

# PARÂMETRO DE EXATIDÃO PARA APROXIMAÇÃO DE FUNÇÕES UTILIZANDO MULTILAYER PERCEPTRONS NOS DOMÍNIOS REAL, COMPLEXO E DE CLIFFORD

Thalles S. Torchi<sup>1</sup>, Milton R. Romero<sup>2</sup> e Evandro M. Martins<sup>3</sup>

Depto. de Eng. Elétrica, UFMS, Caixa Postal 549, CEP 79070-900, Campo Grande, MS  
<sup>1</sup>thalles.torchi@gmail.com, <sup>2</sup>miltonr@del.ufms.br e <sup>3</sup>evandro.mazina@ufms.br

**Resumo** - Na fase de utilização, as Redes Neurais Artificiais (RNA) *Multilayer Perceptrons*, treinadas com o algoritmo de *Backpropagation*, não conseguem aproximar a função de interesse para 100% dos dados de entrada. Este trabalho propõe uma metodologia para abordagem de dois pontos de interesse: **1º**) estimar um parâmetro de exatidão para as saídas de RNA na fase de utilização, com o objetivo de definir quais saídas podem ser consideradas confiáveis e quais não, definindo como confiáveis as saídas que se aproximam do comportamento da função de interesse; e **2º**) estabelecer o número de padrões a serem utilizados na fase de treinamento, que permitam a convergência e a generalização da rede na metodologia proposta. A metodologia baseia-se no treino e utilização de duas redes: a RNA Direta (RNAD), utilizada para aproximar a função de interesse, e a RNA Inversa (RNAI), utilizada para aproximar a inversa (FI) da função de interesse. Caso a função a ser aproximada não tenha FI definida, o domínio é restringido para onde exista. Na utilização destas redes será computada a diferença entre a entrada da RNAD e a saída da RNAI. Quando a entrada da RNAD e a saída da RNAI forem computacionalmente iguais, ou seja, sua diferença muito próxima de zero, tanto quanto à aplicação exigir, será considerado que a saída da rede direta (RNAD), isto é, a aproximação da função de interesse, é confiável. O método é comprovado experimentalmente a partir de dados sintéticos, utilizando a função  $f(x) = x^2$ , a fim de permitir o controle entre as entradas e saídas das redes com o intuito de validação do método nos domínios Real, Complexo e de Clifford. Os dados sintéticos e não dados de aplicações reais, se utilizam para demonstrar a viabilidade do algoritmo permitindo comparar os três domínios, pois possíveis erros contidos nos dados reais se mesclariam com possíveis erros no algoritmo dificultando a validação do método proposto. Os resultados mostram que o método é robusto e permite determinar o parâmetro de exatidão para as saídas da RNA, o critério de convergência e a qualidade da generalização das mesmas, permitindo a comparação gráfica dos três domínios.

**Palavras-chave** – RNA, parâmetro de exatidão, *Multilayer Perceptrons*, Domínio Complexo, Domínio Clifford.

## 1. INTRODUÇÃO

As Redes Neurais Artificiais (RNA) são constituídas de unidades de processamento denominadas neurônios artificiais que armazenam conhecimento experimental para posterior uso [1]. Uma prática estatística comum para a escolha do conjunto de treinamento dessas redes, dentre outros métodos [2], é o método *hold-out* [3], o qual reserva a maior parte dos dados disponíveis para treinamento, geralmente mais que 50%, e o restante para validação. Tendo como característica não desejável a admissão de subconjuntos de treinamento maiores ou menores que o necessário, isto incide diretamente nas fases de validação e utilização, pois as redes podem convergir de acordo com o critério de parada, mas não necessariamente generalizar a função desejada. Na fase de utilização, é uma característica inerente a estas redes que alguns valores de saída sejam diferentes do esperado. Em certas aplicações de controle é necessário utilizar unicamente saídas sobre as quais a aproximação da função de interesse seja considerada aceitável, caso contrário, não é recomendável utilizar [4], [5].

Este trabalho apresenta uma metodologia que fornecerá dois parâmetros: **1º**) para aferir a exatidão das saídas de RNA, definindo assim os valores de saída aceitáveis de acordo com a função de interesse; **2º**) para determinar a quantidade de padrões de treinamento necessários para permitir a convergência e a generalização destas redes de acordo com a metodologia.

A metodologia baseia-se no treino e utilização de duas redes: RNA Direta (RNAD), utilizada para aproximar a função de interesse, e a RNA Inversa (RNAI), utilizada para aproximar a Função Inversa (FI) da função de interesse. Caso a função a ser aproximada não tenha FI definida, o domínio é restringido para onde exista. Dependendo da aplicação, uma maneira

---

<sup>2</sup> Bolsista Capes Brasília/Brasil

prática de fazer a verificação da existência da inversa é fazendo um gráfico da função e visualmente verificando se a função é bijetiva. As classes de problemas admissíveis nesta metodologia restringem as aplicações aos problemas que requerem o uso de redes neurais utilizadas na aproximação de funções e que tenham inversa e se não tem inversa, que possam ser tratadas por intervalos onde exista a função inversa. Sobre o treino destas redes é computada a diferença entre a entrada da RNAD e a saída da RNAI, que devem ser computacionalmente iguais. Esta diferença é algebricamente igual a zero, mas pela representação de ponto flutuante no processamento, esta diferença deve ser verificada dentro de um valor muito próximo de zero, tanto quanto à aplicação exigir, onde esta diferença irá definir os parâmetros de convergência e generalização. Na utilização destas redes será computada também a diferença entre a entrada da RNAD e a saída da RNAI. Quando a entrada da RNAD e a saída da RNAI forem computacionalmente iguais, será considerado que a saída da rede direta (RNAD), isto é, a aproximação da função de interesse, é confiável.

Esta metodologia é aplicada para aproximar a função quadrática  $f(x) = x^2$ , utilizando as redes *Multilayer Perceptrons* nos domínios Real, Complexo e de Clifford, com o intuito de determinar os parâmetros propostos nos itens 1º e 2º de aferir a exatidão das saídas de RNA e determinar a quantidade de padrões de treinamento necessários para permitir a convergência e a generalização, respectivamente. A metodologia é demonstrada, utilizando-se a função quadrática (dados sintéticos), que tem a vantagem de: 1º permite a representação nos três domínios para poder comparar graficamente os resultados e permite definir um isomorfismo entre o domínio complexo e de Clifford; 2º não tem inversa para todo o domínio e pode se mostrar como se tratam essa classe de funções; 3º podem se controlar todos os parâmetros da metodologia para poder atribuir qualquer problema ao algoritmo e não aos dados; 4º é suficientemente suave para não precisar muito treinamento (precisa de poucas amostras para ser aproximada); 5º é suficientemente complexa para mostrar como a metodologia deve tratar com conjuntos que não são totalmente ordenados. Os dados sintéticos se utilizam para demonstrar a validade do algoritmo, pois ao se utilizar dados de aplicações reais os possíveis erros contidos nos dados se mesclariam com possíveis erros no algoritmo dificultando a validação do método proposto.

Este trabalho está organizado como segue: a Seção 2 apresenta os conceitos relevantes para a metodologia proposta das Redes Neurais Artificiais *Multilayer Perceptron* no domínio Real com o algoritmo de treinamento *Backpropagation*; a Seção 3 apresenta os aspectos relevantes da álgebra de Clifford para o algoritmo proposto; a Seção 4 estende as Redes Neurais Artificiais *Multilayer Perceptron* para os domínios Complexo e de Clifford com o algoritmo de treinamento *Backpropagation*; a Seção 5 apresenta a metodologia proposta; a Seção 6 ilustra os resultados obtidos e a discussão desses resultados; e a Seção 7 sugere os trabalhos futuros e as conclusões finais.

