Computation with Clifford valued Feed-Forward Networks

Justin Pearson
Computer Science Department
Royal Holloway
University of London
Egham
Surrey TW20 0EX
e_mail: justin@dcs.rhbnc.ac.uk
Tel. +44 (0) 1784 443426 Fax. +44 (0) 1784 443420

D.L. Bisset*
Electronic Engineering Laboratories,
University of Kent,
Canterbury, Kent, CT2 7NT, UK.

Tel. +44 1227 764000 Fax. +44 227 456084
e_mail: D.L.Bisset@ukc.ac.uk

October 31, 1996

Abstract

Recent research has focused on feed-forward networks with complex weights and activation values such as [GK92, Hir92b, Hir92a, Hir93]. This paper extends this formalism to feed-forward networks with weight and activation values taken from a Clifford algebra (see also [PB92, PB94b]). A Clifford algebra is a multi-dimensional generalization of the complex numbers and the Quaternions. Essentially a Clifford algebra is obtained by extending vector spaces to allow an associative multiplication compatible with the natural metric on the vector space.

This paper presents an extension of the well known back-error propagation algorithm to Clifford valued feed-forward networks, and presents some experimental results with simple encoder-decoder problems. A discussion of the difference between real and Clifford valued networks is also included. Finally a Universal Approximation similar to the results found in [HSW89] is proved.

1 Introduction

Most current research into neural networks focuses on the use of real valued weights and activiation values. More recently the use of complex weights has

been explored with some success [GK92, Hir92b, Hir92a, Hir93]. Complex numbers have been used extensively in engineering and science as a useful analytical and modeling tools. However they are only one instance of a class of algebras that can, and have been, extensively used (e.g. the use of Spin algebras in mathematical physics [CC86, BLJM89]). It is therefore important to try and extend neural networks to cover not only the complex numbers but the other forms of multi-dimensional number. In order that that they can be applied to problem domains that might benefit from their use. For example it is possible to represent a colour image as three corresponding arrays of red, green and blue points respectively, and for three neural networks to operate in parallel on each image one for each colour. Using a Clifford valued system, such as proposed in this paper, a single neural network could be employed that is able to operate on three dimensional numbers where each colour is coded as one of the elements of the number (complex numbers have two elements, quaternions four). Such a coding would allow the neural network to process the whole colour image without splitting it into its separate components.

Clifford algebra provides a formalism for describing a general class of algebras that encompass the Complex numbers, the Quaternions and various matrix algebras. This paper describes how neural networks can be made to operate on Clifford numbers and therefore on any particular number, this then covers the general case. The paper also gives some insight into implementation issues, and provides a Universal Approximation proof for Clifford Networks.

2 Clifford algebras

A Clifford algebra is the answer to the question, how can a vector space be provided with an associative vector-valued multiplication? Given a real vector space $\mathbf{R}^{p+q}$ of dimension $p+q$ (the reason using $p+q$ rather than n say, will become apparent) with basis $e_1, \ldots e_p \ldots e_{p+q}$, the addition of two elements is well defined, for instance:

$$x = \sum_{i=1}^{p+q} x_i e_i$$

$$y = \sum_{i=1}^{p+q} y_i e_i$$

$$x + y = \sum_{i=1}^{p+q} (x_i + y_i) e_i$$

Multiplication is more problematic, neither the scaler product or the vector product are any use, since the scaler product yields a scaler and the vector product is limited to three dimensions. Proceeding purely formally, given two vectors:

$$x = 4e_1 + e_2$$

$$y = e_2$$

For instance, the product would be:

$$xy = 4e_1 + 4e_1e_2 + e_2^2$$

The question is then what to do with the extra elements e_1e_2 and e_2^2. If the following conventions are adopted (which arise naturally in the context of quadratic form theory see [Por81]):

$$e_i^2 = 1 \quad , \quad i = 1, \ldots, p \tag{1}$$

$$e_i^2 = -1 \quad , \quad i = p+1, \ldots, p+q \tag{2}$$

$$e_ie_j = -e_je_i \quad , \quad i \neq j \tag{3}$$

With for $1, \leq h_1, < \ldots h_r \leq n$,

$$e_{h_1} \cdot e_{h_2} \cdots e_{h_r} = e_{h_1 \ldots h_r}. \tag{4}$$

Then all the extra elements can be removed, generating a vector space of dimension 2^{p+q} with basis elements:

$$\{e_A = e_{h_1 \ldots h_r} | A = (h_1, \ldots, h_r), 1 \leq h_1 < \ldots < h_r \leq n\}.$$

For example the Clifford algebra generated by $\mathbf{R}^2$ will have the basis:

$$1, \ e_1, \ e_2 \ e_{12}$$

Multiplication can be expressed more compactly by:

$$e_Ae_B = (-1)^{\#((A \cap B) \setminus P)}(-1)^{p(A,B)}e_{A \Delta B} \tag{5}$$

where P stands for the set $1, \ldots p$, and $\#X$ represents the number of elements in X,

$$p(A,B) = \sum_{j \in B} p'(A,j), \quad p'(A,j) = \#\{i \in A | i > j\} \tag{6}$$

and the sets A, B and $A \Delta B$(the set difference of A and B) are ordered in the *natural* way. It will be useful in the derivation of the back-propagation algorithm to define the quantity $\kappa_{A,B}$ for two basis elements A and B as:

$$\kappa_{(A,B)} = (-1)^{\#((A \cap B) \setminus P)}(-1)^{p(A,B)} \tag{7}$$

It turns out that there is essentially only one way of providing an associative multiplication to a vector algebra and that is a Clifford algebra. The reader can check that $\mathbf{R}_{0,1}$ is isomorphic to the complex numbers and $\mathbf{R}_{0,2}$ is isomorphic to the Quaternions, for more details again check [Por81]. For applications of Clifford algebras to mathematical physics see [CC86].

