

# Multiple-Valued and Branching Neural Networks

Alexander Makarenko<sup>1</sup>

*Institute of Applied Systems Analysis NTUU "KPI"*

Kyiv, Ukraine

[makalex51@gmail.com](mailto:makalex51@gmail.com)

## Abstract

The generalizations of common single-valued artificial neural networks are proposed. The main proposition is to consider multiple-valued networks. Such neural networks can have multiple values of elements and multivalued connections between elements at given time moment. Modified Hopfield neural networks with strong anticipation property are considered as the examples. New aspects of learning processes considered for the case of multiple-valuedness. Branching networks are described. Presumable applications of multivalued neural networks are proposed. Also some new research problems are described.

**Keywords**—neuronets, multiple-valued, strong anticipation, Hopfield modification, generalizations, branching networks, applications

## I. INTRODUCTION.

For the establishing of proposed here new problem in the field of artificial neural network it is necessary to do some analysis of basic concepts of artificial neural network. Here in the introduction we first of all give the analyses of the elements of general construction of artificial neural networks and highlight nine issues concerned the principles of operating of neural network. The analysis of such issues is illustrated on the example of Hopfield network. Then after such analysis it is proposed one of the possible ways of generalization for neural networks – namely the multiple-valuedness of networks.

### A. Steps for building and using classic neural networks

Classic artificial neural networks (ANN) have now become one of the most common tools for forecasting, classifying and modeling [1-5]. As one of the standard applications of ANN, you can specify artificial intelligence and machine learning.

At the same time, the theory and practice of using classical neural networks have gone a long way of development, starting from approximately the 50s of the last century (Mc Culloch W.&W. Pitts, F. Rosenblatt, M. Minsky, J. Hopfield, and many others). In general, it is very approximately possible to propose an ANN application scheme consisting of such steps:

- 1) the choice of architecture
- 2) the choice of  $S$  spaces for the set of states of an individual element (neuron)
- 3) the choice of sets (or spaces) for connections between elements
- 4) the choice of the set for the evolutionary variable (equivalent to time)
- 5) the choice of an adequate dynamic law for the evolution of ANN.
- 6) the formal derivation of models and their mathematical research.
- 7) methods, algorithms and approaches for determining relationships and nonlinear activation functions

- 8) interpretation of ANN solutions, especially in applications
- 9) software and hardware implementations

Initially, point 5 was implemented empirically, by analogy of behavior, or from the desire that the behavior of ANN should be in accordance with the principles of the subject area where ANN is applied.

Further, usually point 6 helps to build ANN and better understand the behavior of ANN. In some cases, the ANN model itself can be derived from a rigorous optimization problem.

### B. The Hopfield model as an example of the classic ANN

As a basic example of the implementation of paragraphs 1) -9) for ANN, we present the classical Hopfield model. Here:

- 1) Architecture: the Hopfield model is a fully connected network (in principle, all elements may be connected to all);
- 2)  $S = \{0,1\}$ ;

3)  $J = \{J_{ij}\}$  is the matrix of connections (bonds), where  $J_{ij}$  is the connection of the element  $i$  with the element  $j$  ;  
 $J_{ij} \in R^1$ ;

- 4) Hopfield models most often use discrete time  $t = 0, \tau, 2\tau, \dots, n\tau, \dots$ ,  $\tau$  - time step;

The formulas often use the designation of indices of discrete steps:  $0,1,2,3,\dots$ ;

- 5) the symplest type of dynamics is:

$$s_i(n+1) = f\left(\sum_{i \neq j} J_{ij} s_j(n); \theta\right) \quad , (1)$$

where  $\theta$  - parameters,  $f$  - sigmoidal function,  $s_i(n)$  . – the state of  $i^{th}$  element in moment  $n$  ;

6) The dynamics can be derived from minimization of the “energy” functional  $E = -\sum J_{ij} s_i s_j$ .

Of course, many modifications of Hopfield had been considered: hierarchical, vector, with delay, etc. when  $S, J, f$  have evident extending, with using real values. For example, we may use  $S \in R^1$  or  $S \in R^1 \times R^1 \times \dots \times R^1$ .

### C. Some generalizations of classical ANN

However, more substantial generalizations of the ANN had been developed in parallel to classical case. So, we can mention complex-valued ANN, ( $S \in C^1, J_{ij} \in C^1$ , where

$C^1$  ( $C^1$  is the space of complex numbers, (I. Aizenberd, Kak S., Chisley R., Miranker, M. Perus, Nitta T.); octonion ANN (T. Isokawa, Y. Kuroe, H. Iima); quaternion ANN (C-A. Popa, M. Kobayashi); p-adic ANN (A. Khrennikov, Tirozzi B.); ANN with Grossons (Ya. Sergeev)).

Step 6) Includes methods, algorithms and approaches to defining  $\{J_{ij}\}$ , functions  $f$ , parameters  $\theta$  for achieving ANN solution of the formulated problems. Often such problems are called learning rules (with teacher, without teacher, reinforce learning, training in spike networks and training in dendritic networks).

Note that step 7) includes (when it is possible to investigate) a theoretical and practical assessment of the capacity of a certain class of networks.

