A Geometric Approach to Properties of the Discrete-Time Cellular Neural Network

Holger Magnussen and Josef A. Nossek, *Fellow, IEEE*

Abstract— **Using the available theory on Linear Threshold Logic, the Discrete-Time Cellular Neural Network (DTCNN) is studied from a geometrical point of view. Different modes of operation are specified. A bound on the number of possible mappings is given for the case of binary inputs. The mapping process in a cell of the network is interpreted in the input space and the parameter space. Worst-case and average-case accuracy conditions are given, and a sufficient worst-case bound on the number of bits required to store the network parameters for the case of binary input signals is derived. Methods for optimizing the robustness of DTCNN parameters for certain regions of the parameter space are discussed.**

I. Introduction

THE Discrete-Time Cellular Neural Network (DTCNN) is a spin glass like architecture with parallel, zero-temperature dynamics, translationally invariant weights and only local interconnections. The DTCNN was introduced in [1] as a discrete-time version of the Cellular Neural Network (CNN) [2], and it is related to the work by Walter Little [3].

The mapping from input images onto output images, which is performed by the DTCNN, is determined by a set of real-valued network parameters. Finding these network parameters for a certain desired task is called learning. The Learning Problem, which is nontrivial due to the local interconnection structure and the recurrence of the DTCNN architecture, is not addressed in this paper.

Each cell of the DTCNN is a Linear Threshold Element (LTE) with a moderate number of inputs. These LTEs, or *Perceptrons* [4], have been thoroughly studied, and a large body of theory is available about them. Efficient algorithms for mapping a set of input images onto desired output values with a single LTE are known [4]–[6]. In the case of the DTCNN, the issue is much more complicated, since the set of input images for each cell not only depends on the global input images of the whole network, but also, due to the recurrence, on the network parameters. This work is concerned with geometrical properties of the DTCNN, which are derived from LTE theory.

The organization of this paper is as follows: The DTCNN is introduced in Section 1, and different modes of operation are specified. Bounds on the number of possible mappings are given. A geometric interpretation, which is very helpful for the understanding of the mapping process performed by one cell of the network, is given in Section 2. Issues of accuracy are very important for hardware realizations. Worst-case and average-case scenarios are discussed in Section 3. Finally, a summary and conclusions are presented in Section 4.

Manuscript received December 20, 1993. Holger Magnussen is supported by a grant from the Ernst-von-Siemens Foundation. This paper was recommended by Associate Editor Mona E. Zaghloul.

The authors are with the Institute for Network Theory and Circuit Design, Technical University Munich, Munich, Germany.

IEEE Log Number 9405335.

DTCNN Equations

The DTCNN is a first-order, discrete-time dynamical system consisting of M identical cells on a (usually) one- or two-dimensional cell grid $\mathcal{CG}$. M input ports are placed on the input grid $\mathcal{IG}$, which has the same dimensions as the cell grid $\mathcal{CG}$. The operation of the DTCNN is described by a state equation

$$x_c(k) = \sum_{d \in \mathcal{N}_Y(c)} a_{d-c} y_d(k) + \sum_{d \in \mathcal{N}_U(c)} b_{d-c} u_d(k) + i \quad (1)$$

an output equation

$$\begin{aligned} y_c(k) &= \mathrm{SGN}(x_c(k-1)) \\ &= \begin{cases} 1 & \text{for } x_c(k-1) \geq 0 \\ -1 & \text{for } x_c(k-1) < 0 \end{cases} \end{aligned} \quad (2)$$

and the initial state

$$y_c(0) = y_{c,0} \quad (3)$$

k is a nonnegative integer corresponding to the time-step. $y_c(k) \in \{-1, 1\}$ is the output of cell c at time-step k. The input signals $u_c(k) \in [-1, 1]$ at time-step k are taken from the input grid $\mathcal{IG}$. The restriction to the interval $[-1, 1]$ is necessary to guarantee the boundedness of the state variables [1]. For convenience, the input signals and the cell output signals are sometimes written in vector notation, i.e. $\mathbf{u}(k) = (u_1(k), \ldots, u_M(k))^{\mathrm{T}}$ and $\mathbf{y}(k) = (y_1(k), \ldots, y_M(k))^{\mathrm{T}}$ with $\mathbf{y}(k) \in \{-1, 1\}^M \ \forall k \in \mathrm{Z}^{0+}$. The initial state $\mathbf{y}_0 = \mathbf{y}(0)$ is either fixed or derived from the input signal $\mathbf{u}(0)$ at time-step $k = 0$ by simple operations. In general, we have $\mathbf{u}(k) \in [-1, 1]^M \ \forall k \in \mathrm{Z}^{0+}$.

$\mathbf{a} = (a_\nu)$ and $\mathbf{b} = (b_\nu)$ are the weighted connections between the cells, i is the cell bias. The network parameters $\mathbf{a}$, $\mathbf{b}$ and i are translationally invariant, i.e. they are identical for each cell in the network. For this reason, $\mathbf{a}$ and $\mathbf{b}$ are also referred to as the *templates*. The "$*_{d-c}$"-notation is used in a symbolical sense, since only the relative position of the two cells c and d with respect to each other, not their absolute location on the cell grid, is important for the value of the template coefficient $*_{d-c}$.

$\mathcal{N}_Y(c)$ and $\mathcal{N}_U(c)$ are the *neighborhoods* of cell c on the cell grid $\mathcal{CG}$ and the input grid $\mathcal{IG}$, respectively. For an r-neighborhood, $\mathcal{N}(c)$ is a square region of dimensions $(2r + 1) \times (2r + 1)$ around the center cell c. Usually, $r = 1$ or $r = 2$. Nonsquare or even asymmetrical neighborhoods as well as different neighborhoods for the $\mathbf{a}$- and $\mathbf{b}$-templates could be used as well. The use of the neighborhoods implies that the cells are only *locally interconnected*. Let N_a and N_b

denote the number of cells in the neighborhoods $\mathcal{N}_Y(c)$ and $\mathcal{N}_U(c)$, respectively. If both the **a**- and **b**-template have the same neighborhood size r (square neighborhoods), each active cell of the network has $N = N_a + N_b = 2(2r+1)^2$ inputs plus a separate bias input. If both the **a**- and **b**-template exist, then N will be even. Cells outside the active grids $\mathcal{CG}$ and $\mathcal{IG}$ are assigned constant values $\varepsilon_U \in [-1,1]$ for the input grid $\mathcal{IG}$ and $\varepsilon_Y \in \{-1,1\}$ for the cell grid $\mathcal{CG}$.

In order to simplify the notational complexity, we will introduce a vector notation for the DTCNN cell. Let as before N be the number of inputs of each cell. Let $\mathbf{p} \in \mathbb{R}^{N+1}$ denote the *parameter vector*. It contains all template coefficients $\mathbf{p}^T = (p_0,\ldots,p_N) = (i,(a_\nu),(b_\nu))$ (in this order). The *cell input vector* $\mathbf{e}_c^T(k)$ collects a constant term for the bias, the cell output signals, and the input signals (in this order) for cell c of the network at time-step k, i.e. $\mathbf{e}_c(k) = (1,(y_{\mathcal{N}_Y(c)}(k)),(u_{\mathcal{N}_U(c)}(k)))$. Note that the indices of $\mathbf{e}$ and $\mathbf{p}$ run from 0 to N. The cell output of cell c at time-step $k+1$ can be written as (cf. (1) and (2))

$$y_c(k+1) = \mathrm{SGN}\big(\mathbf{p}^T\mathbf{e}_c(k)\big) \tag{4}$$

By the definition of linear separability, this choice of outputs $y_c(k+1)$ implies that the mapping is linearly separable. We define the cell input set $\mathcal{E}_0$ and the positive cell input set $\mathcal{E}_0^P$:

Definition 1: [Cell input sets] The *cell input set* is defined by

$$\mathcal{E}_0 := \{\mathbf{e}_c(k) : \forall\, c,k\}$$

It consists of all cell input vectors $\mathbf{e}_c(k)$ ocurring at the input of any active cell at any time-step during the operation of the DTCNN.

Let the SGN function be defined as in (2), and let $\mathbf{p}$ be an arbitrary parameter vector. Then, the *positive cell input set* is given by

$$\mathcal{E}_0^{\mathbf{p}} := \big\{SGN(\mathbf{p}^T\mathbf{e}_c(k))\cdot\mathbf{e}_c(k) \;:\; \mathbf{e}_c(k)\in\mathcal{E}_0\big\}$$

In the rest of this section, the cell index c and the argument k denoting the time-step will be omitted to simplify the notation.

For further studies, it is necessary to distinguish between different operating modes of the DTCNN. With respect to the signal levels, we distinguish between a *continuous mode* with $u_c(k),\varepsilon^U \in [-1,1]$ and a *binary mode* with $u_c(k),\varepsilon^U \in \{-1,1\}$. Let $\mathcal{U} = \{\mathbf{u}(k)\}$ denote the set of input signals presented to the DTCNN during its operation at any time-step k and in any training example. Now, depending on the set of input signals and the parameter vector $\mathbf{p}$, we define

Mode I: No restrictions on the set of input signals, i.e. $\mathcal{U} = [-1,1]^M$ in the continuously-valued input case (''continuous Mode I''), and $\mathcal{U} = \{-1,1\}^M$ in the binary-valued input case (''binary Mode I''). The parameter vector $\mathbf{p}$ can take on arbitrary values.

Mode II: In this case, the set $\mathcal{U}$ is a fixed subset of the corresponding set in Mode I. Let the net be operated for K time-steps. Let L be the number of different input sequences $\mathbf{u}(k)$ (training examples) with $k = 0,\ldots,K-1$. Therefore, the size of $\mathcal{U}$ can be bounded by $|\mathcal{U}| \le LK$. We have $\mathcal{U} \subset [-1,1]^M$ in the continuously-valued case and $\mathcal{U} \subset \{-1,1\}^M$ in the binary-valued case. The parameter vector $\mathbf{p}$ can take on arbitrary values.

