

Dynamic Logic Learning in Cognitive-Based Situation Models

Roman Ilin, Leonid Perlovsky

Sensors Directorate, Air Force Research Laboratory, Hanscom AFB, MA
roman.ilin@hanscom.af.mil

Submission type: Oral presentation paper

Abstract—we present a cognitive modeling framework called Neural Modeling Fields (NMF) and its application to situation learning and categorization. We discuss how this framework is related to the perceptual symbol systems theory of cognition (PSS). Essentially, the mathematical apparatus of NMF is a way to learn the frames and simulators described qualitatively by PSS. For the purposes of this work, a situation is modeled as a set of objects and relationships that exist among them. Here we consider object recognition problem solved and demonstrate how the NMF framework is used to learn high level concepts such as situations.

Index Terms—Neural Modeling Fields, Dynamic Logic learning, Perceptual symbol systems, Situation modeling, Situation learning

I. INTRODUCTION

Our approach to situation modeling and learning is based on the cognitive modeling framework called Neural Modeling Fields (NMF) with its main operating principle referred to as Dynamic Logic learning [1-9]. After describing the main ideas of NMF in section II we relate it to the Perceptual Symbol Systems theory (PSS) in section III. Both frameworks aim at describing a fully functional cognitive system capable of representing types, categorization, categorical inference, abstract concepts, among others. We discuss how the basic concepts from PSS - frames and simulations – correspond to the basic ideas from NMF. The task of situation learning is a partial case of learning high level abstract concepts. Both NMF and PSS outline a hierarchical cognitive model which can be applied for situation learning.

Our methodology of situation modeling is described in section IV. Section V discusses the hierarchical model that can be built based on the technique presented in this contribution. We provide an example of our approach with synthetic data in

section VI. The discussion of further research in section VII is followed by conclusion in section VIII.

II. NMF AND DL

Neural Modeling Fields together with Dynamic Logic learning form a mathematical framework for learning from data. This framework and its applications are described in [1]. The basic premise of NMF is the idea that human cognition must combine the prior knowledge with adaptive learning. Moreover, the proper balance between these two sides of cognition is the key to its successful operation. The dominance of prior knowledge leads to the view that cognitive processes manipulate logical statements, which makes learning very difficult. The dominance of adaptive learning leads to the connectionist approach, which does not provide a systematic way of maintaining the prior knowledge. The NMF proposes a possible middle ground by giving an architecture consisting of parametric models corresponding to the concepts manipulated by cognitive processes. The models provide a placeholder for the prior knowledge, and the model parameters are adapted by learning.

The NMF cognitive system thus contains various models of external objects and also of complex concepts. The models are fuzzy initially since they do not correspond to any particular object. Learning begins when the external data come through the sensors and interact with the models. The models must adapt to better match the data and thus better correspond to the reality. Such adaptation occurs on multiple levels, from simple object classification to learning to identify complex situations.

Dynamic Logic learning is the mechanism of efficient learning in the NMF system described above. The main idea of DL is that learning is always accomplished by the gradual transition from fuzzy to definite model parameters and data associations.

The *data association* problem is a major obstacle to efficient learning of concepts. Every piece of sensor data comes from one or sometimes several objects in the environment. The number of possible mappings between the data and the models grows exponentially with the number of models and the amount of data. Testing each of the possible data association mappings is a huge computational task. The DL approach is to adapt the data associations and the model parameters simultaneously. In the beginning of the process, each model is associated almost equally with all of the data. As the model parameters begin to adapt to the data, the associations become more definite. At the end of the process, correct data associations emerge together with correct model parameters. This conceptual description has been formulated mathematically.