## 2. REDES NEURAIS ARTIFICIAIS

As Equações (1) e (2) descrevem o funcionamento do neurônio artificial [6], Fig. 1, e as relações entre os pesos  $w_{ij}$ , as entradas  $x_i$ , bias  $b_j$ , valor de ativação  $net_j$ , a Função de Ativação (FA)  $f(\cdot)$  e as saídas  $o_j$ , onde o estado do ambiente é representado pelas entradas e o conhecimento adquirido no treinamento é armazenado nos pesos.

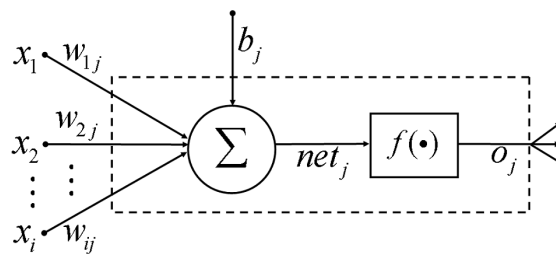


Fig. 1. Neurônio artificial.

$$net_j = \sum_i w_{ji}x_i + b_j \quad (1)$$

$$o_j = f(net_j) \quad (2)$$

$$o_j = f(net_j) = \frac{e^{net_j} - e^{-net_j}}{e^{net_j} + e^{-net_j}} \quad (3)$$

Este trabalho utiliza a FA Tangente Hiperbólica (FATH), Equação (3), a qual é escrita na forma de função elementar transcendental, que permite a mesma representação matemática para os domínios Real, Complexo e de Clifford, tendo apenas como diferença os valores assumidos por  $net_j$  [7]. O valor de  $net_j$  pertence ao domínio Real, Complexo ou de Clifford, de acordo com o domínio da rede neural artificial escolhida. O modelo de RNA adotado por este trabalho é o *Multilayer Perceptrons* (MLP) [8], Fig. 2, no qual se utiliza o algoritmo de treinamento *Backpropagation* (BP), que aproxima com apenas duas camadas ocultas qualquer função contínua [9]. O modelo MLP tem como característica uma camada de entrada onde são apresentados os dados de entrada  $X_i$ , uma camada de saída onde se apresentam as saídas  $O_j$  e uma ou mais camadas ocultas, situadas entre as camadas de entrada e de saída. Este modelo é utilizado com frequência para resolver problemas que não sejam linearmente separáveis, os quais são maioria no campo da Engenharia. Os valores atribuídos as entradas e saídas dessas camadas são modificadas de acordo com o domínio no qual a rede está definida.

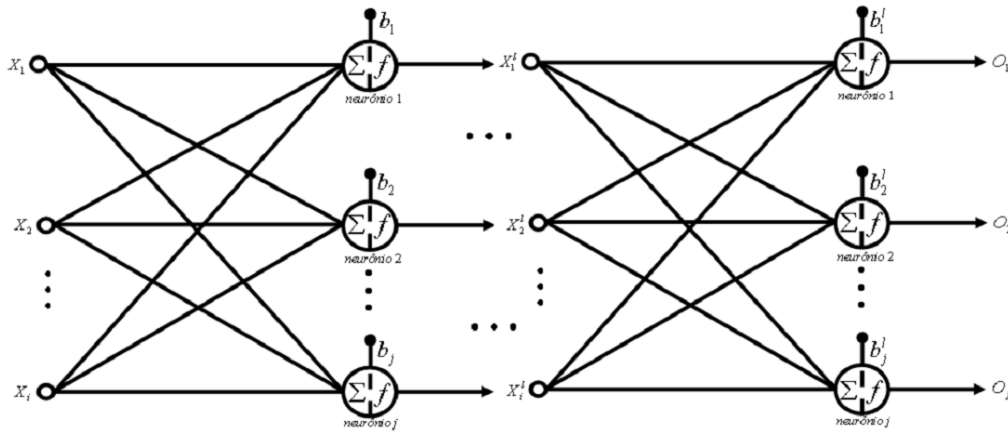


Fig. 2. Rede *Multilayer Perceptron* (MLP).

## 2.1 BACKPROPAGATION NO DOMÍNIO REAL

O BP é um algoritmo de aprendizagem supervisionada, que pode ser executado padrão a padrão [1] da seguinte maneira: Cada um dos  $i$ -ésimos exemplos  $x_i$  são aplicados à rede e esta produz as saídas  $o_j$ . As saídas estão baseadas no estado corrente dos pesos sinápticos, que são inicialmente definidos de forma aleatória. Cada uma destas saídas  $o_j$  é comparada com uma saída desejada  $t_j$  para determinar a diferença entre a saída desejada e a saída obtida  $e_j$ , Equação (4), onde  $n$  se refere ao passo de execução do algoritmo. A seguir é computado o erro  $E(n)$ , conhecido como *Sum of Squared Error* (SSE), Equação (5), onde  $N$  é a quantidade de padrões de treinamento. Esse erro é retropropagado pela rede, a atualização dos pesos de cada camada é proporcional a ele e tem como objetivo reduzi-lo a cada nova iteração. Esta rotina é repetida até que o SSE atinja o critério de parada que é um valor definido a *priori*. Alcançando esse valor, diz-se que a rede aprendeu ou está treinada [8].

$$e_j(n) = t_j(n) - o_j(n) \quad (4)$$

$$E(n) = \frac{1}{2} \sum_{j=1}^N (e_j(n))^2 \quad (5)$$

A regra *delta* de atualização dos pesos  $w_{ij}$  é descrita na Equação (6), onde  $\eta$  refere-se à taxa de aprendizagem que controla a velocidade da convergência,  $\delta_j$  é o termo de propagação do erro na saída e  $w_{ij}^{new}$  e  $w_{ij}^{old}$  são os valores do peso  $w_{ij}$  na iteração  $n+1$  e  $n$ , respectivamente. Caso o peso pertença à camada de saída,  $\delta_j$  deve ser o descrito na Equação (7), onde  $f'(\cdot)$  é a derivada primeira da FA e, portanto, a função de ativação deve ser derivável. Caso pertença a uma camada oculta,  $\delta_j$  deve ser o descrito na Equação (8), onde  $k$  é o número de neurônios na camada anterior ao se retropropagar o erro na saída da rede. O termo  $n$  (iterações) dessas equações é suprimido para melhor visualização das relações. Detalhes sobre o algoritmo de aprendizagem *Backpropagation* podem ser encontrados em [1], [8].

$$w_{ij}^{new} = w_{ij}^{old} + \eta \cdot \delta_j \cdot x_i \quad (6)$$

$$\delta_j = e_j \cdot f'(net_j) \quad (7)$$

$$\delta_j = \sum_k (\delta_k \cdot w_{jk}) \cdot f'(net_j) \quad (8)$$

### 3. ÁLGEBRAS DE CLIFFORD

As Álgebras de Clifford são um tipo de álgebra associativa, e são uma das possíveis generalizações dos números Complexos e dos *quaternions* [10], [11], [12], as quais são definidas da seguinte maneira: sejam  $a, b, c, \dots$  vetores pertencentes ao espaço vetorial  $V_n$   $n$ -dimensional sobre os reais e  $\alpha$  qualquer escalar Real. A álgebra geométrica  $Cl_{p,q} = Cl(V_n)$  com  $n = p + q$  é gerada a partir do espaço vetorial  $V_n$  onde é definido o produto geométrico  $\otimes$ , Equação (9), que é a combinação linear do produto interno  $\vee$  e do produto exterior  $\wedge$  onde o primeiro representa os escalares e o segundo os bivectores [12].

$$a \otimes b = a \vee b + a \wedge b = \frac{1}{2}(a \otimes b + b \otimes a) + \frac{1}{2}(a \otimes b - b \otimes a) \quad (9)$$

Este produto guarda as seguintes propriedades sobre todos os vetores:

1.  $a \otimes (b \otimes c) = (a \otimes b) \otimes c$ ;
2.  $a \otimes (b + c) = a \otimes b + a \otimes c$ ;
3.  $(b + c) \otimes a = b \otimes a + c \otimes a$ ;
4.  $b\alpha = \alpha b$ ;
5.  $b^2 = \pm |b|^2$ .

