OSTRAVSKÁ UNIVERZITA

DIPLOMOVÁ PRÁCE

2018

BC. TOMÁŠ VOZNIČKA

OSTRAVSKÁ UNIVERZITA
PŘÍRODOVĚDĚCKÁ FAKULTA
KATEDRA INVOFMATINY A POČITAČŮ

Celularní neuronové sítě

Diplomová práce

Autor práce: bc. Tomáš Voznička
Vedoucí práce: doc. RNDr. PaedDr. Eva Volná, PhD.

2018

UNIVERSITY OF OSTRAVA
FACULTY OF SCIENCE
DEPARTMENT OF INFORMATICS AND COMPUTERS

Cellular neural networks

Diploma thesis

Author: bc. Tomáš Voznička
Supervisor: doc. RNDr. PaedDr. Eva Volná, PhD.

2018

Detail tématu čekajícího na realizaci

Katedra: *Katedra informatiky a počítačů*

Obor:

Název VŠKP: *Celulární neuronové sítě*

Anglický překlad názvu: *Cellular neural networks*

Typ práce: *diplomová (Nav.)*

Vedoucí práce: Volná, Eva

Pokyny pro vypracování:

1. Rešerše problematiky celulárních neuronových sítí a jejich zařazení do oblasti umělých neuronových sítí a celulárních automatů s důrazem na jejich aplikaci.
2. Výběr vhodného problému a jeho implementace.
3. Zhodnocení výsledků experimentální části.
4. Zhodnocení přínosu práce.

Doporučená literatura:

Chua, L. O., Yang, L., Cellular Neural Networks: theory. IEEE Transactions on Circuits and Systems, Vol.35, No.10, 1988, pp. 1257-1272.

Chua, L. O., Yang, L., Cellular Neural Networks: Applications. IEEE Transactions on Circuits and Systems, Vol.35, No.10, 1988, pp. 1273-1290.

Dolan, R., DeSouza, G. GPU-based simulation of cellular neural networks for image processing, IJCNN, pp.730-735, 2009 International Joint Conference on Neural Networks, 2009.

Yang, T. Cellular Neural Networks and Image Processing. Nova Science Publishers, 2002.

Abstrakt

Abstrakt v prvním jazyce

Klíčová slova: Klíčová, slova

Abstract

Abstract in the second language

Key Words: Key, words

Já, níže podepsaný student, tímto čestně prohlašuji, že text mnou odevzdané závěrečné práce v písemné podobě i na CD nosiči je totožný s textem závěrečné práce vloženým v databázi DIPL2.

Prohlašuji, že předložená práce je mým původním autorským dílem, které jsem vypracoval samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.

V Ostravě dne 2. 4. 2018 ....

podpis

Contents

1 Introduction

This work is focused on introducing the Cellular neural networks (CNN) to the reader as such it is separated into several sections. First some theoretical introduction to the field of CNN , some information about the origins and explanations of how they work. Second part of this work is dedicated to an experiments that are supposed to show basic capabilities of CNN followed up by an chapter about the practical implementation of the supporting program and it's usage.

1.1 Cellular neural networks

Cellular Neural Networks (CNN) is analog, continuous time, non-linear dynamic systems and formally belongs to the class of recurrent Neural Networks. Since their introduction in 1988 by Chua and Yang [1] they have been the subjects of intense research.
CNN can have almost any size and dimensions as the math and principles are valid, they are locally connected, cells have connection only to is neighbour but further influence is present due to the usage of continuous time.

2 Experiments

This chapter will deal with the examples of experiments plausible with this work. Each of the following reports contains description, parameters and results of a given experiment.

2.1 Common properties.

Following experiments are using several common properties in setup, in order to keep the reports clear all go over them here.
Input: inputs are general gonna be ether gray-scale pictures or binary ones with the values -1/1.
Boundary conditions: Boundary conditions defines the networks behavior on the edges of the picture. Two of them are used, Fixed where the outline of the picture is "surrounded" by fixed value, or Flux where the edge pixels are copied outwards.
Initial output: this setting states the initial values of output matrix, commonly its not an important since the calculations are not working with it. but i some experiments its set to some picture or other specific value.

2.2 Edge detection in gray-scale picture.

2.2.1 Description

This example shows the ability to use Cellular neural networks to find edges of different objects in grey-scale picture. It uses a grey-scale picture as an input, matrix setting shown bellow(1) and fixed value boundary condition to produce a binary picture with the edges.

