UDC 004.89
doi:10.31799/1684-8853-2020-6-12-20

Articles

Interpretation of a trained neural network based on genetic algorithms

Pimenov V. I.[a], *Dr. Sc., Tech., Professor, orcid.org/0000-0002-7228-3009, v_pim@mail.ru*
Pimenov I. V.[b], *PhD, Tech., Associate Professor, orcid.org/0000-0002-1954-6463*
[a]*Saint-Petersburg State University of Industrial Technologies and Design, 18, B. Morskaya St., 191186, Saint-Petersburg, Russian Federation*
[b]*Admiral Makarov State University of Maritime and Inland Shipping, 5/7, Dvinskaya St., 198035, Saint-Petersburg, Russian Federation*

Introduction: *Artificial intelligence development strategy involves the use of deep machine learning algorithms in order to solve various problems. Neural network models trained on specific data sets are difficult to interpret, which is due to the "black box" approach when knowledge is formed as a set of interneuronal connection weights.* ***Purpose:*** *Development of a discrete knowledge model which explicitly represents information processing patterns encoded by connections between neurons.* ***Methods:*** *Adaptive quantization of a feature space using a genetic algorithm, and construction of a discrete model for a multidimensional OLAP cube with binary measures.* ***Results:*** *A genetic algorithm extracts a discrete knowledge carrier from a trained neural network. An individual's chromosome encodes a combination of values of all quantization levels for the measurable object properties. The head gene group defines the feature space structure, while the other genes are responsible for setting up the quantization of a multidimensional space, where each gene is responsible for one quantization threshold for a given variable. A discrete model of a multidimensional OLAP cube with binary measures explicitly represents the relationships between combinations of object feature values and classes.* ***Practical relevance:*** *For neural network prediction models based on a training sample, genetic algorithms make it possible to find the effective value of the feature space volume for the combinations of input feature values not represented in the training sample whose volume is usually limited. The proposed discrete model builds unique images of each class based on rectangular maps which use a mesh structure of gradations. The maps reflect the most significant integral indicators of classes that determine the location and size of a class in a multidimensional space. Based on a convolution of the constructed class images, a complete system of production decision rules is recorded for the preset feature gradations.*

Keywords *— classification, deep machine learning, neural network, genetic algorithm, multidimensional OLAP cube, decision rule, semantic interpretation, visualization of classes.*

For citation: Pimenov V. I., Pimenov I. V. Interpretation of a trained neural network based on genetic algorithms. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 6, pp. 12–20. doi:10.31799/1684-8853-2020-6-12-20

Introduction

It is known, that the up-to-date artificial intelligence research and technology uses deep machine learning algorithms, which improves quality of modern business processes in the areas of logistics management, optimize supply planning, financial operations, production processes, predict risks, increase customer satisfaction, diagnose diseases, selects dosages of drugs and solve other narrow classification problems, as well as the creation of a strong artificial intelligence, universal in application to various tasks [1–7].

But, the deep neural network models, which trained on specific data sets, are difficult to interpret for both human mind and machine algorithms. Also, the creation of a strong artificial intelligence, which capable of adapting and interacting with the external environment is an actual complex scientific challenge [8, 9].

The difficulty of verbalizing the output of deep learning and clearly clarification of the obtained result (i. e. why the model made those or another decisions) is associated with the using of the "black box" model [10], in which in the process of training neural network, the "knowledge" is formed from the sets of links weight between the neighbor neurons. Herewith, visualization and synthesis of new solutions can be carried out using generative adversarial networks [11, 12]. In this case, one network generates artificially created examples of complex objects, and the other network evaluates their reality based on a training set, which allows performing creative tasks, generating variants and prototypes of multidimensional objects.

The creation of a universal algorithm for strong artificial intelligence can be based on the method of complex use of multidimensional data analysis, aimed at transforming a multidimensional feature space into a finite set of classes, and then building a basic discrete code that stores information in a compressed form about a set of features characteristic of a given class. This discrete form of knowledge, not only provides the ability to interpret themselves by the various methods, e. g., mathematical production rules, but also allows to made cognitive visual-

ization of multidimensional classes using descriptive (explanatory) variables.

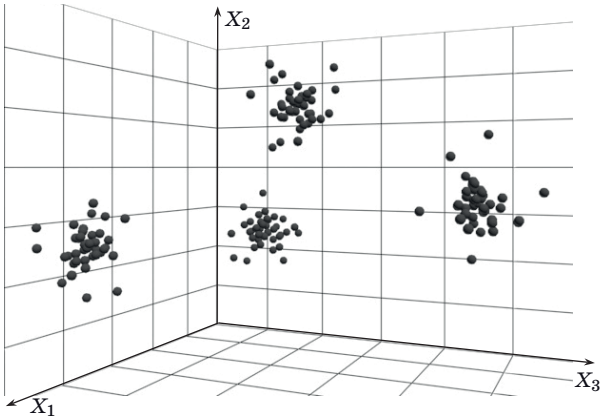
Neural network as a discretization model of the signs space

Classifying neural network uses object data at the training stage $\omega_i, i = \overline{1, n}$, which can be aggregated from different sources, e. g. the Internet, or can be inclusions of a variety of sensors in a process control loop or some technical object. The geometric paradigm of machine learning uses an attributive description of objects of the training sample and their representation by the points in a multidimensional coordinate system. Using conversion of nominal and ordinal variables to a binary type is applied we are providing a numerical representation of qualitative properties.