Mode III: In Mode III, $\mathcal{U}$ is a fixed subset of $[-1,1]^M$ or $\{-1,1\}^M$ as in Mode II, but the parameter vector $\mathbf{p}$ is fixed at some point $\mathbf{p}_0$. This implies that the output signals $\mathbf{y}(k)$ are explicitly known for each time-step k.

TABLE I
BOUNDS ON $|\mathcal{E}_0|$ FOR DIFFERENT OPERATING MODES

	continuous input	binary input
Mode I	unbounded	$\lvert\mathcal{E}_0\rvert = 2^N$
Mode II	$\lvert\mathcal{E}_0\rvert \le 2^{N_a}\cdot KLM$	$\lvert\mathcal{E}_0\rvert \le \min\{2^{N_a}\cdot KLM, 2^N\}$
Mode III	$\lvert\mathcal{E}_0\rvert \le KLM$	$\lvert\mathcal{E}_0\rvert \le \min\{KLM, 2^N\}$

$|\mathcal{E}_0|$, the cardinality of the cell input set, critically depends on the mode of operation of the DTCNN. Table I summarizes the different bounds on $|\mathcal{E}_0|$ for the different modes of operation. For constant inputs $\mathbf{u}(k) = \mathbf{u}$, we can set $K = 1$ in the formulas in the table. Note that in binary Mode I, we have $\mathcal{E}_0 = \{\mathbf{e} \in \{-1,1\}^{N+1} : e_0 = 1\}$. In Mode III, the inputs and outputs for each cell in the network are known, hence the whole sequence of output patterns $\mathbf{y}(k)$ is known, and the DTCNN reduces to the standard perceptron case.

Further properties of the DTCNN, like stability, boundedness of states, convergence properties etc. can be proven by using the existing theory for synchronously updated two-state Hopfield networks, see for example [7]–[9], and also [1].

Mapping Properties

We can write for the output signal $\mathbf{y}(k)$ of the DTCNN at time-step $k \in \mathrm{Z}^{0+}$

$$\mathbf{y}(k) := G(k,\mathbf{y}_0,\mathbf{u}(0),\mathbf{u}(1),\ldots)$$
$$G : \mathrm{Z}^{0+}\times\{-1,1\}^M\times[-1,1]^{M^{Z^{0+}}} \to \{-1,1\}^M$$

For the special case of a DTCNN in binary Mode I with $L = 1$ constant input $\mathbf{u}(k) = \mathbf{u}$, i.e. only one training example, we can give an upper bound on the number of performable mappings. Assuming that the network is stopped after K time-steps, we have

$$\mathbf{y}(K) := G_b(\mathbf{y}_0,\mathbf{u})$$
$$G_b : \{-1,1\}^M\times\{-1,1\}^M \longrightarrow \{-1,1\}^M$$

Let $\mathcal{G}_b$ be the set of all possible mappings G_b from an initial state and a constant and binary input to the output after a fixed number of time-steps. Then

$$|\mathcal{G}_b| = 2^{M2^{M+M}} = 2^{M4^M} \tag{5}$$

Since the cells of a DTCNN are only locally interconnected and the network parameters are translationally invariant, the number of different realizable mappings is reduced considerably. This is the price which has to be paid for the advantages of simple hardware realizability of the DTCNN. The global

mapping properties of the DTCNN are determined by the local mapping (4) performed by each cell of the network. Each cell performs a linearly separable mapping

$$y_c(k+1) = F_{\text{ls}}(\mathbf{e}_c(k)) = \text{SGN}\big(\mathbf{p}^{\text{T}}\mathbf{e}_c(k)\big)$$
$$F_{\text{ls}} : \mathcal{E}_0 \longrightarrow \{-1, 1\}$$

Let $\mathcal{F}_{\text{ls}}$ be the set of all possible linearly separable mappings F_{ls}. By virtue of the Function-Counting Theorem [10], we can give a bound on $|\mathcal{F}_{\text{ls}}|$, the number of homogenously linearly separable dichotomies of $|\mathcal{E}_0|$ points in $N+1$ space:

$$|\mathcal{F}_{\text{ls}}| \leq 2\sum_{k=0}^{N}\binom{|\mathcal{E}_0|-1}{k}$$

Equality only holds when the $|\mathcal{E}_0|$ points are in *general position*, which is generally not guaranteed in binary Mode I. In this case, we can use a bound due to Winder, which can be found in the appendix of [11].

$$|\mathcal{F}_{\text{ls}}| \leq 2\sum_{k=0}^{N}\binom{2^N-1}{k} \leq \frac{2^{N^2+1}}{N!}$$

Hence the number of different possible mappings of the DTCNN in the binary input case grows with the order $O(\frac{2^{N^2}}{N!})$. Since usually $N \ll M$, this is a big loss when compared to (5). Still, even for simple problems like binary input DTCNNs with a 1-neighborhood in **a**- and **b**-template ($N = 18$), the number of different possible mappings is bounded by approximately $2 \cdot 10^{82}$, and for a 2-neighborhood in **a**- and **b**-template ($N = 50$), by approximately $3 \cdot 10^{688}$. Even if the bound on the number of mappings performable by a DTCNN is the same as in the case of a perceptron with N inputs, the mappings performed by a DTCNN can be much more complex.

II. Geometric Interpretation

Since each cell of a DTCNN is based on a Linear Threshold Element (LTE), the existing theoretical background on threshold logic gives valuable insight [11]–[13]. This section deals with all those operating modes, in which $|\mathcal{E}_0|$ can be bounded by a fixed number, i.e. all modes except continuous Mode I.

Input Space and Parameter Space

Linearly separable functions of N inputs have a very figurative geometric interpretation. Let $\mathcal{I} = \mathbb{R}^{N+1}$ denote the *input space*. The $|\mathcal{E}_0|$ cell input vectors $\mathbf{e} \in \mathcal{E}_0$ correspond to $|\mathcal{E}_0|$ points in $\mathcal{I}$. Let those points $\mathbf{e}$, for which the desired output is $+1$, be labelled "+", and those points, for which the desired output is -1, be labelled "$-$". The equation $\mathbf{p}^{\text{T}}\mathbf{e} = 0$ defines a hyperplane through the origin of $\mathcal{I}$. It divides the set of vertices into two disjoint sets, one set, for which $\mathbf{p}^{\text{T}}\mathbf{e} \geq 0$, and a set with $\mathbf{p}^{\text{T}}\mathbf{e} < 0$. Thus only those dichotomies are possible, where the "+" points can be separated from the "$-$" points by a hyperplane (linearly separable mapping).

The space which is dual to $\mathcal{I}$ with respect to the Euclidian norm is more useful to visualize the mapping process of a DTCNN cell. Let $\mathcal{P} = \mathbb{R}^{N+1}$ denote the *parameter space*. The parameter vector $\mathbf{p} \in \mathcal{P}$ corresponds to a point in $\mathcal{P}$. Each of the $|\mathcal{E}_0|$ equations $\mathbf{e}^{\text{T}}\mathbf{p} = 0$ with $\mathbf{e} \in \mathcal{E}_0$ defines a hyperplane through the origin of $\mathcal{P}$. The binary-valued response of a cell to a cell input vector $\mathbf{e}$ depends on which side of the corresponding hyperplane the point $\mathbf{p}$ is located, either $\mathbf{e}^{\text{T}}\mathbf{p} \geq 0$ or $\mathbf{e}^{\text{T}}\mathbf{p} < 0$. Each of these hyperplanes divides $\mathcal{P}$ into two half spaces (convex sets). The intersection of a finite number of half spaces is called a *convex cone* $\mathcal{C}_{\mathbf{p}}$:

Definition 2: [Convex cone] Let $\mathbf{p} \in \mathcal{P}$ so that $\mathbf{e}^{\text{T}}\mathbf{p} \neq 0 \ \forall \ \mathbf{e} \in \mathcal{E}_0$. The set

$$\mathcal{C}_{\mathbf{p}} = \big\{\bar{\mathbf{p}} \in \mathcal{P} \ : \ \mathbf{e}^{\text{T}}\mathbf{p} \cdot \mathbf{e}^{\text{T}}\bar{\mathbf{p}} > 0 \ \forall \ \mathbf{e} \in \mathcal{E}_0\big\} \tag{6}$$

is called a *convex cone* $\mathcal{C}_{\mathbf{p}}$.

Note that from this definition and the definition of the positive cell input set $\mathcal{E}_0^{\mathbf{P}}$, we have $\mathbf{e}^{\text{T}}\bar{\mathbf{p}} > 0$ for all $\mathbf{e} \in \mathcal{E}_0^{\mathbf{P}}$ and $\bar{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}$. The hyperplanes are excluded in Definition 2, because parameter vectors $\bar{\mathbf{p}}$ on the hyperplanes require an infinite accuracy of the network parameters (see Section 3). This exclusion is legal, since without loss of generality, an infinitesimal perturbation can be added to the parameter vector, so that $\mathbf{e}^{\text{T}}\bar{\mathbf{p}} > 0$ and the functionality of the network remains unchanged [9]. Each convex cone $\mathcal{C}_{\mathbf{p}}$ corresponds to a linearly separable Boolean function, since the cell output values are identical for all $\bar{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}$. The mapping $\{\mathcal{C}_{\mathbf{p}}\} \longrightarrow \mathcal{F}_{\text{ls}}$ is injective, but the mapping $\mathcal{P} \longrightarrow \mathcal{F}_{\text{ls}}$ is not, since each linearly separable Boolean function can be realized by infinitely many parameter vectors $\bar{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}$.

Properties of the Parameter Space Description

It is important to note that from Definitions 1 and 2 we can conclude

Lemma 1: [Identical behavior in a convex cone] Let $\mathcal{C}_{\mathbf{p}}$ denote a convex cone. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two identical DTCNNs with different parameter vectors $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{C}_{\mathbf{p}}$. Then $\mathcal{D}_1$ and $\mathcal{D}_2$ will go through the same sequence of output signals $\mathbf{y}(k)$, if both nets are operated with identical inputs $\mathbf{u}(k)$ and initial states $\mathbf{y}_0$.

