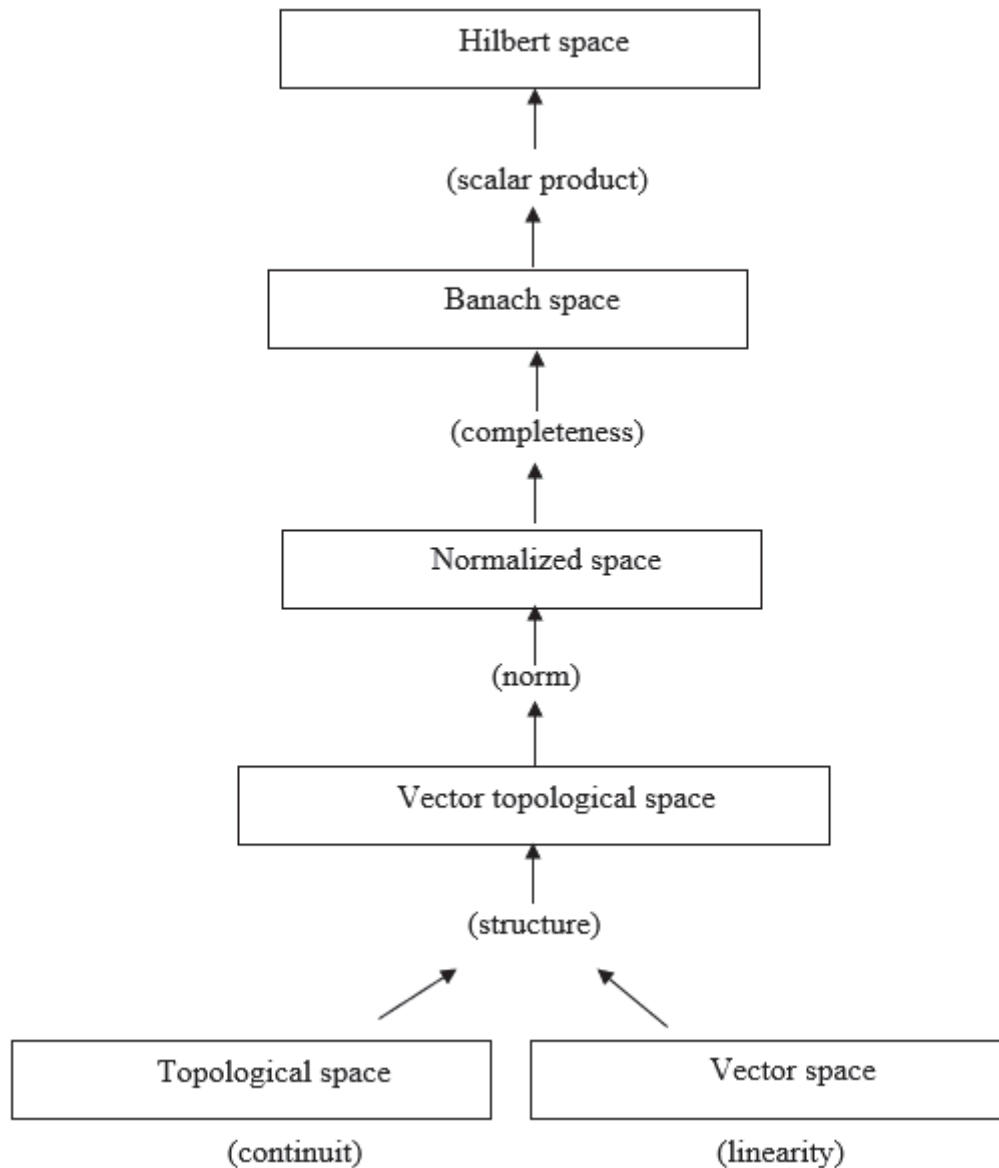


Figure 1. A diagram showing the constructions of the mathematical structure of a Hilbert space. shows that the construction of Hilbert spaces is based on two properties, continuity, and linearity. Source [6]

3. Hilbert space as the basis for the description of quantum operations

To interpret phenomena that take place in reality, e.g., physical, economic, etc., their mathematical models are built. The issues related to quantum mechanics have come to several different models, such as:

- 1) Model based on Hilbert space.
- 2) Feynman model.

- 3) Description model using C * - algebras.
- 4) Statistical presentation using a density matrix.

For this research, two models will be used, i.e., the use of a density matrix and one based on the Hilbert space. Hilbert space (H) together with the set of permissible operations in this space is widely recognized as a tool for describing the states of quantum mechanics. The notation introduced is also commonly used by Dirac² introducing the concept of a state vector, which is a vector in a Hilbert space. All permissible operations on a state vector are described by the properties of Hilbert spaces. An interesting and non-intuitive property of the state of a system at a given moment is the fact that its value only describes its direction (properties such as length or sense do not matter).

