Coordinate independent update formulas for versor Clifford neurons

Sven Buchholz
Cognitive Systems Group
University of Kiel
24118 Kiel, Germany
Email: sbh@ks.informatik.uni-kiel.de

Eckhard Hitzer
Department of Applied Physics
University of Fukui
910-0024 Fukui, Japan
Email: hitzer@mech.fukui-u.ac.jp

Kanta Tachibana
Frontiers of Computational Science
Nagoya University
464-8603 Nagoya, Japan
Email: kanta@fcs.coe.nagoya-u.ac.jp

***Abstract*—We study the optimization of neural networks with Clifford geometric algebra versor and spinor nodes. For that purpose important multivector calculus results are introduced. Such nodes are generalizations of real, complex and quaternion spinor nodes. In particular we consider nodes that can learn all proper and improper Euclidean transformations with so-called conformal versors. Thus a single node can correctly compute full 3D screws and rotoinversions with off-origin axis and off-origin points of inversion. The latter is a unique property of our proposed versor neuron. Computing inversions by ordinary real-valued networks is not easily possible due to its nonlinear nature. Simulation on learning inversions illustrating these facts are provided in the paper.**

I. INTRODUCTION

Clifford neurons is a handle for neurons in the Clifford algebra domain. Clifford algebras subsume, for example, the real numbers, the complex numbers and the quaternions of Hamilton. In the 1960s David Hestenes started to study Clifford algebras as a universal language for geometry for which he coined the term ”Geometric algebra”.

Complex-valued neural networks are a very vital research topic with a lot of interesting applications (see e.g. [1]). Geometrically, complex multiplication can be seen as a transformation of the Euclidean plane, namely dilation-rotation. Hence complex-valued neural networks model a point of the plane as a single entity on which geometrical operations are carried out. A similar behavior for quaternionic-valued neural networks (w.r.t. points and transformations of the Euclidean space), however, requires a particular architecture based on so-called *spinor* nodes [2]. For quaternions, this spinor architecture turned out to be very useful in applications [3].

Here we want to extend the known spinor nodes[1] to nodes that can compute more powerful transformations. In particular we consider nodes that can learn all proper and improper Euclidean transformations with so-called conformal versors. Where we use the term *versor* for Lipschitz elements of Clifford groups [8], [12], as explained below. This includes, for example, inversions, which are not easy to compute by ordinary real-valued networks because they are nonlinear transformations. An interesting and new feature of our node theory is a priori coordinate independence. We consider a variety of new multivector node models, their error functions and (optimal) weight update rules: Clifford group versor neurons, unitary conformal versor neurons, homogeneous conformal versor neurons, and weight unitarity by Lagrange multipliers. The a priori invariance gives deep geometric insight and allows for easy systematic generalization to even higher dimensional spaces. Based on these results it is also straightforward to devise versor (and spinor) MLP back-propagation algorithms.

[1]Real and complex neurons can be viewed as spinor nodes too.

II. GEOMETRIC ALGEBRA

Definition 1 (Clifford geometric algebra)**.** *A Clifford geometric algebra $\mathcal{G}_{p,q}$ is defined by the associative geometric product of elements of a quadratic vector space $\mathbb{R}^{p,q}$, their linear combination and closure. $\mathcal{G}_{p,q}$ includes the field of real numbers $\mathbb{R}$ and the vector space $\mathbb{R}^{p,q}$ as subspaces. The geometric product of two vectors is defined as*

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}, \tag{1}$$

where $\mathbf{a} \cdot \mathbf{b}$ indicates the standard inner product and the bivector $\mathbf{a} \wedge \mathbf{b}$ indicates Grassmann's antisymmetric outer product. $\mathbf{a} \wedge \mathbf{b}$ can be geometrically interpreted as the oriented parallelogram area spanned by the vectors $\mathbf{a}$ and $\mathbf{b}$. Geometric algebras are graded, with grades (subspace dimensions) ranging from zero (scalars) to $n = p + q$ (pseudoscalars, n-volumes).