In what follows a Clifford number will be represented as:

$$x = \sum_A x_Ae_A \tag{8}$$

Where A ranges over all the basis elements in the Clifford algebras. The A'th part of an element is denoted as $[x]_A$. In general Clifford algebras are non-commutative.

Clifford algebras are used extensively in mathematical physics, because they can arise as representations of symmetry groups and can aid calculations. For example they can be used to present a compact form of Maxwell's equations for the propagation of electro-magnetic waves, or to simplify computational problems, see [CC86] for examples in other areas of physics.

3 Clifford back error propagation

In what follows the norm[1] $|\cdot|$ will be used, where,

$$|x| = \left(\sum_{A} [x]_A^2 \right)^{\frac{1}{2}} \tag{9}$$

where $[x]_A$ represents the A'th part of the Clifford number x, although this norm is not the standard norm on a Clifford algebra (except in the case $\mathbf{R}_{0,n}$) it does facilitate the derivation of a useful learning algorithm.

A feed-forward Clifford network with n inputs and m outputs can be thought of as a function,

$$\Psi : (\mathbf{R}_{p,q})^n \to (\mathbf{R}_{p,q})^m \tag{10}$$

Where $(\mathbf{R}_{p,q})^n$ is the n-dimensional left module[2]. over the Clifford algebra $\mathbf{R}_{p,q}$

The following error metric will be used:

$$E = \frac{1}{2} \int_{x \in X} \|\Psi - \Phi\|^2 \tag{11}$$

where X is some compact subset of the Clifford module $(\mathbf{R}_{p,q})^n$ with the product topology derived from the norm (9).

It is convenient from the point of view of the derivation of the BEP equations to define $\|\cdot\|$ as,

$$\|x\|^2 = \sum_{i=1}^{k} |(x)_i|^2 \tag{12}$$

where $(x)_i$ is a Clifford number representing the i'the part of x in the m dimensional Clifford module over $\mathbf{R}_{p,q}$.

Assume that each node in the network has the same Clifford valued activation function $f : \mathbf{R}_{p,q} \to \mathbf{R}_{p,q}$.

The output o_j of the j'th neuron can be written as,

$$o_j = f(net_j) = \sum_{A} w_A^j e_A \tag{13}$$

with w_A^j a function from $\mathbf{R}_{p,q}$ to $\mathbf{R}$ and

$$net_j = \sum_{l} \omega_{lj} o_k \tag{14}$$

where k sums over all the inputs to neuron j.

It is important to notice since $\mathbf{R}_{p,q}$ is in general non-commutative the order of multiplication in the above equation is a priori important. That is networks with right and left multiplication with the same weights will have different behaviors. Although it can be shown that the approximation capabilities of left and right multiplication networks are the same.

[1] This is only a norm on the underlying vector space of the algebra, not on the whole of the algebra. But it is sufficient for deriving a minimization algorithm.

[2] If the reader is not familiar with the concept of a module, it is enough to view these Clifford modules as weaker forms of n dimensional vectors with Clifford valued scalars instead of real values scalars. Also see section 6 for the general theory.

In the real case E depends on the number of weights in the network. In the Clifford case E depends not only on all the weights but on the components of each of the weights. Define $\lambda = \|\Psi - \Phi\|^2$, then:

$$\frac{\partial E}{\partial [\omega_{ij}]_A} = \frac{1}{2} \int_{x \in X} \frac{\partial \lambda}{\partial [\omega_{ij}]_A} \tag{15}$$

using the chain rule,

$$\frac{\partial \lambda}{\partial [\omega_{ij}]_A} = \sum_B \left(\frac{\partial \lambda}{\partial u_B^j} \left(\sum_C \frac{\partial u_B^j}{\partial [net_j]_C} \frac{\partial [net_j]_C}{\partial [\omega_{ij}]_A} \right) \right) \tag{16}$$

The partial derivative

$$\frac{\partial [net_j]_C}{\partial [\omega_{ij}]_A}$$

needs a bit of care. Using equation (14):

$$\frac{\partial [net_j]_C}{\partial [\omega_{ij}]_A} = \sum_l \frac{\partial [\omega_{lj} x_l]_C}{\partial [\omega_{ij}]_A} = \frac{\partial [\omega_{ij} o_i]_C}{\partial [\omega_{ij}]_A}$$

Then using the fact that

$$\omega_{kj} o_k = \sum_{D,E} [\omega_{kj}]_D [o_k]_E \kappa_{(D,E)} e_D \Delta E$$

with κ defined as in (5), then,

$$\frac{\partial [\omega_{kj} o_k]_C}{\partial [\omega_{kj}]_A} = \kappa_{(D,E)} [o_k]_E \tag{17}$$

where E is the basis element satisfying the equation:

$$\kappa_{(A,E)} e_A e_E = e_C$$

For example in the algebra $\mathbf{R}_{2,0}$ the table of derivatives would look like,