Actions on state vectors are performed using operators. Operators are certain mappings on a Hilbert space that transform a state vector described in this space into another state vector also described in the same space. The description of the space is made using observables [], which are defined as Hermitian operators [] with real eigenvalues. The real eigenvalues are the basis of the Hilbert space, i.e., they are used to describe the state vectors in this space.

Description of the state vector on a Hilbert space:

$$\psi = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha |0\rangle + \beta |1\rangle, \quad (13)$$

where:

ψ - state vector,

α - probability modulus expressed in the set of real or imaginary numbers that the state vector will be in the state described by the observable $|0\rangle$,

β - probability modulus expressed in the set of real or imaginary numbers that the state vector will be in the state described by the observable $|1\rangle$.

The quantum state is described by the direction, therefore only the relations (proportions) between α and β are important, not their values. After the measurement, the system is on one of the observables, and the probability of finding the state vector on a given

² Dirac also introduced such concepts as a bracket which consists of the vector ket $|\varphi\rangle$, the conjugate vector bra $\langle\varphi|$ and the operations and properties that occur between them.

observable, e.g., represented by α , is $|\alpha|^2$. Complex numbers³ α i β are called probability amplitudes and the square of their modulus is the actual probability number.

Vector norm state:

$$||\psi|| = \sqrt{\langle \psi | \psi \rangle} = \sqrt{[\alpha \ \beta] \begin{bmatrix} \alpha \\ \beta \end{bmatrix}} = \sqrt{\alpha^2 + \beta^2}, \quad (14)$$

An example of the description of quantum states in H^{2n} dimensional space:

Clean state $|00\rangle$, which is the state of the vector in H^4 dimensional space corresponds to the Kronecker product $H^2 \otimes H^2$ in this case $|0\rangle \otimes |0\rangle$, so:

$$|00\rangle = \begin{matrix} 1 \\ 0 \end{matrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

4. Chosen operation in Hilbert space

4.1. Linear operators

Generally, operator A on a vector space is a mapping of a given vector belonging to this space to a vector belonging to the same vector space described by operator A , which can be written:

$$A: |\psi\rangle \rightarrow |\psi'\rangle. \quad (15)$$

The transformation by the operator is linear, so the operator can also be called linear if it satisfies the axioms of linearity:

$$A(|\psi_1\rangle + |\psi_2\rangle) = A|\psi_1\rangle + A|\psi_2\rangle, \quad (16)$$

$$A(\alpha|\psi\rangle) = \alpha A(|\psi\rangle). \quad (17)$$

³ Complex numbers consist of real and imaginary parts, so in their nature their values are also two-dimensional.

Based on equations 16 and 17, it can be concluded that each observable is represented by a linear operator in a Hilbert space being the eigenstate of the Hermitian operator representing a measurable property. Thus, a measurement operation is an action on a state vector with a particular operator represented by the observable.

4.2. Density matrices

In the case of a complex system of Hilbert spaces, the information about the state vector is based on the so-called partial trace (Tr). Determination of the partial trace is based on the density matrix or the density operator (ρ). The density matrix is determined for a given Hilbert space and is expressed by the formula:

$$|\psi\rangle\langle\psi| = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \alpha & \beta \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha\beta \\ \beta\alpha & \beta^2 \end{bmatrix}. \quad (18)$$

It is known from the principle of superposition that

$$\alpha^2 + \beta^2 = 1.$$

As you can see, such values are on the main diagonal of the density matrix, so the trace value Tr is determined as the sum of the diagonal of the density matrix. The sum of the diagonal values of the density matrix is called the trace of the density matrix and is called Tr.

The concept of the density matrix is also related to the density operator ρ , which is determined for quantum states in more than one Hilbert space and is defined as:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| = \sum_i p_i \begin{bmatrix} \alpha_i^2 & \alpha_i\beta_i \\ \beta_i\alpha_i & \beta_i^2 \end{bmatrix}, \quad (19)$$

where:

p_i – the probability of being in the given quantum state described by given vector.