As álgebras de Clifford têm diversas assinaturas  $Cl_{p,q}$ , onde:  $p$  é a quantidade de elementos de base  $e_i$  que, elevados ao quadrado, são iguais a  $(+1)$  e  $q$  é a quantidade de elementos de base  $e_i$  que, elevados ao quadrado, são iguais a  $(-1)$ . Essas geram exatamente  $2^n$  combinações linearmente independentes entre a unidade escalar e os vetores de base  $\{1, e_1, e_2, \dots, e_n\}$  através do produto geométrico  $\otimes$ . Estes elementos satisfazem às seguintes relações:

$$(e_i)^2 = +1 \quad \text{com } i = 1, 2, 3, \dots, p$$

$$(e_i)^2 = -1 \quad \text{com } i = p + 1, p + 2, \dots, n$$

$$e_i \wedge e_j = -e_j \wedge e_i \quad \text{com } i = 1, 2, 3, \dots, p$$

Os elementos utilizados pela álgebra são chamados *k-blades*. Esses elementos são os escalares (*0-blades*), vetores (*1-blade*), segmentos de plano orientados (*2-blades*) e volumes orientados (*3-blades*) como mostra a Fig. 3, a qual exibe uma possível representação destes subespaços [13], [14]. Os *k-blades* nada mais são do que subespaços orientados [15], onde o produto geométrico  $\otimes$ , Equação (9), permite a modelagem dos espaços vetoriais conhecidos através desses elementos.

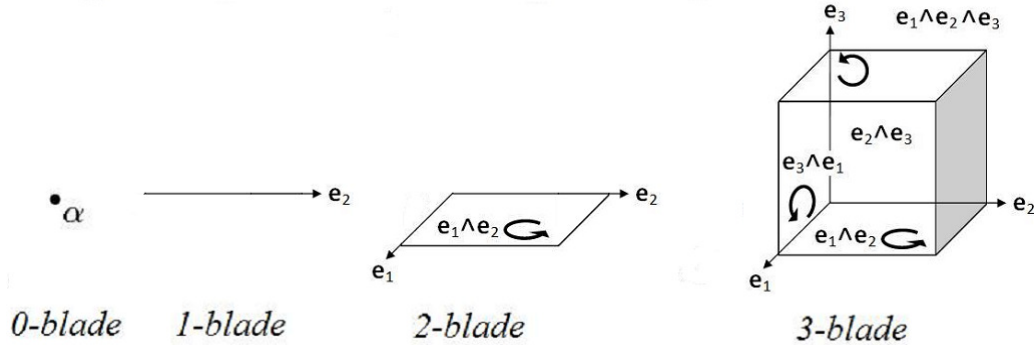


Fig. 3. *k-blades*: subespaços orientados.

Dependendo da assinatura da álgebra de Clifford, essa pode ter a álgebra dos números Complexos como subespaço [14], [15]. Como é o exemplo da assinatura  $Cl_{2,0}$  que tem a álgebra dos números Complexos expandida sobre os elementos de base  $\{1, e_1 \wedge e_2\}$  onde  $(e_1 \wedge e_2)^2 = -1$  é equivalente à unidade imaginária do plano Complexo. Para  $n=3$  as  $2^n$  combinações linearmente independentes definem a álgebra de Clifford de assinatura  $Cl_{3,0}$ , isomorfa ao espaço Euclidiano 3D que possui 8 elementos: escalar (1), vetores  $(e_1, e_2, e_3)$ , bivectores  $(e_1 \wedge e_2, e_2 \wedge e_3, e_3 \wedge e_1)$  e o trivetor ou pseudoescalar  $(e_1 \wedge e_2 \wedge e_3)$ , onde o símbolo  $(\wedge)$  identifica o produto exterior. O produto geométrico, para a álgebra de Clifford com assinatura  $Cl_{3,0}$ , está definido como segue, na Tabela 1, onde  $e_{12} = e_1 \wedge e_2$ ;  $e_{23} = e_2 \wedge e_3$ ;  $e_{31} = e_3 \wedge e_1$ ;  $e_{123} = e_1 \wedge e_2 \wedge e_3$ .

$\otimes$	$e_1$	$e_2$	$e_3$	$e_{12}$	$e_{23}$	$e_{31}$	$e_{123}$
$e_1$	1	$e_{12}$	$-e_{31}$	$e_2$	$e_{123}$	$-e_3$	$e_{23}$
$e_2$	$-e_{12}$	1	$e_{23}$	$-e_1$	$e_3$	$e_{123}$	$e_{31}$
$e_3$	$e_{31}$	$-e_{23}$	1	$e_{123}$	$-e_2$	$e_1$	$e_{12}$
$e_{12}$	$-e_2$	$e_1$	$e_{123}$	-1	$-e_{31}$	$e_{23}$	$-e_3$
$e_{23}$	$e_{123}$	$-e_3$	$e_2$	$e_{31}$	-1	$-e_{12}$	$-e_1$
$e_{31}$	$e_3$	$e_{123}$	$-e_1$	$-e_{23}$	$e_{12}$	-1	$-e_2$
$e_{123}$	$e_{23}$	$e_{31}$	$e_{12}$	$-e_3$	$-e_1$	$-e_2$	-1

Tabela 1. Produto geométrico com assinatura  $Cl_{3,0}$ .

#### 4. REDES NEURAIS ARTIFICIAIS NOS DOMÍNIOS COMPLEXO E DE CLIFFORD

Do ponto de vista estrutural, projetar RNA que possam processar sinais Complexos ou de Clifford, este último apresentado na Seção 5, é semelhante a projetar RNA de domínio Real. Os conceitos de topologia de rede, modos de treinamento, critérios de parada e validação do treinamento são análogos às RNA de domínio Real [11], [16], [17], [18].

A diferença encontra-se nos valores de entrada, alvos e pesos que são ou números Complexos ou números de Clifford e, por consequência, necessitam de FA definidas em seu domínio de atuação. Entretanto, certos cuidados devem ser tomados ao se escolher a FA a ser implementada, pois essa escolha tem influência direta na adaptação do algoritmo BP, tanto para o domínio Complexo quanto para o de Clifford. Esta influência está ligada aos tipos de função que as RNA implementadas com determinada FA complexa ou de Clifford podem aproximar [17],[18] e na adaptação do algoritmo BP para estes domínios onde há a necessidade de que a FA seja derivável para poder efetuar o treinamento. O algoritmo pode ser implementado usando as derivadas totais ou parciais [11], [19], [20], [21], [22], [23]. Tanto as FA complexas quanto as de

Clifford podem ser divididas em três classes de interesse: as **inteiras não-limitadas**, **não-inteiras limitadas** e as **inteiras limitadas** [11]. Essa última classe limita a quantidade de funções que as RNA complexas podem aproximar e consequentemente esta limitação também estará presente em álgebras de Clifford de assinatura isomorfa à álgebra dos números Complexos. Isso é explicado pelo teorema de **Liouville**, o qual estabelece que qualquer função complexa inteira e limitada no domínio Complexo é uma função constante e, portanto, não é adequada para a implementação do algoritmo de treinamento BP [11]. No domínio Complexo e no domínio de Clifford, a FATH é chamada de *fully* tangente hiperbólica, Equação (3). Essa função representa a classe de funções inteiras não-limitadas, Fig. 4, onde  $a_j$  é a parte real e  $d_j$  a parte imaginária do argumento da função, e  $O_j$  a saída da função.

