

## 4. Sistemas Neuro-Fuzzy

Neste capítulo será apresentado o sistema híbrido Neuro-Fuzzy, o qual é a combinação das técnicas de fuzzy e redes neurais. Alguns modelos Neuro-Fuzzy já desenvolvidos na literatura são descritos.

### 4.1. Introdução

Neste capítulo é apresentada a fundamentação de conceitos para a implementação de sistemas híbridos, tais como o sistema Neuro-Fuzzy. Nas últimas décadas, como uma alternativa aos métodos convencionais de modelagem do controle, surgiram o projeto de controle baseado em *Lógica Fuzzy (FL)* e *Redes Neurais Artificiais (ANN)*. Estas duas técnicas são aplicadas com sucesso em diversas áreas nas quais o controle convencional tem falhado na modelagem do controlador.

Os sistemas fuzzy são apropriados para a modelagem de controladores a partir do conhecimento explícito fornecido pelo especialista, nos quais o desempenho do controlador depende da experiência do especialista. E as *ANN* são adequadas na criação do controle baseado no conhecimento implícito embutido em um conjunto de dados, entretanto o desempenho das redes neurais é afetado pelo ajuste de seus parâmetros (número de neurônios em cada camada, número de camadas escondidas, etc.). Portanto, muitos pesquisadores têm tentando integrar estas duas técnicas para gerar um modelo híbrido que possa aproveitar as vantagens de cada uma delas e minimizar suas deficiências. Nos últimos anos, diversos tipos de sistemas híbridos foram desenvolvidos na literatura, geralmente pela integração das técnicas de *Algoritmos Genéticos (GA)*, *ANN* e *FL*.

Neste trabalho, apresentam-se diversas arquiteturas híbridas baseadas na integração de *Sistemas Fuzzy* e *ANN*.

## 4.2. Sistemas Híbridos

Os sistemas híbridos são a sinergia obtida pela combinação de duas ou mais técnicas de modelagem. O foco destes sistemas está em obter um sistema mais poderoso e com menos deficiências. Dependendo da forma básica de construção, uma técnica pode ser aplicada para melhorar as deficiências do outro em um maior ou menor grau. Algumas destas formas são descritas a seguir.

### 4.2.1. Sistema Híbrido Seqüencial

Em um sistema híbrido seqüencial, a saída de um subsistema com “*Técnica 1*” atua como entrada de outro subsistema com “*Técnica 2*”, como apresentado na Figura 4.1.

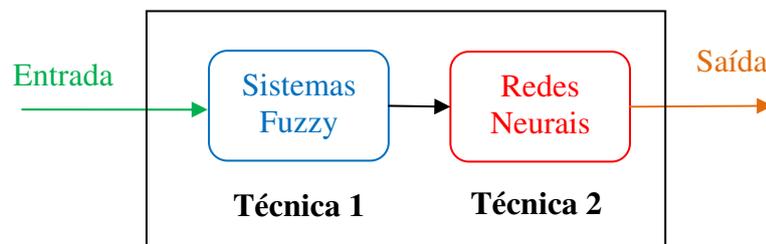


Figura 4.1. Sistema Híbrido Seqüencial.

Na figura anterior, mostra-se um sistema híbrido seqüencial constituído de duas técnicas ou subsistemas: a primeira técnica é um sistema fuzzy que trabalha como um pré-processador fuzzy dos dados, e a segunda técnica é uma *ANN*. Na literatura, freqüentemente este tipo não é considerado como sistema híbrido, por ser a forma mais fraca de hibridização.

### 4.2.2. Sistema Híbrido Auxiliar

Um sistema híbrido auxiliar, é constituído de dois subsistemas um principal e outro auxiliar. O subsistema principal com “*Técnica 1*” chama ao subsistema auxiliar com “*Técnica 2*” para realizar uma determinada tarefa.

No exemplo da Figura 4.2, o subsistema principal baseado em uma *ANN*, invoca um subsistema auxiliar baseado em um algoritmo genético para a otimização de seus pesos. Este tipo de sistema tem um maior grau de hibridização que o sistema híbrido seqüencial.

