

Otimização do Processo de Construção de Redes Neurais Artificiais com Método de Segmentações Geométricas Sucessivas Utilizando Projeções de Estimador de Densidade para Quebra em Hipervolumes

Lucas Corrêa Netto Machado Leonardo de Mello Honório Augusto Santiago Cerqueira
Universidade Federal de Juiz de Fora Universidade Federal de Juiz de Fora Universidade Federal de Juiz de Fora
Grupo de Robótica Inteligente - GRIn Grupo de Robótica Inteligente - GRIn Grupo de Robótica Inteligente - GRIn
Juiz de Fora, Minas Gerais, Brasil Juiz de Fora, Minas Gerais, Brasil Juiz de Fora, Minas Gerais, Brasil
Email: lucas.correa@engenharia.ufjf.br Email: leonardo.honorio@ufjf.edu.br Email: augusto.s.cerqueira@gmail.com

Abstract—This work presents a new technique for optimization of the process of artificial neural networks assembling and training named Successive Geometric Segmentation Method (SGSM). The SGSM groups the data of each class into hyperboxes (HB) aligned in accordance with the largest axis of its points distribution. If the HB are linearly separable, a separating hyperplane may be identified resulting a neuron. If it is not, the data is divided into smaller classes for new HB. In this case, the chosen technique uses the probability distribution of the data in a hyperbox.

Resumo—Este trabalho apresenta uma técnica a fim de otimizar o processo de construção e treinamento de redes neurais no Método de Segmentações Geométricas Sucessivas (MSGs). O MSGs agrupa os dados de cada classe em hipercaixas (HC) onde cada caixa é alinhada de acordo com os eixos de maior distribuição de pontos. Sendo as caixas linearmente separáveis, um hiperplano de separação é identificado originando um neurônio. Caso não seja possível a separação por um único hiperplano, os dados são divididos em conjuntos menores para obter novas HC. Neste caso, a técnica de divisão estima a densidade de probabilidade dos pontos na hipercaixa para escolher o ponto de corte.

I. INTRODUÇÃO

Para se treinar uma Rede Neural Artificial (RNA) é comum precisar de definir a topologia a ser utilizada, o que pode influenciar na qualidade da rede. Para o caso das redes feed-forward, foi demonstrado que são necessárias no mínimo três camadas de neurônios para que a rede consiga aproximar qualquer tipo de função, isto é conhecido como teorema de Komogorov [1]. Apesar disto, não existe maneira para definir a quantidade ótima de neurônios necessários para garantir o bom funcionamento da rede. Algumas abordagens foram desenvolvidas com este intuito, obtendo apenas sucessos parciais [2].

Já o processo de classificar é uma necessidade real presente em diversas áreas, sendo duas abordagens bastante utilizadas atualmente: uma baseada em técnicas de classificação geométricas, como *support vector machines* ou SVM [3], e a segunda em teorias de inteligência computacional, como as redes neurais

artificiais [4]. No caso da técnica baseada em SVM, obtém-se altas taxas de acerto com boa fundamentação matemática, no entanto, apresentam dificuldades para escolha da função kernel e determinação de parâmetros e margem. As redes neurais artificiais podem facilmente ser aplicadas em praticamente todos os problemas de classificação. Entretanto necessita da especificação de diversos parâmetros, principalmente os relacionados à topologia da rede [5], como o número de camadas e o número de neurônios por camada, que são dependentes do tipo de problema a ser tratado.

Uma das metodologias mais rápidas de geração de classificadores é a técnica de separação geométrica através de intervalos. Estes métodos utilizam funções lineares ou não-lineares [6] para diferenciar as classes. Quando funções lineares são utilizadas o treinamento é rápido, entretanto, a resposta pode não ser satisfatória devido a uma série de fatores. Por outro lado, funções não-lineares apresentam uma melhor resposta, contudo, assim como metodologias de treinamento de redes neurais [2], necessitam de muito tempo de treinamento.

II. DESCRIÇÃO DO MSGS

Como proposta de uma nova metodologia de construção de redes neurais, este trabalho apresenta o Método de Segmentações Geométricas Sucessivas com segmentação de dados utilizando um estimador de densidade de probabilidade. O fluxograma do MSGS utilizado neste trabalho é mostrado na Figura 1, e se inicia através da construção do envelope dos dados através do uso de hipercaixas orientadas: *Oriented Bounded HyperBox (OBHB)*. Com cada classe que se deseja identificar sendo representada por um OBHB, inicia-se a segunda etapa onde é necessário verificar se os OBHBs estão ou não em colisão. Uma colisão significa que estão ocupando o mesmo lugar no espaço e não podem ser classificados corretamente. Para tanto, será utilizado o Teorema do Hiperplano de Separação (THS) que é apresentado em detalhes em [7].

Como o método de separação de classes é baseado em funções lineares, poderão existir colisões. Uma forma de contornar esta situação é utilizar árvores de colisão [8]. Originalmente esta metodologia propõe a representação de um objeto através de uma árvore hierárquica utilizando volumes geométricos como, no caso, OBHBs. Uma árvore de colisão

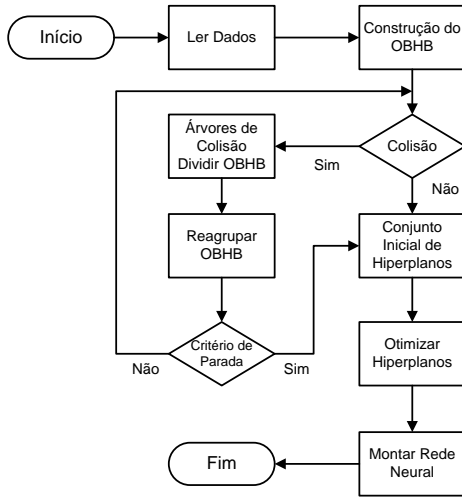


Fig. 1: Fluxograma do Método de Segmentações Geométricas Sucessivas

divide, segundo um critério heurístico, os dados presentes em um único OBHB, em dois. Cada novo OBHB gerado a partir do anterior aumenta a precisão do contorno do envelope em relação aos dados originais. Este processo será descrito na Subseção II-C. Adicionalmente, verifica-se que esta divisão gera a necessidade de novos testes de colisão, o que incorre na identificação de novos hiperplanos de separação. Desta forma, para evitar excesso de OBHBs torna-se necessário acrescentar uma etapa de agrupamentos conforme descrito na Subseção II-D. Este processo ocorre recursivamente até que não sejam detectadas mais colisões ou até que se chegue a um critério de parada.

