A NEW SYNAPTIC OPERATOR AND ITS APPLICATIONS FOR EFFICIENT IMPLEMENTATION OF CELLULAR NEURAL NETWORKS IN DIGITAL TECHNOLOGIES

by

Radu Dogaru, Kenneth R. Crounse, and Leon O. Chua

Memorandum No. UCB/ERL M97/40

3 June 1997

A NEW SYNAPTIC OPERATOR AND ITS APPLICATIONS FOR EFFICIENT IMPLEMENTATION OF CELLULAR NEURAL NETWORKS IN DIGITAL TECHNOLOGIES

by

Radu Dogaru, Kenneth R. Crounse, and Leon O. Chua

Memorandum No. UCB/ERL M97/40

3 June 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A New Synaptic Operator and its Applications for Efficient Implementation of Cellular Neural Networks in Digital Technologies

Radu Dogaru[1], Kenneth R. Crounse and Leon. O. Chua
University of California Berkeley,
Nonlinear Electronics Laboratory, 258 M Cory Hall,
Berkeley, CA-94720 1770
E-mail: radu_d@fred.eecs.berkeley.edu

June 3, 1997

Abstract

In this paper we show that the multiplication operator used in defining the conventional CNN synapses can be expressed as a particular case of a more general operator, defined here as a "generalised synapse". Instead of the conventional multiplication operation, a different particular case of this generalized synaptic operator was found to be an excellent candidate for VLSI implementation of CNN synapses in digital technology since it may be defined by using only the addition, subtraction and absolute value operations. The effectiveness of this new operator is demonstrated with discrete-time CNN examples operating in all possible dynamic modes (equilibrium, periodic and chaotic).

1. Introduction

In a fully parallel CNN [3] implementation, for both continuous-time and discrete-time models, one goal is to reduce as much as possible the area occupied by the processing units. In such structures the input of each processing cell is computed as a weighted summation of the outputs of neighbouring cells. The weight values define a "cloning template" which specifies the behavior (overall function) of the CNN. The weighting operation is often associated with information processing at the synaptic level in biological neural systems. Basically, only three main processing functions associated with each VLSI cell are needed to implement a template programmable CNN: an analog memory for storing the cloning templates, an analog circuit for emulating the cell body and the axons, and a synaptic interconnection circuit. In terms of the occupied area, the last function is the most important since the number of synapses in a CNN is usually at least 9 times the number of neurons (cells).

Despite evidence that biological synapses do not actually perform a mathematical multiplication of the synaptic strength (weight) with the dendritic stimulus, in most artificial models of neural networks a linear additive model defined by $x = \sum_{i \in N} y_i \cdot w_i$ is generally used to compute the state x (input) of a specific neuron (cell) when the outputs y_i and synaptic strengths w_i of the neurons belonging to the set N are known. Here x can also be considered as a value which indicates the correlation between the inputs and the weights. The multiplication operator is found not only in neural networks but also in many information-processing techniques. This situation is motivated by the existence of powerful mathematical and symbolic

---

[1] On leave from University "Politehnica" of Bucharest with a Fulbright grant (permanent e-mail address: radu_d@atm.neuro.pub.ro and http://atm.neuro.pub.ro/~radu_d).
The work was also partially suported by the Office of Naval Research under grant N00014-96-1-0753.

tools around this operator which allows a rigorous analysis and design of all functions built as compositions of the addition and multiplication operators. For the particular case of CNN formal models, more general synaptic models have been considered [2]. Even if such non-linear synaptic operators may lead to complications when the symbolic design of weights is considered, the approach to be presented below can lead to very efficient hardware implementations and may be applied to many cases where heuristics, adaptation and genetic algorithms are used to explore the weight space for optimal solutions. Non-linear synapses other than multiplier-based ones have already been reported in some artificial neural network models [1]. Also, real biological systems are able to perform many tasks using complex non-linear synapses where the weight values are established through such mechanisms as adaptation, self-organization and evolution. What is important in the neural processing of information is the idea of correlation between the inputs and the prototype patterns stored as weights of the synapses to which these inputs are applied. If the correlation with a particular pattern is strong enough, the associated neuron fires. How this correlation is effectively implemented (i.e. the nature of the synaptic operator) is not important as long as the main feature is preserved, i.e. the correlation function reaches its maximum value only when both weights and input vector are correlated.