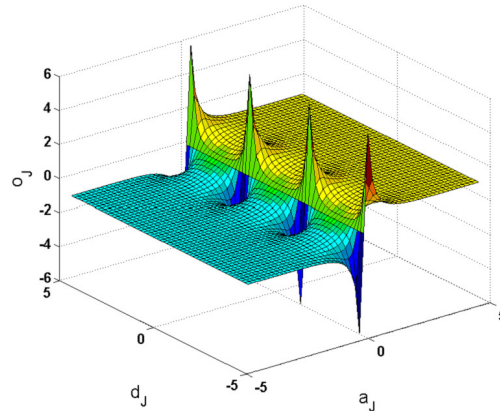


Fig. 4. Função de Ativação Fully Tangente Hiperbólica.

No domínio Complexo, essa FA possui valores que tendem ao infinito, valores singulares, periódicos em  $net_j \in 0 \pm i(2q + \frac{1}{2})\pi$  com  $q \in \mathbb{N}$ . Caso a saída do neurônio situe-se em alguma singularidade, representada pelos picos na Fig. 4, onde  $net_j = a_j + id_j$ , a derivada da FA assumirá um valor elevado e consequentemente a correção dos pesos poderá ser maior que o necessário, podendo elevar o tempo de convergência ou até mesmo comprometê-lo [21]. Para contornar esse problema, em [20] sugere-se que os padrões de treinamento, validação, utilização e os pesos iniciais devam ser escalados para uma região do plano Complexo onde as singularidades tenham menor influência. Essas mesmas medidas podem ser tomadas quanto à FA *fully* tangente hiperbólica de Clifford, desde que a assinatura da álgebra seja isomorfa à álgebra dos números Complexos. Definições e características detalhadas com relação aos tipos de FA de domínio Complexo podem ser encontradas em [7] e para o domínio de Clifford em [11],[21].

#### 4.1 BACKPROPAGATION COMPLEXO E DE CLIFFORD

Para que o algoritmo BP possa processar no domínio Complexo ou de Clifford, é necessário redefinir para esses domínios o SSE, Equação (5), e as Equações (6), (7) e (8) que compõem a regra *delta* de atualização dos pesos e consequentemente a FA. Este trabalho adota a regra *delta* de domínio Complexo e de Clifford, esse último de assinatura  $Cl_{0,3}$ , para o uso da FA *fully* tangente hiperbólica definidos nas Equações (11), (12), (13) e (14). Essa regra possui a mesma representação matemática para os dois domínios devido ao isomorfismo.

A generalização dessas relações para qualquer álgebra de Clifford de assinatura  $Cl_{p,q}$  pode ser encontrada em [11]. O erro na saída da rede complexa e na rede de Clifford é definido de forma análoga ao domínio Real, Equação (4), com a diferença que  $e_j(n) \in \{\mathbb{C} \text{ ou } Cl_{0,3}\}$ . Há consenso que a função custo, Equação (10), tanto no domínio Real [1], Complexo [7], [18], [19], [20], [21] ou de Clifford [11], [13], [16], [22], deva ser uma função Real. Nas Equações (10), (11), (12) e (13), o operador utilizado nas equações deve ser mudado de acordo com o domínio, portanto, se o domínio for complexo, o operador  $\diamond \Leftrightarrow \bullet$  representa a multiplicação convencional; caso o domínio seja o de Clifford  $Cl_{0,3}$ , o operador  $\diamond \Leftrightarrow \otimes$  representa o produto geométrico.

$$E = \frac{1}{2} \sum_{j=1}^N (e_j \diamond e_j^*) \quad (10)$$

$$\delta_j = e_j \diamond f'(net_j)^* \quad (11)$$

$$\delta_j = \left( \sum_k \delta_k \diamond w_{jk}^* \right) \diamond f'(net_j)^* \quad (12)$$

$$w_{ij}^{new} = w_{ij}^{old} + \eta \cdot \delta_j \diamond x_i^* \quad (13)$$

O asterisco (\*) sobrescrito implica no conjugado complexo ou de Clifford [24]. No domínio complexo, o complexo conjugado de um número  $z = a \pm jb$  é dado por  $z^* = a \mp jb$ , ou seja, basta inverter o sinal da parte imaginária. Já no domínio de Clifford, o processo é semelhante com a diferença que o complexo conjugado de cada um dos *k-blades* depende do *grade k*, onde *grade* corresponde à dimensão do *blade*, à qual o *blade* pertence. Tomemos como exemplo o multivetor **A** dado por:

$$\mathbf{A} = \underbrace{\alpha_1}_{0\text{-blade}} + \underbrace{\alpha_2 \mathbf{e}_1 + \alpha_3 \mathbf{e}_2 + \alpha_4 \mathbf{e}_3}_{1\text{-blades}} + \underbrace{\alpha_5 \mathbf{e}_1 \mathbf{e}_2 + \alpha_6 \mathbf{e}_1 \mathbf{e}_3 + \alpha_7 \mathbf{e}_2 \mathbf{e}_3}_{2\text{-blades}} + \underbrace{\alpha_8 \mathbf{e}_1 \mathbf{e}_2 \mathbf{e}_3}_{3\text{-blades}}$$

Os sinais de cada um dos *k-blades* do complexo conjugado de Clifford são dados pela expressão  $(-1)^{k_m \frac{k_m+1}{2}}$ , onde  $k_m = (k \bmod 4)$ . A Tabela 2 descreve essa relação.

<i>k-blade</i>	$k_m = k \bmod 4$	<i>sinal do conjugado</i>
0-blade	0	+
1-blade	1	-
2-blade	2	-
3-blade	3	+
4-blade	0	+
5-blade	1	+
⋮	⋮	⋮

Tabela 2. Sinal do conjugado do *blade* de *grade k*.

Logo, o complexo conjugado do multivetor **A** será:

$$\mathbf{A}^* = \underbrace{\alpha_1}_{0\text{-blade}} - \underbrace{\alpha_2 \mathbf{e}_1 - \alpha_3 \mathbf{e}_2 - \alpha_4 \mathbf{e}_3}_{1\text{-blade}} - \underbrace{\alpha_5 \mathbf{e}_1 \mathbf{e}_2 - \alpha_6 \mathbf{e}_1 \mathbf{e}_3 - \alpha_7 \mathbf{e}_2 \mathbf{e}_3}_{2\text{-blade}} + \underbrace{\alpha_8 \mathbf{e}_1 \mathbf{e}_2 \mathbf{e}_3}_{3\text{-blade}}$$

## 5. METODOLOGIA DIRETA-INVERSA: DOMÍNIO REAL

A metodologia proposta por este trabalho baseia-se no treinamento de duas RNA: uma rede para aproximar a função de interesse, RNAD, e uma segunda para aproximar a função inversa da primeira, RNAI, como mostra a Fig. 5. Sejam, respectivamente:

- *X, T* e *O*: o vetor de entradas, alvos e saídas da RNAD;
- *O, X* e  $X^\alpha$ : o vetor de entradas, alvos e saídas da RNAI;
- *N*: o número de padrões de treinamento.

Inicialmente, foi realizada apenas uma inicialização da rede neural para cada caso. Sabe-se que o desempenho da rede pode variar em função dos pesos iniciais e do critério de parada utilizado, mas este aspecto não é estudado neste trabalho. Posteriormente é fornecido para a RNAD um conjunto de treinamento, padrão a padrão,  $[X \ T]$  com  $N$  inicialmente pequeno. Esta rede é treinada até se atingir o critério de parada e os pesos representativos deste treino são salvos. Logo após, a RNAI começa seu treinamento com o conjunto  $[O \ X]$ . Seu treinamento é executado até se atingir o critério de parada, que pode ser ou não igual ao critério de parada da RNAD. Seus pesos representativos são salvos e a rede fornece como conjunto de saídas atuais o vetor  $X^\alpha$ . Este fluxo de treinamento pode ser visualizado na Fig. 5.