The geometric algebra $\mathcal{G}_3 = \mathcal{G}_{3,0}$ of three-dimensional Euclidean space $\mathbb{R}^3 = \mathbb{R}^{3,0}$ has an eight-dimensional basis of scalars (grade 0), vectors (grade 1), bivectors (grade 2) and trivectors (grade 3). Trivectors in $\mathcal{G}_3$ are also referred to as oriented volumes or pseudoscalars. Using an orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ for $\mathbb{R}^3$ we can write the basis of $\mathcal{G}_3$ as

$$\{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_2\mathbf{e}_3, \mathbf{e}_3\mathbf{e}_1, \mathbf{e}_1\mathbf{e}_2, i = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\}. \tag{2}$$

In (2) i is the unit trivector, i.e. the oriented volume of a unit cube. Let us point out, that the even subalgebra $\mathcal{G}_3^+$ of $\mathcal{G}_3$ is isomorphic to the quaternions $\mathbb{H}$ of Hamilton. We therefore call elements of $\mathcal{G}_3^+$ rotors (shorthand for rotation operators), because they can be used to implement rotations of vectors (and all other elements) of $\mathcal{G}_3$. The role of quaternion conjugation is naturally taken by reversion

$$(\mathbf{a}_1\mathbf{a}_2\dots\mathbf{a}_s)^\sim = \mathbf{a}_s\dots\mathbf{a}_2\mathbf{a}_1, \quad \mathbf{a}_1,\mathbf{a}_2,\dots,\mathbf{a}_s \in \mathbb{R}^{p,q}, \;\; s\in\mathbb{N}. \tag{3}$$

The inverse of a non-null vector $\mathbf{a}$ with respect to the geometric product is defined as

$$\mathbf{a}^{-1} = \frac{\mathbf{a}}{\mathbf{a}^2}. \tag{4}$$

A reflection at a hyperplane with normal vector $\mathbf{a}\in\mathbb{R}^{p,q}$ can be formulated as

$$\mathbf{x}' = -\mathbf{a}^{-1}\mathbf{x}\mathbf{a}. \tag{5}$$

A rotation by the angle θ in the plane of a unit bivector $\mathbf{i}$ can thus be given as the product $R=\mathbf{ab}$ of two vectors $\mathbf{a}$, $\mathbf{b}$ from the $\mathbf{i}$-plane (i.e. geometrically as a sequence of two reflections) with angle $\theta/2$.

Blades of grade k, $0\leq k\leq n=p+q$ are the outer products of k vectors $\mathbf{a}_l$ $(1\leq l\leq k)$ and directly represent the k-dimensional vector subspaces V spanned by the set of vectors $\mathbf{a}_l$ $(1\leq l\leq k)$. This subspace representation is also called outer product null space (OPNS) representation.

$$\mathbf{x}\in V = \mathrm{span}[\mathbf{a}_l, 1\leq l\leq k] \;\;\Leftrightarrow\;\; \mathbf{x}\wedge\mathbf{a}_1\wedge\mathbf{a}_2\wedge\dots\wedge\mathbf{a}_k = 0. \tag{6}$$

Extracting a certain grade part from the geometric product of two blades A_k and B_l has a deep geometric meaning.

One example is the grade $l-k$ part of the geometric product A_kB_l, that represents the orthogonal complement of A_k in B_l, provided that A_k is contained in B_l

$$A_k \rfloor B_l = \langle A_kB_l\rangle_{l-k} \tag{7}$$

Because of its geometrical significance this grade part is called contraction [9] of A_k on B_l.

Another important grade part of the geometric product of A_k and B_l is the maximum grade $l+k$ part, also called the outer product part

$$A_k\wedge B_l = \langle A_kB_l\rangle_{l+k}. \tag{8}$$

If $A_k\wedge B_l$ is non-zero it represents the union of the disjoint (except for the zero vector) subspaces represented by A_k and B_l.

The geometric algebras $\mathcal{G}_{p+1,q+1}$ are of special interest and are called conformal geometric algebras. One reason is that all orthogonal transformation groups $O(p,q)$ of vector spaces $\mathbb{R}^{p,q}$ are cases of Clifford (or Lipschitz) groups (defined below) of these spaces. Conformal transformation groups $C(p,q)$ preserve inner products (angles) of vectors in $\mathbb{R}^{p,q}$ up to a change of scale. Now the conformal group $C(p,q)$ is isomorphic to the orthogonal group $O(p+1,q+1)$. The metric affine group (consisting of orthogonal transformations and translations) of $\mathbb{R}^{p,q}$ is a special subgroup (specified below) of the orthogonal group $O(p+1,q+1)$, and can thus be implemented as a Clifford group in $\mathcal{G}_{p+1,q+1}$.