From the implementation perspective, multiplication is highly area consuming, particularly in digital implementations where such advantages as good reproducibility and precise weight storage are to be exploited. For fixed point-arithmetic with a prescribed resolution *res*, the area required by a combinational multiplication unit is $O(res^2)$, while for many other operators (addition, subtraction, finding the minimum value, e.t.c.) the area is only $O(res)$ [11].

Instead of other solutions proposed in the literature based on emulating multipliers by shift registers, we propose, in section 2, a new synaptic model called a "comparative synapse". We demonstrate in this section that this operator is a particular case of a more general synaptic operator which also includes the ordinary multiplication operation as a particular case. For information tasks that do not need symbolic computations it is thus much more convenient to think in terms of a generalized product and then to choose the particular case which maximizes the efficiency in VLSI implementations. Some examples of applying this principle to the DT-CNN synapse design are presented in section 3 in order to show that independent of the particular synaptic model used (ordinary multiplier or comparative synapse), the overall CNN functionality remains unchanged. Some comments on the implications to VLSI implementations and conclusions are given in section 4.

2. From a new synapse model to a generalised product operator

In [8] the synapse model

$$y = \mathrm{sgn}(x)\mathrm{sgn}(w)\min\left(|x|,|w|\right) \tag{1}$$

was proposed for computing the contribution y of a particular synapse to a specific cell input where $\mathrm{sgn}(\bullet)$ denotes the signum function[2] and where x and w are the output of another cell in the net, and the synaptic weight value, respectively. This model has the advantage of having a very simple VLSI implementation in either analog or digital form. It has also been successfully tested in a feed-forward neural structure for signal classification tasks.

---

[2] The signum function is defined by $\mathrm{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$

Theorem: $\forall x, w \in \mathrm{IR}$ the following two functions $f_1(x,w)$ and $f_2(x,w)$ are equivalent;

$$f_1(x,w) = \mathrm{sgn}(x)\mathrm{sgn}(w)\min(|x|,|w|) \quad (2)$$

$$f_2(x,w) = \tfrac{1}{2}(|x+w| - |x-w|) \quad (3)$$

Proof: For $x=w=0$ it is straightforward to show that $f_1(0,0) = 0 = f_2(0,0)$

In what follows $x \in \mathrm{IR}$ and w is considered as a parameter and an explicit canonical representation for both $w>0$ and $w<0$ will be derived for $f_2(x,w)$.

I. ($w>0$) : $f_1(x,w) = \mathrm{sgn}(x)\min(|x|,w)$ where the function has the graphical representation shown in Fig.1.a. It is clear that $f_1(x,w)$ represents a cross section with respect to the parameter w of a piece-wise linear two-dimensional function and thus, using results in [4], has the following canonical representation:

$$f_1(x,w) = \tfrac{1}{2}(|x+w| - |x-w|) \quad (4)$$

II. ($w<0$): $f_1(x,w) = -\mathrm{sgn}(x)\min(|x|,-w)$ where the function has the graphical representation shown in Fig.1.b. The corresponding canonical representation is given by:

$$f_1(x,w) = \tfrac{1}{2}(-|x-w| + |x+w|) = \tfrac{1}{2}(|x+w| - |x-w|) \quad (5)$$

