

Capturing the Dynamics of Cellular Automata, for the Generation of Synthetic Persian Music, Using Conditional Restricted Boltzmann Machines

Sahar Arshi¹[s.arshi@2014.hull.ac.uk] and Darryl N. Davis¹[d.n.davis@hull.ac.uk]

¹ School of Engineering and Computer Science,
University of Hull, UK, HU6 7RX.

Abstract. In this paper the generative and feature extracting powers of the family of Boltzmann Machines are employed in an algorithmic music composition system. Liquid Persian Music (LPM) system is an audio generator using cellular automata progressions as a creative core source. LPM provides an infrastructure for creating novel Dastgāh-like Persian music. Pattern matching rules extract features from the cellular automata sequences and populate the parameters of a Persian musical instrument synthesizer [1]. Applying restricted Boltzmann machines, and conditional restricted Boltzmann machines as two family members of Boltzmann machines provide new ways for interpreting the patterns emanating from the cellular automata. Conditional restricted Boltzmann machines are particularly employed for capturing the dynamics of cellular automata.

Keywords: Computational Creativity, Dastgāh, Persian Music, Conditional Restricted Boltzmann Machines, Computer Music, Cellular Automata.

1 Introduction

Persian traditional music is often recognized and investigated in the framework of Dastgāh music [2]. Each Dastgāh consists of smaller musical pieces called Gushe which share some similarity in respect to musical intervals, frequency of appearing notes, and repeated melodic motives. Different Dastgāh systems share some similar traits, for example the quality of the beginning and ending pieces and the presence of pieces having special characteristics as well as their respective time intervals. Dastgāhs, by their nature, have their complexities, which makes the emergence of new musical pieces reliant on the nascence of genius master musicians, who are intimately familiar with the concept of Dastgāh. This fact has made the creation of new Dastgāhs nearly impossible, and the quantity of compositions in different Dastgāhs intricate. Yet, there are immense possibilities for new music to be created in this genre [3, 4]. Techniques and tools from artificial intelligence are able to assist in the navigation of music space for creating the possibility of novel music.

One of the targets in the computational creativity arena is the manifestation of new artefacts with the help of computational algorithms. It is obvious that a clear viewpoint about the quality of creativity and how it happens is required from the outset. One of the ways for defining creativity is the study of its appearance or expression in human being when an artefact is created. Boden [5], [6] identifies three

types of creativity in this respect: Combinational, Exploratory, and Transformational. In combinational creativity, previous ideas are populated and associated with each other in an artistic way. In exploratory creativity, a conceptual space is explored in the hope of finding novel forms; this navigation may result in transformational creativity by altering some of the involved dimensions of the elements and/or the conceptual space. Boden further describes creativity in two broad categories: Historical-Creativity, and Personal-Creativity with respect of their origin. H-creativity refers to a type of creativity that has never happened before within a civilization or society. P-creativity is allocated to a type of creativity that is new to the person who created it, though it has happened before throughout history. Observation of the requirements involved for the appearance of a creative act, provides guidelines in many respects for enabling the recreation process. Algorithmic music generation can be considered as a manifestation of computational creativity.

Algorithmic music composition have been performed for various purposes from generating musical motifs for inspiring musicians and usually in the form of computer aided algorithm composition software to more complex tasks as well as mechanization of music itself [7]. The later target is a hard core problem which needs the studying the creativity behavior in human. Likewise algorithmic composition can improve knowledge about how the creativity is performed by humans. Nevertheless various methodologies are applied for algorithmic composition; for example machine learning tools, knowledge based systems, evolutionary algorithms, and computational intelligence models [7]. Computational intelligence models as well as Cellular Automata (CA) are able to generate materials without contributing to human domain knowledge [4]. Therefore they have been in the attention of the computational creativity community. Noteworthy, hybrid methodologies which benefit from the characteristics of various tools are likely to be the subject of future research. In the current project the aim is to explore the possibility of creating new types of Persian music, by navigating the spectrum of various types of creativity by a hybrid approach consisting of CA, and a machine learning tool.

Liquid Persian Music (LPM) is a Cellular Automata based auditory system, which relies on pattern matching rules for extracting features from CA progressions, and feeding these to a Persian musical instrument synthesizer. Pattern matching rules extract features from consecutive CA progressions. The synthesizer accepts parameters such as the length of the musical instrument string, and the ADSR envelopes and produces the audio signal. More information about the software can be found in [8].

One of the aims in this paper is to elevate the pattern matching rules to the next level. For this work two major concepts have been taken into account. One of them refers to associative memories and their crystal like nature. Every time an associative memory system reaches its equilibrium stability, new associations are produced across their constituent elements. This in fact, determines their stochastic nature. The other concept refers to the nature of cellular automata, as stochastic, yet determinable systems which makes them storable.

