

# Neuro-Fuzzy Network com taxa de aprendizagem ótima

Inteligência Computacional

Sávio Rodrigues

Design de gestão e informática

CEFET-MG - Campus V

Divinópolis, Brasil

saviorrodrigues012@gmail.com

**Resumo**—O presente trabalho visa a implementação de uma rede Neuro Fuzzy com taxa de aprendizagem ótima.

**Palavras-chave**—NFN, neuro fuzzy network, fuzzy

## I. INTRODUÇÃO

Um Sistema Neuro-Fuzzy é um tipo de sistema híbrido incorporado constituído pela combinação de duas técnicas de modelagem, as redes neurais artificiais e a Lógica Fuzzy. A ideia básica é a construção de um sistema de inferência Fuzzy, numa estrutura paralela distribuída de tal forma que os algoritmos de aprendizado das redes neurais possam ser aproveitados neste sistema híbrido para ajustar os parâmetros.

## II. IMPLEMENTAÇÃO

A Figura 1 mostra a representação ilustrativa da rede testada com a definição de 3 funções de pertinência complementares e pesos que serão ajustados para cada uma. Nela, é possível identificar as operações feitas com o valor de pertinência e os pesos para gerar uma saída.

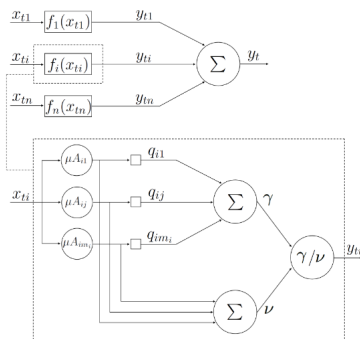


Fig. 1. Rede Neuro Fuzzy

Inicialmente foram lidos os valores de entrada  $x_1$ ,  $x_2$  e  $x_3$  e um valor esperado  $y_1$ . Em seguida, foram definidas as 3 funções de pertinência com centro em 0, 0.5 e 1 respectivamente.

As variáveis iniciais são:

- **weights:** Os pesos são valores aleatórios que variam de  $-0.5$  a  $0.5$ . Para cada função de pertinência possui um peso associado, ou seja, em toda a rede é um total 9 pesos.
- **$y_i$ :** É declarado o vetor que armazenará os valores preditos.
- **segments:** Vetor para armazenar o index da função com maior ativação.
- **activation:** Vetor para armazenar o valor de ativação da função.
- **alpha:** Vetor para armazenar o valor do alpha, responsável por ajustar os pesos.

O valor de saída  $y_{ti}$  é definido pela seguinte formula

$$y_i = \frac{a}{b} = \frac{\sum \mu_{A_{ij}} q_{ij}}{\sum \mu_{A_{ij}}} \quad (1)$$

Por se tratar de funções complementares o valor de  $b$  sempre será em uma unidade, uma vez que a soma das ativações das duas funções de pertinência sempre será 1. Após o cálculo da saída, é feito o reajuste dos pesos o qual é realizado a partir da regra delta definida pela seguinte fórmula:

$$q_{ij} = q_{ij} + \alpha \mu_{A_{ij}} (Y_p - Y_E) \quad (2)$$

em que  $y_p$  é o valor predito e  $y_e$  o valor esperado.

$\alpha$  é definido da seguinte forma:

$$\alpha = \frac{1}{\sum \mu_{A_{ij}}^2} \quad (3)$$

Todos esses passo é feito para todos os pontos como segue o pseudocódigo ilustrado na Figura 2.

## III. RESULTADOS

Ao executar os algoritmo em 30% das amostras da base dados fornecida, para mais fácil visualização dos resultados, obteve-se o gráfico ilustrado na Figura 3.

**Entrada:**  $x_t, \hat{y}_t, m$ ;  
**Saída:**  $y_t$ ;  
 Inicializar  $b_{ij}$ ;  
**for**  $t = 1, 2, \dots$  **do**  
     Ler  $x_t, \hat{y}_t$ ;  
     **for**  $i = 1 : n$  **do**  
         Calcular  $\mu_{A_{ik_i}}(x_{ti}), \mu_{A_{ik_i+1}}(x_{ti})$ ;  
         Calcular  $y_{ti}$ ;  
     **end for**  
     Calcular  $y_t$ ;  
     Calcular  $\alpha$ ;  
     Atualizar  $q_{ik_i}, q_{ik_i+1}$ ;

Fig. 2. Rede Neuro Fuzzy

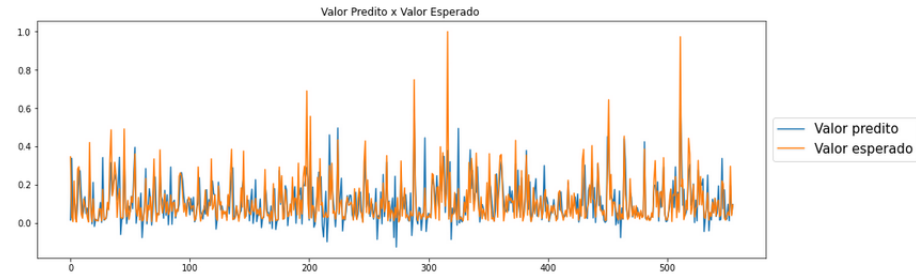


Fig. 3. Rede Neuro Fuzzy

#### IV. CONCLUSÃO

O algoritmo do sistema Neuro-Fuzzy com taxa de aprendizagem ótima se mostrou bastante eficiente para prever os resultados pelo reajuste de pesos. Pela análise gráfica, é possível perceber que a margem de erro se mostra pequena e em muito casos, se aproximando do valor real.