Also it is very important to use ANN in quantum mechanics and, most importantly, the use of the ideas of quantum mechanics in ANN in connection with quantum computing and parallelization of computational processes (D. Deutch, M. Perus, Chisley R. and many others).

However, certain systems, processes lead to the need for further theoretical expansion of the possible construction of ANN, which appeared explicitly, as it seems, in the works of the author. Namely, we are talking about the ANN schemes, in which the input parameters, variables, and the bonds can simultaneously have MANY values (multiple-valued), unlike the previously known ANN schemes, when variables can, for example, take a single value not from  $S = \{0,1\}$ , but, for example, the set  $m$  of the values  $m > 1$  from  $S_k = \{0,1,2,\dots,K\}$  at given time moment.

In this article, we further develop a description of such multiple-valued ANN. The main goals of this paper are to describe existing multiple-valued generalization of neural networks and to pose new research problems in developing and investigation of such objects. In section II we pose the description of basic construction of proposed generalization. Also we give the description of some aspects of learning processes for such networks. The problems of interpretation of the calculation results on with multiple-valued networks are posed in the section III. One important example of ANN with multiple-valuedness – namely ANN with strong anticipation is described in section IV. Some propositions for further

research in the field of multiple-valued neural networks are describe in the section V of the paper.

## II. STATEMENTS OF RESEARCH TASKS FOR THE CASE OF MULTIPLE-VALUED ANN.

Here in this section we describe some aspects of realization of ideas of accounting multiple-valuedness in artificial neural networks. The multiple-valuedness of neuron state, bounds and propositions for presumable learning of networks are considered in subsections below.

### A. General considerations for possible multi-valued generalizations of the ANN

Here we will discuss the new possibilities of expanding the scheme for considering steps 1) -9) from classical ANN neural networks into the case of neural networks with multiple-valued solutions (MANN). Specific examples of such neural networks will be given in section 3, where ANNs are described with the strong anticipation property.

Architecture (steps 1) -2)). Suppose that ANN elements can be multiple-valued, i.e. let us assume that each element  $i$  from the set  $N$  of network elements can have several values from the set  $\Omega_i$  at the same time moment. As the simplest case, we can consider only two branches for all of the elements  $|\Omega_i| = 2$ , where  $||$  is the power of the set. But in principle there may be case when  $|\Omega_i| = +\infty$ . For simplicity, we can at first take the case  $\Omega_i = \Omega, \forall i$ .

However, in principle, possible an heterogeneous case, when  $\Omega_i \neq \Omega_j$ . In the simple examples it is natural to consider  $\Omega$  as a subset  $R^1$  (real Euclidean space). However, in the future one can also study more complex cases, for example  $\Omega$ , consisting of a set of sets  $\Omega = \{A, B, \dots, G\}$ , where  $A, B, \dots, G$  are sets in a certain space (for example, in Hausdorff space). In the following, we confine ourselves to the presentation of case  $\Omega = (\omega_1, \omega_2, \dots, \omega_K)$  with  $\omega_k \in R^1, k = 1, 2, \dots, K$  unless otherwise specified.

3) Bonds. Suppose that each of the presumable multiple states of one element can be associated with each of the multiple states of other element. That is, we have a set of bonds  $\{J_{ij}^{k_i k_j}\}$  between elements  $i$  and  $j$  which accounting states  $k_i$  and  $k_j, k_i, k_j = 1, 2, \dots, K$  of elements.

For fixed  $i$  and  $j$  then we have a matrix of relations between these elements (in this case, the size  $K \times K$ ). An example of connections in real systems can be connections between two individuals in society: family, jobs, sympathies, etc. relations.

4) Time As a parameter describing the evolution of the network, for simplicity, we take a discrete set  $t = 0, 1, \dots, n, \dots$ . For the Hopfield network,  $t$  is usually the time, but it is possible in physics to find models where one of

the spatial variables acts as evolution variable. Also the case of continuous time may be considered just as for single-valued ANN.

5) Dynamic laws. The choice of a dynamic law strongly depends on the choice of variable values in points 1) - 4). Even in the case  $\Omega \in R^1 \times R^1 \times \dots \times R^1$  there are already a large number of possible options. For the classical Hopfield network  $f$  is a usual function,  $f : R^1 \rightarrow R^1$ , which is called the activation function and is a function from the class of sigmoidal functions [1]. In the multiple-valued case, a function  $f$  can also, in general, be a multivalued non-linear function of a multivalued argument:

$$f^M : \left\{ \begin{matrix} \omega_1^i(n) \\ \omega_2^i(n) \\ \dots \\ \omega_K^i(n) \end{matrix} \right\} \Rightarrow \left\{ \begin{matrix} \omega_1^i(n+1) \\ \omega_2^i(n+1) \\ \dots \\ \omega_K^i(n+1) \end{matrix} \right\}. \quad (2)$$

That is, the function  $f$  in such case generates multiple-valued states of element at the time moment  $(n+1)$ , depending on all the values of all elements at the moment  $(n)$ . But even more interesting variants are possible, when a multi-valued function depends on the past and the future, i.e.

$$f^M : (\{\varpi(n-l)\}, \{\varpi(n-l+1)\}, \dots, \{\varpi(n)\}, \{\varpi(n+1)\}, \{\varpi(n+2)\}, \dots, \{\varpi(n+m)\}) \Rightarrow (\{\varpi(n+1)\}) \quad (3)$$

where  $\{\varpi(n)\} = \{\omega_1(n)\}, \{\omega_2(n)\}, \dots, \{\omega_K(n)\}$ .

