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 and their applications. 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 (CNNs) 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.
history description application

2 Examples of Image processing

This chapter will deal with the examples of image processing plausible with CNNs. Each of the following reports contains description, parameters and results of a given examples.

2.1 Common properties.

Due to the nature of CNNs all following examples shares the same parameters that need to set. In order to keep the individual reports clear here are explanations of those parameters.

Input 1: Inputs in these examples are ether gray-scale pictures or binary ones with the values -1/1.In some cases can be also Arbitrary (all 0) in these the input picture is used as an initial value state for the network.

Initial state/Input 2: This set the initial values of the network, can be Arbitrary (all 0) or can be set to be some picture.

Gene: A gene is a one string representation of Z, A and B (in this order) separated by ";". Separator can by any symbol, ";" is used here as its used in the supporting program.

Boundary conditions: Boundary conditions defines the networks behaviour on the edges of the picture ("boundary layer"). Three of them are used in the examples bellow.

Fixed: The "boundary layer" of cells doesn't not make any computation and doesn't change it's value.

Flex: During the computations on "boundary layer" values from outside of picture are substituted by taking value of nearest pixel of the picture.

Arbitrary: During the computations on "boundary layer" values from outside of picture are substituted by a defined user given 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.

2.2.2 Setup

Input 1: Grayscale picture.
Initial state/Input 2: Arbitrary
Boundary conditions: Flux.
Output: Binary.
Gene: -0.5;0;0;0;0;5;0;0;0;0;-1;-1;-1;-1;8;-1;-1;-1;-1

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

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

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 only pixels with the same color around them.

Figure 2: Input

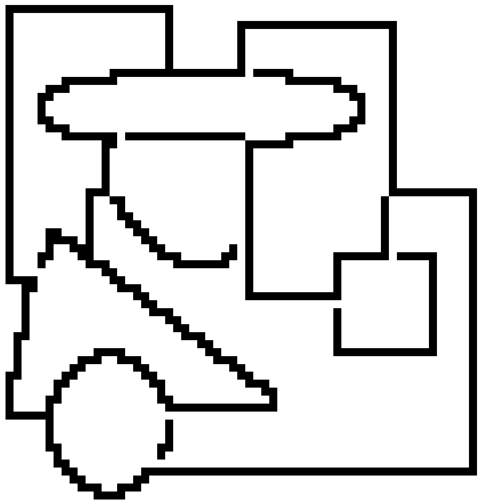

Figure 3: Output

2.3 Directional deletion.

2.3.1 Description

This example shows the ability to use Cellular neural networks to delete lines in specific directions.

2.3.2 Setup

Input 1: Binary picture.
Initial state/Input 2: Arbitrary
Boundary conditions: Arbitrary (0).
Output: Binary.
Gene: -2;0;0;0;0;1;0;0;0;0;0;-1;0;0;1;0;0;-1;0

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

Figure 4: Chosen values of A,B and Z for horizontal deletion
B for other directions, A and Z are the same.

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

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

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

Figure 5: Vertical

Figure 6: Top right

Figure 7: Top left

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.

