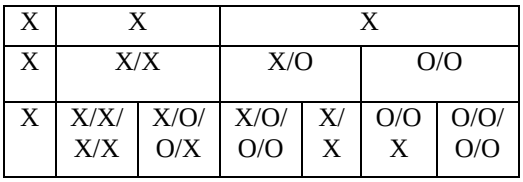

Fig. 5 Case of increasing the number of elements and increasing the number of states with time. Here we represent three time steps (from top to down). We show 3 elements at initial moments, 4 elements at second and 7 elements at third step. Some multiple states of elements displayed for third time step (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 $2^1 + 2^2 + ... + 2^N$ "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 $2^2 + 4^2 + 8^2 + ... + M^2$ 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 $"nonactive"_{ik} \to "active"_{\{i+1,1,...,M\}}$ 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" $\to$ "inactive", "inactive" $\to$ "inactive", "old" $\to$ "new", "new" $\to$ "new". However, it may turn out that the "active" $\to$ "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 $K \to \infty$. 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. 3rd. 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. Symposum , 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 Sunchronization. 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 30th 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. 4th 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 & Aplications. 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