In equation (3), both memory effects (where  $l$  is the depth of delay) and advancing (where  $m$  is the horizon of foresight) are taken into account. Note that it can have a complex shape (for example, an integral dependence), and the effects of delay and lead can be infiniti ( $l = +\infty, m = +\infty$ ). But even the simplest case ( $l = 1, m = 1$ ) or even ( $l = 0, m = 1$ ) leads to completely new properties of such ANN. The fact is that then the nonlinear equation:

$$f : (\{\varpi(n)\}) \Rightarrow (\{\varpi(n+1)\}). \quad (4)$$

in the simplest cases, takes the form:

$$\varpi_i(n+1) = f(\{\varpi_i(n)\}, \{\varpi_i(n+a)\}, \{J_{ij}^{k_l k_j}\}). \quad (5)$$

This equation is generally non-linear and can have many solutions, one solution or no solution for the state of an individual element. Then, complementing the set of values  $\Omega$

with the empty set  $\emptyset$ , we call any sequence consisting of values chosen uniquely from all possible states at specified times by the branch of the MANN solution. The sequence of transitions by the branch of solution consisting of the values selected in a unified way from all possible states at specified points in time is named the MANN branch of the transitions:

$$\omega_i^{p(1)}(1), \omega_{j(i)}^{p(2)}(2), \dots, \omega_{j(n-1)}^{p(n)}, \dots, \omega_{j(n+m-1)}^{p(n+m)}(n+m), \quad (6)$$

where it is assumed that the values  $\omega_i^{(1)}(1)$  go into value  $\omega_{j(i)}^{(2)}(2)$ , etc. We assume that the transition from the empty value of  $\emptyset$  is only possible to the empty value of  $\emptyset$ .

In case of reaching the empty value of  $\emptyset$ , we say that the branch is terminated.

At this stage of the research, it has so far been possible to investigate the scheme of equations (3) - (5) only in the simplest case, when is  $f$  the piecewise linear.

6) Formalism. So far, mostly single-valued ANN cases have been investigated. Examples of MANN were chosen empirically, following the samples of classical ANN and those that under certain conditions go into the usual unambiguous ANN networks. However, many classical ANNs (for example, Hopfield) and their classical generalization can be obtained as solutions of minimization (i.e., as optimization problems). We can expect that this way of formalization and derivation of neural networks is basic. Application of general minimization can also be applied in the case of MANN. However, it is clear that in this case it will be necessary to apply the modern apparatus of the calculus of variations and optimization for multivalued functional (see the discussion in [6]).

### B. Training and identification of multiple-valued networks

Here we describe other important features of MANN, which are closely related to their mathematical interpretation, and at the same time to the interpretation of the behavior of solutions for real problems.

The first important new formulation is the problem of learning of MANN. For previously used, primarily "classical" networks - Hopfield models, multilayer perceptron, etc., the learning process (with teacher, without teacher, reinforce learning, etc.) is important. According to certain basic examples, the certain procedures (algorithms) exist of computing matrix of connections  $\{J_{ij}\}$  between elements  $i$  and  $j$  of neural networks. Examples are the back-propagation process for a multilayer perceptron and the Hebb rule for the Hopfield model [1-6], and many others. Note, that for classical ANN, these algorithms (learning) are proved by mathematical reasoning. At the same time, even to generalize the "classical" ANN, the rules of learning are chosen mostly empirically (without evidence), but by analogy with the structure of known rules (for example, Hebb rules) and by analogy with the criterion of the correctness of the choice from the ANN prototype.

However, for MANN, the formulation of possible learning schemes is a difficult task (just in empirical approach, and moreover, in terms of mathematical justification in the general case). And the complexity here arises in connection with the possible multiple-valuedness of both the MANN states and the possible multiple-valuedness of connections between the elements.

As with most classical and non-classical methods of identification, it is assumed that there is an original process, a system, which generates certain data sets. It is assumed that for an approximate description of the behavior of such systems (the generation of approximations to exact, in a certain sense, data), a model is used that provides such approximations. Then the most common task of identification is to choose a class of models for approximation, then choose a narrower subclass of models from a general class and then select the parameters of such sub-models, which ensures the convergence between real data sets and generated approximation models of solutions (data sets). We restrict ourselves to subclass of models with discrete time, namely artificial neural networks. Then the general problem of identification for such objects (MANN) is as follows:

Identification task for MANN. Let there be some measure  $E(U, V)$  of the deviation of the approximated solution (or approximation data)  $V$  from the exact solution (data set of the initial system  $U$ ). For multivalued neural networks (MANN) it depends on many components: the rules for changing the state of the elements, the state space of the bonds and many others. Then the task of identification of MANN is to determine the parameters of the MANN (or part of the parameters) at which the minimum deviation  $E(U, V)$  is achieved on a certain subset of the initial data.

