Neuro-symbolic architecture for experiential learning in discrete and functional environments

Anton Kolonin[1,2,3]

[1]Aigents, Novosibirsk, Russian Federation
[2]SingularityNET Foundation, Amsterdam, Netherlands
[3]Novosibirsk State University, Novosibirsk, Russia
akolonin@gmail.com

Abstract. The paper presents a "horizontal neuro-symbolic integration" approach for artificial general intelligence along with elementary representation-agnostic cognitive architecture and explores its usability under the experiential learning framework for reinforcement learning problem powered by "global feedback".

Keywords: artificial general intelligence, cognitive architecture, domain ontology, experiential learning, global feedback, local feedback, neuro-symbolic integration, operational space, reinforcement learning

1 Introduction

The current agenda of artificial general intelligence research is focused on neuro-symbolic architectures (NSA) [1, 2] with reinforcement learning (RL) capabilities [3]. There are studies concentrated on how the graph-based approaches can be combined with artificial neural networks and deep learning, particularly, deep reinforcement learning [4]. The latter involves variations end extensions of learning based on local feedback [5] such as Q-learning [6], involving incremental feedback or error propagation across the states of a studied behavioral program.

In this work we attempt to bridge the gap between symbolic and symbolic approaches within representation-agnostic cognitive architecture. This architecture is considered to be invariant to whether the operational space of an agent possessing it is represented by unstructured raw discrete data or a structured system of functions describing the state of an agent's environment.

From a practical standpoint, we anticipate that it might be plausible to build so called "horizontal neuro-symbolic integration" systems capable to perform both "System 1" ("slow") and "System 2" ("fast") thinking [7] – depending on the stage of a learning process and the explainability requirements for the system. In this work we suggest something rather different compared to the modern "vertical neuro-symbolic integration" systems with neural networks and knowledge graphs residing at different levels of cognitive architecture [8].

At the same time, we are considering the perspective of replacing the so-called "local feedback" (and local "error propagation") [9,10] with the so-called "global feedback" known in neuroscience and psychology [11,12,13] to see if it enables reinforcement learning and what are the conditions and circumstances that make it possible.

For the purpose of the above, we will first consider an overall approach to neuro-symbolic integration; next, describe our view on the global feedback, then draft principal architecture of an elementary module of a cognitive system and, finally, experimentally explore the reference implementation of the described architecture, discuss the results and draw some practical conclusions.

It worth noticing that the widely used RL term seem too much associated with only a limited scope of what can be called experiential learning (EL) [14] which involves any forms of learning, including unsupervised learning based on observing the states of the environment from an agent's standpoint, self-supervised learning [15] based on guidance and feedback provided by an agent to itself relying on different performance metrics possessed innately or inferred during the life-cycle, semi-supervised learning involving different forms of guidance given by an external agent being a teacher, and finally reinforcement learning per se – based on explicit feedback provided by a reinforcing instructor or an environment.