Por outro lado, caso os OBHBs testados não estejam em colisão, tem-se através do THS, utilizado para verificar colisões, o conjunto inicial de hiperplanos. No entanto, como será mostrado na Subseção II-E, ocorre a necessidade de otimizar o conjunto inicial para tornar a rede mais eficiente. Finalmente, na Subseção II-F será apresentado a estratégia adotada para montagem da rede neural.

A. Construção da OBHB

A primeira etapa do MSGS é a representação dos dados por um envelope. Na metodologia adotada os dados são representados por uma caixa alinhada ao eixo de maior distribuição de pontos, dado pelo autovetor da matriz de covariância com maior autovalor associado. O método *OBB - Oriented Bounded Box* foi inicialmente desenvolvido para o espaço tridimensional e visando exclusivamente testes de colisão em sistemas de realidade virtual. A seguir uma generalização desta técnica é apresentada para \mathbb{R}^n .

Seja \vec{X} um conjunto de m pontos em \mathbb{R}^n . A construção de uma hipercaixa envolvente alinhada com a distribuição de \vec{X} se inicia através do cálculo do conjunto de autovetores $\vec{\Pi}$ e autovalores $\vec{\Lambda}$ da matriz de covariância $Cov(\vec{X})$. Onde $\vec{\Pi}$ determina vetores alinhados com a dispersão dos pontos ao longo de \mathbb{R}^n . Estes autovetores são utilizados para gerar um novo sistema de coordenadas $\mathcal{C}(\vec{X})$ onde a origem se encontra no ponto médio \vec{x}_c dos elementos de \vec{X} .

Logo, para uma dada classe de pontos \vec{X} é realizada a transformação de cada um de seus pontos para o novo sistema

de coordenadas $\mathcal{C}(\vec{X})$ obtendo-se $\vec{V} = \{\vec{v}^1, \dots, \vec{v}^m\}$ onde para um dado $\vec{v}^k \in \vec{V}$ têm-se

$$\vec{v}^k = \{Pr(\vec{x}^k, \pi_1), \dots, Pr(\vec{x}^k, \pi_n)\} \quad (1)$$

e

$$Pr(\vec{x}^k, \pi_i) = \frac{(\vec{x}^k - \vec{x}_c) \cdot \vec{\pi}_i}{|\vec{\pi}_i|} \quad (2)$$

Assim sendo, utilizando \vec{V} é possível obter os limites da projeção (valor máximo e mínimo) dos elementos de \vec{X} sobre cada um dos eixos existentes em $\vec{\Pi}$. Logo pode-se definir uma função de projeção responsável por determinar um segmento de reta formado pelos limites de \vec{X} sobre um dado eixo π_p como sendo

$$L\vec{P}r(\vec{X}, \pi_p) = \{v_p^{min}, v_p^{max}\} \quad (3)$$

onde

$$v_p^{min} = MIN(v_p^K), v_p^{max} = MAX(v_p^K)$$

e K é o conjunto de todos os elementos da classe. Considerando estes segmentos de reta como arestas é possível definir uma hipercaixa $\vec{\mathcal{H}}$ envolvente

$$\vec{\mathcal{H}} = \{L\vec{P}r(\vec{X}, \pi_1), \dots, L\vec{P}r(\vec{X}, \pi_n)\} = \{L\vec{P}r(\vec{X}, \vec{\Pi})\} \quad (4)$$

A Figura 2 ilustra o exemplo e mostra a construção dos limites da hipercaixa $\vec{\mathcal{H}}^{1L}$ que circunscreve o conjunto de pontos \vec{X}^{1L} . Os eixos π_1 e π_2 são construídos a partir do ponto médio \vec{x}_c segundo os autovetores da matriz de covariância. Cada ponto $\vec{x}_c \in \vec{X}^{1L}$ é transformado e projetado para os eixos deste novo sistema de coordenadas. Os valores mínimo e máximo da projeção dos pontos em cada eixo definirão os limites da hipercaixa.

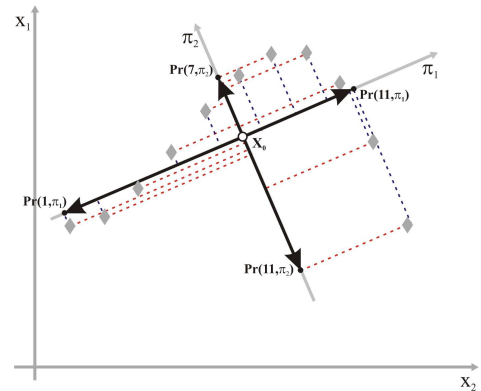


Fig. 2: Projeções de \vec{X}^{1L} definindo os limites da Caixa Envolvente $\vec{\mathcal{H}}^{1L}$

Com o centro da hipercaixa calculado pode-se definir Δ_p e δ_p , sendo Δ_p o comprimento da projeção completa, resultado do tamanho total da caixa projetada sobre o eixo π_p , e δ_p o tamanho da projeção parcial, comprimento do centro da caixa à sua borda sobre o eixo π_p como

$$\Delta_{\vec{\mathcal{H}}, p} = |L\vec{P}r(\vec{X}, \pi_p)| = \sqrt{(v_p^{max} - v_p^{min})^2}. \quad (5)$$

e

$$\delta_{\vec{\mathcal{H}}, p} = \frac{\Delta_{\vec{\mathcal{H}}, p}}{2}, \quad (6)$$

$$\vec{\delta}_{\vec{\mathcal{H}}, p} = \delta_{\vec{\mathcal{H}}, p} \times \vec{\pi}_p \quad (7)$$

e o centro da hipercaixa representado por $\vec{\mathcal{H}}_0$ é dado por

$$\vec{\mathcal{H}}_0 = \left\{ \frac{v_1^{max} + v_1^{min}}{2}, \dots, \frac{v_n^{max} + v_n^{min}}{2} \right\} \quad (8)$$