It follows from (4) and (5) that for all values of w, $f_1(x,w)$ is equivalent to the representation $f_2(x,w)$:

$$f_1(x,w) = \tfrac{1}{2}(|x+w| - |x-w|) = f_2(x,w)$$

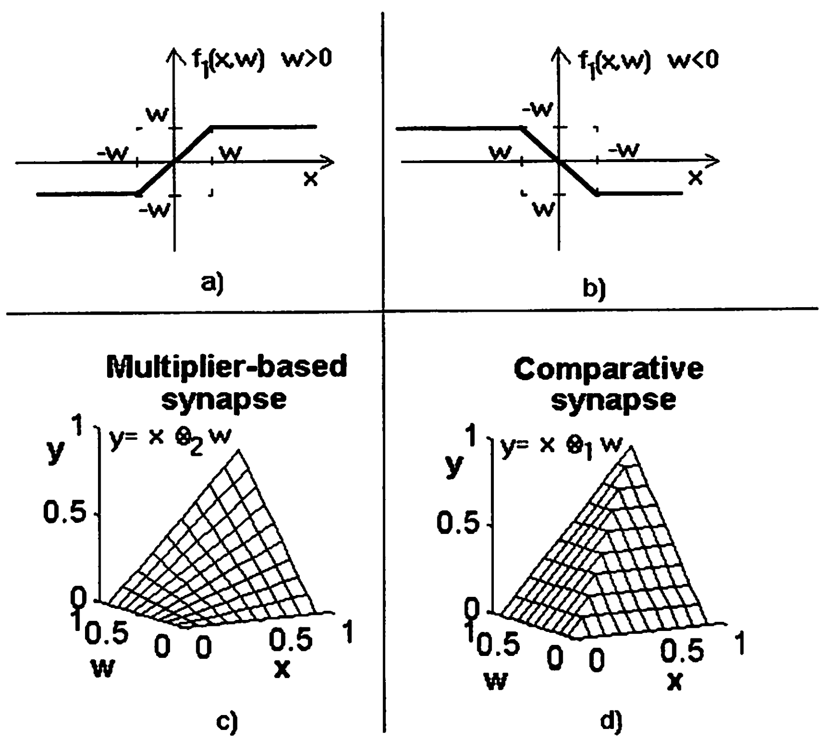

Figure 1: Different piece-wise linear function representations;
a) $f_1(x,w)$ representation for $w>0$; b) $f_1(x,w)$ representation for $w<0$;
c) The ordinary multiplication ; d) The "comparative synapse".

Let us now define a generalized synaptic operator $\otimes_\alpha$ between two n-vectors **x** and **y** via the relationship:

$$\mathbf{x} \otimes_\alpha \mathbf{y} = \left(\left\|(\mathbf{x}+\mathbf{y})/2\right\|_\alpha\right)^\alpha - \left(\left\|(\mathbf{x}-\mathbf{y})/2\right\|_\alpha\right)^\alpha \tag{6}$$

where α is a positive real number and $\|\mathbf{z}\|_\alpha = \left(\sum_{i=1}^{n} |z_i|^\alpha\right)^{1/\alpha}$ defines a particular norm " α " of the n-dimensional vector **z**. In the general case, both **x** and **y** are vectors representing an arbitrary input and weight, respectively, for a specific neuron cell. The operator $\otimes_\alpha$ performs a generalised inner product (and thus the result is a scalar), corresponding to the soma activation in biological neurons. It follows from definition (6), which is applicable for any vector dimension including 1, that $\mathbf{x} \otimes_\alpha \mathbf{y} = \sum_{i=1}^{n} x_i \otimes_\alpha y_i$, where x_i and y_i are scalar components associated with each particular synapse. Thus, the main property of the generalised operator is exploited at the synaptic level, each contribution being summed independently of α to compute the soma activation of a cell neuron. In what follows we will consider the following two particular synaptic cases;

a) $\alpha = 1 \Rightarrow \|z\|_1 = |z|$ (Manhattan norm) and b) $\alpha = 2 \Rightarrow \|z\|_2 = \|z\| = \sqrt{(z)^2}$ (Euclidean norm). Then, we will find the following particular expressions for the generalised synaptic operator:

$$x \otimes_1 y = f_2(x,y) = f_1(x,y) \tag{7}$$

$$x \otimes_2 y = (x/2 + y/2)^2 - (x/2 - y/2)^2 = xy \tag{8}$$

Observe that (8) corresponds to the conventional, multiplicative synapse (8) while (7) to the "comparative synapse". Both (7) and (8) are generalised expressions of the logical "AND" operator (obtained when the arguments are restricted to: $x, y \in \{0,1\}$). Moreover, (7) is equivalent to one of the most often used "Fuzzy-AND" operator when $x, y \in [0,1]$. Thus, we can conclude that (6) represents a more general expression for a synaptic operation which is based only on summations/subtractions and on one-variable norm evaluation.

