

UMA METODOLOGIA PARA GERAÇÃO AUTOMÁTICA DE REDES NEURAIAS NEBULOSAS A PARTIR DE ÁRVORES DE DECISÃO

ANDRÉ P. LEMOS*, RENATO D. MAIA†, MAURILIO J. INÁCIO‡, WALMIR M. CAMINHAS*

**Departamento de Engenharia Elétrica - Universidade Federal de Minas Gerais*

†*Departamento de Ciências da Computação - Universidade Estadual de Montes Claros/ Departamento de Engenharia de Controle e Automação - Faculdade de Ciência e Tecnologia de Montes Claros*

‡*Departamento de Engenharia da Computação - Faculdade de Ciência e Tecnologia de Montes Claros*

Emails: andrepl@cpdee.ufmg.br, renato.dourado@unimontes.br/renato@femc.edu.br,
maurilio@femc.edu.br, caminhas@cpdee.ufmg.br

Abstract— This work proposes a new methodology for developing fuzzy neural networks based on information extracted from decision trees for pattern classification problems. This approach was proposed in order to solve the main deficiency of some of these networks, the exponential growth of free parameters given the number of features. Experiments with datasets containing lots of features were performed in order to demonstrate the efficiency of the proposed methodology.

Keywords— Fuzzy Neural Networks, Decision Tree, Pattern Classification

Resumo— Este trabalho propõe uma nova metodologia para a construção de redes neurais nebulosas baseada em informações extraídas a partir de árvores de decisão para problemas de classificação de padrões. Essa abordagem foi proposta para resolver a principal deficiência de algumas dessas redes, o crescimento exponencial dos parâmetros livres em função do número de características. Experimentos são apresentados para comprovar a eficácia da metodologia em bases de dados contendo um número elevado de características.

Palavras-chave— Redes Neurais Nebulosas, Árvores de Decisão, Classificação de Padrões

1 Introdução

A utilização de modelagem baseada em dados vem sendo largamente utilizada em diversos campos da engenharia para construir modelos capazes de reproduzir resultados semelhantes aos de sistemas complexos, através de um conjunto de amostras. O principal objetivo desse tipo de modelagem é a construção de modelos com alta acurácia, reproduzindo resultados semelhantes ao sistema modelado, com o menor erro possível. Atualmente, outro objetivo importante vem sendo desejado: o desenvolvimento de modelos com estruturas transparentes, de forma que seja possível interpretar os resultados obtidos e/ou inserir conhecimento a priori na sua estrutura.

Um exemplo de modelos baseados em dados, com alto grau de adaptabilidade e transparência, são as redes neurais nebulosas, definidas como redes neurais compostas por neurônios nebulosos. Essas redes possuem como principal característica a possibilidade de extração de conhecimento em forma de regras, a partir de sua topologia, bem como a possibilidade de inserir conhecimento a priori. Essas redes são formadas por uma colaboração sinérgica entre as teorias de conjuntos nebulosos e redes neurais, permitindo uma vasta gama de habilidades de aprendizado, fornecendo assim modelos que integram o tratamento de informação incerta provido pelos sistemas nebulosos e a habilidade de aprendizado proporcionada pelas redes neurais (Gomide

and Pedrycz, 2007).

Porém, a principal deficiência de alguns desses modelos é dada pelo fato de realizarem uma partição do espaço de entrada através da combinação de conjuntos nebulosos construídos para cada variável (Caminhas et al., 1999; Hell, 2008). Cada combinação possível dos conjuntos define uma regra, que representa uma região do espaço de entrada e, geralmente, o número de parâmetros livres desses modelos é proporcional ao número de regras. Por exemplo, um modelo com 2 entradas e 3 conjuntos nebulosos por entrada gera $3^2 = 9$ regras. Caso o modelo possua 10 entradas, $3^{10} = 59049$ regras são geradas, tornando praticamente impossível o treinamento, dado o excessivo número de parâmetros livres a serem otimizados.