Como \vec{X} e $\vec{\mathcal{H}}$ são vetores formados por pontos em \mathbb{R}^n com $\vec{\mathcal{H}}$ sendo uma projeção de \vec{X} em $\vec{\Pi}$ pode-se ainda definir que para o sistema de coordenada \mathcal{C} , têm-se que as projeções de \vec{X} e $\vec{\mathcal{H}}$ são iguais, ou seja:

$$Pr(\vec{X}, \vec{\Pi}) = Pr(\vec{\mathcal{H}}, \vec{\Pi}) \quad (9)$$

B. Teste de Colisão

Neste ponto é necessário um procedimento para identificar se duas hipercaixas $\vec{\mathcal{H}}^1$ e $\vec{\mathcal{H}}^2$ ocupam o mesmo lugar no espaço. A proposta inicialmente demonstrada em [9] apresenta um método denominado *Teorema do Hiperplano de Separação (THS)* para realizar o teste de colisão. Os parâmetros de um hiperplano são utilizados para a criação de um neurônio correspondente.

1) *Teorema do Eixo de Separação (TES)*: Sejam duas hipercaixas $\vec{\mathcal{H}}^1$ e $\vec{\mathcal{H}}^2$ em \mathbb{R}^n com seus respectivos centros em $\vec{\mathcal{H}}_0^1$ e $\vec{\mathcal{H}}_0^2$ formando o conjunto de pontos $\vec{X}_c = \{\vec{\mathcal{H}}_0^1, \vec{\mathcal{H}}_0^2\}$. Se existir um eixo $\pi_p \in \mathbb{R}^n$ que atenda à inequação

$$|LPr(\vec{X}_c, \pi_p)| \geq \frac{|LPr(\vec{\mathcal{H}}^1, \pi_p)| + |LPr(\vec{\mathcal{H}}^2, \pi_p)|}{2} \quad (10)$$

ou, de forma análoga,

$$\Delta_{\vec{X}_c, p} \geq \delta_{\vec{\mathcal{H}}^1, p} + \delta_{\vec{\mathcal{H}}^2, p}. \quad (11)$$

onde

$$\delta_{\vec{\mathcal{H}}, p} = \sum_{i=1}^n \left| \frac{\vec{\delta}_{\vec{\mathcal{H}}, i} \cdot \vec{\pi}_p}{|\vec{\pi}_p|} \right| \quad (12)$$

os conjuntos representados por $\vec{\mathcal{H}}^1$ e $\vec{\mathcal{H}}^2$ estão devidamente separados. Logo, para se determinar que não existe colisão, é necessário encontrar um eixo onde a projeção das imagens das hipercaixas não é contínua, ou seja, um eixo sobre o qual a projeção do vetor que liga os centros das hipercaixas tenha módulo menor que as projeções das hipercaixas somadas.

A Figura 3 ilustra este procedimento onde $\{\pi_1, \pi_2\}$ e $\{\pi_3, \pi_4\}$ são, respectivamente, as bases dos sistemas de coordenadas \mathcal{C}^1 e \mathcal{C}^2 que contêm duas hipercaixas $\vec{\mathcal{H}}^1$ e $\vec{\mathcal{H}}^2$. Seja $\vec{\Delta}_{\vec{\mathcal{H}}_0^1, \vec{\mathcal{H}}_0^2}$ um vetor que vai do centro de $\vec{\mathcal{H}}^1$ para $\vec{\mathcal{H}}^2$. As projeções de $\vec{\mathcal{H}}^1$ sobre os eixos π_1 e π_2 e as projeções de $\vec{\mathcal{H}}^2$ sobre π_3 e π_4 são representadas por $\delta_{1,1}, \delta_{1,2}, \delta_{2,3}, \delta_{2,4}$ respectivamente. Aplicando o *TES* para o eixo $\pi_3 = x_2$ têm-se que a soma de $|\delta_{\vec{\mathcal{H}}^1, 3}|$ com $|\delta_{\vec{\mathcal{H}}^2, 3}|$ é inferior a projeção dos centros das caixas $|\vec{\Delta}_{\vec{\mathcal{H}}_0^1, \vec{\mathcal{H}}_0^2}|$, atendendo à Equação 11. Desta forma o eixo π_3 é considerado um eixo de separação.

2) *Teorema do Hiperplano de Separação (THS)*: Considerando duas hipercaixas $\vec{\mathcal{H}}^1$ e $\vec{\mathcal{H}}^2$ no espaço \mathbb{R}^n , se as caixas não estiverem em colisão, existe um eixo de separação π_p que contém as projeções $\delta_{\vec{\mathcal{H}}^1, p}$ e $\delta_{\vec{\mathcal{H}}^2, p}$, conforme demonstrado na Subsecção II-B1.