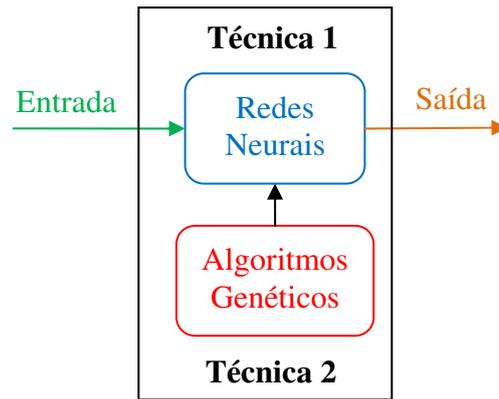


Figura 4.2. Sistema Híbrido Auxiliar.

#### 4.2.3. Sistema Híbrido Incorporado

Na prática, os sistemas híbridos incorporados são os mais utilizados, pois apresentam o maior grau de hibridização, ao ponto que não é possível a separação entre os dois subsistemas. Nestes sistemas, o grau de hibridização é tão elevado que se pode dizer que o primeiro subsistema contém o segundo, ou vice-versa.

Na Figura 4.3, apresenta-se um exemplo de um sistema híbrido incorporado, o qual é constituído de um *Sistema Fuzzy* e uma *ANN*. Estes tipos de sistemas híbridos são conhecidos como Sistemas Neuro-Fuzzy, nos quais um sistema de inferência fuzzy é implementado conforme a estrutura paralela de uma *ANN*.

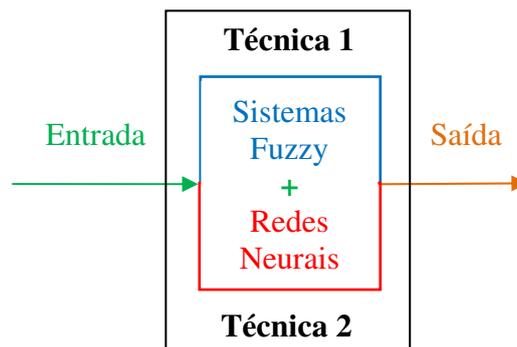


Figura 4.3. Sistema Híbrido Incorporado.

Nesta dissertação, utiliza-se o sistema híbrido incorporado Neuro-Fuzzy, pois este sistema tem recebido uma grande atenção dos pesquisadores em aplicações na área de controle, pela necessidade de controladores cada vez mais eficientes.

### 4.3. Sistemas Neuro-Fuzzy

Um *Sistema Neuro-Fuzzy (SNF)*, é um tipo de sistema híbrido incorporado constituído pela combinação de duas técnicas de modelagem muito conhecidas como as *ANN* e a *FL*. Na atualidade, os *SNF* estão se tornando de grande interesse, pois trazem os benefícios tanto de *ANN* quanto de sistemas de *FL*, removendo assim as desvantagens individuais ao combinar as características comuns. Além disso, diferentes arquiteturas de *SNF* vêm sendo pesquisadas em diversas áreas de aplicação, especialmente no controle de processos [23].

A idéia básica de um *SNF* é a construção de um *Sistema de Inferência Fuzzy (FIS)*, numa estrutura paralela distribuída de tal forma que os algoritmos de aprendizado das redes neurais possam ser aproveitados nestes sistemas híbridos para ajustar os parâmetros do *FIS*. A Figura 4.4 apresenta a estrutura de um *SNF* que é dividido em 5 camadas.

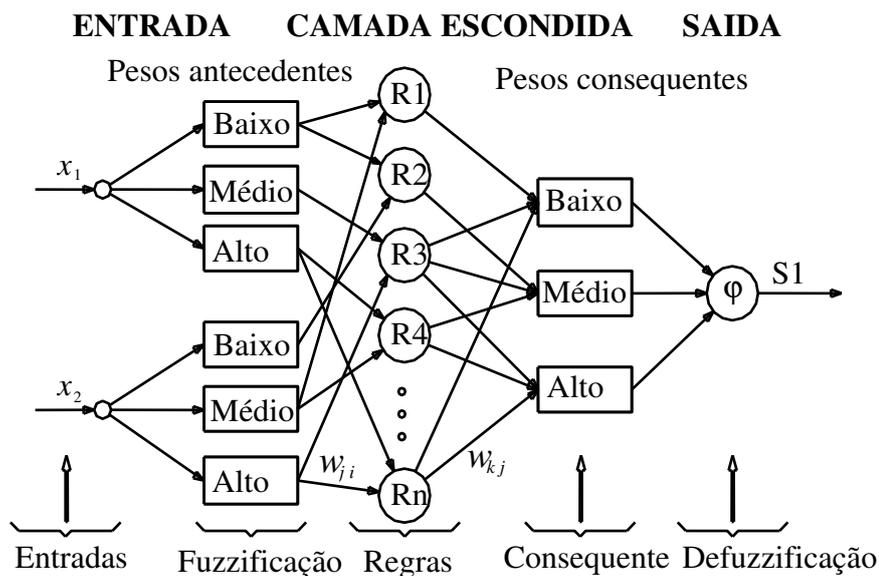


Figura 4.4. Arquitetura básica de um Sistema Neuro-Fuzzy.