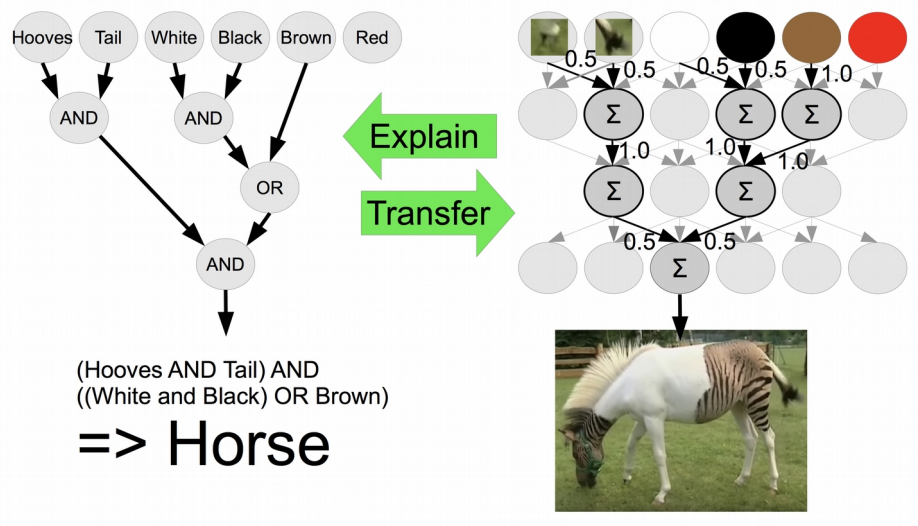

Fig. 1. An example of "horizontal neuro-symbolic integration". A symbolic representation of knowledge about the properties of a horse is presented on the left, with the semantic knowledge graph with labeled vertices performing either abstract concepts referring to specific properties and their values and edges performing as weighted links maintaining the probabilistic predicate logic structure. The sub-symbolic representation of the same knowledge is on the right, with the same knowledge stored in a distributed form as parameters connecting artificial neurons across layers in a deep neural network being an unlabeled weighted graph.

2 Neuro-symbolic integration

The “horizontal neuro-symbolic integration” framework concept is rendered on Fig.1. While the same knowledge may be represented in a form on the right or on the left, different phases of acquisition (learning) and execution (application) of knowledge may be performed using either of the two representations or both of them concurrently. The left (Fig.1) “symbolic” representation corresponds to Kahneman’s “System 2” of reasonable, explainable [16] and interpretable [17] “slow thinking” mode while the right “sub-symbolic” one corresponds to “System 1” associative and intuitive unexplainable “fast thinking” [7] modality.

The system implementing both of such knowledge representations would be able to learn acquiring new knowledge and perform applying this knowledge – using any of the two systems. “System 1” would be capable to learn more slowly but perform faster and “System 2” would be learning faster but acting more slowly [7].

At the same time, there would be possibility to “transfer” knowledge from the “interpretable” representation fast acquired earlier (“System 2”) into an “intuitive” representation to be applied fast when needed (“System 1”). In some cases, knowledge acquisition in the “symbolic” form may be inferred in the course of conventional probabilistic reasoning [18] and in some cases it can be obtained by symbolic input obtained from outer agents of external knowledge storage systems using a symbolic knowledge representation language such as “Aigents Language” [19].

The other way around, knowledge learned in the course of experiential learning by “System 1” during the training process could be “explained” being translated into a reasonable representation of “System 2” for either verification by means of probabilistic reasoning or communication of knowledge to external agents and knowledge storage systems via a symbolic language.

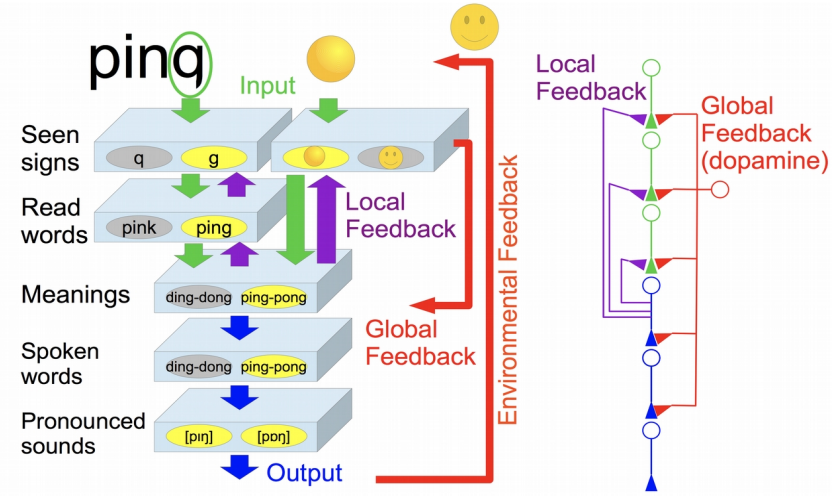

Fig. 2. Global feedback and local feedback loops in artificial cognitive architecture (left) and neuro-cortical architecture (right) with uncertain multi-modal perception and reinforcement.