It is clear that the hardware for implementing the norm-taking operation α depends on the choice of the technology, and thus, (6) provides a higher degree of freedom than that of conventional multiplication. Indeed, for digital technology, it is obvious that α=1 is the best choice for norm taking, since in this case, taking the absolute value of a number is equivalent to finding its magnitude, which requires no hardware but simply copying the magnitude bits. For comparison, in the same digital technology, taking the α=2 norm will require $O(res^2)$ logic gates where *res* denotes the representation resolution of the digital numbers (usually at least 8 bits).

Figure 1.c and 1.d presents the graphical representation of the above two operators where both weights and inputs are positive and less than one. For $|w| \leq 1$ and $|x| \leq 1$ one may also consider the $\otimes_1$ operator as a piece-wise linear approximation of the ordinary multiplication operation $\otimes_2$.

Properties of the $\otimes_1$ synaptic operator:

1. ***The synaptic output is bounded by either the input or the weight value;***

This follows from (1) and for CNNs it has the following consequence: For the feedback template A it makes no sense to use weights greater than one if the neuron outputs are bounded by one (this is the case in conventional CNN models). Indeed, these outputs are inputs to other synapses and according to (1) the output of the $\otimes_1$ synapse is always bounded by one even if the weights are greater than one.

Even though this property seems to appear as a limitation, from the VLSI implementation perspective it is very useful since all weights and state variables are restricted to vary within a specified bounded domain. In order to obtain effects which are specific for weights larger than 1, one may simply choose an output saturation level $w_{\max} > 1$. Here, $w_{\max}$ is the largest absolute value of the weights in the cloning template matrix A. Also, in many cases, the cloning template can be simply scaled down , that is, the actual size of the weights is irrelevant but only the relative size is important.

2. ***The $\otimes_2$ operator may be replaced by the $\otimes_1$ operator in linear threshold gates to perform the same Boolean function without any change in weight values as long as their absolute values are bounded by 1.***

In a threshold gate, any input $x_i \in \{-1,1\}$. Thus, for any synapse we can write $x_i \otimes_2 w_i = w_i \mathrm{sgn}(x_i)$. According to (1) and taking into consideration that $|x_i| = 1$ if $x_i \in \{-1,1\}$, we have $x_i \otimes_1 w_i = [\mathrm{sgn}(w_i)\min(1,|w_i|)]\,\mathrm{sgn}(x_i)$. But, while $|w_i| \le 1$ and taking into account that $x = |x|\mathrm{sgn}(x)$, it follows that $x_i \otimes_1 w_i = [\mathrm{sgn}(w_i)|w_i|]\,\mathrm{sgn}(x_i) = w_i\,\mathrm{sgn}(x_i) = x_i \otimes_2 w_i$. As long as the non-linear function used in the threshold gates is a hard limiter defined by $y = \mathrm{sgn}\left(\sum_i w_i \mathrm{sgn}(x_i)\right)$, the condition $|w_i| \le 1$ is not a restriction. Indeed, if this condition is not fulfilled, one may simply rescale the weights as $w_i' = (1/|w_{\max}|)w_i$. Thus, the implemented Boolean function remains unchanged while $y = \mathrm{sgn}\left(\frac{1}{|w_{\max}|}\sum_i w_i \mathrm{sgn}(x_i)\right) = \mathrm{sgn}\left(\sum_i w_i \mathrm{sgn}(x_i)\right)$. As a consequence of this property, all CNN designs using hard-limiter neurons or threshold-like operators either remains unchanged, or require only proper scalings of the cloning templates in order to ensure the condition $|w_i| \le 1$.