Proof: Trivial, by induction. Both DTCNNs start from the same initial state $\mathbf{y}_0$. From Definition 2 we have identical cell outputs $\text{SGN}(\mathbf{e}^{\text{T}}\mathbf{p}_1) = \text{SGN}(\mathbf{e}^{\text{T}}\mathbf{p}_2)$ for all identical cell inputs $\mathbf{e} \in \mathcal{E}_0$ and all $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{C}_{\mathbf{p}}$. Hence the two DTCNNs will perform identical mappings from one time-step to the next, and thus the two nets behave identically. ■

Obviously, only a subset of all hyperplanes $\mathbf{e}^{\text{T}}\mathbf{p} = 0$ actually bound a convex cone ("bounding hyperplane"), while others only touch the cone in the origin of $\mathcal{P}$.

Definition 3: [Bounding hyperplane] Let $\mathcal{C}_{\mathbf{p}}$ be a convex cone. A hyperplane $\mathbf{e} \in \mathcal{E}_0^{\mathbf{P}}$ is called *bounding hyperplane* of the convex cone $\mathcal{C}_{\mathbf{p}}$, if and only if

$$\exists \bar{\mathbf{p}} \in \mathcal{P} \ : \ \bar{\mathbf{p}}^{\text{T}}\mathbf{e} \leq 0 \text{ and } \bar{\mathbf{p}}^{\text{T}}\mathbf{e}_\nu > 0 \ \forall \mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$$

For each convex cone $\mathcal{C}_{\mathbf{p}}$, the set $\mathcal{E}_0^{\mathbf{P}}$ can be divided up into two complementary subsets $\mathcal{E}_1^{\mathbf{P}} \subseteq \mathcal{E}_0$ and $\mathcal{E}_2^{\mathbf{P}} = \mathcal{E}_0^{\mathbf{P}} \backslash \mathcal{E}_1^{\mathbf{P}}$. Let

$$\mathcal{E}_1^{\mathbf{P}} := \{\mathbf{e} \in \mathcal{E}_0^{\mathbf{P}} \text{ is a bounding hyperplane of } \mathcal{C}_{\mathbf{p}}\}$$

To explore some properties of the two subsets $\mathcal{E}_1^{\mathbf{P}}$ and $\mathcal{E}_2^{\mathbf{P}}$ and to ease the following proofs, we introduce Motzkin's Theorem [14]. In the Theorem, the ordering relations $\mathbf{u} \geq \mathbf{v}$ and $\mathbf{u} \gneqq \mathbf{v}$ between two vectors $\mathbf{u}$ and $\mathbf{v}$ both imply that $u_i \geq v_i$, but $\mathbf{u} \geq \mathbf{v}$ additionally implies $\mathbf{u} \neq \mathbf{v}$.

Theorem 1: [Motzkin's Theorem of the Alternative] Let $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ be given matrices, with $\mathbf{A}$ being nonvacuous. Then either

$$(I) \quad \exists \mathbf{x} \; : \; \mathbf{A}\mathbf{x} > \mathbf{0} \quad \mathbf{B}\mathbf{x} \geq \mathbf{0} \quad \mathbf{C}\mathbf{x} = \mathbf{0}$$

or

$$(II) \quad \exists \mathbf{y}_1 \geq \mathbf{0}, \; \mathbf{y}_2 \gtreqless \mathbf{0}, \; \mathbf{y}_3 \; : \; \mathbf{A}^{\mathrm{T}}\mathbf{y}_1 + \mathbf{B}^{\mathrm{T}}\mathbf{y}_2 + \mathbf{C}^{\mathrm{T}}\mathbf{y}_3 = \mathbf{0}$$

but never both.

Proof: See for example [14]. ■

Corollary 1: [Linear Combination] A hyperplane $\mathbf{e} \in \mathcal{E}_0^{\mathbf{P}}$ belongs to the set $\mathcal{E}_1^{\mathbf{P}}$, if and only if

$$\nexists \lambda_\nu \in \mathbb{R}^{0+} : \mathbf{e} = \sum_\nu \lambda_\nu \mathbf{e}_\nu \; \forall \mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$$

i.e. $\mathbf{e}$ cannot be written as a nontrivial positive linear combination of vectors $\mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$.

Proof: The proof is trivial using Theorem 1. Let $\mathbf{A}$ be a $(|\mathcal{E}_0^{\mathbf{P}}| - 1) \times (N+1)$ matrix, where the rows of $\mathbf{A}$ are equal to the vectors $\mathbf{e}_\nu^{\mathrm{T}} \in \mathcal{E}_0 \backslash \{\mathbf{e}\}$. Let $\mathbf{B} = -\mathbf{e}^{\mathrm{T}}$ be an $1 \times (N+1)$ matrix, and let $\mathbf{C} = \mathbf{0}$.

"if": If $\mathbf{e} \in \mathcal{E}_1^{\mathbf{P}}$, then there is a solution for (I) in Theorem 1, which implies that there is not solution for (II), i.e. $\mathbf{e}$ cannot be written as a nontrivial positive linear combination.

"only if": If $\mathbf{e} \notin \mathcal{E}_1^{\mathbf{P}}$, then we have $\mathbf{e}^{\mathrm{T}}\mathbf{p} > 0$ for all $\mathbf{p} \in \mathcal{P}$ satisfying $\mathbf{e}_\nu^{\mathrm{T}}\mathbf{p} > 0$ for all $\mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$. It follows that there is no solution for (I) in Theorem 1, and hence there must be a solution for (II), i.e. there exist a nonnegative value λ and nonnegative values λ_ν with at least one strictly positive λ_ν, so that

$$\lambda \mathbf{e} = \sum_\nu \lambda_\nu \mathbf{e}_\nu \quad \forall \mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$$

If $\lambda > 0$, then we are done after dividing both sides by λ. The fact that $\lambda > 0$ will be shown by contradiction. Assume that $\lambda = 0$. Then there is a nontrivial solution $\lambda_\nu \geq 0$, so that

$$0 \cdot \mathbf{e}^{\mathrm{T}}\tilde{\mathbf{p}} = \sum_\nu \lambda_\nu \mathbf{e}_\nu^{\mathrm{T}}\tilde{\mathbf{p}} = 0 \quad \forall \tilde{\mathbf{p}} \in \mathcal{P}, \; \mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$$

This is a contradiction to the fact that for all $\tilde{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}$, we have $\mathbf{e}_\nu^{\mathrm{T}}\tilde{\mathbf{p}} > 0$. Thus such a solution with $\lambda = 0$ cannot exist and λ is strictly positive. This completes the proof. ■

Lemma 2: [Reduced linear combination] If for each cell input vector $\mathbf{e} \in \mathcal{E}_2^{\mathbf{P}}$ there is a positive linear combination, i.e. there are nontrivial $\lambda_\nu \geq 0$ so that

$$\mathbf{e} = \sum_\nu \lambda_\nu \mathbf{e}_\nu \qquad \forall \mathbf{e}_\nu \in \mathcal{E}_0 \backslash \{\mathbf{e}\}$$

then there are nontrivial $\bar{\lambda}_\nu \geq 0$ so that

$$\mathbf{e} = \sum_\nu \bar{\lambda}_\nu \bar{\mathbf{e}}_\nu \qquad \forall \bar{\mathbf{e}}_\nu \in \mathcal{E}_1^{\mathbf{P}}$$

Proof: Let $\bar{\mathbf{e}} \in \mathcal{E}_1^{\mathbf{P}}$ and $\hat{\mathbf{e}} \in \mathcal{E}_2^{\mathbf{P}}$. By Corollary 1 there are nontrivial $\bar{\lambda}_\nu^\mu \geq 0$ and $\hat{\lambda}_m^\mu \geq 0$ so that for any $\hat{\mathbf{e}}_1 \in \mathcal{E}_2^{\mathbf{P}}$ we can write

$$\hat{\mathbf{e}}_1 = \sum_\nu \bar{\lambda}_\nu^1 \bar{\mathbf{e}}_\nu + \sum_{m \neq 1,\mu} \hat{\lambda}_m^1 \hat{\mathbf{e}}_m + \hat{\lambda}_\mu^1 \hat{\mathbf{e}}_\mu$$

Now pick a vector $\hat{\mathbf{e}}_\mu \in \mathcal{E}_2^{\mathbf{P}} \backslash \{\hat{\mathbf{e}}_1\}$ with $\mu \neq 1$. Again, we have

$$\begin{aligned} \hat{\mathbf{e}}_\mu = & \sum_\nu \bar{\lambda}_\nu^\mu \bar{\mathbf{e}}_\nu + \sum_{m \neq 1,\mu} \hat{\lambda}_m^\mu \hat{\mathbf{e}}_m + \hat{\lambda}_1^\mu \hat{\mathbf{e}}_1 = \\ = & \sum_\nu \bar{\lambda}_\nu^\mu \bar{\mathbf{e}}_\nu + \sum_{m \neq 1,\mu} \hat{\lambda}_m^\mu \hat{\mathbf{e}}_m + \\ + & \hat{\lambda}_1^\mu \sum_\nu \bar{\lambda}_\nu^1 \bar{\mathbf{e}}_\nu + \hat{\lambda}_1^\mu \sum_{m \neq 1,\mu} \hat{\lambda}_m^1 \hat{\mathbf{e}}_m + \hat{\lambda}_1^\mu \hat{\lambda}_\mu^1 \hat{\mathbf{e}}_\mu \end{aligned}$$

Now we can write

$$\begin{aligned} & \hat{\mathbf{e}}_\mu \left(1 - \hat{\lambda}_1^\mu \hat{\lambda}_\mu^1\right) = \\ & = \sum_\nu \underbrace{\left(\bar{\lambda}_\nu^\mu + \hat{\lambda}_1^\mu \bar{\lambda}_\nu^1\right)}_{=:\tilde{\lambda}_\nu \geq 0} \bar{\mathbf{e}}_\nu + \sum_{m \neq 1,\mu} \underbrace{\left(\hat{\lambda}_m^\mu + \hat{\lambda}_1^\mu \hat{\lambda}_m^1\right)}_{=:\tilde{\lambda}_m \geq 0} \hat{\mathbf{e}}_m \end{aligned}$$