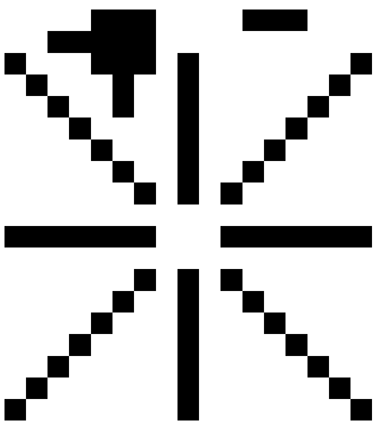

Figure 8: Input

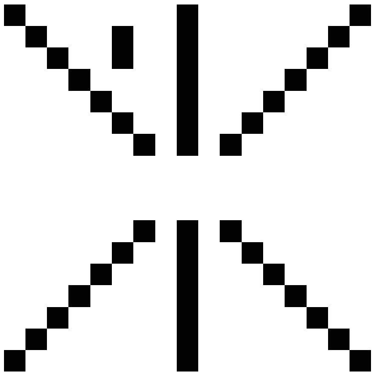

Figure 9: Horizontal deletion

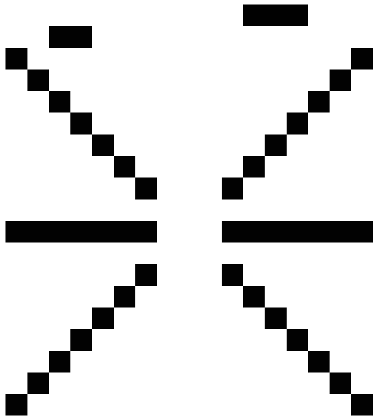

Figure 10: Vertical deletion

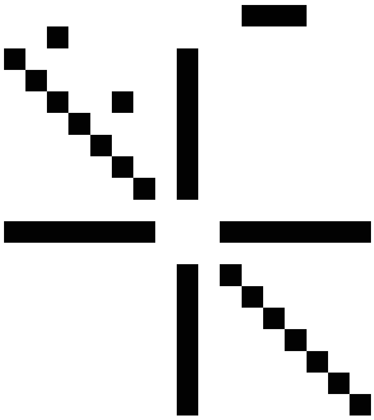

Figure 11: Top right diagonal deletion

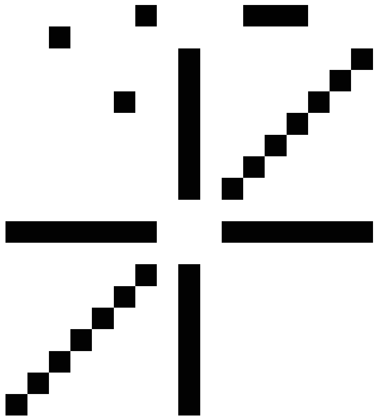

Figure 12: Top left diagonal deletion

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.

2.4.2 Setup

Input 1: Arbitrary
Initial state/Input 2: Grayscale picture.
Boundary conditions: Arbitrary (0).
Output: Binary.
Gene: 0;0;1;0;1;2;1;0;1;0;0;0;0;0;0;0;0;0;0

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

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

2.4.3 Results

Figure 14 show input used in this example, it is grayscale picture. The Figure 15 shows typical result of this example after running 5 cycles when this example became stable.

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.

2.5.2 Setup

Input 1: Arbitrary.
Initial state/Input 2: Grayscale picture
Boundary conditions: Flux.
Output: Binary.
Gene: 3.75;0.25;0.25;0.25;0.25;3;0.25;0.25;0.25;0.25;0;0;0;0;0;0;0;0;0

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

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

2.5.3 Results

Figure 17 show input used in this example, it is a picture with several black objects. The Figure 18 shows typical result of this example after running 5 cycles, each cycle makes the black object grow a new layer of black pixels around them.

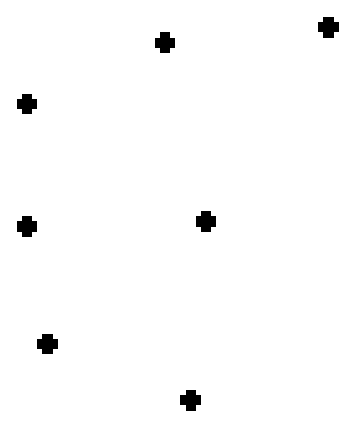

Figure 17: Input

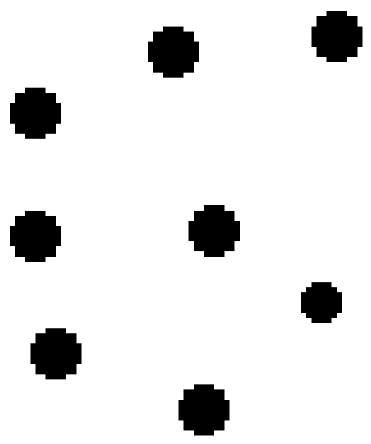

Figure 18: Output

2.6 Pattern match.

2.6.1 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.
Patter is set up as B matrix by the positions of -1/0/1, where -1 represents white, 1 black and 0 is for non specific color.

