A recurrent perceptron learning algorithm for cellular neural networks

Abstract A supervised learning algorithm for obtaining the template coefficients in completely stable Cellular Neural Networks (CNNs) is analysed in the paper. The considered algorithm resembles the well-known perceptron learning algorithm and hence called as Recurrent Perceptron Learning Algorithm (RPLA) when applied to a dynamical network. The RPLA learns pointwise defined algebraic mappings from initial-state and input spaces into steady-state output space; despite learning whole trajectories through desired equilibrium points. The RPLA has been used for training CNNs to perform some image processing tasks and found to be successful in binary image processing. The edge detection templates found by RPLA have performances comparable to those of Canny’s edge detector for binary images.

Key words Cellular neural networks · Learning · perceptron learning rule · Image processing

1 Introduction

A Cellular Neural Network (CNN) is a 2-dimensional array of cells (Chua and Yang 1988). Each cell is made up of a linear resistive summing input unit, and R-C linear dynamical unit, and a 3-region, symmetrical, piecewise-linear resistive output unit. The cells in a CNN are connected only to the cells in their nearest neighborhood defined by the following metric:

\[ d(i, j; \delta_{ij}) = \max \{|i - i|, |j - j|\} \]

where \((i, j)\) is the vector of integers indexing the cell \(C(i, j)\) in the \(i\)th row, \(j\)th column of the 2-dimensional array. The system of equations describing a CNN with the neighborhood size of 1 is given in Eqs. 1–2.

\[
\begin{aligned}
\dot{x}_{i,j} &= -A \cdot x_{i,j} + \sum_{k, l \in \{-1, 0, 1\}} w_{k,l} \cdot y_{i+k,j+l} \\
&+ \sum_{k, l \in \{-1, 0, 1\}} z_{k,l} \cdot u_{i+k,j+l} + I \\
y_{i,j} &= f(x_{i,j}) = \frac{1}{2} \cdot \left( \left| x_{i,j} + 1 \right| - \left| x_{i,j} - 1 \right| \right),
\end{aligned}
\]

where, \(A, I, w_{k,l}\) and \(z_{k,l} \in \mathbb{R}\) are constant parameters. \(x_{i,j}(t) \in \mathbb{R}, y_{i,j}(t) \in [-1, 1]\), and \(u_{i,j} \in [-1, 1]\) respectively denotes the state, output, and (time-invariant) external input associated to a cell \(C(i, j)\).

It is known in Chua and Yang (1988) that a CNN is completely stable if the feedback connection weights \(w_{k,l}\) are symmetric. Throughout the paper, the input connection weights \(z_{k,l}\) are chosen to be symmetric for reducing computational costs while the feedback connection weights \(w_{k,l}\) are chosen symmetrically for ensuring complete stability, i.e., \(w_{-1,-1} = w_{1,1} := a_1, w_{-1,0} = w_{1,0} := a_2, w_{-1,-1} = w_{1,-1} := a_3, w_{0,-1} = w_{0,1} := a_4, w_{0,0} := a_5\); \(z_{-1,-1} = z_{1,1} := b_1, z_{-1,0} = z_{1,0} := b_2, z_{-1,-1} = z_{1,1} := b_3, z_{0,-1} = z_{0,1} := b_4, z_{0,0} := b_5\). Hence, the number of connection weights to be adapted is a small number, 11, for the chosen neighborhood size of 1. So, the learning is accomplished through modification of the following weight vector \(w \in \mathbb{R}^{11}\) whose entries are the feedback template coefficients \(a_i\)’s the input template coefficients \(b_j\)’s, and the threshold \(I\).