3. Examples of using generalised synapses in CNNs

Consider the discrete-time CNN (DT-CNN) model with *NxN* cells defined by:

$$\mathbf{X(0)} = \mathbf{X0}\ ;\ \mathbf{X}(t) = \mathbf{A} * \mathbf{F}(\mathbf{X}(t-1)) + \mathbf{B} * \mathbf{U} + I\ \ ;\ \mathrm{t} = 1,..\mathrm{T} \tag{9}$$

where **X0** is an *NxN* matrix associated with a planar monochrome (black and white) image to be processed. An image matrix **U** having elements associated with pixel brightness and represented as positive numbers less than 1, can be also applied as an input and filtered using the feed-forward cloning template **B.** The system may evolve to an equilibrium if for $t > T_e$ $\mathbf{X}(t+1) = \mathbf{X}(t)$, or to a periodic attractor if T exists for which $\mathbf{X}(t+T) = \mathbf{X}(t)$, or even to a chaotic attractor. In the simulations presented below we used *N=64* and *T=200*. Each of the *NxN* cells is characterised by the same non-linear output function *y=f(x),* feedback cloning template **A,** and feed-forward cloning template **B.** The matrix function **F** collects in a compact representation all independent cells' output nonlinearities *f(x).* The * operator denotes a generalised spatial convolution defined as: $(\mathbf{A} * \mathbf{X})_{i,j} = \sum_{k,l \in N_r(i,j)} a_{k,l} \otimes_\alpha x_{i,j}$ where $N_r(i,j) = \{k,l \mid \max(|k-i|,|l-j|) \le r\}$ is the set of indices associated with all cells lying in a neighborhood of radius *r* of the cell *i,j*.

For the experiments presented below, we have considered different non-linear output functions *f(x)* and cloning templates in order to get different dynamic behaviours in a 64x64 cell DT-CNN. The image to be processed was considered both as an initial state **X(0)** or as an input **U**. In all cases, *r=1* and thus both feed-forward and feed-back cloning templates are 3X3 matrices. For each example we have considered both $\otimes_1$ and $\otimes_2$ operators while the design procedure for templates was based on heuristics and on previously designed templates for such functions as image halftoning [12] [13]. In what follows we show that there is no major influence on the overall CNN functionality when the synaptic operator is changed from $\otimes_2$ .to $\otimes_1$.

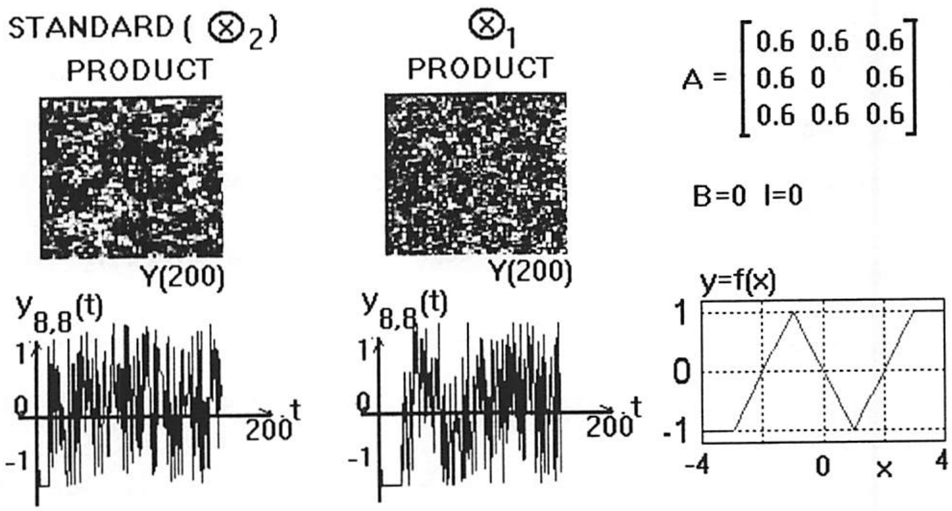

Figure 2: Hyper-chaotic discrete-time CNN simulation with different synaptic operators.