3 Global feedback versus local feedback

The most of RL works referenced above [6,9,10] are focused on feedback propagation over a series of states probabilistically associated with eventual reinforcement. The reinforcing feedback is propagated step by step across the preceding behavioral trajectory which makes the latest steps collect more feedback even if they are irrelevant to delayed reinforcement. This can be called "local feedback" as it is propagated on a step-by-step basis so the reward of the next step is locally shared with the previous step. Also, this makes training longer because of slow incremental propagations of reinforcements.

We explore the alternative scheme of the "global feedback" [11,12,13] with full amount of reward shared evenly with all steps being in the attention focus at the time of reinforcement. Then the main problem becomes how to figure out the time span of the attention focus so it captures the complete sequence of steps leading to either reinforcement or failure. In the following experiments we were considering an event of either a positive or a negative stimulus to set a boundary of the attention interval. In turn, positive and negative stimuli were considered as a source of either positive (reinforcement or reward) or negative (punishment) feedback.

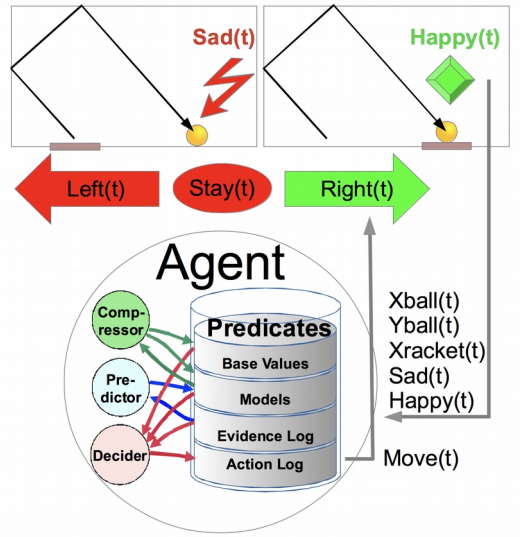

Fig. 3. Cognitive architecture and operational space for experiential learning in an arbitrary operational space represented by domain ontology – an example for a simplified "self-pong" game. Agent memories and cognitive processes – at the bottom. Sample operational space – at the top.

4 Cognitive architecture

Cognitive architecture of an elementary agent is inspired by the task-driven approach [18,20] implementing the theory of functional systems (TFS) of P. Anokhin. It is

shown on the bottom of Fig.3, where the agent possesses three processes acting upon four different memories, extending the cognitive model described in our earlier work [21].

The four types of memory are: a) “Base Values” or fundamental goals like avoidance of negative stimuli (“Sad”) and searching for positive ones (“Happy”); b) “Models” keeping probabilistic relationships between different state transitions experienced by an agent, with every state transition keeping the input environmental state and output action; c) “Evidence Log” of environmental states; d) “Action Log” of actions directed toward the environment.

Three types of processes are: 1) “Predictor” inferring the “Models” based on the “Evidence Log” and “Action Log” experiences, 2) “Decider” intended to make a choice relying on probabilities obtained based on the experience state and predictions evaluated by the “Predictor”; 3) “Compressor” which is supposed to keep the amount of stored memories in a reasonable range eliminating occasional and irrelevant models and logs to keep consumption of resources under control.