Diversas abordagens vêm sendo encontradas na literatura para resolver esse problema. Su and Chang (2000) utilizam conjuntos nebulosos multivariáveis para realizar a partição do espaço de entrada, reduzindo assim o número de regras geradas, mas reduzindo também a sua interpretabilidade. Ishibuchi et al. (1995) utilizam algoritmos genéticos para a seleção de um número limitado de regras a serem utilizadas diante de todas as possíveis combinações dos conjuntos nebulosos das entradas. Essa abordagem pode reduzir drasticamente o número de parâmetros do modelo, porém a um elevado custo computacional. Gobi and Pedrycz (2008) utilizam ferramentas de minimização de funções booleanas para diminuir o número de

conjuntos nebulosos de entrada e o número de regras.

Este trabalho propõe uma nova metodologia para a solução desse problema. Para isso, redes neurais nebulosas *and/or*, propostas por Caminhas et al. (1999), são modificadas, de modo que a sua topologia seja definida a partir de regras extraídas de árvores de decisão, para resolver problemas de classificação de padrões. Árvores geradas pelo algoritmo CART (Breiman et al., 1984) são utilizadas para selecionar características definidas como entradas da rede, definir os conjuntos nebulosos que realizam o particionamento do espaço de entrada e o número de neurônios *and* e *or*, assim como suas respectivas conexões. Uma vez definida a topologia, os parâmetros dessa rede são ajustados através de um algoritmo genético com codificação real, a fim de minimizar o erro de classificação, dado um conjunto de amostras.

2 Rede Neural Nebulosa *and/or*

A rede neural nebulosa *and/or*, proposta por Caminhas et al. (1999), tem como principal característica a representação de regras de classificação de forma explícita, possibilitando a extração de regras do tipo se-então a partir de sua topologia. Essa rede é formada por neurônios lógicos nebulosos de agregação do tipo *and* e *or* (Pedrycz, 2004). Neurônios lógicos nebulosos são unidades funcionais que realizam operações não-lineares multivariáveis entre hipercubos unitários $[0, 1]^n \rightarrow [0, 1]$.

O neurônio lógico nebuloso de agregação *or* é interpretado como uma expressão lógica que realiza uma agregação local do tipo *and* das entradas \mathbf{x} com pesos \mathbf{w} , através de uma *t-norma*, seguida de uma agregação global do tipo *or* dos resultados, realizada por uma *s-norma* (Pedrycz, 2004). O neurônio *and* é interpretado de maneira análoga, com a agregação local do tipo *or* e global do tipo *and*, tal como apresentado em (1), em que T e t representam *t-normas* e S e s *s-normas*.

$$\begin{aligned} y &= or(x, w) = S_{i=1}^N x_i t w_i \\ y &= and(x, w) = T_{i=1}^N x_i s w_i \end{aligned} \quad (1)$$

A partir desses neurônios, é possível extrair regras do tipo se-então, de forma que a operação de agregação local possa ser interpretada como uma ponderação das entradas. Assim, um neurônio do tipo *and* pode ser interpretado como uma regra do tipo: *se* $(x_1|w_1)$ *e* $(x_2|w_2)$ *e* \dots $(x_n|w_n)$ *então* y ; e um neurônio do tipo *or* como: *se* $(x_1|w_1)$ *ou* $(x_2|w_2)$ *ou* \dots $(x_n|w_n)$ *então* y , sendo $|$ o operador de ponderação, de forma que a expressão $(x_n|w_n)$ é in-

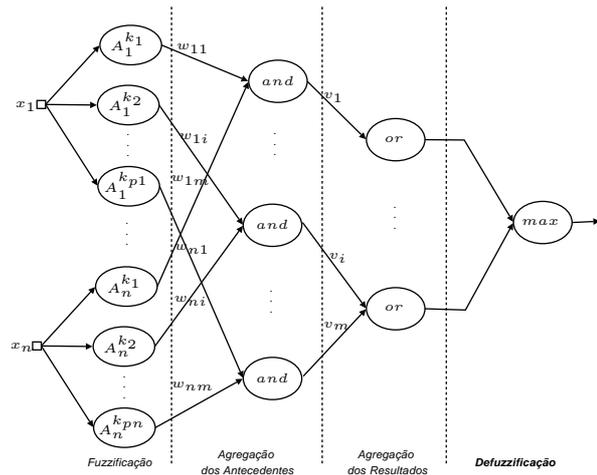


Figura 1: Rede neural nebulosa *and/or*

terpretada como a ponderação da entrada x_n pelo peso w_n .