The formulation of the identification problem in the proposed form allows for a very large scope for detailing of specific tasks and used neural networks. So, the measure of deviation  $E(U, V)$  can take a different form depending on the task and purpose. In the simplest case  $E(U, V)$  may have norms  $L_2$  or  $C$ . However, even in this case  $E(U, V)$  may depend on the values only at the final moment of time, or on all the values of the elements on the trajectories at all moments in time. So the dimension of the space of adjustable (identifiable) parameters can be relatively large even for relatively simple neural networks. We also note that for neural networks with strong anticipation, we discovered the possibility of many-valued solutions, which in principle can significantly increase the dimension of the parameter space. As already indicated the idea of identification has already been encountered in various statements. So, one of the methods was the use of genetic algorithms. Another approach is the use of gradient methods. In principle the same schemes may be used also in multiple-valued case. But it is the task for future investigation because it needs the adaptation and development of multivalued analysis and optimizations.

There is still a lot of research to be done on this issue. However, it is already possible to express some thoughts about the formulation of problems, which, it seems, can help in further research.

Option 1. Both elements and connections are multiple-valued. Multiplicity can have a different character, for example, just infinite sets for values (a more complex variant). In addition, the number of possible states may depend on time. More simple is Option 1a.

Option 1a. Let the number of simultaneously possible states for each of the  $N$  elements be fixed (for example  $K = const, K > 1$ ), and let the number of connections between the elements  $i$  and  $j$  be also less than or equal to that of  $K^2$ . It is interesting that, even for a case  $K = 2$ , the differences from the classical case begin to appear. (In classical case ( $K = 1$ ) there is only one state for the element for the connection ( $J_{ij}$ ) between the elements  $i$  and  $j$ ).

In the case  $K = 2$  (let's call this as the "minimum" MANN), the elements  $i$  and  $j$  have 2 possible states and 4 possible connections ( $J_{ij}$ ).

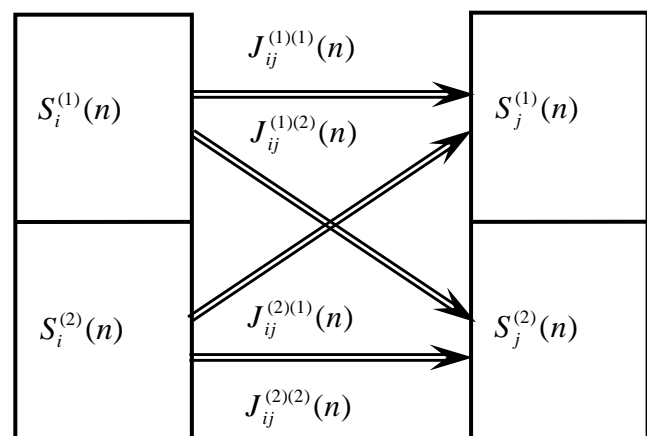


Fig. 1 Relationship of two two-state elements at moment  $n$ .

Then, when training, it is necessary to train 4 connections between elements  $i$  and  $j$ . If the branches (or trajectories) described above did not intersect (did not interact), then formally it would be possible to double the number of elements (for the case  $K = 2$ ; the case of large, but finite  $K$  ones could be interpreted in a similar way). and if separate states from two different possible multiple - states did not interact with states of another type, (for example,  $S_i^{(1)}$  and  $S_j^{(2)}$ ) then formally, we could assume  $J_{ij}^{(1)(2)} \equiv 0$  for this case and we would get two networks that are not connected with each other, each of which could be trained in its "classic" Hebb rule.



In the case of the connection of such sub-states, the task of learning becomes much more complex. Interestingly, in this case, possible analogies with the convolution neural network are seen.

But the importance of this complex option (multi-valued states and multi-valued connections) is still a consequence of the architecture for ANN. The point is in the storage capacity of information in ANN in general, and even more so in the MANN (which can be shown on the example of the Hopfield model and the hypothetical MANN, which is its generalization). It is recognized that in the Hopfield ANN information about the training images is stored in the connections between the elements [1-3]. Sometimes this property is called an analogue of holography.

Moreover, for the Hopfield model, there are estimates of the storage  $N$  capacity of training images, for example,  $M_{save} < 0.14N$  where  $N$  is the number of network elements,  $M_{save}$  the number of recognizable different images

by the network or  $M_{save} < \sqrt{N}$  in other works.

Even in a simple case of a network with possible division ( $K = 2$ ) into two separate subnets, capacity has an estimate:

$M_{save} < 0.14N \times 2$ , which is already a good result.

With state correlation (and branches) due to a much larger number of links  $\{J_{ij}^{(1)(2)}\}$  an even greater increase in capacity can be expected with the same number of elements. This can probably be very important in possible implementation of such networks as technical objects or as objects implemented as programs on computers.

A simpler option is when only one single connection is allowed (the same for all states) between the elements  $i$  and  $j$ , i.e. when  $K = 2$ ,  $J_{ij}^{(1)(1)} = J_{ij}^{(1)(2)} = J_{ij}^{(2)(1)} = J_{ij}^{(2)(2)}$ .

Such a case is still very interesting, since it allows us to study and use the case of multiple-valued elements with the usual connections of the elements  $i$  and  $j$ .

Another important aspect of the general case of the MANN is the study of the possible types of the activation function  $f$  and its influence on the properties and behavior of the MANN solutions.

