



Contents lists available at ScienceDirect

## Biologically Inspired Cognitive Architectures

journal homepage: [www.elsevier.com/locate/bica](http://www.elsevier.com/locate/bica)

Research article

## Biologically inspired cellular automata learning and prediction model for handwritten pattern recognition

Aamir Wali\*, Mehreen Saeed

Department of Computer Science, FAST-NUCES, Faisal Town, Lahore, Pakistan

## ARTICLE INFO

## Keywords:

Cellular automata  
CALP  
Conway's game of life  
Handwritten pattern recognition  
Data generation and over-sampling techniques  
Neural ensemble  
Ensemble methods

## ABSTRACT

In this study, we propose an ensemble learning architecture called “Cellular Automata Learning and Prediction” (CALP) model, for classification of handwritten patterns. We further propose that every handwritten pattern is an array of living cells or organisms that both interact and are affected by one another. Since the cells impact one another, and have the ability to die and reproduce, we extend this analogy to growth and evolution. Thus every pattern can grow and evolve. We use cellular automata (CA) to model this behavior as it has been used as a default model for various biological systems. Proposed architecture allows the handwritten patterns to evolve or grow using various parameters that control how the cells interact with each other. Then these different evolved patterns are used to train independent classifiers which are then combined together to form an ensemble. The idea is to capture more variations in a handwritten data set than the typical standalone classifiers or their ensembles. The method is applied on 5 handwritten data sets using 5 different classifiers. The experimental results show that our model obtains better classification accuracy on all 5 data sets, even on a small-sized training data. We also compare the performance of CALP with other over-sampling methods.

## 1. Introduction

The idea of cellular automata (CA) was originally conceived by Stanislaw Ulam and John von Neumann in 1940s. However, a more systematic study on CA was done much later in 1990 and then in 2002 by Wolfram in his well known book *A new kind of Science* (Wolfram, 2002). Later on, many interesting properties of CA were discovered, and they were applied in different areas of computer science such as image processing (Popovici & Popovici, 2002; Rosin, 2010), machine learning (Wongthanavas & Ponkaew, 2013), data mining (Fawcett, 2008), cryptography (Seredynski, Bouvry, & Zomaya, 2004), allocation hub location problem (Saghiri & Meybodi, 2018), etc. Recently, it has been shown that CA can be successfully applied to simulate cell and bacterial growth, and to model urban growth to predict the evolution of society and population (Abolhasani, Taleai, Karimi, & Rezaee Node, 2016; Berberoğlu, Akun, & Clarke, 2016). They have also been applied for generating test cases for software testing (Bhasin, 2014; Bhasin, Singla, & Sharma, 2013). This is because CA being a dynamic system that consists of cells, and a set of rules that explain how the cells evolve with time, gives them the ability to reproduce using self-generating patterns, and to model evolution and growth of a pattern.

In this paper, we first propose that a handwritten pattern is a collection of living cells, and then propose one such model that exploits the

important properties of interacting cells and the self generating characteristics of cells to evolve data. We also show how the fascinating abilities of cells modeled using CA can be amalgamated with handwritten pattern recognition system techniques to give rise to a methodology, which evolves the patterns using CA and use these evolved patterns for classification. The CA was used particularly because it is used in various biological systems (Akdur, 2011; Vitvitsky, 2016; Youssef, 2013).

Since our pattern can evolve and grow to new patterns, this is also related to synthetic data generation. The idea of synthetic pattern generation is not uncommon in the area of classification and pattern recognition. Synthetic data set generation, besides being used for the purpose of increasing number of training samples, is also used to balance imbalanced data sets (Charte, Rivera, del Jesus, & Herrera, 2015; Fernández-Navarro, Hervás-Martínez, & Gutiérrez, 2011; Sáez, Krawczyk, & Woźniak, 2016). Data generation is also of interest to the deep learning community. A major disadvantage of state-of-the-art convolutional neural network is the need for huge amount of data, whereas our proposed method is capable of performing well on small-sized data by simulating variations of existing data.