A rede neural nebulosa *and/or* é formada por quatro camadas distintas. A primeira camada, denominada *fuzzificação*, é responsável pela granularização das entradas através de conjuntos nebulosos. Em seguida, a camada de *agregação dos antecedentes* realiza a agregação dos resultados da granularização através de neurônios nebulosos *and*, gerando regras que realizam a partição do espaço de entrada, atribuindo uma das possíveis classes a cada região gerada. Essas regras podem ser interpretadas como: *se* $(x_1 \text{ é } A_1^{k_1}|w_{j_1}) \dots \text{ e } (x_i \text{ é } A_i^{k_i}|w_{j_i}) \dots \text{ e } (x_n \text{ é } A_n^{k_n}|w_{j_n})$ *então* x *pertence a classe* l *com peso de conexão* v_j . A camada de *agregação dos resultados* é formada por n neurônios do tipo *or*, sendo n o número de classes. Essa camada realiza a agregação das regras pertencentes à mesma classe, ponderando-as pelos pesos de conexão. Dessa forma, cada neurônio do tipo *or* representa a agregação de um conjunto de regras de classificação dos padrões de entrada para uma determinada classe. O resultado dessa agregação é o grau de certeza de que uma dada amostra pertence a uma determinada classe. Finalmente, a camada de *defuzzificação* define a saída final do sistema. A classe final da amostra é definida como a classe com maior grau de certeza, calculada através da função máximo. A figura 1 ilustra essa rede.

Uma vez que cada neurônio *and* representa uma regra de classificação para uma determinada classe, e essas regras são agregadas por um neurônio *or*, o resultado final dessa rede é um conjunto de regras nebulosas de classificação para cada uma das classes do problema tratado. Essa alta transparência dos resultados é a principal característica dessas redes.

3 Metodologia Proposta

A transparência dessa rede facilita a introdução de conhecimento a priori sobre o problema. Esse fato motivou a utilização de árvores de decisão para a definição de sua topologia, com o objetivo de torná-la viável para classificação de padrões em problemas com um elevado número de características.

A abordagem proposta neste trabalho utiliza regras de classificação extraídas a partir de árvores de decisão, construídas pelo algoritmo CART, para a definição da topologia da rede. As entradas são as características selecionadas durante construção da árvore de decisão. O número de conjuntos nebulosos por variável de entrada é definido como o número de pontos de corte presentes na árvore para a variável em questão mais um. Cada conjunto possui função de pertinência trapezoidal, de forma que funções adjacentes podem ser sobrepostas, cruzando nos valores dos pontos de corte definidos pela árvore para a variável, com valor de pertinência igual a 0.5. Finalmente, as regras extraídas da árvore são utilizadas para definir o número de neurônios *and* e *or* da rede, assim como as suas respectivas conexões.

Para esclarecer a metodologia proposta, segue um exemplo de sua aplicação. Os dados utilizados neste exemplo são provenientes da base de dados *Iris*, do repositório de aprendizado de máquina UCI (<http://ics.uci.edu>), comumente utilizados na literatura para avaliar metodologias de classificação de padrões. Essa base de dados possui 4 características e 3 possíveis classes.

Inicialmente, uma árvore de decisão é construída a partir dos dados, utilizando-se o algoritmo CART. A figura 2 ilustra a árvore obtida. Analisando-se essa figura, percebe-se que apenas 2 das 4 características foram selecionadas para realizar a classificação. Além disso, a variável x_3 possui dois pontos de corte, $[2.45 \ 4.95]$, e x_4 apenas um, 1.75. Dessa forma, as variáveis de entrada da rede são definidas como essas duas variáveis. Além disso, x_3 possuirá 3 conjuntos nebulosos, e x_4 , 2. A figura 3 ilustra os conjuntos gerados para cada uma das entradas. Analisando-se essa figura, percebe-se que esses conjuntos possuem funções de pertinência trapezoidais, e que funções adjacentes sempre se cruzam nos pontos de corte com valor de pertinência igual a 0.5.