Instead of focusing on the mappings between the data and the models, we can define the total similarity between all the models and all the data points. The total similarity will depend on the similarity between individual data elements and individual models. This similarity can be described as

$$L(\mathbf{x}|\mathbf{M}(\mathbf{S})) = \prod_{n=1}^N \left[\sum_{h=1}^H l(n|h) \right], \quad (1)$$

where \mathbf{M} stands for the set of all models, \mathbf{S} is the set of model parameters, N is the total number of data points, H is the total number of models, and $l(n|h)$ is the similarity between data point n and model h . If the function $l(n|h)$ is formulated in probabilistic terms, the total similarity can be interpreted as the total likelihood of the data given the models, making the framework similar to the finite mixture models in statistics. The maximization of the total similarity with respect to the model parameters provides the best match between the data and the models. Note that (1) does not explicitly contain the associations. It seems that if we maximize L with respect to model parameters \mathbf{S} , the data association problem will be solved automatically. Unfortunately, the total similarity usually contains many local maxima and its maximization by itself involves combinatorial computational complexity. In DL approach the associations are modeled explicitly as association weights $f(h|n)$, for $n=1\dots N$ and $h=1\dots H$. Each of these quantities varies between 0 and 1 corresponding to weaker or stronger associations between model h and data element n . The learning algorithm consists of iterative computation of the association weights and the model parameters. The crucial difference between the DL algorithm and other iterative optimization algorithms is 1) model initialization

guaranteeing fuzzy data association and 2) the presence of additional model parameters \mathbf{S}^{DL} controlling the fuzziness of data associations in the course of learning. The algorithm is summarized in Table 1. For more detailed description of this algorithm, please see [1].

TABLE 1
Dynamic Logic Learning Algorithm

1. Initialize model parameters $\mathbf{S}_h^I = \mathbf{S}_h^0, h = 1..H$
2. Compute association weights $f^I(h n) = l(n h) / \sum_{h_2=1}^H l(n h_2), h = 1..H$
3. Estimate model parameters, α is learning rate $\mathbf{S}_h^{I+1} = \mathbf{S}_h^I + \alpha \cdot \sum_{n=1}^N \partial \log l(n h) / \partial \mathbf{S}_h^I, h = 1..H$
4. Adjust fuzziness parameters $\mathbf{S}^{DL}_h^{I+1}$
5. Repeat steps 2-4 until convergence criteria are satisfied

III. PSS AND DL

The perceptual symbol system was described in [10]. A comprehensive discussion of PSS is beyond the scope of this paper and we will only mention the main ideas related to the subject of situation learning.

The main idea of PSS is that concepts are formed by cognition as a result of consolidation of multiple perceptual memories. This contrasts with the idea that our perceptions are transformed into different internal structures, referred to as *amodal* symbols. Instead, the concepts remain modal, and are referred to as perceptual symbols. Because of the consolidation, the perceptual symbols stop corresponding to a particular external object and, instead, correspond to an averaged object from its category, essentially turning into abstract models of the reality.

The key terms used by PSS are *frame*, *simulation*, and *simulator*. A frame is a system of perceptual symbols used to construct a specific perceptual category. For example, a frame called “car” contains memories of the previous encounters with cars that contribute to the ability to recognize cars. A frame can contain a single perceptual symbol or a hierarchy of symbols related to different parts of a complex object. For example a perceptual symbol for a car contains sub-symbols for doors, wheels, etc. Simulation is the process of constructing an instance of an object from the category. It can be thought of

as imagining an object based on the information contained in the frame. The frame and all the simulations it produces are called a simulator. In the language of PSS, object categorization is the process of matching the perceived entity with one of the simulations produced by the frames. If one of the frames is capable of producing a simulation that is similar enough to the perception, the perceived object is categorized.

Another important idea is that perceptual symbols do not correspond to holistic images of the perceived objects. Rather they are componential. Different object features are stored separately and are retrieved by the simulator when necessary. This property allows the flexibility necessary to explain the many properties of a fully functional conceptual system.

An attempt to implement a computational model of the PSS encounters the combinatorial complexity of data associations. Indeed, learning a frame requires making the decisions about which features of the corresponding concept are important and which are not. Since many frames are learned at the same time, the problem of data association appears, just like in the case of MHT.

A parallel can be drawn between the ideas coming from PSS and those of NMF. Specifically, the frames can be identified with models. The simulations can be identified with the DL process of maximizing the similarity between the data and the models and used in the processes of learning and categorization.

We suggest that the mathematical framework of DL is a possible implementation of the PSS. At this stage we are only concerned with efficient category learning. The other cognitive processes, such as productive reasoning, language, etc. can be addressed by similar methods in the future.