<u>Example 1:</u> (Chaotic behaviour):

In Fig.2 the structure and simulation results for a high-dimensional hyper-chaotic DT-CNN are presented. Such CNNs may be used as efficient ciphering sequence generators in communication systems since their high dimensionally makes it very difficult for such systems to be identified. In each step, complex spatio-temporal patterns are generated. A sample obtained at the stopping moment (T=200) is presented in Fig.2 for both synaptic operators. The temporal evolution of a particular cell (the middle one) is also presented. Observe that for identical CNN structures using different particular cases of the generalised product as synaptic operators, the same overall behaviour was obtained.

Moreover, the weight space exploring method (WSE) described in [7] was successfully applied to CNNs using both types of synaptic operators. In this way, we have designed chaotic and even synchronizable DT-CNNs with the "comparative synapse" as easily as when using the conventional multiplier synapse.

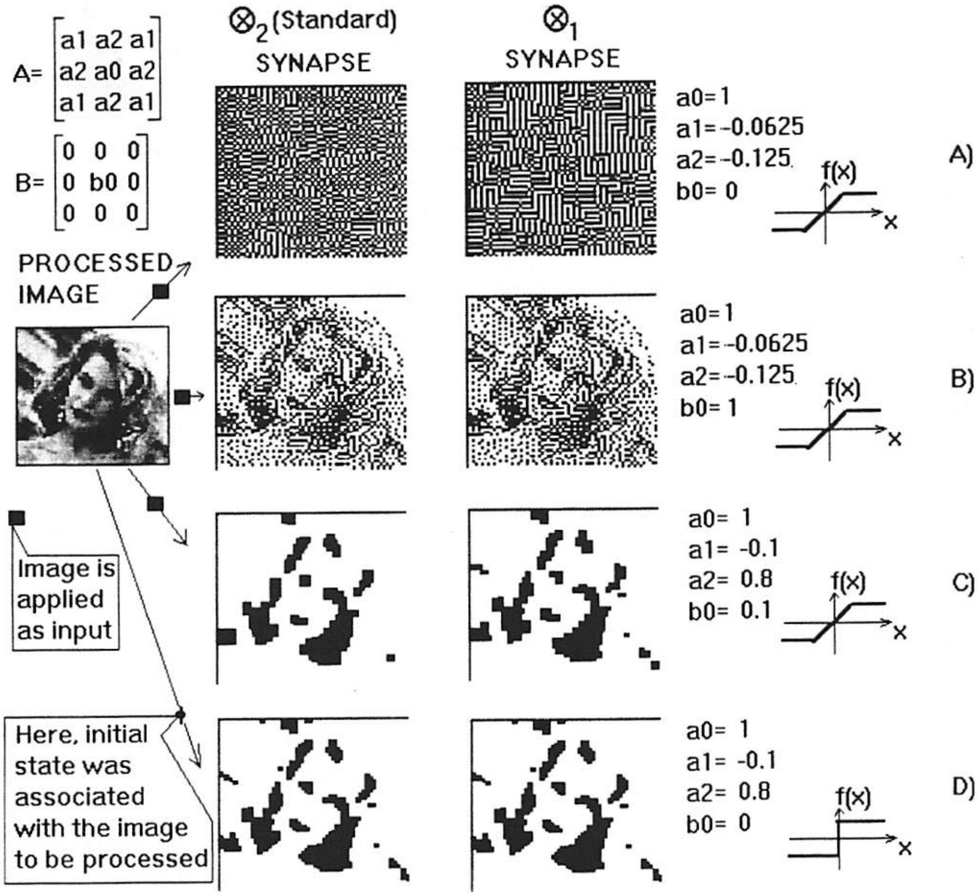

Figure 3: Information processing with equilibrium states in DT-CNNs with different synaptic operators: A) Labyrinth-like pattern formation ; B) Image halftoning ; C) Feature enhancement I ; D) Feature enhancement II