2.2.2 Setup

Input: Grayscale picture.
Boundary conditions: Flux.
Initial output: Unimportant (all zeros)

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \; B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \; Z = -0.5 \tag{1}$$

Figure 1: Chosen values of A,B and Z for this experiment

2.2.3 Results

Figure 2 show input used in this example, it is a picture with several objects of different shade of grey. The Figure 3 shows typical result of this experiment. This functionality is based on removing all pixels that have all only pixels with the same color around them.

Figure 2: Input

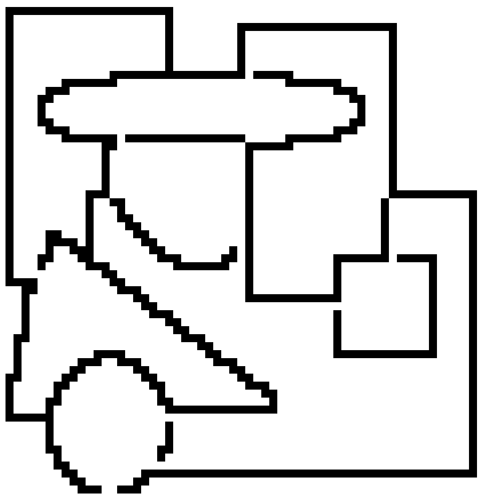

Figure 3: Output

2.3 Directional deletion.

2.3.1 Description

This example shows the ability to use Cellular neural networks to delete lines in specific directions. It uses a binary picture as an input, matrix setting shown bellow and fixed value boundary condition to produce a binary picture with the edges.

2.3.2 Setup

Input: Binary picture.
Boundary conditions: Fixed.
Initial output: Unimportant (all zeros)

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} Z = -2 \tag{2}$$

Figure 4: Chosen values of A,B and Z for horizontal deletion

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} Z = -2 \tag{3}$$

Figure 5: Chosen values of A,B and Z for vertical deletion

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} Z = -2 \tag{4}$$

Figure 6: Chosen values of A,B and Z for top left diagonal deletion

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} Z = -2 \tag{5}$$

Figure 7: Chosen values of A,B and Z for top right diagonal deletion

2.3.3 Results

This functionality works by deleting pixel if there is any other pixel right next to it in chosen direction. Bellow you can see the input used for this experiment and the standard outputs for the several directions mention above.