Combining several reflections leads to an overall sign (parity) for odd and even numbers of (reflection plane) vectors $\mathbf{a}$, $\mathbf{b}$, etc. due to (5). This can be generally taken care of by introducing the grade involution of multivectors $A\in\mathcal{G}_{p,q}$

$$\widehat{A} = \sum_{k=0}^{n}(-1)^k\langle A\rangle_k. \tag{9}$$

Now we can define [12] the powerful notion of Clifford (or Lipschitz) group which includes Pin(p,q), Spin(p,q), and $\mathrm{Spin}_+(p,q)$ groups as covering groups of orthogonal $O(p,q)$, special orthogonal $SO(p,q)$ and $SO_+(p,q)$ groups, respectively. A Clifford group is the subgroup in $\mathcal{G}_{p,q}$ generated by non-null vectors $\mathbf{x}\in\mathbb{R}^{p,q}$ in the following way

$$\Gamma_{p,q} = \{m\in\mathcal{G}_{p,q} \mid \forall\mathbf{x}\in\mathbb{R}^{p,q},\, \widehat{m}^{-1}\mathbf{x}\,m \;\in\mathbb{R}^{p,q}\} \tag{10}$$

For every $m\in\Gamma_{p,q}$ we have $m\widetilde{m}\in\mathbb{R}$.

A. *Conformal geometric algebra* $\mathcal{G}_{4,1}$

Given an orthonormal basis for $\mathbb{R}^{4,1}$

$$\{\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3,\mathbf{e}_+,\mathbf{e}_-\} \tag{11}$$

with

$$\mathbf{e}_1^2=\mathbf{e}_2^2=\mathbf{e}_3^2=\mathbf{e}_+^2=-\mathbf{e}_-^2=1, \tag{12}$$

we introduce a change of basis for the two additional dimensions $\{\mathbf{e}_+,\mathbf{e}_-\}$ by

$$\mathbf{e}_0=\frac{1}{2}(\mathbf{e}_++\mathbf{e}_-), \qquad \mathbf{e}_\infty=\mathbf{e}_--\mathbf{e}_+. \tag{13}$$

The vectors $\mathbf{e}_0$ and $\mathbf{e}_\infty$ are *isotropic* vectors, i.e.

$$\mathbf{e}_0^2=\mathbf{e}_\infty^2=0, \tag{14}$$

and have inner and outer products of

$$\mathbf{e}_0\cdot\mathbf{e}_\infty=-1, \qquad E=\mathbf{e}_\infty\wedge\mathbf{e}_0=\mathbf{e}_+\wedge\mathbf{e}_-. \tag{15}$$

We further have the following useful relationships

$$\begin{aligned}&\mathbf{e}_0E=-\mathbf{e}_0,\;\; E\mathbf{e}_0=\mathbf{e}_0, \mathbf{e}_\infty E=\mathbf{e}_\infty,\\ &E\mathbf{e}_\infty=-\mathbf{e}_\infty,\;\; E^2=1.\end{aligned} \tag{16}$$

III. Geometric objects in conformal geometric algebra

A. *Points, point pairs, circles and spheres*

Apart from the group theoretic reasons explained above, the conformal geometric algebra $\mathcal{G}_{4,1}$ provides us with a model [8], [9], [10] of Euclidean geometry, which has a number of computational advantages. The basic geometric objects in conformal geometric algebra are homogeneous conformal points given by

$$P=\mathbf{p}+\frac{1}{2}p^2\mathbf{e}_\infty+\mathbf{e}_0, \tag{17}$$

where $\mathbf{p}\in\mathbb{R}^3, p=\sqrt{\mathbf{p}^2}$. The $+\mathbf{e}_0$ term shows that we include projective geometry. The second term $+\frac{1}{2}p^2\mathbf{e}_\infty$ ensures, that conformal points are isotropic vectors

$$P^2=PP=P\cdot P=0. \tag{18}$$

A point pair is conformally represented (in OPNS) by

$$Pp = P_1 \wedge P_2. \tag{19}$$

The conformal outer product null space spanned by three conformal points is a Euclidean sphere in 2D, i.e. a circle

$$Circle = P_1 \wedge P_2 \wedge P_3. \tag{20}$$

The conformal outer product null space spanned by four conformal points is a 3D Euclidean sphere

$$Sphere \quad = \quad P_1 \wedge P_2 \wedge P_3 \wedge P_4. \tag{21}$$

B. Flat objects of flat points, lines, planes and the 5D space $\mathbb{R}^{4,1}$

In the conformal representations of circles (20) and spheres (21) the point at infinity $\mathbf{e}_\infty$ is not excluded. Indeed if one of the points is at infinity, we get conformal lines as flattened circles passing through infinity

$$Line = P_1 \wedge P_2 \wedge \mathbf{e}_\infty = Pp \wedge \mathbf{e}_\infty, \tag{22}$$

and conformal planes as spheres passing through infinity

$$Plane = P_1 \wedge P_2 \wedge P_3 \wedge \mathbf{e}_\infty = Circle \wedge \mathbf{e}_\infty. \tag{23}$$

The second form clearly shows the intimate relationship of point pairs with lines (circles with planes). In a sense all geometric properties of a line (plane) are already encoded in the wedge product of two (three) conformal points of the line (plane).

