Analogy Making and Logical Inference on Images using Cellular Automata based Hyperdimensional Computing

Ozgur Yilmaz[*]
Department of Computer Engineering
Turgut Ozal University
Ankara Turkey
ozyilmaz@turgutozal.edu.tr

Abstract

In this paper, we introduce a framework of reservoir computing that is capable of both connectionist machine intelligence and symbolic computation. Cellular automaton is used as the reservoir of dynamical systems. A cellular automaton is a very sparsely connected network with logical nodes and nonlinear/logical connection functions, hence the proposed system corresponds to a binary valued and nonlinear neuro-symbolic architecture. Input is randomly projected onto the initial conditions of automaton cells and nonlinear computation is performed on the input via application of a rule in the automaton for a period of time. The evolution of the automaton creates a space-time volume of the automaton state space, and it is used as the reservoir. In addition to being used as the feature representation for pattern recognition, binary reservoir vectors can be combined using Boolean operations as in hyperdimensional computing, paving a direct way symbolic processing. To demonstrate the capability of the proposed system, we make analogies directly on image data by asking 'What is the Automobile of Air'?, and make logical inference using rules by asking 'Which object is the largest?'

1 Introduction

We have introduced a holistic intelligence framework capable of simultaneous pattern recognition Yilmaz (2015b) and symbolic computation Yilmaz (2015a,c). Cellular automaton is the main computational block that holds a distributed representation of high order input attribute statistics as in neural architectures (Figure 1 b). The proposed architecture is a cross fertilization of cellular automata, reservoir computing and hyperdimensional computing frameworks (Figure 1 a). The cellular automata (CA) computation can be viewed as a feedforward network with logical nodes and connections, as shown in Figure 1 c. In this paper, we analyze the symbolic computation capability of the system on making analogies and rule based logical inferences, directly on the image data. The results show that (Figure 2), binary vector representation of images derived through CA evolution provide very precise analogies and accurate rule based inference, even though very small number of examples are provided. In the next subsection we review cellular automata [1], then introduce relevant neuro-symbolic computation studies. Finally, we state our contribution.

[*] Web: ozguryilmazresearch.net

[1] The literature review is narrowed down in this paper due to space considerations. Please visit our published papers to get a wider view of our architecture among existing reservoir and hyperdimensional computing approaches.

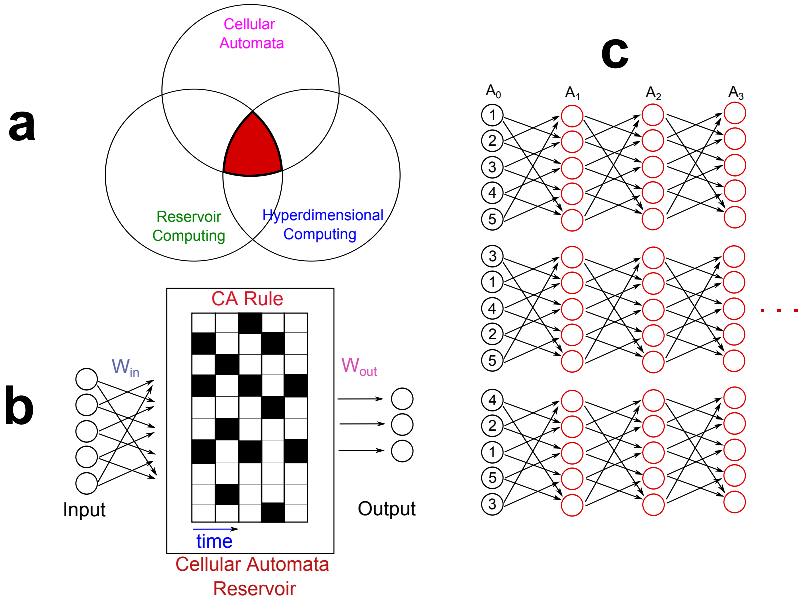

Figure 1: **a.** Our work is a cross fertilization of cellular automata, reservoir computing and hyperdimensional computing frameworks. **b.** In cellular automata reservoir, data is projected onto cellular automaton instead of a real valued node as in classical neural networks. **c.** The network formed by the cellular automaton feature space of rule 90. It can be viewed as a time unrolled feedforward network, however the connections are not all-to-all between layers due to partitioning of different permutations, given as separate rows. And the connections are not algebraic but logical, i.e. XOR operation. See Yilmaz (2015b) for details.