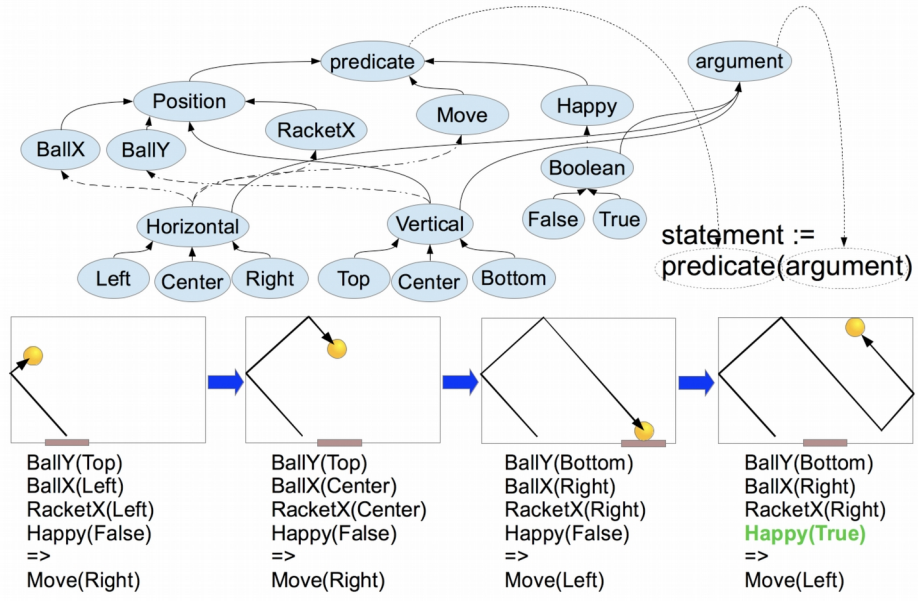

Fig. 4. Operational space – “functional”. Domain ontology – at the top. Representation of a sequence of states and actions by means of respective predicates – at the bottom.

An example of an operational space for an agent with such cognitive architecture is provided for a simplified “self-pong” game at the top of Fig. 3. The goal of a player in this game is to reflect the ball with the racket. The agent is provided a negative stimulus (“Sad”) if the ball hits the “floor”. The agent is given a positive stimulus (“Happy”) if the ball hits either the racket or the ceiling right after being reflected by

the racket successfully – depending on the game setup. Both stimuli may be considered as boolean predicates with time *t* as an argument. In turn, the action space of an agent is limited to the choice between moving the racket "Left" or "Right" or keeping it in place ("Stay"), which are other predicates with *t* as an argument as well. Other predicates of the environment can be coordinates of the ball ("Xball", "Yball") and the racket ("Yracket"), in case of the "functional" representation discussed further.

While the example above describes the operational space as a specific domain ontology including environmental variables (coordinates and stimuli) and agent actions, the cognitive architecture itself is assumed to be agnostic in respect to particular domain ontology as long as the ontology is described by any consistent set of predicates.

5 Operational spaces

An attempt has been made to evaluate possibility of experiential learning for the same physical problem applied to different operational spaces and corresponding domain ontologies. For this purpose, we have represented the above-described "self-pong" game using two completely different representations - "functional" and "discrete".

In the first case, illustrated on Fig. 4, we consider a "functional" operational space where behaviors of the ball and the racket are expected to be known and represented by distinct functions for different coordinates of the two. That could be a typical case for using a symbolic probabilistic reasoning system that operates conventional predicates describing the properties of identified concepts and objects and makes predictions on that basis.

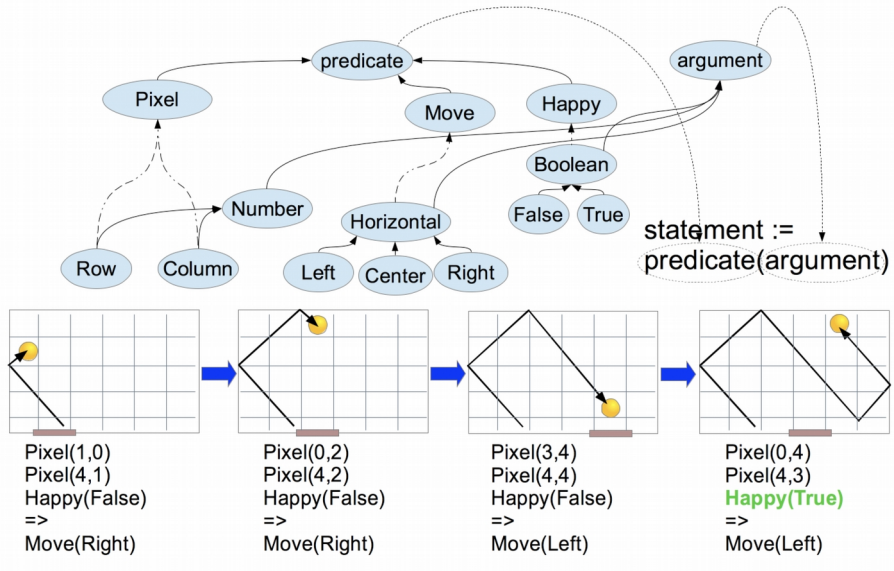

Fig. 5. Operational space – "discrete". Domain ontology – at the top. Representation of sequence of states and actions by means of respective predicates – at the bottom.