Consequently one can further introduce two more flattened objects,

$$P \wedge \mathbf{e}_\infty, \tag{24}$$

a (flat) finite–infinite point pair, and

$$-i_s E = Sphere \wedge \mathbf{e}_\infty, \tag{25}$$

a 5D pseudoscalar proportional to the 5D unit pseudoscalar $I = iE$ representing the (flat) 5D conformal space $\mathbb{R}^{4,1}$, i.e. the embedded (flat) 3D Euclidean space $\mathbb{R}^3$.

IV. Important multivector calculus

An important result of multivector differential calculus is given in Proposition 43 of [4] (see also [5])

$$(A * \partial_X) X = \dot{\partial}_X (\dot{X} * A) = P_{\text{subspace}}(A), \tag{26}$$

where $X = F(X)$ is the identity function on some linear subspace of $\mathcal{G}(I)$ of dimension d, and $P_{\text{subspace}}(A)$ is the projection[2] into this $d-$dimensional subspace of $\mathcal{G}(I)$. I is the pseudoscalar of the geometric algebra $\mathcal{G}(I) = \mathcal{G}(\mathbb{R}^{p,q})$.

We obtain therefore the following lemma [6].

Lemma 1. *For any constant multivector* $A \in \mathcal{G}(I)$

$$\partial_X \langle X^{-1} A\rangle = -P_{\text{subspace}}(X^{-1} A X^{-1}). \tag{27}$$

The brackets $\langle \dots \rangle$ in Lemma 1 mean the scalar part of the enclosed multivector $\langle X^{-1} A\rangle = \langle X^{-1} A\rangle_0 = X^{-1} * A$. Based on Lemma 1 we get the following theorem [6].

Theorem 1. *For any two constant multivectors* $A, B \in \mathcal{G}(I)$ *and* $A' = XAX^{-1}$ *we have*

$$\begin{aligned} \partial_X \langle XAX^{-1}B\rangle &= \partial_X \langle A'B\rangle \\ &= P_{\text{subspace}}(X^{-1}[A'B - BA']). \end{aligned} \tag{28}$$

An alternative form of Theorem 1 is the following corollary.

Corollary 1. *For any two constant multivectors* $A, B \in \mathcal{G}(I)$ *and* $A' = XAX^{-1}$, $B'' = X^{-1}BX$ *we have*

$$\begin{aligned} \partial_X \langle A'B\rangle = \partial_X \langle AB''\rangle &= \partial_X \langle B''A\rangle \\ &= P_{\text{subspace}}([AB'' - B''A]X^{-1}). \end{aligned} \tag{29}$$

V. Conformal geometric algebra (versor) neurons

As mentioned in the introduction, a quaternion spinor neuron (QSN) is excellent for implementing rotations in 3D space. Quaternions are a subalgebra of the larger geometric algebra $\mathcal{G}(\mathbb{R}^{4,1})$. This algebra contains the conformal model of Euclidean geometry which allows to implement rotoinversions (alias rotoreflections), screw motions, dilations, and transversions in Euclidean space in a very analogous way to rotations by general versor transformations [9], [14]. These transformations include therefore reflections at arbitrary planes, rotations around arbitrary axis, translations, central expansions (or contractions), and inversions relative to any point or sphere. Neural networks with these nodes will therefore be ideally suited to learn these transformations with high accuracy and efficiency.

Conformal versors V describe in conformal Clifford group [12] representations the above mentioned transformations of arbitrary conformal geometric object multivectors $X \in \mathcal{G}(\mathbb{R}^{3+1,1})$ of section III

$$X' = (-1)^v V^{-1} X V, \tag{30}$$

where the versor[3] is a geometric product of v *invertible* vectors $\in \mathbb{R}^{3+1,1}$.

We now introduce a neuron that extends the QSN and is a special instance of a Clifford neuron [7]. Hence the new type of neuron is characterized by a two-sided multiplication of a single multivector weight. A *conformal versor* (transformation) *neuron* (CVN) with input multivectors $X \in \mathcal{G}(\mathbb{R}^{3+1,1})$, weight versors $W \in \mathcal{G}(\mathbb{R}^{3+1,1})$, and multivector thresholds $\Theta \in \mathcal{G}(\mathbb{R}^{3+1,1})$ computes

$$Y = (-1)^w W^{-1} X W + \Theta, \tag{31}$$

where w represents the number of vector factors (parity) in W. The usual norm–type error function is associated with the CVN, compare eq. (39). Deriving a coordinate independent

[2]Please note that this projection is a natural result of differentiation, just as differentiation of a curve leads to the concept of tangent (generalized to tangent space in differential geometry).