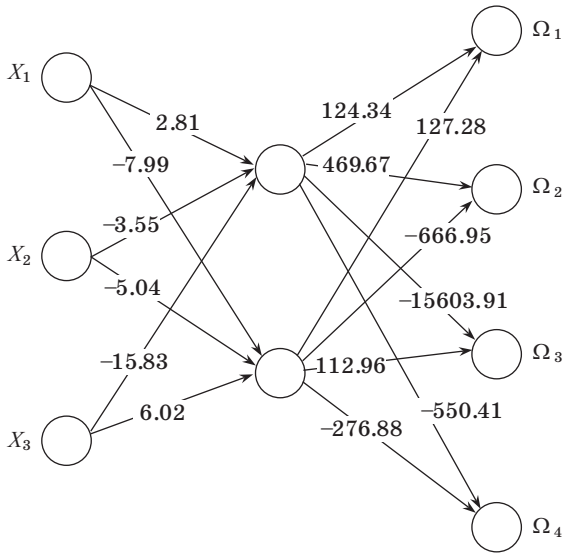
Descriptive signs $\{X_j | j = \overline{1, N}\}$, entered to the input layer of the neural network, characterize the properties of objects of the training sample. The classifying output attribute indicates the belonging of objects ω_i to the one of the class sets Ω_m, $m = \overline{1, M}$. Having an adequate set of signs **X**, it is possible to form an individual space, in which the objects of the training sample are separated by non-intersecting class hulls (Fig. 1).

By the classification process, the neural network transforms a continuous signs space into a discrete set of classes. So, trained on data corresponding to Fig. 1, a three-layer (one input and output layer, one middle layer) neural network transform a combination of the values of three signs into one of four specified classes. The model defined by a set of weighting coefficients shown in Fig. 2. This is uses the activation function likes $f(S) = \dfrac{1}{1 + e^{-S}}$, where S is the signal on the input layer.

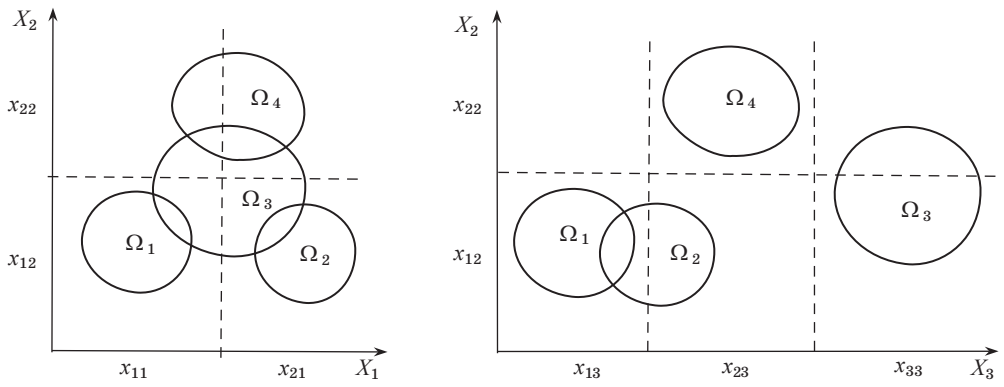
For the clearly interpret the constructed neural network, the information processing should be presented explicitly as connections between combinations of values of N signs X_j and classes Ω_m. Such a view can be attracted using a discrete model of a



■ ***Fig. 1.*** Training set objects in a multidimensional space



■ ***Fig. 2.*** Structure and weight coefficients of the classifying neural network



■ ***Fig. 3.*** Separation of cluster shells in the X_1–X_2 and X_2–X_3 subspaces

multidimensional OLAP (online analytical processing) cube with binary measures (cell values) [13].

The key step in this case, is the quantization of the multidimensional space into the minimum allowable number of cells that preserve the separating power of the original dictionary of signs **X**. Accordingly, for each signs X_j the minimum number of thresholds t_j, is set, at which the distinguishability of classes not violated (Fig. 3).

The number of thresholds t_j is determined by the number of class pairs separable by the X_j signs. If several pairs of classes have a common gap, then one threshold is used.

Method of neural network interpretation

A discrete carrier of knowledge should be built in the form of a binary decision matrix [14] or a multidimensional OLAP cube with binary measures and measurement labels, which are gradations of signs values.

The number of signs gradations and the location of the thresholds are determined in the process of adaptive quantization of the signs space using a genetic algorithm.

The creations of the intervals of changes in the initial signs X_j within the specified classes Ω_m, is performed by independently changing the value of X_j at the input of the neural network. Herewith, we using the set of average values for the remaining sign, when the m-th output neuron is triggered.

If an object of the m-th class has a binary signs (attribute) X_j, or the values of the quantitative signs X_j belong to the interval $(d_{(i-1)\,j},\ d_{ij})$, then the gradations of the signs value x_{ij} for the class Ω_m in the cells of the OLAP cube take single values

$$x_{ij}(m) = \begin{cases} 1,\ \exists\omega\in\Omega_m,\ x_j\in(d_{(i-1)j},\ d_{ij}),\ m=\overline{1,\ M},\ i=\overline{1,\ t_j}; \\ 0,\ \text{otherwise}, \end{cases}$$

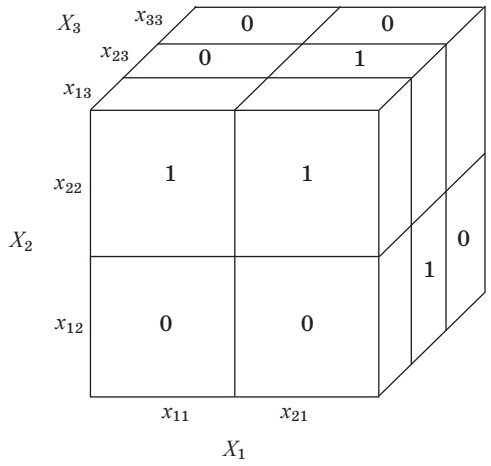
where t_j — the number of gradations of the sign X_j (so-called, the nominal values).