Considerando π_p como sendo o eixo de separação é possível definir um segmento de reta \mathcal{S} sobre este eixo tal que

$$|\mathcal{S}| = \Delta_{\vec{X}_c, p} - \delta_{\vec{\mathcal{H}}^1, p} - \delta_{\vec{\mathcal{H}}^2, p}. \quad (13)$$

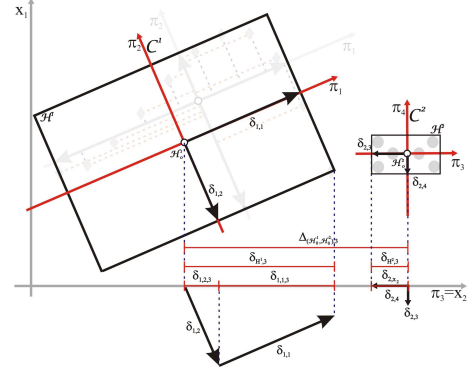


Fig. 3: Identificação do Eixo de Separação

Isto significa que existem pontos pertencentes ao eixo π_p que estão localizados entre as projeções das hipercaixas. Utilizando assim o segmento de reta \mathcal{S} é possível então definir um hiperplano $\chi = 0 \in \mathbb{R}^n$ que seja normal à π_p e que contenha um ponto \vec{x}^r sobre o segmento de reta \mathcal{S} , logo

$$\chi = \vec{\pi}_p \cdot \vec{X} - \vec{\pi}_p \cdot \vec{x}^r \quad (14)$$

onde \vec{X} é um conjunto de pontos pertencentes a \mathbb{R}^n que atendem a equação 14, formando assim o hiperplano χ . Logo $\chi(\vec{\mathcal{H}}^i, \vec{\mathcal{H}}^j)$ é definido como sendo um hiperplano de separação entre as hipercaixas $\vec{\mathcal{H}}^i$ e $\vec{\mathcal{H}}^j$.

A Figura 4 exemplifica este processo onde $\vec{\mathcal{H}}^1$ e $\vec{\mathcal{H}}^2$ são duas hipercaixas em \mathbb{R}^2 . O segmento de reta \mathcal{S} é definido considerando os valores das projeções $\delta_{\vec{\mathcal{H}}^1, p}$ e $\delta_{\vec{\mathcal{H}}^2, p}$ sobre o eixo de separação π_p , e os centros das caixas $\vec{\mathcal{H}}_0^1$ e $\vec{\mathcal{H}}_0^2$. Utilizando o ponto \vec{x}^r sobre o segmento de reta \mathcal{S} , criou-se o hiperplano de separação $\chi(\vec{\mathcal{H}}^i, \vec{\mathcal{H}}^j)$ perpendicular ao eixo π_p . Uma observação importante é que *pode existir mais de um eixo de separação entre duas hipercaixas pois qualquer ponto sobre \mathcal{S} é candidato para a construção de $\chi(\vec{\mathcal{H}}^i, \vec{\mathcal{H}}^j)$* . Neste trabalho o ponto considerado para análise foi o ponto médio de \mathcal{S} .

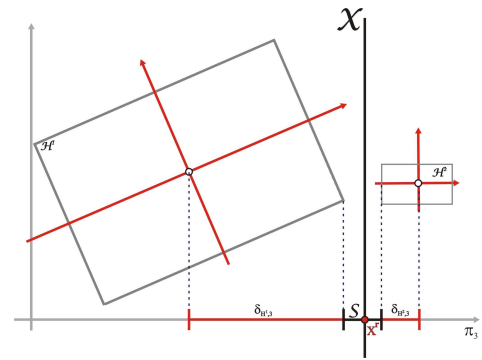


Fig. 4: Definição do Hiperplano de Separação

C. Segmentação Geométrica dos OBHB

Dado que ocorreu colisão, torna-se necessário aumentar a representatividade da hipercaixa com a base de dados em análise. Uma técnica apresentada na literatura [7] é através da segmentação geométrica dos dados, gerando uma busca denominada de árvore de Colisão (AC). Similar às tradicionais buscas em árvore, a técnica de AC divide os dados de uma

classe em uma forma hierárquica onde a cada novo nível o conjunto total de pontos é dividido e, para cada divisão, uma nova hipercaixa é criada aumentando a precisão do envelope.

A metodologia utilizada se baseia em um estudo considerando a projeção da densidade de probabilidade sobre os eixos da hipercaixa. Esta análise será tratada a fundo na seção III.

D. Reagrupamento de OBHB

Neste ponto as hipercaixas já segmentadas em caixas filhas apresentam separação umas das outras. Com isso consegue-se obter um conjunto de neurônios que divide o espaço em diversas regiões que formam instâncias de uma dada classe.

A Figura 5b mostra a construção inicial das caixas onde pode-se visualizar a existência de colisão. Seguindo o procedimento descrito na Seção II-C, a quebra continua sempre pela maior caixa que apresenta colisão até que o conjunto esteja separado. A Figura 5c mostra o final do processo onde as caixas não apresentam nenhuma colisão. Entretanto, apesar de todas as caixas estarem separadas, é possível observar que algumas regiões não foram corretamente identificadas. Apesar desta situação não ser necessariamente um erro, seria mais desejável uma melhor especialização de cada região.

São adotadas estratégias na tentativa de solucionar o problema na identificação ressaltada: algumas caixas são quebradas novamente por apresentarem indícios de não serem representativas de uma só região e, após isso, as caixas resultantes são reagrupadas de uma outra forma. Cada uma das caixas encontradas é avaliada novamente sobre a possibilidade de quebra buscando permanecer apenas com caixas representativas da distribuição dos pontos. Posteriormente um reagrupamento é realizado da seguinte forma: uma lista das regiões é indexada da menor para a maior área. Seguindo esta ordem, cada elemento da lista é reagrupado com o seu par mais próximo pertencente a mesma instância de saída. Se esta nova hipercaixa não apresentar colisões, a união se mantém, uma nova lista indexada é criada e o processo recomeça do menor elemento da lista. Entretanto, caso a caixa esteja em colisão, as caixas originais se mantêm e o processo continua normalmente para o restante da lista. A Figura 5d mostra o resultado final deste processo.

E. Otimização dos Hiperplanos

Com o prosseguimento do algoritmo de busca chega-se a um ponto onde as caixas, já segmentadas em caixas filhas, apresentam separação umas das outras. Neste ponto é possível encontrar todos os planos de separação conforme apresentados na Figura 5e. Pode-se observar na figura que o excesso de hiperplanos encontrados iria gerar uma rede neural muito mais complexa e especializada do que o necessário.