Example 2: (Pattern formation, Fig.3.A):

In this example, the input image is not accessible since $b0=0$ (in Fig.3.A) and thus the CNN system is an autonomous one, evolving to an equilibrium pattern. Such patterns were first described in [12] and have been exploited for halftoning applications. The initial state of the system is chosen to be random and small in both cases. While there are some differences in the roughness of the final pattern one may see that replacing the classic synapse with the comparative synapse does not change the essential functionality (in this case, labyrinth-like pattern formation). Moreover, patterns with the same degree of roughness may be obtained by re-adjusting the cloning template values when a comparative synapse is used.

Example 3: (Image halftoning, Fig.3.B):

When a non-zero feed-forward cloning template **B** is considered, one may exploit the pattern formation propoerty presented above in order to get a halftoned version of a greyscale input image. The results for both synaptic operators, using the same template matrix which was designed according to procedures presented in [12], show that there is a very slight difference in the respective final patterns (halftoned images) and thus the particular synaptic operator does not influence the overall functionality in this case. Moreover, we have tested the same cloning templates for continuous-time CNNs with the same result.

Example 4: (Fixed-point convergence, Fig.3.C)

In this example we consider different cloning templates designed for "primary feature extraction" of an input image when a standard synapse was used. Replacing it with a comparative synapse does not change the basic results even if there are some slight differences between the equilibrium patterns. However, if required, such small differences may be compensated by slightly changing the values of the template matrices.

In all previous examples, the neuron function was the one most often used in actual CNN implementations, i.e. the linear "saturated" function which may be actually written as a particular case of the $\otimes_1$ synaptic operator $y = f(x) = x \otimes_1 1$, where the variable weight value was simply replaced by the constant 1. Thus, when comparative synapses are used, the only two cells needed to construct a CNN are the memory cells for storing the cloning template values and the corresponding synaptic operator. It must also be noticed that based on Property 1 (bounded synaptic output) even if linear neurons ($y=f(x)=x$) are used, the states of the system still remain constrained to evolve within a bounded domain. Such systems may have some implementation advantages while preserving the convergence to equilibrium states in order to perform different useful processing tasks.

Example 5: (hard-limiter nonlinearity, Fig. 3. D);

For our final example, we will consider a hard-limiter activation function in order to test the effects of Property 2. Indeed, one may see that the final pattern is now independent of which synapse (multiplier-based or comparative) is used. In this case, the image is applied as an initial state of the system and each neuron along with the associated synapses may be considered as performing a Boolean operation. It may be inferred that such systems are the most robust for changes in either the models, or in the synaptic weight values. Based on other previous results [6], [8] we suggest that for a neural cell there is a strong relationship between the robustness of the synaptic weights and the number of output levels allowed for that cell. The lower the number of output levels allowed (minimum is 2 in the case of threshold logic) the more robust the cell is with respect to changes in both the synaptic weight value and an non-linear synaptic operator. Here, robustness is considered with respect to the overall functionality of the CNN. This may also explain why slight differences in the final patterns may appear for the two synaptic models when the continuous-valued neuron function was used (examples 1-4). However, in all cases the overall functionality was always maintained despite the choice of a different synaptic model.