[3]Please note that the overall sign of a conformal entity does not influence the representation of the Euclidean geometric object in question, just like in projective geometry.

weight update rule for the CVN requires to formulate this error function coordinate free as in (40). A necessary prerequisite is the socalled *principal involution*. The principal involution of the basis vectors (11) is defined as

$$\overline{\mathbf{e}_1} = \mathbf{e}_1,\ \overline{\mathbf{e}_2} = \mathbf{e}_2,\ \overline{\mathbf{e}_3} = \mathbf{e}_3,\ \overline{\mathbf{e}_+} = \mathbf{e}_+,\ \overline{\mathbf{e}_-} = -\mathbf{e}_-, \quad (32)$$

i.e. we always multiply with the sign of the quadratic form in (11). By linearity the principal involution of any vector $\mathbf{X} \in \mathbb{R}^{3+1,1}$ is

$$\overline{\mathbf{X}} = X_1\mathbf{e}_1 + X_2\mathbf{e}_2 + X_3\mathbf{e}_3 + X_\infty 2\mathbf{e}_0 + X_0\frac{1}{2}\mathbf{e}_\infty. \quad (33)$$

The principal involution of a versor is defined as the principal involution of its vector factors followed by reversion of the order of all vector factors

$$\begin{aligned}&\overline{a_1 a_2 \dots a_k} = \overline{a_k} \dots \overline{a_2}\ \overline{a_1},\\ &a_1, a_2, \dots, a_k \in \mathbb{R}^{3+1,1}, \quad k \in \mathbb{N}_0.\end{aligned} \quad (34)$$

All multivectors are linear combinations of blades, and all blades are versors. Linearity allows to extend the definition of the principal involution to arbitrary conformal multivectors [11]. The principal involution is an anti-involution

$$\overline{\overline{X}} = X, \qquad \overline{XY} = \overline{Y}\ \overline{X}, \qquad \forall X, Y \in \mathcal{G}(\mathbb{R}^{3+1,1}). \quad (35)$$

The principal involution does not change the grade of a multivector expression, grade extraction and principal involution commute therefore

$$\begin{aligned}&\overline{\langle X \rangle}_k = \langle \overline{X} \rangle_k,\\ &\forall X \in \mathcal{G}(\mathbb{R}^{3+1,1}), \qquad \forall k \in \mathbb{N}_0, 0 \le k \le 5.\end{aligned} \quad (36)$$

Scalars are invariant under the principal involution

$$\overline{\alpha} = \alpha, \qquad \forall \alpha \in \mathbb{R}. \quad (37)$$

We therefore get the useful identities

$$\begin{aligned}\langle \overline{X} Y \rangle =& \langle Y \overline{X} \rangle = \overline{\langle Y \overline{X} \rangle} = \langle \overline{Y \overline{X}} \rangle = \langle X \overline{Y} \rangle = \langle \overline{Y} X \rangle,\\ &\forall X, Y \in \mathcal{G}(\mathbb{R}^{3+1,1}).\end{aligned} \quad (38)$$

The use of the principal involution has the distinct advantage that the scalar product of any multivector $X \in \mathcal{G}(\mathbb{R}^{3+1,1})$ with its principal involution $\overline{X} \in \mathcal{G}(\mathbb{R}^{3+1,1})$ is positive definite:

$$\langle X\overline{X} \rangle = \langle \overline{X} X \rangle \ge 0, \qquad \langle X\overline{X} \rangle = 0 \Leftrightarrow X = 0. \quad (39)$$

We therefore can rewrite the conformal error function (norm) of a CVN as

$$E_C = \frac{1}{2}\langle error\ \overline{error} \rangle, \qquad error = D - Y, \quad (40)$$

where $D \in \mathcal{G}(\mathbb{R}^{3+1,1})$ represents a target output multivector for the CVN calculation.