Para simplificar a rede neural gerada, é necessário encontrar qual o menor conjunto de hiperplanos que separe todas as regiões pertencentes à diferentes classes. Isto é um problema de otimização discreta e pode ser abordado por diversos métodos. Entretanto, como a principal finalidade deste trabalho é mostrar a eficácia da nova abordagem de quebra de hipervolumes, bem como do método de construção da rede neural, a solução apresentada aqui baseia-se em um critério de busca em profundidade simples sem que nenhuma otimização mais elaborada tenha sido aplicada.

Adotando o método de busca em profundidade, o critério de escolha de um dado hiperplano é um índice determinando quantas regiões ele separa. Desta forma todos os hiperplanos são indexados de acordo com este índice, e a cada iteração do

algoritmo de busca um novo hiperplano é adicionado à lista selecionada. Para evitar redundância, as regiões separadas são marcadas não sendo necessária a adição de novos hiperplanos para a sua classificação. Este processo continua até que todas as hipercaixas estejam separadas. Aplicando este procedimento no resultado mostrado na Figura 5e o número de hiperplanos caiu de 798 para apenas 54 (Figura 5f).

F. Montagem da Rede Neural

A rede neural gerada pela metodologia proposta é composta de uma camada de entrada, três camadas ocultas e uma camada de saída. A funcionalidade de cada camada é descrita abaixo:

- Camada de Entrada: responsável por distribuir as entradas para os neurônios da camada oculta.
- Primeira Camada Oculta: cada hiperplano encontrado para separar as hipercaixas gera um neurônio seguindo a Equação (14), o que torna possível avaliar como um determinado ponto se encontra em relação a um certo hiperplano. Caso o valor de saída do neurônio seja positivo, o ponto se encontra acima do hiperplano, caso o valor seja negativo, o ponto se encontra abaixo, e caso zero, sobre o hiperplano.
- Segunda Camada Oculta: responsável por processar o conjunto de hiperplanos que definem a pertinência de um dado ponto a uma região definida do espaço. Cada hipercaixa encontrada gera um neurônio na camada. Um valor positivo indica que o ponto se encontra dentro da região limitada pelos hiperplanos que separam a hipercaixa de todas as outras.
- Terceira Camada oculta: possui número de neurônios igual ao número de instâncias que se deseja classificar e retorna valores reais para ambas as instâncias. A saída positiva do neurônio indica que o ponto testado se encontra em alguma das regiões pertencentes à classe que se refere ao neurônio.
- Camada de Saída: processa os valores da terceira camada oculta. Esta camada possui o mesmo número de neurônios que a camada anterior e além de examinar os resultados, retorna um valor positivo para o neurônio que representa a instância a qual o ponto testado foi classificado. Caso o ponto testado não pertença a nenhuma região definida, o valor de ativação será igual a uma média ponderada dos valores de saída, ou seja, um índice que determina com um certo grau de confiabilidade a região de fronteira da instância mais próxima do ponto testado.

III. QUEBRA DE HIPERCAIXAS CONSIDERANDO A DENSIDADE DE PROBABILIDADE

O presente capítulo trata a análise da densidade de probabilidade dos dados de cada caixa, projetados sobre os eixos da própria OBHB a fim de otimizar a quebra de hipercaixas mostrada na Seção II-C e evitar alguns problemas de representatividade dos dados pela caixa como pôde ser observado na Figura 5c.

Para isso, são obtidos estimadores da densidade de probabilidade dos pontos sobre algum dos eixos da OBHB e analisando essa função, normalmente se observa um comportamento típico quando os dados não são da mesma fonte: um vale separando dois picos centrados nos locais de maior concentração dos dados. Considerando a convergência de funções densidade de probabilidade, quanto mais afastados um ponto do outro,

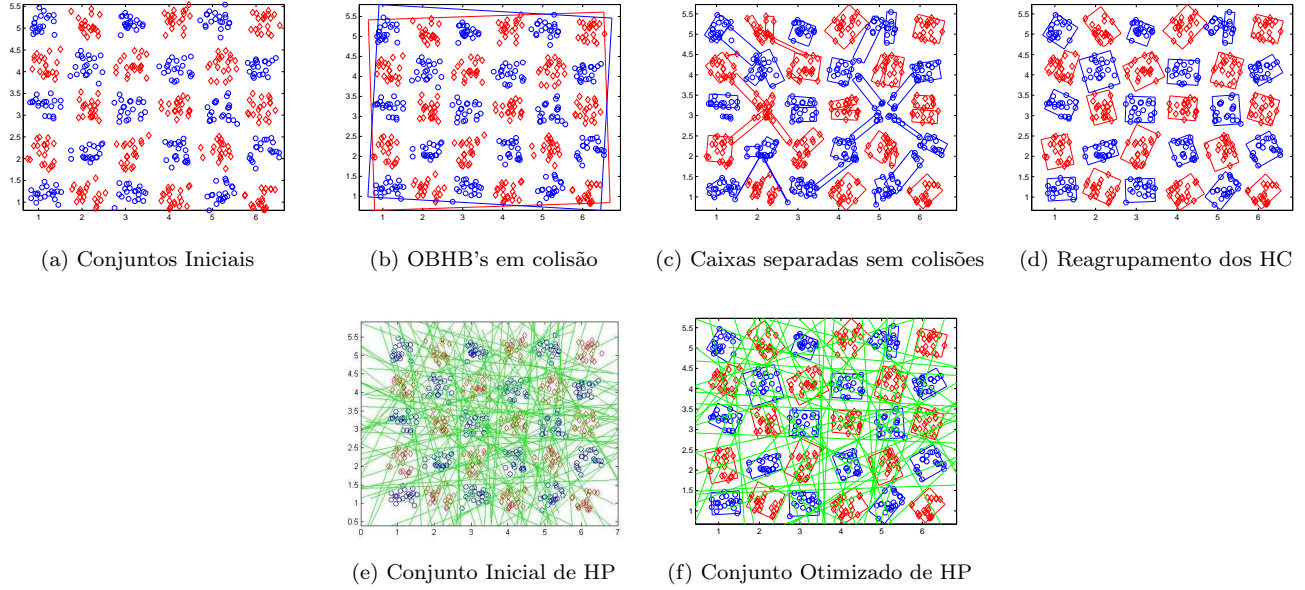


Fig. 5: Exemplo de Geração e otimização de Rede Neural utilizando o TES

menor a interferência construtiva na formação da densidade da hipercaixa.