In the second case, illustrated on Fig. 5, we consider a “discrete” operational space where functional behaviors of the ball and the racket are expected to be unknown so everything is represented by “pixels” of a virtual display where any pixel could be corresponding to either the ball or the racket. That could be a typical case for using a neural network architecture with input from a raster display like Atari Breakout RL test from Open AI Gym framework resembling the “self-pong” discussed here [6].

6 Learning model

Three different learning models were explored during the following experiments.

First, “Sequential” - “symbolic” matching of the sequences of experienced states leading to positive or negative feedback since the last known feedback event. The model is represented by a set of successful sequences of states and actions ended up with either positive or negative feedback. Making a decision, the currently perceived sequence in the evidence log is used to find the nearest successful sequence in the model memory and apply it executing the corresponding action, or a random action is made if no match is found. The extended version of it called “SequentialAvoidance” is different so when no successful sequence is found and a random choice is being made, the unsuccessful sequences ended up with a negative feedback are discarded. Both versions may be extended with an option to make “fuzzy matching” so if no exact match for a successful sequence is found, the most similar one in the model memory is considered, based on the specified threshold in the range 0.0-1.0.

Second, “State-Action” model - the “sub-symbolic” one - was employed as a three-layer network connecting states to actions, with the input layer corresponding to the values of input predicates and a hidden layer representing compound states. The state-to-action connection weights were updated on every positive feedback event with positive correction. Optionally, if configured so, the weights could be updated also in case of any negative feedback with negative correction. Based on the “global feedback” principle, the state-action weights in a network were updated for every state and action pairs contained in the scope of the attention focus. The attention focus scope was being accumulated with every new state transition and reset upon any feedback arrival. When a decision was necessary, this model was operating in either a) the “non-fuzzy” mode when an action was selected only in case if the current state was perfectly predicting an action or b) the “fuzzy” mode when an action was selected only in case if it was predicted with certainty above the specified threshold in the range 0.0-1.0.

Third, the “Change-Action” model was a variant of the “State-Action” model where each state in a model was actually a “state transition” or a change between the previous and the current state, so the actions were associated not with states per se but with state transitions including the previous state and the current state.

7 Experimental results

All three learning models were applied to a simplified version of the Atari Breakout game [6] called "self-pong" as described above. The results are presented on Fig.6. The experiments were run for the same agent employing the same learning models with the inputs consisting of predicates representing environmental states in either a "functional" or a "discrete" operational space accordingly to the respective domain ontologies.

		Immediate feedback					Delayed feedback				
Environment	**Player Algorithm**	**2X4**	**4X6**	**6X8**	**8X10**	**Avg**	**2X4**	**4X6**	**6X8**	**8X10**	**Avg**
Functional	Sequential	89	88	88	92	89	70	73	72	85	75
Functional	SequentialAvoidance	92	90	90	93	91	67	73	81	85	77
Functional	SequentialAvoidance 0.5	**93**	**93**	**93**	93	93	80	83	81	89	83
Functional	State-Action	94	88	91	94	92	64	71	79	80	74
Functional	State-Action 0.5	93	88	87	93	90	64	68	75	83	73
Functional	Change-Action	91	86	89	92	90	64	73	76	79	73
Functional	Change-Action 0.5	93	90	90	93	92	63	69	80	84	74
Discrete	Sequential	89	88	88	92	89	70	73	72	85	75
Discrete	SequentialAvoidance	92	90	90	93	91	67	73	81	85	77
Discrete	SequentialAvoidance 0.5	93	91	88	92	91	70	76	80	83	77
Discrete	State-Action	94	88	91	94	92	64	71	79	80	74
Discrete	Change-Action	91	86	89	92	90	64	73	76	79	73