IV. SITUATION MODELING

The perception of elements in the environment and comprehension of their meaning are the building blocks of situation awareness [11]. The meaning refers to categorization of environment observations into higher level abstract categories, referred to as situations. In this contribution we define a situation as a set of objects and relationships that exist between them. Other, more complex definitions can be found in [12],[13]. For the purposes of this demonstration, we limit the sensor input to image data. In this case the types of relationships that are allowed to exist between objects can be limited to spatial relationships. Following [14] we allow five types of relationships necessary to characterize a visual scene. For a particular object, they are

1. Presence: is the object present in the scene.

2. Size: relative size of the object
3. Position: where the object is located
4. Support: object(s) supporting this object
5. Interposition: objects occluding or touching the object

Size and position are divided into subsets as follows. Size = {big, small, medium}, Position = {top, bottom, left, right, middle}.

We transform the perceived scene into a vector of binary features. This is schematically illustrated in Fig. 1. The left hand side of the figure contains the visualization of a situation. The multi-colored squares correspond to different objects identified in the scene, the interposed objects are located next to each other and the supporting objects are located below the supported objects. The size and the location of the squares correspond to the size and position of the objects in the scene. The locations of the squares do not correspond to the locations of the objects in the real scene as the visualization is only used for illustration. The scene is transformed into a binary feature vector displayed on the right hand side of Fig.1.

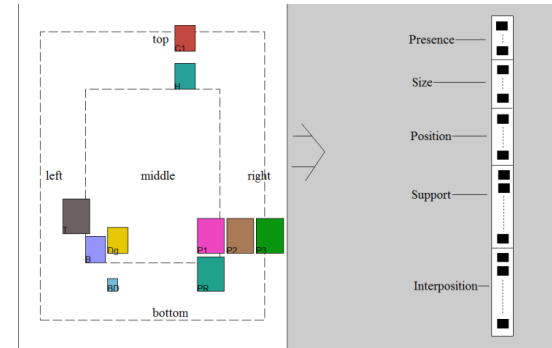


Figure 1. Situation visualization (left) and transformation to a binary feature vector (right). The black squares stand for 0 or 1. Please see text for more explanation.

We assume that there are n object categories that can be recognized by our sensor. The length of the binary feature vector is $n + 3n + 5n + n^2 + n^2$. The terms of this expression correspond to the five relationships: each object is either present or absent, each object has three possible sizes and five possible positions, there is a possible support relationship between each pair of objects and there is a possible interposition relationship between each pair of objects.

V. MODEL HIERARCHY

Obviously a scene does not enter our cognition as a binary feature vector. However it is easy enough to imagine a hierarchical system where such

transformation can take place. Such a system has been described in [15] and we will outline it here for completeness. The bottom layer of the hierarchy consists of models corresponding to object categorization. Examples of DL applications to object categorization and tracking can be found in [16-26]. The level of activation of each model in the bottom layer can be determined based on how similar the model is to a subset of the data. High level of model activation corresponds to the presence of the corresponding object in the scene and low level corresponds to its absence. The normalized activations of the bottom layer form a vector with features which can be approximated by 0 and 1. The size and position of each of the detected objects can be easily deduced from the corresponding models. The detection of relationships such as support and interposition can be done by considering model pairs. This can be implemented as separate NMF models. Thus the high and low levels of activations from the bottom layer form the binary feature vector, which serves as input to the top layers for subsequent categorization.

The DL algorithm uses similar functional models for each scene type and a similarity measure between the data elements and the model. In the case of binary feature vector, the model can be given by the vector of probabilities $\mathbf{p}_h = (p_{h_1} \dots p_{h_i} \dots p_{h_{D_x}})$, where D_x is the length of the binary vector, and each component of the vector stands for the probability of the corresponding feature to equal one. We essentially assume that the feature vectors are generated by a multivariate binomial distribution. The similarity between the feature vector \mathbf{x}_h and the model \mathbf{p}_h representing situation h is given as the conditional probability of the vector given the model [27]

$$l(n|h) = \prod_{i=1}^{D_x} p_{h_i}^{x_{n_i}} (1 - p_{h_i})^{(1-x_{n_i})}. \quad (2)$$

With the models and the similarities defined we can derive all the expressions in Table 1 and implement the algorithm. The parameter initialization plays an important role in the proper initialization of this algorithm. All the components of the probability vector \mathbf{p}_h must be initialized with values close to 0.5, corresponding to the maximum initial variance of the probability distribution.