Boltzmann machines are a type of associative memory, and in this paper two families of Boltzmann machines, Restricted Boltzmann Machines (RBM), and Conditional Restricted Boltzmann Machines (CRBM), are investigated. RBMs have been used for learning static data and producing new representations of data.

Conditional Restricted Boltzmann machines expand RBM to be capable of prediction by learning time-series data. Cellular automata progressions are stored inside RBMs, and CRBMs. In this paradigm the stochastic nature of Boltzmann machine family have been used to benefit the generation of musical forms in the LPM system.

The arrangement of the sections of this paper are as follows: Section two presents Liquid Persian Music system, as a test bed for creating Persian-like music. Section three is dedicated to cellular automata, and basins of attraction models for studying the dynamics of CA. Section four is dedicated to Boltzmann machines in general and RBM, and CRBM in detail. The architectures of these two members of family of Boltzmann machines are described which then form the basis of the current experiments. The design approach and experiment are presented in sections five, and six. The results are available in section seven. The paper concludes in section eight, and future possibilities for research are presented.

2 Liquid Persian Music Software & Previous Research Agenda

The fundamental idea of LPM, likewise its former version Liquid Brain Music [9, 10] is the exploration of artificial life systems in the production of audio. Outputs of such system in relation to different parameterizations, are conventionally referred to as voices in the series of LPM associated publications. Voices resemble musical motives in music terminology. In [4, 11], it has been inferred that there are $88^7 * 20^7 * t$ number of such voices, considering the number of 88 unique CA behaviors, 20 pattern matching rules, t number of CA progressions, and 7 as the count of the synthesizer parameters.

One of the present considerations have been towards sequencing those voices in a musical manner. In [3], the LPM voices have been examined both individually and in relation to each other to study their aesthetical aspects regarding Zipf's law [12]. This study have been performed, acquiring suitable aesthetical means for evolving them in a musical manner. A computational framework for creating Persian Dastgāh-like music have been proposed in [4], in which sequencing the voices in a competent manner have been considered as a search problem. The musical critiques have been determined to be Zipfian metrics, and random sequences of voices as genotypes. Fitness functions based on Zipfian metrics are computable aesthetical critiques. In a later experiment in [3], Zipfian metrics have been extracted from a traditional Persian music data base. These metrics have been employed to train a Support Vector Machine Regression (SVMR) model, against, features extracted from LPM random sequences. The features which produced overlap between Persian music and LPM sequences were kept while the other ones have been discarded. This has been done due to the fact that SVMR model would decide solely based on a few strong detaching feature categories. On this basis, SVMR model would have cast off most of the LPM sequences without any chances for guiding them to the next evolutionary phase. On the other hand, it has been demonstrated that only a few number of features out of about 400 Zipfian features were sufficient for the conducting task. More details on this experiment can be found in [3].

The major issue in these experiments was the appearance of vivid jumps between the musical motives in some of the sequences. The randomized sequences in the

initial population, left a dramatic effect throughout consequent generations which could not have been detected with the usage of Zipfian metrics solely. To counter this, there would be too many generations involved, and/or the fitness function possess sufficient amount of intelligence to unravel the competent individuals, and operators to erode the jumping effects and then evolve the pieces in a desired manner. Although fitness function based on Zipfian metrics are capable of discarding incompetent individuals and giving credits to those who have further similar Zipfian metrics to Persian music, the need for a more intelligent criticizer became self-evident. It should not be left unsaid that during algorithm execution some desirable pieces have been created which were detected while performing auditory investigations. Some of the LPM generated audio have also been evaluated in an auditory survey [13]; some examples are available in [8]. In the current paper, unlike the previous publications in this series [3, 4, 11], the aesthetical aspects of the produced audio are not directly and particularly the main focus. Instead the attention is shifted towards approaches for interpreting the patterns derived from cellular automata sequences and applying them in the problem of sequencing LPM voices. This kind of approach is not only considered as a generative model; multiplying the power of creativity of cellular automata; it also has the potential of maintaining and producing smoother transitions between LPM voices.

3 Cellular Automata

Cellular Automata are dynamical systems whose manifestation are assigned to Von Neumann back in 1940s [14, 15]. Later in 1960s CA have been investigated as dynamical systems [16]. In this viewpoint, CA are discrete dynamical systems consisting of identical elementary individuals with same set of defined behaviors, which account to the global dynamics of the whole system.

Elementary CA consists of an assembly of cells which are arranged in a sequence of one dimensional array. In this work, each of the cells take two binary states. The progression of CA in each time step t , emerges as a two-dimensional lattice. The state of a cell at a time step is determined by its state and the states of its adjacent neighbors at the previous time step. The neighborhood radius is defined by r , which accounts to 2^r specifications, resulting from permuting the states of a cell and its neighbors. Allocating these permutations to binary states produces 2^{2^r} CA rules [16]. Considering the states of the left, and right neighbors of a cell, the number of rules for an elementary CA becomes 256.