Desta forma obtêm-se de maneira prática um melhor ponto de quebra do que quando se considera apenas o sinal da projeção sobre o eixo, ou área das caixas filhas após uma quebra (Seção III-B). Esta melhora pode ser percebida tanto visualmente através de figuras que serão mostradas, quanto no tempo gasto durante o processo. Essa abordagem, para os casos testados, permitiu uma redução na quantidade de testes de colisão realizados, uma vez que as primeiras quebras efetuadas já criam envelopes mais representativos do conjunto de dados.

A. Kernel Density Estimation

A fim de se obter a projeção da densidade de probabilidade sobre o eixo π_j , de um determinado conjunto de pontos \vec{X}^i delimitado pela caixa \vec{H}^i , cada ponto $\vec{x}^k \in \vec{X}^i$ com $k = 1, 2, \dots, m$ é tratado por uma distribuição normal com média $\mu = x_j^k$ e desvio padrão σ fixo para todos os pontos conforme mostra a Equação (15). Na qual $x \in [L\vec{P}r(\vec{X}, \pi_j)]$.

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (15)$$

A densidade segue a Equação (16), dada pelo somatório das funções considerando todos os m pontos de \vec{X}^i . Desta forma, existe contribuição de todos os elementos para gerar a densidade de probabilidade da hipercaixa e quanto mais afastado um ponto está do outro, mais nítido o vale entre eles.

$$dp(\vec{H}^i, \pi_j) = \frac{1}{m} \sum_{k=1}^m f(x, x_j^k, \sigma) \quad (16)$$

A escolha de um valor σ permite determinar a distância até onde um ponto afetará na densidade. A medida que dois pontos se afastam, a contribuição conjunta para o valor final

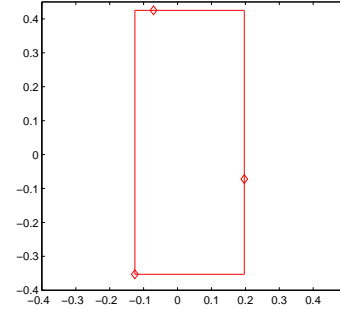


Fig. 6: OBHB gerado com três pontos

diminui, uma vez que a probabilidade de eles estarem num mesmo conjunto é menor.

São mostrados exemplos utilizando dados sintéticos que seguem um padrão conhecido. O primeiro, ilustrado na Figura 6 é formado por três pontos que são gerados em duas dimensões com distribuição uniforme de média zero em ambas coordenadas.

A Figura 7a mostra um exemplo da estimação da densidade de probabilidade, na qual as curvas tracejadas são referentes à gaussiana devido a cada um dos pontos (Equação (15)), e a curva contínua representa o somatório das três gaussianas (Equação (16)). O eixo das abcissas representa o eixo $\pi_i \in \mathcal{C}(\vec{X})$

É possível perceber que entre os pontos mais afastados da HC (Figura 6) surge um indício da formação de um vale, o que se torna mais nítido a medida que os pontos se afastam, ou diminuindo o valor do desvio padrão σ considerado.

O valor de desvio padrão escolhido para gerar a gaussiana devido a cada instância, está intimamente ligado à interferência entre os dados de um conjunto na construção da densidade, e

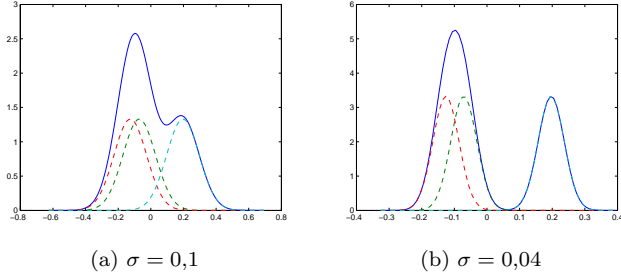


Fig. 7: distribuição gaussiana para três pontos de mesma fonte e somatório das distribuições

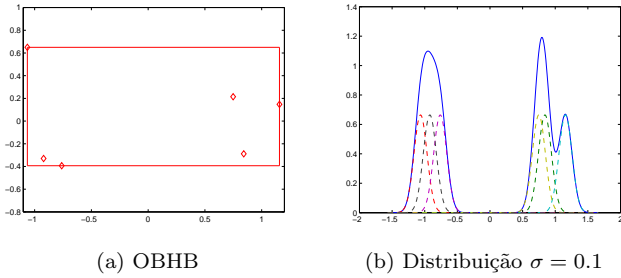


Fig. 8: OBHB com seis pontos gerados, pertencentes a duas casas diferentes e sua distribuição de probabilidade

desta forma quanto menor o valor de σ mais próximo devem estar os pontos para que haja contribuição entre as gaussianas, e assim, gerar uma distribuição contínua, sem vales. A Figura 7 mostra um comparativo com curvas geradas usando dois valores diferentes de σ , na qual fica possível perceber a diferença na interação entre as curvas realçando a descontinuidade na distribuição dos pontos dentro da hipercaixa (Figura 6).

Já no segundo exemplo, mostrado na Figura 8a, a OBHB é formada por instâncias geradas seguindo o mesmo processo do primeiro exemplo, porém se diferenciam pela média no eixo das abcissas, três com média $x = -1$ e três com média $x = 1$. A Figura 8b ilustra a distribuição encontrada para esta HC, que deixa visualmente perceptível uma separação entre dois conjuntos de dados, já que o vale fica nítido e é formado por um grande intervalo no qual o valor da distribuição é zero. Esta forma de curva de densidade facilita a quebra, como será abordado na Seção III-B.

B. Quebra de HC

A quebra de uma hipercaixa considerando a estimação da densidade de probabilidade é realizada utilizando um valor de limiar, sendo assim, valores de distribuição menores que este limiar são tratados como pertencentes a um intervalo para quebra da OBHB.

