

UMA ABORDAGEM INTERVALAR PARA OS MAPAS AUTO-ORGANIZÁVEIS DE KOHONEN

FABIANA T. SANTANA*, ADRIÃO DUARTE DÓRIA NETO*, REGIVAN HUGO NUNES SANTIAGO†

* *Universidade Federal do Rio Grande do Norte*
 Departamento de Engenharia de Computação e Automação
 Natal, RN, Brasil

† *Universidade Federal do Rio Grande do Norte*
 Departamento de Informática e Matemática Aplicada
 Natal, RN, Brasil

Emails: fabianatsantana@dca.ufrn.br, adriao@dca.ufrn.br, regivan@dimap.ufrn.br

Resumo— A Matemática Intervalar vem ganhando espaço em muitas aplicações, por exemplo, em Redes Neurais e Processamento Digital de Imagens podemos pensar em representar dados numéricos e até mesmo matrizes de pixel com valores intervalares e assim preservar possíveis incertezas. Com isso, surge a necessidade de uma fundamentação matemática e ferramentas adequadas, como algoritmos que lidam com dados dessa natureza. Neste trabalho apresentamos as principais definições da Aritmética Intervalar utilizadas para se trabalhar com Redes Neurais Intervalares. Definimos a distância para elementos provenientes de produtos cartesianos entre intervalos, utilizando a distância de Moore, além de mostrar que a propriedade de métrica é preservada. Considerando os estudos feitos à respeito dos Mapas Auto-Organizáveis de Kohonen, fizemos uma extensão para o caso intervalar propondo uma nova abordagem para o algoritmo Self-Organizing Map. Para concluir, aplicamos o algoritmo a um exemplo numérico, exibimos o processo algébrico na primeira iteração da rede e observamos o deslocamento dos neurônios para formar os aglomerados do mapa de características.

Palavras-chave— Aritmética Intervalar, Redes Neurais Intervalares, Algoritmo SOM Intervalar, Mapas Auto-Organizáveis.

1 Aritmética Intervalar de Moore

Em muitos problemas a representação de dados matemáticos se faz necessária e muitas vezes esta representação vem acompanhada de uma taxa de erro que pode ser originado por limitação de máquina, por truncamento ou por arredondamentos. A aritmética intervalar surgiu, proposta por Moore na década de 60, com o objetivo de desenvolver mecanismos de controle de erros e tornar os resultados de problemas práticos mais confiáveis e com maior precisão. Esse estudo vem ganhando espaço em muitas aplicações, como por exemplo, em Redes Neurais e Processamento Digital de Imagens. Com isso, surge a necessidade de uma fundamentação matemática adequada e algoritmos que lidam com dados intervalares.

Nesta seção apresentaremos as operações fundamentais e algumas propriedades da aritmética intervalar proposta por Moore que serão utilizadas em Redes Neurais Intervalares.

Definição 1.1. (*Conjunto dos Intervalos de Números Reais*) Definimos por \mathbb{IR} o conjunto dos intervalos de números reais, isto é, $\mathbb{IR} = \{[x, \bar{x}] \mid x, \bar{x} \in \mathbb{R} \text{ e } x \leq \bar{x}\}$

Definição 1.2. (*Operações Aritméticas em \mathbb{IR}*) Dados os intervalos reais $X = [x, \bar{x}]$ e $Y = [y, \bar{y}]$ as operações elementares são definidas por:

- (a) $X + Y = [x + y; \bar{x} + \bar{y}]$;
- (b) $X - Y = [x - \bar{y}; \bar{x} - y]$;
- (c) $X \times Y = [\min\{x.y, \underline{x}.\underline{y}, \bar{x}.\bar{y}, \bar{x}.\underline{y}\};$

$$\max\{x.y, \underline{x}.\underline{y}, \bar{x}.\bar{y}, \bar{x}.\underline{y}\}];$$

$$(d) \frac{X}{Y} = X \times Y^{-1} = [\min\{\frac{x}{\underline{y}}, \frac{x}{\bar{y}}, \frac{\bar{x}}{\underline{y}}, \frac{\bar{x}}{\bar{y}}\}, \max\{\frac{x}{\underline{y}}, \frac{x}{\bar{y}}, \frac{\bar{x}}{\underline{y}}, \frac{\bar{x}}{\bar{y}}\}], \text{ onde } 0 \notin \mathbb{R}.$$