Wolfram recognized four major classes of behavior in CA, consisting of fixed, cyclic, chaotic, and complex behaviors [16]. CA have been in the attention of artists' community as well, for their emergent behavior. Various attempts have taken place for contributing CA both independently as an inspiring source for artists, or as a computational creative tool for generating artefacts. Some of these works can be traced back in the works of pioneers as well as Iannis Xenakis in the generation of music. who used CA for attaining the general structure of compositions [7, 19]. However, these kind of compositions were often accompanied by heavy editing. Some others inferred CA as a source of raw material, which could not have been highly relied on for music composition in an independent manner [20].

Beyls Cellular Automata Explorer is one of the early musical systems based on CA. He experimented with various CA configurations including the cell neighborhood from previous and future CA progressions, and various rule specification in the search for finding complex musical structures [21]. Millen developed the CAM system, which exploits two and three dimensional game of life cellular automata, and maps the data to musical space for producing melody [7], [21]. CAMUS and Chaosynth [22, 23] are two other well-known CA musical generators. CAMUS uses Game of Life together with Demon Cyclic Space to simulate the diffusing effect of musical patterns in time. CAMUS system later exploits Cartesian mapping to achieve triplet musical forms [23]. Chaosynth is a CA based audio system which relies on producing sound granules resulting from additive synthesis; the produced audio is not presently similar to music as we know, it rather resembles the nature sounds as well as the flow of water, or a cluster of amphibians singing. Chaosynth underlying system is based on the model of the chemical reaction of a catalyst which in the visual domain appears as hypnotic contraction and detention effects of chemicals [20].

3.1 The Dynamics of Cellular Automata

The dynamics of cellular automata can be better illustrated in structures known as basin of attraction [24–26]. In order to clarify this notation, the meaning of state space is briefly overviewed. A state space consists of all possible configurations or patterns. For instance a binary vector of size N has 2^N patterns, and a binary matrix of size 4×4 has 2^{16} different configurations. For a cellular automata rule number, this state space can be divided to sections with each section having a structure relating its consisting patterns. These structures are known as basins of attraction. They often consist of a central point or a collection of patterns arranged on a circular path. There might be branches with so many sub branches connecting to these figures. The links between patterns in a basin of attraction is formed according to the accessibility by previous observed patterns. These outer branches are pre-images of the inner ones.

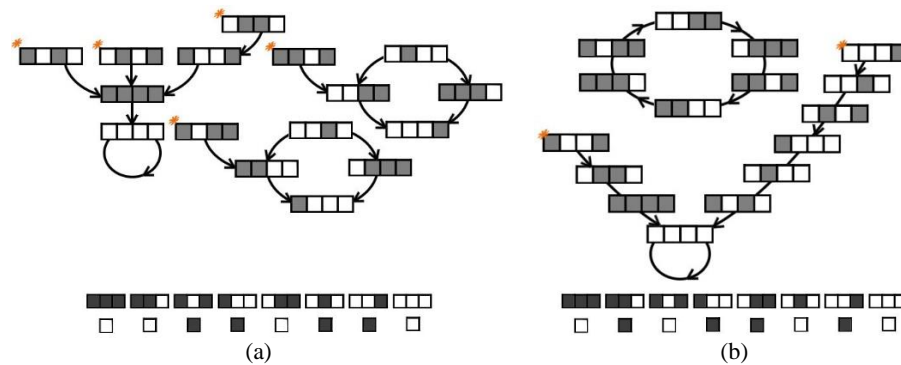


Fig. 1. Examples of Basin of Attraction for Cellular automata configuration for (a) rule 54, and (b) rule 90. The star signs shows the Garden of Eden states. These graphs are manually obtained, however, there are algorithms available for obtaining basin of attraction models for any CA rule space with different configurations [17].

The leaves are Garden of Eden states, which are not accessible by any previous states [24–26]. Figure 1 shows two examples of basin of attraction models for four-cell configurations for Wolfram rule numbers 54, and 90. Rule 54 produces 3 basins of attraction, while rule 90 produces 2 of them. Note that all different 16 configuration for 4 cells have been classified in the basin of attraction field model.

4 Restricted Boltzmann Machines

Boltzmann machines are stochastic neural networks invented by Hinton and Sejnowsky back in 1985 [27]. They can also be considered as Hopfield networks which replace the binary threshold rules associated to the units with stochastic decision rules [28]. Boltzmann machine dynamics revolve around an energy concept. Adhering to a fully connected architecture for Boltzmann machines make them inefficient for training. On the other hand, constraining the connections between the units will turn them into devices with tractable formulation for the required training procedures [29]. Restricted Boltzmann machines [30] are graphical models with undirected symmetrical connections which have only maintained their pair-wise connections between visible and hidden units. Figure 2-(a) shows the architecture of RBM.

