Hyperdimensional Representations in Semiotic Approach to AGI

Alexey K. Kovalev[1,2], Aleksandr I. Panov[2,3](✉), and Evgeny Osipov[4]

[1] National Research University Higher School of Economics, Moscow, Russia
[2] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia
[3] Moscow Institute of Physics and Technology, Moscow, Russia
panov.ai@mipt.ru
[4] Luleå University of Technology, Luleå, Sweden

Abstract. The paper is dedicated to the use of distributed hyperdimensional vectors to represent sensory information in the sign-based cognitive architecture, in which the image component of a sign is encoded by a causal matrix. The hyperdimensional representation allows us to update the precedent dimension of the causal matrix and accumulate information in it during the interaction of the system with the environment. Due to the high dimensionality of vectors, it is possible to reduce the representation and reasoning on the entities related to them to simple operations on vectors. In this work we show how hyperdimensional representations are embedded in an existing sign formalism and provide examples of visual scene encoding.

Keywords: Cognitive agent · Sign-based world model · Semiotic network · Causal tensor · Distributed representation · Symbol grounding

1 Introduction

When constructing intelligent systems that control the functioning of agents in a real, rather than a virtual environment, one of the main problems is the symbol grounding problem. In other words, for each concept that the system can operate with, it is necessary to map some idea, which is based on the signals coming from the agent sensors. It is human nature to operate with symbols, i.e. some indivisible entities representing the concepts, while existing computer architectures restrict the low-level representation of information in intelligent systems where binary numbers are commonly used.

At an early stage of the rise of artificial intelligence, one of the leading hypotheses that captured the minds of researchers for a long time and determined the development of the field for years to come was the hypothesis that "a physical symbol system has the necessary and sufficient means for general intelligent action" proposed by Allen Newell and Herbert Simon [1]. However, in the practical implementation of such systems, researchers encountered several problems, the main among which was the symbol grounding problem mentioned above.

B. Goertzel et al. (Eds.): AGI 2020, LNAI 12177, pp. 231–241, 2020.
https://doi.org/10.1007/978-3-030-52152-3_24

Despite the fact that the research and development of symbolic artificial intelligence methods continue, at present the connectionist approach using artificial neural networks is leading in the number of applications and the attention of researchers [2]. In recent years, the neuro-symbolic approach, which combines the advantages of both connectionism and symbolism, is gaining more and more popularity. As characteristic representatives, Markov Logic Networks [3, 4] and Logic Tensor Networks [5] can be distinguished.

Another direction of the neuro-symbolic representation can be called the approach to the use of hyperdimensional representations, proposed in [6]. Despite the fact that artificial neural networks are not used explicitly in this approach, the representations themselves obtained using hyperdimensional computing are very well suited for working with neural networks [7].

In this paper, we approach the solution of the symbol grounding problem using the previously proposed sign-based cognitive architecture [8–10], in which the processing of sensory information occurs in the image component of the sign representing some entity. We propose using hyperdimensional vectors to encode the precedent component of the sign image and demonstrate that this allows us to preserve the main advantages of the sign approach – the ability to represent operations and relationships based on operations with vectors and matrices. A new interpretation of the image component allowed us to describe complex visual scenes in a simpler language. In this work, we consider the capacity of the proposed mechanism for encoding sensory information.

The structure of the paper is as follows: Sect. 2 briefly provides the necessary information about sign-based cognitive architecture. Section 3 describes the use of hyperdimensional vectors as a representation of the image component of a sign and the operations on such representations. The fourth part shows the possibility of using hyperdimensional vectors for encoding elements of the causal matrix, with which the image structure is formalized, and provides an example of representing some visual synthetic scene in the form of a hyperdimensional vector for which simple reasoning schemes on the properties of objects presented on the stage are carried out.

2 Sign-Based World Model

In [8], the principles of the organization of sign-based cognitive architecture (SBWM) [9, 10] were described in detail, in particular, the process of reasoning expressed by applying certain mental actions by cognitive agents on their representation of the environment was described. Next, we briefly outline the basic principles of the SBWM following [11, 12].