The subcube of the discretized multidimensional space for class Ω_4 is shown in Fig. 4.

In such a discrete classifying space, the values of signs are set in the form of single elements of the OLAP cube and threshold levels. By this way, it is provided an easy semantic interpretation of the decision rule, based on the trained neural network.

Interpreting an OLAP cube with binary measures, based on a system of mathematical production (decision) rules of the form

$$\text{“if } \bigwedge_{j=1}^{N}(x_j\in(d_{(i-1)j},\ d_{ij})_m),\ \text{then } \omega\in\Omega_m\text{”},\ m=\overline{1,\ M},$$



■ ***Fig. 4***. Subcube Ω_4 of a discretized multidimensional space

which use gradations $(d_{(i-1)j},\ d_{ij})_m$ values of signs x_j, $j=\overline{1,\ N}$, for each class Ω_m.

The object signs values points to the cells in the OLAP cube. During the recognition process, occurs element-by-element conjunction (logical AND) of cells, resulting to distinguish the single cell, corresponding to the class code. The space of “own” gradations point out to the found object.

After the coding process in a discretized multidimensional signs space, the images of the classes are rendered using rectangular maps, that use a mesh structure of gradations. On the basis of the such constructed maps (with the gradations sets of signs) we can create a complete system of mathematical production rules.

Genetic model for optimizing discretized feature signs

To describe the discretization algorithm and the choice of the signs space, we use genetic methods concepts, used for the solving common optimization tasks [15–18].

Individual objects in a population represent a discretized multidimensional space $X_1\times X_2\times\ldots\times X_N$ using phenotype — a set of combinations of levels of signs of the working vocabulary $\mathbf{X}_w$, $\mathbf{X}_w=\{X_j\,|\,j=\overline{1,\ N_w}\}$, containing a list of measurable properties of objects.

The match function (so-called, fitness-function) of an individual objects determined by its separating ability — the proportion of combinations of levels of signs, indicating that the object ω belongs to the one of the pairwise disjoint classes Ω_m, $\Omega_m\subset\mathbf{\Omega}$, $\mathbf{\Omega}=\Omega_1\cup\Omega_2\cup\ldots\cup\Omega_M$.

At the level of the heritable structures, information about space is determined by the genotype —

a set of genes of a given individual objects, aggregated in a chromosome series. An individual objects in a population can be represented by a genotype or a single chromosome, when the genotype consists of one chromosome. The coding system for heritable information is a genetic code.

We use a kind of genetic-like algorithm that represents chromosomes using bit strings. Only one gene in a chromosome corresponds to each level of quantization of a signs in a phenotype. A gene is a fixed length bit string containing the value of this level. Thus, a combination of values of all quantization levels for measurable properties of an object is encrypted in the chromosome of an individual.

Improving the quality of the individual's matching function is associated with minimizing the volume of the signs space

$$V(N_w, t_j) = \prod_{j=1}^{N_w} t_j \to \min$$

providing $I(\mathbf{X}_w) = 1$, ensure error-free division of the sample into M classes in the discretized space of the working vocabulary, and natural limits $x_j \in [x_{j\min}, x_{j\max}]$, where $j = \overline{1, N_w}$, $N_w = |\mathbf{X}_w|$. Thus, for choosing the best individual, we should reduce both the number of object signs and the number of their gradations t_j, which makes it possible to increase the extrapolating power of the classifying rule [19].

For these conditions, the length of the chromosome depends on the unknown number of gradations of the signs.

Therefore, the size of the chromosome is fixed by specifying for each signs the minimum number of thresholds, which makes it possible to separate all completely separable classes for which the intervals of change in the values of the signs do not intersect.

Chromosome G consists of two gene groups: $G = \{g_x, g_d\}$.

Gene groups g_x contains single-bit genes bit(x_j), indicating the occurrence of a signs X_{ij} in optimizing space $\mathbf{X}_w$:

$$g_x = \{\text{bit}(x_1), ..., \text{bit}(x_j), ..., \text{bit}(x_N)\}.$$

Gene groups g_d combines genes that in binary format represent quantization threshold values d_j sign X_j, $i = \overline{1, p_j}$, $p_j = t_j - 1$, where t_j — minimum number of sign quantization levels:

$$g_d = \{\text{bin}(d_{11}), ..., \text{bin}(d_{ij}), ..., \text{bin}(d_{pN\,N})\}.$$

Number of bits to represent the threshold gene bit string

$$K_j = \log_2\left(\frac{x_{j\max} - x_{j\min}}{\delta_j} + 1\right),$$

where δ_j — accuracy of representation of sign X_j.