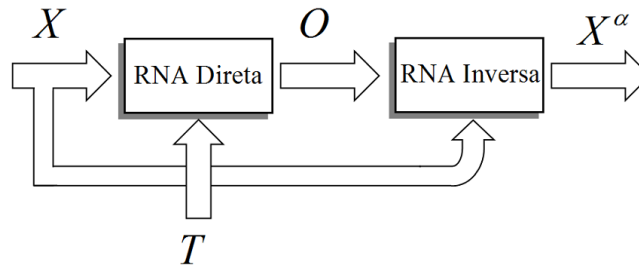


Fig. 5. Fluxo de dados no treinamento e utilização das redes direta e inversa.

Após o treinamento seqüencial das RNAD e RNAI, nesta ordem, o valor absoluto da diferença entre cada componente do vetor  $X$  e  $X^\alpha$  é calculado de acordo com o erro  $e_{inv}$ , Equação (14), e logo em seguida é computada a média desses erros  $E_{inv}$ , Equação (15).

$$e_{inv}(t) = |x(t) - x^\alpha(t)|, \quad \text{com} \quad 1 \leq t \leq N \quad (14)$$

$$E_{inv} = \frac{1}{N} \sum_{t=1}^N e_{inv}(t) \quad (15)$$

Esse último parâmetro será utilizado para separar as saídas confiáveis das não-confiáveis na fase de utilização, onde uma saída confiável é aquela que é menor que o erro médio de treinamento  $E_{inv}$ . O uso desse na fase de utilização processa-se da seguinte maneira: um conjunto de utilização  $X$  é aplicado à RNAD e o fluxo da Fig. 5 é procedido usando-se os pesos armazenados após o treinamento das redes, desconsiderando-se os vetores alvos, para fornecer a saída  $X^\alpha$ . Logo após, o erro  $e_{inv}$  é computado para cada componente dos vetores  $X$  e  $X^\alpha$  de utilização. Esses são classificados de acordo com a desigualdade da Equação (16). Para qualquer quantidade  $N$  de padrões de treinamento, o parâmetro  $E_{inv}$  consegue classificar os valores confiáveis e não-confiáveis.

$$o(t) \text{ será } = \begin{cases} \text{não-confiável,} & \text{se } e_{inv}(t) > E_{inv} \\ \text{confiável,} & \text{se } e_{inv}(t) \leq E_{inv} \end{cases} \quad (16)$$

Para se obter a quantidade de padrões que permitirá à RNAD generalizar, aumenta-se gradativamente a quantidade de padrões  $N$  de treinamento. Efetua-se um novo treinamento seqüencial da RNAD e RNAI e avalia-se o novo valor para o parâmetro  $E_{inv}$ . Esse novo valor tenderá a ser menor que o do treinamento anterior. Este processo é repetido até que  $E_{inv}$  não tenha mais mudanças significativas em seu valor, isto é, se mantém dentro dos valores da definição de computacionalmente iguais. Para esse último treinamento, o valor  $N$  associado a ele é a quantidade de padrões que permitirá à RNAD generalizar a função de interesse satisfatoriamente.

## 5.1 METODOLOGIA DIRETA-INVERSA: DOMÍNIO COMPLEXO E DE CLIFFORD

Esta metodologia é utilizada no domínio Complexo e de Clifford. Nestes domínios, é necessária a utilização de outro parâmetro para aferir a quantidade de padrões de treinamento que garanta a generalização da função de interesse. Este fato deve-se à característica inerente destes domínios: ambos não são totalmente ordenados. Este parâmetro é dado como módulo da diferença dos valores absolutos entre a entrada da RNAD e a saída da RNAI,  $E_{inv_{dm}}$ , como mostra a Equação



(17), lembrando que a definição de módulo para o domínio Complexo e para o domínio de Clifford [24] deve ser respeitada. No caso do domínio de Clifford, o módulo é dado por  $|A| = \sqrt{A \otimes \tilde{A}}$ , onde  $\tilde{A}$  é definido como reverso de um multivetor  $A$  e é obtido invertendo-se a ordem dos fatores do *k-blades* com *grade*  $k \geq 2$ .

$$E_{inv_{dm}} = \frac{1}{N} \sum_{t=1}^N \left| |x(t)| - |x^\alpha(t)| \right| \quad (17)$$

O parâmetro  $E_{inv}$ , Equação (16), permanece sendo o valor que fará a classificação entre os valores confiáveis e os não-confiáveis, na fase de utilização nestes domínios.

## 6. TREINAMENTOS E RESULTADOS

A função quadrática Real, Complexa e de Clifford não tem sua FI garantida por todo domínio, razão pela qual os domínios dessa função são restringidos para a elaboração dos resultados. Os algoritmos BP para o domínio Real [1], [8], Complexo, [7], [18], [19], [20], [21] e de Clifford [11], [14], [15], [22] foram desenvolvidos no ambiente Matlab®, versão 7.1.0.246 (R14), e escritos de forma a não conter nenhuma melhoria de otimização ou adaptações no algoritmo para facilitar a convergência ou qualquer outra mudança que vise a incrementar o desempenho. Esta medida foi tomada para analisar as capacidades da metodologia proposta. A topologia em cada domínio, definida por tentativa e erro [1], sem um estudo detalhado, foi escolhida de maneira que permita a função quadrática convergir.

A Tabela 3 descreve os parâmetros de treinamento das redes nos três domínios de atuação.

	Domínio		
	Real	Complexo	Clifford
Entradas RNAD	entre [1, 5]	$r \cdot \exp(\theta i)$ :	$r \cdot \exp(\theta \mathbf{e}_1 \wedge \mathbf{e}_2)$ :
		com $\theta$ entre $[5^\circ, 40^\circ]$ e $r = 1$	
Taxa de aprendizado ( $\eta$ )	$10^{-1}$		
SSE desejado	$10^{-4}$	$7 * 10^{-5}$	
Topologia RNAD e RNAI	1-4-4-1	1-5-1	1-3-1
Algoritmo de treinamento	Backpropagation		
Modo de treinamento	Padrão a Padrão		

Tabela 3. Parâmetros de treinamento RNAD e RNAI.

Para a avaliação da capacidade de classificação das saídas confiáveis e das não-confiáveis, na fase de utilização foram gerados 500 padrões de utilização que foram submetidos a cada uma das redes treinadas, as quais foram treinadas com diferentes quantidade de padrões, todos esses valores pertencentes aos intervalos de treinamento especificados para cada uma das redes.

### 6.1 RESULTADOS NO DOMÍNIO REAL

Os conjuntos de treinamento fornecidos às redes reais são valores igualmente espaçados entre [1,5]. Os dados de treinamento não foram gerados aleatoriamente, para ter controle sobre a sua localização. O treinamento inicial com dois padrões mostra que a metodologia consegue classificar as saídas de forma satisfatória, como pode ser observado na Fig. 6, onde fica claro que os