Fig. 6. Experimental results with columns: Environment: a "functional" or a "discrete" operational space and the respective domain ontology; Player Algorithm: a learning model, with 0.5 indicating fuzziness threshold. Numbers indicate the success rates (%) during the training period till the Agent is capable of playing without failures, so they correspond to the speed of learning. The "Avg" column indicates the average success rate across different game field sizes (2X4, 4X6, 6X8, 8X10) for each of the kinds of the feedback ("immediate" or "delayed").

All models were explored with different sizes of the game field (2X4, 4X6, 6X8, 8X10) under the conditions of experiencing negative and positive feedbacks. In the simplest case, the "Immediate feedback" was assumed so the positive stimulus ("Happy") was directed to the Agent by the environment at the point when a racket is successfully meeting the ball. In a more complex case of "Delayed feedback", the positive feedback was communicated only upon the ball hitting the ceiling being successfully reflected by a racket earlier.

Evaluation of the learning process success has been made based on success rate in percent during the training phase. The success rate was identified as the total number of positive feedbacks denominated by the sum of all positive and negative feedbacks. The training phase duration was selected as a number of epochs spent till an agent can play totally avoiding perception of negative feedback. The training phase duration was adjusted to be the same across all the learning models ("Player Algorithm" on Fig.3) for specific size of the game field and sort of feedback (immediate or delayed).

The code implementing the cognitive architecture, the models, the game environment and all of the experiments may be found on GitHub: https://github.com/aigents/aigents-java/tree/master/src/main/java/net/webstructor/agi .

A video featuring the process of learning can be watched on YouTube: https://www.youtube.com/watch?v=2LPLhJKh95g .

For all or the experimental conditions discussed above, the Agent was able to learn the game without failures, eventually. The presented approach has turned out to be practical in terms of shortening the learning times and implementing the “one-shot” learning concept. As it would be expected, expanding the game field and replacing immediate feedback with delayed feedback increased the learning times and decreased the success rates. The following conclusions were made.

1) Both “Functional” and “Discrete” representations of the environment are close to be equivalent from the accuracy (the learning speed on epochs) perspective.

2) Functional representation is much better from the run-time performance (response time and energy saving) perspective.

3) Both avoidance of negative feedback and fuzzy matching of experiences are helpful for increasing accuracy and learning speed.

4) Delayed reward decreases learning speed to the extent of ~10-15%.

5) Replacing explicit “symbolic” memories of successive behaviors with global feedback on combinations of “sub-symbolic” state-action contexts effects in: a) a dramatic increase in run-time performance, b) a minor decrease in learning speed.

6) Negative "global feedback" significantly worsens accuracy; learning may get impossible in some cases.

Still, the delayed reward problem is not solved in full, so an increase of the game field along with further delay of either positive or negative reinforcement was making it impossible to get reasonable learning results in the limited scope of this research. This is assumed to take place due to the inability to bound attention focus clearly so occasional positive feedbacks were allocated to multiple random state-action transitions loosely relevant to the eventual sparse feedback.

8 Conclusion

We have evaluated both “interpretable” functional representation and “non-interpretable” discrete representation of operational environment. We have done it using both “interpretable” symbolic representation and “non-interpretable” sub-symbolic versions of behavioral processes and their underlying models. Based on the study, we conclude that interpretable “one-shot” reinforcement learning is achievable to the same extent in all explored configurations and can be successfully done in both “interpretable” space and “non-interpretable” one. It has been found that acting within an “explainable” operational space saves memory and computing resources due to its more “structured” compact functional representation.

Converting a “non-explainable” discrete space to an “explainable” functional one, remains a challenge, however, which can potentially be solved with hybrid neuro-symbolic architectures. For this purpose, further studies on both “vertical” and “horizontal” neuro-symbolic integration architectures are necessary.