$\frac{\partial [x]_B}{\partial [\omega_{jl}]_A}$	$B=0$	1	2	12
$A=0$	$[x_{jl}]_0$	$[x_{jl}]_1$	$[x_{jl}]_2$	$[x_{jl}]_{12}$
1	$[x_{jl}]_1$	$[x_{jl}]_0$	$[x_{jl}]_{12}$	$[x_{jl}]_2$
2	$[x_{jl}]_2$	$-[x_{jl}]_{12}$	$[x_{jl}]_0$	$-[x_{jl}]_1$
12	$-[x_{jl}]_{12}$	$[x_{jl}]_2$	$-[x_{jl}]_1$	$[x_{jl}]_0$

(18)

The error derivative is now quite easy to calculate, if j is an output neuron then,

$$\frac{\partial \lambda}{\partial u_A^j} = \frac{\partial}{\partial u_A^j} \|\Psi - \Phi\|$$

$$\frac{\partial}{\partial u_A^j} |o_j - \Phi_j^2|^2 = 2[o_j - \Phi_j]_A$$

If j is not an output unit then the chain rule has to be used again.

$$\frac{\partial \lambda}{\partial u_A^j} = \sum_k \frac{\partial \lambda}{\partial u_A^k} \left(\sum_{B,C} \frac{\partial u_B^k}{\partial [net_k]^C} \frac{\partial [net_k]^C}{\partial u_A^j} \right) \tag{19}$$

with k running over the neurons that receive input from neuron j.

The term:

$$\frac{\partial [net_k]_C}{\partial [u_j]_A}$$

is calculated in a similar manner to (17),

$$\frac{\partial [net_l]_C}{\partial u_A^j} = (-1)^{\kappa'} [\omega_{jk}]_D \tag{20}$$

where κ' and D satisfy the same conditions as before. The derivatives:

$$\frac{\partial u_B^k}{\partial [x_k]_C}$$

play the same rôle as $f'(net_j)$ does in the real valued case and depends on the activation function used, this will be discussed in the next section.

Bringing this all together we have,

$$\frac{\partial E}{\partial [\omega_{ij}]_A} = \frac{1}{2} \int_{x \in X} \sum_B \lambda_j^B \left(\sum_C \frac{\partial u_B^j}{\partial [net_j]_C} \kappa_{(A,E)} [o_k]_E \right) \tag{21}$$

with

$$\lambda_j^B = \frac{\partial \|\Psi - \Phi\|^2}{\partial u_B^j} = 2[o_j - \Phi_j]_B \tag{22}$$

if j is an output neuron. Again E is the basis element satisfying the equation:

$$\kappa_{(A,E)} e_A e_E = e_C$$

If j is not an output unit then the chain rule has to be used again.

$$\lambda_j^B = \sum_k \lambda_k^B \left(\sum_{B,C} \frac{\partial u_B^k}{\partial [net_k]^C} \kappa_{(A,D)} [\omega_{jk}]_D \right) \tag{23}$$

with k running over the neurons that receive input from neuron j D defined in same way as E is in (3).

3.1 Choice of activation function

For the complex case it might be assumed that the rich field of complex analysis would provide a suitable class of activation functions. There exists a complex extension of the sigmoid function:

$$f(z) = \frac{1}{1 + e^{-z}}$$

where e^{-z} is the complex exponential function. This function is analytic(in the sense of complex analysis) but it is not bounded, as it is on the real line. It

is an unfortunate fact that any function that is complex analytic and bounded is necessary constant by Liouville's theorem(see any standard text on complex analysis for a proof).

The most important characteristic of a complex activation function, to ensure learning can take place: is that it should be bounded and nonlinear in its components, its partial derivatives should exist and be continuous and the partial derivatives must be such that learning always takes place in the presence of non-zero error. For a fuller discussion see [GK92]. In [GK92] a simple complex activation is proposed:

$$f(z) = \frac{z}{c + \frac{1}{r}|z|} \tag{24}$$

With c and r real values. When restricted to the real domain the function f looks like a sigmoid. This activation function has been used successfully in some simple applications, for example the complex encoder-decoder problem. The activation function above extends to the Clifford case, the partial derivatives are easy to work out(using the notation of the previous section):

$$\frac{\partial u_A}{\partial [x]_B} = \begin{cases} -\frac{r[x]_A[x]_B}{(c+\frac{1}{r}|x|)^2 r|x|} & \text{if } |x| > 0 \\ 0 & \text{if } |x| = 0 \end{cases} \tag{25}$$

if $A \neq B$ and if $A = B$ then,

$$\frac{\partial u_A}{\partial [x]_B} = \begin{cases} \frac{r(|x|^2 - [x]_A^2 + cr|x|)}{|x|(cr+|x|)^2} & \text{if } |x| \neq 0 \\ \frac{1}{c} & \text{if } |x| = 0. \end{cases} \tag{26}$$

The norm being the Clifford norm defined in the previous section.

3.2 Results on Encoder-decoder problems

The encoder-decoder problem is often used to test new techniques in back error propagation. While it is not a formal benchmark it us useful to get a feel of how new algorithms can perform. Essentially for a network to solve the encoder-decoder problem a training set is presented to the network which forces the network to encode the training set in some say. For instance in the real case with a 3-2-3 network if the network is trained on the set of vectors $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ then the two hidden units will learn a binary coding of the input signals.