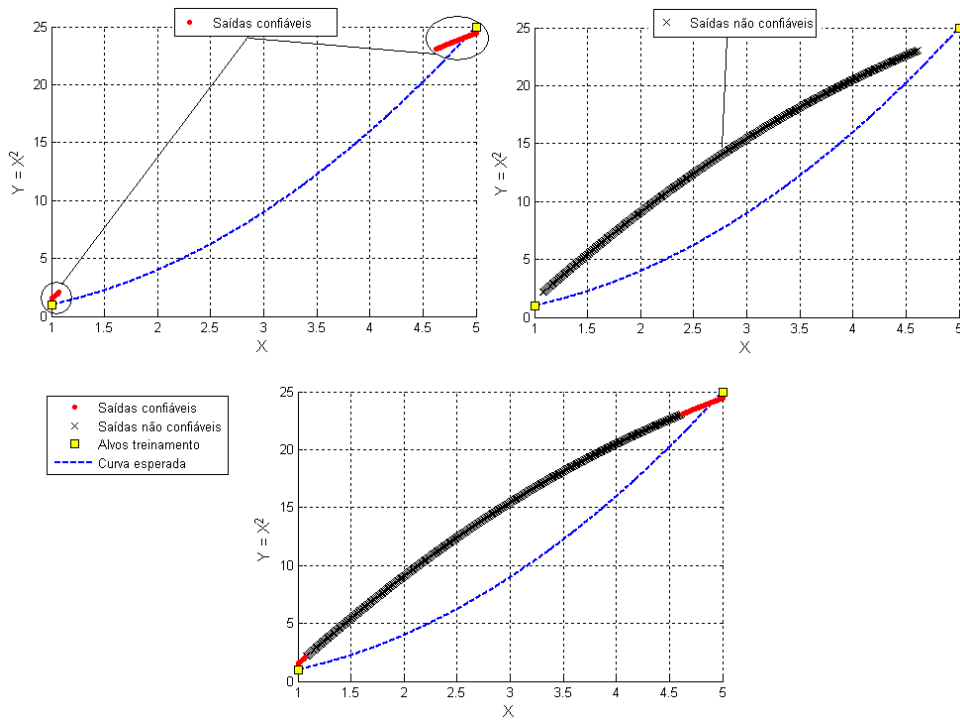


Fig. 6. Saídas da RNAD treinada com 2 padrões: domínio Real.

valores confiáveis estão próximos dos padrões de treinamento (na figura superior esquerda), como era esperado, já que dois padrões de treinamento são insuficientes para representar a função quadrática no intervalo especificado. Os valores não confiáveis estão mostrados na figura superior direita e todos os resultados juntos na figura inferior. Com o aumento progressivo da quantidade de padrões de treinamento, verificou-se também o aumento da quantidade de valores classificados como confiáveis (na figura superior esquerda), como pode ser notado na Fig. 7. Os valores não confiáveis estão mostrados na figura superior direita e todos os resultados juntos na figura inferior. É notório também o comportamento do erro *Ein*v, que tem seu valor reduzido conforme o aumento de padrões de treinamento.

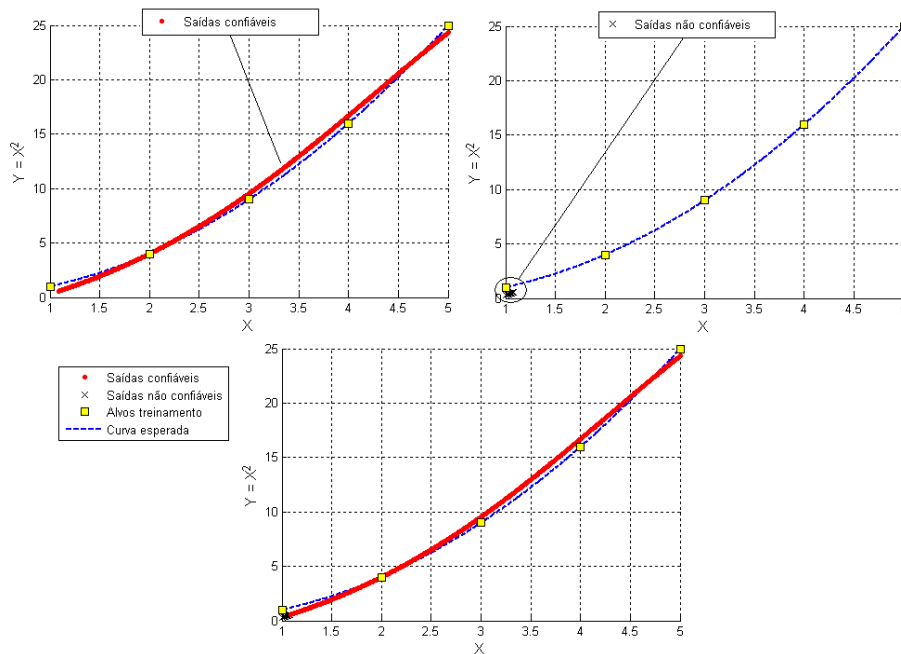


Fig. 7. Saídas da RNAD treinada com 5 padrões: domínio Real.

Observando-se a Tabela 4, podemos verificar que, a partir de cinco padrões de treinamento, o parâmetro *Einv* já não sofre alterações significativas. Conclui-se, de acordo com a metodologia proposta, que essa é a quantidade suficiente para generalizar esta função no intervalo admitido.

padrões	<i>Einv</i>	épocas de treino	
		RNAD	RNAI
2	0,0119	457	155
3	0,0108	2688	222
4	0,0108	3028	533
5	0,0093	2406	104
6	0,0099	2107	58
7	0,0098	1437	214
8	0,0093	2350	261

Tabela 4. Comportamento do erro *Einv*: domínio Real.

No domínio Real, o parâmetro *Einv* proposto consegue classificar a exatidão das saídas da RNA e ao mesmo tempo quantificar o tamanho do conjunto de treinamento necessário para generalizar a função.

Verificou-se necessário usar duas camadas ocultas, visto que o aumento gradativo de padrões aumenta demasiadamente o número de épocas de treinamento, quando utilizada apenas uma camada oculta. Com duas camadas ocultas, as redes convergem para quaisquer quantidades de padrões de treinamento.

## 6.2 RESULTADOS NO DOMÍNIO COMPLEXO

Pelo fato de o domínio Complexo não ser um conjunto totalmente ordenado, como os números reais sobre a reta Real, foi utilizada a representação de um número Complexo na forma exponencial  $z = r \cdot \exp(\theta \cdot i)$  para construir os conjuntos de treinamento e de utilização da função quadrática complexa. Esta forma permite gerar números Complexos igualmente espaçados pela fase  $\theta$  sobre um determinado raio  $r$  de interesse. Tal como a função quadrática real, a função quadrática complexa não tem sua FI garantida por todo o seu domínio. Assim, optamos por uma faixa de valores para a fase no raio unitário, onde a função quadrática tenha sua função inversa garantida.

Os parâmetros de treinamento para a RNAD e RNAI de domínio Complexo são mostrados na Tabela 3, na Seção 6. O comportamento da metodologia neste domínio vai ao encontro dos resultados no domínio Real. Com algumas ressalvas, discutidas na Seção 7, neste domínio *Einv* é o parâmetro que afere a quantidade de padrões que permite a generalização da função. Como no domínio Real, o erro *Einv* consegue classificar satisfatoriamente as saídas confiáveis das não-confiáveis, na fase de utilização da RNAD, como se pode observar nas Figs. 8 e 9, as quais também mostram que os valores confiáveis se encontram próximos dos valores de treinamento, como esperado. Nas figuras os valores confiáveis estão mostrados na figura a esquerda, os valores não confiáveis estão mostrados na figura central e todos os resultados juntos na figura direita.

Neste domínio também se verifica que o aumento de padrões de treinamento melhora a exatidão dos valores de saída da RNA complexa. Observando-se a Tabela 5, podemos notar que seis padrões de treinamento são suficientes para garantir a generalização da rede direta.

padrões	<i>Einv</i>	<i>Einv<sub>dm</sub></i>	épocas de treino	
			RNAD	RNAI
2	0,00946	0,00770	117	445
3	0,01042	0,00517	1485	1559
4	0,01002	0,00495	3114	215
5	0,01010	0,00538	2134	545
6	0,01141	0,00412	701	771
7	0,01128	0,00423	923	703
8	0,01119	0,00431	959	562

Tabela 5. Comportamento dos Erros *Einv* e *Einv<sub>dm</sub>*: domínio Complexo.