As mentioned above, the poorly investigated problems are tasks related to the activation function  $f$ . Theoretically, the tasks related to  $f$  can also be discussed in a similar way with the possibility of the multivalued activation function  $f$  of states and connections of elements above. That is, in the most general case, the problem of MANN dynamics can be reduced to a general mathematical scheme of the dynamics of multivalued map  $f: \Omega \Rightarrow \Omega$ . In the most general case,  $f$  is a multivalued mapping, and is the function on the space  $\Omega$  of subsets of the space. A well-developed mathematical apparatus of the behavior of multivalued dynamic systems and mappings can already be applied to such a formulation. It is

theoretically known that in such case a language of dynamical systems and attractors, the task of learning ANN is equivalent to constructing (synthesizing) dynamical systems with the desired configuration of attractors. Such a theoretical (and, in principle, complete answer) way is complex and the subject of further numerous studies. However, due to the complexity of this way, and especially the need for computational experiments, it is important investigate simpler cases. The simplest (although still new and interesting case) is the case of a single-valued function  $f$ , but with the existence of a set of branches as solutions of nonlinear dynamic MANN equations. The results of such studies for simplest example will be presented (very briefly) in section IV.

In this case, as the first attempts, you can take for analogues of the Hopfield network and analogs with the well-known Hebb rules. But at this stage it is generally possible to consider different versions of the functions  $f$ . We also note that now in the case of unambiguous ANN, various machine learning options, especially deep learning, are becoming more applicable.

You can call this the methods of identifying rules (the laws of evolution ANN). As applied to the MANN variant with multi-valued activation functions  $f$ , this leads to the need for further development in the direction of identification of multi-valued dynamic systems. In particular, finding multi-valued functions  $f$  with different machine learning options should be developed, but in the case of multi-valued components. Remark here that the examples of learning processes for quantum neural networks [7-10] may be useful as the prototype of learning for more complex networks with multivaluedness.

Note that the identification of the MANN for really important tasks is closely connected with the interpretation of the MANN solutions.

### III. TOWARD THE INTERPRETATION OF MANN SOLUTIONS.

In classical ANN, the postulate is initially and even automatically built in concept that the ANN solution is unambiguous (or their trajectories is single-valued, like in single-valued dynamic systems). Then usually there are no problems with the interpretation of the solution in single-valued case. In the case of discrete time, at these times, any quantities that uniquely characterize the states of a real system are measured. The values of the ANN solution at the same times are approximations (single-valued) of the true values of the system states.

Regarding the MANN, due to the potential ambiguity of their solutions, there are much many opportunities for interpretations. It is important that the systems for the study and management, which are used by the ANN, are usually unambiguous. However, in connection with the MANN, it becomes necessary to allow as the possibility and ambiguity of the behavior of the source systems, i.e. the existence of

many branches. So far, this has been accepted in the interpretations of quantum mechanics. But it looks like this behavior may be interesting for other systems.

If there are many branches for a general MANN, then the presence of many branches represents a larger set of presumable interpretations. If it is possible to obtain (measure) the value from each branch, then a set of values is obtained both as measurements and as the values of the MANN. However, then the perceived picture of the world would be perceived as multi-valued. However, usually as a result of measurements, a single-valued value of the variable being measured is obtained (not multiple-valued). Remark that remember the collapse of wave function to single state of quantum system under measurement. Then even in the case of many branches, there may exist an operator that turns many values into one perceived value. Often this operator may be an averaging operator. The rules for such averaging operator are very important for interpretation. So, the building of such operators constitute separate research problem and may including the investigation of history of considered multiple-valued network. At the same time, we note that according to this unambiguous sequence, one can try to train or ANN for unambiguous interpretation, or at the same time restore the approximate learning rules for MANN along the only accessible measurable trajectory.

#### IV. NEURAL NETWORKS TAKING INTO ACCOUNT STRONG ANTISPATION AS AN EXAMPLE OF ANN WITH MULTIPLE-VALUEDNESS

In the previous subsections of the article, thoughts were given on the new possible features of multi-valued neural networks and the resulting prospects for theory and practice. It seems that with the further development of research in this direction, the proposals for the design of MANN and their applications will increase.

However, it is already possible to specify an example of MANN with multiple values. Namely, this may be an example of ANN, taking into account the strong anticipation property introduced in the 90s by Daniel Dubois. Details of the study of such ANN are given, in particular, in [10] and others. Here we present only a very short summary, including information about not yet very common information about strong anticipation.

Since the beginning of the 90th, D. Dubois introduced the system with strong anticipation [11]: “An incursive discrete system is a system which computes its current state at time  $t$ , as a function of its states at past times  $\dots, t-3, t-2, t-1$ , present time,  $t$ , and even its states at future times  $t+1, t+2, t+3, \dots$ ”

$$x(t+1) = A(\dots, x(t-2), x(t-1), x(t), x(t+1), \dots) \quad (7)$$

where the variable  $x$  at future times  $t+1, t+2, t+3, \dots$  is computed in using the equation itself”.

In relation to MANN, this leads to the need of considering dynamic equations of the form:

$$s_i(n+1) = f(\{s_j(n)\}; \{s_j(n+1)\}; \dots; \{s_j(n+m)\}; \{J_{ij}\}) \quad (8)$$

$i = 1, 2, \dots, N$ ;  $f$  - activation function,  $s_i(n)$  - element state at the time moment  $t = n$ ,  $n = 0, 1, 2, \dots$ .