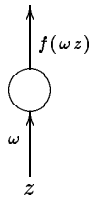
Clifford networks have been trained on a variety of encoder-decoder problems. Figure 1 shows the mean square error against epochs and figure 8 shows the output of the network after training of a 3-2-3 encoder problem, more extensive results can be found in [PB94a]. It is interesting to note that the epoch count is approximately the same as would be expected for a similar problem on a conventional BEP network, this suggests that the added complexity of the Clifford numbers it not affecting the convergence or pattern extraction efficiency of the network.

4 Relation to real valued networks

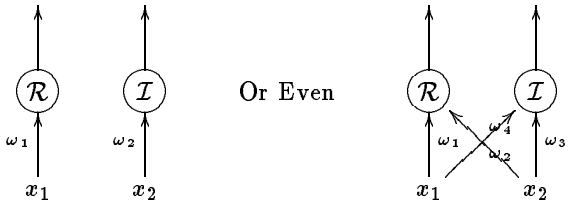
Given x inputs, y hidden neurons and z outputs, what is its relation to a real valued network with $x2^n$ inputs $y2^n$ hidden units and $z2^n$ outputs (where 2^n is

the dimension of the algebra in question, all Clifford algebras have dimension 2^n for some n)? Although real and Clifford networks can represent the same class of functions as real valued networks [3], there is not a direct and simple mapping between them. To make things simpler the complex case is concentrated on, these observations scale up to any Clifford algebra without difficulty.

Consider a single complex neuron with one input and no threshold:



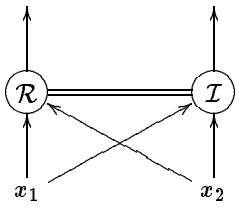
It might be thought that this is translatable into an ordinary real valued network in either of the following ways::



This is not so, writing out the equations for a single neuron we have:

$$f(\omega z) = \frac{\omega z}{1+|\omega z|} = \frac{(w_1 + w_2 i)(z_1 + z_2 i)}{1+|wz|}$$
$$= \frac{(\omega_1 z_1 - \omega_2 z_2)}{1+|\omega z|} + i\frac{(\omega_2 z_1 + \omega_1 z_2)}{1+|\omega z|} \tag{27}$$

This indicates the second diagram is more appropriate, but each term in equation (27) has $1+|\omega z|$, this involves weight values from both neurons and hence implies that there is some form of cross linkage between neurons.



Where the linkage terms are supplying the extra contributing factors to make up the denominators. This clearly shows that there is no simple relationship between a Clifford valued network and a real valued network. It also indicated that Clifford networks may have the potential to form hidden unit encodeings of input data that are in some way more efficient, or that they are able to represent more complex pattern relationships.

[3] See section 7.

5 Metric space theory for Approximation

Mathematically some care has to be taken with the notion of approximation. What is required is some function E which given two functions ϕ,ψ gives:

$$E(\phi, \psi) = 0 \text{ if and only if } \phi = \psi \tag{28}$$

and $E(\phi, \psi)$ be close to zero if ϕ is 'close' to ψ. Formalizing these requirements leads to the notion of a metric space (see [Cop88] for a good introduction), which is defined as a set X together with a distance metric $d : X \to \mathbf{R}$ satisfying the following axioms:

$$d(x, x) = 0 \qquad \text{for all } x \in X \tag{29}$$

$$d(x, y) = d(y, x) \qquad \text{for all } x, y \in X \tag{30}$$

$$d(x, z) \leq d(x, y) + d(y, z) \qquad \text{for all } x, y, z \in X \tag{31}$$

.

Various choices of metrics are available for function spaces, first the metric of uniform convergence:

$$E(\phi, \psi) = \sup\{|\phi(x) - \psi(x)| \ | x \in X\} \tag{32}$$

where X is a subset on which both ϕ and ψ are defined, more formally $X \subseteq \text{dom}(\phi) \cap \text{dom}(\psi)$. To guarantee this value is defined extra conditions either have to be imposed on X or on ϕ and ψ. If X is stipulated to be compact (i.e. X is bounded), or extra conditions or imposed on the asymptotic behavior of the functions ϕ and ψ as they both tend to infinity then in both cases the equation (32) will be well defined. In this paper only compactness assumptions will be needed. Uniform convergence metrics are useful where networks are required to perform equally well over the whole of the input space.

Second there are the L_p metrics which are much easier to deal with mathematically and are defined as:

$$\rho_{p,\mu}(\phi, \psi) = \left(\int |\phi - \psi|^p d\mu \right)^{\frac{1}{p}} \tag{33}$$

Again extra conditions have to be stipulated to make the metric always well defined; similar to the conditions for the metric (32) above, see [Hor91] for details.

In order to ask the question whether a class of networks can approximate a class of continuous functions, the concept of density is needed. Given a class of functions C and a function norm $|\cdot|$ a subset S is said to be dense in C if the closure of S is the whole of C. What this means in practical terms for neural networks is that a class S of neural networks is dense in a function space C, if given a function $f \in C$ and an arbitrary $\epsilon > 0$ there exists a $g \in S$ such that $|f - g| < \epsilon$. Many theorems prove that Neural networks are universal approximators by showing the the function space of Neural networks is dense in an appropriate function space.