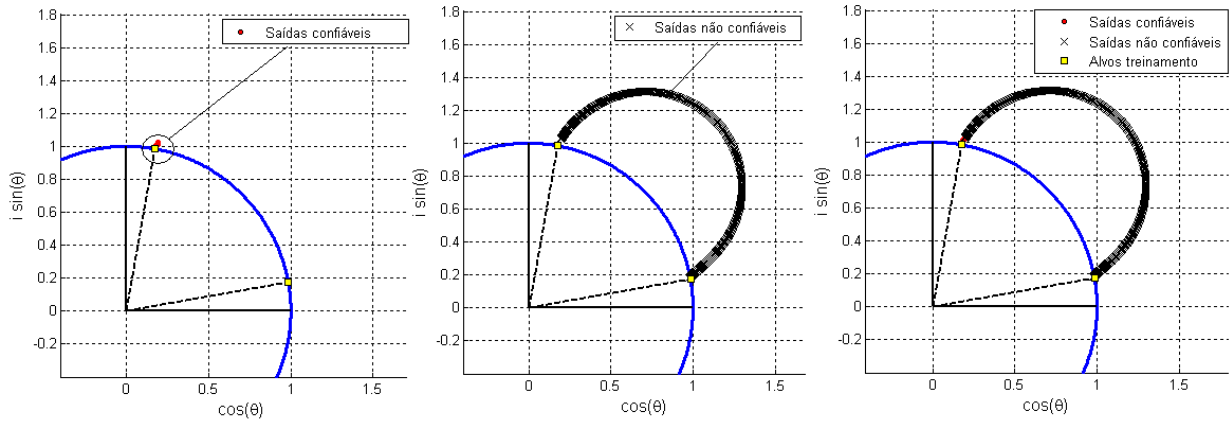


Fig. 8. Saídas da RNAD treinada com 2 padrões: domínio Complexo.

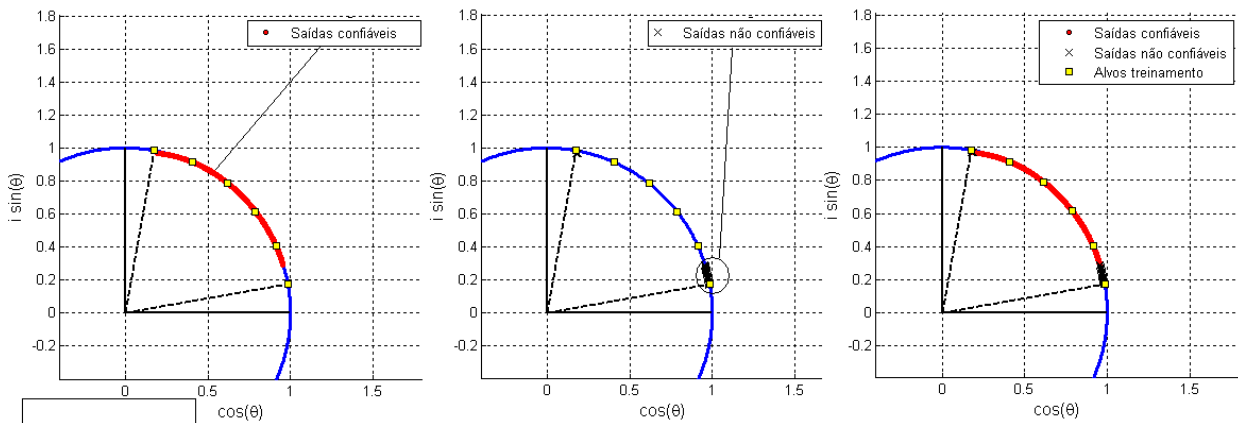


Fig. 9. Saídas da RNAD treinada com 6 padrões: domínio Complexo.

### 6.3 RESULTADOS NO DOMÍNIO DE CLIFFORD

Conforme discutido na Seção 3, as álgebras de Clifford podem ter, dependendo da sua assinatura, a álgebra dos números Complexos como subespaço. A fim de facilitar a ponderação dos resultados e estabelecer parâmetros observáveis, foram fornecidos conjuntos de treinamento e utilização para as RNA de Clifford de assinatura  $Cl_{0,3}$ , a qual contém o subespaço gerado pelos elementos de base  $\{1, e_1 \wedge e_2\}$ , isomorfo à álgebra dos números Complexos.

Na Tabela 3, na Seção 6, é possível observar que os parâmetros de treinamentos das redes complexas e de Clifford são semelhantes diferenciando-se apenas na topologia. Esta semelhança é proposital para a comparação e avaliação da metodologia nos dois domínios. No domínio de Clifford, a metodologia também conseguiu classificar as saídas confiáveis das não-confiáveis por meio do parâmetro  $Einv$ , resultado que pode ser observado nas Figs. 10 e 11. Nas figuras os valores confiáveis estão mostrados na figura a esquerda, os valores não confiáveis estão mostrados na figura central e todos os resultados juntos na figura direita.

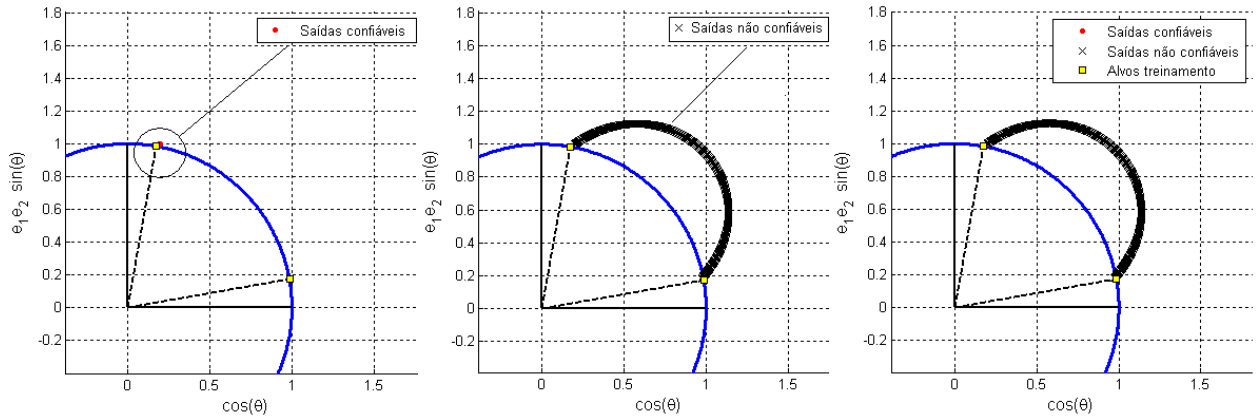


Fig. 10. Saídas da RNAD treinada com 2 padrões: domínio de Clifford.

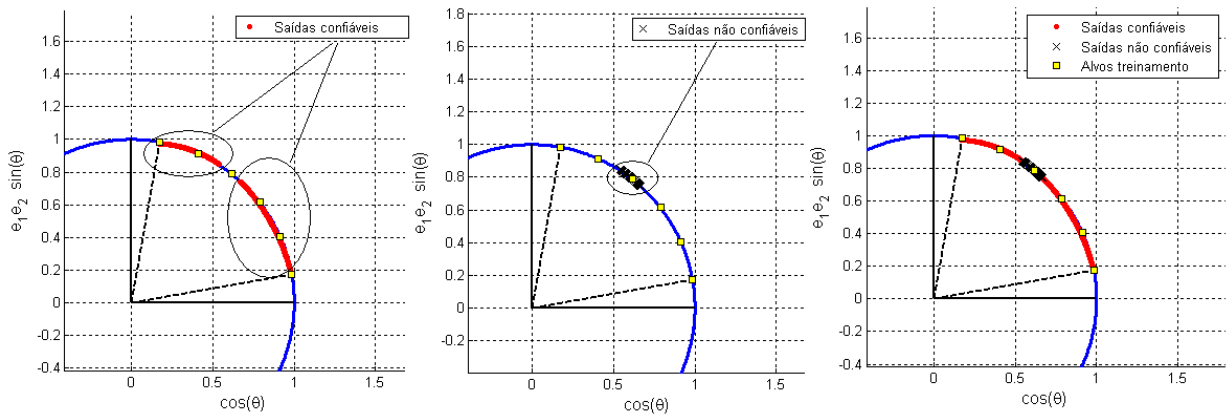


Fig. 11. Saídas da RNAD treinada com 6 padrões: domínio de Clifford.