RBM are capable of modelling the probability distribution of the input data. These new representations of the observable data, are accumulated in the intra-layer weights. Consequently, these networks are widely used in feature extraction, dimensionality reduction, and classification applications. Some examples of these kinds are available in [31, 32], [33]. For instance, once this network is trained, the features which are now embedded in weights in the bipartite graph can be used as data for further machine learning applications [34],[35]. Boltzmann machines have also been utilized in music generation applications in [36], more examples of such systems are provided in the CRBM section.

The behavior of the RBM system is directed by the energy function defined in formula 1. In which v , and h , stand for visible and hidden layer units in vectors, W is the weight matrix. a , and b , are bias vectors. On this account, the energy function in RBM is a sum of the linear products of the hidden vector, the connection weights matrix, and the input vector. The remaining terms in the following formulas stand for the products of the visible and hidden units and their contributing bias terms.

$$E(v, h) = -(h^T W v + a^T v + b^T h) \quad (1)$$

The probability of observation of a particular configuration over v visible input vectors and h hidden units are obtained by the notation 2, in which Z is the normalization term. The Z normalizing value is the result of calculating $e^{-E(v, h)}$ throughout all configurations for v , and h .

$$p(v, h) = e^{-E(v, h)} / Z \quad (2)$$

Obtaining Z (which is also known as the partition function) is nearly impossible, originating from the fact that so many binary units are involved. Inferring the joint probability distribution of v , and h can be performed by easier ways than computing the value of partition function. In order to track the value of the $p(v, h)$ a procedure is taken by marginalizing out the value of h from formula 2 [37].

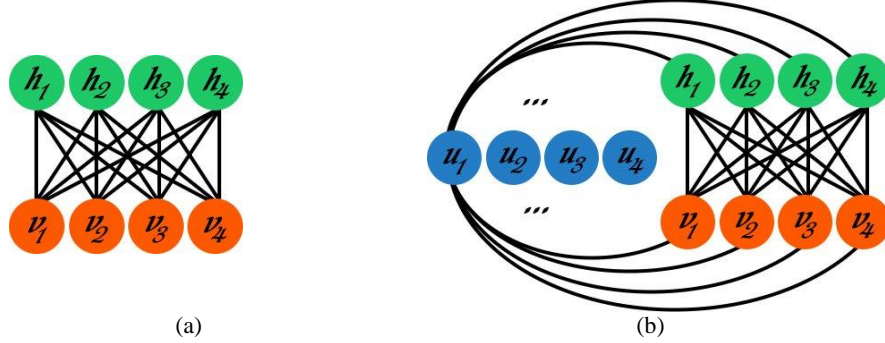


Fig. 2. The architectures of (a) Restricted Boltzmann machine, and (b) Conditional Restricted Boltzmann machine.

The marginal probability distribution of a visible vector equals the sum of the exponential of the energy function over all possible configurations as

$$p(v) = \sum_h e^{-E(v,h)/Z} \quad (3)$$

This formulation provides a basis for attaining the logarithm likelihood of $p(v)$ which its derivation can be traced back in [38]. The ultimate target would be to maximize $\log(p(v))$ for the training data in the visible units. This would become the foundation for performing gradient descent method for training RBMs by differentiating the negative of logarithm likelihood of $p(v)$. On this account the parameter update [34] for the weights can be obtained as:

$$\Delta w = \gamma(E_{P_{data}}(vh^T) - E_{P_{model}}(vh^T)) \quad (4)$$

Equation 4 consists of two terms. The first term $E_{P_{data}}(.)$ is the positive gradient, also known as the data dependent term which is the expectation over the data distribution. The second term $E_{P_{model}}(.)$ is the negative gradient and is the data independent term also presenting the expectation over the model distribution. Computing these expectations are intractable, however, they are estimable by performing alternating Gibbs sampling between hidden, and visible units. This procedure is followed in Contrastive Divergence algorithm.

4.1 Contrastive Divergence Algorithm

Contrastive Divergence algorithm [39, 40] is a sort of gradient descent method which is applied for training weights in RBMs.

The Contrastive Divergence algorithm works with the following procedure:

For each of training vectors the following steps are taken:

- The visible units are populated with a training vector.
- The probabilities of the units in the hidden layer are calculated and their states are determined by the following notation:

$$s(h_m) = \begin{cases} 1, & p(h_{m=1}|v) > rnd \\ -1, & otherwise \end{cases} \quad (5)$$

where rnd is a random number from uniform distribution in the range $[0,1]$.