Structure of chromosomal thread **Ch**

$$\underbrace{1001101101}_{N\ positions}\ \underbrace{\underbrace{00101011}_{K_1\ positions}\ldots\underbrace{10100010}_{K_1\ positions}}_{p_1}\ \ldots\ \underbrace{\underbrace{0110}_{K_j\ positions}\ \ldots\ \underbrace{1011}_{K_j\ positions}}_{p_j}\ \ldots\ \underbrace{\underbrace{01100010}_{K_N\ positions}\ldots\underbrace{10001011}_{K_N\ positions}}_{p_N}.$$

The head gene group determines the structure of the signs space, the rest of the genes are responsible for setting the quantization of the multidimensional space, where each gene is responsible for one quantization threshold for a given variable.

The values of the quantization thresholds are determined by the genes of the found individual

$$d_{ij} = \frac{\text{bin}(d_{ij})}{2^{K_j} - 1}(X_{j\max} - X_{j\min}) + X_{j\min}.$$

Therms "individual" means the value of the chromosome vector belonging to the range of permissible values, $Ch \in \mathbf{Ch}_{permissible}$:

$$\mathbf{Ch}_{permissible} = \{Ch | I(\mathbf{X}_w) = I(\mathbf{X})\},$$

where $I(\mathbf{X}_w)$ — separating power of the signs system $\mathbf{X}_w$, $\mathbf{X}_w \subset \mathbf{X}$, which is defined as the number of class pairs completely separable by a given system to the total number of class pairs $M(M-1)/2$.

The work of the genetic algorithm is generally described as follows [16, 18].

1. Initialization. An initial population is randomly generated from N_I binary chromosomes.
2. Computation of the match function and assessment of the fitness of chromosomes in the population.
3. Selection of parents for crossing (performed using a selection operator).
4. Execution of the operator of crossing.
5. Mutation of offspring (descendant) chromosomes.
6. Formation of a new population by selecting the best individuals in a generation.
7. Switch to the next generation of parents and descendants by repeating steps 2–6 until the stop rule is met.

This algorithm implements adaptive quantization of the signs space. The solution of this algorithm is the discrete neural chromosome code. It describes the space of signs of the minimum volume, while maintaining the separating power of the trained neural network.

Individuals of the initial population contain randomly filled threshold genes, they limited by the values of signs, and the chromosome vector Ch belongs to the range of permissible values.

The algorithm has the following parameters:

— size of population of individuals N_I;
— number of pairs selected for reproduction;
— mutation probability P_{mut};
— crossing probability P_{cross}.

For each population, we determine the number of mutating chromosomes, the number of pairs of crossing chromosomes, a given level of convergence of the algorithm ε.

Probability of selection of an individual for reproduction

$$P_i = \frac{f_i}{\sum\limits_{i=1}^{N} f_i},$$

where i — individual number; $f_i = V(N) - V(N_w)$, $i = \overline{1, N_I}$ — individuals match function.

Probability of using the crossing operator $P_{cross} = 0.9...1$. We use some elite individuals in the crossing procedure with quantity $N_e = (1 - P_{cross})N_I$. In conditions when $P_{cross} < 1$ the best individuals of the current population moves into the population without any changes.

In relation to our task, the crossing operator must ensure the process of study of the set of signs for which the head gene group is responsible, and the set of thresholds, encoded by the corresponding genes. Parent-encodings transfer genetic material to new descendant-encodings. To reproduce them, we use a two-parent crossover, which exchanges parts of the bit string of genes at break points.

Reproduction of parent individuals with chromosomes Ch^l and Ch^k looks as follows

$$b^l_{N-1}\ldots b^l_r\ldots b^l_2 b^l_1 b^l_0 \;\; b^l_{K_1-1}\ldots b^l_r\ldots b^l_2 b^l_1 b^l_0 \;\; \ldots \;\; b^l_{K_j-1}\ldots b^l_r\ldots b^l_2 b^l_1 b^l_0 \;\; \ldots \;\; b^l_{K_N-1}\ldots b^l_r\ldots b^l_2 b^l_1 b^l_0$$

$$+$$

$$b^k_{N-1}\ldots b^k_r\ldots b^k_2 b^k_1 b^k_0 \;\; b^k_{K_1-1}\ldots b^k_r\ldots b^k_2 b^k_1 b^k_0 \;\; \ldots \;\; b^k_{K_j-1}\ldots b^k_r\ldots b^k_2 b^k_1 b^k_0 \;\; \ldots \;\; b^k_{K_N-1}\ldots b^k_r\ldots b^k_2 b^k_1 b^k_0$$

$$\downarrow$$

$$b^k_{N-1}\ldots b^k_{p+1} b^l_p\ldots b^l_1 b^l_0 \;\; b^k_{K_1-1}\ldots b^k_{p+1} b^l_p\ldots b^l_1 b^l_0 \;\; \ldots \;\; b^k_{K_j-1}\ldots b^k_{p+1} b^l_p\ldots b^l_1 b^l_0 \;\; \ldots \;\; b^k_{K_N-1}\ldots b^k_{p+1} b^l_p\ldots b^l_1 b^l_0$$

and

$$b^l_{N-1}\ldots b^l_{p+1} b^k_p\ldots b^k_1 b^k_0 \;\; b^l_{K_1-1}\ldots b^l_{p+1} b^k_p\ldots b^k_1 b^k_0 \;\; \ldots \;\; b^l_{K_j-1}\ldots b^l_{p+1} b^k_p\ldots b^k_1 b^k_0 \;\; \ldots \;\; b^l_{K_N-1}\ldots b^l_{p+1} b^k_p\ldots b^k_1 b^k_0,$$

where the p-th bits of genes act as the breaking point, $b_p = \text{random}(0, K-1)$, K — number of gene encoding bits.