Since we know that for all $\tilde{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}$ and $\mathbf{e} \in \mathcal{E}_0^{\mathbf{P}}$ the inequalities $\tilde{\mathbf{p}}^{\mathrm{T}}\mathbf{e} > 0$ are satisfied, we can conclude that

$$\left(1 - \hat{\lambda}_1^\mu \hat{\lambda}_\mu^1\right) > 0$$

and hence there is a positive linear combination for all $\hat{\mathbf{e}}_\mu \in \mathcal{E}_2^{\mathbf{P}}$ which does not contain $\hat{\mathbf{e}}_1$. By recursively executing the above procedure, all vectors $\hat{\mathbf{e}}_\nu \in \mathcal{E}_2^{\mathbf{P}}$ can be eliminated, and thus a positive linear combination for each $\hat{\mathbf{e}}_\mu \in \mathcal{E}_2^{\mathbf{P}}$ can be found, which only depends on the $\bar{\mathbf{e}}_\nu \in \mathcal{E}_1^{\mathbf{P}}$. This completes the proof. ■

Theorem 2: [Sufficiency of bounding hyperplanes]

$$(\mathbf{e} \in \mathcal{E}_2^{\mathbf{P}}) \Rightarrow \left((\forall \mathbf{p} \in \mathcal{P} : \bar{\mathbf{e}}^{\mathrm{T}}\mathbf{p} > 0 \; \forall \bar{\mathbf{e}} \in \mathcal{E}_1^{\mathbf{P}}) \Rightarrow \mathbf{e}^{\mathrm{T}}\mathbf{p} > 0\right)$$

Proof: From Corollary 1 and Lemma 2 follows that each $\mathbf{e} \in \mathcal{E}_2^{\mathbf{P}}$ can be written as a nontrivial positive linear combination of vectors $\bar{\mathbf{e}} \in \mathcal{E}_1^{\mathbf{P}}$. From $\bar{\mathbf{e}}^{\mathrm{T}}\mathbf{p} > 0$, it then follows by multiplying the positive linear combination with $\mathbf{p}^{\mathrm{T}}$ that $\mathbf{e}^{\mathrm{T}}\mathbf{p}$ cannot be zero or negative, and thus $\mathbf{e}^{\mathrm{T}}\mathbf{p} > 0$. ■

This last theorem implies that a specification of the cell output $\bar{y}$ for all cell input vectors $\bar{\mathbf{e}} \in \mathcal{E}_1^{\mathbf{P}}$ automatically sets the cell output for the cell input vectors $\hat{\mathbf{e}} \in \mathcal{E}_2^{\mathbf{P}}$.

In the case of binary Mode I, a statement concerning the tightness of the upper bound on $|\mathcal{E}_1^{\mathbf{P}}|$ can be made:

Corollary 2: [Tightness of bound] In binary Mode I, i.e. if $\mathcal{E}_0 = \{\mathbf{e} \in \{-1, 1\}^{N+1} : e_0 = 1\}$, the bound

$$|\mathcal{E}_1^{\mathbf{P}}| \leq 2^N$$

is tight.

Proof: Equality will by shown by providing an example for which $\mathcal{E}_2^{\mathbf{P}} = \emptyset$. Examine the convex cone $\mathcal{C}_{\mathbf{p}}$ with $\mathbf{p} = (1, 0, \ldots, 0)^{\mathrm{T}}$. In this case, all vectors $\mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}}$ will have "+1" as their first component. Assume that $\mathcal{E}_2^{\mathbf{P}}$ is nonempty. Then by Corollary 1 all vectors $\mathbf{e} \in \mathcal{E}_2^{\mathbf{P}}$ can be written as a positive linear combination of vectors $\mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}}$, i.e.

$$\mathbf{e} = \sum_\nu \lambda_\nu \mathbf{e}_\nu \; , \quad \lambda_\mu \geq 0, \; \mathbf{e}_\nu \in \mathcal{E}_0^{\mathbf{P}} \backslash \{\mathbf{e}\}$$

The equation for the first component of $\mathbf{e}$ will be

$$\sum_\nu \lambda_\nu = 1$$

Since for any other element but the first element the summation will contain "+1"-entries as well as "−1"-entries, it is impossible to obtain a vector with binary entries as the result

of the summation. Thus there is no such positive linear combination, and hence $\mathcal{E}_2^{\mathbf{P}}$ must be empty and $|\mathcal{E}_1^{\mathbf{P}}| = 2^N$.

We will continue by providing an example where $\mathcal{E}_2^{\mathbf{P}}$ is nonempty. Examine the convex cone $\mathcal{C}_{\mathbf{P}}$ with $\mathbf{p} = (1.01, 1, \ldots, 1)^{\mathrm{T}}$. Then the $\mathcal{E}_0^{\mathbf{P}}$ contains all $N+1$ vectors with N "+1" elements and one "−1" element. Adding these vectors and dividing the sum by $N-2$ will result in a vector consisting of only "+1" elements, which belongs to $\mathcal{E}_0^{\mathbf{P}}$ as well. Thus the vector $\mathbf{e} = (1, \ldots, 1)^{\mathrm{T}}$ can be written as a positive linear combination of other vectors, and thus $|\mathcal{E}_1^{\mathbf{P}}| < 2^N$. ■

At the end of this section, we will provide a simple example for $N = 3$. We have $\mathcal{E}_0 = \{(1, \pm 1, \pm 1, \pm 1)^{\mathrm{T}}\}$ with $|\mathcal{E}_0| = 8$. Now let $\mathbf{p} = (1.01, 1, 1, 1)^{\mathrm{T}}$. In this case, the sets $\mathcal{E}_0^{\mathbf{P}}$, $\mathcal{E}_1^{\mathbf{P}}$, and $\mathcal{E}_2^{\mathbf{P}}$ are

$$\left.\begin{array}{ll} \mathbf{e}_1 = (& -1,\ 1,\ 1,\ 1)^{\mathrm{T}} \\ \mathbf{e}_2 = (& 1,\ 1,\ -1,\ -1)^{\mathrm{T}} \\ \mathbf{e}_3 = (& 1,\ -1,\ 1,\ -1)^{\mathrm{T}} \\ \mathbf{e}_4 = (& 1,\ -1,\ -1,\ 1)^{\mathrm{T}} \end{array}\right\} \mathcal{E}_1^{\mathbf{P}}$$

$$\left.\begin{array}{ll} \mathbf{e}_5 = (& 1,\ 1,\ 1,\ 1)^{\mathrm{T}} \\ \mathbf{e}_6 = (& 1,\ 1,\ 1,\ -1)^{\mathrm{T}} \\ \mathbf{e}_7 = (& 1,\ 1,\ -1,\ 1)^{\mathrm{T}} \\ \mathbf{e}_8 = (& 1,\ -1,\ 1,\ 1)^{\mathrm{T}} \end{array}\right\} \mathcal{E}_2^{\mathbf{P}}$$

$$\Big\} \mathcal{E}_0^{\mathbf{P}}$$

III. Accuracy Requirements

For any realization of a DTCNN, the actual network parameters will deviate from the nominal parameters. Let $\mathbf{p}$ denote the nominal parameter vector, $\hat{\mathbf{p}}$ the actual parameter vector, and $\Delta\mathbf{p} = \mathbf{p} - \hat{\mathbf{p}}$ the error. The desired operation of the network can only be guaranteed by Lemma 1 as long as $\hat{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}$, hence from Definition 2

$$\mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}}\hat{\mathbf{p}} > 0 \ \forall \mathbf{e} \in \mathcal{E}_0 \tag{7}$$

is a sufficient condition for the desired operation of the network. Further, let $x_{\min}$ be the *minimum absolute cell state* given by

$$x_{\min} := \min_{\mathbf{e} \in \mathcal{E}_0} \left\{ \left| \mathbf{e}^{\mathrm{T}}\mathbf{p} \right| \right\} \tag{8}$$

Worst-Case Bound

Assume that the network parameters $\mathbf{p} = (p_0, \ldots, p_N)^{\mathrm{T}}$ can be realized with a guaranteed error limit $\Delta p_\nu^{\max}$, where $|\Delta p_\nu| \le \Delta p_\nu^{\max}$.

Lemma 3: [Worst-case bound] Let $|\Delta p_\nu| \le \Delta p_\nu^{\max} = \alpha_1 |p_\nu|$ for all $\nu = 0, \ldots, N$ and a real-valued, positive *accuracy* α_1. Let $x_{\min}$ be defined as in (8). A sufficient condition for correct operation of the network, i.e. for (7) is

$$\alpha_1 < \frac{x_{\min}}{\|\mathbf{e}\|_\infty \cdot \|\mathbf{p}\|_1} \Bigg|_{\forall \mathbf{e} \in \mathcal{E}_0} \le r_w(1, \mathbf{p} := \min_{\mathbf{e} \in \mathcal{E}_0} \left\{ \frac{|\mathbf{e}^{\mathrm{T}}\mathbf{p}|}{\|\mathbf{e}\|_\infty \|\mathbf{p}\|_1} \right\} \tag{9}$$

Proof: We have

$$\alpha_1 \cdot \|\mathbf{e}\|_\infty \cdot \|\mathbf{p}\|_1 = \alpha_1 \cdot \|\mathbf{e}\|_\infty \sum_{\nu=0}^{N} |p_\nu| =$$

$$= \max_\nu \{e_\nu\} \sum_{\nu=0}^{N} \Delta p_\nu^{\max} \ge \sum_{\nu=0}^{N} |e_\nu| |\Delta p_\nu| \ge \left| \mathbf{e}^{\mathrm{T}} \Delta \mathbf{p} \right|$$

Now

$$\left| \mathbf{e}^{\mathrm{T}}\mathbf{p} \right| \ge x_{\min} > \left| \mathbf{e}^{\mathrm{T}} \Delta \mathbf{p} \right|$$
$$\left(\mathbf{e}^{\mathrm{T}}\mathbf{p} \right)^2 = \left| \mathbf{e}^{\mathrm{T}}\mathbf{p} \right|^2 > \left| \mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}} \Delta \mathbf{p} \right| \ge \mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}} \Delta \mathbf{p}$$
$$\left(\mathbf{e}^{\mathrm{T}}\mathbf{p} \right)^2 - \mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}} \Delta \mathbf{p} = \mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}} \hat{\mathbf{p}} > 0$$

■

Note that in (9), $r_w(1, \mathbf{p})$ is exactly the *(relative) robustness in weight space with respect to the 1-norm* as defined by Nachbar in [15].