- k -levels of Gibbs sampling takes place: the vector v' is reconstructed by sampling from the hidden units and the hidden units h' are resampled from v' .
- The weights and biases are updated considering the following learning rules, in which γ is a learning rate:

$$W^t = W^{t-1} + \gamma(h^t v^{tT} - h'^t x'^{tT}) \quad (6)$$

$$a^t = a^{t-1} + \gamma(v^t - v'^t) \quad (7)$$

$$b^t = b^{t-1} + \gamma(h^t - h'^t) \quad (8)$$

- The algorithm iterates, until a stopping criteria is visited (as well as the number of Gibbs sampling to be performed).

4.2 Conditional Restricted Boltzmann Machines

Conditional restricted Boltzmann machines are an extended version of RBM systems, which model the dependability of data to its previous time frames. An architecture of CRBMs are shown in figure 2-(b). They are best for capturing the dynamics of time-series data. CRBMs has had successful applications in modeling the dynamics of systems, as well as human motion [41, 42]. It is notable that there have been research in the area of music and CRBMs both for classification tasks, and for composition. CRBMs have been applied for the task of auto-tagging music in [43, 44]. This proposed model have been able to outperform some other machine learning classifiers, as well as support vector machines, and multi-layer perceptron. In [45] forward CRBMs have been employed to reconstruct music and improvise in the desired musical style, based on the provided first notes.

In CRBM model, the visible and hidden units are conditioned on n previous steps of visible states. Visible and hidden units receive this temporal information by established direct connection to an additional layer. Previous temporal information are provided in this new added layer. The topology of this model is illustrated in figure 2-(b). The energy function of the system is inferred as:

$$E(v, h, u) = -v^T W h - u^T U^{uv} v - u^T U^{uh} h - v^T a - h^T b \quad (9)$$

In which v, h, u correspondingly stand for observables, hidden neurons, and additional layer (associated to visible units in the previous time steps). Likewise, W are the weights between visible layer, and hidden layer, U^{uv} relates to the weights between visible layer and additional layer, and at last, U^{uh} is the weight matrix between conditional layer, and hidden units. a, b are bias matrixes. The weights, and biases in CRBM models can still be learned by the application of Contrastive Divergence algorithm.

5 Approach

The stochastic nature of Boltzmann machines proposes that there would be no unique equilibrium status for the system. This fact contributes to the generative nature of Boltzmann machine family models. This effect has been resembled to crystals which settle on their different facades (In fact Richards in [46] has first assimilated memories as crystals). A crystal with plasma like origin is gradually formed or

crystallized. Starting from an initial configuration for the elements of the plasma, every time the system is run, different associations between the elements are formed. This causes for attaining various crystal shapes. However, using the term of a solid crystal for BMs is a little bit tricky. A stable equilibrium of the system is where a local minimum energy is achieved in the energy hyper-surface. This does not necessarily mean that the state of neurons are not changing any longer. Actually the states of the neurons may still be oscillating, but that is happening around the local minima. Therefore the states of the neurons are not necessarily frozen.

Since the dynamics of CA is stochastic yet determinable, they can be treated as static memories. This relaxes the architecture of the LPM system from the requirement of navigations in the search spaces as discussed previously. This provides the ability to apply associative memory architectures. The dynamics of CA can be stored as memories. This scheme lets the individuals of an associative memory model be resembled to fossilized ants (as an example of individuals in a swarm) in a crystal. The associations of the ants would be different in every single crystallization process. Moreover, the angle of looking into the crystal changes the relations and the proportions in which they are perceived. This latter phenomenon is an assimilation to show the importance of the choice of collecting output from different hidden units in various extensions of Boltzmann machines. To be clearer, it should be noted that, although all the hidden units can be used for populating the synthesizer parameter, the audio result would change, considering which neuron is taken. This is how the generative powers of Boltzmann machine family can be manifested in LPM system.

6 Experiments

In this section the details of the performed experiments are presented; there are different perspectives towards applying CA to Boltzmann machine model family. In one of the approaches, 88 one-dimensional CA progressions are considered as static images with white and black cells to be applied as input vectors to the visible units. In this approach the number of cells in the visible layer would equal the total number of cells of a cellular automata progression. The same visible units are associated with the same cells in the CA progression over all the training samples. It is obvious that various possible initial seeds exist for the CA progression, which determine the emerging patterns in the upcoming generations. In fact they would end up in different basin of attractions for a CA rule. For this experiment, all the possible binary configurations for the initial seed have been taken into account for all the 88 unique CA behaviors.

