New Digital Approach to CNN On-chip Implementation for Pattern Recognition

Daniela Durackova

Slovak University of Technology, Bratislava
Slovakia

1. Introduction

There are many cases where image processing is performed using the neural network. The content of this Chapter is implementation of CNN (Cellular Neural Network) on a chip in a new original digital approach. First, testing of some features of the proposed cell of the neural network is evaluated. Then the design of the basic circuit containing the cell for the CNN will be introduced. The idea is to use it for a more complex chip with image processing application.

CNN have been extensively used in various image processing applications.

One of the most important problems in signal processing is noise removal. The input signals mostly arrive from the real world and therefore they contain relatively high amounts of noise. The CNN circuits are capable of removing this noise partly. The possibility of noise removing depends on the template (weight) coefficients between the cells.

CNN networks are based on relatively simple principles, very similar to biological neurons. The input signals are multiplied by appropriate weights, the weight matrix being given intuitively, as it is typical for CNN networks. Then they are added and multiplied by the transfer function. We describe also the settings of weights coefficients. CNN have been extensively used in various image processing applications (Matsumoto & Yokohama, 1990) or (Szirányi & Csicsvári, 1993).

The main problem of CNN implementations on a chip is the chip area consumption. The most area is reserved for the multiplexer, so we looked for alternative multiplications. We describe the achieved results with the designed chip. During the creation of the chip architecture we proposed and introduced special original coding for the weight coefficients. After simulations we recognized that the results were better with rounding than without it, but we found that the rounding during multiplication was not as important as we previously expected. Therefore we decided – instead of a hardware multiplexer – to use multiplication realized by simple gate timing of the special signal.

The circuit was designed as a digital synchronized circuit. For the design and simulation we used the Xilinx tool. The cell is based on sub-circuits as it will be shown in detail.

The Chapter describes the designed circuit. We introduce also our novel simulator for the CNN using the program tool Visual Basic for Application, and its algorithm which was based on the same principle as the planned designed circuit. The network can process the patterns with 400 points of recognition.

The aim of our work is an original approach to the designed digital chip architecture using special coding of the weights coefficients. This implementation of the CNN on a chip widens the family of the previous designed circuits.

2. Some notes to using CNN for image processing

The modern world needs image processing very often. There exist various methods for image processing, among them the methods based on the principle of neural networks are also useful. The advantage of the neural network is parallel processing and the implementation on the chip is designed as an analogue or digital circuit, however, the principle for both approaches is similar. According to the theory it is based on the sequences of single inputs (x1... x9) multiplied by weights (w1....w9), then the conversion through the transfer function (as we see in Fig. 1) prepares the signal for the next processing.
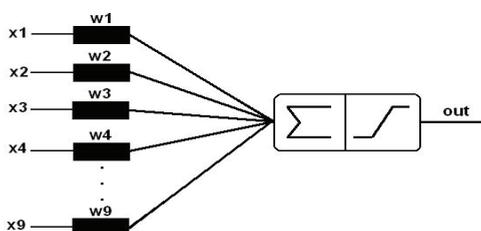

Fig. 1. Graphic representation of the cell.

There exist various transfer functions, as for example the sigmoid function, hard-limiter function or threshold logic function. For image processing the sigmoid function is commonly used. The properties of the network, as for example noise removing, depend on correct choices of the weight matrix. Every image is characterized by its weight matrix. We skip here the basic details well known from the classical theory introduced by Prof. L.O.Chua (Chua & Yang, 1988a; 1988b). As we will see in the next parts of the Chapter, according to the theory of CNN networks the weight matrix is set intuitively.

3. The Novel simulator of CNN

The basic circuit of the neural network is called the cell. The network contains cells connected together according to the proposed rule. The cellular neural network used neighbourhood 1, which means that each cell is connected with only the nearest neighbours. For the theoretical results we first created a simulator for CNN programmed as a macro in Visual Basic for Application. The results from the simulator were first used for evaluation of CNN behaviour, then to compare the results with those obtained from the designed chip. The results of simulations were achieved in matrix and graphical representations. The input and output values are in the range from –15 to + 15 and the graphical form is represented using spectra of two colours, as shown in Fig. 2.