Average-Case Bound

For this subsection, we will assume that the network parameters are random variables with a Gaussian probability distribution. Let $\chi = (\chi_0, \ldots, \chi_N)^{\mathrm{T}}$ be a vector with $N+1$ random variables with Gaussian probability distributions characterized by the expected values $E_{\chi_n} = p_n$ (the nominal network parameters) and the (invertible and symmetric) covariance matrix $\mathbf{C}$. Let $p(\chi)$ be the joint probability density function of the random variables, which obeys

$$p(\chi) = \frac{1}{\sqrt{2\pi}^{N+1} \sqrt{\det \mathbf{C}}} \cdot e^{-\frac{1}{2}\left((\chi - \mathbf{p})^{\mathrm{T}} \mathbf{C}^{-1} (\chi - \mathbf{p}) \right)} \tag{10}$$

Now let P_{ok} be the probability that the network is functioning correctly. Then

$$P_{\mathrm{ok}} = \underbrace{\int \cdots \int}_{\chi \in \mathcal{C}_{\mathbf{p}}} p(\chi)\, d^{N+1}\chi \tag{11}$$

Fig. 1(a) shows the two-dimensional parameter space for the simple case $N = 1$. The shaded area corresponds to the chosen convex cone. In practice, the integral over $\mathcal{C}_{\mathbf{p}}$ is very difficult to evaluate, because the number of bounding hyperplanes is very large in general, and in binary Mode I, it can even grow exponentially with N (see Corollary 2). Still, it is possible to give a lower bound on P_{ok}.

Let $\mathbf{Q}$ be a symmetric $(N+1) \times (N+1)$ matrix so that $\mathbf{C} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q}$. Since $\mathbf{C}$ is a covariance matrix, such a $\mathbf{Q}$ will exist, and it will be invertible. We introduce the norm body $\mathcal{NB}(\mathbf{p}, \beta)$

$$\mathcal{NB}(\mathbf{p}, \beta) = \{ \hat{\mathbf{p}} \in \mathcal{P} \ : \ \hat{\mathbf{p}} = \mathbf{p} - \Delta\mathbf{p},\ \|\Delta\mathbf{p}\|_{\mathbf{C}^{-1}} \le \beta \}$$

The elliptical norm $\|\Delta\mathbf{p}\|_{\mathbf{C}^{-1}}$ is given by

$$\|\Delta\mathbf{p}\|_{\mathbf{C}^{-1}} = \sqrt{\Delta\mathbf{p}^{\mathrm{T}} \mathbf{C}^{-1} \Delta\mathbf{p}}$$

$\mathcal{NB}(\mathbf{p}, \beta)$ is a hyper-ellipsoid, which is introduced, because a closed-form expression can be given for the integration of (10) over $\mathcal{NB}(\mathbf{p}, \beta)$.

Lemma 4: [Average-case bound] A sufficient condition for $\mathcal{NB}(\mathbf{p}, \beta) \subset \mathcal{C}_{\mathbf{p}}$ is

$$\beta < \beta_{\max} = \min_{\mathbf{e} \in \mathcal{E}_0} \left\{ \frac{|\mathbf{e}^{\mathrm{T}}\mathbf{p}|}{\|\mathbf{Q}\mathbf{e}\|_2} \right\} \tag{12}$$

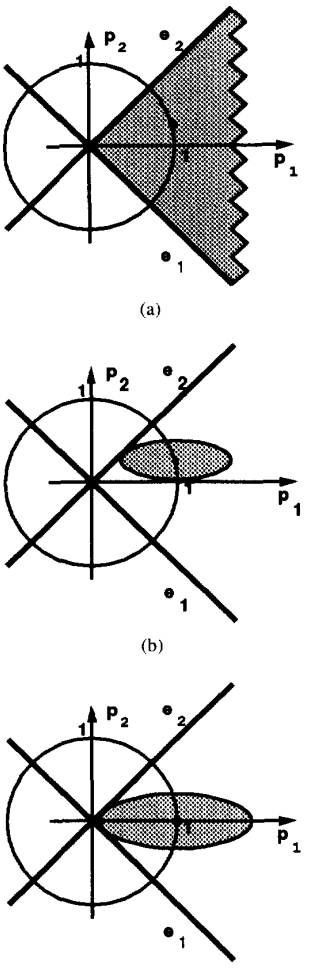


Fig. 1. Parameter space for $N+1 = 2$: (a) convex cone; (b) inscribed norm body $\mathcal{NB}(\mathbf{p}, \beta_{\max})$ (shaded ellipsoid); c) optimal inscription of $\mathcal{NB}(\mathbf{p}^{\text{opt}}, \beta_{\max}^{\text{opt}})$

Proof: We show that from $\mathbf{p} \in \mathcal{NB}(\mathbf{p}, \beta)$, it follows that $\mathbf{p} \in \mathcal{C}_{\mathbf{p}}$. Using the Hoelder inequality for the Euclidian norm, we can write

$$\begin{aligned} |\mathbf{e}^{\mathrm{T}}\Delta\mathbf{p}| = \; & |\mathbf{e}^{\mathrm{T}}(\mathbf{Q}^{-1}\mathbf{Q})^{\mathrm{T}}\Delta\mathbf{p}| \le \|\mathbf{Qe}\|_2 \cdot \|\mathbf{Q}^{-1,\mathrm{T}}\Delta\mathbf{p}\|_2 \\ \le \; & \|\mathbf{Qe}\|_2 \cdot \beta < |\mathbf{e}^{\mathrm{T}}\mathbf{p}| \end{aligned}$$

Then,

$$\begin{aligned} &\mathbf{e}^{\mathrm{T}}\Delta\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}}\mathbf{p} \le |\mathbf{e}^{\mathrm{T}}\Delta\mathbf{p}| \cdot |\mathbf{e}^{\mathrm{T}}\mathbf{p}| < |\mathbf{e}^{\mathrm{T}}\mathbf{p}|^2 = (\mathbf{e}^{\mathrm{T}}\mathbf{p})^2 \\ &\mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}}(\mathbf{p} - \Delta\mathbf{p}) = \mathbf{e}^{\mathrm{T}}\mathbf{p} \cdot \mathbf{e}^{\mathrm{T}}\hat{\mathbf{p}} > 0 \end{aligned}$$

■

Let the σ_ν be the standard deviations of the random variables. In the case of binary inputs, and when additionally the random variables χ_ν are mutually uncorrelated, i.e. $\mathbf{Q} = \operatorname{diag}(\sigma_0, \ldots, \sigma_N)$, then we can further simplify (12) using

$$\|\mathbf{Qe}\|_2^2 = \sum_{\nu=0}^{N} \sigma_\nu^2 e_\nu^2 = \sum_{\nu=0}^{N} \sigma_\nu^2 := \bar{\sigma}^2$$

With the assumption $\sigma_\nu = \alpha_2 |p_\nu|$, we get from (12) and (8)

$$\beta < \beta_{\max} = \frac{x_{\min}}{\bar{\sigma}} = \frac{x_{\min}}{\alpha_2 \|\mathbf{p}\|_2} \tag{13}$$

Fig. 1(b) shows the norm body $\mathcal{NB}(\mathbf{p}, \beta_{\max})$. If (12) is satisfied, then we get from (11) and the nonnegativity of $p(\chi)$

$$P_{\text{ok}} > \underbrace{\int \cdots \int}_{\mathcal{NB}(\mathbf{p}, \beta_{\max})} p(\chi)\, d^{N+1}\chi \tag{14}$$

The integral in this last expression can be evaluated. The derivation is given in Appendix I. We obtain for even N

$$\begin{aligned} P_{\text{ok}} > \; & P_N(\beta_{\max}) := \operatorname{erf}\left(\tfrac{1}{\sqrt{2}}\beta_{\max}\right) - \\ & - \sqrt{\tfrac{2}{\pi}} \cdot e^{-\frac{1}{2}\beta_{\max}^2} \sum_{l=0}^{\frac{N}{2}-1} \frac{\beta_{\max}^{2l+1}}{1 \cdot 3 \cdot \ldots \cdot (2l+1)} \end{aligned} \tag{15}$$

For $N = 0$, expression (15) reduces to the standard Gaussian error integral. Fig. 2 shows $P_N(\beta)$ for values of $N = 0$ (standard Gaussian error integral), $N = 18$ (DTCNN with 1-neighborhood), and $N = 50$ (2-neighborhood). From (13) and (15) we finally get a lower bound on P_{ok}, the probability that the network is functioning correctly:

$$P_{\text{ok}} > P_N\left(\frac{x_{\min}}{\alpha_2 \cdot \|\mathbf{p}\|_2}\right) \tag{16}$$

For $\beta_{\max} \ge 6$ and a 1-neighborhood, the cell will work with a probability of more than 98.9%. In this case, we have

$$\alpha_2 < \frac{x_{\min}}{6 \cdot \|\mathbf{p}\|_2} \tag{17}$$

Computation of $x_{\min}$

The value of $x_{\min}$ in (9) and (16) is still unknown. In continuous Mode I, its definition does not make sense, because $|\mathcal{E}_0|$ is not bounded. In addition, it is easily possible, apart from trivial cases, to choose input values $\mathbf{u}$ such that $x_{\min} = 0$, which would require an infinite precision (see (9) and (17)). In Modes II and III, $x_{\min}$ can usually be computed by running the network and keeping track of the values of $x_c(k)$. In binary Mode I, this is not always possible, since $|\mathcal{E}_0| = 2^N$ grows exponentially with N. Thus the rest of this subsection deals exclusively with binary Mode I, where $|\mathcal{E}_0| = 2^N$. It turns out that even the apparently simple problem of determining the hyperplane $\mathbf{e}_{\mathbf{p}} \in \mathcal{E}_0$, for which $|\mathbf{e}_{\mathbf{p}}^{\mathrm{T}}\mathbf{p}|$ is minimal, is inherently difficult:

Theorem 3: [Complexity of finding a closer hyperplane] The problem of determining whether there is a hyperplane $\mathbf{e}_{\mathbf{p}} \in \mathcal{E}_0$ for a given parameter vector $\mathbf{p}$ with rational elements, which obeys

$$\left|\mathbf{p}^{\mathrm{T}}\mathbf{e}_{\mathbf{p}}\right| \le \epsilon$$

where $\epsilon > 0$ is a rational constant, is NP-complete.