6 Clifford modules

This section deals with a generalization of vector spaces, the theory of Modules over rings: specifically Clifford modules. Various theorems are stated which are

generalizations of traditional theorems such as the Hahn-Banach theorem and the Riesz representation theorem [Rud66, Rud73]; all proofs are omitted, but these can be found in [BDS82].

From now on the convention adopted in [BDS82] is used, where a Euclidean Clifford algebra, that is an algebra of signature $0, n$, is referred as an $\mathcal{A}$ algebra. A module is a generalization of a vector space, where the set of coefficients come from some ring instead of a field, thus modules have a different geometrical structure from vector spaces.

Definition 1 *A unitary left* $\mathcal{A}$-module $X_{(l)}$ *is an Abelian group* $X_{(l)}, +$ *and an operation* $(\lambda, f) \to \lambda f$ *from* $\mathcal{A} \times X_{(l)}$ *into* $X_{(l)}$ *s.t. for all* $\lambda, \mu \in \mathcal{A}$ *and* $f, g \in X_{(l)}$ *the following hold:*

$$(\lambda + \mu)f = \lambda f + \mu f \tag{34}$$

$$(\lambda\mu)f = \lambda(\mu f) \tag{35}$$

$$\lambda(f + g) = \lambda f + \lambda g \tag{36}$$

$$e_0 f = f \tag{37}$$

Definition 2 *Let* $X_{(l)}$ *be a unitary left* $\mathcal{A}$*-module, then a function* $p : X_{(l)} \to \mathbf{R}$ *is said to be a* proper semi-norm *if there exists a constant* $C_0 \geq 0$ *s.t. for all* $\lambda \in \mathcal{A}$ *and* $f, g \in X_{(l)}$ *the following conditions are satisfied:*

$$p(f + g) \leq p(f) + p(g) \tag{38}$$

$$p(\lambda f) \leq C_0 |\lambda| p(f) \tag{39}$$

$$p(\lambda f) = |\lambda| p(f) \quad \text{if } \lambda \in \mathbf{R} \tag{40}$$

$$\text{If } p(f) = 0 \text{ then } \quad f = 0 \tag{41}$$

Definition 3 *Given a module* $X_{(l)}$ *the* algebraic dual $X^{*alg}_{(l)}$ *is defined to be the set of left* $\mathcal{A}$*-linear functionals from* $X_{(l)}$ *into* $\mathcal{A}$.

That is the set of functionals $T : X_{(l)} \to \mathcal{A}$ *s.t.*

$$T(\lambda f + g) = \lambda T(f) + T(g) \tag{42}$$

$f, g \in X_{(l)}$ *and* $\lambda \in \mathcal{A}$.

Definition 4 *The set of bounded* T *functionals with respect to a semi-norm* p *is denoted* $X^*_{(l)} \subset X^{*alg}_{(l)}$. *Explicitly for all functionals* T *and for all* $f \in X_{(l)}$*:*

$$|T(f)| \leq C p(f) \tag{43}$$

for some real constant C.

The following theorem is a a corollary to a Hahn-Banach type theorem for Clifford modules for details and proof see sections 2.10-2.11 in [BDS82].

Theorem 1 *Let* $X_{(l)}$ *be a unitary left* $\mathcal{A}$*-module provided with a semi norm* p *and let* $Y_{(l)}$ *be a submodule of* $X_{(l)}$. *Then* $Y_{(l)}$ *is dense in* $X_{(l)}$ *if and only if for each* $T \in X^*_{(l)}$ *such that* $T|Y_{(l)} = 0$[4] *we have* $T = 0$ *on* $X_{(l)}$.

[4] T restricted to $Y_{(l)}$ equal to zero

Now a useful class of function spaces is introduced.

Definition 5 *The space $C^0(\mathcal{K};\mathcal{A})$. Let $\mathcal{K}$ be a compact subset of $\mathbf{R}^r$ ($r \geq 1$). Then $C^0(\mathcal{K};\mathcal{A})$ stands for the unitary bi-$\mathcal{A}$-module of $\mathcal{A}$-valued continuous functions on $\mathcal{K}$.*

This can be thought of as a product of classical real valued functions i.e.:

$$C^0(\mathcal{K};\mathcal{A}) = \Pi_A C^0(\mathcal{A};\mathbf{R})e_A \tag{44}$$

where A runs over all the basis elements in the Clifford algebra in question. A norm can be defined for each $f \in C^0(\mathcal{K};\mathcal{A})$:

$$||f|| = \sup_{x\in\mathcal{K}} |f(x)| \tag{45}$$

This norm is equivalent to the product norm taken from (44).

Definition 6 *Given an open set $\Omega \subset \mathbf{R}^n$ and a sequence $(\mu_B)_B$ of real valued measures on Ω. Then for any open set in Ω an $\mathcal{A}$ valued measure can be defined:*

$$\mu(I) = \sum_B \mu_B(I)e_B \tag{46}$$

Definition 7 *An $\mathcal{A}$-valued function:*

$$f = \sum_A f_A e_A \tag{47}$$

is said to be μ-integrable in Ω if for all A and B ranging over the basis elements of $\mathcal{A}$ each f_A is μ_B integrable.