In the second experiment, various configurations of CA progression are provided for a Conditional Restricted Boltzmann machine. This model have been inspired from basin of attraction models which are applied for studying the dynamics of CA. There are also some literature about neural networks which learn the dynamics of CA [47]. Each of the CA rules have their own basin of attraction fields regarding the configurations in which the basin of attraction is studied for that CA rule. Identifying each of those attractors and employing them as inputs to CRBMs, requires the consideration of various topologies for the input layer, and the connecting weights. Although this is not an impossible task, it is a labor intensive one. Therefore, another

tactic has been taken into account which is based on CA progressions history as training series. In this experiment all the transients starting from the initial seeds have been taken into account. Notably, all the possible initial seeds have been explored as well.

CA progressions can be trained to CRBMs for seizing their dynamics with various configurations. This work can be performed for each generation of CA, or even an arbitrary configuration from CA progression. For instance, one may wish to choose a complete number of cells from each CA generation, or partially select some of the cells. It is possible to include cells from more than one generation to form $p * q$ matrices of cells. The number of previous cells are also adjustable. For example, one may pick n number of $p * q$ previous configurations to constrain visible and hidden layers.

The number of the CA progressions to be trained by the CRBM system has been empirically determined by the behavior of CA rules themselves, whether they are in the classes of fixed, cyclic, chaotic, or complex. For instance, the number of the involved generations of CA rules in the fixed, and cyclic categories are noticeably less than the rules in the third, and fourth classes. These groups tend to achieve their stable behavior faster than the other groups, and would reach loopy behavior, and/or repetitions which would put more emphasis on these kinds of patterns and bring bias in the training set. Instead the possibility of having more generations out of the CA progressions have been allocated to chaotic, and complex behavior rule categories.

The experiments in this project are implemented in Matlab and with inspiration from the software and codes provided in [48],[49], and [50].

7 Results

In this section more details on the conducted experiments are provided, and the results are presented. For the first experiment with RBM, a number of 88 main CA behaviors over 24 generations have been taken into account. The initial seeds have been taken constant for all the CA rules. The training set consists of 88, $25*25$ metrics in 20 batches. This produces 625 visible units. The number of hidden neurons was selected to be 400 units. Overall number of 200 epochs were involved. Figure 3 demonstrates a portion of the connecting weights between visible, and hidden units. Each square presents one hidden unit. The square shapes are represented in this way to be analogous to the input metrics.

In the second experiment which is performed with CRBM, the CA progression have been divided into a series of $3*3$ metrics respecting their history of appearance. Each of these metrics have been conditioned to their three previous ancestors. This would result in having 9 visible units, and 27 past units. The count of hidden units are 400. Figure 4 demonstrates portions of weight metrics between visible-hidden, past-visible, and past-hidden. For this experiment all the 88 CA behaviors have been involved, and all the possible initial seeds have been allowed.



Fig. 3. Looking into the mind of RBM. Each of the small squares are associated to one of the units in the RBM architecture and all its connecting weights. These squares are a portion of 65 units out of the total hidden neurons.

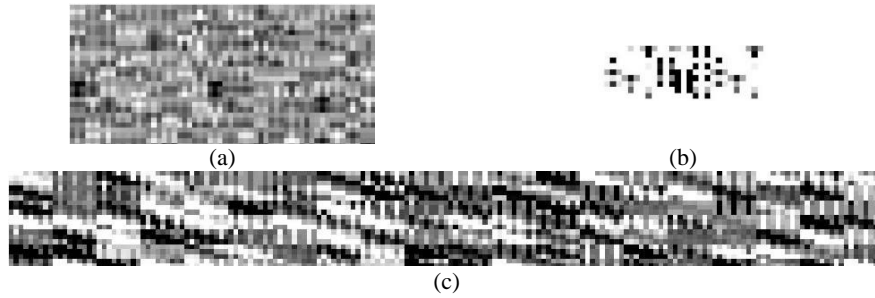


Fig. 4. Looking into the mind of CRBM. In the configuration of the CRBM for this experiment, Figure (a) relates to the weights between the visible and hidden units, figure (b) demonstrates the weights in between past and visible units, while figure (c) shows the weight values between past and hidden units.

So far the experiments suggested in this paper have resulted in producing patterns which can later be accompanied in music production. This means that new pattern matching rules are introduced to LPM system. The interesting thing about this architecture is the generation of various patterns by changing the initial configuration of CA, the initial seeds, and the number of involved hidden units which also contribute to the generative power of Boltzmann machine family. The outputs of the pattern matching rules in the original LPM system are employed for feeding the parameters of a Persian musical instrument synthesizer. The parameters vary from the physical parameters of a musical instrument as well as string length to ADSR envelopes of musical notes.

What have been discussed in this paper are some technical issues revolving around the sub system of a bigger LPM architecture for creating Persian-Dastgāh like music. The presentation of the whole system is out of the scope of the current paper. The mapping of the results of this research to musical space would be in association to other parts of the system and subject to future publications. The interested readers are welcome to follow the creations of the system as the research progresses in [8], related to LPM web page.