Note that in order to find out whether for a certain hyperplane $\mathbf{e}_{\mathbf{p}}^0 \in \mathcal{E}_0$ the scalar product $|\mathbf{p}^{\mathrm{T}}\mathbf{e}_{\mathbf{p}}^0|$ is minimal, one has to decide whether there is another hyperplane $\mathbf{e}_{\mathbf{p}} \in \mathcal{E}_0$ with $|\mathbf{p}^{\mathrm{T}}\mathbf{e}_{\mathbf{p}}| \le |\mathbf{p}^{\mathrm{T}}\mathbf{e}_{\mathbf{p}}^0|$.

Proof: see Appendix II. ■

Remark: In Appendix II it is actually shown that the problem from Theorem 3 and the Value-Independent Knapsack are polynomially equivalent. Thus pseudo-polynomial algorithms like Branch-and-Bound methods or Cutting-Plane algorithms (see for example [16] or [17] for an overview) can be used to solve instances of the above problem in practice.

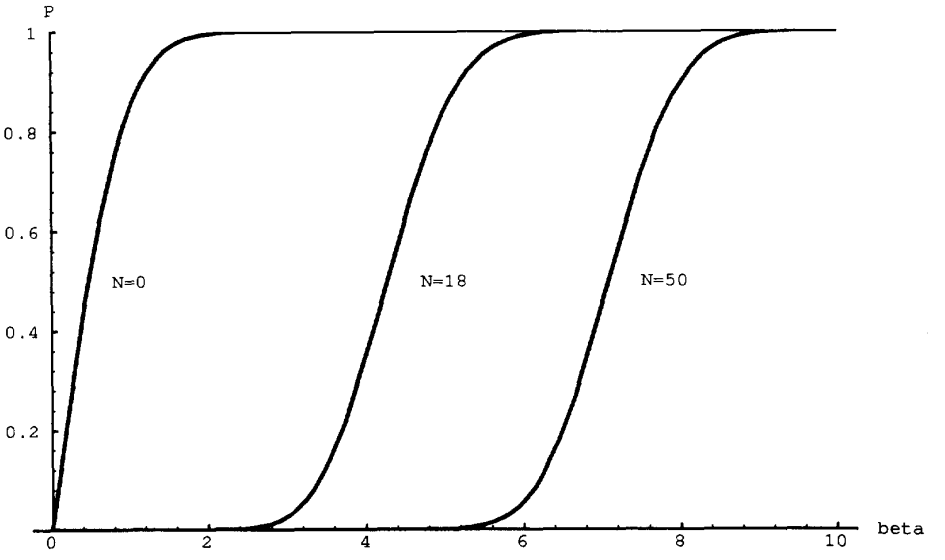


Fig. 2. Probability $P_N(\beta)$

Still, a lower bound on $x_{\min}$ depending on N can be found. Since we have $\mathbf{e}^T\tilde{\mathbf{p}} \neq 0$ for all $\tilde{\mathbf{p}} \in \mathcal{C}_\mathbf{p}$ and all $\mathbf{e} \in \mathcal{E}_0$, we have $x_{\min} > 0$. Now let

$$\tilde{p}_\nu := \frac{p_\nu}{x_{\min}} \qquad \nu = 0, 1, \ldots, N \tag{18}$$

Then we have

$$\left|\mathbf{e}^T\tilde{\mathbf{p}}\right| \geq 1 \quad \forall \mathbf{e} \in \mathcal{E}_0$$

Using Muroga's terminology, this corresponds to the *normalized system of inequalities in a majority expression* (Definition 3.2.2 in [13]). For any positive linearly separable Boolean function of N variables, a bound on the sum of the weights can be given

Theorem 4: [Bound on the weight sum]

$$0 \leq \sum_{\nu=0}^{N} \tilde{p}_\nu \leq 2(N+1)\left(\frac{N+1}{4}\right)^{\left(\frac{N+1}{2}\right)}$$

Proof: see proof of Theorem 9.3.2.2 in [13] ■

Since any linearly separable Boolean function can be converted into a positive linearly separable Boolean function by just flipping the sign of input variables, the above result is also true for the general case. Thus we can replace the sum over the $\tilde{p}_\nu$ by $\|\tilde{\mathbf{p}}\|_1$. Using (9) and (18), we get a lower bound on the relative robustness (worst-case) in the parameter space

$$\alpha_1 < \frac{1}{2(N+1)}\left(\frac{N+1}{4}\right)^{-\left(\frac{N+1}{2}\right)} \leq r_w(1, \mathbf{p}) \tag{19}$$

Expression (19) confirms the known result that the number of bits required to store the weights is of the order $O(N \log N)$ [18]. For a 1-neighborhood in **a**- and **b**-template ($N = 18$), 27 bits are needed to be able to realize *any* possible linearly separable Boolean function. The corresponding number of bits for a 2-neighborhood is 101 bits.

Remark 1: Note that (19) is a worst-case bound, i.e. it is a sufficient condition that the network will work for *any* desired linearly separable Boolean function. The bound in Theorem 4 is not tight, as has been shown experimentally for values $N \leq 7$ by Muroga [13]. On the other hand, certain linearly separable functions are known, which can only be realized by weights that grow exponentially in N (see Theorem 9.3.1.1 in [13], where a linearly separable function requiring $\Omega(N)$ bits to store the weights is constructed). Given the achievable accuracy of analog VLSI realizations, it seems doubtful whether it will be possible to build a DTCNN that can perform *any* linearly separable Boolean mapping even with only a 1-neighborhood in both templates. Practical experience though shows that there are applications, i.e. linearly separable mappings, for which the required accuracy is feasible with an analog realization. An example is the edge detector in [15], where $\|\tilde{\mathbf{p}}\|_1 = 9$, $\|\mathbf{e}\|_\infty = 1$, and thus an accuracy of $\alpha_1 = 11\%$ or 4 bits is sufficient for the desired task of the network, although a 1-neighborhood in **a**- and **b**-template is used.

Remark 2: Due to the much smaller number of inputs of each cell, the DTCNN compares very favorably to a fully connected Hopfield network, where the required accuracy for the network parameters is a lot more restrictive.

Optimization of Robustness for a Given Convex Cone

In practice, an optimally robust parameter vector $\mathbf{p}$ is desirable. This is necessary to obtain a high yield in a fabrication process, and on the other hand, it is reported that robust network parameters might have a positive effect on the generalization ability of the network [19]. In Mode III, the cell input vectors $\mathbf{e}_c(k)$ and desired cell outputs $y_c(k+1)$ are known for all cells c at any time-step k. Due to the translational invariance of the DTCNN template parameters, the problem of finding the optimally robust template parameters is equivalent to the problem of finding the optimally robust weights for a single perceptron, which maps all inputs $\mathbf{e}_c(k)$ onto the corresponding $y_c(k+1)$. Therefore, it is possible to

apply existing algorithms for the perceptron to optimize the robustness, i.e. to find the parameter vector so that the bounds on α_1 or α_2 in (9) and (17) are optimal. In this case, the convex cone $\mathcal{C}_{\mathbf{p}}$ and the actually occurring cell input vectors $\mathbf{e} \in \mathcal{E}_0^{\mathbf{p}}$ (or even better $\mathbf{e} \in \mathcal{E}_1^{\mathbf{p}}$, see Theorem 2) are known. Due to the redundancies in the input data and the interconnection structure, $|\mathcal{E}_0|$ is usually much smaller than suggested by the bound $\min\{KLM, 2^N\}$ in Table I. In [20], a DTCNN with 1-neighborhood in **a**- and **b**-template ($N = 18$) is used for a simple classification task of $L = 60$ 10×10 ($M = 100$) binary patterns. The longest trajectory was about $K = 40$. However, it turned out that $|\mathcal{E}_0| = 7101 \ll 262144 = 2^{18}$. An optimal worst-case robustness is thus achieved by solving (see (8) and (9))

$$\max_{\bar{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}} \left\{ \min_{\mathbf{e} \in \mathcal{E}_0^{\mathbf{p}}} \left\{ \frac{\mathbf{e}^{\mathrm{T}} \bar{\mathbf{p}}}{\|\mathbf{e}\|_\infty \|\bar{\mathbf{p}}\|_1} \right\} \right\} \tag{20}$$

This is the *Perceptron of Optimal Stability* problem with respect to the 1-norm, and it has been shown that this case can be reduced to a Linear Programming Problem [19] and is thus efficiently solvable.