Fig. 2. The spectra of used colours.

In the first row there is a red colour scale which represents positive numbers, the blue colour in the second row represents negative numbers.

Our first simulator contained 16 cells. We tested some features of the neuron network, as e.g. detection of vertical lines (see Fig. 3) or filling of the picture (Fig. 4).
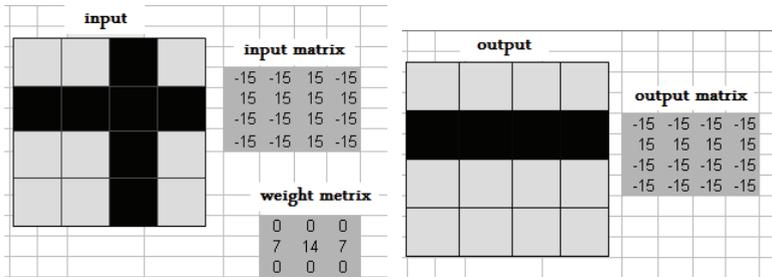

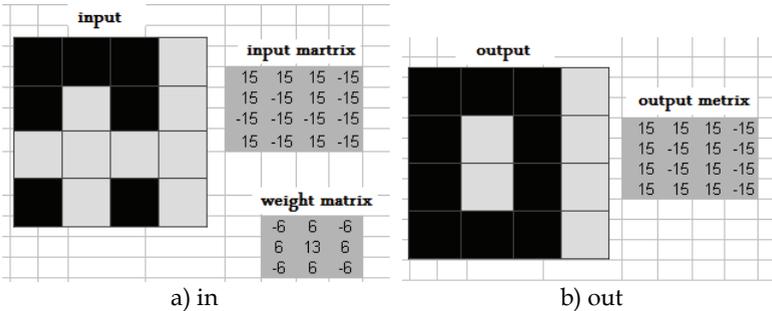Fig. 3. Filling of the lines for the given pattern.

a) in　　　　　　　　　　　　　　　　b) out

Fig. 4. Completing rectangle rims

These figures illustrate the simple tasks for image processing. The left parts show the pattern input, the right parts are the output results.

From this pattern we can recognize that the network is able to satisfy the given conditions. For more complicated image processing we need to widen the network to 400 cells. As the input matrix we use the 20×20 input values and a 3×3 input weight matrix. During the simulations we can change the slope of the sigmoid function, to fix the number of iterations, or to finish it automatically. The simulator gives the results with rounding, or without it. The results were mostly similar, however, the proposed rounding gave more precise results.

The next experiment was focused on noise filtering. As an input we used letter "E", with the random noise (around 35 %). The results are in Fig. 5. The quality of the results was dependent on the weight matrix and the position of the fault information. The weight matrix was given intuitively, as it is typical for CNN networks. In the first example (Fig. 5b), five points are not removed because we chose an incorrect weight matrix. Figure 5c shows perfect noise removal.

The weight matrices for Figs. 5b and 5c are as follows:

$$\text{For figure } b: \begin{bmatrix} -4 & 6 & -4 \\ 6 & 7 & 6 \\ -4 & 6 & -4 \end{bmatrix} \quad \text{and for figure } c: \begin{bmatrix} -3 & 6 & -3 \\ 6 & 7 & 6 \\ -3 & 6 & -3 \end{bmatrix}$$
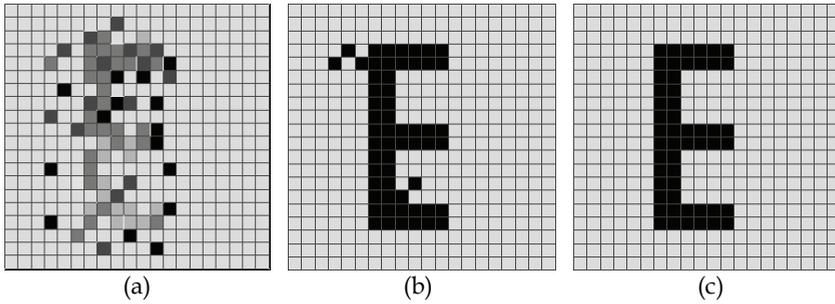
| (a) | (b) | (c) |

Fig. 5. Noise filtering

> a) input to the network

> b) output after the use of an inappropriate weight matrix

> c) output after perfect noise removing

We have developed a novel simulator for the CNN using the program tool Visual Basic for Application, and its algorithm was based on the same principle as the planned designed circuit. The network can process the patterns with the recognition of 400 points. The created universal simulator can change various simulation parameters.

