Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 123 (2018) 373–379

Procedia Computer Science

www.elsevier.com/locate/procedia

8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017

Molecular Associative Memory with Spatial Auto-logistic Model for Pattern Recall

Dharani Punithan[1] and Byoung-Tak Zhang[2]*

[1] Institute of Computer Technology, Seoul National University, South Korea
[2] Dept. of Computer Science and Engineering, Seoul National University, South Korea
punithan.dharani@gmail.com, btzhang@snu.ac.kr

Abstract

We propose a molecular associative memory model, by combining auto-logistic specifications, which capture statistical dependencies within the local neighborhood systems of the exposed knowledge, with the bio-inspired DNA-based molecular operations, which store and evolve the memory. Our model, characterized by only the local dependencies of the spatial binary data, allows to capture only a fewer features. Our memory model stores the exposed patterns and recalls the stored patterns through bio-inspired molecular operations. Our molecular memory simulation exemplifies the applications of associative memories in pattern storage and retrieval with high recall accuracy, even with lower order memory traces (pair-wise cliques) and thus exhibits brain-like content-addressing cognitive abilities.

Keywords: Auto-logistic models, Markov Random Field (MRF), second-order (8-point) neighborhood, pair-wise cliques, DNA-DNA hybridization, mutation, associative memory, recall

1 Introduction

Learning and memory are closely related critical concepts for understanding human intelligence and developing intelligent systems. Learning is a biological process of acquiring new knowledge, while memory is a process of encoding, storing and retrieving that acquired knowledge [7]. Associative memory or content-addressable memory (CAM) [8] is a function of brain that recalls (retrieves) previously stored data that closely matches the given partial cues. Associative memory is useful in applications requiring high-speed searches of large databases and search-intensive operations such as information retrieval, pattern matching, image processing, machine vision, etc [12]. In contrast to conventional location-based memory, associative memory requires a higher depth of processing (a huge amount of cognitive effort) to re-access the information. DNA (deoxyribonucleic acid) is found in every cellular organism as storage medium for genetic information, that defines the biological processes of life. The characteristics such as vast storage, massive parallelism and self-assembly are similar between brain [6, 18, 19] and DNA [1, 2, 16]. These properties are useful in realizing the brain-like associative

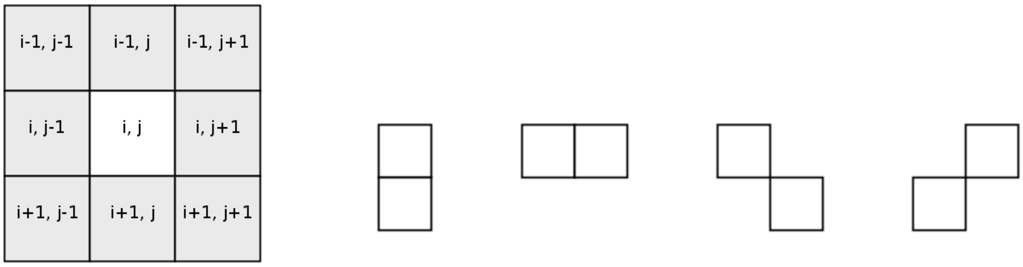
*Corresponding Author

10.1016/j.procs.2018.01.058

memories in molecular systems, which can be vaster than human brain [2, 15]. Recent works [11, 14] show that molecular systems can exhibit brain-like cognitive behaviors.

In this work, we demonstrate the simulation of molecular associative memory for pattern storage and recall. We combine the auto-logistic model [4, 5], a widely used model in image processing, with bio-inspired molecular operations such as hybridization and mutation to demonstrate associative memory for pattern recall. The auto-logistic model, first proposed by Besag in his seminal work [4], is a pair-wise interaction Markov random field (MRF) for binary (0-1) spatial data. It defines spatial dependency among random variables within the local neighborhood system. The proposed memory model learns the patterns (digits from 0 to 9) when exposed to MNIST [9] training examples. It defines only the local spatial features, specified by auto-logistic models, of the images, encodes them into DNA sequences and recodes to vector-based representation for computational efficiency and theoretical analysis. We then apply bio-inspired molecular operations (hybridization and mutation) to extract the information from examples and store the patterns with weights in the memory. The stored patterns are retrieved by the hybridization of the memory strands with the query strands. Our model requires only a few features, thus achieving lesser training time and lesser computation during the pattern recall. Our molecular learning algorithm is based on hypernetwork model [20] with the incorporation of auto-logistic specifications.