Definição 1.3. (*Distância de Moore*) Dados dois intervalos reais $X = [x, \bar{x}]$ e $Y = [y, \bar{y}]$, definimos a distância intervalar de X a Y pela maior distância em módulo entre os extremos, isto é

$$d_M(X, Y) = \max\{|x - y|, |\bar{x} - \bar{y}|\}.$$

Definição 1.4. (*Módulo de um Intervalo*) Dado um intervalo real $X = [x, \bar{x}]$ o módulo de X é definido pela maior distância em módulo de X até $\mathbb{O} = [0, 0]$, isto é $|X| = \text{dist}(X, \mathbb{O}) = \max\{|x|, |\bar{x}|\} \geq 0$.

Corolário 1.1. *Dados dois intervalos reais $X = [x, \bar{x}]$ e $Y = [y, \bar{y}]$ temos que $X = Y$ se, e somente se, $\text{dist}(X, Y) = 0$.*

Observação 1.1. *A função distância $d_M(X, Y) = \max\{|x - y|, |\bar{x} - \bar{y}|\}$ define uma métrica em \mathbb{IR} por satisfazer as seguintes propriedades:*

- (a) $d_M(X, Y) = 0 \Leftrightarrow X = Y$;
- (b) $d_M(X, Y) = \text{dist}(Y, X), \forall X, Y \in \mathbb{IR}$;
- (c) $d_M(X, Z) \leq \text{dist}(X, Y) + \text{dist}(Y, Z), \forall X, Y, Z \in \mathbb{IR}$.

Teorema 1.1. *O Conjunto \mathbb{IR} munido da função distância $d_M(X, Y)$ é um espaço métrico completo.*

Definição 1.5. (*Produto Cartesiano de Espaços Métricos*) Sejam M e N espaços métricos, cuja métrica indicaremos com o símbolo d . O produto cartesiano $M \times N$ é, como conjunto, formado pelos pares ordenados $z = (x, y)$, onde $x \in M$ e $y \in N$.

Podemos dotar o produto $M \times N$ de uma métrica, definindo a distância de $z = (x, y)$ a $z' = (x', y')$ como sendo:

1. $d'(z, z') = d(x, x') + d(y, y')$;
2. $d''(z, z') = \max\{d(x, x'), d(y, y')\}$;
3. $d(z, z') = \sqrt{(d(x, x'))^2 + (d(y, y'))^2}$

A generalização para o produto de n fatores é imediata [1]. Dados os espaços métricos M_1, M_2, \dots, M_n , cujas métricas indicaremos com o mesmo símbolo d , o produto cartesiano $M = M_1 \times M_2 \times \dots \times M_n$ é o conjunto das listas $x = (x_1, x_2, \dots, x_n)$, onde $x_1 \in M_1, \dots, x_n \in M_n$.

Como vimos acima, $\mathbb{I}\mathbb{R}$ munido da função distância $d_M(X, Y)$ é um espaço métrico completo, então podemos definir a distância entre elementos proveniente de produtos cartesianos entre $\mathbb{I}\mathbb{R}$ ou subconjuntos não vazios de $\mathbb{I}\mathbb{R}$.

Definição 1.6. (*Distância entre Elementos Intervalares de Produtos Cartesianos*) Para $X = (X_1, X_2), Y = (Y_1, Y_2) \in \mathbb{I}\mathbb{R} \times \mathbb{I}\mathbb{R}$ definimos a distância de X a Y por

$$d(X, Y) = d_M(X_1, Y_1) + d_M(X_2, Y_2)$$

Proposição 1.1. Para $X = (X_1, X_2), Y = (Y_1, Y_2) \in \mathbb{I}\mathbb{R} \times \mathbb{I}\mathbb{R}$, a função distância $d(X, Y) = d_M(X_1, Y_1) + d_M(X_2, Y_2)$ define uma métrica em $\mathbb{I}\mathbb{R}$.