The main element of the system is the sign, which corresponds with the agent's concept of any object, action or situation, then for simplicity, we will call the object, action or situation an entity. The sign consists of four components: image, meaning, significance, and name. The image component corresponds to the characteristic feature of the described entity. In the simplest case, an image refers to signals from the sensors of an agent that correspond to an entity. In the general case, we can say that the image of the sign coheres to the set of entity characteristic features with which the sign relates. The significance of the sign describes the standard application of the entity, adopted on the basis of experience in the interaction of a coalition of agents with the environment. The meaning of the sign is understood as the relation of the agent to the entity or the experience of the interaction of the agent with this entity, thus, the meanings are formed in the process of interaction of a concrete agent with the environment.

The sign components are described by a special structure - the causal matrix. A causal matrix is a tuple $z = \langle e_1, e_2, \ldots, e_t \rangle$ of length t where events e_i are represented by a binary vector of length h. For each index j of the event vector e_i (row of the matrix z), we will associate a tuple, possibly empty, of causal matrices Z_j, such that $z \notin Z_j$. We divide the set of columns indices of the causal matrix z into two disjoint subsets $I^c \subset \mathbb{N}, \forall i \in I^c\ i \leq t$ and $I^e \subset \mathbb{N}, \forall i \in I^e\ i \leq t$, such that $I^c \cap I^e = \emptyset$. The set I^c for the matrix z will be called the indexes of the condition columns, and the set I^e – the indexes of the effect columns of the matrix z. If $|I^e| = \emptyset$, i.e. there are no effect columns in the matrix, then we will say that such a matrix corresponds with the object. If $|I^e| \neq \emptyset$ what the presence of effect columns in the matrix means, then such a matrix corresponds with an action or process. The structure of the causal matrix makes it possible to uniformly encode both static information and features of an object, as well as dynamic processes.

A sign means a quadruple $s = \langle n, p, m, a \rangle$, where the name of a sign n expressed by a word in some finite alphabet, $p = Z^p$, $m = Z^m$, $a = Z^a$ are tuples of causal matrices, which are respectively called the *image*, *significance*, and *meaning* of the sign s. Based on this, the whole set of causal matrices Z can be divided into three disjoint subsets: images Z^p, significances Z^m, and meanings Z^a, such that $Z = Z^p \cup Z^m \cup Z^a$ which are organized into semantic networks, which we will call causal.

Formally, a causal network on images will be a labeled directed graph $W_p = \langle V, E \rangle$ in which:

1. each node $v \in V$ is assigned a causal matrices tuple $Z^p(s)$ of the image of a certain sign s, which will be denoted by $v \rightarrow Z^p(s)$;
2. an edge $e = (v_1, v_2)$ belongs to the set of graph edges E, if $v_1 \rightarrow Z^p(s_1), v_2 \rightarrow Z^p(s_2)$ and $s_1 \in S_p(s_2)$, i.e. if the sign s_1 is an element of the image s_2.

Causal networks on significances and meanings are defined in a similar way. The network on names is a semantic network whose vertices are the names of signs, and the edges correspond to special relationships. The semantic network on names will also be called a causal network.

These four mentioned above causal networks are connected using transition functions $\Psi_i^j,\ i, j \in \{p, m, a, n\}$ to the semiotic network. The transition function Ψ_i^j allows us to switch from i-th component of the sign to the j-th one. A semiotic network can be considered as an agent's knowledge base about the environment, taking into account its experience of interacting with the environment.

Formally, we will call the semiotic network $\Omega = \left\langle W_m, W_a, W_p, W_n, R, \Theta \right\rangle$ a sign-based world model, where W_m, W_a, W_p, W_n are causal networks of significances, meanings, images, and names, respectively, $R = \langle R^m, R^a, R^p, R^n \rangle$ is a family of relations on sign components, Θ is a family of operations on a set of signs. Operations Θ include such actions on signs as unification, image comparison, updating while learning, etc.