2 Background on Auto-logistic Model



(a) Second Order (8-point)

(b) Pair-wise Clique Types

Figure 1: Second-order Neighborhood System with Associated Pair-wise Clique Types

Consider a random field ($X = \{X_{ij}\}$), defined over a discrete finite two-dimensional lattice (L) of points. $\{X_{ij}\}$ represents random variables. A random field is a Markov random field [10] (MRF) if it satisfies the Markov property $p(x_{ij}|x_{kl}, (k,l) \in L, (k,l) \neq (i,j)) = p(x_{ij}|x_{kl}, (k,l) \in \mathcal{N}_{ij}); \forall (i,j) \in L$, where i and k are row indices, j and l are column indices, x_{ij} and x_{kl} are the realizations of the random variables, associated with the specified lattice points and $\mathcal{N}_{ij}$ is the neighborhood of (i,j).

Two widely used neighborhood systems are first-order and second-order neighborhood systems. A first-order neighborhood system (also called 4-point or von Neumann neighborhood) of any given lattice point (i,j) includes only orthogonal (right, left, up, and down) lattice points. A second-order neighborhood system (also called 8-point or Moore neighborhood) includes orthogonal and diagonal lattice points. In sub figure 1a, the shaded gray lattice points, in the second-order neighborhood system, indicate neighbors of the central lattice point (i,j). An MRF, characterized by the conditional distributions called the local characteristics of the random field, models the local interactions. Such spatial local dependencies can be defined by the clique potentials. A clique is either a single point or a subset of points in which each distinct pair of points are within the neighborhoods of each other. The pair-wise clique types in second-order neighborhood system are shown in sub figure 1b.

Auto-models [4, 5] are a subclass of MRFs; which are defined with only single (singleton) and pair-wise (doubletons) cliques. An auto-model is said to be an auto-logistic model if the random variables take binary values (0 and 1). The normalized conditional probability for the auto-logistic model, incorporating only single and pair-wise cliques, is given by the equation 1 [3]

$$p(x_{ij}|x_{kl}, kl \in \mathcal{N}_{ij}) = \frac{exp\{x_{ij}(\alpha_{ij} + \sum_{kl \in \mathcal{N}_{ij}} \beta_{ij,kl} x_{kl})\}}{1 + exp\{\alpha_{ij} + \sum_{kl \in \mathcal{N}_{ij}} \beta_{ij,kl} x_{kl}\}} \tag{1}$$

where α_{ij} is the local parameter and β_{ij} represents the pair-wise interaction strength between two neighboring lattice points (i, j) and (k, l). Owing to this simple form, auto-logistic models involve relatively low computational cost in model construction. When the pair-wise cliques are restricted to the first-order neighborhood system (the nearest neighborhood), the auto-logistic model is reduced to the popular Ising model [10,13]. When we model a digital image, the lattice is interpreted as a regular $2D$-lattice of pixels and the states of random variables as shades of the pixel color; black (0) and white (1) in case of a binary image (binary spatial data).

3 Molecular Associative Memory Model

DNA is made up of four nitrogen bases : Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). A single-stranded DNA sequence is formed when the bases are connected together with phosphodiester bonds. Two single strands chemically bind to form a double-stranded DNA helix by Watson-Crick complementarity [17] : $\overline{A} \equiv T$, $\overline{T} \equiv A$, $\overline{C} \equiv G$, and $\overline{G} \equiv C$. This base-pairing bio-operation is known as hybridization (or annealing). The reverse process, a double-stranded helix yielding its two constituent single strands, is called melting. We apply these operations to implement our memory model.

In our model, we first construct an initial memory as a set of m two dimensional lattices, each of size $N \times N$, representing m patterns (digits from 0 to 9). The lattice length N is 28; as each MNIST image is of size 28 × 28. Each lattice point corresponds to a pixel (x_{ij}) in the image. Initially, all pixels in the memory are black. We encode each pixel with its location and color to DNA molecules. The molecules are formed from the four-letter DNA alphabet $\{A, T, G, C\}$. For example, in a DNA sequence 'GTGGTTA'; 'GTG' (first three bases) represents row index (i) of a pixel, 'GTT' (next three bases) represents column index (j) of that pixel and 'A' (last base) represents the color of that binary pixel. Then, we form the pair-wise cliques (GTGGTTA-GACGTTA) with respect to the second-order neighborhood system of each pixel. These cliques are considered as the memory traces (engrams).