Demonstração: Para $X = (X_1, X_2), Y = (Y_1, Y_2), Z = (Z_1, Z_2) \in \mathbb{I}\mathbb{R} \times \mathbb{I}\mathbb{R}$, temos:

1. $d(X, Y) = 0 \Leftrightarrow d_M(X_1, Y_1) + d_M(X_2, Y_2) = 0$. Como $d_M(X_1, Y_1), d_M(X_2, Y_2) \geq 0$, temos $d(X, Y) = 0 \Leftrightarrow d_M(X_1, Y_1) = 0$ e $d_M(X_2, Y_2) = 0$. Como d_M é uma métrica, temos $d(X, Y) = 0 \Leftrightarrow X_1 = Y_1$ e $X_2 = Y_2$, isto é, $X = Y$.
2. $d(X, Y) = d_M(X_1, Y_1) + d_M(X_2, Y_2) = d_M(Y_1, X_1) + d_M(Y_2, X_2)$, pois d_M é uma métrica. Logo, $d(X, Y) = d(Y, X)$.
3. $d(X, Z) = d_M(X_1, Z_1) + d_M(X_2, Z_2) \leq (d_M(X_1, Y_1) + d_M(Y_1, Z_1)) + (d_M(X_2, Y_2) + d_M(Y_2, Z_2))$. Pois, d_M é uma métrica. Assim, $d(X, Z) \leq (d_M(X_1, Y_1) + d_M(X_2, Y_2)) + (d_M(Y_1, Z_1) + d_M(Y_2, Z_2)) = d(X, Y) + d(Y, Z)$.

Portanto, d é uma métrica. ■

A distância definida acima é uma métrica em relação a distância de Moore. Sempre que formos calcular a distância entre pares com coordenadas

intervalares usaremos a distância d , que pode ser facilmente estendida para o produto cartesiano de n fatores.

2 Redes Neurais Intervalares

Definição 2.1. (*Redes Neurais Intervalares*) Uma rede neural será chamada de rede neural intervalar se algum dos componentes, conjunto de entrada, saída e pesos, forem valores intervalares.

Definição 2.2. (*Neurônio Intervalar*) O neurônio intervalar é a unidade de processamento das redes neurais artificiais intervalares que recebe entradas intervalares (ou reais) e as processa obtendo uma saída.

Denotaremos os sinais de entrada por X_i e os pesos sinápticos, que fazem a conexão da entrada i ao neurônio j , por W_{ij} . Consideraremos que as entradas e os pesos sinápticos são intervalares.

O processamento dos dados de entrada em um neurônio artificial intervalar é feito através das três seguintes funções [2]: *Função Normalizadora*, que analisa a natureza da entrada normalizando os dados para intervalos. *Função de Ativação*, que realiza a soma ponderada dos sinais de entrada pelos seus respectivos pesos sinápticos intervalares. *Função de Transferência*, que determina a saída do neurônio.

2.1 Uma Proposta Intervalar para os Mapas Auto-Organizáveis de Kohonen

Como acontece nas redes neurais supervisionadas, as redes auto-supervisionadas também passam por várias etapas para ter um resultado satisfatório e com isso os dados numéricos passam a conservar algum tipo de erro resultante das limitações dos equipamentos, arredondamentos e truncamentos. Vários problemas complexos tratáveis pelo algoritmo de Kohonen podem ser representados por dados intervalares. Para que isso seja possível é necessário um algoritmo apropriado para lidar com os dados intervalares.

Em um mapa auto-supervisionado cada neurônio se conecta com uma camada computacional [3]. Inicialmente os neurônios ficam organizados em linhas e colunas. Com a apresentação dos padrões de entrada os neurônios da grade tendem a se reagrupar formando aglomerados que irão refletir as principais características dos padrões de entrada, de acordo com os seguintes passos:

1. *Competição:* Para cada padrão de entrada, os neurônios da grade calculam seus respectivos valores de uma função discriminante que fornece a base para a competição entre os neurô-

nios. O neurônio com o maior valor da função discriminante é declarado vencedor da competição.

2. *Cooperação*: O neurônio vencedor determina uma vizinhança topológica de neurônios excitados, isto é, os neurônios nesta vizinhança cooperam de alguma maneira com o processo de aprendizagem.
3. *Adaptação Sináptica*: Este mecanismo faz com que os neurônios excitados aumentem seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes adequados aplicados a seus pesos sinápticos. Os ajustes feitos possibilitam que a resposta do neurônio vencedor à aplicação subsequente de um padrão de entrada similar é melhorada.