Inserting (31) the conformal error function becomes

$$\begin{aligned}E_C =& \frac{1}{2}\langle (D-\Theta)\overline{(D-\Theta)} \rangle - \langle (-1)^w W^{-1} X W \overline{(D-\Theta)} \rangle\\ &+ \frac{1}{2}\langle W^{-1} X W \overline{W^{-1} X W} \rangle.\end{aligned} \quad (41)$$

Using Corollary 1 we get for the the weight multivector derivative of the full conformal error function (41)

$$\begin{aligned}-\partial_W E_C =& P_{SW}\left([\overline{error}(-1)^w W^{-1} X W\right.\\ &\left. -(-1)^w W^{-1} X W \overline{error}] W^{-1}\right).\end{aligned} \quad (42)$$

The concise form of (42) is suitable for deriving optimal weight update rules for neural networks with CVN nodes

$$\triangle W = -\eta\ \partial_W E_C, \qquad 0 < \eta \in \mathbb{R}. \quad (43)$$

The simple modification $W \to W' = W + \triangle W$ does not ensure that the W' is again a versor. In order to ensure this in practice versor composition techniques like in [15] may be necessary.

The optimal value of the constant η is related to the Hesse matrix of E_C. For Clifford neurons with one–sided multiplication the optimal value is [7], [13]

$$\eta_{opt} = \frac{1}{\lambda_{max}}, \quad (44)$$

where λ_{max} is the biggest eigenvalue of the Hesse matrix of the neuron under consideration. For CVN neurons the Hesse matrix may be invariantly computed as the matrix of all second order multivector differentials [4], [5]

$$(A * \partial_W)(B * \partial_W)E_C, \quad (45)$$

where for A and B all basis blades of the weight subspace of $\mathcal{G}(\mathbb{R}^{4,1})$ need to be inserted.

VI. Variants of the Conformal Versor Neuron

The weight dependence of (42) is more complicated than for the QSN [7]. One possibility for simplification would be to work with unitary versors.

A. Unitary conformal versor neuron

Clifford group versors V have the following property [12]

$$V\widetilde{V} = \widetilde{V}V \in \mathbb{R}, \quad (46)$$

where $\widetilde{V}$ indicates the *reverse* of V, i.e. reversing the order of all vector factors of V. It is therefore possible to *normalize* these versors with a real factor, such that

$$V\widetilde{V} = \widetilde{V}V = \pm 1, \quad (47)$$

which corresponds to restricting the versors to members of the normalized subgroup of the Clifford group, called Pin group.

As shown explicitly in Table 16.1 of [9], all versors in the conformal model of 3D Euclidean geometry describing reflections at planes, spheres and points, rotations, translations, scaling and transversions can easily be brought into this form with

$$V\widetilde{V} = \widetilde{V}V = +1. \quad (48)$$

We therefore consider the *unitary conformal versor neuron* (UCN) calculation

$$Y = (-1)^w W^{-1} X W + \Theta, \qquad W\widetilde{W} = \widetilde{W}W = +1. \quad (49)$$

with error function (41) again, for which the weight multivector derivative of E then reads

$$-\partial_W E_U = P_{SW}\left((-1)^w \widetilde{\overline{error}}\; \widetilde{W}\widetilde{X} + (-1)^w \overline{error}\; \widetilde{W} X\right). \tag{50}$$

Compared to (42) the weight dependence of $-\partial_W E_U$ is greatly simplified in (50). For rotors in the conformal model (rotations, translations, scaling and transversions) the parity is even and therefore $(-1)^w = 1$.

The concise result of (50) makes it very suitable for developing optimal weight update rules for neural networks with UCN nodes. In practice the unitarity of the weights can be secured in iterative updates by norming the updated weight $W_1 = W + \triangle W$ with

$$W_1 \longrightarrow \frac{W_1}{\sqrt{|W_1 \widetilde{W_1}|}}\,. \tag{51}$$

B. Homogeneous conformal versor neuron

Another way to construct a conformal versor node, would be to make both the node calculation Y and the error function calculation E_H homogeneous in the versor weights W in the following way:

$$Y = (-1)^w \widetilde{W} X W + \widetilde{W} W \Theta, \tag{52}$$

The error function can then be defined as

$$E_H = \frac{1}{2}\langle error\ \overline{error}\rangle, \qquad error = \widetilde{W} W D - Y. \tag{53}$$

where $D \in \mathcal{G}(\mathbb{R}^{3+1,1})$ represents a target output multivector for the *homogeneous conformal versor neuron* (HCN) calculation, and $error$ the homogeneous conformal error multivector.