A camada de entrada representa as variáveis de entrada, as quais são normalizadas e escalonadas dentro do intervalo numérico  $[0,1]$  ou  $[-1,1]$ . A segunda camada é a etapa de fuzzificação. Nesta etapa, os intervalos de cada variável de entrada são divididos em diversos níveis (Baixo, Médio e Alto), os quais indicam os pesos da rede para cada entrada. A terceira camada é definida

pelas regras do *FIS*, a camada 4 é determinada pelos conseqüentes das regras, e a camada 5 ou camada de saída é a etapa de defuzzificação, onde se calcula o valor numérico de saída.

### 4.3.1. Características de Sistemas Neuro-Fuzzy

Os *SNF* permitem a extração de conhecimento baseada na forma de regras de inferência fuzzy, mediante a integração do conhecimento explícito gerado pela experiência do especialista e do conhecimento implícito obtido a partir de um conjunto de dados. Assim, estes sistemas associam a capacidade de aprendizado e de tolerância a falhas das *ANN*, com a interpretabilidade dos *FIS* 's.

Devido à sua natureza dual, estes sistemas herdam as características de seus “genitores”. Neste sentido, dividiram-se as características em duas categorias principais, como se apresenta na Figura 4.5.

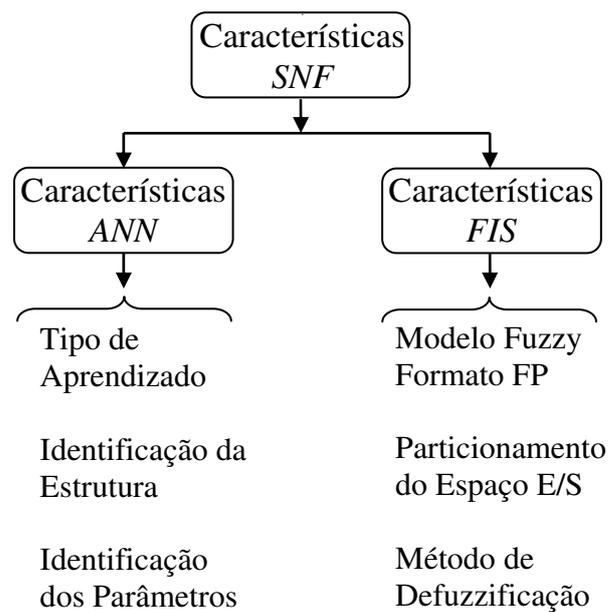


Figura 4.5. Características do Sistema Neuro-Fuzzy.

#### 4.3.1.1. Características Fuzzy do *SNF*

As características fuzzy do *SNF* são agrupadas em 4 sub-classes, classificadas em relação aos parâmetros da estrutura fuzzy, de modo que a combinação destas características determina o modelo do *SNF*. Na Figura 4.6,

apresenta-se esta classificação.