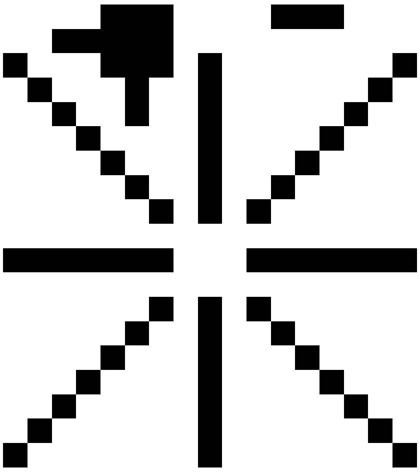

Figure 8: Input

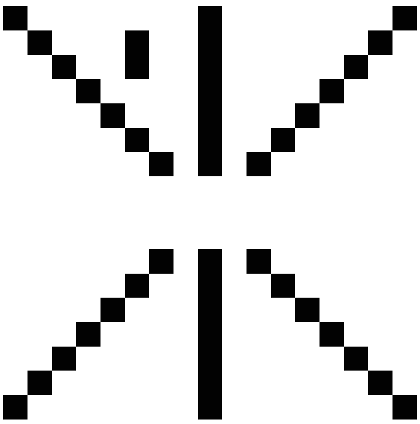

Figure 9: Horizontal deletion

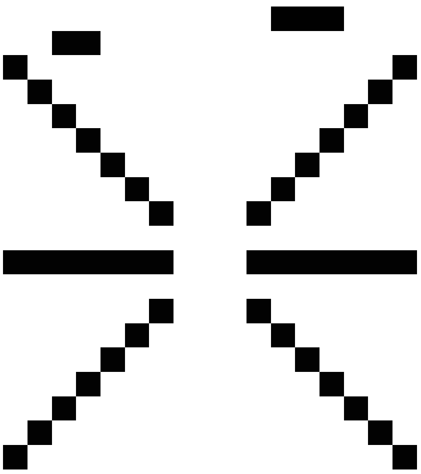

Figure 10: Vertical deletion

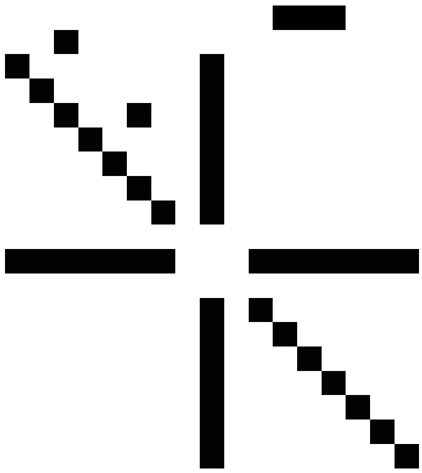

Figure 11: Top right diagonal deletion

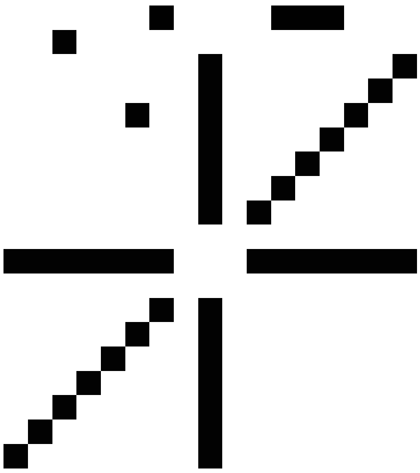

Figure 12: Top left diagonal deletion

2.4 Average.

2.4.1 Description

This example shows the ability to use Cellular neural networks to produce a binary representation of grey-scale picture base on the average of the pixels. It uses a grey-scale picture as an input, matrix setting shown bellow and fixed value boundary condition to produce a binary picture with the edges.

2.4.2 Setup

Input: Unimportant (all zeros)
Boundary conditions: Fixed.
Initial output: Grey-scale picture

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Z = 0 \tag{6}$$

Figure 13: Chosen values of A,B and Z for this experiment

2.4.3 Results

Figure 14: Input

Figure 15: Output

2.5 Black propagation.

2.5.1 Description

This example shows the ability to use Cellular neural networks to make black objects bigger (white smaller) on a binary picture. It uses a binary picture as an input, matrix setting shown bellow and fixed value boundary condition to produce a binary picture with the edges.

2.5.2 Setup

Input: Unimportant (all zeros)
Boundary conditions: Fixed.
Initial output: Grey-scale picture

$$A = \begin{bmatrix} 0.25 & 0.25 & 0.25 \\ 0.25 & 3 & 0.25 \\ 0.25 & 0.25 & 0.25 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Z = 3.75 \tag{7}$$