In the SBWM, the concept of the activity spread is defined, which allows the reasoning processes to occur in the semiotic network. After the activation level of the sign component exceeds a certain threshold, the component is considered as active. If the components of the image, significance, and meaning of the sign are activated, then the sign itself is also activated (its name is activated). At the same time, the activation process can proceed in the opposite direction: first, the name of the sign is activated, and then all sign components are automatically activated. If the activation level of a sign component is nonzero but does not exceed a predetermined activation threshold θ, then such a component is called pre-activated.

Spreading activity on a semiotic network is subject to *globa*l and *local* rules for spreading activity.

The *global rule* is that if one of the components of the sign s becomes active on a step t, the other components become pre-activated.

The group of local rules consists of four rules: ascending, predicting, descending and causal. The *ascending rule* says that if at the time t the component of the sign s becomes active, then all occurrences of this component in the causal matrix of other signs become active. The *predicting rule* determines that if at the time moment t an event e_t is active in any component of the sign s, then the events e_{t+1} of the same component are pre-activated. The *descending rule* establishes that if at the time moment t each event e_t in the tuple of causal matrices of the component $i \in \{p, m, a\}$ of the sign s is active, then the components i of all signs included in the event e_t are pre-activated. The *causal rule*: if an event e_t is active at a time t, then a predictive rule and a descending rule are consistently applied to all event-effects, with the amendment that the maximum activity applies.

3 Representation of Symbols by Hyperdimensional Vectors

In recommender systems and natural language processing, a widely used approach is the one that translates localized one-hot representations of objects that the system works with into distributed representations. Moreover, in both problems, there is a decrease in dimensionality, because initial one-hot vectors can have tens of thousands of dimensions, while the standard length of a distributed vector, for example, for word representation, is 300. A classic example in recommender systems are models with hidden variables that use a singular matrix decomposition of users-items matrix [13], and the modification of such a decomposition, called the truncated singular decomposition, allows one to vary the dimensions of the representation, simultaneously solving the regularization problem.

For the problems of natural language processing, there also have been attempts to use the singular decomposition, for example, for the co-occurrence frequency matrix [14]. However, approaches based on iterative learning of representations in the corpus of texts, such as word2vec [15, 16] and GloVe [17], gained wide popularity. In the original word2vec article, CBOW (continuous bag of words) and skip-gram models are proposed. In CBOW, the central word in the window is predicted from surrounding words by a certain contextual window that runs through text whose size is a hyperparameter. In skip-gram, the inverse problem is solved – according to the central word, it is predicted whether another word enters its context. In essence, the CBOW and skip-gram models are neural networks with one hidden layer with a linear activation function, and the prediction is constructed as softmax from the scalar product of the vector of the central and context words.

In GloVe, the problem is formulated as follows, given a joint co-occurrence matrix whose elements correspond to the occurrence frequency of one word in the context of another, then let the scalar product of the vector representations of the central word and the context word approximate the logarithm of this value.

In [18], examples are given that such representations of words contain some semantic and syntactic information that allows us to solve problems of searching for analogies, for example, of the type "king:man :: woman:queen" using arithmetic operations on vectors:

$$v_{king} - v_{man} + v_{woman} \approx v_{queen},$$

where v_{king}, v_{man}, v_{woman}, v_{queen} vector representations for words "king", "man", "woman" and "queen" correspondingly. Similarly, analogies of the type "big:biggest :: large:largest" are solved.

For recommender systems, this approach allows you to specify the similarity between the vectors of users or items, for example, using the cosine distance, while with the "one-hot" representation, the distance between the vectors does not make any sense.

Similar results were obtained for computer vision problems [19, 20], when using an autoencoder, representations are learned that allow you to add or remove some details of an image by changing a specific coordinate.

All of the above approaches can be summarized as follows: we reduce the dimension of the original vector while simultaneously trying instead of a localized, uninterpreted representation, to obtain representations in which the coordinates carry some, often poorly interpreted, meaning.

On the other side of the scale lies an approach that, in contrast to the first, increases the dimension of the vector representation and deprives individual component of the vector of any interpretability. Moreover, the resulting representations are in some ways symbols, but symbols that can be operated on using vector operations. Let us consider this approach described in [6] in more detail.