We then recode the string-based DNA sequence into a $2 \times n$ matrix, where n is the number of bases of the DNA sequence. Each DNA base is recoded into a vector : A as $[1, 0]^T$, T as $[-1, 0]^T$, G as $[0, 1]^T$, and C as $[0, -1]^T$. The molecular memory is a bag of these DNA single-strands; each representing pixel information such as pixel location (row and column indices) and pixel color (black or white) with pair-wise cliques. This initial memory evolves and gets updated when exposed to MNIST examples. It learns and stores the foreground (white) pixels of the examples in the memory with weights. The weights are computed based on the auto-logistic model generated conditional probabilities. The stored patterns are retrieved on partial cues (corrupted or noisy query patterns) by associative recall. The learning, storing and retrieving of information are performed using the DNA-based bio-molecular operations.

3.1 Learning

We model each MNIST training image as a two-dimensional binary random field with second-order neighborhood system. We first binarize each gray-scale image and remove noise if present. Each pixel (location and color) in the image is encoded into DNA molecules. The DNA molecules, representing the pixel locations, are complementary to memory strands. We then form the pair-wise cliques with the second-order neighborhood system. We recode the molecules into respective vectors, as mentioned before. The DNA single-strands of memory are hybridized with the single-strands of the training images. If the hybridization is complete, the pair-wise clique potential (interaction weight; $\beta_{ij,kl}$) is assigned as 1; otherwise 0. On partial hybridization, we mutate the DNA base, representing the pixel color, of the memory strand at the mutation rate of 0.1. The conditional probability of a foreground pixel ($x_{ij}^m = 1$) given its neighborhood ($x_{\mathcal{N}_{ij}}^m$) at the pixel location (i, j) of the pattern (m) is defined, by the auto-logistic model [3], as in equation 2.

$$p(x_{ij}^m = 1 | x_{\mathcal{N}_{ij}}^m) = \frac{exp\{\alpha + \sum_{kl \in \mathcal{N}_{ij}} \beta_{ij,kl}^m x_{kl}^m\}}{1 + exp\{\alpha + \sum_{kl \in \mathcal{N}_{ij}} \beta_{ij,kl}^m x_{kl}^m\}} \tag{2}$$

We present the results (section 4) with the local parameter $\alpha = -1$. We use these, auto-logistic model generated, conditional probabilities (refer equation 2) to update the weights (w_{ij}^m) of the foreground pixels of a pattern (m) of the memory (refer equation 3).

$$w_{ij}^m(new) = w_{ij}^m(old) + \eta * p(x_{ij}^m = 1 | x_{\mathcal{N}_{ij}}^m) \tag{3}$$

where $\eta = 1/(1 + exp(-\gamma * (iterNum - stepSize)))$ is sigmoid decay learning rate, $\gamma = 0.01$ is decay rate, iterNum is the current training iteration number and stepSize $= 100$. After each step, we remove the set of DNA single-strands, representing the training image, exposed in that iteration; this is known as melting operation. With new examples, learning strengthens the weight of respective foreground pixels of the memory. After learning, we change all the foreground pixel, having smaller weights (< 0.002) in the memory to black. Finally, we normalize the weights such that the sum of all the weights of the foreground pixels is 1.

3.2 Recall

Our model recalls the closest pattern stored in the memory to a given query (corrupted or noisy) pattern by applying bio-molecular operations and computing pair-wise clique strengths. The pair-wise cliques of DNA sequences (complementary strands) at each foreground pixel of the query pattern are formed and are allowed to hybridize with single-strands of the memory. The weighted score (refer equation 4) is computed for each of the stored patterns (digits from 0 to 9) and the softmax (refer equation 5) of scores is computed to retrieve the closest stored memory pattern.

$$score^m = \sum_{x_{ij}^m = 1} w_{ij}^m * \left[\frac{\sum\limits_{kl \in \mathcal{N}_{ij}} \beta_{ij,kl}^m}{|C|} \right]; \quad m = 0, ..., 9. \tag{4}$$

where $|C|$ is the cardinality of the pair-wise clique set.

$$\sigma(score^m) = \frac{\exp(score^m)}{\sum\limits_{l=0}^{9} \exp(score^l)}; \quad m = 0, ..., 9. \tag{5}$$

4 Results

In this section, we demonstrate the basic functionalities of our molecular memory model – (1) learning of the input patterns by encoding and storing them, and (2) recalling of the learned patterns when queries (corrupted or noisy) are presented.