8 Conclusions and Future Work

In this article, RBM and CRBM have been applied for capturing the dynamics of cellular automata. It has been illustrated that these models have generative characteristics. In fact the usage of these two models have manifold targets in LPM system. The first aim is to enhance the pattern matching rules in LPM system and

taking them to another level. The other one is the exploitation of the generative, and stochastic nature of Boltzmann machine family themselves.

Elementary cellular automata have been used as a computational intelligence tool which provides raw material for creativity in LPM system. Complicating the initial configuration of CA, gives raise to the variety of the emerging patterns. Moreover, cellular automata are a sub type of Random Boolean Networks [51]. This fact expands the neighborhood configurations to multiply the level of complexity of the emerging patterns. It is obvious that taking account these types of configurations would increase the generality of the whole system, however, the integrity is suppressed by the current computational resources while the results of this stage are sufficient guidelines for entering the next phases of research.

Deep neural networks have been generating interest in the area of computational creativity due to the resemblance their underlying processes bear to the procedures taking place in human brain. In this work an infrastructure for employing deep architectures are employed which are built on Boltzmann machines. Extending the architecture of RBM into deep Boltzmann machine or a deep belief net [52] architecture by building further hidden layers would provide additional feature extractors or in another viewpoint higher level of features. This implies that further representations of data will be reflected in the top layers which can be derived correspondingly.

Other future steps of this research includes the design of the mapping mechanism to musical space of LPM voices while delving deeper into the sequencing problem. The produced mechanism is capable of being generalized to include various sources of raw materials for creativity rather than CA and different genres of music besides Persian music or any other creative artefacts.

References

1. Arshi, S.: An Implementation of Santur musical instrument and the Synthesis of music pieces using Learning Machines, Master Thesis, (2012).
2. Farhat, H.: The Dastgah Concept in Persian Music. Cambridge University Press (1990).
3. Arshi, S., Davis, D.N.: Generating Synthetic Persian Music. In: Chapter one in third International Conference on New Music Concepts (ICNMC 2016). pp. 11–43. ABEditore s.r.l, Academia Musicale, Milano.
4. Arshi, S., Davis, D.N.: A Computational Framework for Aesthetical Navigation in Musical Search Space. In: AISB symposium on computational creativity. , Sheffield, UK (2016).
5. Boden, M. a.: Computer models of creativity. *Psychologist*. 13, 72–76 (2000).
6. Boden, M.: Computer models of creativity. *Artif. Intell. Mag.* 30, 23–34 (2009).
7. Fernández, J.D., Vico, F.: AI methods in algorithmic composition: A comprehensive survey. *J. Artif. Intell. Res.* 48, 513–582 (2013).
8. Davis, D.N.: Computer and Artificial Music: Liquid Persian Music, <http://www2.dcs.hull.ac.uk/NEAT/dnd/music/lpm.html>.
9. Turner, C.: Liquid Brain Music, (2008).
10. Woods: Liquid Brain Music: Phase II. Available at: <http://www2.dcs.hull.ac.uk/NEAT/dnd/>. Computer Science, University of Hull (2009).
11. Arshi, S., Davis, D.N.: Towards a Fitness Function for Musicality using LPM. 6th York Dr.

Symp. (2015).