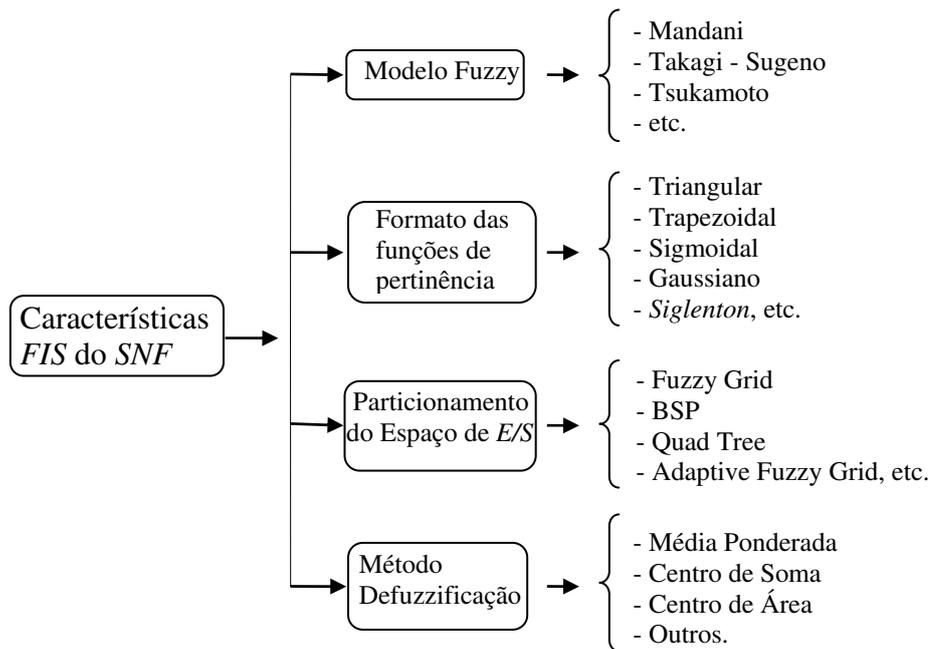


Figura 4.6. Características fuzzy do Sistema Neuro-Fuzzy.

O modelo fuzzy implementado no *SNF* determina o formato das regras fuzzy (modelo de inferência), o qual forma a parte fundamental da estrutura de conhecimento em um sistema de inferência fuzzy. Entretanto, os formatos das funções de pertinência influenciam nos graus de pertinência associados a cada variável, sendo o formato triangular e o trapezoidal os mais usados pela vantagem de serem computacionalmente simples. Assim, as funções de pertinência “*singleton*” geralmente são utilizadas nos consequentes dos sistemas híbridos, isto devido à vantagem de simplificar o processo de defuzzificação do sistema fuzzy.

O particionamento do espaço das variáveis de entrada e saída (*E/S*) define internamente regiões fuzzy de espaço, os quais são relacionados através das regras fuzzy. Assim, o particionamento do espaço de saída geralmente é mais simples pois está associado aos consequentes das regras.

Depois de fazer as avaliações do conjunto de regras fuzzy, o valor real da saída do *SNF* é determinado pelo método de defuzzificação escolhido.

#### 4.3.1.2. Características Neurais do SNF

As características neurais do *SNF* estão relacionadas com a capacidade de

aprendizado do sistema híbrido, estando divididas em 3 sub-classes como apresentado na Figura 4.7.

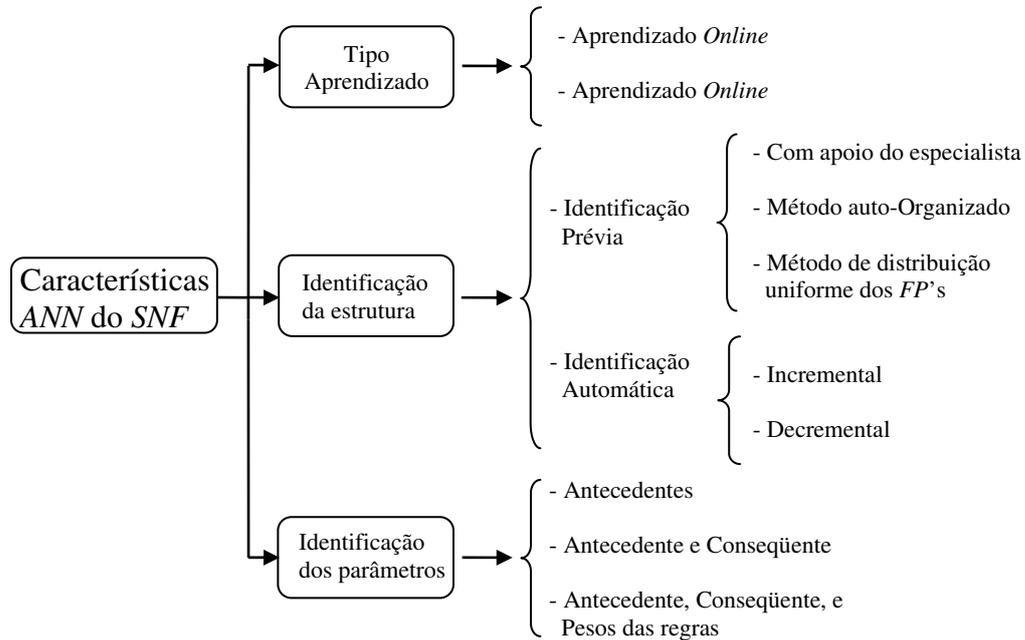


Figura 4.7. Características RNA do Sistema Neuro-Fuzzy.