When exchanging pieces of parental-encodings, the existing fragments of alleles will be redistributed among the genes of the descendant-encodings while preserving their loci.

To enforce the genetic variability of alleles, we use the mutation operator, which leads to the appearance of new alleles from fragments that were not previously contained in the parental genes.

Chromosomes descendants are exposed with random changes with probability P_{mut} (0.001...0.01). The number of changes made to the chromosome is defined as follows

$$K_{mut} = \beta \cdot \text{random}(1, K),$$

where K — size of the chromosomes, $K = N + \sum_{j=1}^{N} p_j K_j$; β — mutation power coefficient, $\beta \in [0; 1]$.

Mutation stands in inverting the binary sequence, which position in the chromosome determined strongly randomly:

$$b_{r_n} := \left| b_{r_n} - 1 \right|,$$

where $r_n = \text{random}(1, K)$, $n = \overline{1, K_{mut}}$.

During the simulation modeling we configure the power of mutation because this is one of the most important properties of the search algorithm.

The rule (decision) for stopping the genetic algorithm is to achieve a given level of convergence $f_{i\ \max} - f_{i\ \min} < \varepsilon$ — determining such power of match of individuals in the population, at which their further improvement does not occur.

The result of the genetic algorithm computation leads to the choice of an individual from a finite population that has the maximum value of the matching function f_i.

The genetic algorithm makes it possible to find the effective value of the volume of the signs space $V(\mathbf{X}_w)$, for neural network prediction models based on a "black box" type and trained on a samples. This type of space provides us with a prediction for those combinations of values of input signs that were not represented in the training sample, which is usually strongly limited by size.

Visualization and interpretation of classes

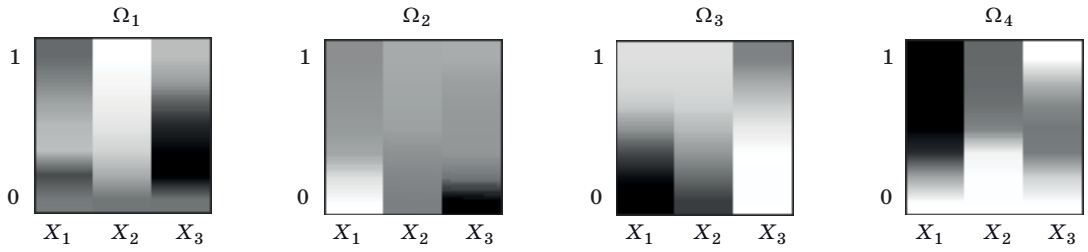
Strictly accurate mapping of characteristic and general signs of object classes is a challenged issue when visualizing solutions in multidimensional continuous spaces [20–27]. It is required to analyze $N_w(N_w - 1)/2$ slices to unambiguously identify a class based on an OLAP cube.

Since the information about the combinations of gradations of the initial features for any class is contained in a compressed form, in a trained discrete knowledge carrier with binary measures, we can use a rectangular map to form a unique image of each class, which use a mesh structure of gradations.

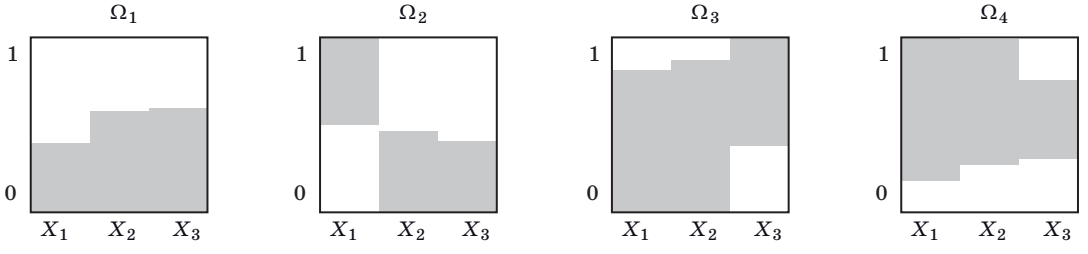
After coding in a discretized multidimensional space of signs, the images of classes reflects the most significant, integral indicators of classes and smooth out the insignificant signs, which observed on image maps, representing the ranges of changes in signs and signals at the input of the output neuron with varying signs.

The class image for each output neuron of the trained network can be mapped in grayscale (Fig. 5) or in 3D. We used the values of linear combinations of inputs coming to the output neurons and the values of the corresponding activation functions. This mappings introduce the proportion of the training sample, objects belonging to the given m-th class (also known as estimation of the "conditional probability" of the class), in which the j-th characteristic lands into the i-th interval.

We use a bar chart (Fig. 6) to assess the interval of changes in a signs within the considered m-th class. The columns formed by independently vary-



■ ***Fig. 5.*** Class images representing signals at the input of the output neuron when signs vary



■ ***Fig. 6.*** Class images representing ranges of signs variation in a normalized space

ing the value of each initial signs at the input of the multilayer neural network (with the set average values of the remaining signs), when the m-th output neuron is triggered. Input indicators (showings) are normalized linearly to the interval 0...1.

Variation ranges of signs at the input of a trained neural network, at which triggered a neuron of the class Ω_1 is: $X_1 = 0...0.4$, $X_2 = 0...0.57$, $X_3 = 0.0...0.59$.