Definition 8 *For any μ-integrable function f define:*

$$\int_\Omega f(x)d\mu = \sum_{A,B} e_A e_B \int_\Omega f_A(x)d\mu_B \tag{48}$$

A Riesz representation type theorem can be obtained.

Theorem 2 *Let T be a bounded $\mathcal{A}$ valued function in $C^0_{(l)}(\mathcal{K};\mathcal{A})$. Then there exists a unique $\mathcal{A}$ valued measure μ with support contained in $\mathcal{K}$ such that for all $f \in C^0_{(l)}(\mathcal{K};\mathcal{A})$:*

$$T(f) = \int_\mathcal{K} f(x)d\mu \tag{49}$$

For a proof again see [BDS82].

7 The Approximation result

A feed-forward network with one output neuron and N inputs units and K hidden units computes a function:

$$\Phi(\mathbf{x}) = \sum_{j=1}^{K} \alpha_j f(\sum_{i=1}^{N} y_{ij}x_i + \theta_j) \tag{50}$$

with f the activation function x_i the i'th input, y_{ij} weight values for the connection between the input layer and the hidden layer and α_j the weights from the hidden layer to the output node.

$\Phi(\mathbf{x})$ can be seen as a function from $\mathbf{R}^{N2^n}$ (where 2^n is the dimension of $\mathcal{A}$) to $\mathcal{A}$ and hence a member of $C^0_{(l)}(\mathbf{R}^{N2^n};\mathcal{A})$. This is why the material of the last section was relevant. The next definition is important. What is shown is that all activation functions satisfying the definition, when used in feed-forward networks, are universal approximators. Then to complete the proof all that is needed to show is that the activation functions considered in section 3.1 satisfy the definition.

Definition 9 *An activation function f (considered as a function from $\mathbf{R}^{N2^n}$ to $\mathcal{A}$) is said to be* discriminating *if for any given Clifford valued measure μ with support I^{N2^n} if:*

$$\int_{I^{N2^n}} f(\sum_{i=1}^{N} y_i x_i + \theta) d\mu(x) = 0 \tag{51}$$

for all $y_i, \theta \in \mathbf{R}_{0,n}$ implies $\mu(x) = 0$.

Theorem 3 *Let f be any continuous discriminating functions. Then finite sums of the form:*

$$\Phi(x) = \sum_{j=1}^{K} \alpha_j f(\sum_{i=1}^{N} y_{ij} x_i + \theta_j) \tag{52}$$

are dense in $C^0_{(l)}(I^{N2^n};\mathcal{A})$

Proof: This proof is essentially a modification of Cybenko's Theorem 1 in [Cyb89] using the theory of Clifford modules in the last section.

Let S be the function space generated by sums of the form (52). Assume that the closure of S is not all of $C^0_{(l)}(I^{N2^n};\mathcal{A})$; denote the closure of S by R. By the Hahn-Banach type theorem (1) there is a bounded linear functional T on $C^0_{(l)}(I^{N2^n};\mathcal{A})$, with $T \neq 0$ but $T(R) = T(S) = 0$. By Theorem 2 this bounded linear functional is of the form:

$$T(h) = \int_{I^{N2^n}} h(x)\mu(x) \tag{53}$$

for some measure μ and $h \in C^0_{(l)}(I^{k2^n},\mathcal{A})$. In particular since $f \in C^0_{(l)}(I^{k2^n},\mathcal{A})$ is in R, for any y_i:

$$T(f) = \int_{I^{N2^n}} f(\sum_{i=1}^{N} y_i x_i + \theta) d\mu(x) = 0 \tag{54}$$

Since f is discriminating this implies $\mu = 0$ contradicting our assumption hence S must be dense in $C^0_{(l)}(I^{N2^n};\mathcal{A})$. □

So to prove that the class of feed-forward networks considered in section 3.1 are universal approximators, we have to show that functions of the form:

$$f(x) = \frac{x}{1 + |x|} \tag{55}$$

are discriminating.

Theorem 4

$$f(x) = \frac{x}{1+|x|} \tag{56}$$

is discriminatory.

Proof: A function $f(x)$ is discriminatory , if:

$$\int_{I^{N2^n}} f(\sum_{i=1}^{N} y_i x_i + \theta) d\mu(x) = 0 \tag{57}$$

for all y_i implies that $\mu(x) = 0$. This is equivalent to saying that:

$$\int_{I^{N2^n}} f(\sum_{i=1}^{N} y_i x_i + \theta) d\mu(x) = \sum_{A,B} e_A e_B \int_{I^{N2^n}} f_A(\sum_{i=1}^{N} y_i x_i + \theta) d\mu_B(x) = 0 \tag{58}$$

for all y_i.

Define $\gamma_A(x) : I^{N2^n} \rightarrow \mathbf{R}$ to be the limit of:

$$\gamma_A(x) = \lim_{\lambda \rightarrow \infty} f_A(\lambda x) \tag{59}$$

(where λx is a Clifford multiplication, with λ a real number). So

$$f_A(\lambda x) = \frac{[\lambda z]_A}{1+|\lambda z|} = \frac{\lambda [z]_A}{1+\lambda |z|} \tag{60}$$

So

$$\gamma_A(z) = \begin{cases} 1 & \text{if} \quad [z]_A > 0 \\ 0 & \text{if} \quad [z]_A = 0 \\ -1 & \text{if} \quad [z]_A < 0 \end{cases} \tag{61}$$

In our case:

$$\gamma_A(\sum_{i=1}^{N} y_i x_i + \theta) = \begin{cases} 1 & \text{if} \quad [\sum_{i=1}^{N} y_i x_i + \theta]_A > 0 \\ 0 & \text{if} \quad [\sum_{i=1}^{N} y_i x_i + \theta]_A = 0 \\ -1 & \text{if} \quad [\sum_{i=1}^{N} y_i x_i + \theta]_A < 0 \end{cases} \tag{62}$$

The sets defined by $[\sum_{i=1}^{N} y_i x_i + \theta]_A = 0$ are hyper-planes, since $[\sum_{i=1}^{N} y_i x_i + \theta]_A$ is just a set of linear equations in the components of x_i.