O tipo de aprendizado, em relação à forma de apresentação dos padrões, divide-se em aprendizado *Off-line* e aprendizado *On-line*. No aprendizado *On-line*, a atualização dos parâmetros da estrutura é feita para cada padrão apresentado. Por outro lado, no aprendizado *Off-line* a atualização dos parâmetros é feita após a apresentação de todos os dados do conjunto de treinamento.

A identificação da estrutura de um *SNF* está relacionada com a determinação do número adequado de regras fuzzy, e de um particionamento satisfatório das E/S. Esta característica divide-se em duas categorias: *Identificação Prévia* e *Identificação Automática*. Na *Identificação Prévia*, o conhecimento da estrutura do sistema é feito *a priori* com apoio de algum especialista ou método auto-organizado, antes do início da atualização dos parâmetros. Na *Identificação Automática*, o conhecimento é obtido de forma incremental ou decremental, de modo que o sistema não precisa de conhecimento prévio.

As características do *SNF* em relação à identificação dos parâmetros são baseadas no método utilizado para o ajuste dos pesos fuzzy que definem os perfis

das *FP*'s dos antecedentes e conseqüentes das regras fuzzy. Geralmente, a maioria dos SNF apresenta identificação dos antecedentes e conseqüentes, com peso fixo de cada regra e de valor unitário. Um exemplo deste tipo de SNF são os modelos ANFIS (*Adaptative Network Based Fuzzy Inference System*). Entretanto, alguns SNF só apresentam ajuste nos antecedentes das regras fuzzy; um exemplo típico é o modelo NEFCLASS (*Neuro Fuzzy Classification*). Além disso, existem sistemas que apresentam ajuste no antecedente, conseqüente, e nos pesos de cada uma das regras fuzzy.

### 4.3.2. Modelos de Sistemas Neuro-Fuzzy

Os modelos dos SNF são determinados pela modelagem das características de seus “genitores”. Nesta seção apresentam-se brevemente alguns dos SNF mais conhecidos: ANFIS, FSOM, NEFCLASS, NEFCON e NFHQ. Com isto, espera-se tornar mais clara a compreensão de um SNF qualquer.

#### 4.3.2.1. Adaptive Network based Fuzzy Inference System (ANFIS)

O sistema Neuro-Fuzzy ANFIS foi criado por Roger Jang. Esta arquitetura foi usada com sucesso em aplicações de previsão e aproximação de funções. Além disso, o autor propõe algumas variações do modelo para outras aplicações, aumentando sua popularidade ao ponto de ser inserido no *software* MATLAB®. A Figura 4.8 ilustra a arquitetura ANFIS.

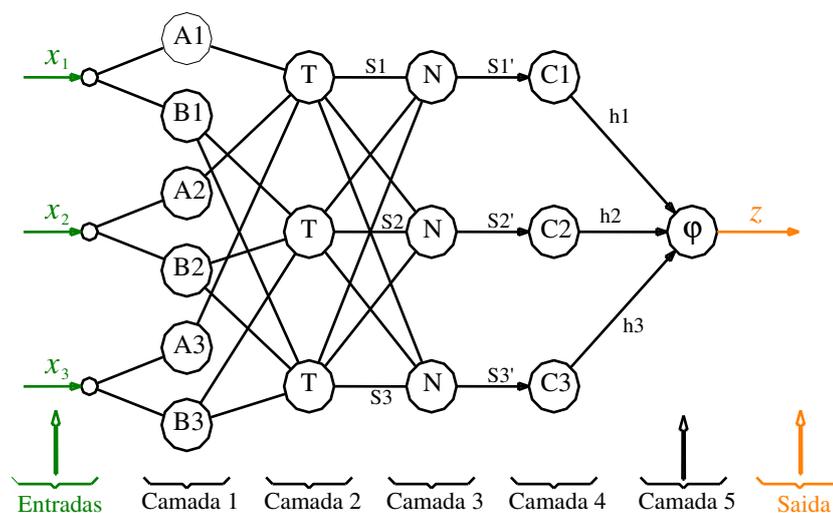


Figura 4.8. Arquitetura ANFIS.

Na figura acima, os nós que compõem uma mesma camada têm funções semelhantes, as quais são descritas a seguir.