4. Conclusions

A general expression for a non-linear synaptic operator is proposed. This operator is called a "generalised synapse" and it is proved that both the classic multiplication synapse and a novel synaptic operator, called the "comparative" synapse, are merely special cases of this generalized operator with respect to the choice of a particular norm. The $\otimes_1$ operator has the advantage over the $\otimes_2$ operator in terms of its VLSI implementation. This property is particularly important in CNNs where most of the silicon area for hardware implementations is allocated to the synaptic functions. Particular examples using such operators in DT-CNN were considered in this paper to emphasise the independence of the overall behaviour on the particular choice of the generalised product. Until powerful symbolic (arithmetical) processing rules are derived for the $\otimes_1$ operator, as is now the case for the $\otimes_2$ operator, the comparative synapse will be restricted to such applications where heuristic rules or other design methods based on non-symbolic searching in weight space (adaptive, evolutionary, stochastic) are used for designing the parameters of the information processing system. However, this is the case in most conventional CNN template design techniques (e.g. [5][10]). Hence, within the CNN framework, we expect the cloning template re-design procedures to be straightforward. Moreover, as a consequence of

Property 2, for all CNN models where binary cell outputs are considered [9] there is no need to re-design the cloning templates.

While the implementation advantages of the comparative synapse are obvious in digital technology, it should be mentioned that in analog technology there are other advantages not related with the occupied area but with the implementation precision required for the synaptic operator. Compared to analog multipliers, implementing such functions as "comparison" or "absolute-value" operations in (1) and (3) should involve less stringent component tolerances.

Acknowledgements:

This research was carried out at University California at Berkeley in the framework of a Fulbright research grant awarded to the first author. We express our thanks to Tao Yang for having many stimulating discussions and for his valuable suggestions.

References

[1] G.A Carpenter, S. Grossberg, and D.B. Rosen, "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system", in *Neural Networks, Vol. 4,* pp. 759-771, 1991.

[2] L.O.Chua, M. Hasler, G.S. Moschytz, "Autonomous cellular neural networks: a unified paradigm for pattern formation and active wave propagation", in *IEEE Trans. on Circuits and Systems-I,* Vol. 42, No. 10, pp. 559-577, Oct. 1995,.

[3] L.O. Chua, and L. Yang, "Cellular neural network: theory and practice", in *IEEE Trans. on Circuits and Systems,* Vol. 35, pp. 1257-1290, Oct. 1988.

[4] L. O. Chua, S. M. Kang, "Section-wise piecewise-linear functions: canonical representation, properties, and applications", in *Proceedings of the IEEE*, Vol. 65, No. 6, June 1977.

[5] G. Destri, "Discrete-time cellular neural network construction through evolution programs", in *Proceedings CNNA-96,* Seville, Spain, pp. 473-478, June 1996.

[6] R. Dogaru, A.T. Murgan, D. Ioan, "Robust oscillations and bifurcations in cellular neural networks", in Proceedings of the Third International Workshop on Cellular Neural Networks and their Applications - CNNA 94, Rome, December 1994, pp. 303-309.

[7] R. Dogaru, A.T. Murgan, M. Glesner, S. Ortmann, "Searching for robust chaos in discrete time neural networks using weight space exploration", in *Proceedings ICNN'96* (International Conference on Neural Networks, Washington, 1996), pp. 688-693.

[8] R. Dogaru, L.O. Chua, "A VLSI-friendly Synapse Model and its Application for Fast and Cheap Signal Classification Problems with Neuro-fuzzy Networks", in *Electronics Research Laboratory Memorandum, University of California, Berkeley*, November 10, 1996.

[9] H. Harrer, J.A. Nossek, "Discrete-time cellular neural networks", in *International Journal of Circuit Theory and Applications,* Vol. 20, pp. 435-467, Sept. 1992.

[10] H. Magnussen, J. Nossek, "Towards a learning algorithm for discrete-time cellular neural networks", in *Proceedings CNNA-92,* Munich, Germany, pp. 80-85, Oct. 1992.

[11] J.M. Rabaey, "Digital integrated circuits, a design perspective" , Prentice Hall 1996.

[12] K.R. Crounse, T. Roska, L.O. Chua, "Image halftoning with cellular neural networks", in *IEEE Trans. on Circuits and Systems - II*, Vol. 40, No. 4, pp. 267-283, April 1993.

[13] T. Sziranyi and M. Csapodi, "Texture classification by Cellular Neural Network using genetic learning", in *Proc. of 12th ICPR (Int. Conf. on Pattern Recognition),* Oct. 1994.