First of all, the case of the Hopfield model modification had been considered taking into account strong anticipation, but when  $f$  is usual single-valued piecewise linear function and  $\{J_{ij}\}$  is a matrix of connections fixed in time. One of the options for taking into account the anticipation is additive:

$$f = f((1-\alpha) \sum_{j \neq i} J_{ij} s_j(n) + \alpha \sum_{j \neq i} J_{ij} s_j(n+1)) \quad (9)$$

where  $\alpha \in [0, 1]$  is the parameter to account for strong anticipation. The case  $\alpha = 0$  corresponds to the absence of an account of anticipation. In the case of relation (9), only the case of a unique matrix of bonds and a unique function of activation is given.

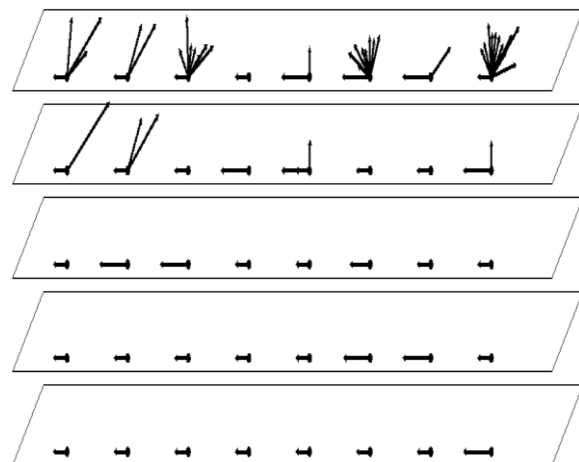


Fig. 2 The increasing of number of solution branches in case with strong anticipation

In Figure 2 below, for an illustration, a graph of the solution of the modified Hopfield network is shown taking into account strong anticipation ( $\alpha = 0.1$ ,  $N = 8$ ),  $N$  is the number of network elements,  $f$  is piecewise linear. Elements of a neural network are shown on each of the horizontal planes corresponding to same time moments. Each arrow on the figure corresponds to a specific value from the set of possible values of the element states at a given moment of time, and

the length of the arrow indicates the amplitude of separate value from set of presumable solutuins. Five time moments are represented at the figure (time moment increase from bottom to top of the figure 2).

The analysis of Figure 2 shows the main possible new features of the multiple-valued MANN behavior. Namely: 1) the possible multivalued behavior of the MANN; 2) possible inhomogeneous solutions (for example, some elements have unambiguous solutions, and some elements have multi-valued values); 3) the potential possibility of "multiplication" of ambiguity; 4) implicit change of structure of solution due presumable change of branches in solution.

The property 4 are very interesting and is new in comparing with case of neural networks with fixed structure. Moreover the useful space of values of elements may be unknown before the computation process.

It is very important that for other MANN with ambiguities we should expect similar opportunities in behavior. Therefore, it makes sense to consider the hypothetical possibilities arising from this in theory and implementation of neural networks. Some of these features will be discussed very briefly in the next subsection.

#### V. POSSIBLE CONSEQUENCES OF THE USE OF ANN WITH MULTIPLE-VALUEDNESS

To demonstrate the potential of possible polysemy in ANN, here we will give very briefly some of the tasks and hypotheses arising from such possibilities. Multiple-valuedness suggested the following research directions: a possible connection with a probabilistic description; proposals for a relationship with quantum mechanics; generalizations of the classical theory of calculations and the theory of automata, and much more [10].

But definitely, one more direction should be emphasized, returning to the original sources of the concept of artificial neural networks. Namely - it is concerned understanding and modeling of mental processes in the brain. It is generally accepted that ANN is one of the main approaches to understanding the processes in the brain (including their modeling) [4, 5, 10, 12, 13] and a lot of other research.

But almost always common in modern approaches is the postulate that considered basic ANNs are with unambiguous solutions. However, it can be proposed the presumable multiple-valuedness as a consequence of the property of possible ambiguity described in subsections 2–4. A discussion of some of these problems is given in the work [13] on models of consciousness. There are some possible consequences of the assumption about the role of polysemy for the processes of consciousness. Thus, the act of consciousness can be associated with the choice of an unambiguously perceived sensation from a whole set of possible states.

Also an important area is the search for the possibilities of implementing multi-valued ANN not only for living systems. Possible candidates for the implementation of MANN can be found in modern photonics, where multi-photon element states

[14] and multi-photon and multi-state message packets of photons between elements [15] are allowed.

#### VI. BRANCHING NEURAL NETWORKS

Branching of solutions of MANN opens absolutely new prospects for application of neural networks. Remark that neural networks had been constructed for definite problems solutions. In case of MANN because of presumable time-dependent branching we can named such case as branching neural network (BNN). Due such possibilities the different networks may have different temporal branching structure. So, we can pose the question of searching networks with temporal structure of branching with optimal structure of set of branches. Moreover it may be posed the problem of optimal control of branches set for optimization the computations. Remark that for single-valued neural networks and just for multiple-valued networks with fixed structure of branches such problem cannot be established.