The basis of this approach is the idea that for a sufficiently large dimension of the space for any randomly extracted and fixed vector from this space ~99% of the remaining vectors of the space will be quasi-orthogonal to this fixed vector. In this case, by quasi-orthogonality we mean that, for example, for binary vectors, the normalized Hamming distance between them will be approximately 0.5, and then any sufficiently small deviation from this value will indicate that these two vectors are not random, and one enters into a superposition of the other. This property of hyperdimensional spaces allows us to reduce the procedure of matching a given object of its hyper-dimensional representation to sampling a random vector. Thus, for each object or property that the system encounters, a random hyper-dimensional vector, for example, a binary one, is generated and put into correspondence with this object or property. All vectors obtained in such a manner are stored in a special memory called "Item Memory", where they are assigned a label corresponding to the encoded entity. For the Item Memory, a search operation that receives a vector and returns the vector closest to this one is defined. The search operation on the Item Memory can be considered as restoring the original vector of an entity from its noisy copy, the need for such an operation will be shown below. The property of hyperdimensional spaces described above just allows avoiding collisions for such an operation with sufficient space capacity and not too much noise in the input vector.

We briefly describe operations on hyperdimensional vectors.

The binding operation to two vectors associates the third, quasi-orthogonal to both initial vectors. For binary vectors, the binding is carried out using the elementwise exclusive or operation. Binding obeys the laws of commutativity and associativity. The semantic meaning of binding can be explained by the following example: let some object have a certain attribute a_i with a value v_j, we put them in correspondence with hyper-dimensional random vectors A_i and $V_j^{A_i}$. Then the binding $A_i \oplus V_j^{A_i}$ corresponds to assigning the value v_j to the attribute a_i. Also, the inverse operation to the binding is defined – an unbinding $A_i \oplus \left(A_i \oplus V_j^{A_i}\right) = V_j^{A_i}$ which returns one of the original vectors.

The bundling operation to a certain set of hyperdimensional vectors associates another hyperdimensional vector that is not quasi-orthogonal with respect to any of the vectors of the set. Bundling is implemented through the threshold sum:

$$[X_0 + X_1 + \dots X_n] = Y,$$

where $y_i \in Y$ and

$$y_i = \begin{cases} \sum_1^n x_i,\ x_i \in X_i, & \text{if } \sum_1^n x_i \leq thr \\ thr, & \text{if } \sum_1^n x_i > thr, \end{cases}$$

where *thr* is a threshold, which is a hyperparameter.

Bundling can be considered as a representation of the set of some objects. The commutativity and associativity of bundling are obvious.

Sometimes it becomes necessary to obtain a quasi-orthogonal vector from the original one, but so that this operation is reversible. To do this, permutation operations are used, which permute the coordinates of the vector according to a certain rule. A special case of permutation is a cyclic shift. Denote $X^{n>}$ whose vector coordinates are cyclically shifted to the right on n positions relative to the original vector X.

Let the state of a system at the initial time moment correspond to a hyperdimensional vector X_0, then all the states of the system at time moments $i = 1, \dots n$ can be expressed as follows:

$$\begin{aligned} X_1 &= X_0^{1>}, \\ X_2 &= X_1^{1>} = X_0^{2>}, \\ &\dots \\ X_n &= X_{n-1}^{1>} = X_{n-2}^{2>} = \dots = X_0^{n>} \end{aligned}$$

Applying unbinding to a bundle

$$\begin{aligned} A_i \oplus B &= \left[A_1 \oplus V_j^{A_1} + \ldots + A_i \oplus V_j^{A_i} + \ldots + A_n \oplus V_j^{A_n}\right] \\ &= \left[A_i \oplus A_1 \oplus V_j^{A_1} + \ldots + A_i \oplus A_i \oplus V_j^{A_i} + \ldots + A_i \oplus A_n \oplus V_j^{A_n}\right] \\ &= \left[Noise + V_j^{A_i}\right] = \tilde{V}_j^{A_i} \end{aligned}$$

we get $\tilde{V}_j^{A_i}$ – a noisy version of the vector $V_j^{A_i}$ by which one can restore the vector $V_j^{A_i}$ by searching through Item Memory.

