Characterization of Canonical Robust Template Values for a Class of Uncoupled CNNs Implementing Linearly Separable Boolean Functions

YIH-LON LIN
Dep. of Electrical Eng.
National Sun Yat-Sen Uni.
Kaohsiung, Taiwan 804
TAIWAN

WEI-CHIH TENG
Dep. of Electrical Eng.
National Sun Yat-Sen Uni.
Kaohsiung, Taiwan 804
TAIWAN

JYH-HORNG JENG
Dep. of Information Eng.
I-Shou University
Kaohsiung County, Taiwan
840
TAIWAN

JER-GUANG HSIEH
Dep. of Electrical Eng.
National Sun Yat-Sen Uni.
Kaohsiung, Taiwan 804
TAIWAN

Abstract: - In this paper, the geometric margin is used as a robustness indicator of a CNN (cellular neural network) implementing a linearly separable Boolean function. For a class of uncoupled CNNs having low template values, characterization of canonical robust template values is made by finding the maximal margin canonical hyperplane. Support vector machine (SVM) technique is employed for the associated optimization problem. Two illustrative examples are provided to illustrate the main result.

Key-Words: - cellular neural network, boolean function, maximal margin

1 Introduction

CNNs (cellular neural networks), first introduced by Chua and Yang [1,2], are large scale nonlinear circuits composed of locally connected cells. CNN has a tremendous variety of applications in the fields of dynamic systems and digital image processing [3-5].

The class of CNNs without feedback interconnection from neighboring cells, namely the uncoupled CNNs, plays an important role in many practical applications. Furthermore, the simplicity of the uncoupled CNN circuit renders it attractive in VLSI implementation. The binary steady-state output in terms of the binary input of the uncoupled CNN can be represented by a linearly separable Boolean function. Most of the elementary applications can be derived and analyzed via Boolean functions [6-8], which is directly related to the CNN template parameters.

One of the crucial issues of VLSI CNN chip design is the robustness of a template set for CNN [9]. Analog VLSI implementations of CNN have numerous limitations that need to be taken into account in the theory of CNN in order to guarantee correct and efficient operations. Template parameters can only be realized with a precision of typically 5~10% of the nominal values and usually only a discrete set of possible values is available [10].

By treating the truth table of a linearly separable Boolean function generated by an uncoupled CNN as the training data set, the geometric margin of the training set is used as a robustness indicator of the given CNN. The maximal margin classifier of a training data set, which provides the maximum geometric margin for the given training data set, will be found by solving an optimization problem in the realm of support vector machine (SVM) theory. SVM is a learning system which was first introduced by Vapnik and his coworkers in 1992 [11]. A unique feature of the SVM is that the final discriminant function for classification problem can be expanded on a small subset of training data, which is referred to as support vectors [12,13]. In the meanwhile, the maximal margin of the optimal separating hyperplane to the nearest vertices can be computed directly from a neat formula.

It is a common practice for robustness consideration to keep the template values of a CNN low. It will be seen that this amounts to providing a guaranteed geometric margin. For a class of uncoupled CNNs having low template values, it is the purpose of this paper to characterize those template values for maximum robustness.

2 Uncoupled CNN

Consider a standard CNN consisting of an $M \times N$ rectangular array of cells. Without loss of generality, we will consider exclusively 3×3 neighborhood for each cell $C(i, j)$, $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$. The CNN parameters are represented by a triple (A, B, z), where $A = [a_{kl}]$ and $B = [b_{kl}]$, $k, l \in \{-1, 0, 1\}$, are 3×3 feedback and control templates, respectively, and z is the threshold value. It is customary to use the following notation to represent the CNN template parameters:

$$A = \begin{array}{|c|c|c|}\hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array},$$

$$B = \begin{array}{|c|c|c|}\hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array}, z.$$

The uncoupled CNN is represented by

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & a_{00} & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad B = \begin{array}{|c|c|c|}\hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array}, z. \quad (1)$$

The governing equations of the uncoupled CNN are given by

$$\dot{x}_{ij} = h_{ij}(x_{ij}, w_{ij}) := g_{ij}(x_{ij}) + w_{ij}, \quad (2a)$$

$$y_{ij} = 0.5 \times |x_{ij} + 1| - 0.5 \times |x_{ij} - 1|, \quad (2b)$$

$$g_{ij}(x_{ij}) = -x_{ij} + a_{00} y_{ij}, \quad (2c)$$

$$w_{ij} = \sum_{k=-1}^{1} \sum_{l=-1}^{1} b_{kl} u_{i+k,j+l} + z_{ij}, \quad (2d)$$

where x_{ij}, y_{ij}, and z_{ij} are the state, output, and threshold of $C(i, j)$, and $u_{i+k,j+l}$, $k, l \in \{-1, 0, 1\}$, are inputs from the neighboring cells including $C(i, j)$. Moreover, $g_{ij}(x_{ij})$ is called the driving-point component and w_{ij} is called the offset level.