4.3. State vector measurement

The measurement of the state vector of a quantum system is determined by the base systems. At the time of measurement, the vector value will assume one of the values represented by these systems, also called observables or bases. Until the moment of measurement, the vector state is in a superposition of states representing the observables. If the state vector is in the eigenvector state of a given Hermitian operator (observables), then after the measurement its value will be

equal to this observable, it is the so-called clean condition and the measurement result is determined. In most cases, however, the state vector is located in the so-called superposition of states, and at the time of measurement, its state undergoes the so-called collapse, i.e., it is in one of the eigenstates of the Hermitian matrix. The probability of finding the state vector on a given observable (i) can be defined as:

$$p(i) = |\psi \rangle M_i^\dagger M_i |\psi \rangle = |\lambda_i|^2, \quad (20)$$

where:

M_i - i-th Hermitian measurement operator,

M_i^\dagger - Conjugate i-th Hermitian measurement operator,

λ_i - i-th observable.

The normalized value of the state vector V on the observable (i) after measurement is determined by the relationship:

$$V_i = \frac{M_i |\psi \rangle}{\sqrt{|\psi \rangle M_i^\dagger M_i |\psi \rangle}}. \quad (21)$$

The use of the dependencies (20) and (21) allows determining both the probability of finding the state vector on a given observable as well as the determination of its normalized size.

5. Preparation of quantum data for the creation of models based on the Artificial Neural Network

The premise of Quantum-Inspired ANN is that it can be implemented for any order of magnitude register describing the quantum state of a given real value. Such a quantum state is created as a result of converting a numerical value stored in the decimal number system into a value stored in the binary system. Assuming of the accuracy taken into research was representing any decimal number by 12 values in the binary system.

The number so stored is then transformed into a 12-element matrix of quantum mixed states. As a result of this approach, each value stored in the decimal number system representing data for the input to the neural network, the values of weights and biases and output data are represented by a 12-element matrix of quantum mixed states (12x2).

The essence of the new approach is to build a Quantum-Inspired ANN for each cubit of the quantum mixed state, which leads to the needs to build 12 quasi-parallel ANNs from the youngest cubit to the oldest.

In the first stage, the input values, i.e., the volume of electricity supplied and sold (dataset of 24-dimensional vector), and the output values ((dataset of 24-dimensional vector), i.e., the volume-weighted average unit price of electricity obtained in each hour of the day, were exchanged.

5.1. Binary representation

As a result of the first stage of data preparation, we obtain a binary set representing a given decimal value (inputs and outputs). In the case of this study, it was a 12-element binary set. Each element of a binary set represents the corresponding order of magnitude, i.e. the first element represents the largest order of magnitude and the last - the smallest.

Due to the adopted assumptions, ANN was initialized in the MATLAB environment based on actual data using the MATLAB function[1] :

```
INIT_test = fitnet(24, 'trainlm');  
INIT_test = configure(INIT_test, input, target);
```

where:

INIT_test – the name of the Artificial Neural Network,
fitnet - built-in Matlab function to adjust the hidden layer of the network INIT_test,
trainlm – function defining the network training method in this case Levenberg - Marquardt,
configure – function to configure the input and output layers of the network to configure the hidden layers and the output layer to input and output values,
input – a set of input data,
target – the target data set.

As a result of the network initialization, randomly generated values of weights and biases were obtained, which, similarly to the input and output data, were also converted into binary values.

5.2. Quantum representation

The next stage of the quantum-inspired ANN construction was the conversion of binary values into state vectors and density matrices in a Hilbert space. It was assumed that individual

binary values represent pure states in this space. So, a given value, e.g. an input data represented by a real value of 0.0996, is represented by a binary value of 0.000110010111.

The next step was the quantization of pure states into quantum mixed states. Thus, quantum mixed states are made up of the pure states ket 0 and ket 1 for the individual bits of the binary number. The idea of creating a quantum mixed state boils down to using the property that after observation, the probability of a quantum bit in the state $|0\rangle$ is $|\alpha|^2$, similarly, that it will be in the state $|1\rangle$ is $|\beta|^2$. So, assuming that $\alpha = \beta$ can be written $2|\alpha|^2=1$, so:

$$\alpha = \beta = \frac{\sqrt{2}}{2} \approx 0,71 \quad (22)$$

Based on the dependence (22), it is possible to determine the probability module for pure states, e.g. for pure state 1, the dominant interval of the probability module is in the range $0.71 \leq \alpha \leq 1$, and this range was randomized. The supplementary interval is determined by the principle of superposition. The same was done for the value of 0. When using the above-mentioned the algorithm, the pure states were changed into quantum mixed states, e.g., for 000110010111, the following was obtained:

0.800	0.890	0.990	0.626	0.475	0.850	0.850	0.661	0.770	0.527	0.558	0.558
0.600	0.456	0.141	0.780	0.880	0.527	0.527	0.750	0.638	0.850	0.830	0.830

5.3. Density matrices

When analyzing the principle of building the ANN model, it can be noticed that the weight matrices are linear operators for the input data, which are vectors. Taking this property into account, in the quantum-inspired ANN, the quantum states of weights were transformed into their density matrices according to the equation (18). Below is an example of successive transformations for hidden layer weight values $\mathbf{W}_{1,1}$.