Recently some efforts have been registered to expand the usual scheme of neural network construction. It is quite new to consider the case of multidimensional neural networks, where elements can take several values simultaneously [10]. One of the steps in the development of such a concept is branching neural networks, which are described in this paper.

**Definition.** Let's call branched artificial neural networks in which the number of possible states (Case 1) or the number of elements increases with time by dividing the elements with multiplication (Case 2).

The Case 2 is more simple and follows in many respects the classical scheme.

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| X |   |   |   | X |   |   |   |
| X | X | X | X | X | X | X | X |
| X | X | X | X | X | X | X | X |

Fig. 3 Simplest example of Case 2. The number of elements double at each time step. X marks the "active" values of element. Here we represent three time steps (from top to down).

Much more interesting is the Case 1 with an increase in the number of possible states, or the case of combining cases one and two. Here on the Fig.4 we give the simplest example for Case 2.

|     |     |     |
|-----|-----|-----|
| O/O | X/X | X/O |
| X/O | X/X | X/O |
| O/O | X/X | X/X |

Fig. 4 Simple example of the network with constant number of elements (here N=3) and many but fixed number of states (here M=2). Here X correspond to "active" states of elements and "O" to "nonactive" (virtual) states. For example X/X corresponds to two "active" states of element at the same time. Here we represent three time steps (from top to down).

|   |             |             |             |         |          |             |
|---|-------------|-------------|-------------|---------|----------|-------------|
| X | X           |             | X           |         |          |             |
| X | X/X         |             | X/O         |         | O/O      |             |
| X | X/X/<br>X/X | X/O/<br>O/X | X/O/<br>O/O | X/<br>X | O/O<br>X | O/O/<br>O/O |

Fig. 5 Case of increasing the number of elements and increasing the number of states with time. Here we represent three time steps (from top to down). We show 3 elements at initial moments, 4 elements at second and 7 elements at third step. Some multiple states of elements displayed for third time step (for example X/X/X/X).

In each of these three cases, one of the main problems is teaching the neural network. Let us first discuss the variant with multiplication of elements as a simpler one. For simplicity we will consider that an element has only two states and the number of elements doubles at each step. One of the common tasks of teaching classic neural networks is the following. There is a set of inputs and corresponding outputs. The training consists in selecting the neural network parameters. Most often the selected parameters are the values of links between elements. First, let's remind the usual training of back-propagation. If initially the structure of elements and links is set, then the learning coincides with back-propagation, only in the classic back-propagation layers and links are taken regardless of time, and we have layers of the network at different points in time.

The learning problem looks quite different when the composition of cells in layers is initially unknown and changes over time. One way out is to initially set the maximum number of cells on each layer that are initially assigned the "inactive" status. Actually working cells are "active". Note that this representation is very uneconomical. Thus, for  $N$  temporary steps, you should work with the  $N \times M$  cells ( $M$  - the maximum number of elements on the layer). For  $2^1 + 2^2 + \dots + 2^N$  "active" cells we need all the elements (so far  $2^N < M$ ) for the simplest case of doubling the "active" elements. At the same time, we still need to process  $M^2 \times N$  links (compared to only  $2^2 + 4^2 + 8^2 + \dots + M^2$  for the "active" elements).

In the case of unknown initial configurations of elements on the layers, it is necessary to specify a rule of cell division "nonactive" <sub>$i_k$</sub>   $\rightarrow$  "active" <sub>$\{i+1,1,\dots,M\}$</sub>  depending on the previous solution, ie. where  $i$  - moment of time  $k$  - the element index on  $i^{th}$  time layer. It is logical to assume that if the element is not divided, then we can assume that the relationships of "active" cells with "active" cells may not change (remaining "trained"). One of the possible options is to "learn" the newly emerging links "active"  $\rightarrow$  "inactive", "inactive"  $\rightarrow$  "inactive", "old"  $\rightarrow$  "new", "new"  $\rightarrow$  "new".

However, it may turn out that the "active"  $\rightarrow$  "active" links may also be restructured. One way out of this situation is that every time there are new elements (multiplication), it is necessary to retrain all active connections. But this, of course, is very resource-intensive. Therefore, one should look for intermediate (approximation) solutions. Probably, we should rebuild the links locally only for newly born elements. Thus, even in the case of branched neural networks, many research tasks already arise from one case. Note only that in this case the training parameters (and in the tasks of error minimization) are not only the links between elements, but also the number and architecture of elements.

Probably, it is necessary to reconstruct connections locally only for newly born elements. Thus, even in the case of branched neural networks, many research tasks already arise from Case 1. Note only that in this case the training parameters (and in the tasks of error minimization) are not only the links between the elements, but also the number and architecture of the elements.

The Case 2 corresponds to another branching possibility - namely, the multivalence of branching states of the elements, including the increasing number of states in the time of each of the elements. Again, the simplest option is to set the maximum number of states of the element. And initially all states are "inactive", and among them there are active at the moment of time. Then in the process of learning it is necessary to set up all the links between all the states, which is again resource-intensive.