With the static binary inputs, i.e., $u_{i+k,j+l} \in \{1, -1\}$, the steady-state output $y_{ij}(\infty)$ of $C(i, j)$ can be calculated explicitly without integrating (2), which is stated as follows [8, Theorem 6.1].

If $a_{00} > 1$, then, starting from any $x_{ij}(0) \in (-1, 1)$, we have

$$y_{ij}(\infty) = sign\lfloor (a_{00} - 1) x_{ij}(0) + w_{ij} \rfloor.$$

If $a_{00} = 1$, then we have

$$y_{ij}(\infty) = sign\lfloor w_{ij} \rfloor, \text{ if } w_{ij} \neq 0,$$

$$y_{ij}(\infty) = x_{ij}(0) \in [-1, 1], \text{ if } w_{ij} = 0.$$

If $a_{00} < 1$, then we have

$$y_{ij}(\infty) = sign\lfloor w_{ij} \rfloor, \text{ if } |w_{ij}| \geq 1 - a_{00},$$
$$y_{ij}(\infty) = (1 - a_{00})^{-1} w_{ij} \in (-1, 1), \text{ if } |w_{ij}| < 1 - a_{00}.$$

Note that in the $a_{00} > 1$ case, by absorbing the term $(a_{00} - 1) x_{ij}(0)$ into the offset level or by selecting $x_{ij}(0) = 0$, we have

$$y_{ij}(\infty) = sign\lfloor w_{ij} \rfloor.$$

For simplicity, rename the uncoupled CNN (1) as

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & a_{00} & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \quad B = \begin{array}{|c|c|c|}\hline w_1 & w_2 & w_3 \\ \hline w_4 & w_5 & w_6 \\ \hline w_7 & w_8 & w_9 \\ \hline \end{array}, b. \quad (3)$$

Also rename $u_{i-1,j-1}$, $u_{i-1,j}$, $u_{i-1,j+1}$, $u_{i,j-1}$, …, and $u_{i+1,j+1}$ in (2d), as u_1, u_2, u_3, u_4, …, and u_9, respectively. Then we have

$$w_{ij} = w_1 u_1 + \ldots + w_9 u_9 + b = \langle w, u \rangle + b := f(u),$$
$$w := [w_1 \quad \ldots \quad w_9]^T, \quad u := [u_1 \quad \ldots \quad u_9]^T \in \Re^9.$$

Usually w is called the weight vector and b is called the bias.

It is well known that a Boolean function $\beta(u_1, u_2, \ldots, u_9)$ of nine variables is realizable by every cell of an uncoupled CNN if and only if $\beta(\cdot)$ can be expressed by the formula

$$\beta(u_1, u_2, \ldots, u_9) = sign[w_1 u_1 + w_2 u_2 + \ldots + w_9 u_9 + b],$$

where w_i, $i = 1, 2, \cdots, 9$, and b are real constants, and $u_i \in \{1, -1\}$, $i = 1, 2, \cdots, 9$, is the ith Boolean variable [8, Theorem 6.2].

It is important to note that the discriminant function

$$f(u) := w_1 u_1 + w_2 u_2 + \ldots + w_9 u_9 + b$$

is an affine-linear function of $u \in \Re^9$. Thus implementing a linearly separable Boolean function by an uncoupled CNN is a linear classification problem.

3 Maximal Margin Classifier

Let $X \subseteq \Re^n$ and $Y := \{1, -1\}$. Suppose we are given a nontrivial training set

$$S := \{(x_i, y_i)\}_{i=1}^{l} \subseteq X \times Y .$$

The training set *S* is said to be linearly separable if there is a hyperplane of the form

$$f_{w,b}(x) := \langle w, x \rangle + b = 0 , \quad w \in \Re^n , \quad b \in \Re ,$$

that correctly classifies the training data. By treating the truth table of a given Boolean function as the training data set with $l = 512$ training data of nine inputs and one binary output ($y_i \in \{1, -1\}$), this training set must be linearly separable in order for the Boolean function to be realizable by an uncoupled CNN.

As pointed out in [8], since no template parameters can be realized exactly in practice, it is important that the CNN template be designed to be as robust as possible. This guarantees the reliability of CNN hardware implementation.