\[
w := [a^T b^T I]^T = [a_1 a_2 a_3 a_4 a_5 b_1 b_2 b_3 b_4 b_5 I]^T.
\]
Several design methods and supervised learning algorithms for determining templates coefficients of CNNs are proposed in the literature (Chua and Yang 1988; Vanderberghe and Vandewalle 1989; Zou et al. 1990; Nossek et al. 1992; Chua and Shi 1991; Chua and Thiran 1991; Kozek et al. 1993; Schuler et al. 1992; Magnussen and Nossek 1992; Güzelis 1992; Balsi 1992; Balsi 1993; Schuler et al. 1993; Karamamut and Gülzeliș, 1994; Gülzelis and Karamamut 1994; Lu and Liu, 1998; Liu 1997; Fajfar et al. 1998; Zarandy 1999). As template design methods, well-known relaxation methods for solving linear inequalities are used in Vanderberghe and Vandewalle (1989), Zou et al. (1990) for finding one of the connection weights providing that the desired outputs are in the equilibrium set of a considered CNN. However, for the methods in Vanderberghe and Vandewalle (1989), Zou et al. (1990), there is not a general procedure on how to specify an initial state vector yielding the desired output for the given external inputs and the found weight vector. A trivial solution in the determination of such a proper initial state vector is to take the desired output as the initial state; but this requires the knowledge of the desired output which is not available for external inputs outside the training set. On the other hand, a number of supervised learning algorithms to find connection weights of CNNs which yield the desired outputs for the given external inputs and the predetermined initial states have been developed in the past (Kozek et al. 1993; Schuler et al. 1992; Magnussen and Nossek 1992; Gülzelis, 1992; Balsi 1992; Balsi 1993; Schuler et al. 1993; Karamamut and Gülzelis 1994; Gülzelis and Karamamut 1994). (see Nossek (1996) for a review.) The backpropagation through time algorithm is applied in Schuler et al. (1992) for learning the desired trajectories in continuous-time CNNs. A modified alternating variable method is used in Magnussen and Nossek (1992) for learning steady-state outputs in discrete-time CNNs. Both of these algorithms are proposed for any kind of CNNs since they do not impose any constraint needed to be imposed on connection weights for ensuring complete stability and the bipolarity of steady-state outputs. It is described in Gülzelis (1992) that the supervised learning of steady-state outputs in completely stable generalized CNNs (Gülzelis and Chua 1993) is a constrained optimization problem, where the objective function is the output error function and constraints are due to some qualitative and quantitative design requirements such as the bipolarity of this steady-state outputs and complete stability. The recurrent backpropagation algorithm (Pineda 1988) is applied in Balsi (1992) and Balsi (1993) to a modified version of CNN differing from the original CNN model in the following respects: 1) cells are fully-connected, 2) the output function is a differentiable sigmoidal one, and 3) the network is designed as a globally asymptotically stable network. In Schuler et al. (1993), the modified versions of the backpropagation through time and the recurrent backpropagation algorithms are used for finding a minimum point of an error measure of the states instead of the output.

The lack of the derivative of error function prevents using gradient-based methods for finding template, minimizing the error. In order to overcome this problem, the output function can be replaced (Karamamut and Gülzelis 1994) with a continuously differentiable one which is close to the original piecewise-linear function in Eq. 2. Whereas the gradient methods are now applicable, the error surfaces have almost flat regions resulting in extremely slow convergence (Karamamut and Gülzelis 1994). An alternative solution to this problem is to use methods not requiring the derivative of error. Such a method is given in Kozek et al. (1993) by introducing genetic optimization algorithms for the supervised learning of the optimal template coefficients. The learning algorithm analyzed in this paper, RPLA, constitutes another solution in this direction. The RPLA is, indeed, a reinforcement type learning algorithm: it terminates if the output mismatching error is zero, otherwise it penalizes connection weights in a manner similar to the perceptron learning rule.

The RPLA is firstly presented in (Gülzelis and Karamamut 1994) for finding template coefficients of a completely-stable CNN to realise an input-(steady-state) output map which is pointwise defined, i.e., described by a set of training samples. Here, the input consists of two parts: the first part is the external input and the second is the initial state. RPLA is a global learning type algorithm in the sense of Nossek (1996). This means that it aims to learn not only equilibrium outputs but also their basins of attraction. RPLA has been applied to nonlinear B-template CNNs (Yalcın and Gülzelis 1996) as well as linear B-template CNNs; moreover a modified version of it has been used for learning regularization parameters in CNN-based early vision models (Gülzelis and Günsel 1995; Günsel and Gülzeliş 1995).

This paper is concerned with the convergence properties of RPLA as well as its performance in learning image processing tasks. It is shown in the paper that RPLA with a sufficiently small constant learning rate converges, in finite number steps, to a solution weight vector if such a solution exists and if the conditions of Theorem 3 are satisfied. The RPLA is indeed reduced to the perceptron learning rule (Rosenblatt 1962) if feedback template coefficients (except for the self-feedback one) are set to zero, i.e., the corresponding CNN is in the linear threshold class (Chua and Shi 1991). This means that a CNN trained with an RPLA for a sufficiently small constant learning rate is capable of learning any locally defined function $F_{\text{local}}(\cdot) : [-1, 1]^n \rightarrow \{-1, 1\}$ of the external input whenever its domain space specified by a $3 \times 3$ nearest neighborhood, is linearly separable.

The structure of the paper is as follows. Section 2 formulates the supervised learning of completely stable CNNs as the minimization of an error function. The dynamics of the difference equations defining the proposed learning algorithm RPLA is analyzed in Sect. 3. Some simulation results on the image processing applications of the RPLA are reported in Sect. 4.