2.6.2 Setup

Input 1: Arbitrary
Initial state/Input 2: Grayscale picture.
Boundary conditions: Arbitrary (0).
Output: Binary.
Gene: -6.5;0;0;0;0;1;0;0;0;0;1;-1;1;0;1;0;1;-1;1

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} Z = -6.5 \tag{8}$$

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

2.6.3 Results

Figure 20 show input used in this example, it is a picture with several black patterns. The Figure 21 shows typical result of this example with a given B where black pixels marks the position of the middle pixel of recognised pattern. For different patterns B has to be adjusted.

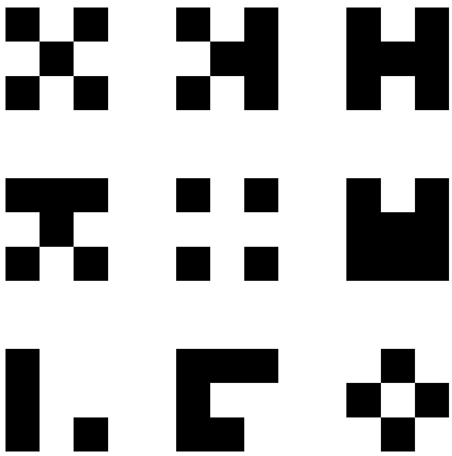

Figure 20: Input

Figure 21: Output

2.7 Logical Not.

2.7.1 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.

2.7.2 Setup

Input 1: Binary picture
Initial state/Input 2: Arbitrary.
Boundary conditions: Flex
Output: Binary.
Gene: 0;0;0;0;0;1;0;0;0;0;0;0;0;0;-2;0;0;0;0

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} Z = 0 \tag{9}$$

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

2.7.3 Results

Figure 23 show input used in this example. The Figure 24 shows typical result of this example, image with negated colors.

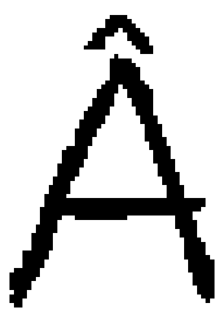

Figure 23: Input

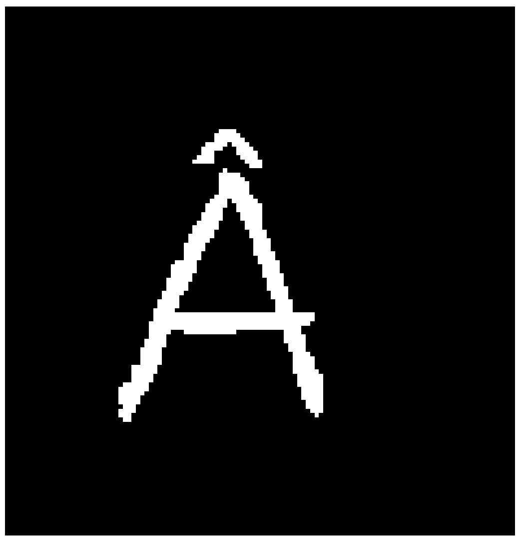

Figure 24: Output

2.8 Logical And/Or.

2.8.1 Description

Cellular neural networks are able to make logical operations on binary pictures, this example shows logical And(Z = -1) and Or(Z = 1)

2.8.2 Setup

Input 1: Binary picture
Initial state/Input 2: Binary picture.
Boundary conditions: Arbitrary (0)
Output: Binary.
Gene: 1;0;0;0;0;2;0;0;0;0;0;0;0;1;0;0;0;0

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} Z = 1 \tag{10}$$

Figure 25: Chosen values of A,B and Z for this experiment (logical Or)

2.8.3 Results

Figure 26 show input used in this example. The Figure 28 shows typical result of this example, image with negated colors.