Uma vez definidas as funções de pertinência, regras de classificação extraídas da árvore são utilizadas para definir o número de neurônios *and* e *or* e suas respectivas conexões. As seguintes regras são extraídas da árvore ilustrada pela figura 2:

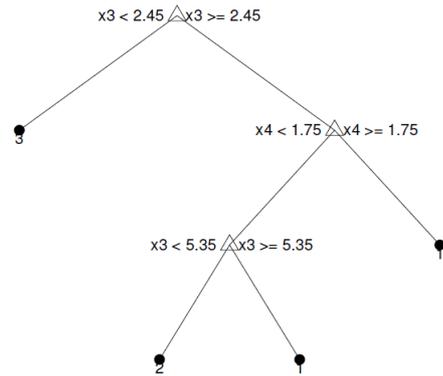


Figura 2: Árvore de decisão gerada a partir de dados da base *Iris*

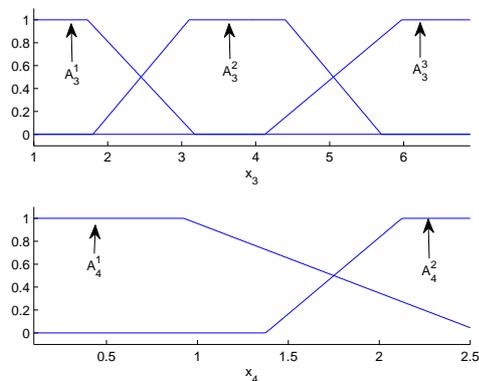


Figura 3: Funções de pertinência geradas a partir da árvore de decisão ilustrada na figura 2

- R_1 : Se $x_3 < 2.45$ então y é 3
- R_2 : Se $x_3 > 2.45$ e $x_4 > 1.75$ então y é 1
- R_3 : Se $2.45 < x_3 < 5.35$ e $x_4 < 1.75$ então y é 2
- R_4 : Se $2.45 < x_3 < 4.95$ e $x_4 > 1.75$ então y é 1

Cada regra é utilizada para definir um neurônio do tipo *and*, suas conexões com os conjuntos nebulosos da entrada, e sua conexão com o neurônio *or* relativo à classe definida no consequente da regra. Porém, algumas regras possuem condições que representam mais de um conjunto nebuloso de uma dada entrada, por exemplo, a condição $x_3 > 2.45$ (segunda regra) é representada pelos conjuntos nebulosos A_3^2 e A_3^3 . Assim, uma nova camada é definida, formada por neurônios do tipo *or*, com pesos de conexão fixos e iguais a 1 representando a agregação de conjuntos nebulosos das entradas, para que estes possam ser mapeados pelas regras extraídas, de forma análoga à metodologia proposta por (Gobi and Pedrycz, 2008). Dessa forma, a condição $x_3 > 2.45$ é representada pela agregação lógica nebulosa A_3^2 *or* A_3^3 . Além disso, algumas regras não necessitam de neurônios *and*, uma vez que seus antecedentes são formados por condições de apenas uma variável de entrada, como por exemplo a primeira regra. Assim, essas

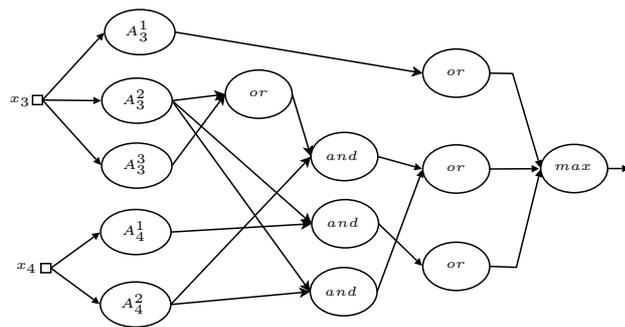


Figura 4: Estrutura da rede *and/or* gerada a partir da árvore de decisão ilustrada na figura 2

regras conectam diretamente conjuntos nebulosos de entrada a neurônios *or*. A topologia final da rede gerada pelas regras extraídas é ilustrada na figura 4.