The rest of the proof is almost verbatim from Lemma 1 of Cybenko [Cyb89]. So let $\Pi^A_{y,\theta} \subset I^{2^n}$ be the hyper-plane defined by:

$$\left\{ x | \left[\sum_{i=1}^{N} y_i x_i + \theta \right]_A = 0 \right\} \tag{63}$$

and let $H^A_{y,\theta}$ be the half space defined by:

$$\{x | [\sum_{i=1}^{N} y_i x_i + \theta]_A > 0\} \tag{64}$$

Then by the Lebesgue bounded convergence theorem we have:

$$0 = \int_{I^{N2^n}} f_A(\lambda x) d\mu_B(x) = \int_{I^{N2^n}} \gamma_A(x) d\mu_B(x) = \mu(H^A_{y,\theta}) \tag{65}$$

Now if μ_B were always a positive measure the result would be trivial, but since μ_B is an arbitrary measure the result is harder (since positive bits of μ might cancel out negative bits of μ_B).

Fix the y_i's and define:

$$F_A(h) = \int_{I^{N2^n}} h([\sum_{i=1}^{K} y_i x_i]_A) \tag{66}$$

for some bounded μ_B measurable function $h : \mathbf{R} \to \mathbf{R}$. F_A is a bounded functional on $L^\infty(\mathbf{R})$.

Let h be the indicator function on the interval $[\theta_A, \infty)$, then:

$$F(h) = \int_{I^{N2^n}} h([\sum_{i=1}^{K} y_i x_i]_A) = \mu_B(\Pi^A_{y,\theta}) + \mu(H^A_{y,\theta}) \tag{67}$$

Similarly $F(h) = 0$. If h is the indicator of any open interval, by linearity $F(h) = 0$ and hence for any simple function. Since the simple functions are dense in $L^\infty(\mathbf{R})$, $F = 0$.

In particular given the two functions $s(x) = \sin(x)$, $c(x) = \cos(x)$:

$$F_A(s(x) + ic(x)) = \int_{I^{N2^n}} s([\sum_{k=1}^{K} y_k x_k]_A) + ic([\sum_{k=1}^{K} y_k x_k]_A) d\mu_B =$$

$$\int_{I^{N2^n}} \exp(i[\sum_{k=1}^{K} y_k x_k]_A) d\mu_B = 0 \tag{68}$$

for all y_k. Therefore the Fourier transform of μ_B is zero, hence μ_B must be zero and hence f is discriminatory. □

One important thing to point out with this proof is that the order of weight multiplication is irrelevant; the whole proof could be repeated with networks where multiplication was done on the right. Thus it does not matter theoretically which sort of nets (left or right weight multiplication) are used for a particular problem. Practically not much is known, but in all the examples the author has tried, the performance of the net does not seem to be affected by the order of weight multiplication.

Thus it has been shown that Clifford valued networks with values taken from Euclidean algebras, which include the complex numbers and the Quaternions, are universal approximators in the sense of [Cyb89]. That is any continuous function on a bounded Clifford domain can be approximated to any degree of accuracy by a Clifford valued network. Further more both left and right valued weight multiplication networks have been shown to have the same computational power. Which due to the non-commutative nature of Clifford algebras was not a priori obvious.

8 Conclusion and further work

This paper has presented the Back Error Propagation algorithm for Clifford valued networks and has demonstrated that feed-forward Clifford networks are able to solve simple problems. In [Pea94] more extensive encoder-decoder problems

are solved with Clifford networks, together with a multi-dimensional extension of Rosenblatt's Perceptron building on the work of [Geo93]. It is still to be demonstrated whether or not representing multi-dimensional signals in single Clifford values will necessarily give a more efficient representation of problem domains, however it is likely that for specific domains this may be the case. It is hoped that applications from physics ([CC86]) will furnish examples. One possibility is, given that Clifford numbers are vectors in some 2^n dimensional space, the direction of the number can be taken as a symbol representing a class or outcome, while the magnitude of the number could represent confidence in the outcome. In the complex case, a number such as $\alpha e^{i\theta}$, θ would represent some value and α some degree of confidence in that value. The benifit of going to higher dimensions would result, not in more symbols, but in more degrees of freedom in representing the symbols.

References

[BDS82] F. Brackx, R. Delenghe, and F. Sommen. *Clifford Analysis*. Research notes in mathematics; 76. Pitman Advanced Publishing Program, 1982.

[BLJM89] H Blaine Lawson Jr. and Marie-Louise Michelsohn. *Spin Geometry*. Princeton University Press, Princeton, New Jersey, 1989.