4.1 Learned and Stored Patterns

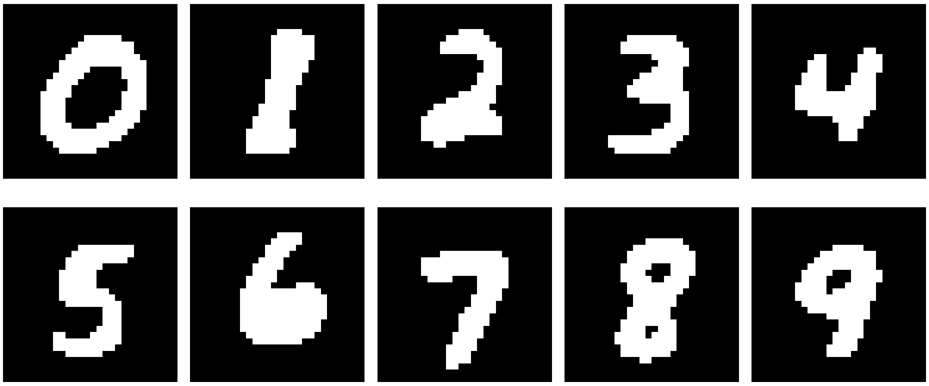


Figure 2: Patterns Stored in Molecular Associative Memory

The learned and stored patterns in the DNA-based memory, when exposed to the MNIST training dataset, are shown in figure 2. We use $50,000$ training examples; 5000 for each digit (from 0 to 9). This memory is constructed with the features incorporating only the pair-wise interactions, defined in the second-order neighborhood system and the auto-logistic model specifications. These patterns are learned from examples and stored in molecular memory by applying the DNA-based bio-molecular operations (hybridization and mutation).

4.2 Evaluation of Recall Task

We create two artificial datasets by (1) corrupting the parts of the stored patterns at various depths (from 0 to 10) and (2) adding random noise to the stored patterns at various noise levels (from 0 to 0.5). We use them for examining the recall task. The evaluation also includes the second best matches.

4.2.1 Corrupted Patterns

We create the corrupted datasets of stored patterns at different depths from 0 to 10. We choose a random foreground pixel and change it to black. Depending on the specified depth, we use depth first search algorithm and change all the foreground pixels in the respective neighborhoods to black. For each pattern (from 0 to 9) and at each depth (from 0 to 10), we create 1000 corrupted patterns. We use these $110,000$ corrupted patterns to evaluate the recall task of the memory. The average recall accuracies for corrupted patterns at various depths (from 0 to 10) are shown in figure 3. The result shows high accuracy (99.12%) even at the depth level of 10. The average recall accuracy at corruption depth of 10 for each pattern is shown in table 1.

4.2.2 Noisy Patterns

We create the noisy datasets of the stored patterns at different noise levels from 0 to 0.5. For each pattern at each noise level, we create 1000 noisy patterns and hence $60,000$ in total. The average recall accuracies for noisy patterns at various noise levels (from 0 to 0.5) are depicted in figure 4. The recall accuracies are high up to the noise level 0.3 (99.95%). The accuracies drop owing to the heavily

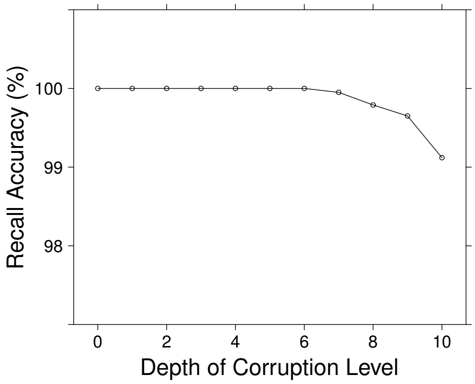


Figure 3: Recall Accuracy at Various Depth Levels

Patterns	Recalled ? yes	no	Acc. (%)
0	1000	0	100
1	991	9	99.1
2	1000	0	100
3	1000	0	100
4	1000	0	100
5	1000	0	100
6	1000	0	100
7	1000	0	100
8	1000	0	100
9	921	79	92.1
	Ave. Acc. (%) :		99.12

