# A Note on Geometric Algebra and Neural Networks


*Kanta TACHIBANA
kanta@cc.kogakuin.ac.jp

MinhTuan PHAM
minhtuan@cmplx.cse.nagoya-u.ac.jp

Tomohiro YOSHIKAWA
{yoshikawa

Takeshi FURUHASHI
furuhashi}@cse.nagoya-u.ac.jp



**Abstract:** This note first explains Clifford's geometric algebra (GA) as a generalization of complex and quaternion algebras. Second, this note describes GA neurons as a natural extension of complex neurons. In any dimension the GA neuron takes a vector input and returns another vector output. The GA neurons are applicable to optimization of Space Folding Model for effective pattern recognition. Next, points with precision are considered using conformal geometric algebra and it is shown that addition of conformal vectors works well for precision update. The GA neuron and its use of vectors with precision (or belief) could be useful for datasets with different levels of precision/detail/belief of any dimension. The conformal vector could be also useful to set a prior distribution of geometric versors.


## 1 Introduction

Let $T(L)$ be the tensor space of the linear vector space $L$. Grassmann's exterior algebra $E(L)$ regards all elements of $T(L)$ which contain tensor products of any $x \in L$ with $x$ itself as the zero element of $T(L)$. The exterior product or wedge product maps an ordered pair $E(L) \times E(L)$ to $E(L)$. The exterior product is bilinear and $x \wedge x$ becomes 0 for any $x$ in $L$. Because $0 = (x+y) \wedge (x+y) = x \wedge x + x \wedge y + y \wedge x + y \wedge y$ where $x$ and $y$ are both elements of $L$, the exterior product is anti-commutative $x \wedge y = -y \wedge x$.

Clifford's geometric algebra $G(L)$ is the exterior algebra of a linear space $L$ equipped with a measure $x \cdot x = |x|^2$. $G(L)$ has a bilinear and associative product, that maps an ordered pair $G(L) \times G(L)$ to $G(L)$, which is called geometric product or Clifford product. For $x, y \in L$, the geometric product (simply written by juxtaposition of the elements) is defined as $xy = x \cdot y + x \wedge y$. Example: Let us think about the two-dimensional Euclidean space $\mathbb{R}^2$. The geometric products of its orthonormal basis vectors $\{e_1, e_2\}$ are $e_1e_1 = e_1 \cdot e_1 = e_2e_2 = e_2 \cdot e_2 = 1$ and $e_1e_2 = e_1 \wedge e_2 = -e_2 \wedge e_1 = -e_2e_1$. Because of the associativity we have $(e_1e_2)^2 = -e_2e_1e_1e_2 = -e_2e_2 = -1$. We therefore denote the unit bivector as $i = e_1e_2$, then $i^2 = -1$. The set of real linear combinations of $\{1, i\}$ is the even grade subalgebra of $G(\mathbb{R}^2)$, which is isomorphic to the set of complex numbers $\mathbb{C}$.

The authors have naturally extended the complex-valued neuron, all of whose input, weight, bias and output are in $\mathbb{C}$ [1, 2]. The proposed neuron uses the so-called Clifford group $= \{s \in G(L) \mid \varphi(s, x) \in L\ \forall x \in L\}$, where $\varphi$ is a function constructed with geometric products, with weight in $G(L)$, and with input, output and bias in $L$ [3]. This note discusses the relationship of the GA neuron with complex and quaternion neurons [4]. This note also considers points with precision using conformal geometric algebra.

## 2 Complex Neuron

The complex neuron in general sums input stimuli weighted by weights plus bias all of which are complex numbers. For simplicity, assume the neuron has an input:

$$u_{\mathbb{C}} = w_{\mathbb{C}} x_{\mathbb{C}} + b_{\mathbb{C}},$$

where $u_{\mathbb{C}}, w_{\mathbb{C}}, x_{\mathbb{C}}, b_{\mathbb{C}} \in \mathbb{C}$. A two-dimensional vector $(x_1, x_2)$ can be represented as a complex number. Its first and second components are the real and the imaginary coefficients respectively: $x_{\mathbb{C}} = x_1 + x_2 i \in \mathbb{C}$. On the other hand, a natural representation of a vector is $x = x_1e_1 + x_2e_2$. Using geometric algebra $G(\mathbb{R}^2)$, we can link it to the complex number representation as $x_{\mathbb{C}} = x_1(e_1e_1) + x_2(e_1e_2) = e_1(x_1e_1 + x_2e_2) = e_1x$. The square root of $w_{\mathbb{C}} = \rho(\cos\theta + i\sin\theta),\ \rho \in [0, \infty),\ \theta \in [0, 2\pi)$ is also a complex number $w'_{\mathbb{C}} = \sqrt{w_{\mathbb{C}}} = \sqrt{\rho}(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2})$. And complex numbers are commutative, i.e. $w'_{\mathbb{C}}(e_1x) = (e_1x)w'_{\mathbb{C}}$. Then, the complex neuron becomes:

$$\begin{aligned} (e_1u) &= w'_{\mathbb{C}}w'_{\mathbb{C}}(e_1x) + (e_1b) \\ &= w'_{\mathbb{C}}(e_1x)w'_{\mathbb{C}} + (e_1b). \end{aligned}$$

Multiply $e_1$ from the left:

$$e_1e_1u = e_1\underline{w'_{\mathbb{C}}e_1}xw'_{\mathbb{C}} + e_1e_1b.$$

Looking at the underlined part, and let $w'_{\mathbb{C}} = \alpha + \beta i$,

$$\begin{aligned} w'_{\mathbb{C}}e_1 &= (\alpha + \beta i)e_1 = \alpha e_1 + \beta e_1e_2e_1 \\ &= e_1(\alpha - \beta i) = e_1\overline{w'_{\mathbb{C}}}, \end{aligned}$$

where $\overline{w'_{\mathbb{C}}}$ is the complex conjugation. Then, the complex neuron is represented as:

$$\begin{aligned} e_1e_1u &= e_1e_1\overline{w'_{\mathbb{C}}}xw'_{\mathbb{C}} + e_1e_1b \\ u &= \overline{w'_{\mathbb{C}}}xw'_{\mathbb{C}} + b. \end{aligned} \tag{1}$$

This $u$ is the result of rotating $x$ by the angle $\theta$, scaling by factor $\rho$, and translation by vector $b$.

## 3 Geometric Algebra Neuron

The geometric algebra neuron is a natural extension of the complex neuron. Let the Clifford group $\Sigma =$

$\{s \in G(L) \mid \varphi(s,x) \in L \ \forall x \in L\}$, whose element transforms a vector to another vector. For $k = 0, 1, \ldots, n$, the set of versors, i.e. multiplications of $k$ linearly independent vectors $\{M_k = v_1 v_2 \ldots v_k \in G(L) \mid v_1, \ldots, v_k \in L\}$ is a subset of $\Sigma$ with $\varphi(M_k, x) = \overline{M_k} x M_k$. The authors have proposed a geometric algebra neuron:

$$u = \sum_{k=0}^{n} \varphi(M_k, x) + b$$

and found optimal learning rates based on the Hessian matrix. Note that the complex neuron of eq. (1) only represents $\varphi(M_2, x)$ part. In the case of $n = 2$, $\varphi(M_0, x)$ is scalar multiplication, $\varphi(M_1, x)$ is a reflection, and $\varphi(M_2, x)$ is a rotation. These three transformations are mixed. The mixing weights are adjusted with the norm $|M_k|^2$. In the case of $n = 3$, $\varphi(M_2, x)$ is isomorphic to the quaternion product which gives rotation and scaling in three-dimensional space.

As the GA neuron learns reflection and rotation of vectors and multivectors in any dimension, a network constructed with GA neurons can be applicable to optimization of Space Folding Model (SFM) [5]. In the network for SFM, a GA neuron is assigned for each Space Folding Vector (SFV) and the feature space is folded to minimize the error function by training $\{M_k\}s$.

## 4 Conformal GA and Update of Precision

Introducing new two basis vectors $\vec{e}_+$ and $\vec{e}_-$ to $G(\mathbb{R}^n) = G(n)$, we have conformal geometric algebra $G(n+1, 1)$. The new vectors have positive and negative signature, i.e. $\vec{e}_+^{\,2} = -\vec{e}_-^{\,2} = 1$. And further new basis vectors are constructed in the $\vec{e}_+ \wedge \vec{e}_-$ plane:

$$\begin{cases} \vec{n}_\infty = \vec{e}_+ + \vec{e}_- \\ \vec{n}_o = \frac{1}{2}\left(\vec{e}_- - \vec{e}_+\right). \end{cases}$$

Because $\vec{n}_\infty^2 = \vec{n}_o^2 = 0$, they are also called null basis vectors. Using these null basis vectors, $n$-dimensional hypersphere with center at $\vec{x} \in \mathbb{R}^n$ and radius $\rho \in \mathbb{R}$ is represented as

$$X = \mu \vec{x} + \frac{\mu}{2}\left(\vec{x}^2 - \rho^2\right)\vec{n}_\infty + \mu \vec{n}_o,$$

where $\mu$ is any nonzero real number.