The difference between CA and other data generation techniques is that the other methods take a set of samples or data points and generate new samples within them using interpolation, nearest neighbor, or

\* Corresponding author.

E-mail addresses: [aamir.wali@nu.edu.pk](mailto:aamir.wali@nu.edu.pk) (A. Wali), [mehreen.saeed@nu.edu.pk](mailto:mehreen.saeed@nu.edu.pk) (M. Saeed).<https://doi.org/10.1016/j.bica.2018.04.001>Received 13 March 2018; Accepted 9 April 2018  
2212-683X/© 2018 Elsevier B.V. All rights reserved.

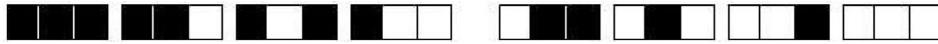


Fig. 1. 8 possible combinations of neighborhood in a 1D, 2-state and 3-site Cellular Automaton. Black cells represent a value of 1, and white represents a value of 0.

presents a value of 0.

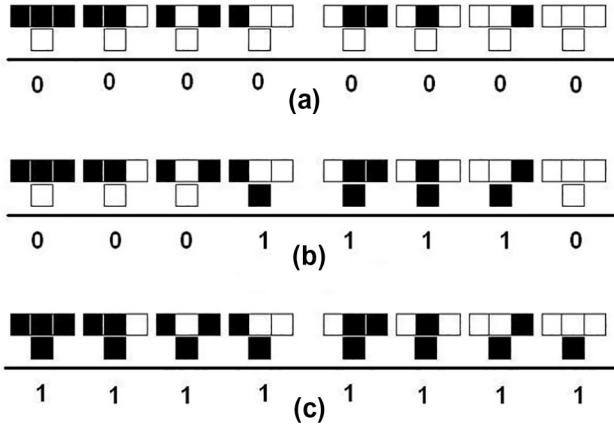


Fig. 2. Behavior of (a) Rule 0 (0000 0000 in binary), (b) Rule 30 (0001 1110 in binary), and (c) Rule 255 (1111 1111 in binary). Each part specifies a pair containing 8 neighborhoods (top), and immediately below them, their respective mapping to 0 or 1.

random re-sampling of the original data set with or without replacement. For example, SMOTE algorithm (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and its variations use nearest neighbors and interpolation, ADASYN (He, Bai, Garcia, & Li, 2008) also uses nearest neighbor with focus on only boundary samples, while techniques like OS (Zhou & Liu, 2006), and bootstrapping (Mooney, Duval, & Duvall, 1993) select random data points from the original data set. CA however, works differently. It takes each data point individually, and evolves it to a new form using the notion of self-generated patterns.

Furthermore, ensembles are used by the machine learning community to boost the performance of classifiers. An ensemble, which consists of a number of base classifiers, has shown its effectiveness in many applications such as handwritten digits classification (Zhang, Bui, & Suen, 2007), writer-independent off-line signature verification (Bertolini, Oliveira, Justino, & Sabourin, 2010), classifiers for decoding fMRI data (Cabral, Silveira, & Figueiredo, 2012), automatic music genre classification (Costa, Oliveira, Koerich, & Gouyon, 2013) and face recognition (Li, Shen, Shen, Yang, & Gao, 2016). Therefore, we also propose an ensemble model built on base classifiers that are trained on multiple evolved versions of the same data set.

In this paper, we propose a data evolution technique using CA and show how it can be combined with conventional machine learning classification models. The aim of this paper is four fold. First, to show that evolving the data set using a programmable CA adds more variations into the data set leading to improved training. Second, to demonstrate how various evolutions of a pattern can be combined to form an ensemble. Third, to show that such an ensemble has the ability to capture more variations in handwritten data or image data sets, and hence improves the overall classification accuracy as compared to using data synthesis methods such as SMOTE, ADASYN, etc. Finally, we illustrate that CALP requires very small training data, and yet performs better, or equally as compared to a single classifier trained on a larger one.