2.2 Algoritmo SOM Intervalar

Apresentaremos nesta seção uma proposta intervalar para o algoritmo SOM (*Self-Organizing Map*) [4], usado para o agrupamento das características no mapa de Kohonen. Para isso usaremos as propriedades aritméticas de soma, subtração e multiplicação intervalar, segundo a aritmética de Moore [5] e algumas funções intervalares. No algoritmo proposto por Kohonen, [4], é usada a distância euclidiana para números reais. Nesta nova abordagem, onde consideramos os exemplos de treinamento e os pesos sinápticos intervalares, necessitamos de uma distância apropriada e usaremos a distância dada na definição (1.6).

Processo Competitivo

Seja m a dimensão do espaço de entrada cujos vetores são constituídos por coordenadas intervalares. Selecione aleatoriamente um padrão dentre os demais exemplos de treinamento. O exemplo selecionado será representado por

$$X = [\underline{x}_0, \overline{x}_0], \dots, [\underline{x}_m, \overline{x}_m]^T \quad (1)$$

onde $i = 1, \dots, m$ representa o índice das coordenadas do vetor de entrada X . Suponha que tenhamos l neurônios organizados na grade e representaremos cada um deles pelo índice j , $j = 1, \dots, l$. O vetor de pesos sinápticos dos neurônios da grade tem a mesma dimensão do espaço de entradas e será representado por

$$W_j = [W_{j1}, \dots, W_{jm}]^T \quad (2)$$

onde W_{ji} é o peso sináptico que conecta o neurônio j a entrada $X_i = [\underline{x}_i, \overline{x}_i]$. O vetor de entrada selecionado é apresentado à rede sem que se especifique a saída desejada e algum neurônio deve se relacionar melhor com esta entrada, ficando conhecido como neurônio vencedor. O critério usado para se determinar o neurônio vencedor será através da menor

distância entre o vetor de entrada X e o vetor peso sináptico W_j de cada neurônio. Para isso usaremos a distância entre vetores dada na definição (1.6).

$$d_j(t) = d(X, W_j) = \sum_{i=1}^m d_M(X_i, W_{ji}) \quad (3)$$

onde $X = [X_1, \dots, X_m]^T$ é o vetor de entrada e $W_j = [W_{j1}, \dots, W_{jm}]^T$ é o vetor peso que liga o neurônio j à entrada i . Sem perda de generalidade vamos supor que o neurônio vencedor seja aquele de índice k . Neste passo, um espaço contínuo de entrada de padrões de ativação é mapeado para um espaço discreto de saída de neurônios por um processo de competição entre os neurônios da grade [3].

Processo Cooperativo

O neurônio vencedor localiza o centro de uma vizinhança topológica de neurônios cooperativos. Os estudos feitos para redes neurais de entradas reais mostraram que a função gaussiana consegue atender às exigências a respeito da vizinhança topológica, estando de acordo com as evidências neurobiológicas. Para definir a vizinhança topológica é necessário definir a distância lateral entre o neurônio vencedor e os demais neurônios da vizinhança. Definimos a distância lateral entre o neurônio vencedor k e o neurônio j pela distância entre vetores intervalares dada na definição (1.6), entre os vetores pesos sinápticos associados com o neurônio k e o neurônio j , isto é:

$$d_{j,k} = d(W_j, W_k) = \sum_{i=1}^l d_M(W_{ji}, W_{ki}) \quad (4)$$

que fornecerá um número real. Assim, a vizinhança topológica será dada por:

$$h_{j,k}(X) = \exp\left(-\frac{(d_{j,k})}{2\sigma^2}\right) \quad (5)$$

A definição acima depende da localização do neurônio vencedor e resultará em um número real. O parâmetro σ é a largura da vizinhança topológica e mede o grau com que os neurônios excitados na vizinhança topológica participam do processo de aprendizagem. A largura efetiva da vizinhança topológica deve diminuir com o tempo tornando a vizinhança mais restrita e mais especializada. Como no caso real, para representar a dependência de σ com o tempo discreto t escolhemos o decaimento exponencial. Logo, temos