1.1 Cellular Automata

Cellular automaton is a discrete computational model consisting of a regular grid of cells, each in one of a finite number of states (Figure 1 c). The state of an individual cell evolves in time according to a fixed rule, depending on the current state and the state of its neighbors. The information presented as the initial states of a grid of cells is processed in the state transitions of cellular automaton and computation is typically very local. Essentially, a cellular automaton is a very sparsely connected network with logical nodes and nonlinear/logical connection functions (Figure 1 c). Some of the cellular automata rules are proven to be computationally universal, capable of simulating a Turing machine (Cook, 2004).

The rules of cellular automata are classified according to their behavior: attractor, oscillating, chaotic, and edge of chaos (Wolfram, 2002). Turing complete rules are generally associated with the last class (rule 110, Conway game of life). Lyapunov exponent of a cellular automaton can be computed and it is shown to be a good indicator of the computational power of the automata (Baetens & De Baets, 2010). A spectrum of Lyapunov exponent values can be achieved using different cellular automata rules. Therefore, a dynamical system with specific memory capacity (i.e. Lyapunov exponent value) can be constructed by using a corresponding cellular automaton. The time evolution of the cellular automata has very rich computational representation Mitchell et al. (1996), especially for the edge of chaos dynamics. The proposed algorithm in this paper exploits the entire time evolution of the CA and uses the states as the reservoir LukošEvičIus & Jaeger (2009); Maass et al. (2002) of nonlinear computation.

1.2 Symbolic Computation on Neural Representations

Uniting the expressive power of mathematical logic and pattern recognition capability of distributed representations (eg. neural networks) has been an open question for decades although several successful theories have been proposed (Garcez et al., 2012; Bader et al., 2008; Marcus, 2003; Miikkulainen et al., 2006; Besold et al., 2014; Pollack, 1990). The difficulty arises due to the very different mathematical nature of logical reasoning and dynamical systems theory. Along with many other researchers, we conjecture that combining connectionist and symbolic processing requires commonalizing the representation of data and knowledge.

Along the same vein, (Kanerva, 2009) introduced hyperdimensional computing that utilizes high-dimensional random binary vectors for representing objects, predicates and rules for symbolic manipulation and inference. The general family of the methods is called 'reduced representations' or 'vector symbolic architectures', and detailed introductions can be found in (Plate, 2003; Levy & Gayler, 2008). In this approach, high dimensionality and randomness enable binding and grouping operations that are essential for one shot learning, analogy-making, hierarchical concept building and rule based logical inference. Most recently (Gallant & Okaywe, 2013) introduced random matrices to this context and extended the binding and quoting operations. The two basic mathematical tools of reduced representations are vector addition and XOR.

In this paper, we borrow these tools of hyperdimensional computing framework, and build a semantically more meaningful representation by removing the randomness and replacing it with the cellular automata computation. This provides not only a more expressive symbolic computation architecture, but also enables pattern recognition capabilities otherwise not possible with random vectors.

1.3 Contributions

We provide a low computational complexity method Yilmaz (2015b) for recurrent computation using cellular automata based hyperdimensional computing. It is shown that the framework has a great potential for symbolic processing such that the cellular automata feature space can directly be combined by Boolean operations as in hyperdimensional computing, hence they can represent concepts and form a hierarchy of semantic interpretations. We demonstrate this capability by making analogies directly on images and infer relationships using logical rules. In the next section, we give the details of the algorithm, and then provide results experiments that demonstrate our contributions.

2 Cellular Automata Feature Expansion

In our reservoir computing method, data are passed on a cellular automaton instead of an echo state network and the nonlinear dynamics of cellular automaton provide the necessary projection of the input data onto an expressive and discriminative space. Compared to classical neuron-based reservoir computing, the reservoir design is trivial: cellular automaton rule selection. Utilization of edge of chaos automaton rules ensures Turing-complete computation in the reservoir, which is not guaranteed in classical reservoir computing approaches.