12. Manaris, B., Sessions, V., Wilkinson, J.: Searching for Beauty in Music. 1, 1–10 (2001).
13. Arshi, S.: Liquid Persian Music Survey, <https://www.surveymonkey.co.uk/r/QPQ77JB>.
14. Burks, A.W.: Von Neumann's Self-Reproducing Automata, (1970).
15. Burks, A.: Essays on Cellular Automata. University of Illinois Press (1970).
16. Wolfram, S.: A New Kind of Science. (2002).
17. Wuensche, A.: Discrete dynamics lab: Tools for investigating cellular automata and discrete dynamical networks. *Artif. Life Model. Softw.* (Second Ed. 215–258 (2009).
18. Li; Packard: The Structure of the Elementary Cellular Automata Rule Space. *Complex Syst.* 4, 281–297 (1990).
19. Hoffmann, P.: Towards an Automated Art: Algorithmic process in Xenakis' Composition. *Contemp. Music Rev.* 21, 121–131 (2002).
20. Miranda, E.R.: Cellular automata music: From sound synthesis to musical forms. *Evol. Comput. Music.* 170–193 (2007).
21. Burraston, D., Edmonds, E., Livingstone, D., Miranda, E.R.: Cellular Automata in MIDI based Computer Music. *Proc. Int. Comput. Music Conf.* 4, 71–78 (2004).
22. Miranda, E.R.: Evolving cellular automata music: From sound synthesis to composition. *Proc. Work. Artif. Life Model. Music. Appl.* 12 (2001).
23. Miranda, E.R.: Sounds of artificial life. In: *Proceedings of the 4th Conference on Creativity & and Cognition*, ACM. pp. 173–177 (2002).
24. Wuensche, A.: Basins of attraction in network dynamics: A Conceptual Framework for Biomolecular Networks. *Modul. Dev. Evol.* 288–314 (2004).
25. Wuensche, a: Classifying cellular automata automatically. *Complexity.* 4, 1–26 (1999).
26. Wuensche, A., Lesser, M.: *The Global Dynamics of Cellular Automata.* (1992).
27. Ackley, D., Hinton, G., Sejnowski, T.: A learning algorithm for boltzmann machines. *Cogn. Sci.* 9, 147–169 (1985).
28. Hinton, G.E.: Boltzmann Machines. Tutorial. 1–7 (2007).
29. Carreira-Perpiñán, M., Hinton, G.: On contrastive divergence learning. *Artif. Intell. Stat.* (2005).
30. Smolensky, P.: Information Processing in Dynamical systems: Foundations of Harmony Theory. In: Rumelhart, D.E. and McClelland, J.L. (eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations.* pp. 194–281. MIT Press (1986).
31. Larochelle, H., Bengio, Y.: Classification using discriminative restricted Boltzmann machines. *ICML*. 536–543 (2008).
32. Gehler, P. V., Holub, A.D., Welling, M.: The rate adapting poisson model for information retrieval and object recognition. *Proc. 23rd Int. Conf. Mach. Learn. - ICML '06.* 337–344 (2006).
33. Salakhutdinov, R., Hinton, G.: Replicated softmax: An undirected topic model. *Adv. Neural Inf. Process. Syst.* 22 - Proc. 2009 Conf. 1607–1614 (2009).
34. Salakhutdinov, R., Hinton, G.: An Efficient Learning Procedure for Deep Boltzmann Machines. *Neural Comput.* 24, 1967–2006 (2012).
35. Hinton, G.E., Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks. *Science* (80-.). 313, 504–507 (2006).
36. Lauly, S.: Music generation using Dynamically Linked Boltzmann Machines. 1–8 (2007).

37. Mnih, V., Larochelle, H., Hinton, G.: Conditional Restricted Boltzmann Machines for Structured Output Prediction. *Uai*. 514–522 (2011).
38. Hinton, G., Sejnowski, T.J.: Optimal perceptual inference. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. , Washington D. C. (1983).
39. Hinton, G.: A Practical Guide to Training Restricted Boltzmann Machines A Practical Guide to Training Restricted Boltzmann Machines. *Computer (Long. Beach. Calif)*. 9, 1 (2010).
40. Hinton, G.E.: Training Products of Experts by Minimizing Contrastive Divergence. *Neural Comput.* 14, 1771–1800 (2002).
41. Taylor, G.W., Hinton, G.E.: Factored conditional restricted Boltzmann Machines for modeling motion style. *Proc. 26th Int. Conf. Mach. Learn. (ICML 09)*. 1025–1032 (2009).
42. Taylor, G.W., School: Learning Representations of Sequences With Applications To Motion Capture and Video Analysis. *Cvpr*. (2012).
43. Mandel, M., Pascanu, R., Larochelle, H., Bengio, Y.: Autotagging music with conditional restricted Boltzmann machines. (2011).
44. Mandel, M., Eck, D., Bengio, Y.: Learning tags that vary within a song. *Proc. 11th Int. Conf. Music Inf. Retr.* 399–404 (2010).
45. Loeckx, A.J., Bultheel, J.: Forward Conditional Restricted Boltzmann Machines for the Generation of Music. In: *ICCC 2015 : 17th International Conference on Computational Creativity*. , Zurich, Switzerland (2015).
46. Richards, I.A.: *Principles of Literary Criticism* (Routledge Classics). Routledge (1924).
47. Wulff, N.H., Hertz, J. a: Learning Cellular Automation Dynamics with Neural Networks. *Adv. Neural Inf. Process. Syst.* 5. 631–638 (1993).
48. Tanaka, M.: Deep Neural Network, Mathworks, Matlab, <https://uk.mathworks.com/matlabcentral/fileexchange/42853-deep-neural-network>.
49. Salakhutdinov, R.R.: Learning Deep Boltzmann Machines Software, <http://www.cs.toronto.edu/~rsalakhu/code.html>.
50. Taylor, G.W.: Matlab implementation of Implicit mixtures of Conditional Restricted Boltzmann Machines, <https://github.com/gwtaylor/imCRBM>.
51. Kauffman, S. a: Homeostasis and differentiation in random Generic Control Networks. *Nature*. 224, 177–178 (1969).
52. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–54 (2006).