The robustness for the average case can be optimized as well. Since $\|\mathbf{e}\|_2 = \sqrt{N+1}$ for all $\mathbf{e} \in \mathcal{E}_0$ in the binary mode, the corresponding optimization problem becomes with (17)

$$\frac{\sqrt{N+1}}{6} \max_{\bar{\mathbf{p}} \in \mathcal{C}_{\mathbf{p}}} \left\{ \min_{\mathbf{e} \in \mathcal{E}_0^{\mathbf{p}}} \left\{ \frac{\mathbf{e}^{\mathrm{T}} \bar{\mathbf{p}}}{\|\mathbf{e}\|_2 \|\bar{\mathbf{p}}\|_2} \right\} \right\} \tag{21}$$

This is the *Perceptron of Optimal Stability* problem with respect to the 2-norm, which can be solved efficiently by the AdaTron algorithm [15], [5]. Fig. 1(c) shows the optimal norm body $\mathcal{NB}(\mathbf{p}^{\mathrm{opt}}, \beta_{\max}^{\mathrm{opt}})$ in this case. Application of the AdaTron algorithm to the above classification example resulted in an improvement of the bound on α_2 from $\alpha_2 = 0.005\%$ to $\alpha_2 = 0.072\%$ (case with $P_{\mathrm{ok}} \geq 98.9\%$).

Relevance of the Bounds

In Mode III, the underlying perceptron problem will have a solution, since a convex cone is uniquely determined. In this case, the optimization problems (20) and (21) can be solved, and the solution of (21) is unique [21], [5]. It follows from the uniqueness that there is a set of $N+1$ linearly independent vectors $\mathbf{e}_0, \ldots, \mathbf{e}_N \in \mathcal{E}_0^{\mathbf{p}}$ such that $\mathbf{e}_\nu^{\mathrm{T}} \mathbf{p}^{\mathrm{opt}} = x_{\min}^{\mathrm{opt}}$ for $\nu = 0, \ldots, N$. $x_{\min}^{\mathrm{opt}}$ is the optimal $x_{\min}$ corresponding to a parameter vector $\bar{\mathbf{p}}$ found in (20) or (21). For all other vectors $\mathbf{e} \in \mathcal{E}_0^{\mathbf{p}} \backslash \{\mathbf{e}_0, \ldots, \mathbf{e}_N\}$, we have $\mathbf{e}^{\mathrm{T}} \mathbf{p}^{\mathrm{opt}} \geq x_{\min}^{\mathrm{opt}}$. Therefore, even in the extreme case where $\mathcal{E}_0^{\mathbf{p}} = \{\mathbf{e}_0, \ldots, \mathbf{e}_N\}$, $x_{\min}^{\mathrm{opt}}$ is still optimal, since any modification of the optimal parameter vector $\mathbf{p}^{\mathrm{opt}}$ would result in an $x_{\min} < x_{\min}^{\mathrm{opt}}$, and thus in stricter accuracy requirements due to (9) and (17).

Therefore, by picking the input signal and the initial state such that these $N+1$ vectors $\{\mathbf{e}_0, \ldots, \mathbf{e}_N\} \subseteq \mathcal{E}_0^{\mathbf{p}}$, i.e. that they actually appear as cell input vectors, the bounds on the required accuracy using (9) and (17) with $x_{\min}^{\mathrm{opt}}$ will be tight.

We will end this section with an example, where a relatively small DTCNN has high accuracy requirements. For small numbers of inputs ($N \leq 8$), all linearly separable Boolean functions have been categorized in tables (see for example [11], which contains cases $N \leq 6$). We will use a DTCNN that consists of a 1-dimensional row of $M = 13$ cells, an **a**- and a **b**-template with 1-neighborhoods, i.e. $N = 6$, and a bias. The input signals are binary, and the input set consists of one input pattern plus the corresponding initial state. The parameter vector is $\mathbf{p} = \frac{1}{35}(8, 7, 6, 5, 4, 3, 2)^{\mathrm{T}}$. Application of optimization algorithms of the type (20) confirms that this parameter vector is the optimal one, i.e. there are 7 vectors $\mathbf{e}_0, \ldots, \mathbf{e}_6 \in \mathcal{E}_0$, so that $|\mathbf{p}^{\mathrm{T}} \mathbf{e}_\nu| = x_{\min} = \frac{1}{35}$, while for all other possible binary cell input vectors $\mathbf{e}$, we have $|\mathbf{p}^{\mathrm{T}} \mathbf{e}| > x_{\min}$. The vectors $\mathbf{e}_\nu$ are

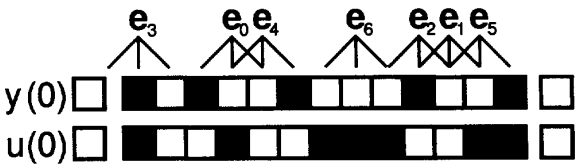


Fig. 3. Input $\mathbf{u}(0)$, initial state $\mathbf{y}(0)$, and the critical cell input vectors $\mathbf{e}_\nu$ for the example

$$\begin{aligned}
\mathbf{e}_0 &= (1, \ 1, -1, -1, -1, \ 1, -1)^{\mathrm{T}} \\
\mathbf{e}_1 &= (1, \ 1, -1, -1, -1, -1, \ 1)^{\mathrm{T}} \\
\mathbf{e}_2 &= (1, -1, \ 1, -1, \ 1, -1, -1)^{\mathrm{T}} \\
\mathbf{e}_3 &= (1, -1, \ 1, -1, -1, \ 1, -1)^{\mathrm{T}} \\
\mathbf{e}_4 &= (1, -1, -1, \ 1, \ 1, -1, -1)^{\mathrm{T}} \\
\mathbf{e}_5 &= (1, -1, -1, \ 1, -1, \ 1, \ 1)^{\mathrm{T}} \\
\mathbf{e}_6 &= (1, -1, -1, -1, \ 1, \ 1, \ 1)^{\mathrm{T}}
\end{aligned}$$

Note that any permutation of the elements of $\mathbf{p}$, where each element can be positive or negative, can be used to construct similar cases. Fig. 3 shows a possible input $\mathbf{u}(0)$, initial state $\mathbf{y}(0)$, so that the critical cell input vectors $\mathbf{e}_\nu$ actually occur. With (9), we get a required worst-case accuracy requirement of 6 bit even for the small network in this example. The bound obtained from (19) is 7 bits.

IV. Conclusions

DTCNNs are examined from a geometrical point of view. Different modes of operation are identified. The DTCNN is related to a standard perceptron, when the input set and the network parameters are fixed. In all other cases, the set of all cell input vectors also depends on the parameter vector. The mapping properties of the DTCNN are studied. An upper bound on the number of possible mappings for the whole network is given for the case of binary inputs, and it is shown that the number of performable mappings is of the order of $O(2^{N^2}/N!)$. This is a large loss when compared to the maximum number of performable mappings, and even compared to the number of possible mappings for a fully-connected Hopfield network.

A geometric view of the mapping at cell level is given in the input space and the parameter space. The parameter space is segmented by a large number of hyperplanes through the origin into a large number of convex cones, each of which uniquely defines a linearly separable mapping. The hyperplanes correspond to possible cell input vectors, and they can be split in two sets: those hyperplanes which bound a

given convex cone, and those which do not. Hyperplanes from the latter set can be written as positive linear combinations of hyperplanes from the first set, and the cell output signals corresponding to hyperplanes from the second set cannot be set independently. In the case of binary inputs, a convex cone can be bounded by as many as 2^N hyperplanes, where N is the number of inputs of each DTCNN cell.

Sufficient conditions on the required accuracy of the network parameters for a worst-case and an average-case scenario are given, so that the network is still functioning correctly. These bounds still depend on $x_{\min}$, the minimum absolute cell state value. For the binary input case, even the problem of deciding whether a certain hyperplane is the closest from a certain point in parameter space is shown to be NP-complete. Still, a lower bound on $x_{\min}$ depending on N can be given. For a DTCNN with 1-neighborhoods in both templates, an accuracy of 27 bit (worst-case result) is sufficient to guarantee the performability of all linearly separable Boolean mappings at cell level. It is confirmed that the number of bits required for storing the network parameters grows with the order $O(N \log N)$. Therefore, the DTCNN has a decisive advantage with respect to a hardware realization when compared to the fully-connected Hopfield network, since the number of network parameters is considerably smaller. In certain applications, however, the required accuracy constraints can be much less critical compared to the worst case bound. However, it is shown that a set of $N+1$ critical cell input vectors are sufficient to enforce the worst-case bounds.

Appendix I
Evaluation of the Error Integral

Let $\mathbf{v} := \mathbf{Q}^{\mathrm{T},-1}(\chi - \mathbf{p})$, and thus

$$d^{N+1}\chi = |\det \mathbf{Q}^{\mathrm{T}}| \cdot d^{N+1}\mathbf{v}$$

From (10) and (14), we then get

$$\begin{aligned} P_{\mathrm{ok}} &> \underbrace{\int \cdots \int}_{\mathcal{NB}(\mathbf{p},\beta)} p(\chi)\, d^{N+1}\chi \\ &= \frac{|\det \mathbf{Q}^{\mathrm{T}}|}{\sqrt{2\pi}^{N+1}\sqrt{\det \mathbf{C}}} \underbrace{\int \cdots \int}_{\|\mathbf{v}\|_2 \le \beta} e^{-\frac{1}{2}\|\mathbf{v}\|_2^2}\, d^{N+1}\mathbf{v} \\ &= \frac{(N+1)\tau_{N+1}}{\sqrt{2\pi}^{N+1}} \int_{\rho=0}^{\beta} \rho^N \cdot e^{-\frac{1}{2}\rho^2} d\rho \\ &= \frac{\sqrt{2}}{1\cdot 3\cdot \ldots \cdot (N-1)\sqrt{\pi}} \int_{\rho=0}^{\beta} \rho^N \cdot e^{-\frac{1}{2}\rho^2} d\rho \end{aligned} \tag{22}$$

τ_{N+1} is the volume of the unit hypersphere in $\mathbb{R}^{N+1}$. Since N is even (c.f. Section 1), we have

$$\tau_{N+1} = \frac{2^{\frac{N}{2}+1}\sqrt{\pi}^N}{1\cdot 3\cdot \ldots \cdot (N+1)}$$