- decimal value: *0.361760584314686*,
- binary value: *0.010111001001*,
- quantum state:

0.820	0.014	0.900	0.243	0.527	0.704	0.890	0.770	0.510	0.920	0.890	0.475
0.572	1.000	0.436	0.970	0.850	0.710	0.456	0.638	0.860	0.392	0.456	0.880

- the density matrix for the first qubit of a quantum state:

0.6724	0.4693
--------	--------

0.4693	0.3276
--------	--------

6. The use of quantum mathematical models to create models based on the Artificial Neural Network

The basic assumption of QiANN is the assumption that it is possible to implement QiANN for each order of magnitude of binary values (12 orders of magnitude), and thus the corresponding quantum states so that the value of a given order for the input set, set of weights, biases and output data is treated as one network. After learning the QiANN created in this way, quantum states were measured and binary values were re-converted to decimal values.

Procedure algorithm

1. Separation of the youngest order of quantum states for input and output and the base weight matrix.
2. Determine the number of learned networks (Ls) for each order of magnitude
3. Calling the learning function for separated sets
4. Record of the obtained QiANN model
5. If the value (Ls) has been reached, go to pt. 6 otherwise go to point. 3.
6. If not all orders of quantum states have been processed, isolation of the next higher order of quantum states for input and output and the base weight matrix and going to p. 3 otherwise go to point 7.
7. Calculation of the mean quantum states for each order.
8. Measurements of average quantum states - reduction to the base 0 or 1.
9. Converting base states into a 12-element numeric sequence analogous to binary values.
10. Converting binary values to real numbers
11. Calculation of the mean square error (MSE) as the difference between the actual values of the target and the values obtained from the QiANN models from point 10.

Description for figure 2:

$uk_1^1 \dots uk_{12}^1$ input values of quantum states for successive networks from 1 to 12,

$wk_1^{1,1} \dots wk_{12}^{1,1}$ - values of the density matrix for the weights of the hidden layer for the successive nets from 1 to 12,

$b_1^1 \dots b_{12}^1$ - density matrix values for the hidden layer bias for successive networks from 1 to 12, $nk_1^1 \dots nk_{12}^1$ - values of the state vector for the sum of the products of the input and weights and biases for the hidden layer for successive networks from 1 to 12,