We have found that rounding by multiplication is not as important as we previously expected. On the basis of the simulations we have design a novel digital CNN cell. This will be used for CNN consisting of 400 cells which will be used for image processing in the future. The circuit contains some service signals. For the cells connected into a CNN network it is inevitable to design a control circuit which will control synchronization.

4. The design of the digital CNN cell

After the simulations we started to design our digital approach for CNN implementation on a chip. The proposed circuit was designed as a digital synchronized circuit. For the design and simulation we used the Xilinx tool. The cell is based on sub-circuits as schematically shown in Fig. 6.
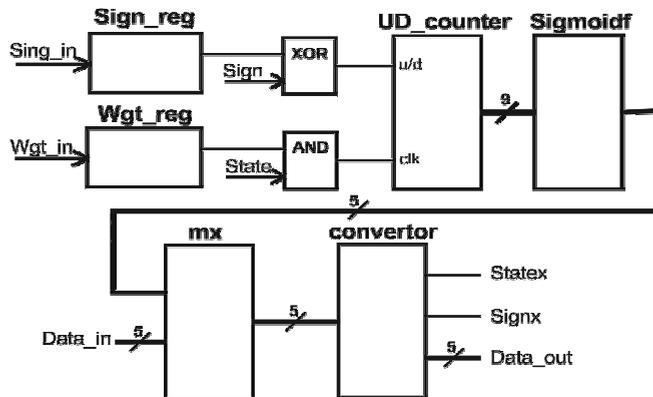

Fig. 6. The block diagram of the digital CNN cell

CNN networks are based on relatively simple principles, very similar to biological neurons. The input signals are multiplied by appropriate weights, then they are added and multiplied by the transfer function. The main problem of the CNN implementations on a chip is the chip area consumption. Most of the area is reserved for the multiplexer, so we looked for alternative multiplications. After the simulations we recognized that the results were almost the same with rounding and without it.

As an input we set *Wgt_reg*, then the sign 9 bit serial register *sign_reg* and input values State from the neighbour cells. Then we included also their sign and the input of the cells themselves because in the CNN theory the cell is the neighbour also to itself.

The inputs from each cell are multiplexed by the input multiplexer. The input has information about the sign of the state of the next cell. These inputs are multiplexed by the weight register in the AND gate, and the inputs are compared with the gate XOR. The weights have to be specially timed, so by timing we can perform multiplication.

Then the results are added in the *Ud_counter* and sequentially transferred through block *sigmoidf* . Function hard limiter (hl) has the values true or false, while functions threshold logic(tl) and sigmoid function(sf) could have values also between these extreme values true and false. We decided that function (sf) is the best as it has ist derivative in the whole range, while function (tl) has no derivative in points –1 and +1. This behaviour is important for neural networks which are able to learn. The plot of function (sf) for various slopes is in Fig. 7.

The results are passing through the block *sigmoidf*, which realized the sigmoid function. The block *converter* transfers value output on the time interval, which sends nearest neighbors and contains register, where is the result storing.

According to Fig. 6. the inputs from each cell are multiplexed by the input multiplexer. The block converter contains a register, where the result is stored.
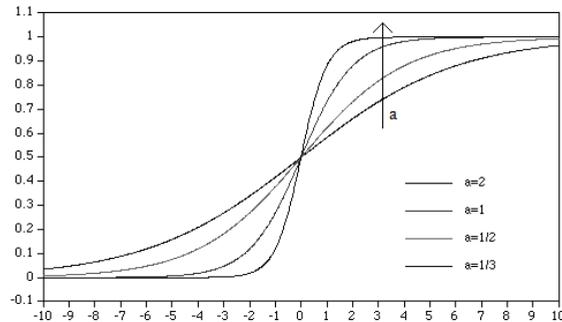

Fig. 7. The sigmoid function

5. Multiplication of signals using the AND gate

The method of multiplication is based on the fact that at multiplication the input value must be converted to the time signal and the weight value has to be special picked, so multiplication starts by timing. We proposed special coding for the weights.