The rest of the paper is organized as follows. Section 2 provides a comprehensive background on CA. Section 3 gives an insight into the motivation behind our model, and gives a layout of our proposed scheme. Results and discussion are given in Section 4 followed by conclusions in Section 5.

## 2. Background

In this section, we familiarize the readers with CA and its various properties. We also discuss some previous methods that used CA as a classification model.

### 2.1. Cellular automata

Wolfram introduced the CA as a  $d$ -dimensional grid that consists of cells (Wolfram, 2002). Each cell can have a value of 1 or 0. A combination or configuration of cells is called a state. A state may evolve at every step resulting in the next generation or state. How a state evolves is determined by a definite rule. The rule specifies the new value of a cell as a function of the cell's previous value, and values of the neighboring cells, collectively called the neighborhood. The rule is sequentially applied on all the cells in the state.

A one-dimensional CA consist of an array of cells, each cell has a neighborhood so for example, for a 3-site automaton, neighborhood is defined by the current cell, and cells to its immediate left and right. This is also called an elementary CA. Mathematically, the overall model of a CA is given by:

$$a_i(t+1) = f(a_{i-1}(t), a_i(t), a_{i+1}(t)) \quad (1)$$

where  $a_i(t)$  is the value of cell  $i$  at time  $t$ . By extending the radius of the neighborhood, a  $w$ -site automaton can be developed.

A 3-site, one-dimensional automaton with binary values has 8 possible configurations of neighborhood or environment that a rule must handle. These are shown in Fig. 1. The white and black cells have a value of 0 and 1 respectively. In case of the first configuration (left-most), the value of the current cell is 1, and its left and right neighbors are also valued as 1. In the second configuration, the current cell and the left neighbor is 1, but the value of the right neighbor is 0, and so on.

In one-dimensional, 3-site CA with only two states, a rule must specify the mapping of each of these 8 possible configurations to a value of 0 or 1. Some examples are shown in Fig. 2. In each sub-figure, the 8 possible configurations of neighborhood are shown. Below each configuration, is the new output value that the center cell in the configuration will evolve to. The rule in Fig. 2(a) maps all configuration onto a value of 0. If we consider the 8 outputs of each configuration as an 8-bit binary number, then the rule in Fig. 2(a) maps to binary [00000000] which has a value of 0. This is why the same rule is called rule 0. This allows a total of 256 possible rules each of which can be indexed by an 8-bit binary number. Fig. 2(b) shows rules 30, while (c) shows rule 255 written as [11111111] in binary. Rule 255 maps all 8 configurations to 1 (Wolfram, 2002).

A cellular automaton can also be 2-Dimensional. In 2D CA, any cell has 8 immediate neighbors. Different rules may be defined on all 8 or fewer cells in the neighborhood. A cell with its 4 adjacent (non-diagonal) cells form Von Neumann neighborhood, and a cell along with its 8 neighboring cells form a Moore neighborhood (Wolfram, 2002).

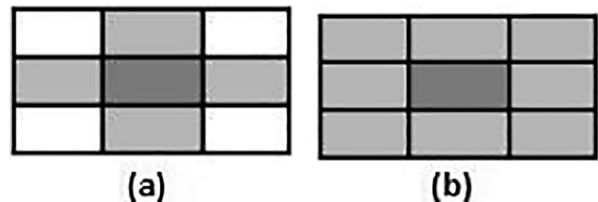


Fig. 3. (a) Von Neumann neighborhood, (b) Moore neighborhood for the center cell.

Download English Version:

<https://daneshyari.com/en/article/6853442>

Download Persian Version:

<https://daneshyari.com/article/6853442>

[Daneshyari.com](https://daneshyari.com)