Table 1: Recall Accuracy at Depth Level 10

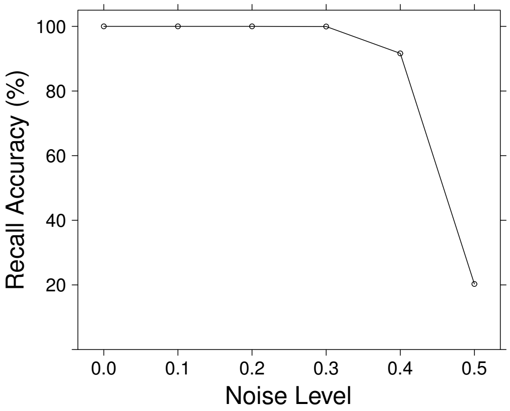


Figure 4: Recall Accuracy at Various Noise Levels

Patterns	Recalled ? yes	no	Acc. (%)
0	1000	0	100
1	998	2	99.8
2	1000	0	100
3	1000	0	100
4	1000	0	100
5	1000	0	100
6	999	1	99.9
7	1000	0	100
8	1000	0	100
9	998	2	99.8
	Ave. Acc. (%) :		99.95

Table 2: Recall Accuracy at Noise Level 0.3

distorted patterns over the noise level 0.3. The average recall accuracy at noise level of 0.3 for each pattern is shown in table 2.

5 Conclusion

The ability of DNA bio-molecules to store information, which can be manipulated by bio-inspired molecular operations such as hybridization and mutation, is used to implement the associative and recall mechanisms of memory. Our results demonstrate that biological DNA can be used as a cognitive learning tool to store and retrieve patterns. The patterns similar to the stored DNA molecules are recalled correctly, even with only pair-wise cliques in the local second-order neighborhood system which allow to capture only a fewer features. Our simulation of the molecular associative memory model, like human brain, is able to recall the patterns with high accuracy, when presented with partial information.

6 Acknowledgments

The authors thank Hyo-Sun Chun, Christina Baek, JeHwan Ryu and Jiseob Kim for insightful discussions. This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1401-12. The Institute of Computer Technology (ICT) at Seoul National University provided research facilities for the study.

References

[1] L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[2] E. B. Baum. Building an associative memory vastly larger than the brain. *Science*, 268(5210):583–585, 1995.

[3] Martin Beckerman. *Adaptive cooperative systems*. John Wiley & Sons, Inc., 1997.

[4] J. E. Besag. Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(1):75–83, 1972.

[5] J. E. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

[6] D. B. Chklovskii, B. W. Mel, and K. Svoboda. Cortical rewiring and information storage. *Nature*, 431(7010):782–788, 2004.

[7] Eric R. Kandel, Yadin Dudai, and Mark R. Mayford. The molecular and systems biology of memory. *Cell*, 157(1):163–186, 2014.

[8] T. Kohonen. *Self-organization and associative memory*. Springer-Verlag New York, Inc., 3rd edition, 1989.

[9] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist`.

[10] Stan Z. Li. *Markov random field modeling in image analysis*. Springer Publishing Company, Inc., 3rd edition, 2009.

[11] Kevin MacVittie, Jan Halámek, Vladimir Privman, and Evgeny Katz. A bioinspired associative memory system based on enzymatic cascades. *Chemical Communications*, 49(62):6962–6964, 2013.

[12] Behrooz Parhami. Associative memories and processors: an overview and selected bibliography. *Proceedings of the IEEE*, 61(6):722–730, 1973.

[13] Dharani Punithan and Byoung-Tak Zhang. Ising model based molecular associative memory for pattern recall. In *Proceedings of Korea Computer Congress*, pages 941–942. Korean Institute of Information Scientists and Engineers (KIISE), 2017.

[14] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with dna strand displacement cascades. *Nature*, 475(7356):368–372, 2011.

[15] John H. Reif, Thomas H. LaBean, Michael Pirrung, Vipul S. Rana, Bo Guo, Carl Kingsford, and Gene S. Wickham. Experimental construction of very large scale dna databases with associative search capability. In *International Workshop on DNA-Based Computers*, pages 231–247. Springer, 2001.

[16] Paul W. K Rothemund, Nick Papadakis, and Erik Winfree. Algorithmic self-assembly of dna sierpinski triangles. *PLoS biology*, 2(12):e424, 2004.

[17] J. D. Watson and F. H. C. Crick. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acids. *Nature*, 171:737–738, 1953.

[18] George M. Whitesides and Bartosz Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[19] Eran Zaidel and Marco Iacoboni. *The parallel brain: the cognitive neuroscience of the corpus callosum*. MIT press, 2003.

[20] Byoung-Tak Zhang. Hypernetworks: A molecular evolutionary architecture for cognitive learning and memory. *IEEE Computational Intelligence Magazine*, 3(3):49–63, 2008.