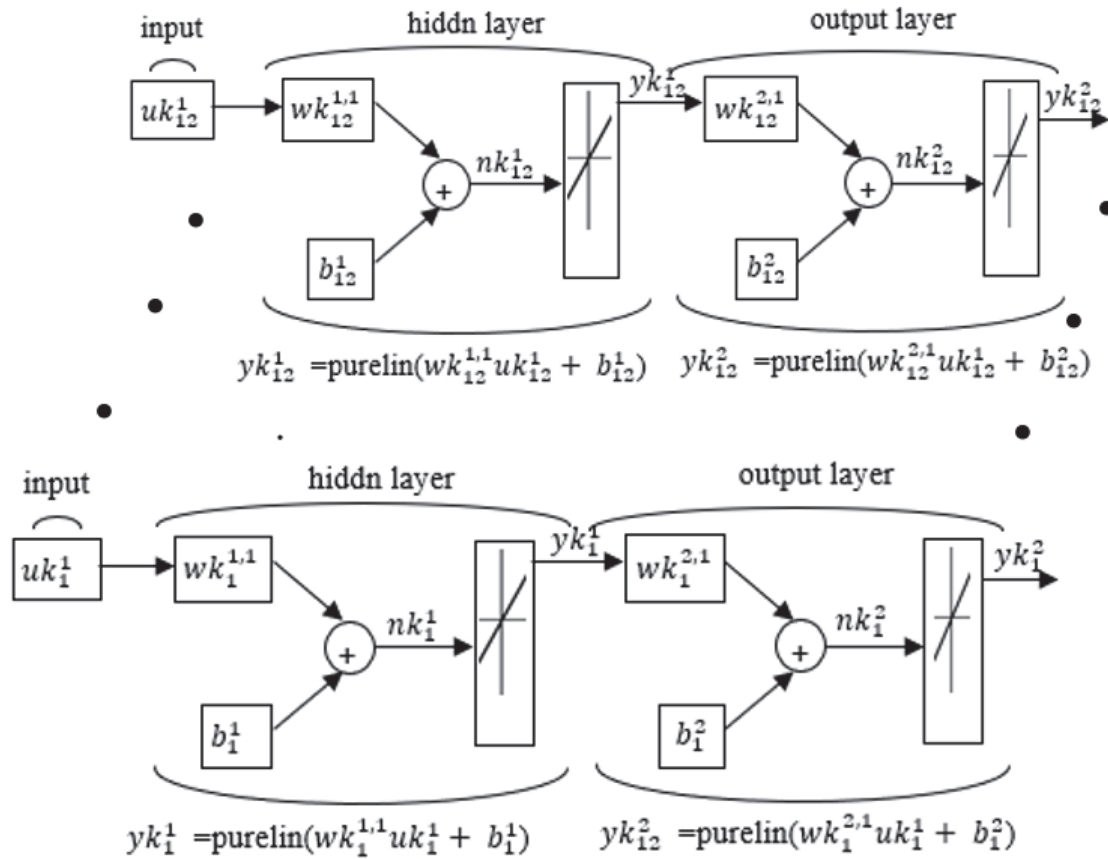


Figure 2. The idea of quantum computing. Source: own elaboration based on [1].

$yk_1^1 \dots yk_{12}^1$ - values of the state vector of the hidden layer activation function for successive networks from 1 to 12, $wk_1^{2,1} \dots wk_{12}^{2,1}$ - the values of the density matrix for the weights of the output layer for the successive nets from 1 to 12,

$b_1^2 \dots b_{12}^2$ - values of the density matrix for the bias of the output layer for subsequent networks from 1 to 12,

$nk_1^2 \dots nk_{12}^2$ - values of the state vector for the sum of the products of the values of the hidden layer activation function and weights and biases for the output layer for successive networks from 1 to 12,

$yk_1^2 \dots yk_{12}^2$ - values of the vector of states of the output layer activation function for successive networks from 1 to 12.

Table 1. Representation of the data used in the calculations point 5.1. Source: Own study.

Quantum register of input values	Hidden layer weights	Output layer weights	Quantum register of output values
$uk_1^{1,181}$ $uk_1^{1,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$	wk_1^1 $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	wk_1^2 $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$yk_1^{2,181}$ $yk_1^{2,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$
...
uk_{12}^{181} uk_{12}^1 $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$	wk_{12}^1 $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	wk_{12}^2 $\begin{bmatrix} \alpha\alpha & \alpha\beta \\ \beta\alpha & \beta\beta \end{bmatrix}$	$yk_{12}^{2,181}$ $yk_{12}^{2,1}$ $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$ [...] $\begin{bmatrix} \widehat{\alpha} \\ \widehat{\beta} \end{bmatrix}$

6.1. Learning quantum-inspired neural network

Due to its specificity, limited by the activities described in the Hilbert space, it came down to the proper implementation of network learning. QiANN was learned in a manner analogous to the classical network learning algorithm, i.e.:

- 1) Determining the value of nk - for the hidden layer.
- 2) Determining the yk value - for the hidden layer.
- 3) Determination of the value of nk - for the output layer.
- 4) Determination of the yk value - for the output layer.
- 5) Changes in the values of the weights and biases density matrices for both learned layers using the error backpropagation algorithm.

Determining the value of nk comes down to operating on state vectors, which are uk for the hidden layer and yk for the output layer, using the k operators for the appropriate network layer. An important issue was the assessment of the value of nk in terms of its value, if the value of nk exceeded its order of magnitude, its excess was transferred to the next larger order of magnitude (based on the transfer bit). This operation is aimed at transferring information about the value of a given state vector between successive networks. Due to constraints related to the state vector, purelin was used as an activation function for all layers.

6.2. Backpropagation error algorithm

The error backpropagation algorithm was also based on the classical approach and consisted of changing the value of the weight and bias density matrix in the following steps:

- determining the difference between the training data and the data from the Δnk model,
- determining changes in weights based on Δnk for individual weights (for the purelin function the derivative net equals 1),
- calculating new values of the weight density matrix for the weights of the initial layer,
- determination of Δnk for the hidden layer,
- calculating changes in weights and biases for the weights of the hidden layer,
- calculation of new values of the hidden layer weight density matrix.

All the above operations (as opposed to the classical backpropagation algorithm) were carried out based on vector-matrix calculus.

6.3. Measurement of quantum state

To compare the results of the Quantum Inspired ANN (QiANN) with the "Classic" ANN, a procedure for measuring the quantum states of the QiANN was developed.

It consists of:

- Collapses of quantum states to the base value, as a result of which individual registers take the value 1 or 0,
- the vector obtained in this way, representing a binary value, is converted into a real value,
- the actual values are compared with the target values from the Day-Ahead Market of the Polish Power Exchange.