VI. NUMERIC SIMULATION

Suppose that we want to be able to automatically distinguish the situation of a small group of people marching in a single file from the situation of a small

group of people having a picnic, and also to determine when the observed situation is neither of the two.

We compiled a list of possible objects, which includes twenty five items, such as person, road, tree, car, chair, building, etc ($n=25$). This means that the length of binary feature vectors equals to 1475. The situations can be described as follows.

Single file: three or four people walking after each other, on the road or in the field, trees, bushes, cars may be present.

Picnic: People standing or sitting on the ground, chairs, blankets, food may be present. Trees, cars, buildings may be present.

Random situation: any of the objects may be present.

All three situations include obvious restrictions on the relationships between different types of objects, such as that an object either supports another object or is supported by it; roads are located on the bottom or in the middle of the scene, etc.

We used the descriptions above to generate several data sets to test the algorithm. In addition to the essential objects and relationships characterizing each situation we added random objects and relationships. One of the data sets is illustrated in Fig. 2.

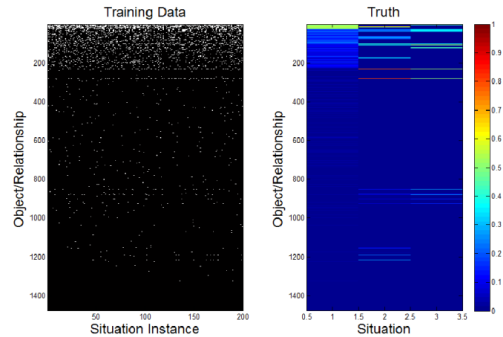


Figure 2. Visualization of the training data for situation learning. On the left, 200 binary feature vectors are shown with binary features along the vertical axis. On the right, the mean values for each feature are shown. These values correspond to the probabilities that need to be learned

The algorithm is initialized with $H=4$ models. The parameters of each model are set to probabilities randomly distributed between 0.48 and 0.52. This ensures that all the models are initially associated with all of the data. In the course of the algorithm execution, the models quickly converge to the correct probabilities. We purposely start with more models than there are true situations. Since in a real world application the true number of situations is unknown we have to assume a large enough number of situations. There are two mechanisms that help get

rid of the extra models. The algorithm automatically computes the probability of each model being present in the data. Low model probability means that very few situation instances have been associated with the model and therefore it can be discarded at the end. If two models have similar parameters and have become associated with the same subset of the data, they can be merged into one model. After a model is merged or discarded it becomes inactive and does not participate in the subsequent iterations of the algorithm. These two mechanisms ensure that the algorithm finishes with the number of active models corresponding to the true number of distinct situation categories contained in the input data.

The execution of the algorithm on the training data set with 200 situation instances is illustrated in Fig. 3. The top portion of the figure shows the evolution of model parameters and the bottom portion shows the evolution of the data associations. The top leftmost image shows the initial state with all of the model parameters close to 0.5. The bottom leftmost image shows the corresponding data associations with all 200 situation instances assigned to all of the models. As the algorithm executes, the data associations and the model parameters become more definite. By iteration number 15 most of the data are already assigned to the correct models. Some of the data are assigned to the extra model that is discarded at a later iteration. The final state after 49 iterations is shown on the rightmost images where the model parameters and the data associations are definite.

TABLE 2
Confusion Matrices

Training – 200 samples				
		DL Categorization		
		Random	Picnic	File
Truth	Random	118	2	0
	Picnic	2	38	0
	File	0	0	40
Testing – 1000 samples				
		DL Categorization		
		Random	Picnic	File
Truth	Random	329	0	1
	Picnic	0	328	2
	File	0	0	340

After the models have been trained on 200 situation instances, we generated 1000 different situation instances and used them for testing. In the testing phase the model parameters do not change. The data associations are computed and the testing data are assigned to the models with the strongest

associations. Since the true situation category is known we are able to estimate the efficiency of categorization. It is illustrated in Table 2. The accuracy of categorization is 99.7% in this case.