We used a special system of 15parts, e.g., one cycle is divided into 15 parts. In Fig. 8 we see the first 15 weights and their original coding.

An example of special rounding and coding of the weights is shown in Fig. 9. The real value of the multiplication is x.w=0.293 and the result after Fig. 9. is 4/15=0.26. We used the simulator to recognize that we can neglect this rounding.
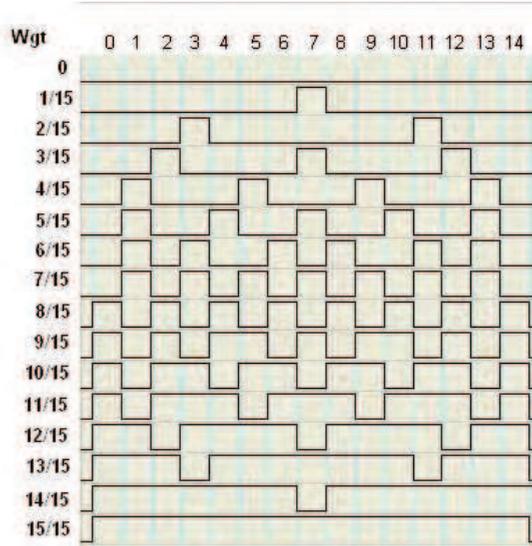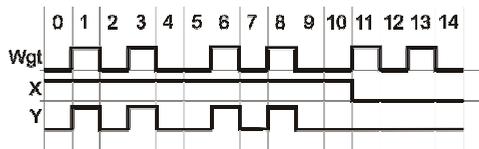
Fig. 8. The first fifteen weights in the proposed system

Fig. 9. An example of evaluation for weight wgt=6/15 and input x=11/15

6. The results of the simulations of the designed CNN cell

After creating the architecture of our implemented CNN we made some simulations on the circuit. In Fig. 10 we can see the filling of the lines for the given pattern. At the input a) we see the corners of the input pattern. After 15 iterations we get the result as we see in Fig. 10b.
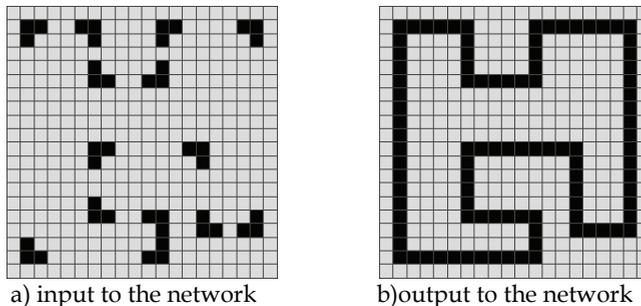

a) input to the network   b)output to the network

Fig. 10. The network completes the corners into a continuous line

The next experiment was focused on noise filtering. As an input we used letter "A" with random noise (around 10 %). Figure 11b shows the noise filtering output after the 3rd

iteration. The quality of the results was dependent on the weight matrix and the position of the fault information. For the noise filtering of letter "A" we use the following weight

matrix: $\begin{bmatrix} -1 & 2 & -1 \\ 2 & 8 & 2 \\ -1 & 2 & -1 \end{bmatrix}$
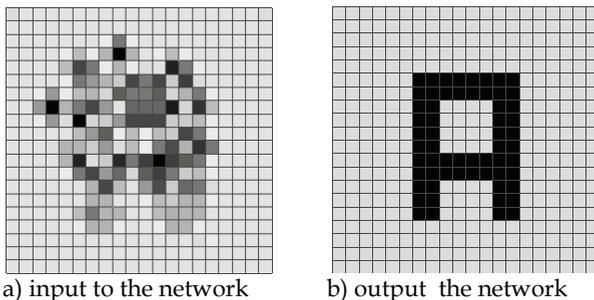

a) input to the network   b) output the network

Fig. 11. Filtering of the noise from the letter.