The reservoir computing system receives the input data. First, the encoding stage translates the input into the initial states of a 1D elementary cellular automaton. For binary input data, each feature dimension can randomly be mapped onto the cells of the cellular automaton. For this type of mapping, the size of the CA should follow the input data's feature dimension. After encoding, suppose that the cellular automaton is initialized with the vector ${A_0}^{P_1}$, in which P_1 corresponds to a random permutation of raw input data. Then, cellular automata evolution is computed using a prespecified rule, Z (1D elementary CA rule), for a fixed period of iterations (I):

$$\begin{aligned}
{A_1}^{P_1} &= Z({A_0}^{P_1}),\\
{A_2}^{P_1} &= Z({A_1}^{P_1}),\\
&\dots\\
{A_I}^{P_1} &= Z({A_{I-1}}^{P_1}).
\end{aligned}$$

The evolution of the cellular automaton is recorded such that, at each time step a snapshot of the whole states in the cellular automaton is vectorized and concatenated. Therfore, we concatenate the evolution of cellular automata to obtain a reservoir for a single permutation:

$$A^{P_1} = [{A_0}^{P_1}; {A_1}^{P_1}; {A_2}^{P_1}; ...{A_I}^{P_1}]$$

It is experimentally observed that multiple random permutation mappings significantly improve accuracy. There are R number of different random mappings, i.e., separate CA reservoirs, and they are combined into a large reservoir feature vector:

$$\mathbf{A^R} = [A^{P_1}; A^{P_2}; A^{P_3}; ...A^{P_R}].$$

The computation in CA takes place when cell activities due to nonzero initial values (i.e., input) mix and interact. Both prolonged evolution duration (large I) and existence of different random mappings (large R) increase the probability of long-range interactions, hence improve computational power and enhance representation.

3 Symbolic Processing and Non-Random Hyperdimensional Computing

Hyperdimensional computing uses random very large-sized binary vectors to represent objects, concepts, and predicates. Then, appropriate binding and grouping operations are used to manipulate the vectors for hierarchical concept building, analogy-making, learning from a single example, etc., that are hallmarks of symbolic computation. The large size of the vector provides a vast space of random vectors, two of which are always nearly orthogonal. Yet, the code is robust against a distortion in the vector due to noise or imperfection in storage because after distortion it will still stay closer to the original vector than the other random vectors.

The grouping operation is normalized vector summation, and it enables forming sets of objects/concepts. Suppose we want to group two binary vectors, V_1 and V_2. We compute their elementwise sums, and the resultant vector contains 0, 1 and 2 entries. We normalize the vector by accepting the 0 entries as they are, transforming 2 entries into 1. Note that, these are consistent within the two initial vectors. Then, the inconsistent entries are randomly decided: 1 entries as transformed into 0 or 1. Many vectors can be combined iteratively or in a batch to form a grouped representation of the bundle. The resultant vector is similar to all the elements of the vector due to the fact that consistent entries are untouched. The elements of the set can be recovered from the reduced representation by probing with the closest item in the memory, and consecutive subtraction. Grouping is essential for defining 'a part of', 'contains' relationships. $+$ symbol will be used for normalized summation in the following arguments.

There are two binding operations: bitwise XOR (circled plus symbol, $\oplus$) and permutation [2]. Binding operation maps (randomizes) the vector to a completely different space, while preserving the distances between two vectors. As stated in Kanerva (2009), "...when a set of points is mapped by multiplying with the same vector, the distances are maintained, it is like moving a constellation of points bodily into a different (and indifferent) part of the space while maintaining the relations (distances) between them. Such mappings could play a role in high-level cognitive functions such as analogy and the grammatical use of language where the relations between objects is more important than the objects themselves.

A few representative examples to demonstrate the expressive power of hyperdimensional computing:

1. We can represent pairs of objects via multiplication. $O_{A,B} = A \oplus B$ where A and B are two object vectors.

2. A triplet is a relationship between two objects, defined by a predicate. This can similarly be formed by $T_{A,P,B} = A \oplus P \oplus B$. These types of triplet relationships are very successfully utilized for information extraction in large knowledge bases Dong et al. (2014).