(see for example [22]). For the term depending on ρ we use the recursion

$$\int_{\rho=0}^{\beta} \rho^N e^{-\frac{\rho^2}{2}} d\rho = (N-1) \int_{\rho=0}^{\beta} \rho^{N-2} e^{-\frac{\rho^2}{2}} d\rho - \beta^{N-1} e^{-\frac{\beta^2}{2}} \tag{23}$$

Since N is even, we can reduce the integration involving ρ to an expression containing the Gaussian error integral plus a sum of other terms. Applying (23) $\frac{N}{2}$ times, we finally arrive at the expression

$$\begin{aligned} \int_0^{\beta} \rho^N \cdot e^{-\frac{\rho^2}{2}} d\rho &= 1\cdot 3\cdot \ldots \cdot (N-1)\cdot \\ &\cdot \left[\int_0^{\beta} e^{-\frac{\rho^2}{2}} d\rho - e^{-\frac{1}{2}\beta^2} \sum_{l=0}^{\frac{N}{2}-1} \frac{\beta^{2l+1}}{1\cdot 3\cdot \ldots \cdot (2l+1)} \right] \end{aligned} \tag{24}$$

Since the sum is finite, the expression will converge for all β due to the influence of the exponential function. Putting together (22) and (24), we get

$$\begin{aligned} P_{\mathrm{ok}} > P_N(\beta) := \operatorname{erf}\left(\frac{\beta}{\sqrt{2}}\right) - \\ - \sqrt{\frac{2}{\pi}} \cdot e^{-\frac{1}{2}\beta^2} \sum_{l=0}^{\frac{N}{2}-1} \frac{\beta^{2l+1}}{1\cdot 3\cdot \ldots \cdot (2l+1)} \end{aligned}$$

This is equal to (15). The "erf(*)"-function is the Gaussian error integral

$$\operatorname{erf}(z) := \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

Appendix II
Proof of Theorem 3

In order to prove Theorem 3, we first have to introduce the *Knapsack* Problem (also called *0-1 Knapsack* problem, see for example the appendix of [23]):

Definition 4: [Knapsack problem]

INSTANCE: Finite set U, for each $u \in U$ a size $s(u) \in Z^+$, a value $v(u) \in Z^+$, and positive integers B and K.

QUESTION: Is there a subset $U' \subseteq U$ such that

$$\sum_{u\in U'} s(u) \le B \qquad \text{and} \qquad \sum_{u\in U'} v(u) \ge K \quad ?$$

The Knapsack problem is NP-complete [23], even if $s(u) = v(u)$ for all $u \in U$ (*Value-Independent Knapsack* problem). It can be solved in pseudo-polynomial time.

Proof: (Theorem 3) It is obvious that the problem in Theorem 3 is in NP, since if we are given a correct guess $\mathbf{e}_{\mathbf{p}}^{\mathrm{cor}}$ by an oracle machine, we can figure out in $O(N)$ steps by just evaluating the scalar product, that $|\mathbf{p}^{\mathrm{T}}\mathbf{e}_{\mathbf{p}}^{\mathrm{cor}}|$ is indeed smaller or equal than ϵ.

Secondly, we will give a polynomial reduction from the Value-Independent Knapsack problem. Let $N = |U|$. Introduce the binary variables $e_\nu \in \{-1, 1\}$ with $\nu = 0, 1, \ldots, N$. Let $e_0 = 1$ and

$$e_\nu = \begin{cases} +1 & \text{if } u_\nu \in U' \\ -1 & \text{otherwise} \end{cases} \qquad \nu = 1, 2, \ldots, N$$

Let $p_\nu = s(u_\nu)$ for $\nu = 1, 2, \ldots, N$, let $\epsilon = (B - K)$, and

$$p_0 = \sum_{\nu=1}^{N} s(u_\nu) - K - B$$

This transformation is polynomial in N. We now claim that the Value-Independent Knapsack problem has a solution if and only if the problem of finding $e_\nu \in \{-1, 1\}$ with $\nu = 1, 2, \ldots, N$ so that

$$|\mathbf{e}^{\mathrm{T}}\mathbf{p}| = \left| \sum_{\nu=0}^{N} e_\nu p_\nu \right| \leq \epsilon$$

has a solution. This is our original problem from Theorem 3. The fact that the p_ν in the original problem are rational is not a restriction, since it is possible to multiply all p_ν with one positive integer, so that in this case all p_ν become integers.

Using

$$\sum_{u \in U'} s(u) = \frac{1}{2} \sum_{\nu=1}^{N} (e_\nu + 1) \cdot s(u_\nu)$$

and thus

$$\begin{aligned} & 2K \leq 2 \sum_{u \in U'} s(u) \leq 2B \\ \iff \quad & 2K - B - K \leq \sum_{\nu=1}^{N} (e_\nu + 1) \cdot s(u_\nu) - B - K \\ & \leq 2B - B - K \\ \iff \quad & -\epsilon \leq \sum_{\nu=1}^{N} e_\nu \cdot s(u_\nu) + 1 \cdot p_0 \leq \epsilon \\ \iff \quad & \left| \sum_{\nu=0}^{N} e_\nu \cdot p_\nu \right| \leq \epsilon \end{aligned}$$

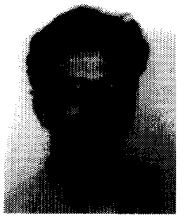
This proves the claim, and thus we have polynomially reduced the Knapsack problem to the problem of Theorem 3. This completes the proof. ■

ACKNOWLEDGMENT

The first author would like to thank Peter Nachbar, Andreas Schuler, and Wolfgang Utschick for many helpful comments and suggestions. We would also like to thank Prof. L. O. Chua for making possible the research stay of the first author at the Electronics Research Laboratory at the University of California, Berkeley.

REFERENCES

[1] H. Harrer and J. A. Nossek, "Discrete-time cellular neural networks," *Int. J. Circuit Theory and Applicat.*, vol. 20, pp. 453–467, Sept. 1992.
[2] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits and Syst.*, vol. 35, pp. 1257–1272, Oct. 1988.
[3] W. A. Little, "The existence of persistent states in the brain," *Math. Biosciences 19*, pp. 102–120, 1974.
[4] M. Minsky and S. Papert, *Perceptrons - An Introduction to Computational Geometry (Expanded Edition)*. Cambridge, MA: MIT Press, 1988.
[5] M. Biehl, J. K. Anlauf, and W. Kinzel, "Perceptron learning by constrained optimization: The AdaTron algorithm," in *IX. ASI Summer Workshop on Mathematical Physics "Neurodynamics 90"*, 1990.
[6] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415–1440, 1990.
[7] E. Goles-Chacc, F. Fogelman-Soulie, and D. Pellegrin, "Decreasing energy functions as a tool for studying threshold networks," *Discrete Appl. Math.*, vol. 12, pp. 261–277, 1985.
[8] J. Bruck, "On the convergence properties of the Hopfield model," *Proc. IEEE*, vol. 78, pp. 1579–1585, Oct. 1990.
[9] E. Goles and J. Olivos, "The convergence of symmetric threshold automata," *Informat. and Cont.*, vol. 51, pp. 98–104, 1981.
[10] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC 14, pp. 326–334, 1965.
[11] P. M. Lewis and C. L. Coates, *Threshold Logic*. New York: Wiley, 1967.
[12] C. Sheng, *Threshold Logic*. Toronto: Ryerson Press, 1969.
[13] S. Muroga, *Threshold Logic and its Applications*. New York: Wiley-Interscience, 1971.
[14] O. L. Mangasarian, *Nonlinear Programming*. New York: McGraw-Hill, 1969.
[15] P. Nachbar, J. A. Nossek, and J. Strobl, "The generalized AdaTron algorithm," in *Proc. Int. Symp. Circuits and Syst.*, vol. 4, pp. 2152–2156, 1993.
[16] N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, eds., *Combinatorial Optimization*. New York: Wiley, 1979.
[17] R. G. Parker and R. L. Rardin, *Discrete Optimization*. New York: Academic Press, 1988.
[18] P. Raghavan, "Learning in threshold networks," in *Proc. 1988 Workshop on Computational Learning Theory: COLT'88*, 1988.
[19] P. Nachbar, *Robuster Entwurf von Neuronalen Netzen*. Ph.D. dissertation, Technical University Munich, Munich, Germany, Dec. 1993.
[20] H. Magnussen, J. A. Nossek, and L. O. Chua, "The learning problem for discrete-time cellular neural networks as a combinatorial optimization problem," Tech. Rep. UCB/ERL M93/88, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, 1993.
[21] J. K. Anlauf and M. Biehl, *Properties of an Adaptive Perceptron Algorithm*, pp. 153–156. New York: Elsevier, 1990.
[22] O. Forster, *Analysis 3* . Vieweg, 1981 (in German).
[23] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.



Holger Magnussen received a Diploma degree in 1990 from the Technical University Munich, Germany, and a M.S. degree in 1991 from Stanford University, USA, both in electrical engineering. Holger Magnussen is currently working towards his Ph.D. degree at the Institute for Network Theory and Circuit Design at the Technical University Munich. His special interests include learning algorithms for two-state recurrent neural networks.

Josef A. Nossek (S'72–M'74–SM'81–F'92) received the Dipl.-Ing. and Dr. degrees in electrical engineering from the Technical University of Vienna, Austria, in 1974 and 1980, respectively.

In 1974 he joined Siemens AG, Munich, Germany, where he was engaged in the design of passive and active filters for communication systems. In 1977, he became a supervisor and, in 1990, head of a group of laboratories concerned with the design of electromechanical and microwave filters. Since 1982, he has been head of a group of laboratories designing digital radio systems within the Transmission Systems Department. In 1984, he spent a month as a visiting professor at the University of Capetown. From 1987 to 1989, he was head of the Radio Systems Design Department, and since April 1989, he has been professor of circuit theory and design at the Technical University of Munich. He is teaching graduate and undergraduate courses in the field of circuit and system theory and conducting research in the areas of real-time signal processing, neural networks, and dedicated VLSI architectures. He has published more than 50 papers in scientific and technical journals and conference proceedings. He holds a number of patents.

Dr. Nossek received the ITG Prize in 1988.