[CC86] J.S.R Chisholm and A.K. Common, editors. *Clifford algberas and theier applications in mathematical physics*, volume 183 of *NATO ASI series, C:Mathematical and physical sciences*, 1986.

[Cop88] E.T. Copson. *Metric Spaces*. Cambridge University Press, first paper back edition edition, 1988.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals and Systems*, pages 303–314, 1989.

[Geo93] George M. Georgiou. The multivalued and continuous perceptrons. In *World Congress on Neural Networks*, volume IV, pages 679–683, Portland, OR, 1993.

[GK92] George M. Georgiou and Cris Koutsougeras. Complex domain back-propagation. *IEEE Transactions on Circuits and Systems*, pages 330–334, 1992.

[Hir92a] A. Hirose. Continuous complex-valued back-propagation learning. *Electronics Letters*, 20(20):1854–1855, September 1992.

[Hir92b] A. Hirose. Dynamics of fully complex-valued neural networks. *Electronics Letters*, 28(16):1492–1493, July 1992.

[Hir93] Aikra Hirose. Motion controls using complex-valued neural networks with feedback loops. In *IEEE International conference on neural networks*, 1993.

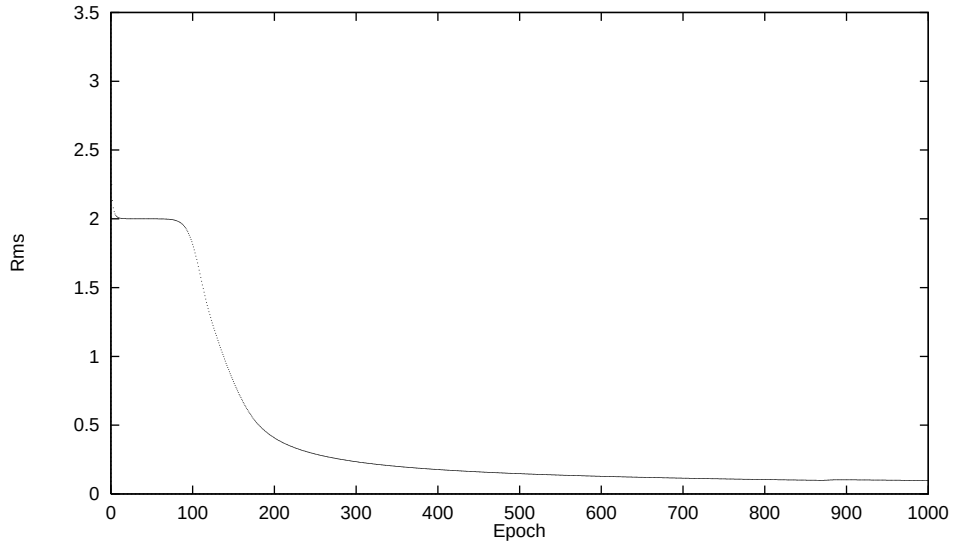


Figure 1: Graphs of RMS error for a 3-2-3 encoder-decoder problem over the algebra $\mathbf{R}_{0,2}$

[Hor91] K Hornick. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

[HSW89] K Hornick, Stinchcombe, and White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[PB92] J.K. Pearson and D.L. Bisset. Back Propagation in a Clifford Algebra. In *ICANN Brighton*, September 1992.

[PB94a] J.K. Pearson and D.L. Bisset. Clifford networks an introduction. Avalible for ftp from `ftp.cis.ohio-state.edu` as `/pub/neuroprose/pearson.clifford.ps.Z`, February 1994.

[PB94b] J.K. Pearson and D.L. Bisset. Neural Networks in the Clifford Domian. In *IEEE94 symposium on Neural Networks Orlando*, June 1994.

[Pea94] Justin K. Pearson. *Neural Networks over Clifford Algebras.* PhD thesis, University of Kent, 1994.

[Por81] I.R. Porteous. *Topological Geometry.* Cambridge University Press, 1981.

[Rud66] Walter Rudin. *Real and complex analysis.* McGraw-Hill series in higher mathematics. McGraw-Hill, 1966.

[Rud73] Walter Rudin. *Functional analysis.* McGraw-Hill series in higher mathematics. McGraw-Hill, 1973.

Pattern 0	(1.00000,0.00000,0.00000,0.00000) (0.00000,0.00000,0.00000,0.00000) (0.00000,0.00000,0.00000,0.00000)
Output	(0.82741,0.00180,-0.00034,0.00075) (-0.00896,-0.01346,0.03117,-0.00653) (-0.02300,0.00890,-0.00371,-0.00493)
Pattern 1	(0.00000,0.00000,0.00000,0.00000) (1.00000,0.00000,0.00000,0.00000) (0.00000,0.00000,0.00000,0.00000)
Output	(0.01103,0.02384,-0.00875,-0.00106) (0.82393,-0.00412,-0.00112,-0.00050) (-0.02084,0.00501,-0.00617,0.00553)
Pattern 2	(0.00000,0.00000,0.00000,0.00000) (0.00000,0.00000,0.00000,0.00000) (0.00000,0.00000,0.00000,-1.00000)
Output	(0.01646,0.03726,-0.02079,-0.00340) (-0.01691,-0.02265,0.03517,-0.00731) (-0.00301,0.00194,-0.00061,-0.82819)

Table 1: Outputs for a trainied 3-2-3 encoder-decoder network