Uma vez definida a topologia da rede, um algoritmo genético de codificação real é utilizado para ajustar seus parâmetros livres. Cada indivíduo da população é constituído pelos pesos dos neurônios *and* e *or* e pelos parâmetros das funções de pertinência. O algoritmo utiliza como função objetivo a ser minimizada o erro de classificação do conjunto de treinamento.

Embora as funções de pertinência possuam 4 parâmetros, conforme definido em (2), apenas 2 são definidos como livres e sujeitos a otimização. Optou-se por essa abordagem para se manter o ponto de cruzamento dessas funções como os pontos de corte extraídos da árvore de decisão. Dessa maneira, apenas os parâmetros *b* e *c* são otimizados, e os parâmetros *a* e *d* são calculados, dado que *b* e *c* devem possuir valores de pertinência unitários, e os pontos de corte adjacentes valores de pertinência iguais a 0.5.

$$f(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \geq d \end{cases} \quad (2)$$

Para cada função de pertinência, foram definidos valores máximos e mínimos para seus parâmetros livres. Cada uma dessas funções é definida entre dois pontos de corte extraídos da árvore de decisão. Dessa forma, calcula-se o valor médio desses dois pontos de corte e define-se esse valor como o valor máximo do parâmetro *b* e mínimo do parâmetro *c*. Em seguida, o ponto de corte inferior ao ponto médio é definido como o valor mínimo do parâmetro *b*, e o ponto de corte superior como o valor máximo do parâmetro *c*. Vale ressaltar que a primeira e a última funções de pertinência de cada

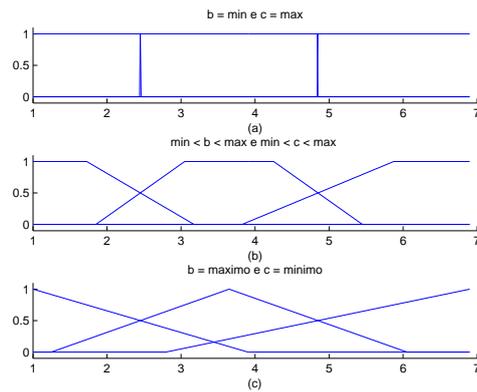


Figura 5: Três formatos para uma função de pertinência, dados valores distintos dos seus parâmetros livres

entrada possuem apenas um parâmetro: a primeira possui apenas o parâmetro *c*, e a última apenas o *d*.

A figura 5 ilustra três possíveis formatos para as funções de pertinência, dados os parâmetros *b* e *c*. A figura 5a ilustra o caso em que *b* assume valor mínimo e *c* valor máximo: nesse caso não há sobreposição entre funções de pertinência adjacentes. A figura 5b ilustra um caso em que *b* e *c* assumem valores entre o mínimo e máximo, definindo regiões de sobreposição entre funções adjacentes. Finalmente, a figura 5c ilustra o caso em que *b* e *c* assumem valores máximo e mínimo, respectivamente. Essa configuração faz com que as funções de pertinência tornem-se triangulares e a sobreposição entre funções adjacentes seja máxima.

Esses valores limítrofes dos parâmetros das funções de pertinência são utilizados como restrições do algoritmo genético. As demais variáveis a serem otimizadas, os pesos dos neurônios constituintes da rede, restringem-se ao intervalo unitário.

4 Experimentos

Foram realizados experimentos para avaliar o desempenho da metodologia proposta. Para isso foram utilizadas bases de dados de problemas de classificação do repositório de aprendizado de máquina UCI. Foram utilizadas as bases *Iris*, *Statlog Heart* e *Ionosphere*. Essas bases representam problemas com diferentes níveis de complexidade. A base *Statlog Heart* possui 13 características e 2 classes. Já a base *Ionosphere* possui 34 características e 2 classes.