A metodologia também conseguiu aferir a quantidade de padrões de treinamento para permitir a generalização da RNAD, utilizando-se o parâmetro  $Einv_{dm}$ , o qual mostrou que seis padrões são suficientes para tal conclusão, como se evidencia na Tabela 6.

padrões	$Einv$	$Einv_{dm}$	épocas de treino	
			RNAD	RNAI
2	0,00984	0,00610	99	293
3	0,01132	0,00776	1677	1033
4	0,01158	0,00723	2527	2136
5	0,01097	0,00496	3418	549
6	0,00955	0,00470	3533	960
7	0,01100	0,00463	2430	1028
8	0,01056	0,00468	9052	2864

Tabela 6. Comportamento dos erros  $Einv$  e  $Einv_{dm}$ : domínio de Clifford.

Esses resultados eram esperados, já que este domínio de atuação é isomorfo ao domínio Complexo. Verificou-se também que uma menor quantidade de neurônios na camada oculta, ver Tabela 3, consegue guardar informação sobre o comportamento da função quadrática complexa, resultado esse que vai ao encontro do trabalho em [13], no qual se verificou que o modelo de neurônio artificial de Clifford tem uma capacidade maior de armazenamento de informação em seus pesos sinápticos, em comparação aos neurônios definidos nos domínios Real e Complexo. Porém, o aumento do custo computacional é evidente na comparação das épocas de treino, Tabelas 5 e 6, resultados esses já confirmados pelo trabalho em [16]. Neste domínio, a metodologia também demonstra que o aumento de padrões de treinamento melhora a

representação da função e conseqüentemente a exatidão nos dados de saída. Pode-se, também, notar que o aumento indiscriminado de padrões não iria melhorar substancialmente a representação da função nem o grau de exatidão nos dados de saída, e esta conclusão é constante nos três domínios de atuação.

## 7. CONCLUSÕES

Este trabalho propôs, validou e demonstrou a **metodologia Direta-Inversa** como uma possível solução para: **1º)** estimar a confiabilidade das saídas das redes na fase de utilização; e **2º)** estabelecer o número mínimo de padrões a serem utilizados na fase de treinamento, para garantir a convergência e a generalização das redes. A metodologia foi desenvolvida para a abordagem de problemas que possuam funções inversas ou que tenham inversa definida em domínios restritos, como foi demonstrado na aproximação da função quadrática  $f(x) = x^2$  para os domínios Real, Complexo e de Clifford. Os resultados mostram a eficiência da metodologia nesses três domínios na separação dos resultados confiáveis dos não-confiáveis. O modelo de neurônio artificial de Clifford tem uma capacidade maior de armazenamento de informação em seus pesos sinápticos, em comparação aos neurônios definidos no domínio Complexo, que, por sua vez, têm uma capacidade maior de armazenamento de informação em seus pesos sinápticos, em comparação aos neurônios definidos no domínio Real. Os resultados experimentais nos três domínios mostram que o aumento indiscriminado de padrões não iria melhorar substancialmente a representação da função nem a confiabilidade, neste trabalho definida como a exatidão, nos dados de saída. Deixa-se como sugestão para trabalhos futuros a aplicação da metodologia para problemas práticos em Engenharia, bem como nas áreas onde as RNA e as álgebras de Clifford são utilizadas com frequência, tais como: visão computacional, robótica, controle, etc.

## 9. REFERÊNCIAS

- [1] Haykin S., “Neural Networks. A Comprehensive Foundation”, *Macmillan College Publishing, New York, 1994*.
- [2] Yamasaki, K., Ogawa, H., “Methods for choosing a training set which prevents over-learning”, *Neural Networks, IEEE World Congress on Computational Intelligence, vol.1 pp. 551 – 554, 1994*.
- [3] Kohavi R., “A study of cross-validation and bootstrap for accuracy estimation and model selection”, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, vol.2, no.12, pp.1137-1143, 1995*.
- [4] K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop, “Neural networks for control systems--A survey”, *Automatica, Volume 28, Issue 6, November 1992, Pages 1083-1112, ISSN 0005-1098, DOI: 10.1016/0005-1098(92)90053-I*.
- [5] Hunt, K.J.; Sbarbaro, D.; , “Neural networks for nonlinear internal model control”, *Control Theory and Applications, IEE Proceedings D , vol.138, no.5, pp.431-438, Sep 1991*.
- [6] Minsky, M., Papert, S., “Perceptrons”, *Cambridge, Mass: MIT Press, 1969*.
- [7] Kim, T., “Complex Backpropagation Neural Network Using Elementary Transcendental Activation Functions”, *IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.1281-1284, 2001*.
- [8] Lawrence J., “Introduction to Neural Networks and Expert Systems”, *Nevada City, CA: California Scientific Software, 1992*.
- [9] Cybenko G., “Continuous Valued Neural Networks with Two Hidden Layers Are Sufficient”, *Technical Report, Department of Computer Science, Tufts University, 1988*.
- [10] Hestenes D., “Reforming the mathematical language of physics”, *American Journal of Physics, vol.71, no.2, pp.104-121, 2003*.
- [11] Qing Yi., “Learning rules for low-dimensional Clifford neural networks”. *Master thesis, Portland State University, 2004*.
- [12] Hestenes D. and Sobczyk G., “Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics”, *Springer, 1984*.
- [13] Sommer G., “Geometric computing with Clifford algebras”, publisher: *Spinger-Verlag, London, UK, 2001*.

- [14] Dorst, L., Mann, S., “Geometric Algebra: A Computational Framework for Geometrical Applications (Part 1)”, *IEEE Computer Graphics and Applications*, vol.22, no.3, 2002.
- [15] Mann, S., Dorst, L., Bouma, T. , “The making of a geometric algebra package in Matlab”, *Relatório Técnico CS-99-27, University of Waterloo, Canada, 26 pages, 1999.*
- [16] Buchholz S. and Sommer G., “On Clifford neurons and Clifford multilayer perceptrons”, *Neural Networks Journal*, vol.21, no.7, pp.925-935, 2008.
- [17] Arena P., Fortuna L., Re R. and Xibilia M. G., “Multilayer perceptrons to approximate quaternion valued functions”. *Neural Networks*, v. 10, n. 2, p. 335–342, 1997.
- [18] Arena P., Fortuna L., Re R. and Xibilia M. G., “On the capability of neural networks with complex neurons in complex valued functions approximation”, *International Symposium on Circuits and Systems*, PP.2168-2171,1993.
- [19] N. Benvenuto and F. Piazza, “On the Complex Backpropagation Algorithm”, *IEEE Transactions on Signal Processing*, vol.40, no.4, pp.967-969, 1992.
- [20] Henry Leung and Simon Haykin, “The Complex Backpropagation Algorithm”, *IEEE Transactions on Signal Processing*, vol.39, no.9, pp.2101-2104, 1991.
- [21] G. M. Georgiou and Koutsougeras C, “Complex domain backpropagation”, *IEEE Transactions on Circuits and Systems. II: Analog and Digital Signal Processing*, vol.39, no.5, pp.330-334, 1992.
- [22] Pearson J. and Holloway R. and Bisset D. L., “Computation with Clifford valued Feed-Forward Networks”, *Oxford University Press, USA, 1999.*
- [23] Needham T., “Visual Complex Analysis”, *Oxford University Press, USA, 1999.*
- [24] Dorst, L., Fontijne, D.; Mann, S., “Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry (The Morgan Kaufmann Series in Computer Graphics)”, *Morgan Kaufmann Publishers Inc., San Francisco, 2007.*