3. A composite object can be built by binding with attribute representation and summation. For a composite object C,

$$C = X \oplus A_1 + Y \oplus A_2 + Z \oplus A_3,$$

where A_1, A_2 and A_3 are vectors for attributes and X, Y and Z are the values of the attributes for a specific composite object.

4. A value of an attribute for a composite object can be substituted by multiplication. Suppose we have assignment $X \oplus A_1$, then we can substitute A_1 with B_1 by, $(X \oplus A_1) \oplus (A_1 \oplus B_1) = X \oplus B_1$. It is equivalent to say that A_1 and B_1 are analogous. This property is essential for analogy making.

[2] Please see Kanerva (2009) for the details of permutation operation, as a way of doing multiplication.

5. We can define rules of inference by binding and summation operations. Suppose we have a rule stating that "If x is the mother of y and y is the father of z, then x is the grandmother of z" [3]. Define atomic relationships:

$$
\begin{aligned}
M_{xy} &= M_1 \oplus X + M_2 \oplus Y, \\
F_{yz} &= F_1 \oplus Y + M_2 \oplus Z, \\
G_{xz} &= G_1 \oplus X + G_2 \oplus Z,
\end{aligned}
$$

then the rule is,

$$R_{xyz} = G_{xz} \oplus (M_{xy} + F_{yz}).$$

Given the knowledge base, "Anna is the mother of Bill" and "Bill is the father of Cid", we can infer grandmother relationship by applying the rule R_{xyz}:

$$
\begin{aligned}
M_{ab} &= M_1 \oplus A + M_2 \oplus B, \\
F_{bc} &= F_1 \oplus B + M_2 \oplus C, \\
G^{'}_{ac} &= R_{xyz} \oplus (M_{ab} + F_{bc}),
\end{aligned}
$$

where vector $G^{'}_{ac}$ is expected to be very similar to G_{ac}, which says "Anna is the grandmother of Cid". Please note that the if-then rules represented by hyperdimensional computing can only be if-and-only-if logical statements because operations used to represent the rules are symmetric.

Without losing the expressive power of classical hyperdimensional computing, we are introducing cellular automata to the framework. In our approach, we use binary cellular automata reservoir vector as the representation of objects and predicates instead of random vectors to be used for symbolic computation. There are two major advantages of this approach over random binary vector generation:

1. Reservoir vector enables connectionist pattern recognition and statistical machine learning (as demonstrated in Yilmaz (2015b)) while random vectors are mainly tailored for symbolic computation.

2. The composition and modification of objects can be achieved in a semantically more meaningful way. The semantic similarity of the two data instances can be preserved in the reservoir hyperdimensional vector representation, but there is no straightforward mechanism for this in the classical hyperdimensional computing framework.

4 Experiments on Analogy Making

In order to demonstrate the power of enabled logical operation, we will use analogy making. Analogy making is crucial for generalization of what is already learned. We tested the capability of our symbolic system using images. The example given here follows "What is the Dollar of Mexico?" in Kanerva (2009). However in the original example, sensory data (i.e. image) is not considered because there is no straightforward way to introduce sensory data into the hyperdimensional computing framework. The benefit of using non-random binary vectors is obvious in this context.

We have previously shown that binarization of the hidden layer activities of a feedforward network is not very detrimental for classification purposes Yilmaz et al. (2015). For an image, the binary representation of the first hidden layer activities holds an indicator for the existence of Gabor like corner and edge features. In order to test the symbolic computation performance of CA features on binarized hidden layer activities, we use CIFAR 10 dataset Krizhevsky & Hinton (2009). We used the first 500 training/test images and obtained single layer hidden neuron representation using the algorithm in Coates et al. (2011) (200 number of different receptive fields, receptive fields size of 6 pixels). The neural activities are binarized according to a threshold and, on average, 22 percent of the neurons fired with the selected threshold. After binarization of neural activities, CA features can be computed on the binary representation as explained in section 2. We formed a separate concept vector for each class (total 10 classes, 50 examples for each class) using binary neural representation of CIFAR training data and vector addition defined in Snaider (2012). These are the basis class concepts extracted from the visual database.