Experience of working with multivalued neural networks has revealed the existence of the following situation. With a fixed number of elements, the number of "active" states of elements can reach tens and hundreds of thousands over time. And there is a problem what to do with such reproduction (probably, exponential over time). It would seem that it was possible to multiply the number of elements so that a single element had a maximum permissible (limited with a fixed restriction) number of elements. However, all this is resource-intensive and difficult to implement. The solution is coarsening of the element state space. In this case each element will have the maximum number of state classes  $K$ . Probably, at the following researches it will be proved that such coarsening will be used with adequate approximation of the case of infinitely multiplying states  $K \rightarrow \infty$ . Perhaps the statistical mechanics with a brutal apparatus will be useful here.

Here arises also a whole complex of proposals for the construction of neural networks with a given optimal structure for a specific given problem.

It is natural to raise a question about practical applicability of branched neural networks. The first example is dendritic neural networks, especially in brain and its models. The second example of systems with branching structure is interpretation of Everett quantum mechanics and branching neural networks are suitable for their modeling. And thirdly, branching neural networks can be useful as models for classical branching processes.

Presumable multiple-valued structure of MANN (especially of branching networks) allows considering the problem of



parallel computation on the new background. As usual the new properties follows to the new research problems. For example new aspects arises in the capacity problem of neural network which earlier had been considered for case of single-valued networks (see [16]). Multi-valuedness can increase the capacity. Also the branching of networks may follows to the super-computation ability of networks (see the discussion of cellular automata case in [17]).

## VII. CONCLUSIONS

Thus, in the present paper, we give considerations on artificial neural networks with multi-valued solutions. Architecture and learning processes are discussed. Generalization of Hopfield model with strong anticipation is considered as the example of neural networks with multiple-valuedness. Both general considerations and examples of how to implement the concept in the form of the simplest Hopfield model with strong anticipation are given. Opportunities for developing the concept and setting specific research objectives are also discussed. It also should be stressed implicit multiple-valuedness of the solutions by temporal branching of solutions and the possibilities of branching neural networks.

## REFERENCES

- [1] S. Haykin Neural Networks. A Comprehensive Foundation. Prentice Hall. New Jersey: Prentice Hall, 1999.
- [2] S. Russell, P. Norwig Artificial Intelligence: A odern Approach. 3<sup>rd</sup>. ed. New Jersey: Prentice Hall, 2009.
- [3] D. Graupe Principles of Artificial Neural Networks. Singapoure: World Scientific ed., 2013.
- [4] R. Kozma, W. Freeman Phase Transitions in the Cerebral Cortex – Enhancing the Neuron Doctrine by Modeling Neural Fields. Springer, 2015.
- [5] Computational Neurology and Psychiatry. Eds. P. Erdi, B. Bhattacharya, A. Cochran, Cham: Springer, 2017.
- [6] Chrisley R.L. Quantum learning. In: PulkanenP. Pylko (eds.), New directions in cognitive science: Proc. Of Int. Symposium , Saariselka, 4-9 Augusy, Lapland, Finland, Helsinki, Finnish Association of Artificial Intelligence, 1995. P.77-89.
- [7] Perus M. Multi-level synergetic computation in brain.. Nonlinear Phenomena in Complex Systems, 2001. vol, 4, n. 2. P. 157-193.
- [8] Miranker L. W. Quantum neuron Yale University, 1993. 18 p.
- [9] Ventura D., Martinec T. A quantum computational learning algorithm. ArXiv quant-ph/9807052
- [10] Makarenko A. Multivaluedness Aspects in Self-Organization, Complexity and Computatios Investigations by Strong Anticipation. Chapter in Book: Recent Advances in Nonlinear Dynamics and Synchronization. EDs. Kyamakya K., Mathis W., Stoop R., Chedjou J., Li Z. , Springer; Cham, 2018. Pp.33-54.
- [11] Dubois D. Incursive and hyperincursive systems, fractal machine and anticipatory ence. Published by the American Institute of Physics, AIP Conference Proceedings 573, 437 – 451, (2001).
- [12] Makarenko A. Neural Networks with Strong Anticipation and Some Related Problems in Complexity Theory. Chapter 12, in. Ser. Studies in Systems, Decisions and Control. Vol. 55. Ed. George H. Dimirovski. Springer, 2016. Pp. 267-281.
- [13] Makarenko A. Hyperincursion in modeling of brain activity phenomena. Proceedings Of 30<sup>th</sup> Int. Anniversary Conf. On System Research, Informatics and Cybernetics. Baden-Baden, August 2018. Baden-Baden, August 2018. vol.5, 5 p.
- [14] Behrman E.C., Niemel J., Steck J.E., Skinner S.R. A quantum dot neural network. Proc. 4<sup>th</sup> Workshop on Physics on Computation, Proc. Int. Joint. Conf. on Neral Network, November 1996. P. 22-24
- [15] Erhard M., Fickler R., Krenn M., Zeilinger A. Twisted photons: new quantum perspectives in high dimensions. Lights: Science & Applications. 2018. Vol. 7, 17146
- [16] Kurkova V, Kolmogorov’’s theorem and multilayer neural networks. Neural Networks, 1992. Vol. 5, issue 3. Pp. 501-506
- [17] Makarenko A. Multivaluedness in cellular automata with strong anticipation and prospects for computation theory.WSEAS Transactions on Information Science and Applications. 2020. Vol.17. pp. 69-78