UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Sistemas Nebulosos

Adaptive Neuro-Fuzzy Inference System (ANFIS)

Cristiano Leite de Castro

Adaptação de material didático do Prof. André Paim Lemos (DELT)

Modelagem utilizando Sistemas Nebulosos

- Sistemas nebulosos podem ser utilizados em tarefas de
  - Aproximação funcional
  - Controle de processos
  - Identificação de sistemas
  - Previsão de séries temporais
  - Classificação de padrões
  - Entre outros.

Sistemas Nebulosos Adaptativos

- Evolução das Técnicas de Modelagem
  - Primórdios:
    - definição do modelo a partir de conhecimento extraído de especialistas;
  - A partir da década de 90:
    - regras iniciais são geradas a partir do conhecimento de especialistas e/ou a partir de dados;
    - os parâmetros do modelo são ajustados a partir de dados;
  - Tendência atual
    - Modelos evolutivos;

Sistemas Nebulosos Adaptativos

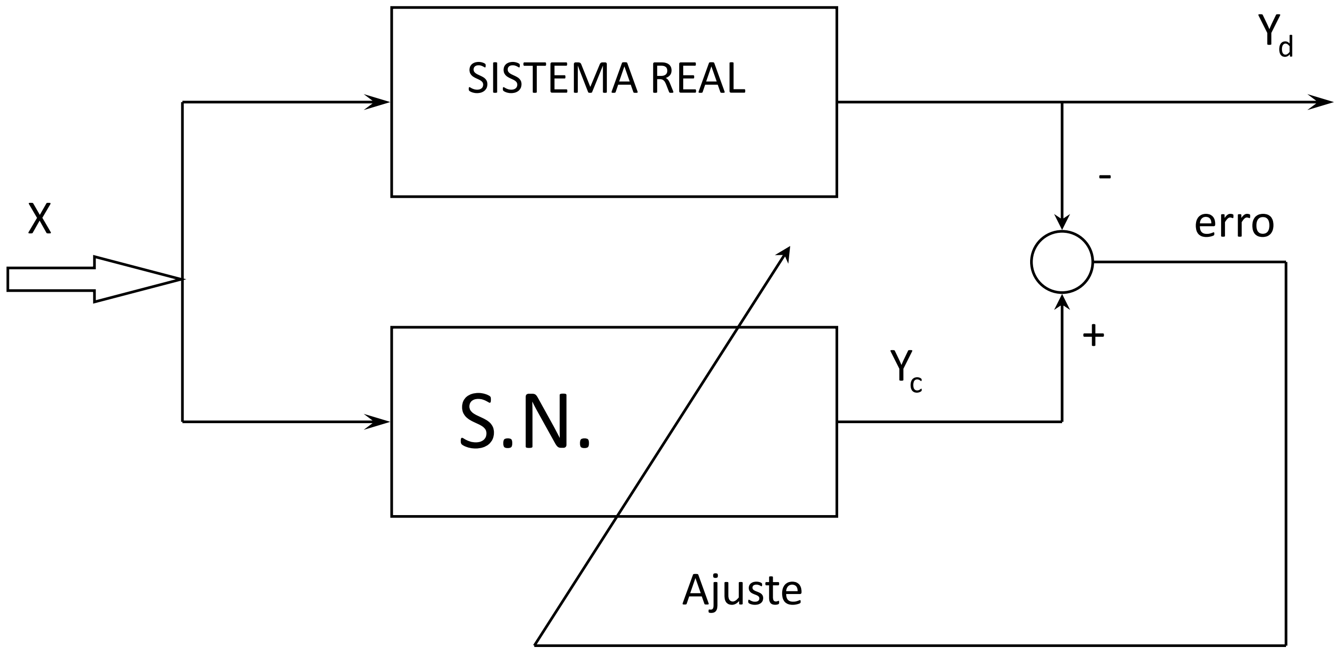
- Modelagem Atual:
  - pode-se ou não utilizar conhecimento de especialistas para definição da topologia inicial do modelo;
  - Caso não utilize:
    - topologia é definida a partir de dados de treinamento, utilizando, por exemplo, algoritmos de agrupamento;
    - parâmetros são ajustados utilizando algoritmos de otimização;
      - Exemplo: **Algoritmo do Gradiente Descedente**

Modelagem utilizando Sistemas Nebulosos

- Etapas do processo de modelagem:
  1. Definição das regras
  2. Escolha dos operadores
  3. Ajuste dos parâmetros

- Solução:
  – *Sistemas Nebulosos Adaptativos com parâmetros ajustados via gradiente descendente;*