- *Camada 1:* A saída desta camada são os graus de pertinências das entradas, baseado na premissa de cada regra. Neste caso, cada entrada apenas tem duas funções de pertinência ( $A_i = Alto$  e  $B_i = Baixo$ ), podendo ser este número maior.
- *Camada 2:* Nesta camada, calcula-se o grau de pertinência ao qual é submetido o conseqüente de cada regra. Cada nó ou neurônio desta camada executa a operação de  $t\_norm$  e corresponde a uma regra:

$$\begin{aligned} S_1 &= A_1(x_1) * A_2(x_2) * A_3(x_3) \\ S_2 &= B_1(x_1) * B_2(x_2) * A_3(x_3) \\ S_3 &= B_1(x_1) * B_2(x_2) * B_3(x_3) \end{aligned} \quad (4.1)$$

onde “\*” representa a  $t\_norm$ .

- *Camada 3:* Esta camada realiza a normalização dos graus de ativação das regras. Cada nó desta camada executa a função normalização, a qual é utilizada como um pré-processamento para a defuzzificação:

$$\begin{aligned} S'_1 &= S_1 / (S_1 + S_2 + S_3) \\ S'_2 &= S_2 / (S_1 + S_2 + S_3) \\ S'_3 &= S_3 / (S_1 + S_2 + S_3) \end{aligned} \quad (4.2)$$

- *Camada 4:* Nesta camada, a saída de cada neurônio é calculada pelo produto da saída normalizada da camada anterior e o grau de ativação do conseqüente. Este valor de saída é dado por:

$$\begin{aligned} h_1 &= S'_1 \cdot C_1 \\ h_2 &= S'_2 \cdot C_2 \\ h_3 &= S'_3 \cdot C_3 \end{aligned} \quad (4.3)$$

onde  $C_i$ 's correspondem aos valores dos conseqüentes, os quais, por exemplo, podem ser funções de pertinência “*singlenton's*”.

- *Camada 5*: A saída desta camada fornece a saída precisa do sistema ANFIS. Esta etapa de defuzzificação facilita-se pelo cálculo das camadas 3 e 4, dado por:

$$Z = \frac{\sum S_i \cdot f_i}{\sum S_i} = \sum S'_i \cdot f_i \quad (4.4)$$

$$Z = h_1 + h_2 + h_3$$

O espaço de particionamento das *E/S* utilizadas é do tipo *Fuzzy Grid Adaptativo*. Seu processo de aprendizado do sistema para a identificação da estrutura e parâmetros é feito em duas etapas, repetidas até atingir o critério de parada:

- *Etapa 1*: Os parâmetros dos conseqüentes são ajustados pelo método de *MQO* (*Mínimos Quadrados Ordinários*), enquanto os antecedentes permanecem fixos.
- *Etapa 2*: Os parâmetros dos antecedentes são ajustados pelo método de *GD* (*Gradient Decrescent*), enquanto os conseqüentes se mantêm fixos.

A idéia principal do sistema ANFIS é implementar um sistema fuzzy numa rede neural, onde geralmente as funções de pertinência utilizadas são do tipo sigmóides.

#### 4.3.2.2. Neuro Fuzzy Classification (NEFCLASS)

Este modelo foi proposto por *Nauck* e *Kruse*, e é aplicado com sucesso basicamente em problemas de classificação. Além disso, *Nauck* desenvolveu outras duas arquiteturas, o modelo *Neuro-Fuzzy Control (NEFCON)* para aplicações em controle e o modelo *Neuro-Fuzzy Function Approximation (NEFPROX)* para aplicações em previsão e aproximação de funções. Todos estes modelos utilizam um modelo genérico denominado *Fuzzy Perceptron*. A Figura 4.9 ilustra uma arquitetura *NEFCLASS*.