Figure 16: Chosen values of A,B and Z for this experiment

2.5.3 Results

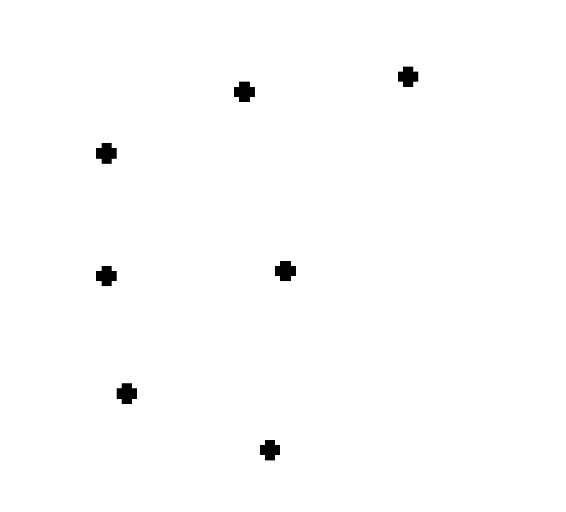

Figure 17: Input

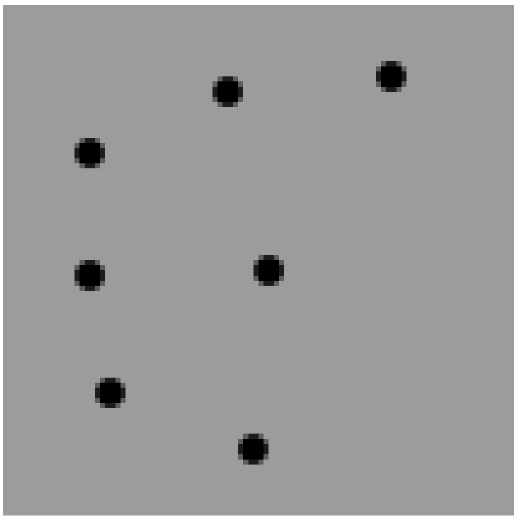

Figure 18: Output

2.6 Logical Not.

2.6.1 Description

Cellular neural networks are able to make logical operations on binary pictures, this example shows logical negation. Black pixels to white and white to black. It uses a binary picture as an input, matrix setting shown bellow and fixed value boundary condition to produce a binary picture with the edges.

2.6.2 Setup

Input: Unimportant (all zeros)
Boundary conditions: Fixed.
Initial output: Grey-scale picture

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad Z = 0 \tag{8}$$

Figure 19: Chosen values of A,B and Z for this experiment

2.6.3 Results

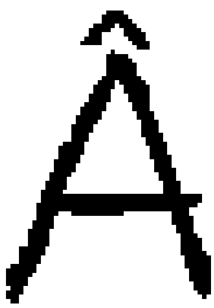

Figure 20: Input

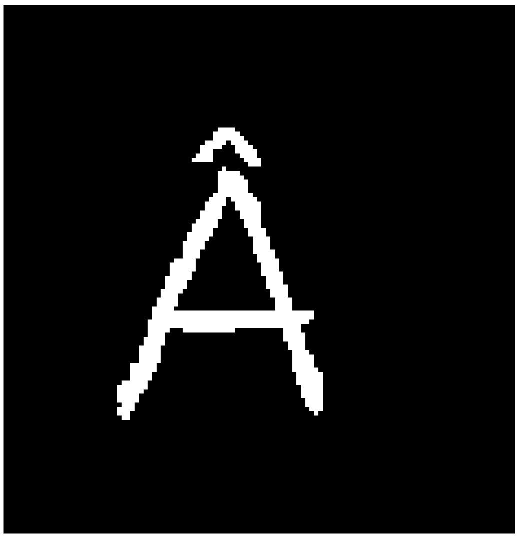

Figure 21: Output

2.6.4 Other logical functions

2.7 Pattern match.

2.7.1 Description

This example shows the ability to use Cellular neural networks to find patterns in picture, this example shows a search for 3x3 pattern designated by B matrix. It uses a binary picture as an input, matrix setting shown bellow and fixed value boundary condition to produce a binary picture with the edges.
Patter is set up by B matrix by the positions of -1/0/1, where -1 represents white, 1 black and 0 is for non specific color.

2.7.2 Setup

Input: Unimportant (all zeros)
Boundary conditions: Fixed.
Initial output: Grey-scale picture

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} Z = -6.5 \tag{9}$$

Figure 22: Chosen values of A,B and Z for this experiment

2.7.3 Results

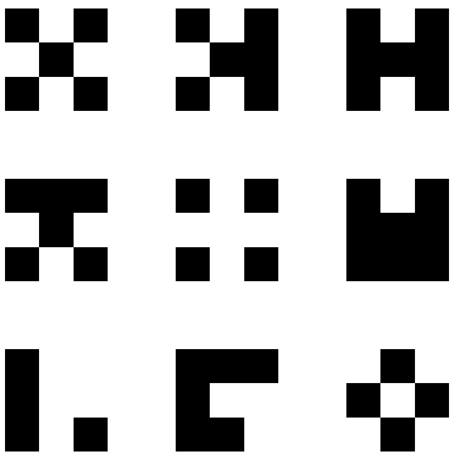

Figure 23: Input

Figure 24: Output

3 Implementations

3.1 Code

3.2 UI elements

Resumé

Resumé v prvním jazyce

Summary

Summary in the second language

References

[1] *L.O. Chua, and L. Yang* Cellular Neural Networks: Theory and Applications, IEEE Trans. on Circuits and Systems, Vol.35, 1998, pp. 1257- 1290.

[2] *Ryanne Dolan and Guilherme DeSouza* GPU-Based Simulation of Cellular Neural Networks for Image Processing, 2009, International Joint Conference on Neural Networks, 2009.