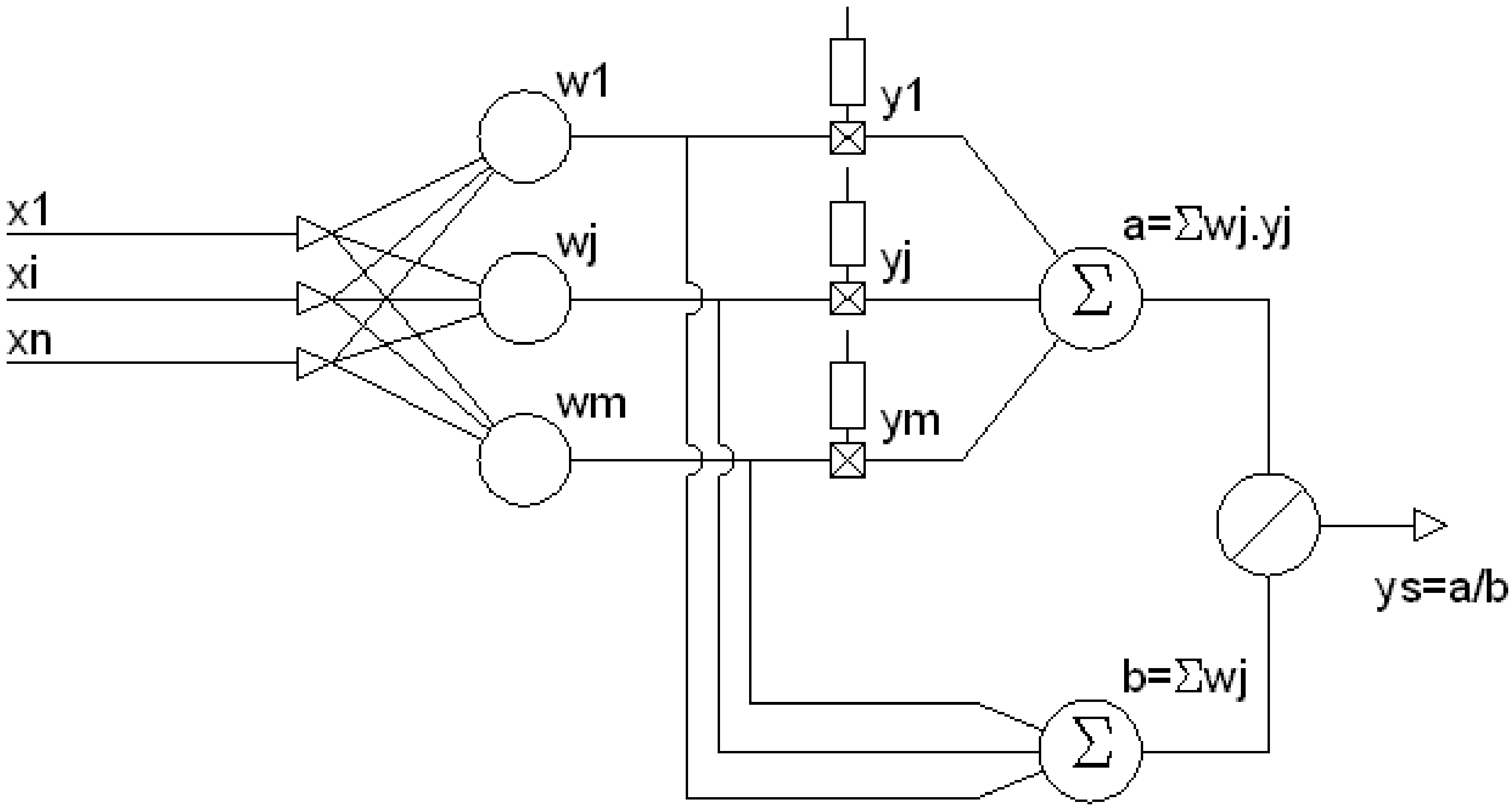
SISTEMAS NEBULOSOS ADAPTATIVOS



PARÂMETROS DE AJUSTE

- funções de pertinência
- parâmetros dos modelos de saída
- base de regras

Estrutura da rede Neurofuzzy vista em sala de aula



Regras implementadas

Se $(x_1$ é $A_{1j})$ e $(x_2$ é $A_{2j})..$e $(x_i$ é $A_{ij})..$e $(x_n$ é $A_{nj})$

Então $y_j = P_{1j}x_1 + P_{2j}x_2 + \ldots + P_{ij}x_i + P_{nj}x_n + q_j$

$$y_j = \sum_{i=1}^{n} P_{ij} . x_i + q_j$$

$$w_j = \prod_{i=1}^{n} \mu_{Aij}(x_i)$$

$$w_j = \mu_{A1j}(x_1).\ \mu_{A2j}(x_2).\ \ldots\ .\mu_{Aij}(x_j).\ \mu_{Anj}(x_n)$$

$$\mu_{Aij}(x_i) = \exp\left[\frac{-1}{2}\left(\frac{x_i - c_{ij}}{\sigma_{ij}}\right)\right]^2$$

$$ys = \frac{\sum_{j=1}^{m} w_j . y_j}{\sum_{j=1}^{m} w_j} = \frac{a}{b}$$

$$\min\ e = \tfrac{1}{2}\left(y_s - y_d\right)^2$$

$$e = f\left(x, c_{ij}, \sigma_{ij}, p_{ij}, q_j\right)$$

Gradiente Descendente

- Regra de aprendizagem *Widrow-Hoff:*

$$w_j = w_j - \alpha(h_w(x^{(i)}) - y^{(i)})x_j^{(i)}$$

  – também conhecida como método dos mínimos quadrados (*Least Mean Squares – LMS*)

Algoritmo Incremental (ou Estocástico)

Repita até convergir

{

    Para I = 1…m // para todos os padrões de treinamento

    {

        Para j = 1…n // para todos os pesos

        {

$$w_j = w_j - \alpha(h_w(x^{(i)}) - y^{(i)})x_j^{(i)}$$

        }

    }

}

Algoritmo

- Exemplo – MatLab.

Leitura Recomendada

- Capítulo 12 do Livro:
  – Jyh-Shing Roger Jang and Chuen-Tsai Sun. 1996. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.