For given (w, b) defining a hyperplane, the functional margin $\mu_S(w, b)$ and the geometric margin $\eta_S(w, b)$ of (w, b) with respect to the training set *S* are defined by, respectively,

$$\mu_S(w, b) := \min_{i=1}^{l} y_i \cdot [\langle w, x_i \rangle + b],$$
$$\eta_S(w, b) := \min_{i=1}^{l} y_i \cdot \left[\left\langle \|w\|^{-1} w, x_i \right\rangle + \|w\|^{-1} b\right].$$

Note that positive scaling of *w* and *b* will affect the functional margin, but with geometric margin unchanged. This geometric margin will be used in this study as the robustness measure of an uncoupled CNN. The margin γ_S is defined to be the maximal geometric margin over all hyperplanes, i.e.,

$$\gamma_S := \max_{w,b} \eta_S(w, b).$$

A hyperplane realizing this maximum is called a maximal margin hyperplane or optimal hyperplane. Any template realizing a maximal margin will be called a maximal margin template or optimal template. See Fig. 1. The optimal canonical hyperplane, with the functional margin equal to 1, can be obtained by solving the following primal optimization problem:

(P) minimize $2^{-1} w^T w$
subject to $y_i \cdot [\langle w, x_i \rangle + b] \geq 1$ for $i = 1, 2, \cdots, l$.

Suppose (w^*, b^*) solves the primal optimization problem (P). Then the maximal margin hyperplane is given by $f^*(x) = \langle w^*, x \rangle + b^* = 0$ with margin $\gamma_S = \|w^*\|^{-1}$.

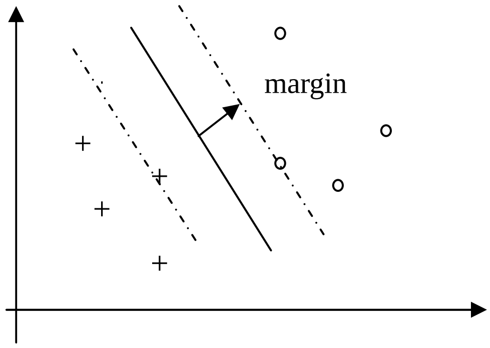

Fig. 1. Maximal margin hyperplane.

For CNN template design, we assume the initial state $x_{ij}(0)$ to be zero so that the value of a_{00} would not affect the resulting optimal values of *B* template and the threshold z_{ij}. Note again that once the optimal w^* and b^* have been found, if we choose $a_{00} < 1$ in the template design, then, for correct binary output, w^* and b^* must be multiplied by the constant $1 - a_{00}$ for actual template parameters and bias in order to provide a functional margin $1 - a_{00}$.

4 Optimal Uncoupled CNN Templates

It is a common practice for robustness consideration to keep the template values of a CNN low. Since the maximum geometric margin is given by the reciprocal of the Euclidean norm of *B* template values corresponding to an optimal canonical hyperplane, this amounts to providing a guaranteed robustness. In this paper, we consider the class of uncoupled CNNs (3) with $w_i \in \{-1, 0, 1\}$, $i = 1, 2, \cdots, 9$.

Recall that the steady-state output $y_{ij}(\infty)$ for binary inputs is given by

$$y_{ij}(\infty) = sign[w_{ij}] = sign[\langle w,u \rangle + b]$$

with the discriminant function given by

$$f(u) = \sum_{i=1}^{9} w_i u_i + b = \langle w,u \rangle + b \,.$$

It is interesting to characterize the template parameters such that the uncoupled CNN determines a maximal margin hyperplane.

Suppose there are *r* nonzero entries in *B* template. It is easy to see that

$$\langle w,u \rangle \in \{-r, -r+2, -r+4, \cdots, r-2, r\}.$$

In order to provide a unity functional margin for optimal canonical hyperplane, it is necessary to choose the bias *b* from the set

$$Z = \{-r+1, -r+3, \cdots, r-3, r-1\}.$$

This guarantees that the values of $f(u)$ will never be zero for all training patterns of the Boolean truth table and the functional margin is unity. It is not difficult to show that if we choose the bias from the set *Z*, the corresponding uncoupled CNN determines a maximal margin classifier with maximum margin given by $1/\sqrt{r}$. This can also be justified by first constructing the Boolean truth table from a given uncoupled CNN. Next, by treating the truth table as the training data set, we solve the SVM optimization problem (P). Finally, the resulting weight vector and the bias then determine the *B* template values and the threshold value.