7. Conclusions and directions for further research

Below are examples of the obtained results for normalized⁴ real data, table 2, the normalized price obtained from the Perceptron ANN model, table 3, and the normalized price obtained from the quantum-inspired Artificial Neural Network, table 4.

⁴ The process of preparing and learning the ANN system model is related to the need for initial preparation of training data (volume of energy and weighted by volume average price for given hour of the day), including normalization. In this case, normalization was carried out based on min-max normalization which bringing the data to the range $\langle 0; 1 \rangle$.

Table 2. The first three days of normalized actual training data. Source: Own elaboration.

Day no	The next hours of the day											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.259	0.221	0.213	0.193	0.183	0.177	0.154	0.143	0.13	0.128	0.146	0.146
2	0.115	0.093	0.094	0.095	0.095	0.132	0.278	0.422	0.43	0.423	0.427	0.423
3	0.499	0.485	0.471	0.473	0.495	0.538	0.529	0.559	0.5	0.5	0.501	0.498
	13	14	15	16	17	18	19	20	21	22	23	24
1	0.148	0.164	0.166	0.17	0.273	0.269	0.284	0.29	0.271	0.233	0.213	0.196
2	0.422	0.43	0.454	0.5	0.547	0.56	0.55	0.537	0.518	0.517	0.479	0.473
3	0.508	0.516	0.541	0.557	0.625	0.625	0.615	0.607	0.59	0.574	0.599	0.592

Table 3. The first three days of baseline data from ANN. Source: Own elaboration.

Day no	The next hours of the day											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.4329	0.397	0.3911	0.401	0.4239	0.4439	0.4467	0.4703	0.4783	0.4884	0.4708	0.4504
2	0.3277	0.3057	0.2858	0.2916	0.3166	0.3116	0.3247	0.3824	0.3396	0.3281	0.3325	0.3201
3	0.4083	0.3847	0.3674	0.3732	0.3963	0.4447	0.4904	0.5047	0.4845	0.4767	0.4773	0.4779
	13	14	15	16	17	18	19	20	21	22	23	24
1	0.4424	0.4339	0.4578	0.4632	0.4249	0.4182	0.4187	0.4395	0.475	0.4936	0.4426	0.4787
2	0.3103	0.3127	0.3448	0.3611	0.3711	0.3644	0.3869	0.4091	0.408	0.4785	0.4299	0.4478
3	0.4845	0.4876	0.5081	0.5139	0.5319	0.535	0.54	0.5365	0.5261	0.5205	0.5345	0.5244

Table 4. The first three days of output data obtained from QiANN

Day no	The next hours of the day											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.5425	0.3489	0.0605	0.4668	0.45	0.676	0.302	0.4329	0.5881	0.1138	0.1074	0.103
2	0.8989	0.0459	0.1426	0.4885	0.134	0.6736	0.9275	0.4045	0.3176	0.6345	0.6084	0.3489
3	0.7244	0.1765	0.7139	0.1377	0.3899	0.3845	0.4272	0.6387	0.0039	0.6355	0.5769	0.6899
	13	14	15	16	17	18	19	20	21	22	23	24
1	0.9138	0.0063	0.2209	0.147	0.5564	0.0603	0.2542	0.8062	0.647	0.1948	0.3345	0.4219
2	0.0486	0.0369	0.1147	0.1797	0.6147	0.8416	0.5237	0.4617	0.2661	0.9429	0.5283	0.8684
3	0.7886	0.1592	0.2542	0.0613	0.2463	0.281	0.2996	0.3149	0.1277	0.1716	0.5352	0.4021

By analysing the obtained results, it can be concluded that it is possible to implement a quantum-inspired Artificial Network to teach the model of the Day-Ahead Market System at TGE S.A. Referring to the obtained results, it can be seen that QiANN tends to average the results more, i.e., it is more resistant to disturbances. However, means error of MSE value is greater for it than for the Perceptron neural network and is 0.09 and for Perceptron ANN MSE is 0.003.