The weight (wgt-out) is timed so that we can multiply it with the input. The result (here denoted as (sucin)) after multiplication with the sigmoid function and converting to the time interval is stored in (statex) multiplied with the sign (signx). In Fig. 12 we see the result after 15 tacts from the beginning.

7. Conclusion

We developed a novel simulator for the CNN using the program tool Visual Basic for Application. Its algorithm is based on the same principle as the planned designed circuit. The network can process the patterns with 400 point recognition. The created universal simulator can change various simulation parameters.

We found that the rounding at multiplication is not as important as we previously expected. On the basis of the simulations we designed a novel digital CNN cell implemented on a chip. This will be used for the CNN consisting of 400 cells. The architecture of the designed digital chip is cascadable, so we can create various capacity of CNN networks.

We expect the applications of the designed chip in CNN for interactive education of the deaf (particularly children) trying to learn how to use the dactyl alphabet.

The new experiment with the Data sensor glove used for this purpose is the topics of our present research.

8. References

Chua, L.O. & Yang.L. (1988). Cellular neural networks: Theory, IEEE Trans. On Circuits and Systems, Vol. 35, No. 10, (1988). Pp.1257-1272

Chua, L.O. & Yang.L. (1988). Cellular neural networks: Applications, IEEE Trans. On Circuits and Systems, Vol. 35, No. 10, (1988). Pp.1273-1290

Durackova, D & Raschman, E. (2007). Pattern recognition for the letters of dactyl alphabet by using a novel neural network simulator, Proceedings of the 5th IEEE Conference on Computational Cybernetics, 19-21 October 2007, Gammarth, Tunisia, pp. 97-99.

Matsumoto, T., Yokohama,T., Suzuki, H., Furukawa, (1990). Several image processing examples by cnn, Proceedings of IEEE International Workshop on Cellular Neural Networks and their Applications, Budapest

Szirányi, T.; & Csicsvári, J. (1993). High speed character recognition using a dual cellular neural network architecture (cnnd), Analog and Digital Signal Processing, Vol. 40, No. 3, pp. 223-231.
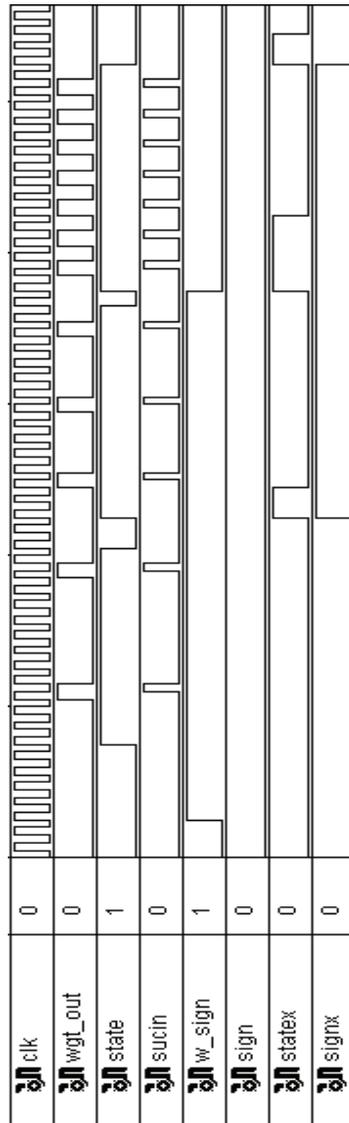

Figure 12. The output results from the Xilinx simulations

Pattern Recognition Techniques, Technology and Applications

Edited by Peng-Yeng Yin

A wealth of advanced pattern recognition algorithms are emerging from the interdiscipline between technologies of effective visual features and the human-brain cognition process. Effective visual features are made possible through the rapid developments in appropriate sensor equipments, novel filter designs, and viable information processing architectures. While the understanding of human-brain cognition process broadens the way in which the computer can perform pattern recognition tasks. The present book is intended to collect representative researches around the globe focusing on low-level vision, filter design, features and image descriptors, data mining and analysis, and biologically inspired algorithms. The 27 chapters coved in this book disclose recent advances and new ideas in promoting the techniques, technology and applications of pattern recognition.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH

open science | open minds