VII. DISCUSSION

The main motivation for this contribution was to demonstrate the feasibility of using NMF for situation learning. We derived and implemented the NMF algorithm capable of learning to categorize large binary feature vectors representing situation instances. In our synthetic examples we used situations with up to twenty five objects and five types of relations. The results demonstrated fast and accurate learning of situation categories.

The future direction of this research is the development of a multilevel system incorporating the perception and categorization of situations. Each level of this multilevel system forms the input into the next higher level as a set of signals produced by models identified, learned, or recognized at the given level. The more general and abstract higher-level models at the next level are learned as combinations of the lower-level models in the same way as the situations are learned in this contribution. In this way the hierarchical cognition of the mind can be modeled [28].

There is a very strong connection between NMF/DL and PSS in that the NML/DL methodology provides a mathematical foundation for PSS explaining how the perceptual symbols can be efficiently learned. We believe that combining the insights from PSS and NMF will result in a powerful cognitive architecture.

Another important direction of future research involves adding linguistic capabilities to the cognitive model. Language can be learned from the environment similarly to the way situations are learned [29-38]

Finally, the hypothesis of Dynamic Logic learning recently received support from the neuroscience community. In a recent study [39] it has been demonstrated that the object recognition by human subjects occurring in the temporal cortex is facilitated by the top-bottom signals originating in the orbitofrontal cortex. The initial top-down signals have been shown to correspond to low spatial frequency components of the incoming image, thus supporting the idea of transitioning from fuzzy to definite in the process of visual recognition.

VIII. CONCLUSION

This work demonstrated an approach to situation modeling and learning based on the cognitive model called Neural Modeling Fields. We described the main ideas of NMF and discussed its relationship to the Perceptual symbol systems theory. Numeric simulations using synthetic situation data demonstrated the feasibility and efficiency of our approach.

ACKNOWLEDGEMENT

The authors would like to thank the United States Air Force for continuing support of this research. This work was supported in part by AFOSR PM Dr. Jon Sjogren and PM Dr. Doug Cochran.

REFERENCES

- [1] Perlovsky, *Neural Networks and Intellect*. Oxford, 2001.
- [2] L. I. Perlovsky, "Physical Concepts of Intellect," *Proceedings of Russian Academy of Sciences*, vol. 354, no. 3, pp. 320-323, 1997.
- [3] L. I. Perlovsky, "Physical Theory of Information Processing in the Mind: concepts and emotions," *SEED On Line Journal*, vol. 2, no. 2, pp. 36-54, 2002.
- [4] L. I. Perlovsky, "Toward Physics of the Mind: Concepts, Emotions, Consciousness, and Symbols.," *Phys. Life Rev.*, vol. 3, no. 1, pp. 22-55, 2006.
- [5] L. I. Perlovsky and R. Kozma, Eds., *Neurodynamics of Higher-Level Cognition and Consciousness*. Heidelberg, Germany: Springer-Verlag, 2007.
- [6] L. I. Perlovsky, "Neural Mechanisms of the Mind, Aristotle, Zadeh, & fMRI," *IEEE Trans. Neural Networks*, vol. 21, no. 5, pp. 718-733, 2010.
- [7] L. I. Perlovsky, "'Vague-to-Crisp' Neural Mechanism of Perception," *IEEE Trans. Neural Networks*, vol. 20, no. 8, pp. 1363-1367, 2009.
- [8] R. Mayorga and L. I. Perlovsky, Eds., *Sapient Systems*. London, UK: Springer, 2008.
- [9] L. I. Perlovsky, "Modeling Field Theory of Higher Cognitive Functions," in *Artificial Cognition Systems*, A. Loula, R. Gudwin, and J. Queiroz, Eds. Hershey, PA: Idea Group, 2007, pp. 64-105.
- [10] L. W. Barsalou, "Perceptual symbol systems," *Behavioral and brain sciences*, vol. 22, pp. 577-660, 1999.
- [11] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, pp. 32-64, 1995.
- [12] G. Jakobson, J. Buford, and L. Lewis, "A Framework of Cognitive Situation Modeling and Recognition," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Washington, DC, 2006, pp. 1-7.
- [13] M. M. Kokar, J. J. Letkowski, R. Dionne, and C. J. Matheus, "Situation tracking: The Concept and a Scenario," in *Military Communications Conference, 2008. MILCOM 2008*, San Diego, CA, 2008, pp. 1-7.
- [14] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz, "Scene perception: detecting and judging objects undergoing relational violations," *Cognitive psychology*, vol. 14, pp. 143-177, 1982.
- [15] L. I. Perlovsky, "Cognitive high level information fusion," *Information sciences*, vol. 177, pp. 2099-2118, 2007.
- [16] R. Deming, J. Schindler, and L. Perlovsky, "Concurrent Tracking and Detection of Slowly Moving Targets using Dynamic Logic," in *2007 IEEE Int'l Conf. on Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Evolution, and Engineering (KIMAS 2007)*, Waltham, MA, April 30 – May 3, 2007.
- [17] R. Deming, J. Schindler, and L. Perlovsky, "Multitarget/Multisensor Tracking using only Range and Doppler Measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, Apr. 2009.
- [18] R. Deming, J. Schindler, and L. Perlovsky, "Track-Before-Detect of Multiple Slowly Moving Targets," in *IEEE Radar Conference 2007*, Waltham, MA, April 17-20, 2007.
- [19] R. Ilin and R. Deming, "Simultaneous Detection and Tracking of Multiple Objects in Noisy and Cluttered Environment using Maximum Likelihood Estimation Framework," in *IEEE International Conference, OCEANS '10*, Sydney, Australia, 2010.
- [20] R. Deming, "Automatic Buried Mine Detection Using the Maximum Likelihood Adaptive Neural System (MLANS)," in *Proc. of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, Gaithersburg, MD, Sept. 14-17, 1998.
- [21] R. Deming and L. Perlovsky, "Concurrent multi-target localization, data association, and