Triggered a neuron of the class Ω_2 is: $X_1 = 0.50...1.0$, $X_2 = 0...0.47$, $X_3 = 0.0...0.43$.

Triggered a neuron of the class Ω_3 is: $X_1 = 0...0.86$, $X_2 = 0...0.89$, $X_3 = 0.63...1.0$.

Triggered a neuron of the class Ω_4 is: $X_1 = 0.17...1.0$, $X_2 = 0.30...1.0$, $X_3 = 0.31...0.77$.

As it was disclaimed early, after the coding process, we get the images of the classes that rendered using rectangular maps with a mesh structure of gradations. Note, that as we says early, this algorithms use a discretized multidimensional signs space. The maximum number of gradations T set to according to the most featured (discrete) sign (Fig. 7). We set "free" gradations, if the signs values in the class correspond to the highest gradation — that's need for the maximum conformity of the images and a bar chart with continuous ranges of signs values.

With the using of the cognitive images, we can clearly determine the classes that have the minimum and maximum values of integral indicators (showings) — the sum of gradations for all binarized signs $Sg(\Omega_m)$ and the spread of signs values $R(\Omega_m)$

$$\Omega'_m = \arg \operatorname*{extr}_{\Omega_m \in \mathbf{\Omega}} Sg(\Omega_m);$$

$$\Omega''_m = \arg \operatorname*{extr}_{\Omega_m \in \mathbf{\Omega}} R(\Omega_m).$$

Small signs values have a class Ω_1, $Sg(\Omega_1) = 1 \cdot (1 + 1 + 1) = 3$. Classes with the highest characteristic values follows Ω_3 и Ω_4: $Sg(\Omega_3) = 1 \cdot (1 + 1) + 2 \cdot (1 + 1) + 3 \cdot (1 + 1 + 1) = 15$, $Sg(\Omega_4) = 1 \cdot 1 + 2 \cdot (1 + 1 + 1) + 3 \cdot (1 + 1) = 14$. Class Ω_1 has the smallest spread of signs values $R(\Omega_1) = \sqrt{1 \cdot 1 \cdot 1} = 1$. Class with the highest spread of values Ω_3, $R(\Omega_3) = \sqrt{3 \cdot 3 \cdot 1} = 3$.

Using the convolution of the constructed images (of classes) for the set gradations of signs, we can produce a complete system of mathematical production rules as follows:

"if $(X_1 \in X_{12})$ and $(X_2 \in X_{21})$ and $(X_3 \in X_{32})$, then $\omega \in \Omega_2$".

Thus, by varying the values of the descriptive variables at the input of the trained neural network, we used the genetic algorithm to extract a discrete carrier of knowledge. This makes it possible to clearly interpret the classes using cognitive maps and produce a full system of mathematical production rules.

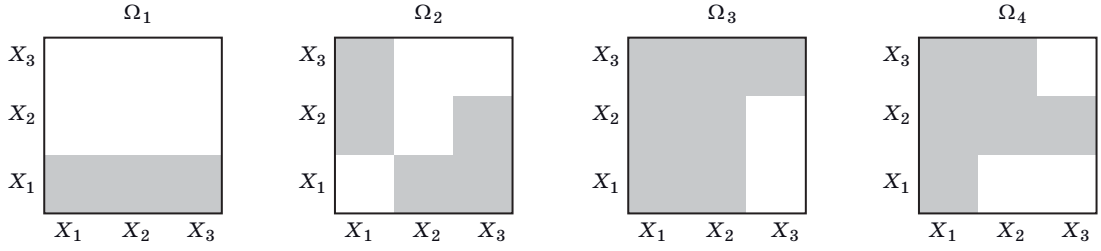
Conclusion

As it was noted before, the complex challenge of verbalizing the output of deep learning and clearly clarification of the obtained result (i. e. why the model made those or another decisions) related to the using of the common "black box" model — by the learning process, the "knowledge" organized in form of set of the weight coefficients of the links between neurons.

Neural network converts a continuous feature space into a discrete set of classes by the process of classification. For the interpretation of the trained neural network decision, the data can be represented in an obvious form as mappings between combinations of values N of signs of X_j and the classes Ω_m, using discrete model of a multidimensional OLAP cube with binary measures.

The discrete knowledge model is formed by the process of adaptive quantization of a signs space using a common genetic algorithm. Individual's chromosome encrypts a set of values of all quantization levels for measurable properties of an object. The head gene group define the structure of the signs space, the remaining genes responsible for configuring the quantization of the multidimensional space, where each gene in charge for one quantization threshold of a given variable.

The genetic algorithm makes it possible to find the effective value of the volume of the signs space $V(\mathbf{X}_w)$, for neural network prediction models based on a "black box" type and trained on a samples. This type of space provides us with a prediction for those



■ ***Fig. 7.*** Images of classes after encoding in a discretized multidimensional signs space

combinations of values of input signs that were not represented in the training sample, which is usually strongly limited by size.

Using the proposed discrete model we can form a unique images of each class based on rectangular maps with cellular structure of gradations. Maps reflect the most significant, integral indicators (showings) of classes, which strongly determine the location and size of a class in multivariate space.

Thus, we can form a complete set of mathematical production decision rules, both in the process of directly interpreting a discrete model of a multidimensional OLAP cube, and on the convolution of class images for signs gradations.