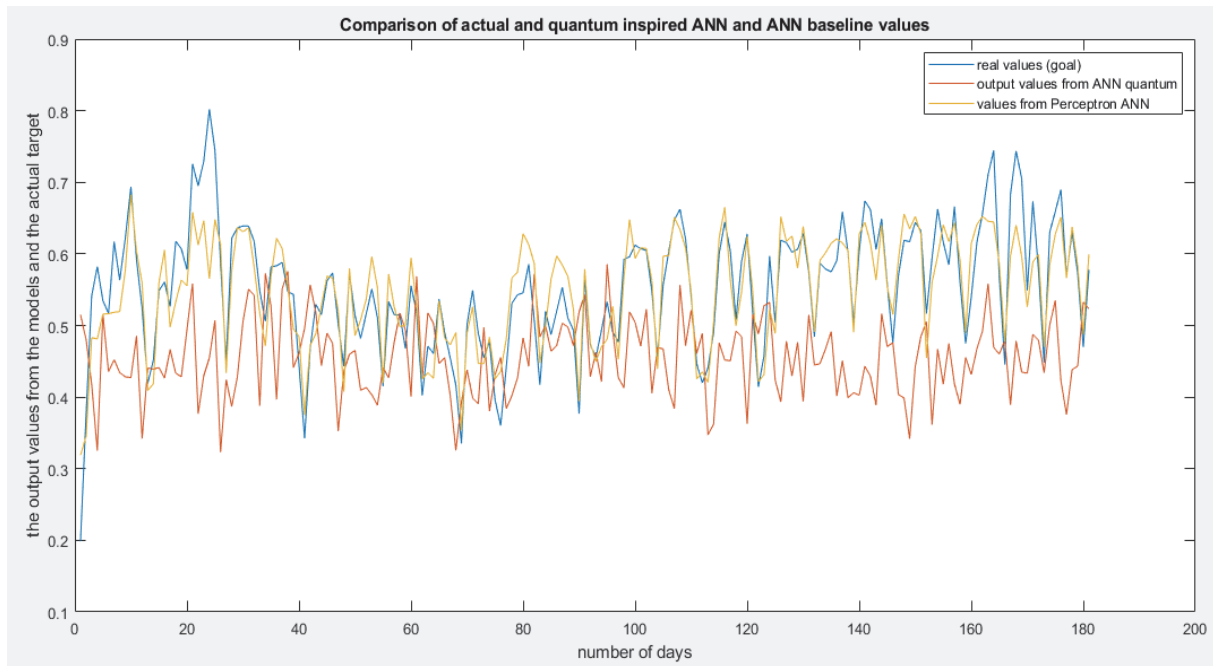


Figure 3. Comparison of the actual normalized values of the price course with the Perceptron ANN model and the quantum-inspired model. Symbols: x-axis(number of the days) - consecutive days of the examined period (in this case 181 days), y-axis (average value of the normalized price). Color blue – real normalized values of price, color yellow – Perceptron ANN output, the color red – quantum inspired ANN output. Source own elaboration in the MATLAB environment.

It can also be seen that quantum-inspired neural networks, whose operation is based on mathematical models appropriate for quantum computing, are limited to the actions possible for these models, i.e., on Hilbert space. The obtained QiANN model, despite the fact that it is the result of many studies and trials, is a proposal for a certain method of implementing neural models and is the subject of further work aimed at using the potential of neural models as well as using quantum inspirations.

References

1. Beale M. H. [at all] (1992-2019) Neural Network Toolbox™ User's Guide, by The MathWorks, Inc, pp. 846.
2. Bernhardt Ch. (2020) Obliczenia kwantowe dla każdego (English: Quantum computing for everyone). WN PWN, Warszawa, stron 202.
3. Chudy. M. (2012) Wprowadzenie do informatyki kwantowej (English: Introduction to quantum computing). AOW EXIT, Warszawa, stron 84.
4. Faliński M. (2020) Wstęp do sztucznej inteligencji (English: Introduction to artificial

- intelligence). WN PWN, Warszawa, stron 332.
5. Giaro K., Kamiński M. (2003). Wprowadzenie do algorytmów kwantowych (English: Introduction to quantum algorithms). AOW EXIT, Warszawa, stron 165.
 6. Heller M. (2016) Elementy mechaniki kwantowej dla filozofów.(English: Elements of quantum physics for philosophers). Copernicus Center Press, stron 191.
 7. Hirvensalo M. (2004) Algorytmy kwantowe (English: Quantum algorithms). WSiP, stron 244.
 8. Jankowski N. (2003) Ontogeniczne sieci neuronowe. O sieciach zmieniających swoją strukturę (English: Ontogenic neural networks. With networks changing their structure). AOW EXIT, Warszawa, stron 312.
 9. Johnston E. R., Harrigan N., Gimeno-Segovia M. (2020) Komputer kwantowy. Programowanie, algorytmy, kod (English: Quantum computer. Programming, algorithms, code). Hellion, Warszawa, stron 278.
 10. Kosiński R. A. (2002) Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos (English: Artificial neural networks. Nonlinear dynamics and chaos). WNT, Warszawa, stron 195.
 11. Korbicz J., Obuchowicz A., Uciński D. (1994) Sztuczne sieci neuronowe. Podstawy i zastosowania (English: Artificial neural networks. Basics and applications), Problemy Współczesnej Nauki. Teoria i zastosowania, Informatyka, AOW PLJ, stron 251.
 12. Mańdziuk J. (2000) Sieci neuronowe typu Hoppfielda (English: Hoppfield neural networks). Teoria i przykłady zastosowań. AOW EXIT, Warszawa, stron 262.
 13. Marecki J. (2001) Metody sztucznej inteligencji (English: Artificial intelligence methods),. WSiZ, Bielsko-Biała, stron 115.
 14. Monteiro, [at all] (2016) Short-Term Price Forecasting Models Based on Artificial Neural Networks for Intraday Sessions in the Iberian Electricity Market. *Energies*, 9, 721 stron 24.
 15. Obuchowicz A., Patan K. (1998) 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, s. 101-106.
 16. Osowski S. (2013) Sieci neuronowe do przetwarzania informacji (English: Neural