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_1}\right) \quad (6)$$

onde σ_0 é o valor inicial de σ , t é o número de iterações e τ_1 é uma constante de tempo. Desta forma, a vizinhança topológica torna-se variável com o tempo, isto é

$$h_{j,k}(t) = \exp\left(-\frac{(d_{j,k})(t)}{2\sigma^2(t)}\right) \quad (7)$$

O Processo Adaptativo

Da mesma forma que as demais redes neurais, o aprendizado de um mapa auto-organizável se dá pelo ajuste de seus pesos sinápticos [3]. Para que a grade seja auto-organizável é necessário que o vetor de pesos sinápticos W_j se modifiquem ao se apresentar novas entradas X . Considerando-se o peso sináptico W_{ji} , que conecta o neurônio j à entrada i , o ajuste intervalar proposto é dado por

$$W_{ji}(t+1) = W_{ji}(t) + [\eta(t)h_{j,k}(t), \eta(t)h_{j,k}(t)](X_i - W_{ji}) \quad (8)$$

onde definimos o intervalo degenerado $[\eta(t)h_{j,k}(t), \eta(t)h_{j,k}(t)]$ para garantir a multiplicação pelo intervalo resultante da operação $(X_i - W_{ji})$, uma vez que não temos definida a multiplicação de escalares por intervalos. O parâmetro η é a taxa de aprendizagem definida pela seguinte expressão

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_1}\right) \quad (9)$$

onde τ_1 é uma constante de tempo e $0 < \eta_0 < 1$ é o valor inicial adotado. Observe que a taxa de aprendizagem decresce com o tempo. Isto evita que ao se apresentar dados novos à rede, após longo período de treinamento, o conhecimento já adquirido não seja comprometido.

2.3 Aplicação do Algoritmo SOM Intervalar

Apresentamos um exemplo de uma rede de Kohonen bidimensional com dados intervalares e mostraremos matematicamente a primeira iteração do processo de treinamento. Considere uma rede com três nós de entrada e seis neurônios dispostos em uma grade de duas linhas e três colunas. Iniciamos o treinamento atribuindo valores aleatórios aos pesos sinápticos W_{ji} , onde j representa o neurônio e i é o índice da coordenada do vetor de entrada. Suponha que na primeira iteração temos os seguintes pesos sinápticos:

- Para o neurônio $j = 1$: $W_{11} = [1.8, 2.2]$; $W_{12} = [1.8, 2.2]$; $W_{13} = [0.8, 1.2]$.
- Para o neurônio $j = 2$: $W_{21} = [2.8, 3.2]$; $W_{22} = [1.8, 2.2]$; $W_{23} = [1.8, 2.2]$.
- Para o neurônio $j = 3$: $W_{31} = [1.8, 2.2]$; $W_{32} = [0.8, 1.2]$; $W_{33} = [0.8, 1.2]$.
- Para o neurônio $j = 4$: $W_{41} = [0.8, 1.2]$; $W_{42} = [1.8, 2.2]$; $W_{43} = [2.8, 3.2]$.
- Para o neurônio $j = 5$: $W_{51} = [1.8, 2.2]$; $W_{52} = [0, 0]$; $W_{53} = [0.8, 1.2]$.

- Para o neurônio $j = 6$: $W_{61} = [0.8, 1.2]$; $W_{62} = [1.8, 2.2]$; $W_{63} = [0, 0]$.

Suponha que o primeiro exemplo de treinamento apresentado à rede tenha as seguintes coordenadas: $X_1 = [0, 0]$, $X_2 = [2.8, 3.2]$ e $X_3 = [4.8, 5.2]$.

Devemos ver qual é o neurônio que mais se identifica com o exemplo apresentado à rede. Utilizando a distância de Moore, definida por

$$d_M(X_i, W_{ji}) = \max\left\{\left| \underline{X_i} - \underline{W_{ji}} \right|, \left| \overline{X_i} - \overline{W_{ji}} \right|\right\}$$

e a relação (3), observamos que $d_4(0) = d_M([0, 0], [0.8, 1.2]) + d_M([2.8, 3.2], [1.8, 2.2]) + d_M([4.8, 5.2], [2.8, 3.2]) = 1.2 + 1 + 2 = 4.2$ é a menor distância. Logo o neurônio vencedor é o neurônio $j = 4$.