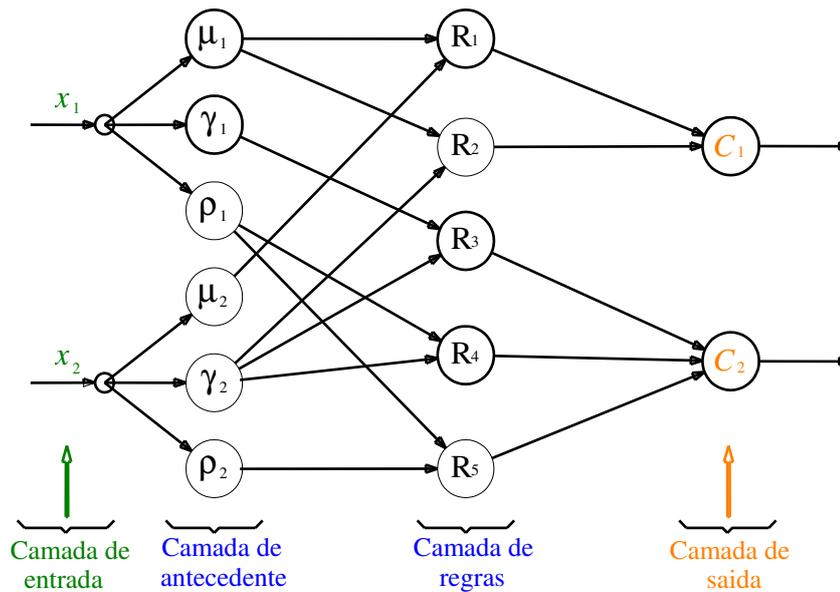


Figura 4.9. Arquitetura NEFCLASS.

Estes tipos de arquiteturas estão constituídas de 4 camadas e são descritas conforme se segue:

- *Camada de Entrada:* Esta camada só tem a função de direcionar os valores das entradas ( $x_1$  e  $x_2$ ) para as entradas das funções de pertinência dos antecedentes das regras. Estas regras fuzzy têm a forma:

**Se  $x_1 \in \mu_1$  e  $x_2 \in \mu_2$  então Padrão  $(x_1, x_2)$  pertence à classe  $i$ .**

- *Camada de Antecedentes:* Esta camada fornece o grau de pertinência dos antecedentes de cada regra, e cada variável de entrada foi dividida em três funções de pertinência (*Baixo, Médio, Alto*). O particionamento do espaço de entrada utilizado nesta camada é “*Adaptative Fuzzy-Grid*”.
- *Camada de Regras:* Nesta camada, executam-se a operação t-norm entre os graus de pertinência dos antecedentes de cada regra, gerando assim seu grau de ativação.
- *Camada de Saída:* Esta camada fornece a saída do sistema, obtida executando a operação t-conorm com os graus de ativação da camada de regras. Os pesos de interligação entre a camada de regras e a camada de saída são unitários.

O aprendizado deste modelo é feito em duas etapas:

- *Etapa 1:* Na primeira etapa, utiliza-se um algoritmo de aprendizado da base de regras, que pode ser construída de duas formas: inicializada a partir de um conhecimento prévio sobre o conjunto de padrões e depois refinada por aprendizado acrescentando ou diminuindo regras ou inicializada com um conjunto vazio de regras e acrescida por aprendizado incremental.
- *Etapa 2:* Após a criação da base de regras, nesta fase executa-se um algoritmo de aprendizado baseado no *GD* (*Gradient Descent*) para ajustar os parâmetros das funções de pertinências dos antecedentes das regras.

O sistema híbrido *NEFCLASS* é uma rede *Fuzzy-Perceptron* que apresenta a vantagem de interpretação da estrutura em forma de regras, e herda todas as características das *ANN* do tipo *Perceptron Multicamadas* (*MLP*).

#### 4.3.2.3. Fuzzy Self-Organized Map (FSOM)

Este modelo foi desenvolvido por *Vuorimaa*, possui uma estrutura muito semelhante ao modelo *ANFIS*, como se ilustra na Fig. 4.10. Este modelo utiliza particionamento *Fuzzy-Box* no espaço de entrada.