[3] The example is adapted from Kanerva (2009).

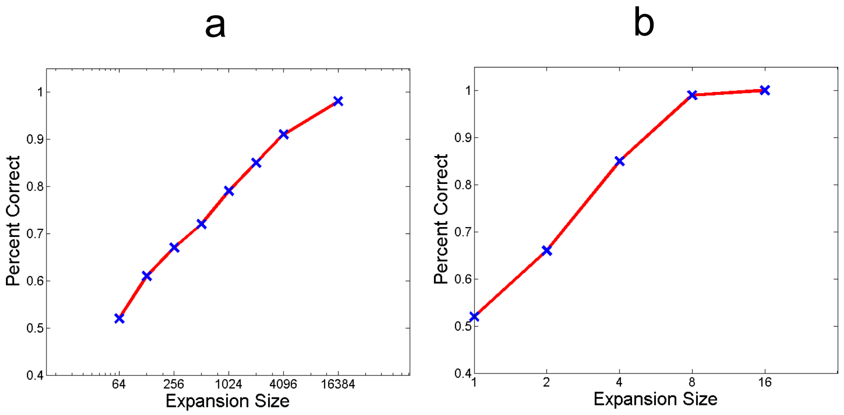

Figure 2: **a.** Analogy making experiment results. The feature expansion (defined as the product $R \times I$) due to cellular automata evolution is given in the x axis (log scale) and the percent correct is given in the y axis. **b.** Rule based logical inference experiment results.

We formed two new concepts called *Land* and *Air*:

$$Land = Animal \oplus Horse + Vehicle \oplus Automobile,$$
$$Air = Animal \oplus Bird + Vehicle \oplus Airplane.$$

In these two concepts, CA features of Horse and Bird images are used to bind with the Animal filler, and CA features of Automobile and Airplane images are used to bind with the Vehicle filler [4]. Animal and Vehicle fields are represented by two random vectors [5], those with the same size as the CA features. Multiplication is performed by xor ($\oplus$) operation and vector summation is again identical to Snaider (2012). The products, Land and Air are also CA feature vectors, and they represent the merged concept of observed animals and vehicles in Land and Air respectively. We can ask the analogical question "What is the Automobile of Air?", AoA in short. The answer can simply be given by this equality (inference):

$$AoA = Automobile \oplus Land \oplus Air.$$

AoA is a CA feature vector and expected to be very similar to Airplane concept vector. We tested the analogical accuracy using unseen Automobile test images (50 in total), computing their CA feature vectors followed by AoA inference, then finding the closest concept class vector to AoA vector (max inner product). It is expected to be the Airplane class. The result of this experiment is given in Figure 2 a [6] for various R and I combinations. The multiplication of the two defines the amount of feature expansion due to cellular automata state space. The analogy on CA features is 98 percent accurate (for both R and I equals 128), whereas if the binary hidden layer activity is used instead of CA features (corresponds to R and I equal to 1), the analogy is only 21 percent accurate. This result clearly demonstrates the benefit of CA feature expansion for symbolic computation.

The devised analogy implicitly assumes that Automobile concept is already encoded in the concept of Land. What if we ask "What is the Truck of Air?"? Even though Truck images are not used in building the Land concept, due to the similarity of Truck and Automobile concepts, we might still get good analogies. The results on this second order analogy is contrasted in Table 1. Automobile and Horse (i.e. "What is the Horse of Air?", the answer should be Bird.) are first order analogies and they result in comparably superior performance as expected, but second order analogy is much higher than chance level (i.e., 10 percent).

Please note that these analogies are performed strictly on the sensory data, i.e., images. Given an image, the system is able to retrieve a set of relevant images that are linked through a logical statement.

[4] There are 50 training images for each class. CA rule 110 is used for evolution. And mean of 20 Monte Carlo simulations is given to account for randomness in experiments

[5] Also 22 percent non-zero elements

[6] These are extended results for our previous publication Yilmaz (2015a). We were unable to test for large R and I values due to hardware limitations.