Para os parâmetros iniciais $\sigma_0 = 0.8$, $\tau = 10$, $\eta_0 = 0.8$, pela relação (6), temos $\sigma(0) = 0.8 \exp\left(-\frac{0}{10}\right) = 0.8$ e a distância lateral entre os neurônio vencedor $j = 4$ e o neurônio $j = 1$ será $d_{1,4} = \sum_{i=1}^3 d_M(W_{1i}, W_{4i}) = 3$. O parâmetro vizinhança topológica do neurônio vencedor $j = 4$ em relação ao neurônio $j = 1$, usando a relação (7) será $h_{1,4}(t) = \exp\left(-\frac{(d_{1,4})(t)}{2\sigma^2(t)}\right) = 0.096$. O parâmetro taxa de aprendizagem para a primeira iteração, de acordo com a relação (9), é dada por $\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_1}\right) = 0.8$.

Utilizando a relação iterativa, dada pela fórmula (8), passamos agora ao ajuste dos pesos sinápticos:

- Para o neurônio $j = 1$: $W_{11} = [1.64, 2.06]$; $W_{12} = [1.69, 2.15]$; $W_{13} = [1.07, 1.54]$.
- Para o neurônio $j = 2$: $W_{21} = [1.77, 2.17]$; $W_{22} = [1.69, 2.15]$; $W_{23} = [0.99, 1.46]$.
- Para o neurônio $j = 3$: $W_{31} = [1.72, 2.13]$; $W_{32} = [1.85, 2.28]$; $W_{33} = [0.92, 1.35]$.
- Para o neurônio $j = 4$: $W_{41} = [0.8, 1.2]$; $W_{42} = [1.8, 2.2]$; $W_{43} = [2.8, 3.2]$.
- Para o neurônio $j = 5$: $W_{51} = [1.77, 2.17]$; $W_{52} = [0.038, 0.043]$; $W_{53} = [0.84, 1.25]$.
- Para o neurônio $j = 6$: $W_{61} = [0.72, 1.14]$; $W_{62} = [1.83, 2.29]$; $W_{63} = [0.31, 0.34]$.

Fazendo uma comparação entre as distâncias dos neurônios ao neurônio vencedor $j = 4$ antes e depois da primeira iteração, temos:

- $d(W_4(0), W_1(0)) = 3$ e $d(W_4(1), W_1(1)) = 2.7$;
- $d(W_4(0), W_2(0)) = 3$ e $d(W_4(1), W_2(1)) = 2.89$;

- $d(W_4(0), W_3(0)) = 3$ e $d(W_4(1), W_3(1)) = 2.89$;
 - $d(W_4(0), W_5(0)) = 5.2$ e $d(W_4(1), W_5(1)) = 5.08$;
 - $d(W_4(0), W_6(0)) = 3.2$ e $d(W_4(1), W_6(1)) = 2.97$.
- 3 ed. Berlin; Heidelberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Singapore; Tokyo. Springer.
- [5] Moore, R. E. Moore. (1966). *Interval Analysis*. Printice Hall, New Jersey.

3 Conclusões

Podemos perceber que todos os neurônios se deslocaram aproximando-se do neurônio vencedor. Dado um conjunto de treinamento com valores intervalares, repetindo o processo descrito acima um número finito de vezes, o algoritmo SOM intervalar tem por objetivo levar um espaço de dados intervalares de alta dimensionalidade para um espaço de dimensão menor que é chamado de mapa auto-organizável de características, como acontece no caso discreto. O uso de dados intervalares tem por objetivo valorizar possíveis incertezas existentes em diversos problemas numéricos. Apesar do estudo em Matemática Intervalar ter se difundido nos últimos anos ainda não temos ferramentas adequadas para abordar problemas que necessitam da utilização de software e algoritmos que lidam com dados intervalares, por isso, é muito importante pensar em novas abordagens, como a proposta do algoritmo SOM intervalar.

Agradecimentos

Os autores agradecem a valiosa colaboração da Sra. Daisy Pierucci que tornou possível uma considerável parcela da organização deste evento. Parte deste trabalho foi realizada com apoio do CNPq, da FAPERJ e da CAPES.

BIBLIOGRAFIA

- [1] Lima, E. L. (2003). *Espaços Métricos*. Rio de Janeiro. Associações Instituto Nacional de Matemática Pura e Aplicada.
- [2] Escarcina, R. E. P. (2004) *Computação Intervalar para Redes Neurais Perceptron*. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte. Centro de Ciências Exatas e da Terra. Departamento de Informática e Matemática Aplicada.
- [3] Haykin S. (1999). *Neural Networks*. 2 ed. Printice Hall, Upper Saddle River, New Jersey.
- [4] kohonen, T. (2001). *Self-Organizing Maps*.