References

1. Tsamoura E. and Michael L.: Neural-Symbolic Integration: A Compositional Perspective. arXiv:2010.11926 [cs.AI] https://arxiv.org/abs/2010.11926 (2020)
2. Garcez A. et. al.: Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning. arXiv:1905.06088 [cs.AI] (2019)
3. Silver D. et. al.: Reward is enough. Artificial Intelligence Volume 299, October 2021, 103535 https://doi.org/10.1016/j.artint.2021.103535 (2021)
4. Francois-Lavet V. et. al.: An Introduction to Deep Reinforcement Learning. arXiv:1811.12560 [cs.LG] https://arxiv.org/abs/1811.12560 (2018)
5. Moreira I. et.al.: Deep Reinforcement Learning with Interactive Feedback in a Human-Robot Environment. arXiv:2007.03363 [cs.AI] https://arxiv.org/abs/2007.03363 (2020)
6. Mnih V. et. al.: Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs.LG] https://arxiv.org/abs/1312.5602 (2013)
7. Kahneman D. :Thinking, Fast and Slow. Farrar Straus & Giroux, January 1, 1994 (1994)
8. Marcus G.: The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. arXiv:2002.06177 [cs.AI] https://arxiv.org/abs/2002.06177v1 (2020)
9. Mostafa H. et. al.: Deep supervised learning using local errors. arXiv:1711.06756 [cs.NE] https://arxiv.org/abs/1711.06756 (2017)
10. Lindsey J. and Ashok Litwin-Kumar A.: Learning to Learn with Feedback and Local Plasticity. arXiv:2006.09549 [cs.NE] https://arxiv.org/abs/2006.09549 (2020)
11. Ciszak M. et. al.: Emergent excitability in adaptive networks of non-excitable units. arXiv:2010.06249 [nlin.AO] https://arxiv.org/abs/2010.06249 (2020)
12. Aljaberi S. et.al.: Global and local synaptic regulation determine the stability of homeostatic plasticity. arXiv:2103.15001 [nlin.AO] https://arxiv.org/abs/2103.15001 (2021)
13. Noh K. et. al.: Impaired coupling of local and global functional feedbacks underlies abnormal synchronization and negative symptoms of schizophrenia. PubMed, BMC Systems Biology 7(1):30, DOI: 10.1186/1752-0509-7-30, April 2013 (2013)
14. Kolb A. and Kolb D.: Experiential Learning Theory. In book: Encyclopedia of the Sciences of Learning, DOI: 10.1007/978-1-4419-1428-6_227, January 2012 (2012)
15. Zbontar J. et. al.: Barlow Twins: Self-Supervised Learning via Redundancy Reduction. arXiv:2103.03230 [cs.CV] https://arxiv.org/abs/2103.03230 (2021)
16. Arrieta A.B. et. al.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion Volume 58, June 2020, Pages 82-115 https://doi.org/10.1016/j.inffus.2019.12.012 (2019)
17. Rudin C. et. al.: Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. arXiv:2103.11251 [cs.LG] https://arxiv.org/abs/2103.11251 (2021)
18. Vityaev E.: Semantic Probabilistic Inference of Predictions. Series «Mathematics». 2017. Vol. 21 https://doi.org/10.26516/1997-7670.2017.21.33 (2017)
19. Kolonin A.: Controlled Language and Baby Turing Test for General Conversational Intelligence. arXiv:2005.09280 [cs.AI] https://arxiv.org/abs/2005.09280 (2020)
20. Vityaev E. et. al.: Logical Probabilistic Biologically Inspired Cognitive Architecture. Lecture Notes in Computer Science book series (LNCS, volume 12177) https://link.springer.-com/chapter/10.1007/978-3-030-52152-3_36 (2020)
21. Kolonin A.: Computable cognitive model based on social evidence and restricted by resources: Applications for personalized search and social media in multi-agent environments. 2015 International Conference on Biomedical Engineering and Computational Technologies https://ieeexplore.ieee.org/document/7361869?arnumber=7361869 (2015)