Automobile	**Horse**	Truck
79	68	52

Table 1: Analogy-making experiment results on CIFAR 10 dataset (subset). R and I are both 32. The accuracy of the analogy is given for first (given in bold) and second order analogies. See text for details.

A very small number of training data is used, yet we can infer conceptual relationships between images surprisingly accurately. However, again it should be emphasized that this is only a single experiment with a very limited analogical scope, and more experiments are needed to understand the limits of the proposed architecture.

It is possible to build much more complicated concepts using hierarchies. For example, Land and Air are types of environments and can be used as fillers in Environment field. Ontologies are helpful to narrow down the set of required concepts for attaining a satisfactory description of the world. Other modalities such as text data are also of great interest, (see Mikolov et al. (2013); Pennington et al. (2014) for state-of-the-art studies), as well as information fusion on multiple modalities (e.g., Image and text).

5 Experiments on Rule Based Inference

In order to test proposed architecture's capability for logical inference, we define a rule and form a knowledge base on image data. Then we make an inference on the knowledge base by applying the rule. The inference may or may not be right, hence the symbolic system is not completely sound. [7] For demonstration of logical inference on our system, we define size relationships among different objects using the following rule set.
Object in image X is larger than object in image Y:

$$L_{xy} = L_1 \oplus X + L_2 \oplus Y.$$

Object in image X is smaller than object in image Z:

$$S_{xz} = S_1 \oplus X + S_2 \oplus Z.$$

And finally, we state the largest object is in image Z:

$$T_z = T_1 \oplus Z.$$

Then the rule is stated as 'If object in X is larger than object in Y and smaller than object in Z, largest object is in Z'. The rule vector is computed as the manipulation of object vectors using hyperdimensional computing framework:

$$R_{xyz} = T_z \oplus (L_{xy} + S_{xz}).$$

Our knowledge base is again formed on the images of CIFAR 10 dataset. First, we use Truck, Automobile, and Airplane images (50 each) to compute X, Y and Z concept vectors (CA rule 110); then we obtain the rule vector R_{xyz} as explained above utilizing the concept vectors. In a completely different set of test image triplet, we make use of single Truck, Automobile, and Airplane images and compute their vector representation; a, b and c respectively. Knowledge base is created on the object vectors in three test images:

$$L_{ab} = L_1 \oplus a + L_2 \oplus b,$$
$$S_{ac} = F_1 \oplus a + M_2 \oplus c,$$

as 'object in image a is larger than object in image b, and object in image a is smaller than object in image c'. Can we infer the largest object? When we apply the rule vector on existing knowledge base, we get an estimate for the vector representation of the largest object:

$$T_{est} = R_{xyz} \oplus (L_{ab} + S_{ac}).$$

We compute the Hamming distance of T_{est} to existing object vectors (i.e. a, b and c), then it is possible to decide on the estimated largest object, i.e. closest vector to T_{est} which should be vector

[7] The completeness of the system requires a proof and it is a future work.

c. The average accuracy of 50 different test image triplets are shown in Figure 2 b. The chance level is 33 percent and it is observed that the binary neural representation (i.e., both R and I is equal to 1) is around 50 percent accurate, whereas cellular automata state space provides a 100 percent inference accuracy for a relatively small reservoir size. Please note that, similar to analogy making experiments logical inference is performed directly on image data and we can make object size inference using a very small number of example images (50).

6 Discussion

Along with the pattern recognition capabilities of cellular automata based reservoir computing Yilmaz (2015b), hyperdimensional computing framework enables symbolic processing. Due to the binary categorical indicator nature of the representation, the rules that make up the knowledge base and feature representation of the data that make up the statistical model live on the same space, which is essential for combining connectionist and symbolic capabilities. It is possible to make analogies, form hierarchies of concepts, and apply logical rules on the reservoir feature vectors [8].

To illustrate the logical query, we have shown the capability of the system to make analogies on image data. We asked the question ”What is the Automobile of Air?” after building Land and Air concepts based on the images of Horse, Automobile (Land), Bird and Airplane (Air). The correct answer is Airplane and the system infers this relationship with 98 percent accuracy, with only 50 training images per class. Additionally we have tested the performance of our architecture on rule based logical inference on images. We defined an object size related rule on image data, provided a knowledge base and inferred the largest object strictly using the image features.