We wish to point out that our main result is useful for speeding up the CFC algorithm which is used to find a series of uncoupled CNNs implementing an arbitrary Boolean function [14].

5 Illustrative Examples

Example 1: Consider the class of templates (A_0, B_0, z_0) given by

$$A_0 = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\,, \quad B_0 = \begin{array}{|c|c|c|}\hline 1 & -1 & 1 \\ \hline 1 & 0 & -1 \\ \hline 0 & -1 & 1 \\ \hline \end{array}\,, \quad z_0 \,.$$

Here $z_0 \in (-9, 9)$ is arbitrary. There are seven nonzero entries in *B* template, so $r = 7$. There are only seven possible maximal margin templates $(A_{opt}, B_{opt}, z_{opt})$ resulting from this class, which are given by

$$A_{opt} = A_0 \,, \quad B_{opt} = B_0 \,, \quad z_{opt} = -6, -4, \cdots, 4, 6 \,.$$

The (maximum) margin of this class is given by $M_{opt} = 1/\sqrt{7}$.

Example 2: The class of uncoupled CNNs investigated in this paper is rather special. The original template values in *B*, which are -1, 0, and 1, remain unchanged for a maximal margin template. Only threshold matters. Consider now the following uncoupled CNN, which does not fall in the class of CNNs considered in this paper:

$$A_0 = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\,, \quad B_0 = \begin{array}{|c|c|c|}\hline 4 & 0 & -3 \\ \hline 0 & 0 & 0 \\ \hline -1 & 1 & 0 \\ \hline \end{array}\,, \quad z_0 = -4 \,.$$

The optimal canonical template values generating the same Boolean function are given by

$$A_{opt} = A_0 \,, \quad B_{opt} = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\,, \quad z_{opt} = -1 \,.$$

It is observed that some template values in *B* and the threshold have been changed. It is interesting to further investigate the characterization of maximal margin CNN template values for a more general class of CNNs.

References:

[1] L. O. Chua and L. Yang, “Cellular neural networks: Theory,” *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1257-1272, Oct. 1988.

[2] L. O. Chua and L. Yang, “Cellular neural networks: Applications,” *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 1273-1290,

Oct. 1988.
[3] I. N. Aizenberg, N. N. Aizenberg, and J. Vandewalle, “Precise edge detection: Representation by boolean functions, implementations on the CNN,” *Proceeding of IEEE International Workshop on Cellular Neural Networks and Their Applications*, London, 1998, pp. 301-306.
[4] I. Szatmári, A. Schultz, C. Rekeczky, T. Kozek, T. Roska, and L.O. Chua, “Morphology and autowave metric on CNN applied to bubble-debris classification,” *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1385-1393, Nov. 2000.
[5] I. N. Aizenberg, “Processing of noisy and small-detailed gray-scale images using cellular neural networks,” *Journal of Electronic Imaging*, vol. 6, no. 3, pp. 272-285, July 1997.
[6] Z. Galias, “Designing cellular neural networks for the evaluation of local boolean functions,” *IEEE Transactions on Circuits and Systems II*, vol. 40, pp. 219-223, Mar. 1993.
[7] L. Nemes, L. O. Chua, and T. Roska, “Implementation of arbitrary boolean functions on the CNN universal machine,” *International Journal of Circuit Theory and Applications*, vol. 26, no. 6, pp. 593-610, Nov. 1998.
[8] L.O. Chua and T. Roska, *Cellular Neural Networks and Visual Computing*. Cambridge, United Kingdom: Cambridge University Press, 2002.
[9] P. Kinget and M. Steyaert, “Evaluation of CNN template robustness towards VLSI implementation,” *International Journal of Circuit Theory and Applications*, vol. 24, no. 1, pp. 111-120, Jan. 1996.
[10] D. Lím and G. S. Moschytz, “A modular g_m-C programmable CNN implementation,” *Proceeding of IEEE International Symposium on Circuits and Systems*, Monterey, California, June 1998, vol. 3, pp. 139-142.
[11] B. E., Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” *Proceeding of the Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, Pennsylvania, United States, 1992, pp. 144-152.
[12] V. N. Vapnik, *The Nature of Statistical Learning Theory*. 2nd ed., New York: Springer, 1999.
[13] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, United Kingdom: Cambridge University Press, 2000.
[14] K.R. Cronus, E.L. Fung, and L.O. Chua. “Efficient implementation of neighborhood logic for cellular automata via the cellular neural network universal machine. IEEE Transactions on Circuits and Systems–I, vol. 44, pp. 355-361, April 1997.