Multivector weight derivation of (53) results in

$$\begin{aligned} -\,\partial_W E_H = {} & -2\widetilde{W}\left\langle error\overline{(D-\Theta)}\right\rangle \\ & + P_{SW}\left((-1)^w \widetilde{\overline{error}}\; \widetilde{W}\widetilde{X} + (-1)^w \overline{error}\; \widetilde{W} X\right). \end{aligned} \tag{54}$$

C. Weight unitarity by Lagrange multiplier

It would also be possible not to assume unitary weights, but to add the constraint via a *Lagrange multiplier* [16] to the conformal versor error function:

$$E_U \longrightarrow E_U + \frac{1}{2}\lambda(1 - (\widetilde{W}W)^2), \tag{55}$$

with constant $\lambda \in \mathbb{R}$. Then $\partial_W E$ of (50) would change to

$$-\partial_W E_U \longrightarrow -\partial_W E_U + 2\lambda \widetilde{W}(\widetilde{W}W). \tag{56}$$

VII. The projection P_{SW} into the weight subspace

Care must be taken for the projection P_{SW} into the weight subspace, because for calculating this projection the subspace under consideration needs to both have a blade basis and a reciprocal blade basis [4], [5], [6]

$$P_{SW}(A) = \sum a^J \langle a_J A\rangle, \tag{57}$$

where the a_J constitute the subspace blade basis and the a^J the corresponding *reciprocal* blade basis of the same subspace.

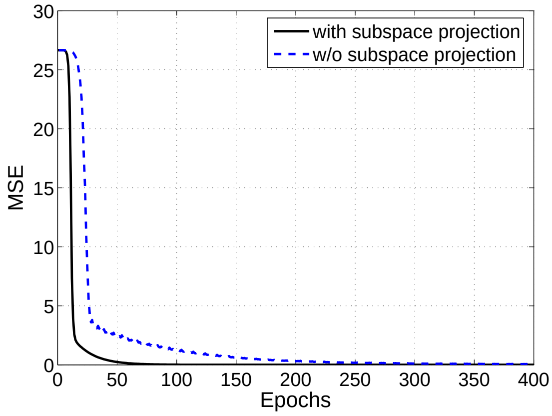

Fig. 1. Learning curves for the CVN. See text for details.

In the conformal model it is *very important* to note that as a result of (15) $-\mathbf{e}_0$ is reciprocal to $\mathbf{e}_\infty$ and vice versa.

For example if only rotations around the origin are considered the 4D rotor weight subspace (isomorphic to $\mathbb{H}$) has the basis

$$\{1, \mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}\}. \tag{58}$$

If only translations are considered the 7D translator weight subspace is spanned [6] by

$$\{1, \mathbf{e}_1\mathbf{e}_\infty, \mathbf{e}_2\mathbf{e}_\infty, \mathbf{e}_3\mathbf{e}_\infty, \mathbf{e}_1\mathbf{e}_0, \mathbf{e}_2\mathbf{e}_0, \mathbf{e}_3\mathbf{e}_0\}, \tag{59}$$

where $\mathbf{e}_1\mathbf{e}_0, \mathbf{e}_2\mathbf{e}_0, \mathbf{e}_3\mathbf{e}_0$ need to be included as the reciprocal bivectors of $\mathbf{e}_1\mathbf{e}_\infty, \mathbf{e}_2\mathbf{e}_\infty, \mathbf{e}_3\mathbf{e}_\infty$.

If rotor and translator weights are required to be combined as so-called motor weights we need the 12D weight subspace

$$\begin{aligned} &\{1, \mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}, \mathbf{e}_1\mathbf{e}_\infty, \mathbf{e}_2\mathbf{e}_\infty, \mathbf{e}_3\mathbf{e}_\infty, \\ &\quad \mathbf{e}_1\mathbf{e}_0, \mathbf{e}_2\mathbf{e}_0, \mathbf{e}_3\mathbf{e}_0, \mathbf{e}_{123}\mathbf{e}_0, \mathbf{e}_{123}\mathbf{e}_\infty\}. \end{aligned} \tag{60}$$

VIII. Simulation Results

The CVN allows to compute a variety of geometrical transformation as outlined before. In the following we report results for sphere inversions. This choice has been made because all conformal transformations can be expressed by combination of inversions. Our setup was as follows. The sphere (21) with center $[0\ 1\ -1]^T$ and radius 4 has been chosen as inversion sphere. Ten points have been uniformly sampled from the unit cube. Their conformal embedding (17) constituted the input training set. The output training set then resulted by versor multiplication (30) with the inversion sphere. A test set has not been generated because the CVN is able to learn inversions exactly. The CVN was trained by batch learning (with learning rate $\eta = 0.005$) with and without subspace projection. The results are shown in Fig. 1.

It can be seen there that subspace projection is not needed for convergence. Of course, convergence is much faster with subspace projection. In that case the MSE dropped below 10^{-10} after 400 epochs, which means that indeed the inversion sphere has been perfectly learned. Without subspace projection, however, MSE after 400 epochs was only 0.0290 with a weight representing a sphere having radius 3.89 and center $[0.0632\ -0.9099\ -0.96479]^T$.