- networks for information processing). OW PW, Warszawa, stron 422.
17. Patterson J., Gibson A. (2017) Deep Learning. Praktyczne wprowadzenie (English: Deep Learning. A practical introduction), Helion, Warszawa, stron 451.
 18. Ruciński D. (2020) 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), pp 25-35
 19. Ruciński D., (2019), 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), pp. 77-93
 20. Ruciński D. (2018) Modelowanie neuronalne cen na Towarowej Gieldzie 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, str. 137.
 21. Ruciński D. (2017) The neural modelling in chosen task of Electric Power Stock Market. *Studia Informatyka. System and Information Technology*. Vol. 1-2(21). Publishing House of University of Natural Science, pp. 1-22.
 22. Ruciński D., Tchórzewski J. (2016). 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) pp. 215-226.
 23. Ruciński D. (2016) 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.
 24. Ruciński D., Kłopotek M., Tchórzewski J. (2005) Self-Organizing Wireless Ad-hock Sensor Networks. *Studia Informatica. Systems and Information Technology*, Volume: 1(5), Publishing House of University of Podlasie, pp. 69-80.
 25. Rutkowska D., Piliński M., Rutkowski L. (1997). *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte* (English: Neural networks, genetic algorithms and fuzzy systems), WN

- PWN, Warszawa - Łódź, stron 411.
26. Rutkowski L. (2020) Metody i techniki sztucznej inteligencji (English:). WN PWN, Warszawa, stron 435.
 27. Sawerwain M., Wiśniewska J. (2015) Informatyka kwantowa. Wybrane obwody i algorytmy (English: Quantum informatics. Selected circuits and algorithms). WN PWN, Warszawa, stron 371.
 28. Sengupta, S. (2014-02). Neural Networks and Applications. <http://nptel.ac.in/courses/117105084>, [dostęp: 15-08-2017, 04-02-2021].
 29. Tadeusiewicz R. (2021) Archipelag Sztucznej Inteligencji (English: The Artificial Intelligence Archipelago). AOW EXIT, Warszawa, stron 125.
 30. Tadeusiewicz R., Korbicz J., Rutkowski L., Duch W. (2013) Sieci neuronowe w inżynierii biomedycznej (English: Neural networks in biomedical engineering). Seria: Inżynieria biomedyczna. Podstawy i zastosowania, tom 9, Akademicka OW EXIT, Warszawa, str. 746.
 31. Tadeusiewicz R. (2013) Biocybernetyka. Metodologiczne podstawy dla inżynierii biomedycznej (English: Biocybernetics. Methodological foundations for biomedical engineering). WN PWN, Warszawa, str. 236.
 32. Tchórzewski J. (2021) 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, stron 343.
 33. Tchórzewski J., Ruciński D. (2019) 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, pp. 1-8.
 34. Tchórzewski J., Ruciński D., Domański P. (2018) Artificial neural network inspired by quantum computing solutions using the movement model of the PR-02 robot. ITM Web of Conferences 19, pp. 1-2.
 35. Tchórzewski J. (1992) Cybernetyka życia i rozwoju systemów (English: Cybernetics of life and systems development). Wyd. WSRP, Siedlce, stron 408.
 36. Tchórzewski J., Kłopotek M. (2001), A case study in neural network evolution. Prace Instytutu Podstaw Informatyki Polskiej Akademii Nauk, Nr 943, pp. 1-12.

37. Żurada J., Barski M., Jędruch W. (1996) Sztuczne sieci neuronowe. Podstawy teorii i zastosowania (English: Artificial neural networks. Fundamentals of theory and application), WN PWN, Warszawa, stron 375.
38. www.tge.pl – the website of Polish Power Exchange. [access: 04-09-2022].