References

1. Martin Prause, Jurgen Weigand. Market model benchmark suite for machine learning techniques. *IEEE Computational Intelligence Magazine*, 2018, vol. 13, iss. 4, pp. 14–24. doi:10.1109/MCI.2018.2866726
2. Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, Mohsen Guizani. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 2018, vol. 20, iss. 4, pp. 2923–2960. doi:10.1109/COMST.2018.2844341
3. Sung-Yu Tsai, Jen-Yuan Changc. Parametric study and design of deep learning on leveling system for smart manufacturing. *IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*, February 8–9, 2018, pp. 48–52. doi:10.1109/SMILE.2018.8353980
4. Abdelrahman M. Shaker, Manal Tantawi, Howida A. Shedeed, Mohamed F. Tolba. Generalization of convolutional neural networks for ECG classification using generative adversarial networks. *IEEE Access*, 2020, vol. 8, pp. 35592–35605. doi:10.1109/ACCESS.2020.2974712
5. Gusev A. V. Prospects for neural networks and deep machine learning in creating health solutions. *Information Technologies for the Physician*, 2017, no. 3, pp. 92–105 (In Russian).
6. Sozykin A. V. An overview of methods for deep learning in neural networks. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*, 2017, vol. 6, no. 3. pp. 28–59 (In Russian). doi:10.14529/cmse170303
7. Guangxin Lou, Hongzhen Shi. Face image recognition based on convolutional neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, vol. 31, iss. 1, pp. 117–124. doi:10.23919/JCC.2020.02.010
8. Rex Martinez. Artificial intelligence: Distinguishing between types & definitions. *Nevada Law Journal*, 2019, vol. 19:3, pp. 1015–1042.
9. Lukyanova O. A., Nikitin O. Y. Selfish general intelligence. *Cloud of Science*, 2019, vol. 6, no. 3. Available at: http://cloudofscience.ru (accessed 10 May 2020) (In Russian).
10. Weiming Xiang, Hoang-Dung Tran, Taylor T. Johnson. Output reachable set estimation and verification for multilayer neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, vol. 25, iss. 11, pp. 5777–5783. doi:10.1109/TNNLS.2018.2808470
11. Na Li, Ziqiang Zheng, Shaoyong Zhang, Zhibin Yu, Haiyong Zheng, Bing Zheng. The Synthesis of unpaired underwater images using a multistyle generative adversarial network. *IEEE Access*, 2018, vol. 6, pp. 54241–54257. doi:10.1109/ACCESS.2018.2870854
12. Mu Zhou, Yixin Lin, Nan Zhao, Qing Jiang, Xiaolong Yang, Zengshan Tian. Indoor WLAN intelligent target intrusion sensing using ray-aided generative adversarial network. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020, vol. 4, iss. 1, pp. 61–73. doi:10.1109/TETCI.2019.2892748
13. Davardoost F., Babazadeh Sangar A., Majidzadeh K. Extracting OLAP cubes from document-oriented NoSQL database based on parallel similarity algorithms. *Canadian Journal of Electrical and Computer Engineering*, 2020, vol. 43, no. 2, pp. 111–118. doi:10.1109/CJECE.2019.2953049
14. Pimenov V. I., Voronov M. V., Pimenov I. V. The cognitive visualization of classifying rules extracted from data based on binary solver matrix model. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 6, pp. 2–11 (In Russian). doi:10.31799/1684-8853-2019-6-2-11
15. Andras Takacs, Manuel Toledano-Ayala, Aurelio Dominguez-Gonzalez, Alberto Pastrana-Palma, Dimas Talavera Velazquez, Juan Manuel Ramos, Edgar Alejandro Rivas-Araiza. Descriptor generation and optimization for a specific outdoor environment. *IEEE Access*, 2020, vol. 8, pp. 2169–3536. doi:10.1109/ACCESS.2020.2975474
16. Abinet Tesfaye Eseye, Matti Lehtonen, Toni Tukia, Semen Uimonen, R. John Millar. Machine learning based integrated feature selection approach for improved electricity demand forecasting in decentralized energy systems. *IEEE Access*, 2019, vol. 7, pp. 91463–91475. doi:10.1109/ACCESS.2019.2924685
17. Hossam M. J. Mustafa, Masri Ayob, Mohd Zakree Ahmad Nazri, Graham Kendall. An improved adaptive memetic differential evolution optimization algorithms for data clustering problems. *PLOS ONE*, 2019, May 28, pp. 1–28. doi:10.1371/journal.pone.0216906
18. Ryadchikov I. V., Gusev A. A., Sechenev S. I., Nikulchev E. V. Genetic algorithm for search PID-controllers parameters of a walking robot stabilization. *Transactions of NNSTU n.a. R. E. Alekseev*, 2019, no. 1(124), pp. 58–65 (In Russian).
19. Jochen L. Cremer, Ioannis Konstantelos, Goran Strbac. From optimization-based machine learning to interpretable security rules for operation. *IEEE*