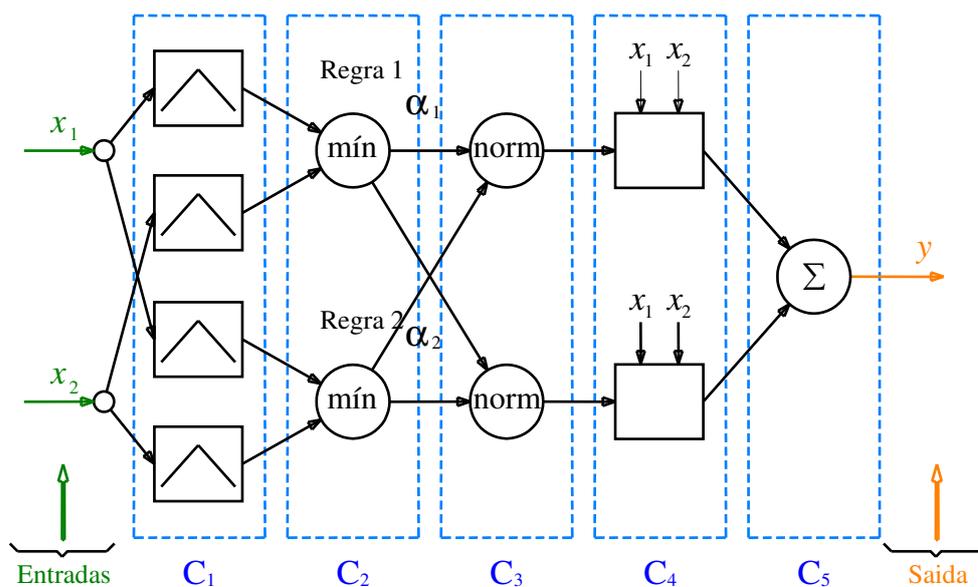


Figura 4.10. Arquitetura FSOM.

A descrição das camadas é semelhante ao modelo *ANFIS*, à exceção de

alguns detalhes descritos abaixo.

Na *Camada 1*, as funções de pertinência dos antecedentes são de formato triangular, e o formato da regras fuzzy são

$$\text{Se } x_1 \in U_{i,1} \text{ e } x_2 \in U_{i,2} \text{ então } y = S_i.$$

onde  $x_1 \in U_{i,1}$  é o grau de pertinência da entrada  $x_j \in U_{i,j}$  no conjunto fuzzy  $U_{i,j}$ .

Na camada 2, a operação *t-norm* para o cálculo do grau de pertinência das regras é

$$\alpha_i = \min\{\mu_{U_{i,1}}(x_1) + \mu_{U_{i,2}}(x_2)\} \quad (4.5)$$

onde  $\alpha$  é o grau de pertinência calculado para cada regra.

Na camada 5, o cálculo da defuzzificação é feito utilizando o método da média ponderada, e é dado por:

$$y = \frac{\sum \alpha_i \cdot S_i}{\sum \alpha_i} = \sum \alpha'_i \cdot S_i \quad (4.6)$$

onde  $S_i$  é o grau de ativação das funções de pertinência do conseqüente de cada regra.

O processo de aprendizado deste modelo é feito em três fases.

- *Etapa 1:* Os parâmetros dos centros  $C_i$  das funções de pertinência triangulares são auto-organizados utilizando o algoritmo *SOM* de Kohonen.
- *Etapa 2:* Os parâmetros dos conjuntos fuzzy são formados em volta aos centros obtidos na etapa 1, usando-se uma largura  $\omega_0$ .
- *Etapa 3:* Nesta etapa, são ajustados os parâmetros dos antecedentes por um algoritmo supervisionado semelhante ao LVQ, o qual também é gerado por Kohonen.

Esta arquitetura pode ser usada na implementação de sistemas para aproximação de funções e controle.

## 4.4. Vantagens e Desvantagens dos *SNF*

Algumas das principais vantagens e desvantagens apresentadas pelos sistemas híbridos em aplicações práticas são:

### 4.4.1. Vantagens das *SNF*

- Os *SNF* combinam as vantagens dos sistemas fuzzy e *RNA*, mostrando um enorme potencial para aplicações que combinem conhecimento qualitativo com robustez.
- A lógica fuzzy provê uma interface de alto nível, de rápida computação e amigável para programar, permitindo que o especialista se concentre nos objetivos funcionais em vez dos detalhes matemáticos. Por outro lado, as *ANN* são convenientes para a extração de conhecimento através do aprendizado.
- O projetista não precisa ter conhecimento prévio do processo, levando a uma fácil adaptabilidade aos diferentes processos.
- Os modelos Neuro-Fuzzy, podem lidar de uma maneira melhor que as *ANN* ao problema de ruído nos dados.
- Capacidade de auto-aprendizado, auto-organização e auto-direcionamento, imitando a capacidade humana do processo de tomada de decisão.

### 4.4.2. Desvantagens dos *SNF*

Algumas das limitações dos *SNF* são:

- Os *SNF* trabalham com um reduzido número de entradas, devido ao problema de explosão combinatória das regras. Portanto, um sistema com muitas entradas demanda um maior esforço computacional.
- Limitação na construção de sua própria estrutura, devido à estrutura fixa; quando há a capacidade de alterar sua estrutura, são limitados pelo elevado número de regras.

No próximo capítulo, é apresentada uma breve descrição da modelagem do sistema servo-hidráulico de interesse desta dissertação, e o projeto do controle por aprendizado acelerado proposto.