A Figura 9a mostra um caso frequente no qual a HC agrupa um conjunto de dados \vec{X} de múltiplas fontes, mas que pertencem a uma mesma classe, que pode ser um exemplo de um conjunto de dados não contínuos no espaço de solução. Como pode ser visto de forma semelhante na Figura 5c

O fato da hipercaixa não ter uma distribuição contínua ao longo de toda sua extensão e o modo como os pontos estão

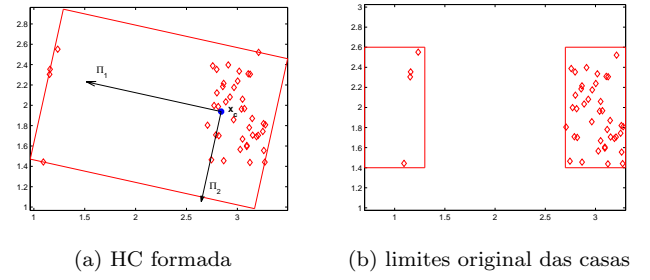


Fig. 9: Instâncias de duas casas diferentes

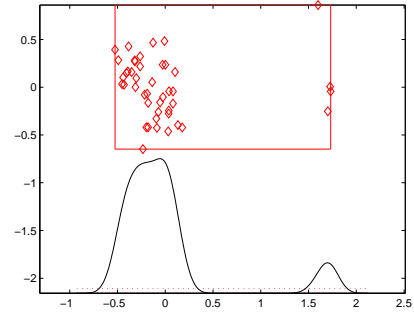


Fig. 10: densidade de probabilidade da HC considerando eixo de maior autovalor

separados dentro da HC, faz com que o ponto médio \vec{x}_c se encontre em uma das nuvens de pontos (Figura 9a).

A quebra da Hipercaixa é realizada tomando o eixo de maior autovalor π_1 . Caso a forma de quebra considerada seja utilizando o ponto médio \vec{x}_c (separando os valores positivos de projeção sobre o eixo π_1 em uma nova caixa, e os valores negativos em outra), percebe-se uma quebra na qual uma das caixas filhas continua contendo pontos de duas casas diferentes, como pode ser visto na Figura 11a que mostra as duas novas caixas geradas ('*' e 'o'). Percebe-se inclusive, que as duas caixas colidem, já que caixa '*' contem pontos que deveriam pertencer à caixa 'o'. Por outro lado, se for levado em conta a densidade de probabilidade da HC, o ponto de quebra escolhido fica entre os dois picos da curva o que favorece a quebra (Figura 10 mostra a caixa em relação ao sistema de coordenadas $\mathcal{C}(\vec{X})$). Desta forma, as caixas filhas resultantes do processo se adaptam melhor à formação dos dados, evitando que uma HC englobe pontos gerados com diferentes distribuições (Figura 11b).

Para os casos testados, este tipo de quebra favorece todo o processo, como pode ser visto analisando a Figura 12, que fornece uma imagem do processo de construção da rede neural usando o MSGS no ponto em que as hipercaixas foram quebradas até não haver mais colisão utilizando o estimador de densidade de probabilidade para quebra, ponto esse que permite uma analogia com a Figura 5c.

IV. RESULTADOS

Serão mostrados dois conjuntos de testes, o primeiro utilizando dados pertencentes a duas classes diferentes descontínuas no espaço de solução com distribuição similar àquela mostrada na Figura 5a. É possível perceber nessa figura os

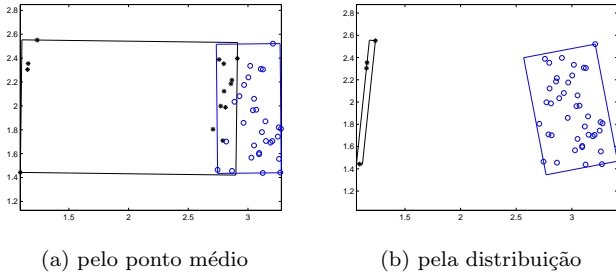


Fig. 11: quebra da HC

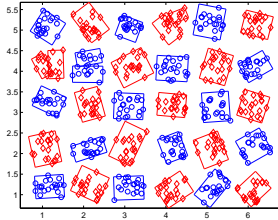


Fig. 12: MSGS com quebras utilizando distribuição de probabilidade

dados distribuídos seguindo a forma de um tabuleiro, intercalando no espaço pequenos conjuntos de dados pertencentes a cada uma das classes. Cada um desses pequenos conjuntos será tratado como "casas" do "tabuleiro" que é o conjunto total dos dados. Para o segundo teste, foram utilizados cinco bancos de dados obtidos do repertório disponibilizado UCI [10] que são detalhados na Tabela I, estes bancos foram escolhidos por serem utilizados na literatura [6] e [11].

Para todos os casos o algoritmo MSGS foi desenvolvido em *Matlab* e as simulações realizadas num computador com processador intel core i7 e 8Gb Ram.

Como dito na Seção II-C, a metodologia utilizada para a seleção do hiperplano de separação definida aqui refere-se ao ponto central do segmento \mathcal{S} . Também é importante mencionar que a metodologia de trabalho para multiclasse escolhida foi a *one-against-all*.

TABLE I: Descrição dos Bancos Públicos

	N. Classes	Instâncias	Atributos
<i>Iris</i>	3	150	4
<i>Glass</i>	6	214	9
<i>Wine</i>	3	178	13
<i>Vowel</i>	10	528	11
<i>Vehicle Silhouettes</i>	4	846	18

A tabela II mostra alguns dos resultados realizados a fim de comparar o funcionamento do método utilizando a quebra de hipercaixas através do ponto médio (Colunas xc1 e xc2), e utilizando a quebra considerando a densidade de probabilidade da hipercaixa (colunas dp1 e dp2).