Utilizou-se um algoritmo genético de codificação real com uma população de 50 indivíduos, seleção por torneio e elitismo simples, durante 50 gera-

Tabela 1: Resultados comparativos dos experimentos realizados

Metodologia	<i>Iris</i>	<i>Ionosphere</i>	<i>Heart</i>
Árvore de Decisão	95.33%	88.90%	78.89%
(Tsipouras et al., 2008)	96.00%	89.74%	81.11%
Rede Proposta	96.00%	89.18%	82.59%

ções. Os parâmetros do algoritmo foram ajustados durante experimentos preliminares. A população inicial do algoritmo genético foi gerada da seguinte maneira: os pesos dos neurônios *and* e *or* foram inicializados aleatoriamente para cada indivíduo; já os parâmetros das funções de pertinência foram inicializados aleatoriamente, dentro dos seus respectivos intervalos. Utilizou-se o produto algébrico como *t-norma* e a soma probabilística como *s-norma*.

Para todos os experimentos, utilizou-se validação cruzada para a avaliação de desempenho. Para isso, o conjunto de dados foi dividido em 10 subconjuntos, sendo 9 utilizados para treinar a metodologia proposta e 1 para validá-la. Esse procedimento é realizado 10 vezes, e a acurácia final do modelo é definida como a acurácia média. Os resultados foram comparados com os resultados obtidos pela árvore de decisão e com outro método encontrado na literatura (Tsipouras et al., 2008), que também realiza a classificação com base em regras nebulosas construídas a partir de árvores de decisão.

A tabela 1 ilustra os resultados obtidos. Analisando-se essa tabela, percebe-se que a metodologia proposta tem desempenho superior ao da árvore de decisão em todos os experimentos, e equivalentes ao de Tsipouras et al. (2008).

Outro ponto importante a ser avaliado é o número de regras gerado para cada problema, dado que um número elevado de regras dificulta a interpretabilidade dos resultados. Para as bases *Iris* e *Heart Statlog* foram geradas em média 6 regras. Já para a base *Ionosphere*, 3 regras foram geradas, sendo que o problema contempla 34 características.

5 Conclusões

Este trabalho propôs uma nova metodologia de construção de redes neurais nebulosas baseada em regras extraídas da estrutura de árvores de decisão. A metodologia descrita propõe uma nova topologia de rede, baseada na rede *and/or*, mas com uma nova camada de agregação das funções de pertinência das entradas. Essa nova topologia foi proposta como solução para o crescimento exponencial de parâmetros livres da rede *and/or* em função do número de características do problema e foi avaliada

a partir de bases de dados comumente utilizadas na literatura. Os resultados obtidos sugerem que essa metodologia é uma alternativa promissora, principalmente para problemas com um elevado número de características.

Agradecimentos

Os autores agradecem à Fapemig e ao CNPq pelo apoio.

Referências

- Breiman, L. and Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and regression trees*, Wadsworth Mathematics Series, San Diego, California.
- Caminhas, W., Tavares, H., Gomide, F. and Pedrycz, W. (1999). Fuzzy sets based neural networks: Structure, learning and applications, *Journal of Advanced Computational Intelligence* **3**(3): 151–157.
- Gobi, A. E. and Pedrycz, W. (2008). Logic minimization as an efficient means of fuzzy structure discovery, *IEEE Transactions on Fuzzy Systems* **16**(3): 553–566.
- Gomide, F. and Pedrycz, W. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*, Wiley Interscience, NJ, USA.
- Hell, M. (2008). *Abordagem Neurofuzzy para Modelagem de Sistemas Dinamicos Nao Lineares*, PhD thesis, Universidade Estadual de Campinas.
- Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H. (1995). Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Transactions on Fuzzy Systems* **3**(3): 260–270.
- Pedrycz, W. (2004). Heterogeneous fuzzy logic networks: Fundamentals and development studies, *IEEE Transactions on Neural Networks* **15**(6): 1466–1481.
- Su, M.-C. and Chang, H.-T. (2000). Application of neural networks incorporated with real-valued genetic algorithms in knowledge acquisition, *Fuzzy Sets and Systems* **112**(1): 85 – 97.
- Tsipouras, M. G., Exarchos, T. P. and Fotiadis, D. I. (2008). A methodology for automated fuzzy model generation, *Fuzzy Sets and Systems* **159**(23): 3201–3220.