- navigation for a swarm of flying sensors," *Information Fusion*, vol. 8, no. 3, pp. 316-330, 2007.
- [22] L. I. Perlovsky and M. M. McManus, "Maximum Likelihood Neural Networks for Sensor Fusion and Adaptive Classification," *Neural Networks*, vol. 4, no. 1, pp. 89-102, 1991.
- [23] L. I. Perlovsky and R. W. Deming, "Neural Networks for Improved Tracking," *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1854-1857, Nov. 2007.
- [24] A. Schatzberg, A. Devaney, and R. Deming, "Maximum Likelihood Estimation of Target Location in Acoustic and Electromagnetic Imaging," in *Expanded Abstracts from the 65th meeting of the Society of Exploration Geophysicists*, Houston, TX, Oct. 1995.
- [25] R. Deming and L. I. Perlovsky, "A Mathematical Theory for Learning, and its Application to Time-varying Computed Tomography," *New Math. and Natural Computation*, vol. 1, no. 1, pp. 147-171, 2005.
- [26] P. L.I., "Multiple sensor fusion and neural networks," DARPA Neural Network Study, 1987.
- [27] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.
- [28] R. Ilin and L. I. Perlovsky, "Cognitively Inspired neural network for Recognition of Situations," *International Journal of Natural Computing Research*, 2010.
- [29] L. I. Perlovsky, "Integrating Language and Cognition," *IEEE Connections*, vol. 2, no. 2, pp. 8-12, 2004.
- [30] J. F. Fontanari and L. I. Perlovsky, "Meaning Creation and Modeling Field Theory," in *IEEE Int. Conf. On Integration of Knowledge Intensive Multi-Agent Sys.*, Waltham, MA., 2005.
- [31] F. J. Fontanari, V. Tikhonoff, A. Cangelosi, R. Ilin, and L. I. and Perlovsky, "Cross-situational learning of object-word mapping using Neural Modeling Fields," *Neural Networks*, pp. 579-585, 2009.
- [32] J. F. Fontanari and L. I. Perlovsky, "How language can help discrimination in the Neural Modeling Fields framework," *Neural Networks*, vol. 21, no. 2-3, pp. 250-256, 2008.
- [33] L. I. Perlovsky, "Language and Cognition," *Neural Networks*, vol. 22, no. 3, pp. 247-257, 2009.
- [34] V. Tikhonoff, J. F. Fontanari, A. Cangelosi, and L. I. Perlovsky, "Language and cognition integration through modeling field theory: category formation for symbol grounding," in *Book Series in Computer Science*, v. 4131. Heidelberg: Springer, 2006.
- [35] L. I. Perlovsky, "Symbols: Integrated Cognition and Language," in *Semiotics and Intelligent Systems Development*, R. Gudwin and J. Queiroz, Eds. Hershey, PA: Idea Group, 2007, pp. 121-151.
- [36] L. I. Perlovsky, "Language and Emotions: Emotional Sapir-Whorf Hypothesis," *Neural Networks*, vol. 22, no. 5-6, pp. 518-526, 2009.
- [37] L. I. Perlovsky, "Evolution of Languages, Consciousness, and Cultures," *IEEE Computational Intelligence Magazine*, vol. 2, no. 3, pp. 25-39, 2007.
- [38] L. I. Perlovsky and R. Ilin, "Neurally and Mathematically Motivated Architecture for Language and Thought," *The Open Neuroimaging Journal, Special Issue "Brain and Language Architectures: Where We are Now?"*, p. inress, 2010.
- [39] M. Bar, et al., "Top-down facilitation of visual recognition," *Proceedings of the National Academy of Sciences*, vol. 103, pp. 449-54, 2006.