Transactions on Power Systems, 2019, vol. 34, iss. 5, pp. 3826–3836. doi:10.1109/TPWRS.2019.2911598
20. Yunhai Wang, Kang Feng, Xiaowei Chu, Jian Zhang, Chi-Wing Fu, Michael Sedlmair, Xiaohui Yu, Baoquan Chen. A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2018, vol. 24, iss. 5, pp. 1828–1840. doi:10.1109/TVCG.2017.2701829
21. Yunhai Wang, Xin Chen, Tong Ge, Chen Bao, Michael Sedlmair, Chi-Wing Fu, Oliver Deussen, Baoquan Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 2019, vol. 25, iss. 1, pp. 820–829. doi:10.1109/TVCG.2018.2864912
22. Ruizhen Hu, Tingkai Sha, Oliver Van Kaick, Oliver Deussen, Hui Huang. Data sampling in multi-view and multi-class scatterplots via set cover optimization. *IEEE Transactions on Visualization and Computer Graphics*, 2020, vol. 26, iss. 1, pp. 739–748. doi:10.1109/TVCG.2019.2934799
23. Zhe Wang, Nivan Ferreira, Youhao Wei, Aarthy Sankari Bhaskar, Carlos Scheidegger. Gaussian cubes: real-time modeling for visual exploration of large multidimensional datasets. *IEEE Transactions on Visualization and Computer Graphics*, 2017, vol. 23, iss. 1, pp. 681–690. doi:10.1109/TVCG.2016.2598694
24. Min Lu, Shuaiqi Wang, Joel Lanir, Noa Fish, Yang Yue, Daniel Cohen-Or, Hui Huang. Winglets: visualizing association with uncertainty in multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 2020, vol. 26, iss. 1, pp. 770–779. doi:10.1109/TVCG.2019.2934811
25. Ying Zhao, Feng Luo, Minghui Chen, Yingchao Wang, Jiazhi Xia, Fangfang Zhou, Yunhai Wang, Yi Chen, Wei Chen. Evaluating multi-dimensional visualizations for understanding fuzzy clusters. *IEEE Transactions on Visualization and Computer Graphics*, 2019, vol. 25, iss. 1, pp. 12–21. doi:10.1109/TVCG.2018.2865020
26. Lazutin O. G. Technique of communicating information about the technical state of space vehicles using data compression algorithms and cognitive graphical representation. *Proceedings of the Mozhaisky Military Space Academy*, 2016, vol. 650, pp. 11–17 (In Russian).
27. Emelyanova Ju. G., Fralenko V. P. Methods of cognitive-graphical representation of information for effective monitoring of complex technical systems. *Program Systems: Theory and Applications*, 2018, vol. 9, no. 4(39), pp. 117–158 (In Russian). doi:https://doi.org/10.25209/2079-3316-2018-9-4-117-158

УДК 004.89
doi:10.31799/1684-8853-2020-6-12-20

Интерпретация обученной нейронной сети на основе генетических алгоритмов

В. И. Пименов[а], доктор техн. наук, профессор, orcid.org/0000-0002-7228-3009, v_pim@mail.ru
И. В. Пименов[б], канд. техн. наук, доцент, orcid.org/0000-0002-1954-6463
[а]Санкт-Петербургский государственный университет промышленных технологий и дизайна, Б. Морская ул., 18, Санкт-Петербург, 191186, РФ
[б]Государственный университет морского и речного флота им. адмирала С. О. Макарова, Двинская ул., 5/7, Санкт-Петербург, 198035, РФ

Введение: стратегия развития искусственного интеллекта предполагает применение алгоритмов глубокого машинного обучения для решения задач различного класса. Обученные на конкретных наборах данных нейросетевые модели трудно интерпретировать, что связано с подходом «черного ящика», когда знания формируются как набор весовых коэффициентов связей между нейронами. **Цель:** разработка дискретной модели знаний, представляющей в явной форме закономерности обработки информации, закодированные связями между нейронами. **Методы:** адаптивное квантование признакового пространства с помощью генетического алгоритма и построение дискретной модели многомерного OLAP-куба с бинарными мерами. **Результаты:** генетический алгоритм выполняет извлечение из обученной нейронной сети дискретного носителя знаний. В хромосоме особи зашифровывается комбинация значений всех уровней квантования для измеримых свойств объекта. Головная генная группа определяет структуру признакового пространства, остальные гены отвечают за настройку квантования многомерного пространства, где каждый ген отвечает за один порог квантования заданной переменной. Дискретная модель многомерного OLAP-куба с бинарными мерами представляет в явной форме связи между комбинациями значений признаков объектов и классами. **Практическая значимость:** для нейросетевых моделей предсказания, построенных по обучающей выборке, генетический алгоритм дает возможность найти эффективное значение объема пространства признаков для тех комбинаций значений входных признаков, которые не были представлены в обучающей выборке, обычно ограниченной в объеме. С помощью предложенной дискретной модели формируются уникальные образы каждого класса на основе прямоугольных карт, в которых используется ячеистая структура градаций. Карты отражают наиболее существенные, интегральные показатели классов, которые определяют местоположение и размер класса в многомерном пространстве. На основе свертки построенных образов классов для установленных градаций признаков записывается полная система продукционных решающих правил.

Ключевые слова — классификация, глубокое машинное обучение, нейронная сеть, генетический алгоритм, многомерный OLAP-куб, решающее правило, семантическая интерпретация, визуализация классов.

Для цитирования: Pimenov V. I., Pimenov I. V. Interpretation of a trained neural network based on genetic algorithms. *Информационно-управляющие системы*, 2020, № 6, с. 12–20. doi:10.31799/1684-8853-2020-6-12-20
For citation: Pimenov V. I., Pimenov I. V. Interpretation of a trained neural network based on genetic algorithms. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 6, pp. 12–20. doi:10.31799/1684-8853-2020-6-12-20