We regard $\rho^2$ of the conformal vector as precision, say $\rho^2 = \beta = \sigma^{-2}$, where $\sigma$ represents a standard deviation. In this interpretation, addition of two conformal vectors means:

$$\begin{aligned} X + Y &= \vec{x} + \vec{y} + \frac{1}{2}\left(\vec{x}^2 - \beta_x + \vec{y}^2 - \beta_y\right)\vec{n}_\infty + 2\vec{n}_o \\ &\propto \frac{\vec{x}+\vec{y}}{2} + \frac{1}{2}\left(\frac{\vec{x}^2 + \vec{y}^2}{2} - \frac{\beta_x + \beta_y}{2}\right)\vec{n}_\infty + \vec{n}_o \\ &= \vec{m} + \frac{1}{2}\left(\vec{m}^2 - \beta\right)\vec{n}_\infty + \vec{n}_o, \end{aligned}$$

where $\vec{m} = (\vec{x} + \vec{y})/2$, i.e. the midpoint, and $\beta = (\beta_x + \beta_y)/2 - \left\{(\vec{x} - \vec{m})^2 + (\vec{y} - \vec{m})^2\right\}/2$. The new precision $\beta$ is interpreted as average precision minus variance.

This can be generalized to weighted points. Let $\mu_x \in \mathbb{R}$ be the weight.

$$X = \mu_x \vec{x} + \frac{\mu_x}{2}\left(\vec{x}^2 - \beta_x\right)\vec{n}_\infty + \mu_x \vec{n}_o.$$

The addition of weighted points is:

$$\begin{aligned} X + Y &= \mu_x \vec{x} + \mu_y \vec{y} + \frac{\mu_x(\vec{x}^2 - \beta_x) + \mu_y(\vec{y}^2 - \beta_y)}{2}\vec{n}_\infty \\ &+ (\mu_x + \mu_y)\vec{n}_o \\ &= \mu \vec{m} + \frac{\mu}{2}\left(\vec{m}^2 - \beta\right)\vec{n}_\infty + \mu \vec{n}_o, \end{aligned}$$

where $\mu = \mu_x + \mu_y$ is the new weight and

$$\begin{aligned} \vec{m} &= \frac{\mu_x \vec{x} + \mu_y \vec{y}}{\mu} \\ \beta &= \frac{\mu_x \beta_x + \mu_y \beta_y}{\mu} - \frac{\mu_x(\vec{x} - \vec{m})^2 + \mu_y(\vec{y} - \vec{m})^2}{\mu} \end{aligned}$$

$\vec{m}$ is the internally dividing point (center of mass) and the new precision $\beta$ is interpreted as weighted average precision minus weighted variance.

This fact of good update of precision can be useful in the following cases.

1. Each training/test sample has a different level of precision $\beta$ (or belief).
2. Massive samples must be learnt and coarse graining is effective. Here, a lot of samples are learnt at once as a hypersphere sample.
3. Precision (or belief) characterises a prior distribution of a dataset and neuron parameters.

## 5 Conclusion

This note described the GA neuron as a natural extension of the complex neuron. In any dimension the GA neuron inputs a vector and outputs another vector. Next, points with precision were considered using conformal geometric algebra. And we showed that the addition of conformal vectors works well to update precision. We want to note the possibility of combining GA neuron with conformal representation of precision in the analysis/learning of datasets with various levels of details and in the coarse graining of huge datasets. Bayesian updates of weights could also be represented by conformal vectors.


## acknowledgment

The authors thank Dr. Eckhard Hitzer for his suggestions.


## References


[1] 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.

[2] S. Buchholz, E. Hitzer, K. Tachibana, Coordinate independent update formulas for versor Clifford neurons, SCIS and ISIS 2008, Nagoya, September 2008.

[3] P. Lounesto, Clifford Algebras and Spinors, Cambridge University Press, 2001

[4] T. Nitta (ed.), Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters, Information Science Publishing, 2009.

[5] M. T. Pham, T. Yoshikawa, T. Furuhashi, Pattern Recognition Based on Space Folding Model, Proc. Applied Geometric Algebras in Computer Science and Engineering (AGACSE2010), June 2010.