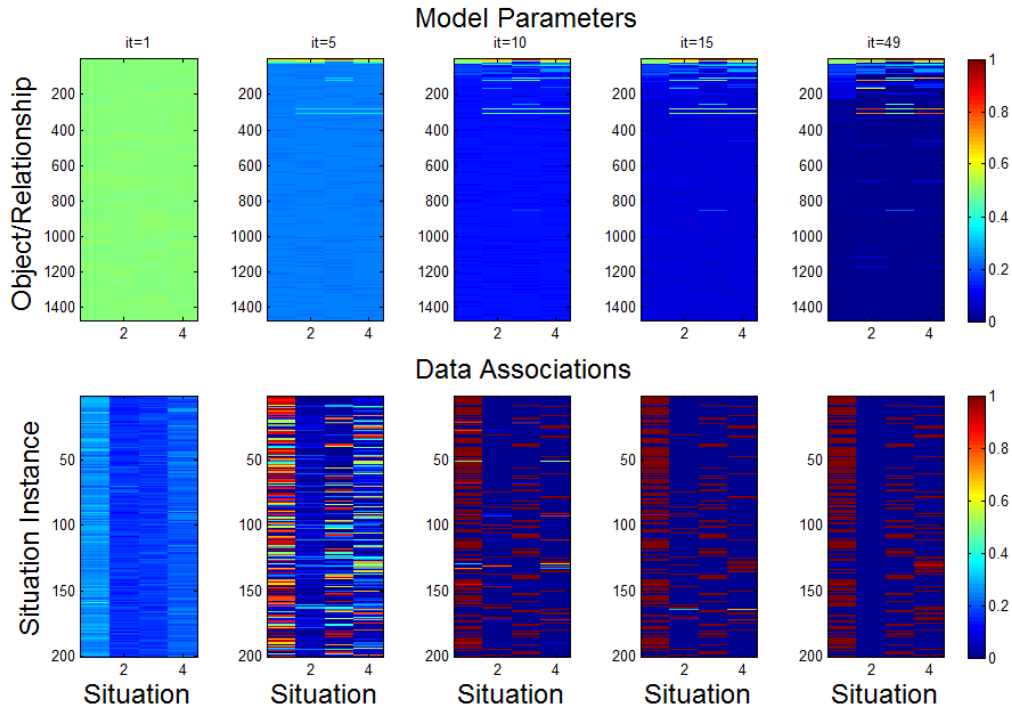


Figure 3. Visualization of the fuzzy to definite process of learning. The probability vector of each model is shown along the vertical axis. In the beginning of the process, on the leftmost image, all probabilities are close to 0.5 – green color. At the end of the process, on the rightmost image, the probabilities of models number 1, 3, and 4 converge to the true values – red and blue colors. Model 2 is discarded at the end. The sequence of images illustrates the gradual emergence of definite patterns of probabilities and data associations.