4 The Use of Hyperdimensional Vectors in the Sign-Based World Model

Let us consider the use of hyperdimensional binary vectors in the Signed-Bases World Model using the causal matrix of an image network as an example. We recall that the causal matrix z is a tuple of events e_i $z = \langle e_1, e_2, \ldots, e_t \rangle$ of a length t. We agree further that the hyperdimensional vector corresponding to the concept will be denoted by the same letter as the concept itself, only in capitals. Then, a vector E_i is assigned to each event e_i, the method of obtaining this vector will be described below. Since a tuple is an ordered set of elements, it is easy to set it through the variety of elements and their order. As described above, the set in hyperdimensional computations is specified by the operation of bundling over the elements included in it, to determine the order, we introduce a special hyperdimensional vector S that will correspond to the first column of the causal matrix. The subsequent columns will be defined through the cyclic shift of the vector S. The fact that some event E_i corresponds to the j-th column will be denoted through $E_i \oplus S^{j>}$. Then in general terms, the vector of the causal matrix can be represented as:

$$Z = \left[E_0 \oplus S + E_1 \oplus S^{1>} + \ldots + E_t \oplus S^{t>}\right].$$

If the causal matrix corresponds to the action, then we introduce two vectors S_c and S_e for the columns of conditions and effects, respectively, then:

$$Z = \left[E_0 \oplus S_c + E_1 \oplus S_c^{1>} + \ldots + E_j \oplus S_c^{k>} + E_{j+1} \oplus S_e + E_{j+2} \oplus S_e^{1>} + \ldots + E_t \oplus S_e^{l>}\right],$$

where $k + l = t$.

It is worth noting that if there is no need to maintain order, for example, for object matrices, then you may not introduce an additional vector S.

Let us return to the representation of an event e_i. An event corresponds to the simultaneous appearance of some attributes, therefore, if each attribute and all possible values of this attribute are associated with hyperdimensional vectors, then the event takes the form:

$$E = \left[A_1 \oplus V^{A_1} + A_2 \oplus V^{A_2} + \ldots + A_m \oplus V^{A_m}\right],$$

where A_i corresponds to i-th attribute and V^{A_i} is its value.

Thus, following the structure of a causal matrix and given HD representation of events we can collapse the whole matrix into the corresponding HD vector CM. This vector may act as an event in the formation of another causal matrix on the next level of abstraction. Properties of operations with HD vectors allow to keep structure inside of such representation and restore it if needed.

Consider an example of representing the causal matrix of a scene depicted in Fig. 1 as a hyperdimensional vector.

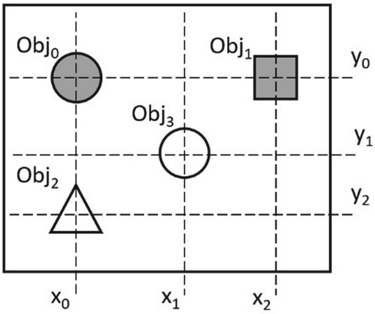

Fig. 1. The model scene

Let us suppose that the objects on the scene have the attributes c – "color", s – "shape", x – "x coordinate", y – "y coordinate" with the corresponding possible values: for the attribute "color" w – "white" and g – "gray", for "form" ci – "circle", t – "triangle" and sq – "square". Let us set the attributes and their values in accordance with the vectors C, S, X, Y, W, G, CI, T, SQ. The value of the attribute "coordinate x" will be encoded as follows. Assume directionality of the process of parsing of the visual scene (for example from top to bottom and from left to right). During parsing of the visual scene we find the leftmost object on the scene, in this case there are two such objects – Obj_0 and Obj_2, and assign a vector X_0 to them, then the next right object – Obj_3 will have a (relative to Obj_0) coordinate value $X_1 = X_0^{1>}$. For object Obj_4, we have $X_2 = X_0^{2>}$. The y coordinate values are encoded in a similar way. This allows us to move from the absolute coordinates to the relative ones. We also introduce vectors $O_0 \ldots O_3$ corresponding to scene objects. Now we can represent the vector corresponding to the causal matrix of the scene as:

$$\begin{aligned} SCENE = {} & [O_o \oplus [C \oplus G + S \oplus CI + X \oplus X_0 + Y \oplus Y_0] \\ & + O_1 \oplus \left[C \oplus G + S \oplus SQ + X \oplus X_0^{2>} + Y \oplus Y_0\right] \\ & + O_2 \oplus [C \oplus W + S \oplus T + X \oplus X_0 + Y \oplus Y_2] \\ & + O_3 \oplus [C \oplus G + S \oplus CI + X \oplus X_1 + Y \oplus Y_1]]. \end{aligned}$$

After that, if we want to find out the value of the object 2 form attribute, we must perform the following operations:

$$SCENE \oplus O_2 \oplus S = Noise + T = \tilde{T}.$$

Such operations allow performing the simplest reasoning on the representation of the scene using hyperdimensional vectors.

5 Discussion

While this paper focused on conceptual aspects of using HD representation in semiotic approach to AGI it is useful to get an intuition about possible applications of the presented encoding for flexible answering to complex queries. Take the last example of the *SCENE* encoding. Suppose the task is to extract objects to the right of object Obj_3. To do this the following computational steps should be performed.

1. Similarly to the example of extracting the value of attribute S, extract the value of attribute X of object Obj_3.
2. Retrieve the clean copy of X_3 from the Item Memory.
3. Construct a bundle of all possible coordinates "to the right of X_3" by circularly shifting X_3 n times binding with X and bundling the result:

$$X_3^{right} = \left[X \oplus X_3^{1>} + X \oplus X_3^{2>} + \ldots + X \oplus X_3^{n>}\right]$$

4. Bind the resulting bundle with the *SCENE* vector. This operation will produce a bundle containing the noisy values of the objects on the queried coordinates.

$$SCENE \oplus X_3^{right} = Noise + \tilde{O}_1$$

5. Passing the result through the Item Memory of objects will reveal the identities of the objects (Fig. 2).

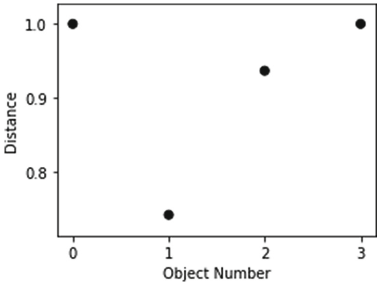

Fig. 2. Distance to objects vectors in Item Memory

Several important issues must be addressed for make the approach work on real use-cases. Specifically, one need to take into account the informational capacity of the bundles of HD vectors. Early results on the capacity were given in [21, 22]. Some ideas for the case of binary/bipolar HD vectors were also presented in [23, 24]. Probably the most comprehensive analysis of the capacity of different VSAs' frameworks has been recently presented in [25]. The practical dimensioning of the architecture for the case of visual questions answering application is a subject for future work and will be reported outside the scope for this article.

6 Conclusion

The paper proposes a new approach that allows solving the symbol grounding problem based on the agent sing-based cognitive architecture using hyperdimensional vector computations to describe the image component of the sign. Due to the use of hyperdimensional vectors to describe the precedent of the causal matrix component, it is possible to interpret the structure of the causal matrix and relations in the causal network as operations on a set of such vectors. The work provides a model example of the use of hyperdimensional vectors to represent a visual scene. In the future, we propose various applications of the sign-based architecture, including for personal cognitive assistants [26] that adapt to a specific user.

Acknowledgements. The reported study was supported by RFBR, research Projects No. 18-07-01011 and No. 19-37-90164.