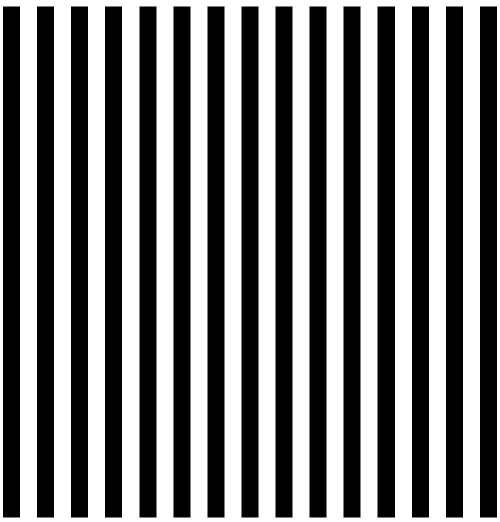

Figure 26: Input1

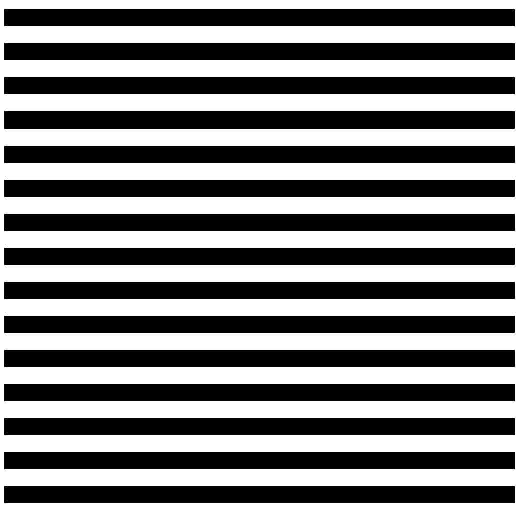

Figure 27: Input2

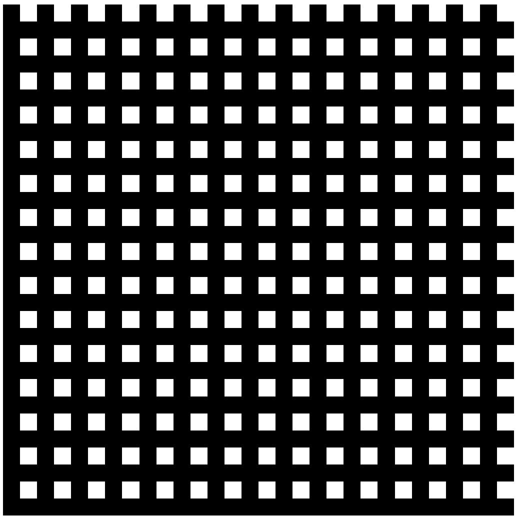
Figure 28: Output for OR

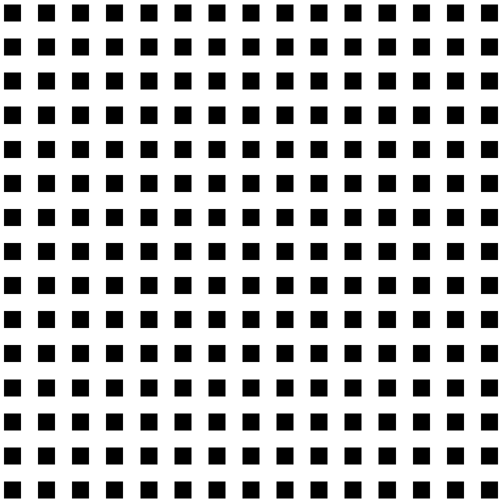
Figure 29: output for and

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.