For comparison, we also tested a linear network and a real-valued multilayer perceptron (MLP) on the above task. None of these networks can learn inversions exactly. Hence we additionally sampled three points from the unit cube to build up a test set. Sphere inversions are nonlinear mappings and hence a linear network can only come up with the best linear approximation. Trained on 3D points (w/o conformal embedding) approximation provided by the linear network has a MSE of 0.0024 (training) and of 0.1224 (test), respectively. Trained on conformal embedded 5D points the errors are 0.0051 and 0.0629, respectively.

MLPs are universal approximators. This means that training performance alone is not so interesting. The critical issue is generalization ability. In fact, a MLP with one hidden layer of only four sigmoidal nodes has been able to learn the 3D training set with MSE below 10^{-6}. However, corresponding MSE on test set is huge 0.0965 in this case. Best test performance with MSE of 0.0198 has been achieved using 6 hidden nodes. Using more hidden nodes resulted in overfitting. Using conformal embedded data renders the task more difficult for the MLP. The best test performance with MSE of 0.0776 has been achieved with 9 hidden nodes. Hence the task was not satisfactorily solvable using a MLP. Of course, the very small training set is a clear disadvantage for the MLP. Note that it is not possible in any case to derive the parameters of the inversion sphere from the MLP weights.

IX. Conclusion

We have introduced the concept of conformal versor neuron. These neurons have conformal geometric algebra versor weights. The input variables of the conformal versor neurons are geometric objects expressed by conformal multivectors (points, point pairs, circles, spheres, flat points, lines, planes and the 5D flat space). The versor weights correspond to proper and improper Euclidean transformations, including scaling and transversion, and the like.

A single node can therefore learn the transformation of a whole object by adapting its multivector weight versor. This has been demonstrated experimentally for inversions. A variety of nodes can be designed. It is expected that they will give rise to powerful networks with interesting applications.

Acknowledgment

The authors would like to thank Robert Valkenburg and the COE Frontiers of Computer Science at Nagoya University. E.H. wishes to thank God, the creator: *Soli Deo Gloria*, and his very supportive family.

References

[1] A. Hirose, *Complex-Valued Neural Networks*, Springer, Berlin, 2006.
[2] S. Buchholz, G. Sommer, *Quaternionic Spinor MLP*, Proceedings of ESANN 2000, d-side publications, 2000, pp. 377–382.
[3] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, H. Nishimura *Quaternion neural network with geometrical operators*, Journal of Intelligent & Fuzzy Systems **15** pp. 149-1164 (2004).
[4] E. Hitzer, *Multivector Differential Calculus*, Advances in Applied Clifford Algebras **12(2)** pp. 135-182 (2002).
[5] D. Hestenes, G. Sobczyk, *Clifford Algebra to Geometric Calculus*, Kluwer, Dordrecht, 1999.
[6] R. Valkenburg, private communication.
[7] S. Buchholz, *A Theory of Neural Computation with Clifford Algebras*, Technical Report Number 0504, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, May 2005.
[8] D. Hestenes, H. Li, A. Rockwood, *New Algebraic Tools for Classical Geometry*, in G. Sommer (ed.), Geometric Computing with Clifford Algebras, Springer, Berlin, 2001.
[9] L. Dorst, D. Fontijne, S. Mann, *Geometric Algebra for Computer Science: An Object-oriented Approach to Geometry*, Morgan Kaufmann Series in Computer Graphics, San Francisco, 2007.
[10] P. Angles, *Conformal Groups in Geometry and Spin Structures*, Progress in Mathematical Physics, Birkhauser, Boston, 2007.
[11] G. Laville, I.P. Ramadanoff, *Stone-Weierstrass Theorem*, eprint, arXiv:math.CV/0411090v2 25 Jan 2007.
[12] P. Lounesto, *Clifford Algebras and Spinors*, 2nd ed., CUP, Cambridge, 2006.
[13] S. Buchholz, K. Tachibana, E. Hitzer, *Optimal Learning Rates for Clifford Neurons*, in J. Marques de Sa et al. (Eds.): Proceedings of ICANN 2007, Part I, LNCS 4668, Springer-Verlag Berlin, 2007, pp. 864–873.
[14] E. Hitzer, K. Tachibana, S. Buchholz, I. Yu, *Carrier method for the general evaluation and control of pose, molecular conformation, tracking, and the like* accepted by Adv. in Appl. Cliff. Algs., 26 pp. (2008).
[15] L. Dorst, *Combining orientation estimates*, preprint, http://www.science.uva.nl/ga/
[16] H. Goldstein, *Classical Mechanics*, Addison-Wesley, London, 1959.