Neural network data embeddings (eg. Kiros et al. (2014); Mikolov et al. (2013)) are an alternative to our approach, in which representation suitable for logical manipulation is learned from the data using gradient descent. Although these promising approaches are showing state-of-the-art results, they are bound to suffer from the dilemma of 'no free lunch' because the representation is data-specific. The other extreme is random embeddings adopted in hyperdimensional computing and reduced vector representation approaches. Although randomness maximizes the orthogonality of vectors and optimizes the effective usage of the space, it does not allow statistical machine learning or semantically meaningful modifications on existing vectors. Our approach lies in the middle: it does not create random vectors, thus can manipulate existing vectors and use machine learning, but do not learn the representation from the data therefore it is less prone to overfitting as well as to the dilemma of 'no free lunch'. Moreover, cellular automata reservoir is orders of magnitude faster than neural network counterparts Yilmaz (2015b).

7 Acknowledgments

This research is supported by The Scientific and Technological Research Council of Turkey (TUBİTAK) Career Grant, No: 114E554.

References

Bader, S., Hitzler, P., & Hölldobler, S. (2008). Connectionist model generation: A first-order approach. *Neurocomputing*, *71*, 2420–2432.

Baetens, J. M., & De Baets, B. (2010). Phenomenological study of irregular cellular automata based on lyapunov exponents and jacobians. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *20*, 033112.

Besold, T. R., Garcez, A. d., Kühnberger, K.-U., & Stewart, T. C. (2014). Neural-symbolic networks for cognitive capacities. *Biologically Inspired Cognitive Architectures*, (pp. iii–iv).

Coates, A., Ng, A. Y., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics* (pp. 215–223).

[8] Linear CA rules, such as rule 90 allow superposition of initial conditions. This property provides a symbolic system with much more powerful expressive capability Yilmaz (2015a).

Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, *15*, 1–40.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., & Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 601–610). ACM.

Gallant, S. I., & Okaywe, T. W. (2013). Representing objects, relations, and sequences. *Neural computation*, *25*, 2038–2078.

Garcez, A. S. d., Broda, K., & Gabbay, D. M. (2012). *Neural-symbolic learning systems: foundations and applications*. Springer Science & Business Media.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, *1*, 139–159.

Kiros, R., Salakhutdinov, R., & Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, .

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, .

Levy, S. D., & Gayler, R. (2008). Vector symbolic architectures: A new building material for artificial general intelligence. In *Proceedings of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference* (pp. 414–418). IOS Press.

LukošEvičIus, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*, 127–149.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, *14*, 2531–2560.

Marcus, G. F. (2003). *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Miikkulainen, R., Bednar, J. A., Choe, Y., & Sirosh, J. (2006). *Computational maps in the visual cortex*. Springer Science & Business Media.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Mitchell, M. et al. (1996). Computation in cellular automata: A selected review. *Nonstandard Computation*, (pp. 95–140).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, *12*, 1532–1543.

Plate, T. A. (2003). Holographic reduced representation: Distributed representation for cognitive structures, .

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*, 77–105.

Snaider, J. (2012). Integer sparse distributed memory and modular composite representation, .

Wolfram, S. (2002). *A new kind of science* volume 5. Wolfram media Champaign.

Yilmaz, O. (2015a). Symbolic Computation using Cellular Automata based Hyperdimensional Computing. *Neural Computation*, .

Yilmaz, O. (2015b). Machine Learning using Cellular Automata based Feature Expansion and Reservoir Computing. *Journal of Cellular Automata*, .

Yilmaz, O. (2015c). Connectionist-Symbolic Machine Intelligence using Cellular Automata based Reservoir-Hyperdimensional Computing. *arXiv preprint arXiv:1503.00851*, .

Yilmaz, O., Ozsarac, I., Gunay, O., & Ozkan, H. (2015). Cognitively inspired real-time vision core. *Technical Report*, . Available at `http://ozguryilmazresearch.net/Publications/NeuralNetworkVisionCore_YilmazEtAl2015.pdf`.