References

1. Newell, A., Simon, H.A.: Computer science as empirical inquiry: symbols and search. Commun. ACM **19**(3), 113–126 (1976). https://doi.org/10.1145/360018.360022
2. Rumelhart, D.E., McClelland J.L.: Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, pp. 318–362, MITP (1987)
3. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1–2), 107–136 (2006). https://doi.org/10.1007/s10994-006-5833-1
4. Wang, J., Domingos, P.: Hybrid markov logic networks. AAAI **137**(1–2), 1106–1111 (2008)
5. Serafini, L., Garcez, A.D.A.: Logic tensor networks: deep learning and logical reasoning from data and knowledge. In: CEUR Workshop Proceedings, p. 1768 (2016)
6. Kanerva, P.: Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. Cogn. Comput. **1**(2), 139–159 (2009). https://doi.org/10.1007/s12559-009-9009-8
7. Bandaragoda, T., De Silva, D., Kleyko, D., Osipov, E., Wiklund, U., Alahakoon, D.: Trajectory clustering of road traffic in urban environments using incremental machine learning in combination with hyperdimensional computing. In: IEEE Intelligent Transportation Systems Conference (ITSC), pp. 1664–1670 (2019)
8. Kovalev, A.K., Panov, A.I.: Mental actions and modelling of reasoning in semiotic approach to AGI. In: Hammer, P., Agrawal, P., Goertzel, B., Iklé, M. (eds) Artificial General Intelligence. AGI 2019. Lecture Notes in Computer Science, vol 11654. Springer, Cham (2019) https://doi.org/10.1007/978-3-030-27005-6_12

9. Osipov, G.S., Panov, A.I.: Relationships and operations in a sign-based world model of the actor. Sci. Techn. Inf. Process. **45**(5), 317–330 (2018)
10. Kiselev, G.A., Panov, A.I.: Sign-based approach to the task of role distribution in the coalition of cognitive agents. In: SPIIRAS Proceedings, pp. 161–187 (2018). https://doi.org/10.15622/sp.57.7
11. Kiselev, G., Kovalev, A., Panov, A.I.: Spatial reasoning and planning in sign-based world model. In: Kuznetsov, S., Osipov, G.S., Stefanuk, V. (eds.) Artificial Intelligence. RCAI 2018. Communications in Computer and Information Science, pp. 1–10. Springer (2018). https://doi.org/10.1007/978-3-030-00617-4_1
12. Panov, A.I.: Goal Setting and Behavior Planning for Cognitive agents. Sci. Tech. Inf. Process. **46**(6), 404–415 (2019)
13. Koren, Y.: The BellKor Solution to the Netflix Grand Prize (2009)
14. Douglas, L.T.R., Laura, M.G., David, C.P.: An improved model of semantic similarity based on lexical co-occurrence. In: Communications of the ACM (2006)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, pp. 1–12 (2013)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, pp 3111–3119 (2013)
17. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation, **31**(6), 682–687 (2017). https://doi.org/10.1080/02688697.2017.1354122
18. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. Proc. OfNAACL-HLT **2013**, 746–751 (2013)
19. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, pp. 1–16 (2016)
20. Hou, X., Shen, L., Sun, K., Qiu, G.: Deep feature consistent variational autoencoder. In: Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, pp. 1133–1141 (2017). https://doi.org/10.1109/WACV.2017.131
21. Plate, T.A.: Holographic Reduced Representations: Distributed Representation for Cognitive Structures. Center for the Study of Language and Information (CSLI). Stanford, Redwood City (2003)
22. Plate, T.A.: Distributed representations and nested compositional structure, University of Toronto, Ph.D. Thesis (1994)
23. Gallant, S.I., Okaywe, T.W.: Representing objects, relations, and sequences. Neural Comput. **25**(8), 2038–2078 (2013)
24. Kleyko, D., Osipov, E., Senior, A., Khan, A.I., Sekercioglu, Y.A.: Holographic graph neuron: a bio-inspired architecture for pattern processing. IEEE Trans. Neural Netw. Learn. Syst. **28**(6), 1250–1262 (2017)
25. Frady, E.P., Kleyko, D., Sommer, F.T.: A theory of sequence indexing and working memory in recurrent neural networks. Neural Comput. **30**, 1449–1513 (2018)
26. Smirnov, I., Panov, A.I., Skrynnik, A., Isakov, V., Chistova, E.: Personal cognitive assistant: concept and key principals. Inform. In: IEEE Primen, vol. 13, pp. 105–113 (2019). https://doi.org/10.14357/19922264190315