As colunas xc1 e xc2 bem como as dp1 e dp2 se diferenciam devido à forma como as redes obtidas no treinamento

foram testadas. Em xc1 e dp1 os dados gerados para teste são encontrados de forma randômica, seguindo uma distribuição similar a dos dados de treino. Por outro lado, em xc2 e dp2 são criados pontos igualmente espaçados entre si em ambas dimensões, cobrindo todo o espaço de soluções.

Algumas configurações diferentes foram consideradas para realizar a comparação, alterando o número de casas (Coluna C_s da tabela), e o número de amostras em cada casa (Coluna Ins da tabela). Os resultados são mostrados de duas formas diferentes: a taxa de eventos classificados corretamente no teste da rede neural obtida pelo método (linha S(%) na tabela), e o tempo levado para obter a rede neural utilizando o MSGS (linha T(s) na tabela).

TABLE II: Resultados comparativos com a FT com 4, 9 e 30 casas. Diferentes quebras.

C_s	Ins	Idx	xc1	xc2	dp1	dp2
4	5	S(%)	99	73	99	73
		T(s)	0,03		0,02	
	10	S(%)	100	82	100	80
		T(s)	0,04		0,03	
	20	S(%)	100	81	100	83
		T(s)	0,02		0,01	
9	5	S(%)	97	74	97	74
		T(s)	0,16		0,16	
	10	S(%)	100	74	100	75
		T(s)	0,26		0,07	
	20	S(%)	100	80	100	79
		T(s)	0,14		0,04	
30	5	S(%)	92	69	94	70
		T(s)	2,04		0,48	
	10	S(%)	99	73	98	72
		T(s)	3,16		0,36	
	20	S(%)	99	75	100	75
		T(s)	3,10		0,34	

Observando a diferença na taxa de sucesso entre as colunas relativas a quebra pelo ponto médio e as colunas relativas a quebra pela densidade de probabilidade, pode-se perceber que em geral, a técnica de quebra mostrada neste trabalho apresenta um melhor número de acertos, o que mostra uma tendência desta quebra a evitar redes que não são capazes de generalizar. Uma outra análise pertinente é devida ao tempo gasto pelo método para gerar a rede, que foi sempre menor ao utilizar a quebra pela densidade chegando a passar de 10x mais rápido.

O segundo conjunto de testes realizado utiliza os bancos de dados disponibilizados pela UCI, e a comparação de desempenho é mostrado na tabela III, na qual é possível ter acesso à taxa de acerto ao testar a rede gerada utilizando o MSGS, e também aos resultados de taxa de sucesso citados em outras referências para os mesmos bancos de dados ([6] e [11]). Para obter a taxa de sucesso mostrada para o MSGS, foi utilizado a metodologia *ten fold*.

O MSGS apresenta um tempo rápido para treinamento, variando de 0,4 segundos para o '*Iris*' até 11,5s para o banco de dados '*Wine*'. A análise quantitativa mostra que o método apresentado foi o mais robusto para os casos analisados, com uma melhor taxa de identificação em 4 dos 5 conjuntos testados, apesar de em alguns casos, apresentar rendimento similar às referências. É interessante ressaltar que mesmo tendo apresentado um resultado melhor, o algoritmo testado ainda é uma

versão inicial do método MSGS, exigindo ainda uma análise em algumas etapas do processo. Análise essa que tende a melhorar tanto o desempenho computacional quanto quantitativo do MSGS, assim como o que foi realizado neste trabalho.

TABLE III: Resultados Quantitativos

	MSGS	Hsu [11]	Fung [6]
	s(%)	s(%)	s(%)
Iris	99,3	97,3	98,7
Glass	89,5	73,8	72,9
Wine	100	99,4	100,0
Vowel	98,5	99,0	98,5
Vehicle	90,8	87,5	82,2

V. CONCLUSÕES FINAIS

Este trabalho apresentou uma técnica buscando a otimização da etapa de quebra de hipervolumes para o Método das Segmentações Geométricas Sucessivas (MSGS) para treinamento e geração autônoma de redes neurais. A análise da projeção da densidade de probabilidade sobre os eixos da HC possibilitou uma quebra eficaz com uma taxa de acerto que pode ser considerada bastante satisfatória tendo em vista que o método não exige o conhecimento da topologia da rede a ser gerada.

AGRADECIMENTOS

Os autores agradecem ao Instituto Nacional de Energia Elétrica (INERGE), CNPq, CAPES/FCT e a EDP pelo suporte dado no desenvolvimento deste trabalho.

REFERENCES

- [1] R. Beale, T. Jackson, and T. Jackson, *Neural computing: an introduction*. Hilger, 1990.
- [2] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Morgan Kaufmann Pub, 2011.
- [3] S. Abe, *Support vector machines for pattern classification*. Springer-Verlag New York Inc, 2010.
- [4] A. Alves da Silva, V. Ferreira, and R. Velasquez, "Input space to neural network based load forecasters," *International Journal of Forecasting*, vol. 24, no. 4, pp. 616–629, 2008.
- [5] H. Huang, G. Feng, and J. Cao, "Robust state estimation for uncertain neural networks with time-varying delay," *Neural Networks, IEEE Transactions on*, vol. 19, no. 8, pp. 1329–1339, 2008.
- [6] G. Fung and O. Mangasarian, "Multicategory proximal support vector machine classifiers," *Machine Learning*, vol. 59, no. 1, pp. 77–97, 2005.
- [7] S. Gottschalk, M. Lin, and D. Manocha, "Obbtrees: a hierarchical structure for rapid interference detection," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 171–180.
- [8] J. Chang, W. Wang, and M. Kim, "Efficient collision detection using a dual obb-sphere bounding volume hierarchy," *Computer-Aided Design*, vol. 42, no. 1, pp. 50–57, 2010.
- [9] S. Gottschalk, "Collision queries using oriented bounding boxes," Ph.D. dissertation, Citeseer, 2000.
- [10] A. Asuncion and D. Newman, "Uci machine learning repository [http://www.ics.uci.edu/~mllearn/mlrepository.html]. irvine, ca: University of california," *School of Information and Computer Science*, 2007.
- [11] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.